CIT **832**
OPERATING SYSTEM CONCEPT AND NETWORKING MANAGEMENT

NATIONAL OPEN UNIVERSITY OF NIGERIA

COURSE GUIDE

CIT 832
OPERATING SYSTEM CONCEPT AND NETWORKING MANAGEMENT

Course Adapter/Coordinator Afolorunso, A. A.
National Open University of Nigeria

NATIONAL OPEN UNIVERSITY OF NIGERIA

National Open University of Nigeria
Headquarters
14/16 Ahmadu Bello Way
Victoria Island
Lagos

Abuja Office
No. 5 Dar es Salaam Street
Off Aminu Kano Crescent
Wuse II, Abuja
Nigeria

e-mail: centralinfo@nou.edu.ng
URL: www.nou.edu.ng

Published by
National Open University of Nigeria

Printed 2008

ISBN: 978-058-102-2

CONTENTS **PAGE**

Introduction

CIT 832 -Operating System Concepts and Networking Management is a three credit unit course of seventeen units. It deals with concepts, structures, features and design mechanism and trends of operating system. The operating system has seen consistent innovations and developments like other fields of computer science. In this course we have made efforts to capture these changes. The trend is towards GUI-based, free platform, independent, secure and network-based operating system. Linux and Windows 2000 have got a very wide coverage in the course. Security and network management are the hot topics today. They are part of modem operating system design. Therefore these topics have also been taken up. The course is divided into four modules.

Module 1 introduces the concept of GUI, fundamental principles of operating system and computer network. The block lays the foundation for the understanding of the subsequent modules. The principles of operating system remain the same although the architecture of the operating system may vary.

Module 2 is about the Linux environment. The first unit is the design principles of Linux and the remaining units detail concepts like hierarchical directory structure, how to change your password, how to combine components together to make useful function, etc. the final unit deals with Linux administration which includes how to boot the operating system, how to maintain and use account and take a back-up.

Module 3 describes Windows 2000. It starts with the architecture and network support. It further elaborates on Windows 2000 client and server architecture, how to log on to network, how to browse network resources and use Windows explorer. Unit 3 is an advanced feature of Windows 2000 networking and finally the unit ends with Windows XP networking.

Module 4 has its focus on Security and network management features of operating system. There is a difference between network security and computer security. The first unit introduces several concepts: the goas of computer security, security problems, threats and vulnerabilities, intrusion detection, cryptography and classification. Unit 2 finally take up Security and management issues pertaining to operating system.

The aim of this course is to equip you with the basic skills of studying and understanding a system as well as lay the foundation of the basic knowledge and tools you need to become a proficient network administrator/manager. By the end of the course, you should be able to confidently work in any networking environment especially 011, either Linux or Windows operating system.

This Course Guide gives you a brief overview of the course content, course duration, and course materials.

What You Will Learn in this Course

The main purpose of this course is to provide the necessary administering computer network as well as working proficiently in a networking environment. It makes available the steps and tools that will enable you to make proper and accurate decision about operating systems whenever the need arises. This, we intend to achieve through the following:

Course Aims

i. Introduce the concepts associated with operating system and their design considerations;
ii. Provide necessary tools for choosing operating system for certain environment;
iii. Expose you to Networking concepts such as network topologies, transmission media, connecting devices such as hubs, repeaters, routers, gateways, etc; and
iv. Expose the relationship between operating systems and computer networks.

Course Objectives

Certain objectives have been set out to ensure that the course achieve its aims. Apart from the course objectives, every unit of this course has set objectives. In the course of the study, you will need to confirm at the end of each unit, if you have met the objectives set at the beginning of each unit. By the end of this course you should be able to:

i. identify important features of several GUIs;
ii. describe stages of evolution of operating systems, classify different types of OS as well as compare different approaches to OS design;
iii. explain the need for computer network and their application, describe various network topologies and how de' connected through repeaters, bridges, routers, gateways, etc.;
iv. understand the functioning of a large number of protocols;
v. connect with other users in Linux through write Command and emailing as well as configure a machine as a domain name server and a network file server;
vi. describe the concept of distributed file system and how a network printer can be managed
vii. explain the concepts of security in computer and networking systems.

Working through this Course

In order to have a thorough understanding of the course units, you will need to read and understand the contents, practise what you have learnt by studying the network of your organization or proposing one if there is none in existence and be committed to learning and implementing your knowledge.

This course is designed to cover approximately sixteen weeks, and it will require your devoted attention. You should do the exercises in the Tutor-Marked Assignments and submit to your tutors.

Course Materials

These include:

1. Course Guide 2. Study Units
3. Recommended Texts
4. A file for your assignments and for records to monitor your progress.

Study Units

There are seventeen study units in this course:

Module 1

Unit 1 Graphical User Interface
Unit 2 Introduction to Operating Systems
Unit 3 Introduction to Networking Concepts
Unit 4 Internetworking: Concepts, Architecture and Protocols

Module 2

Unit 1 Introduction to Linux Operating System
Unit 2 Linux Commands and Utilities
Unit 3 Linux Utilities and Editor
Unit 4 User to User Communication
Unit 5 Unix System Administration

Module 3

Unit 1 Windows 2000 Networking
Unit 2 Managing Windows 2000 Server
Unit 3 Advanced Windows 2000 Networking
Unit 4 Windows XP Networking

Module 4

Unit 1 Security Concepts
Unit 2 Computer Security
Unit 3 Security Management I
Unit 4 Security Management II

Make use of the course materials and do the exercises to enhance your learning.

Textbooks and References

Communication of ACM, April 1993.

Object Orientation: Concepts, Languages, Databases, User Intefaces, Kohsafian, Setrag & Razmik Abnours, New York; Wiley & Sons, 1990.

Operating Systems. Design and Implementation, Tanenbaum, Andrew S.; Woodhull, Albert S. (2006). Upper Saddle River, N.J.: Pearson/Prentice Hall. ISBN0-13-142938-8.

Modem Operating Systems, Tanenbaum, Andrew S. (2001). Upper Saddle River, N.J.: Prentice Hall. ISBN0-13-092641-8.

Operating Systems Concepts by Abraham Silberschatz and James L. Peterson, Addison Wesley.

Computer Network, Tanenbaum, Third Edition, Prentice-Hall 1996.

Data and Computer Communications, William Stalins, Fourth Edition, Macmillan, 1994.

Data Communication and Networkings, Behrouz A. Forouzan 2nd Edition, TMH 2000.

Internetworking with TCPIIP, Gouglas Comer, Volume 1, Fourth Edition, Prentice Hall, 2000.

Computer Networking, Kurose and Ross, Second Edition, Addison-Wesley, 2003.

Computer Network: A Systems Approach, Peterson & Davies, Morgan Kaufmann, Second Edition, 2000 W.

TCP/IP Illustrated, Volume 1, the Protocols, Richard Stevens, Addison-Wesley, 1994.

OSI: A Model for Computer Communication Standards, Uyless Black, Prentice Hall, 1991.

Data Networks, Dimitri Bertsekas and Robert Gallager, Second Edition, Prentice Hall, 1992.

http://www.redhat.com/docs/manuals/linux

http://www.linux.org

www.microsoft.com

White paper for distributed systems at www.microsoft.com

Survey of Operating System, John Holcombe & Charles Holcombe, Tata McGraw Hill.

Cryptography and Network Security, Principles and Practice, William Stallings -SE, PE.

RSA Security's Official Guide to Cryptography, Steve Burnett and Stephen Paine -RSA Press.

Security in Computer, Charles P. Ptleeger and Shari Lawrence Ptleeger, Third Edition, Pearson Education.

Security Bulletins: http://www.microsoft.com/technetisecurilY/

Service Pack: http://www.microsoft.com/windows2000/ downloads/ serviceRacks/

Hotfixes: http://www.microsoft.com/windows2000/downloads/critical/

Microsoft Windows Security: http://www.microsoft.com/securilY

Windows 2000 Commands byAleenFrisch.

Windows 2000 Professional Resource Kit, Microsoft Press.

Assignments File

These are of two types: the Self-Assessment Exercises and the Tutor-Marked Assignments. The self-assessment exercises will enable you monitor your performance by yourself, while the Tutor-Marked Assignment is a supervised assignment. The assignments take a certain percentage of your total score in this course. The Tutor-Marked Assignments w ill be assessed by your tutor within a specified period.

The examination at the end of this course will aim at determining the level of mastery of the subject matter. This course includes seventeen Tutor-Marked Assignments and each must be done and submitted accordingly. Your best scores however, will be recorded for you. Be sure to send these assignments to your tutor before the deadline to avoid loss of marks.

Presentation Schedule

The *Presentation Schedule* included in your course materials gives you the important dates for the completion of tutor marked assignments and attending tutorials. Remember, you are required to submit all your assignments by the due date. You should guard against lagging behind in your work.

Assessment

There are two aspects to the assessment of the course. First is the Tutor-Marked Assignments; second, is a written examination.

In tackling the assignments, you are expected to apply information knowledge acquired during this course. The assignments must be submitted to your tutor for formal assessment in accordance with deadlines stated in the Assignment File. The work you submit to your tutor for assessment will count for 30% of your total course mark.

At the end of the course, you will need to sit for a final three-hour examination. This will also count for 70% of your total course mark.

Tutor-Marked Assignment

There are seventeen tutor-marked assignments in this course. You need to submit all the assignments. The total marks for the best four (4) assignments will be 30% of your total course mark.

Assignment questions for the units in this course are contained in the Assignment File. You should be able to complete your assignments from the information and materials contained in your set textbooks, reading and study units. However, you may wish to use other references to broaden your viewpoint and provide a deeper understanding of the subject.

When you have completed each assignment, send it together with form to your tutor. Make sure that each assignment reaches your tutor on or before the deadline given. If, however, you cannot complete your work on time, contact your tutor before the assignment is due to discuss the possibility of an extension.

Final Examination and Grading

The final examination for the course will carry 70% of the total mark available for this course. The examination will cover every aspect of the course, so you are advised to revise all your corrected assignments before the examination.

This course endows you with the status of a teacher and that of a learner. This means that you teach yourself and that you learn, as your learning capabilities would allow. It also means that you are in a better position to determine and to ascertain the what, the how, and the when of course learning. No teacher imposes any method of learning on you.

The course units are similarly designed with the introduction, table of contents, and then a set of objectives. The objectives guide you as you go through the units to ascertain your knowledge of the required terms and expressions.

Course Marking Scheme

This table shows how the actual course marking is broken down.

Assessment	Marks
Assignment 1- 4	Four assignments, best three marks of the four count as 30% of course marks
Final Examination	70% of overall course marks Total 100% of course marks

Table 1: Course Marking Scheme

Course Overview

Unit	Title of Work	Weeks Activity	Assessment (End of Unit)
	Course Guide	Week 1	
	Module 1		
1	Graphical User Interface	Week 1	Assignment 1
2	Introduction to Operating Systems	Week 2	Assignment 2
3	Introduction to Networking Concepts	Week 3	Assignment 3
4	Internetworking: Concepts, Architecture and Protocols	Week 4	Assignment 4
	Module 2		
1	Introduction to Linux Operating System	Week 5	Assignment 5
2	Linux Commands and Utilities	Week 6	Assignment 6
3	Linux Utilities and Editor	Week 7	Assignment 7
4	User to User Communication	Week 8	Assignment 8
5	Unix System Administration	Week 9	Assignment 9
	Module 3		
1	Windows 2000 Networking	Week 10	Assignment 10
2	Managing Windows 2000 Server	Week 10	Assignment 11
3	Advanced Windows 2000 Networking	Week 11	Assignment 12
4	Windows XP Networking	Week 11	Assignment 13
	Module 4		
1	Security Concepts	Week 12	Assignment 14
2	Computer Security	Week 13	Assignment 15
3	Security Management I	Week 14	Assignment 16
4	Security Management II	Week 15	Assignment 17
	Revision	Week 16	
	Examination	Week 17	
	Total	**17 weeks**	

How to Get the Best from this Course

In distance learning the study units replace the university lecturer. This is one of the great advantages of distance learning; you can read and work through specially designed study materials at your own pace, and at a time and place that suit you best. Think of it as reading the lecture instead of listening to a lecturer. In the same way that a lecturer might set you some reading to do, the study units tell you when to read your set books or other material. Just as a lecturer might give you an in-class exercise, your study units provide exercises for you to do at appropriate points.

Each of the study units follows a common format. The first item is an introduction to the subject matter of the unit and how a particular unit is integrated with the other units and the course as a whole. Next is a set of learning objectives. These objectives enable you know what you should be able to do by the time you have completed the unit. You should use these objectives to guide your study. When you have finished the units you must go back and check whether you have achieved the objectives. If you make a habit of doing this you will significantly improve your chances of passing the course.

Remember that your tutor's job is to assist you. When you need help, don't hesitate to call and ask your tutor to provide it.

1. Read this *Course Guide* thoroughly.

2. Organize a study schedule. Refer to the 'Course Overview' for more details. Note the time you are expected to spend on each unit and how the assignments relate to the units. Whatever method you chose to use, you should decide on it and write in your own dates for working on each unit.

3. Once you have created your own study schedule, do everything you can to stick to it. The major reason that students fail is that they lag behind in their course work.

4. Turn to *Unit* 1 and read the introduction and the objectives for the unit.

5. Assemble the study materials. Information about what you need for a unit is given in the 'Overview' at the beginning of each unit. You will almost always need both the study unit you are working on and one of your set of books on your desk at the same time.

6. Work through the units. The content of a unit itself has been arranged to provide a sequence for you to follow. As you work through the unit you will be instructed to read sections from your set books or other articles. Use the unit to guide your reading.

7. Review the objectives for each study unit to confirm that you have achieved them. If you feel unsure about any of the objectives, review the study material or consult your tutor.

8. When you are confident that you have achieved a unit's objectives, you can then start on the next unit. Proceed unit by unit through the course and try to pace your study so that you keep yourself on schedule.

9. When you have submitted an assignment to your tutor for marking, do not wait for its return before starting on the next unit. Keep to your schedule. When the assignment is returned, pay particular attention to your tutor's comments, both on the tutor-marked assignment form and also on his comments on the assignment. Consult your tutor as soon as possible if you have any questions or problems.

10. After completing the last unit, review the course and prepare yourself for the final examination. Check that you have achieved the unit objectives (listed at the beginning of each unit) and the course objectives (listed in this *Course Guide).*

Facilitators/Tutors and Tutorials

There are 12 hours of tutorials provided in support of this course. You will be notified of the dates, times and location of these tutorials, together with the name and phone number of your tutor, as soon as you are allocated a tutorial group.

Your tutor will mark and comment on your assignments, keep a close watch on your progress and on any difficulties you might encounter and provide assistance to you during the course. You must mail or submit your tutor-marked assignments to your tutor well before the due date (at least two working days are required). They will be marked by your tutor and returned to you as soon as possible.

Do not hesitate to contact your tutor by telephone, or e-mail if you need help. The following might be circumstances in which you would find help necessary. Contact your tutor if:

1you do not understand any part of the study units or the assigned readings,
2you have difficulty with the self-tests or exercises,
3you have a question or problem with an assignment, with your tutor's comments on an assignment or with the grading of an assignment.

You should try your best to attend the tutorials. This is the only chance to have face-to-face contact with your tutor and to ask questions which are answered instantly. You can raise any problem encountered in the course of your study. To gain the maximum benefit from course tutorials, prepare a question list before attending them. You will learn a lot from participating in discussions actively.

Summary

Operating System Concepts and Networking Management introduces you to the basic principles and concepts of operating systems design and functionalities with particular references to Linux and Windows 2000 operating systems as well as concepts of computer network and internetworking. The skills you need to understand the basics of operating system and computer networks, etc. are intended to be acquired in this course. The content of the course material was planned and written to ensure that you acquire the proper knowledge and skills for the appropriate situations. Real-life situations have been created to enable you identify with and create some of your own. The essence is to get you to acquire the necessary knowledge and competence, and by equipping you with the necessary tools, we hope to have achieved that.

We wish you success in the course and hope that you will find it both interesting and useful.

Course Code	CIT 832
Course Title	Operating System Concept and Networking Management
Course Adapter/Coordinator	Afolorunso, A. A. National Open University of Nigeria

NATIONAL OPEN UNIVERSITY OF NIGERIA

National Open University of Nigeria
Headquarters
14/16 Ahmadu Bello Way
Victoria Island
Lagos

Abuja Office
No. 5 Dar es Salaam Street
Off Aminu Kano Crescent
Wuse II, Abuja
Nigeria

e-mail: centralinfo@nou.edu.ng
URL: www.nou.edu.ng

Published by
National Open University of Nigeria

Printed 2008

ISBN: 978-058-102-2

CONTENTS PAGE

MODULE 1 OPERATING SYSTEM FUNDAMENTALS & NETWORKING

UNIT 1 GRAPHICAL USER INTERFACE

CONTENTS

1.0 INTRODUCTION

The Graphical User Interface (GUI) is one of the most revolutionary changes to occur in the evolution of the modem computing system. The interaction between man and computer in the design of operating system has changed from a character-oriented system to the now more graphics-oriented system. This revolution has increased the accessibility and usability of computer systems to the general public.

We will look at GUI features of the various operating systems with little knowledge of operating system or memorized commands. The software providing such features supports Modern User Interface concept such as desktop metaphor, which makes computers available to the majority of

people who are either novices or non-programmers. The personal computer was invented for these users.

We will start with the basic definition move on to basic components and finally look into some systems to find out what GUI facilities they are supporting.

2.0 OBJECTIVES

After going through this unit, you should be able to:

define what GUI is and how it is different from character oriented system
define all the terms related with GUI
identify important features of several GUIs.

3.0 MAIN CONTENT

3.1 What is Graphical User Interface?

The term "user interface" originated in the engineering environment in the late 1970s. Virtually everyone who interacted directly with computers had been an engineer and a programmer, but new kinds of users were emerging: the non-programming user. These users often reacted more negatively to difficulties in dealing with a machine. New forms of interaction were needed, new interfaces, were required, attention flowed to "the user interface".

With the introduction of the Macintosh in 1984, Apple Computer popularised the user interface as it is known today. Apple's user interface is now commonly referred to as a Graphical User Interface or GUI. The GUI has become associated with a common feature set available in a number of product offerings. Common features include:

Secondary user-input devices. Usually a pointing device and typically a **mouse**.
Point and shoot functionality with **screen menus** that **appear** or **disappear** under pointing-device-control.
Icons that represent files, directories and other application and system entities.
Dialog boxes, button, sliders, check boxes and many other **graphical** metaphors that let the programmer and user tell the computer what to do and how to do it.

Today's GUIs have expanded basic functionalities to support not only graphics, but also dimensions, color, height, video, and highly dynamic

interaction. Modem user interfaces can simulate a very realistic view of a real, three-dimensional world.

3.2 Evolution of Human and Machine Interaction

The primary means of communication with computers earlier had been through command- based interfaces. In command interfaces, users have to learn a large set of commands to get their job(s) done. In earlier computer systems paper tapes, cards and batch jobs were the primary means of communicating these commands to the computers. Later, time-sharing systems allowed the use of CRT terminals to interact/communicate with the r computer. These early systems were heavily burdened by users trying to share precious r computer resources such as CPU and peripherals.

The batch systems and time-sharing led to command-driven user interfaces. Users had to memorise commands and options or consult a large set of user manuals. The early mainframe and minicompute1" systems required a large set of instruction manuals on how to use the system. In some systems, meaningful terms were used for command names to help the end-user. But in other systems the end-user had to memorise several sequences of keystrokes to accomplish certain tasks.

Early users of computes were engineers and what we now call expert users; users who had a lot of interest in knowing more about computer systems and the technology. Command line interfaces were acceptable to the majority ofthes8 users. In the l970s, computers were introduced to a new class of users: secretaries, managers and non-technical people. These new users were less interested in learning computer technology and more interested in getting their jobs done through the machine. The command-based interfaces caused many of these new users to develop computer phobia. Imagine the thought of memorising commands made up of "Control-Alt-Del" to boot the system.

To make life easier for the end -user, a large collection of devices have been invented to control, monitor and display information. The early (and still widely used) peripherals are the keyboard and the video terminal. But, it was not until the late 70s, that research projects at some universities led to the invention of pointing devices and windowing systems. The mouse and joystick were among some of the few pointing devices that were invented in this period. Also, research pioneers invented the notion of splitting the screen to allow multiple windows and direct manipulation of objects.

In the 70s; researchers designed powerful new workstations armed with graphical user-interfaces. The basic assumption of these new

workstations was that one user could have a powerful desktop computer totally dedicated to that user's task. Thus, the computer is not only used to perform the task, but can also provide a much more intuitive and easy-to-use environment. In this unit we will examine the common GUIs.

3.3 Common Graphical User Interfaces

This section presents a list of terms used commonly with the graphical user interface (GUI). GUIs are systems that allow creation and manipulations of user interfaces employing windows, menus, icons, dialog boxes-mouse and keyboard. Macintosh toolbox, Microsoft Windows and X-Windows are some examples of GUIs.

1) Pointing Devices

Pointing devices allow users to point at different parts of the screen. Pointing devices can be used to invoke a command from a list of commands presented in a menu. They can also be used to manipulate objects on the screen by:

> Selecting objects on the screen Moving objects around the screen, or Merging several objects into another object.

Since the 960s, a diverse set of tools have been used as pointing devices including the light pen, joystick, touch sensitive screen and a mouse. The popularity of the mouse is due to the optimal coordination of hand and easier tracking of the cursor on the screen.

2) Pointer

A symbol that appears on the display screen and that you move to select object9and commands. Usually the pointer appears as a small angled arrow.

3) Bit-Mapped Displays

As memory chips get denser and cheaper, bit displays are replacing character-based display screens. Bit-mapped display made up of tiny dots (pixels) are independently addressable and much finer resolution than character displays. Bit-mapped displays have advantages over character displays. One of the major advantages is graphic manipulation capabilities for vector and raster graphics, which presents information in the final form on paper (also called WYSIWYG: What You See Is What You Get).

4) Windows

When a screen is split into several independent regions, each one is called a window. Several applications can display results simultaneously in different windows. *Figure* 1 presents a screen with two windows.

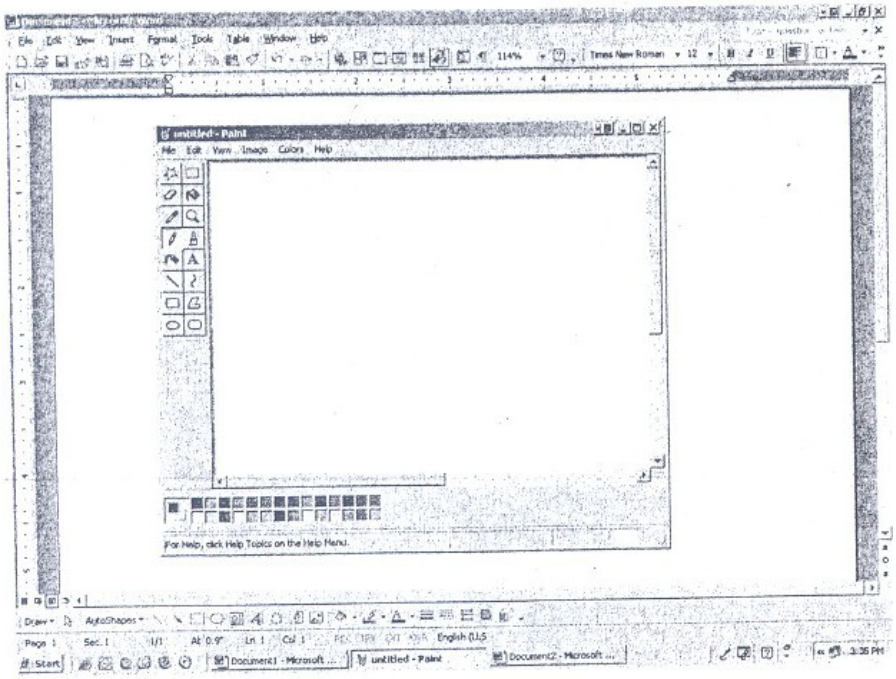


Figure 1: Screen with two windows

The end-user can switch from one application to another or share data between applications. Windowing systems have capabilities to display windows either **tiled or** over-lapped, *Figure* 2. Users can organise the screen by resizing the window or moving related windows closer.

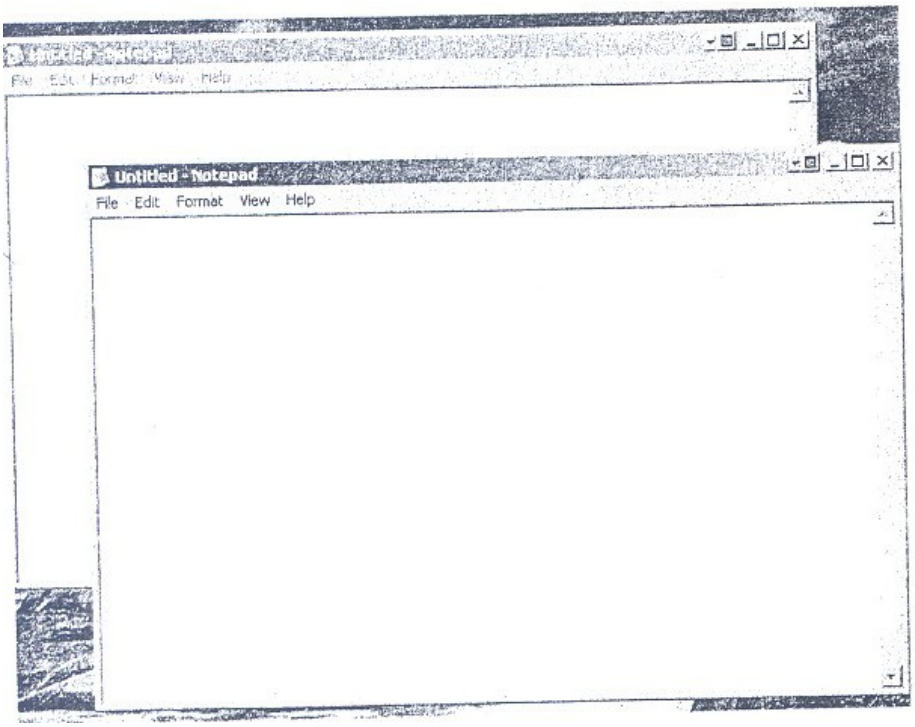


Figure 2: Overlapped Windows

5) Menus

A menu displays a list of commands available within an application *(Figure* 3). From this menu, the end-user can select operations such as File, Edit or Search. Instead of remembering commands at each stage, a menu can be used to provide a list of items. Each menu item can be either a word or an icon representing a command or a function. A menu item can be invoked by moving the cursor on the menu item and selecting the item by clicking the mouse.

Instead of memorising commands to each stage, the user selects a command from a menu bar displaying a list of available commands. For example, *Figure* 3 displays the menu bar. This menu bar displays a list of commands available such as File, Edit and the appropriate action is taken.

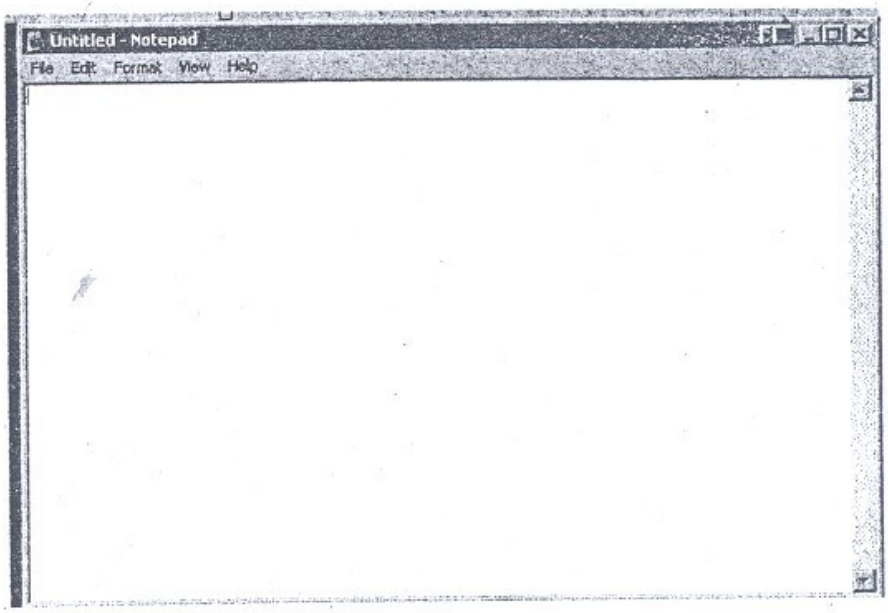


Figure 3: Menu Bar

When a menu item is invoked it could cause other menus, called **pull-down menus**, to appear. **Pull-down menus** *(Figure* 4) are used to present a group of related commands or options for a menu item. *Figure* 4 presents the File pull-down menus.

Pull-down and **pop-up** menus display option commands available for each selection. *Figure* 4 shows the pull-down menu displayed when the Character menu item is selected. The user can then select from different character styles.

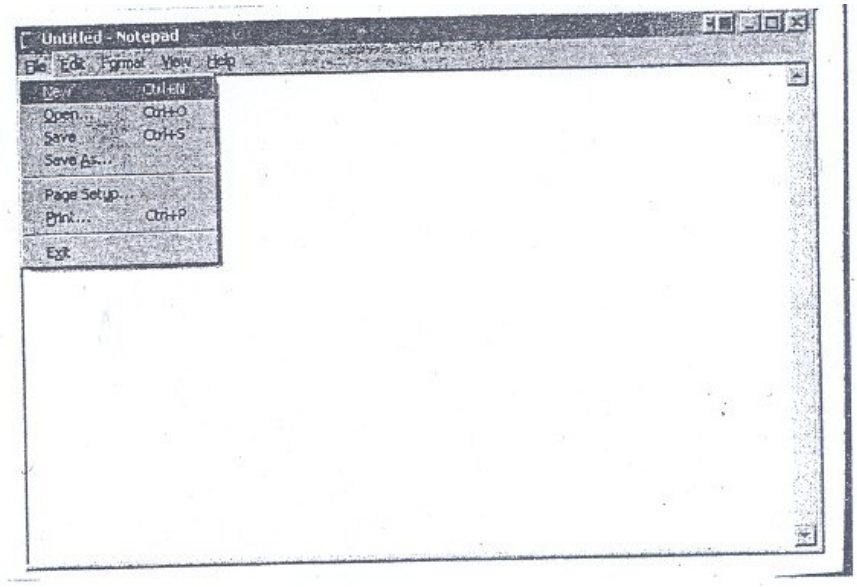


Figure 4: Pull-down Menu

6) Dialog boxes

Dialog boxes *(Figure* 5) allow more complex interaction between the user and the computer. Dialog boxes employ a collection of control objects such as dials, buttons, scroll bars and editable boxes. For example, in *Figure* 5, a dialog box is used to open a file.

In graphical user-interfaces, textual data is not only a form of interaction. Icons represent concepts such as file folders, wastebaskets, and printers. Icons symbolize words and concepts commonly applied in different situations. *Figure* 4 shows paint utility with its palette composed of icons. Each one of these icons represents a certain type of painting behaviour. Once the pencil icon is clicked, for example, the cursor can behave as a pencil to draw lines. Applications of icons to the user-interface design are still being explored in new computer systems and software such as the NeXT computer user interface.

Dialog boxes are primarily **used** to collect information from the user or to present information to the user.

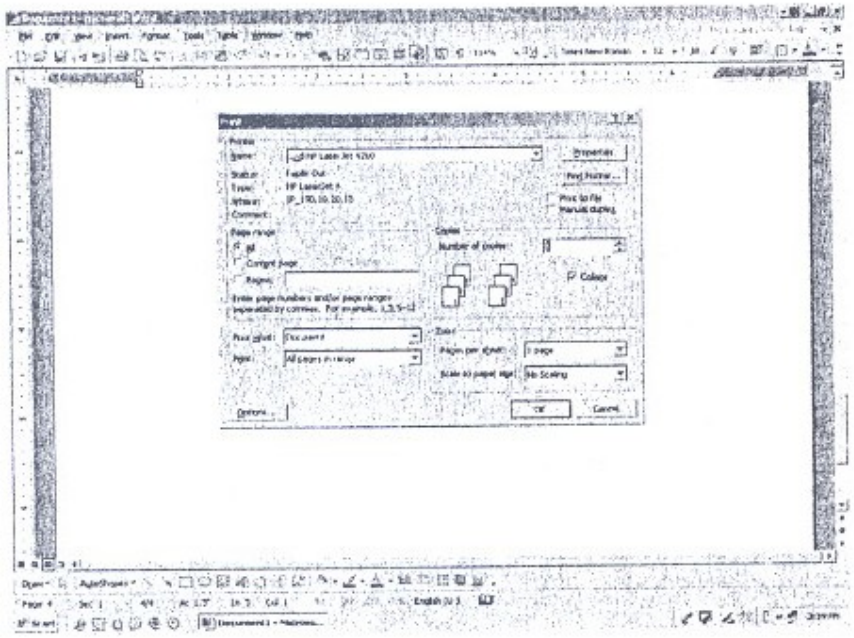
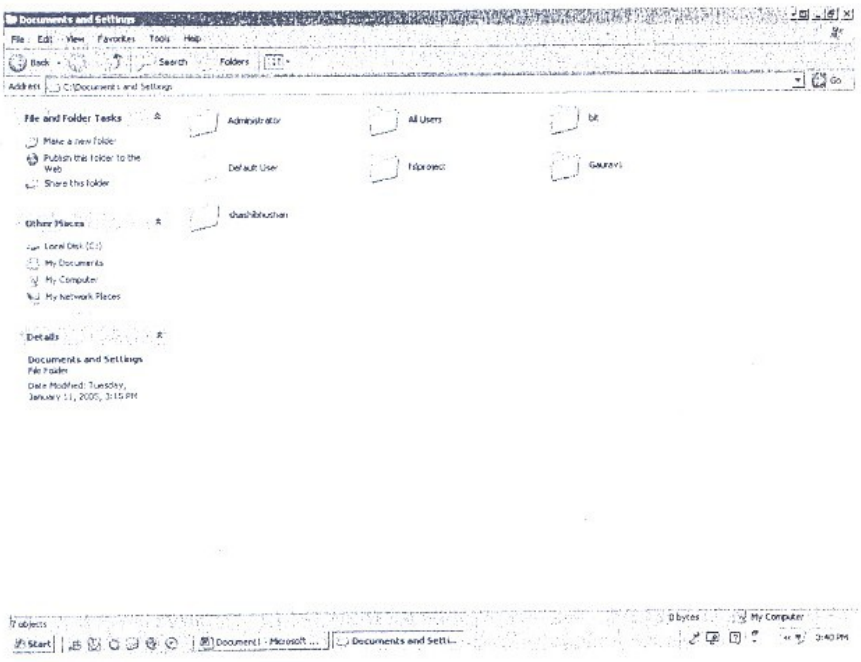


Figure 5: Dialog Box

Among the information obtained are the number of copies and page numbers to be printed. Dialog boxes are also used to indicate error message in the form of alert boxes. Dialog boxes use a wide range of screen control elements to communicate with the user.

7) Icons

Icons are used to provide a symbolic representation of any system/user-defined object such as file, folder, address, book, applications and so on. Different types of objects are represented by a specific type of icon. In some GUIs, documents representing folders are represented by a folder icon *(Figure* 6). A folder icon contains a group of files or other folder icons. Double clicking on the folder icon causes a window to be opened displaying a list of icons and folder icons representing the folder's contents.



D

Figure 6: Icons

8) Desktop Metaphor

The idea of metaphors has brought the computer closer to the natural environment of the end-user. The concept of physical metaphor paradigm, developed by Alan Kay, initiated most of the research for graphic user/interfaces based on a new programming approach called object-oriented programming. Discussion of this subject is beyond this unit. The physical metaphor is a way of saying that the visual displays of a computer system should present the images of real physical objects.

For example, the wastepaper basket icon can be used to discard objects from the system by simply dragging the unwanted objects into the dustbin, as in real life. The desktop metaphor probably has been the most famous paradigm. Because of the large set of potential office users,

this metaphor can have the most dramatic effect. In this paradigm, the computer presents information and objects as they would appear and behave in an office, using icons for folders, in-baskets, out-baskets and calendars.

Ina desktop metaphor; users are not aware of applications. Users deal with files, folders, drawers, a clipboard and an outbox. Instead of starting the word process and loading file, users merely open the report document, which implicitly invokes the word processor. Clicking the mouse on an icon representing the report causes the word processor to get started and to load the report file implicitly. Today, several computing environments provide this capability.

9) The 3D GUI

The desktop metaphor GUI is 2½ D. It is 2D because its visual elements are two- dimensional: they lie in the xy plane, are defined in 2D coordinates, are flat and contain only planar regions (areas). It is 2½ D because where visual elements overlap they obscure each other according to their priority. In a 3D GUI the visual elements are genuinely three-dimensional: they are situated in xyz space, are defined in terms of3D coordinates, need not be flat and may contain spatial regions (volumes).

The design considerations for a 3D GUI appear more complex than for a 2½ D GUI. To begin with, the issues of metaphor and elements arise afresh. The desktop metaphor with its windows, icons, menus and pointing device elements is firmly established for 2½ D GUIs. In contrast no clearly defined metaphor and set of elements for 3D GUIs are manifest -yet. 3D GUIs offer considerably more scope for metaphors than 2½ D GUIs; there are many metaphors which could be based on our physical 3D environment, including the obvious extension of the desktop metaphor into a 3D environment, including the obvious extension of the desktop metaphor into a 3D office metaphor. On the other hand, much more abstract metaphors are possible, such as one based "starmaps" where objects are simply placed somewhere in "cyberspace". Likewise the elements of a 3D GUI may resemble, or differ substantially from, the elements of the 2½ D GUI.

The various prototypes have been developed to design the same elements in the 3D GUI as in the 2½ D desktop GUI: windows, icons, menus, a general space in which to arrange the visual elements, a cursor and an input device to manipulate the cursor.

3.4 Functionality of Graphical User Interfaces

The development environment for most GUIs consists of four major components:

A windowing system,
An imaging model,
An application program interface (API), and
A set of tools and frameworks for creating interfaces and developing integrated applications.

Windowing systems allow programs to display multiple applications at the same time. Windowing systems include programming tools for building movable and resizeable windows, menus, dialog boxes and other items on the display. Some GUIs contain proprietary windowing systems, such as Macintosh. Others use common windowing systems such as X-window or simple X.

An imaging model defines how fonts and graphics are created on the screen. Imaging models handle, for example, typeface and size in a word processor and lines in a drawing program. This component of the system environment has taken on increasing sophistication as applications incorporate complex curves, color, shading and dimension.
Some GUIs support more than one imaging model.

The API is a set of programming language functions that allow the programmer to specify how the actual application will control the menus, scroll bars and icons that appear on the screen. Like windowing models, APIs align with particular GUIs.

Finally, GUI development environments can include toolkits and frameworks. Most of these toolkits are based on object-oriented approach.

Although the structure of the basic development for most GUIs is similar, there are major differences in how the GUI is integrated with the operating system. Some, like the Macintosh and NeXT GUIs, are closely integrated with the operating system.

Others, like X Window or Microsoft's Windows, can be set up as options to be selected by the user when the computer boots up.

Programming of software for GUIs across these components is fairly complex and challenging. Commercial developers who want to support multiple environments find their task further complicated by the absence

of standards across heterogeneous computing platforms. The higher-level toolkit component is intended to mitigate much of this difficulty. Although the graphical user interface has become a standard component of most systems, no standards in windowing systems, imaging models, APIs, or high-level toolkits have emerged. However, three major camps of GUIs dominate. The first camp is IBM's System Application Architecture (SAA), which includes primarily Microsoft's Windows and PM (Presentation Manager). The second camp is UNIX systems, usually built around X Window. The third camp is the Macintosh. In the next section we will describe the object-oriented functionality of representative GUIs in these camps, including Windows, X (upon which most of the UNIX camp is based), NeXT and Macintosh.

SELF ASSESSMENT EXERCISE 1

1) What is GUI and what are its features?
2) Define the features of the following:

   a) Window
   b) Pull-down menu c) Dialog box
   d) Pointing devices

3) What is the difference between Bitmapped and character based displays?

3.5 GUI Design Considerations: Psychological Factors_

There are some empirical studies that have identified basic psychological factors that one should consider in the design of a good GUI. In this section we focus our discussion to three primary contributing human factors, which are: the physical limits of visual acuity, the limits of absolute memory, and the Gestalt Principle.

Visual Acuity

Visual acuity is the ability of the eye to resolve detail. This refers to the amount of information one can put on the screen. The retina of the eye can only focus on a very small portion of a computer screen, or anything for that matter, at anyone time. This is because, at a distance greater than 2.5 degrees from the point of fixation, visual acuity decreases by half. Therefore, a circle of radius 2.5 degrees around the point of fixation is what the user can see clearly.

At a normal viewing distance of 19 inches, 5 degrees translates into about 1.7 inches. Assuming a standard screen format, 1.7 inches is an area about 14 characters wide and about 7 lines high. This is the amount

of information that a user can take in at any one time, and it limits the effective size of icons, menus, dialog boxes, etc. If the user must constantly move his eyes across the screen to clearly focus, the GUI design is causing a lot of unnecessary and tiring eye movement.

Information Limits

Once the user has a: desired fixation point, there is a limit to the amount of information that the person can process at one time. A GUI design rule of thumb is that the range of options or choices should never be more than five or six. It has been shown that absolute identification using one-dimensional criteria was about seven items, plus or minus two. Miller introduced the concept of recoding as a method that people use to store information. It has also been pointed out that by expanding the identification criteria from one to more dimensions people could handle more choices and remember more. Later researchers have expanded this work to develop the concept that people chuck, information together in order to remember more information. This research has a direct impact on GUI design, especially concerning the number of menu items and icons.

Gestalt Principle

The Gestalt Principle states that people use a top-down approach to organising data. This principle can influence how one should organise graphical information on the screen. The Gestalt school of GUI designers have attempted to identify criteria that cause people to group certain items together in a display. Proper grouping results in a necessary redundancy of selection information that aids the user. For example, if the user knows where one item in a group is on a screen, he will expect other like items to be there also. If one groups the items in line with this expectation, it allows for accurate locating and better transfer of information to the user.

The top-down approach also allows for the development of emergent features. An emergent feature is a global property of a set that is not evident when one views each item locally. Since global processing tends to be automatic, one can argue that an emerged feature reduces the attention demand as a user operates a multi-element display. For this performance enhancement, one must use the Gestalt Principle in the initial placement, and the resulting organisation must be compatible with the user's cognitive view of the task.

3.6 GUI Design Considerations: Standards

Considering the above psychological factors, one could come to the conclusion that one could easily extrapolate these factors to the design of a good GUI. Empirical studies of GUI show that this intuition is not always the case. It directly leads to the conclusion that a good GUI would use a lot of icons. Unfortunately, too many randomly placed icons violate the limits of absolute memory. Using the Gestalt Principle, one can group like items together using factors like color to add more informational dimensions. Too many colors, however, destroy the global visual grouping of the items. The user then begins to concentrate on the GUI. Any primary cognitive task attention devoted to the interface may interfere with the primary task. One can derive basic GUI standards from basic human factors, however. The standards are the presentation of information, the grouping of information, and information sequencing.

Amount of Information Presented

The amount of information to present is the most basic of Gill design considerations. H.E. Dunsmore showed that making screens less crowded improves screen clarity and readability. As such, GUI designers usually follow the guidance that the interface should display only what the user needs to perform the current operation. Empirical researchers show that limiting the information to that necessary for the user reduces errors and time to perform tasks. Errors and performance time increase as the GUI presents more information. Of course, it requires a thorough analysis of the tasks that the user must perform in order to display only the necessary amount of information.

Compared to a randomly placed screen, a well-designed screen can reduce the time needed to perform a task by as much as 40%. Ways to conserve screen space are:

1) **Appropriate use of abbreviations:** Many design documents recommend using complete words whenever possible. Due to screen sizing constraints, it is not always possible to use complete words. When complete words are not possible, abbreviations should be contextual and consistent. A good contextual example is "h," which is usually a good abbreviation to use for help. The number of abbreviations should not only be contextual but also be kept to the minimum. As a poor example, in the UNIX system, the "Is" command lists files in a directory. The "Is" command has 17 different one-letter abbreviations that change the output options of the "Is" command. The one-letter abbreviations have

little contextual link to the options they represent. In fact, the UNIX system is a good example of what not to do.

2) **Avoid unnecessary detail:** For example, use whole numbers if one does not need I decimals. Keep the window and icon designs clear and simple. Even when users prefer more complex icons, elaborate icons add nothing to performance. Studies show that when icon designs are too complex, time to complete a task actually increases. In studies with 3-D and 2-D graphical displays, users preferred the 3-D displays. There were no differences in performance between the two graphical displays, however.

3) **Use concise wording:** Screens have limited space. Screen designers should avoid the tendency to place additional data on the screen just because the data is available. More objective limits of screen density vary from the thresholds of 25% to 80%. There is no empirical research that substantiates any performance enhancement with any specific threshold.

4) **Use familiar data formats:** With more familiar formats, the user will need less information to complete the task. An example for data entry is the standard USA address format of street, city, state, and zip code. In addition to requiring less instruction, the user will perform the operation faster than if the format is unfamiliar.

5) **Use tabular formats with column headings:** Tabular formats allow for efficient labeling of related data. It is especially preferable for data location tasks. Simply splitting items on one long line into two lines result in productivity improvements of 20%.

Grouping of Information

Given a set of information to display, there are many ways one can display the information. Proper grouping improves the information's readability and can highlight relationships between the information.

There are several techniques to aid in the grouping of information, which include:

1) **Color:** Presenting different groups in different colors clearly creates some degree of grouping among the elements of the same color. GUIs that utilise color well increase productivity. If like color items are in close proximity, the visual association is

stronger than if the like color items are further apart. In addition to changing the item's colors, one can use different colors for the background and foreground. The effectiveness of this technique decreases as the number of screen colors increases. Overuse of color degrades performance, however.

2) **Graphical Boundaries:** Drawing boundaries around elements is the most common method of grouping elements in GUI. Although there is no empirical evidence to show that these groupings improve performance, users prefer this type of groupings compared to other methods. Another method of grouping is to group tasks within icons. Icon grouping is easy because many icons can have common attributes. Icons are also small and therefore use less space. Another advantage of icons is that recognition is faster for pictures than for text. This makes it easier for the novice to learn a system. Studies also show that icons have smaller error rates than textual interfaces and the same as for menu interfaces. Conversely though, empirical studies have shown that, counter intuitively, icons do not lead to greater increases in performance.

3) **Highlighting:** Besides color, there are several other methods of highlighting including reverse video, brightness, underlining, and flashing. The most common use of highlighting is reverse video to indicate an item that is currently selected. GUIs usually use brightness to show which items are not active at a given time. Underlining is effective if it does not interfere with the legibility of characters. Flashing will both get attention and annoy if the user cannot turn off the flashing. Therefore, one should use flashing only to convey an urgent need. Apple Macintosh uses flashing to signal only program or data destruction. Regardless of which type of highlighting, one needs to apply it conservatively. Overuse of highlighting causes confusion among users and defeats its purpose. Additionally, if one highlights the wrong information, the user has more difficulty detecting the important information.

Information Sequencing

One needs to layout a screen in a manner that allows the user to easily find any information on it. Most designers advocate the use of the de facto GUI screen standards. This is because many users now expect certain modes of operation in al1 GUIs. For example, most users expect the top of screen to contain the headings for the pul1-down menus. The top right is the default location for icons representing the disk

availability. In the Macintosh GUI, the bottom right contains the trash icons used for deleting files.

Within a window, there are also many standard modes. A window title is usually at the top. Scrol1 bars are on the right and bottom for vertical and horizontal window movement. A box for closing the window is at the top left. Icons for resizing the window are at the four comers.
Studies show that most users initial1y scan the screen starting at the upper-left comer. This comer should be the obvious starting point for applications invoked from within the window. This permits a left-to-right and top-to-bottom reading, which is standard for Western cultures.

The optimum sequence for screen presentations is a col1ection of various factors, including:

1) **Sequence of Use:** One needs to present the user the information in the order that I' the user wil1 probably utilise it.

2) **Conventional Usage:** If a common convention is in general usage, the GUI design should continue using it. For example, in the standard window layout, the file option is usual1y to the far left of the menu bar.

3) **Importance:** The designer needs to place the more important information at a prominent location. For example, if several entries are possible, the GUI should lead off with the required ones and end with the optional ones.

4) **Frequency of Use:** One should pace the most frequently utilised commands at the beginning. For example in a menu list, the most frequency utilised commands should be at the top of the list.

5) **Generality versus Specificity:** The more general items should precede the more "specific items, especial1y when there is a hierarchical relationship among the data.

6) **Alphabetical or Chronological:** If there is no other rules for ordering data element, then one should adopt some other technique such as an alphabetical or a temporal listing. Card [11] showed that selection time was faster for alphabetical than for any other functional grouping.

3.7 GUI Examples

The goal of any GUI is to allow the user to work through the computer and application to concentrate on the primary cognitive task. The user should not be concerned with the user interface. Any attention devoted to the interface interferes with the main task.

This section presents Graphical User Interfaces. Some popular GUIs will be covered to provide a broad picture of the subject. There are a great many popular GUIs around including X Windows, Microsoft Windows, NeXT's NeXTStep and others.

3.7.1 Microsoft Windows (MS-WINDOWS)

MS- Windows is the most popular GUI for IBM personal computers. IBM and Microsoft announced OS/2 as a new operating system for 80286 and 80386 personal computers. The OS/2 Standard Edition 1.1 adds Presentation Manager (PM) for its graphical user interface. The user interfaces of Windows and PM are very similar but their APIs are different. Microsoft's strategy is to use Windows as a springboard for PM.

Windows provides an environment that enhances OOS in many ways. The major benefits of Windows are:

1) **Common Look and Feel:** All windows applications have the same basic look and feel. Once you know one or two Windows applications, it is easy to learn another one. .

2) **Device Independence:** Windows presents a device-independent interface to applications. Unlike most of to day's OOS applications, a Windows application is not, bound to the 'underlying hardware such as mouse, key board or display. Windows shields the application from this responsibility. The application deals with tl1e Windows API to manipulate any underlying devices.

3) **Multitasking:** Windows provides non-preemptive multitasking support. Users can have several applications in progress at the same time. Each application can be active in a separate window.

4) **Memory Management**: Windows also provides memory management to break the 640K limitation of MS-DOS. An application has the ability to use the extended memory, share data segments with other applications and unwanted segments to disk.

5) **Support for existing DOS Applications**: Windows allows most standard DOS applications to run under it directly. Any application that does not control the PC's hardware, use the PC BIOS or MS-DOS software interrupts, can run in its own window.

6) **Data Sharing:** Windows allows a data transfer between application Clipboard. Any type of data can be transferred from one window with the clipboard. The Dynamic Data Exchange (DDE) protocol defines how two applications can share information. Information such as bitmap, metafile, character strings and other data formats can be shared.

Support for Object Orientation

In order to create screen objects such as windows, the application developer defines a class (similar to record) specifying the necessary properties. Instances of class can then be created. Several applications can share the same windows simultaneously. To communicate with Instances of a window class, messages are sent and received by a special function called the window function. The windows handle all messages such as re-drawing the screen, displaying icon or pop-up menus and changing the contents of the client area.

Creation and Manipulation of a Window

MS Windows presents a pre-defined style for user-defined windows; it presents the structure of a window, followed by a discussion of windows manipulation.

1) **Structure of a Window:** *Figure* 7 displays possible elements for a window. The caption bar (or title bar) displays the name of application within the window. The system menu box contains names of commands available in all applications, such as minimise, maximize, resize and close. The minimize box clicked once reduces the window to an icon. The maximize box enlarges the window to the full screen. The menu bar contains a list of commands in the application. The client area is the area inside the window, which is under the application control.

2) **Creating Windows:** Each window created on the screen is a member of some user defined window class. Window classes are created by the application program. Several window classes can be active simultaneously. Each window class in turn can have several instances active at the same time. There are no predefined generic window classes that come with MS- Windows.

To create a window the following steps must be taken:

a) Set up a window class structure which defines the attributes of the window class. Attributes that can be defined include:

i) the window function, which handles all messages for this class
ii) the icon and the cursor used for this class of windows
iii) the background color of the client area iv) the window class menu
v) the re-drawing function used when resizing horizontally or vertically.

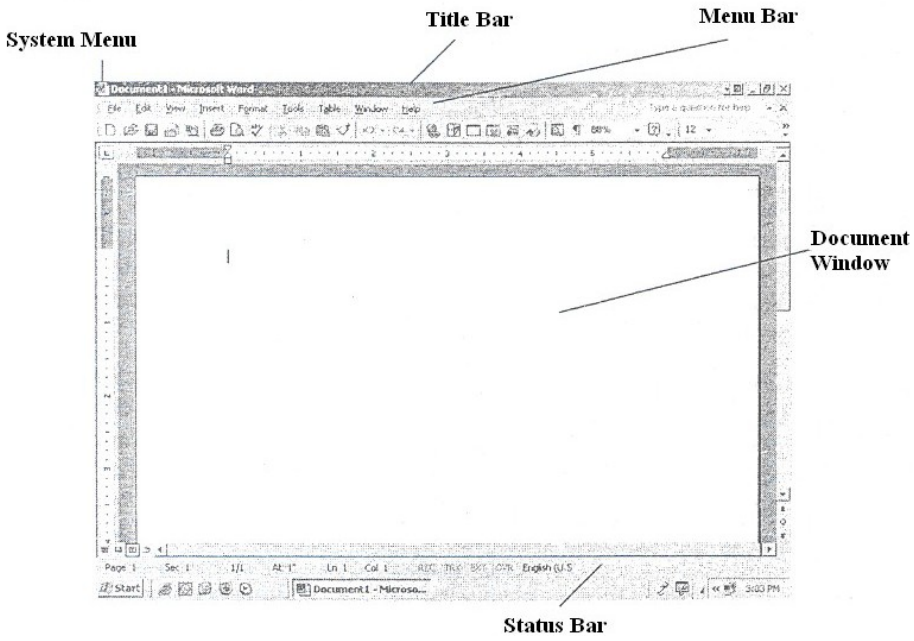


Figure 7: MS-WINDOWS screen element

3) Manipulating windows: An application can choose to display the Windows, resize the window, display additional information in the client area, and so on.

Pop-up and Child Windows

Pop-Up and child windows are special type of windows, and are used to communicate information between the application and the end-user. They remain on the screen for a short period of time. In this example, the pop-up Display information about a given file such as date and time of creation and the size. Dialog boxes, as discussed earlier, are a more sophisticated form of pop-up windows.

MS-Windows provides a collection of predefined child windows. These are the most common usage of child windows. These predefined classes are button, scroll bars, listbox, edit and static class. A developer can also define child windows that can be controlled by any user-defined operation.

Resources

Resources are used to manage windows and user-defined object. MS-WINDOWS provides nine kinds of resources to application developers. These resources are: icons, cursors, menus, dialog boxes, fonts, bitmaps, char strings, user-defined resources, and keyboard accelerators.

1) **Icons and Cursors**: Windows defines a few types of icons and cursors. An icon or a cursor is essentially a bit-mapped region that is used to represent and symbolise a window or cursor. A developer can also define an original icon or cursor using the ICONEDIT utility.

2) **Menus:** Each window can have its own menu bar. A menu item can be a character string or a bitmap. Each item of a menu bar in turn can have a pop-up menu presenting a list of options. Currently, Windows does not support nesting of pop-up menus within other pop-up menus. (Windows 3.0 provides this functionality). But a pop-up menu can invoke a dialog box. When a menu is selected, Windows sends one or more messages to the Window function of the Window containing the menu bar. These messages can be interpreted to perform the function corresponding to the menu item.

3) **Dialog Boxes:** These provide another mechanism besides pop-up menu and menu bar to obtain information from the end-user. Dialog boxes are much more flexible than menu bars or pop-up menus. Dialog boxes usually contain a group of child windows such as buttons, scroll bars, and editable fields. Just like windows, dialog boxes have a function that is used to process messages received from the user upon selection of options. Generally, dialog boxes appear as pop-up-windows. The user selects the option needed from the dialog box and the dialog box disappears. *Figure* 8 depicts an example of a dialog contains an edit box, a list box, and open and cancel buttons. The end-user can specify the name of a file either by selecting from the list box or by typing the name of the file in the edit box. By clicking on the open button, the application will open the selected file.

4) **Fonts**: Windows provides a few families of fonts with different sizes and shapes: Modern, Roman, Swiss, Helvetica, and Script. Application processors and desktop publishing can define additional fonts as needed.

5) **Bitmaps:** They are used to represent icons, cursors, or draw picture on the screen. Both mono and color bitmaps can be defined.

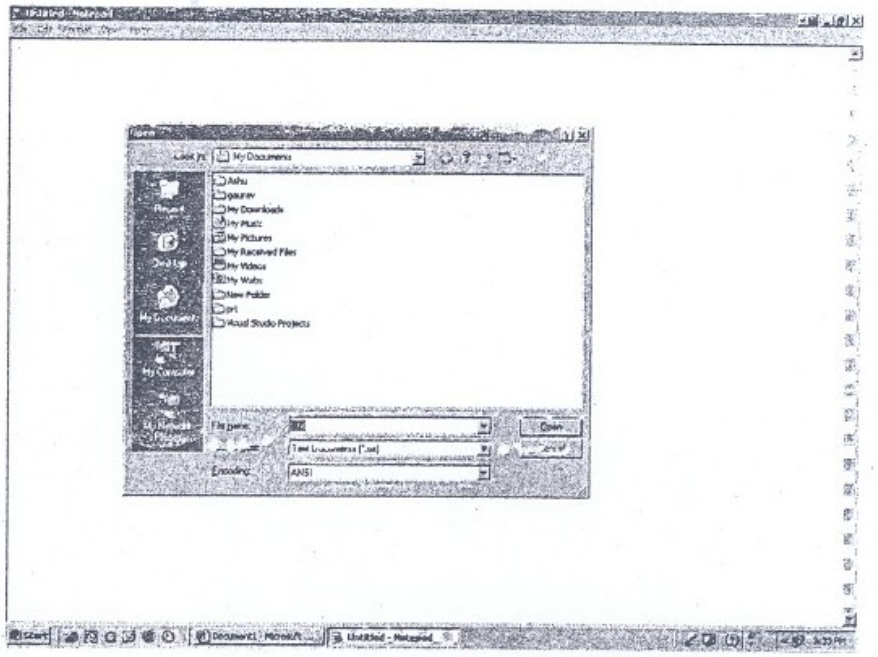


Figure 8: Dialog Box (MS-Windows) with .n edit box, a list box, and push buttons

6) **Character Strings:** Character strings are handled as resources mainly to provide a manageable solution to internationalisation of a window application. Instead of including a character string as part of the code, it can be placed in a resource file handled as a resource. Once in a resource file, different versions of the same character string resource file can exist for different languages. This separation of the character strings from the code makes intemationalising the application much easier, and also makes maintaining the application much simpler.

7) **User-Defined Resources**: These can be used for many purposes and support any user-defined data type. Any arbitrary data can be managed as a user-defined resource.

Resources are defined in a text file called a resource script. They are compiled through a utility called the Resource Compiler and linked with the windows application.

Resources are read only data. Once a resource script is compiled, it can be shared with other window applications.

Graphics Device Interface

Graphics in Windows are handled by the Graphics Device interface (GDI). Both vector and raster color graphics are supported by GDI. GDI supports only 1Wo-dimensional graphics. Vector graphics are supported by a set of vector drawing functions such as drawing a line, point, and polygon etc. pie chart.

Raster graphics are supported by pixel manipulation. Raster graphics can be stored or manipulated either as a bitmap or as a metafile. A bit-mapped representation of the graph can be manipulated using utilities such as BitBIt, PatBlt, and StretchBlt. A metafile provides binary encoding of GDI functions such as to draw vectors and to fill a region with a bitmap. Metafiles take up less disk space than a bit-mapped representation since they do not represent each pixel directly on disk. When metafiles are played, they execute the function encoding and perform the necessary graphics operations to display the graphics output.

3.7.2 Macintosh Toolbox

The tremendous success of the Macintosh computer popularized the window-style menu- driven user interface. Macintosh got its early start from Apple's Lisa.

The Macintosh GUI is called the Toolbox. The Toolbox consists of a collection of utilities to manipulate Macintosh's resources. In this section we present an overview of Toolbox functionally, examine the object-oriented features of the Toolbox and discuss major components of the toolbox's user interface features.

Functional Overview

The Toolbox provides a collection of utilities to access and manipulate Macintosh hardware and software resources. It provides a set of utilities to manipulate interface components such as windows, menu bar, and dialog boxes. These are discussed in the following sections. Some of the other utilities provided are:

1) **Fonts Manager:** allows manipulation of system and user-defined fonts.
2) **Event Manager:** provides monitoring of events generated by keyboard and keypad.
3) **Desk Manager:** provides access to desk utilities such as the calculation.
4) **Text Edit:** provides simple text editing capabilities such as cut and paste.
5) **Memory Manager:** provides a set of routines to manipulate dynamic memory.
6) **File Manager:** provides a set of routines to manipulate files and transfer data between files and applications.
7) **Silver Driver:** provides access to sound and music capabilities.
8) **Toolbox Utilities:** provides a set of general routines to manipulate icons, patterns, strings, fixed-point arithmetic, and so forth.

The Toolbox can be manipulated either from the assembly language or Macintosh Programmer's Workshop (MPW) language (Pascal, C, C++). Toolbox can be accessed from non-MPW languages including Think C. One of the major features of the Toolbox is that most of the Toolbox utilities reside in ROM. This provides a very responsive user interface.
From this point on, we will concentrate on the user components of the Macintosh Toolbox such as the Window Manager, the Manager, the Menu Manager, the Dialog Manager, the Scrap Manager, the Draw, and the Resource Manager.

Object-Oriented Features of Toolbox

Just like Microsoft Windows, the Macintosh Toolbox design is object-oriented. For example, the application developer defines a new type of window by the template for a new class of windows. Messages are sent between Toolbox application to change the behaviour of screen objects such as windows and cursors. When a window needs to be drawn or resized, a message is sent to the window definition function to perform the appropriate action.

The Window Manager

The Window Manager allows the application developer to manipulate windows on the screen. It deals with issues such as overlapping windows, resiling windows, moving windows around the screen, or changing a window from a foreground to a background window. Developers can create and manipulate pre-defined Toolbox windows with any desired shape and form. For example, the user can create a circular or a hexagonal window.

Toolbox comes with a series of pre-defined windows. The most popular of these windows is the document window. The document window, depicted in Figure 9, has the following regions: Title bar, Close Box, Scroll bar(s), Size Box and content region. The Title bar contains the title of the window. By clicking and holding the mouse within the region, the window can be dragged to a different screen. The Close Box is used to shut the window. Clicking inside this box will cause the window to disappear. The Size Box is used to resize the window. The horizontal or vertical Scroll bars can be used to scroll the contents of the window. The content region is the area that is controlled by the application.

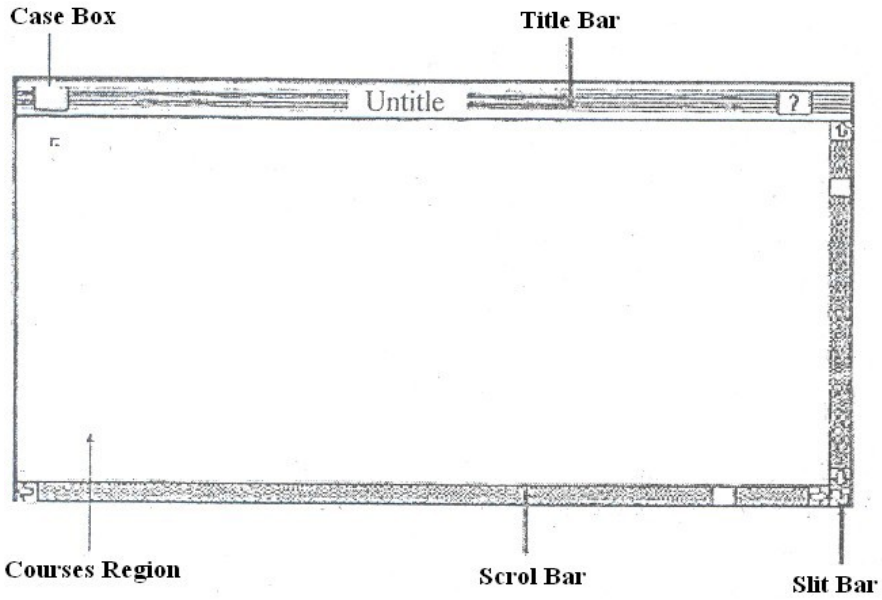


Figure 9: Macintosh Toolbox Document Window

When creating a document window anyone of tile components described above can be omitted. For example, anyone of the Scroll bars can be absent from the document window.

New types of windows can be created through either the Window Manager or the Resource Manager. The Resource Editor can be used to create a window template. A window template essentially defines a class of windows that can be instantiated by an application. For each class of window, defined by the system or there exists a function called the window definition function. This function is used to handle the behaviour and appearance of the window. When a window needs to be drawn or resized, a message is sent to the window definition function to perform the action for that type of window.

The Window Manager provides a large collection of routines to create, display, hide and activate/deactivate windows on the screen.

The Resource Manager

The Resource Manager provides access to various resources such as the windows, fonts, icons and menus. A user-defined resource such as a window is defined as Resource Editor. The template of the window and the window definition function is stored in a file called the resource file. A unique resource identifier is used to access a pre-defined resource. Resource identifier can be used to recognise the type of resource and the actual resource file to be read.

The Menu Manager

The Menu Manager provides a set of routines to create and manipulate menus. A menu bar, depicted in *Figure 10,* can contain a list of one or more menu items. Each menu item highlights itself when clicked with the mouse; this item is selected. The developer can define a menu using tile standard bar. Menus can be stored as a resource and managed by the Resource Manager. Once given to Manager, a unique identifier is used to reference the menu.

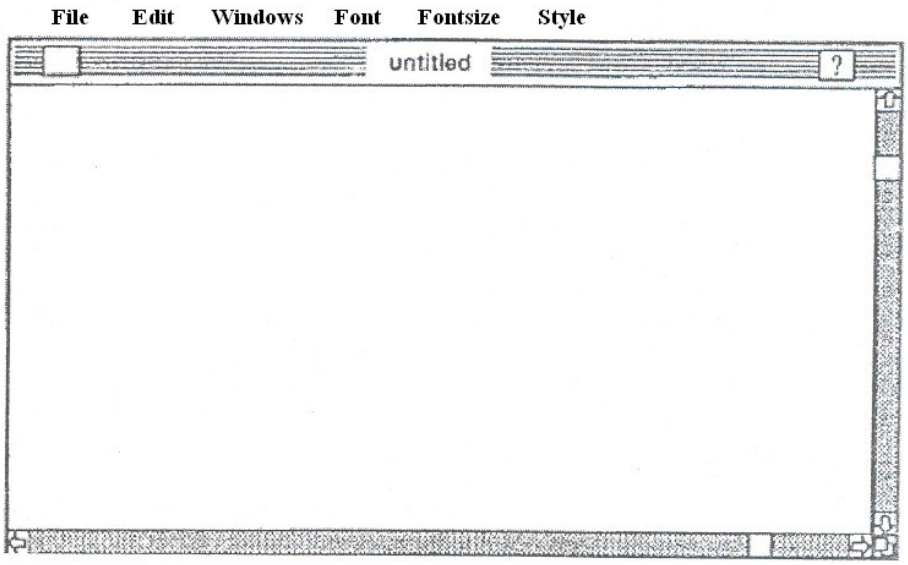


Figure 10: Macintosh Menu Bar

A menu item can even invoke a pull-down menu, which can be used to or choose additional selections, see *Figure* 11. A menu item in a pull-down menu can then invoke a dialog or an alert box. A menu item can be a textual icon. Unlike MS Windows, menus in Toolbox cannot be a part of a window Menus and only used to define the menu bar, which is associated with one Macintosh screen.

The Control Manager

The Control Manager provides a set of routines to define objects like buttons, check boxes and scroll bars. Controls are usually a window. For example, the scroll bars can be defined as part of a document. Most often controls are defined as part of a dialog or an alert box.

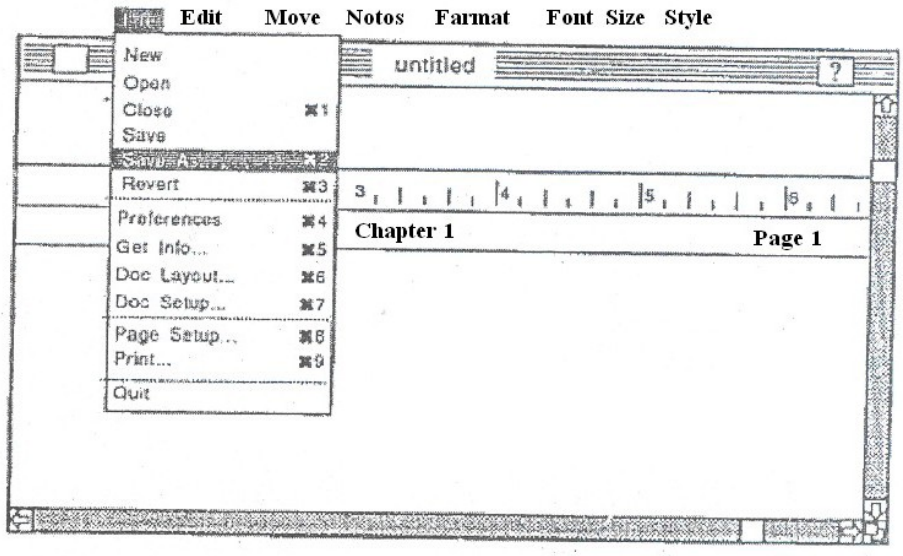


Figure 11: Macintosh Pull-down Menu

The Dialog Manager

The Dialog Manager provides a set of routines and data structures to manipulate dialogs and alert boxes. Dialog boxes are used to get additional information from the end-user of an application. An alert box is used to provide information such as a warning or a cautionary message.

The Scrap Manager allows the user to move data from one application to another. A portion of one window can be cut and placed in tile scrap manager and then pasted into another window. Once a portion of a window is cut, it is placed in the clipboard. The user can paste tile contents of the clipboard into another window.

3.7.3 Windows

The name X, as well as part of the initial design, was derived from an earlier window system called W developed at Standford University. Currently, the X Window system is supported by an X Consortium of primarily UNIX-based hardware and software vendors as a standard

base for user interfaces across most UNIX product lines. *Figure* 12 illustrates the components of UNIX GUIs based on X Windows.

The X Window system does not define any particular style of interface but rather provides a mechanism for supporting many styles. X is also network-based. In contrast with PM, the X architecture is based on the premise that an application can run one computer, while the graphical presentation of the application's output and responses from the user can occur on another computer. As such, X provides a flexible set of primitive window operations but carefully avoids dictating the look and feel of any particular application. X's device-independent functionally allows it to serve as a base for a variety of interface styles. X does not provide user interface components such as buttons, menus, or dialog boxes.

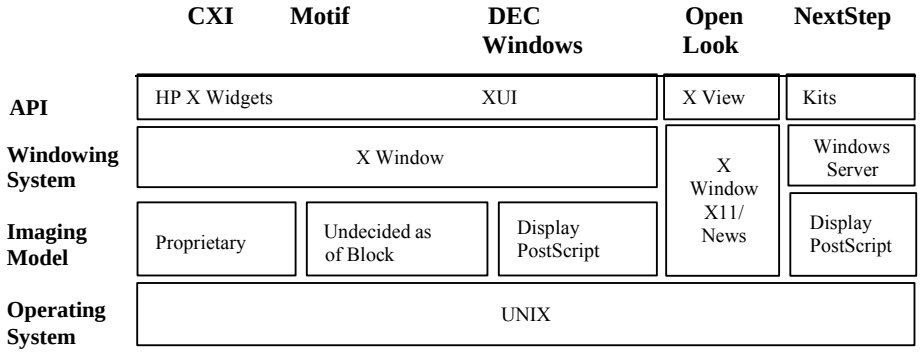


Figure 12: Components of major UNIX-based GUIs

An interface supporting X can theoretically use any X-Window display. The application program sends calls to the X-Window Library, which packages the requests as X packets and sends calls to the X-Window server. The server decodes the X packets and displays them on the screen. Exactly how the packets will be displayed (i.e. will look) on a workstation depends on the set of pre-designed window elements called widgets that are unique to tile particular workstation in the system environment. The ultimate look on one workstation may differ substantially from that on another, but the response to a user's action on these different-looking interfaces will be the same. Several hybrids of X exist.

The application-programming model for X, like PM and Windows, is event-driven, and the categories of events are similar. Like PM, X programs rest in wait loops until the underlying window-management system generates an event. Window hierarchy management is also common to the architecture of both systems. Like PM's window, windows in X are organized as parents and children, so attributes can be inherited and effects of the events applied to related members.

Since X does not support any particular interface style, most programmes build applications with libraries that provide an interface to the base window system. One standard programming interface to X is the Clanguage library known as X Library or Xlib. This library defines functions for access and control over the display, windows and input devices. Although commonly used, the library can be difficult and tedious to work with and often results in hundreds of lines of code for a GUI; A better solution is a higher-level toolkit, such as the X Toolkit (Xt), upon which many other X Window toolkits are based.

X Toolkit is the foundation upon which most of the commercial toolkits are based and is a good example for describing object-oriented functionally. The X Toolkit consists of two parts: a layer known as Xt lntrinsics and a set of widgets. Xt Intrinsics supports many of the available widget sets. The widget set implements user interface components including scroll bars, menus and buttons. Xt intrinsics provides a framework that allows the programmer to combine the components. Both parts are integrated with Xlib, allowing programmers access to higher-level libraries. A typical X programming environment is illustrated in *Figure* 13.

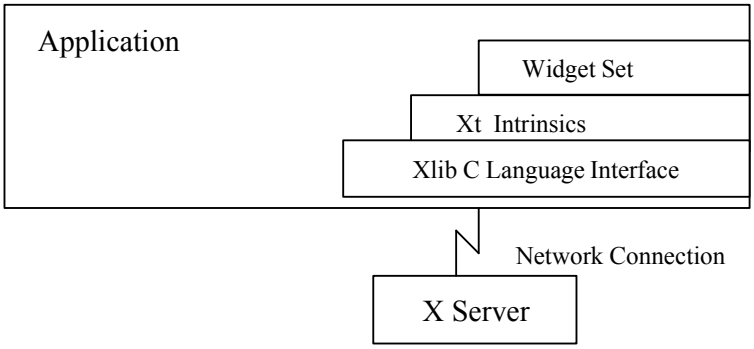


Figure 13: Typical X Window Development Environment

The common distinction in object-oriented programming between an application programmer (consumer) and a class library programmer (producer) fits well into the X environment. The X application programmer is likely to use a combination of a higher- level toolkit (e.g., the Toolkit's Xt Intrinsic), Xlib, and a widget set. The class library programmer, whose job is to create new widgets, is likely to use the capabilities from within one of the higher-level toolkits.

Depending on the toolkit, the widget programmer may be able to take advantage of object-oriented capabilities. Xt Intrinsic, for example, uses an object-oriented approach and organises widgets into classes. Xt Intrinsic defines the basic architecture of a widget. This allows widgets

to work together smoothly when built by different programmers or potentially when selected from different widget sets.

, A typical widget consists of two basic parts: a class record and an instance record. Each of these components is implemented as a C structure containing data and pointers to methods. Intrinsics defines the organisation of each structure. All widgets belonging to a class share a copy of common data methods for that class. Each individual widget has its own copy of instance-specific data. A widget's class record is usually allocated and initialised statically at compile time; a unique copy of the instance record is created at run time for each individual widget.

Since all widgets belonging to the same class share the same class record, the class record must contain only static data that do not relate directly to the state of an individual widget. For example, every widget's class record includes a field containing the widget's class name. The class record also contains methods that define the appearance and behaviour of all widgets in the class. Although most of these methods operate on the data in the widget's instance records, the methods themselves are shared by all widgets in a class.

Many object-oriented languages provide inheritance as a language construct. The Xt Intrinsics is written in the C language, which does not directly support object- oriented programming. Xt itself supports inheritance with a subclassing mechanism. To use Xt's subclassing capabilities, the programmer first finds an existing widget with similar .functions (a common step in object-oriented design), writes a subclass that can inherit existing data and methods, and then adds new data and methods as needed.

If a similar widget cannot be found, then a foundation class called Core is subclassed. All widget classes are subclasses of the core widget class. Like the class Window in PM, Core contains the basic methods for initializing, displaying, and destroying widgets, and reacting the external resizing. Core also stores basic properties (e.g., the geometry) of widgets and the data for handling events.

New widget classes inherit the methods (called resources) defined by their superclass by specifically including the definition of the superclass structure in the definition of tile new class. Xt Intrinsics provides two mechanisms for inheriting the methods defined by a superclass. The first mechanism is referred to as chaining. When a method is chained. Xt Intrinsics invokes the method defined by a widget's superclass first and then invokes the widget's method. This allows a widget to inherit part of a method from its superclass.

Xt Intrinsics also provides a mechanism for inheriting methods that are not chained. This is done by using special symbols to specify methods in the widget's class records. Each symbol is defined by the superclass that added the method to the widget's class record. These symbols can be used by a subclass of the widget class Core and do not have to be redefined by each widget class. Only classes that contribute new methods to the class record need to define new symbols. When a widget class specifies one of these symbols in the class record, Intrinsics copies the corresponding method used by the widget's superclass into the widget's class structure at class initialisation time. *Figure* 14 illustrates this architecture and shows the relationship between the class record and instance records of several widgets belonging to widget class core.

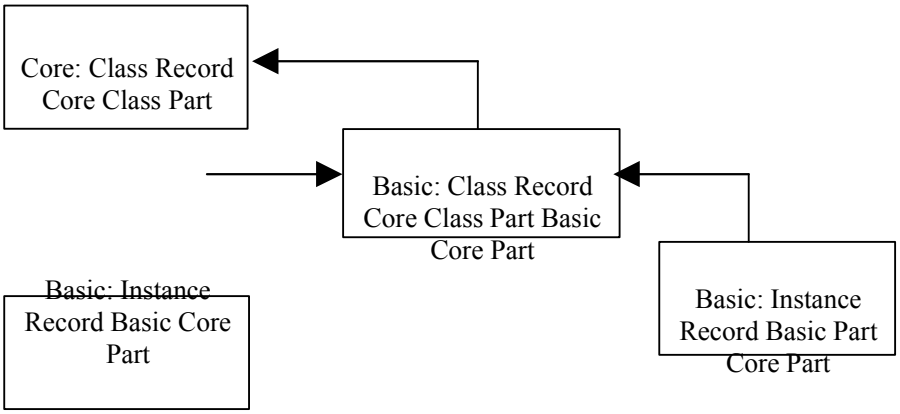


Figure 14: Inheriting from Widget Class Core

The Xt Intrinsics uses a data-abstraction technique to hide the implementation of a widget from applications that use the widget. Applications that use a widget see only the incomplete definition of the widget and therefore cannot directly access fields in the widget structure.

Applications declare all widgets as type Widget, which is known as opaque type. This means the application has a pointer to the widget structure but does not have access to the real definition of the data that it represents. The application cannot access the contents of the data structure.

Another object-oriented technique promoted in X Windows is to extract some general functionality from two or more new widget classes and create a metaclass. A metaclass is an abstract class that is not intended to be instantiated directly but serves as a superclass for other similar widgets. When creating a complete new widget se4it is frequently useful to create a class of hierarchy or metaclasses. With this approach, the top metaclass in the hierarchy defines elements that all widget classes in tile set have in common, and each subclass becomes more and more

specialised. Such an organisation allows the widget programmer to create new widget classes with the least amount of effort, although there is some extra initial effort required to design and create metaclasses; For example, this approach is used by the X Widget set from HP, where most basic widgets inherit from the primitive metaclass and most composite widgets inherit from the Manager widget class.

3.7.4 NeXT

The NeXT computer, with its three-dimensional user interface, was introduced in 1988. It has grabbed the attention of tile computer industry. The machine has been hailed as the most innovative computer invented in recent times. The computer was initially intended for the educational market. But the NeXT Corporation decided to widen the market for its machine to the commercial arena.

We provide a brief overview of the NeXT software tools and capabilities, followed by a discussion of the capabilities of the user-interface design.

Overview of NeXT Software

The NeXT computer is designed to reach a wide range of users from non-technical to power users. The non-technical users can deal with the graphic user interface to perform tasks by manipulating menus, icons, and dialog boxes. The power user can directly interact with its Mach Operating system.

The NeXT system software comprises three major pieces: the Mach operating system, applications and the NeXT user interface. The Mach operating system, developed at Carnegie Mellon University, is a re-designed UNIX. Mach re-designs the UNIX kernel to reduce the size.
NeXT also comes with a set of bundled applications. Currently, there are new applications being developed that take advantage of NeXT hardware and software capabilities. Some of the applications supported on NeXT are NeXT SQL Database Server, Mathematica (symbolic mathematics package, WYSIWYG editors and word processors, and more.

NeXT User Interface

The third and last component, the NeXT user interface, is the most impressive piece of NeXT technology. The NeXT user interface draws heavily on direct manipulation and modem graphic user interfaces. The NeXT user interface is composed of four components. Workspace Manager, Interface Builder, Application Kit, and NeXT Window Server.

The Workspace Manager allows the user to manage files and directories and to execute programs, when a user logs into a NeXT machine, the Workspace Manager is started, *Figure* 15, The Directory Browser window is used to navigate through the files on disk.

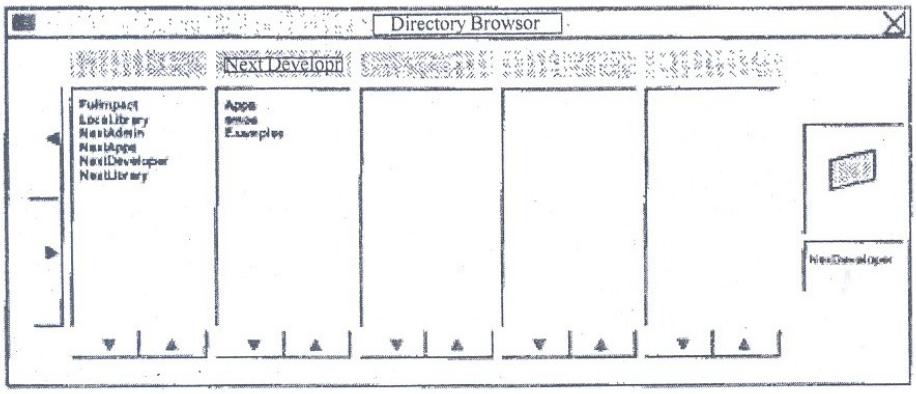


Figure 15: NeXT Workspace Manager Directory Browser

The Interface Builder lets the user create interfaces on tile screen without writing a single line of code. Users simply select the menu, control and screen objects from a palette, and then move the controls to the desired location.

The Application Kit is a library of user-interface objects. It is used in conjunction with the Interface Builder to design user interfaces. The Application Kit is discussed in the next section.

The Window Server handles all screen activities, such as drawing windows and handling events such as mouse clicks. The window Server itself does not perform the drawing and screen I/O commands. Display PostScript, designed by Adobe and NeXT, handles all such activities. Up to now, PostScript has been used only as a language for printer engines. With the advent of the PostScript, both screen and printer share the same protocol. Therefore one drawing method is used to display objects on the screen and the printer.

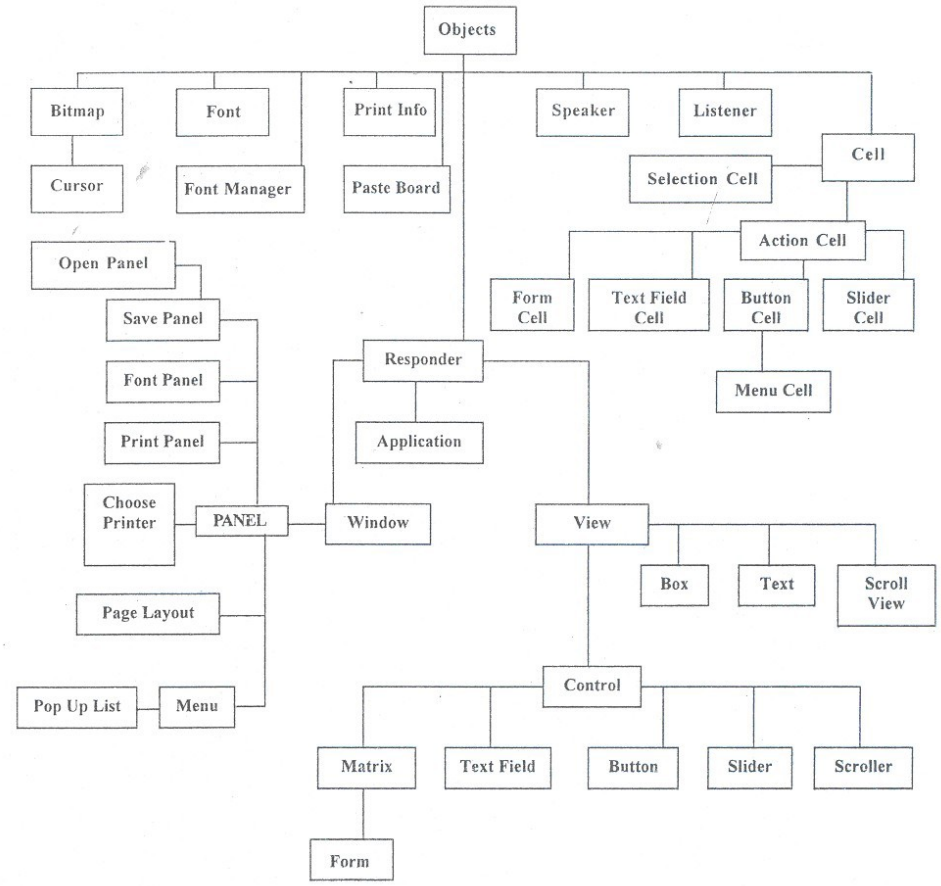


Figure 16: Application Kit

Application Kit

The Application Kit provides an extensive library of predefined classes. The hierarchy of classes is shown in *Figure* 16. These classes provide functionality to define user interfaces composed of menus, windows, buttons, slide bars, and sound. Each class within the hierarchy defines the behaviour of its objects. For example, a menu knows how to handle mouse events, when clicking on a menu item. Window objects know how to resize the window.

The Application Kit can be extended by adding new subclasses to the hierarchy. To add new subclasses, the class definition is written using the Objective C language. Each NeXT machine comes with a copy of Objective C. Objective C is an object- oriented extension of C. The language is a superset of C incorporating object orientation features from Smalltalk. Just like Smalltalk, it comes with a large collection of predefined classes to simplify the software development task. The language supports abstract data types, inheritance, and operator overloading. Unlike C++, Objective C does not extend the definition of any existing C language construct. It relies totally on the introduction of

new constructs and operates to perform tasks such as class definition or message passing.

To develop user interfaces, designers can use Application Kit from Objective C directly. This would be very much like developing an application using MacApp. They can create instances of objects from the Application Kit hierarchy and modify the attributes by calling the methods attached to the class definition. But the other method of defining user interface, using the Interface Builder, is much easier than coding it entirely in Objective C.

Designing User Interfaces with Interface Builder

The Interface Builder provides an easy to use utility to design a user interface using the muse and the screen. The Interface Builder is similar to an icon editor or screen painter. Designers can define the screen layout by selecting the screen objects from the Interface Builder palettes. The Interface Builder also helps to define the user-defined class and make connections between objects. Not all of the coding is automatic; the Application Kit and the Objective C language are also needed.

Defining a user interface using the Interface Builder requires the following steps:

1) **Define Layout of Screen**: the interface designer defines a user interface by simply selecting screen objects from the Interface Builder palettes, see *Figure 17. After picking an object from a palette using the mouse, the object can be dragged into the destination window and resized as desired.*

   The Interface Builder palettes include objects such as windows, menus, buttons, fields' radio buttons and more. At the top of the palettes window, three buttons allow the interface developer to select a wide array of user-interface objects.

2) **Define the User-Defined Classes:** the developer defines a new class definition I using the Classes Window. The Classes Window allows the developer to extend .the Application Kit class hierarchy. The developer navigates through the class hierarchy and creates a new subclass within the hierarchy. Methods and outlets (see next step) are defined for this new class definition. When a class is defined this way, only the template of the class is created.

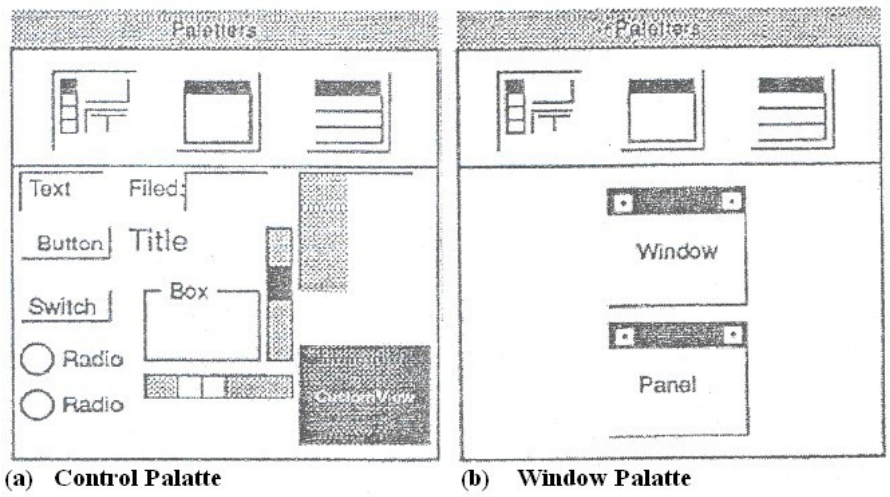


Figure 17: NeXT interface Builder palette

3) **Making Connections**: up to this point we have defined the layout of the user I interface. At this step, the developer needs to make connections among application objects. For example, when a scroller's slide bar is moved, a message is sent to an application object to perform certain actions like shifting the text. Again the Inspector is used to connect user interface objects with methods of classes within the application.

4) **Actual Application Code:** the previous steps are handled by Interface Builder directly. The last step is accomplished by writing the application code in Objective C. When the developer is done with the first two steps, the Interface Builder defines the source files necessary to build the application. These files contain the template for the class definitions and the connections made among objects. At this stage, the developer needs to extend these source files. Extensions are made to specify the logic of the program and the code for method definitions.

4.0 CONCLUSION

In this unit, you have been introduced to several GUI terms and their functionalities and GUI design considerations. It has also taken you through several GUI standards and it is our belief that by now you should be able to identify important features of several GUIs.

5.0 SUMMARY

The GUI has already made a tremendous contribution to the increased usability of computer systems. It is with great excitement that we look forward to future innovations in human-computer interfaces. GUI

development is at the vanguard of creativity in many areas, such as ergonomics, software development tools, computer graphics and linguistics to name just a few. The past decade has seen a rapid change il1 the understanding and definition of GUIs, but we have only been through the infancy of GUIs. There remains much to be done in terms of increasing the productivity of computer users, standardising operations across different architectures and adapting the human-computer interface to non-traditional applications. In this unit we discussed several issues related to GUI including functioning of several popular packages.

6.0 TUTOR-MARKED ASSIGNMENT

1) What are the four major components of GVI? Explain the functioning of any one component.
2) How does MS-Windows enhance DOS environment?
3) How are graphics supported in MS-Windows?
4) What types of utilities are provided in Toolbox? Explain their features.
5) Explain the functioning of Resources Manager and Menu Manager of Toolbox.
6) What is the basic philosophy of X-windows? How is it different from the rest of GUIs?
7) What are the major components of NeXT STEP? How do these elements
8) How are applications written in NeXTSTEP environment?

7.0 REFERENCES/FURTHER READINGS

Communication of ACM, April 1993.

Object Orientation: Concepts, Languages, Databases. User interfaces, Khosafian, Setrag & Razmik Abnours, New York; Wiley & Sons, 1990.

UNIT 2 INTRODUCTION TO OPERATING I SYSTEM I

CONTENTS

1.0 INTRODUCTION

An operating system is a system software which may be viewed as an organized collection of software consisting of procedures for operating a computer and providing an environment for execution of programs. It acts as an interface between users and the hardware of a computer system.

There are many important reasons for studying operating systems. Some of them are:

1) User interacts with the computer through the operating system in order to accomplish his task since it is his primary interface with a computer.
2) It helps users to understand the inner functions of a computer very closely.
3) Many concepts and techniques found in the operating system have general applicability in other applications.

The introductory concepts and principles of an operating system will be the main issues for discussion in this unit. The unit starts with the basic definition and then goes on to explain the stages of evolution of operating systems. It further gives details of several approaches to operating system design.

In the last two subsections of the unit we classify an advanced operating system and explain some characteristics of modern operating systems.

2.0 OBJECTIVES

After going through this unit, you should be able to:

list stages of evolution of operating systems
classify different types of operating systems
compare different approaches to operating system design.

3.0 MAIN CONTENT

3.1 What is an Operating System?

An operating system is an essential component of a computer system. The primary objectives of an operating system are to make the computer system convenient to use and to utilise computer hardware in an efficient manner.

An operating system is a large collection of software, which manages the resources of the computer system, such as memory, processor, file system and input/output devices. It keeps track of the status of each resource and decides who will have control over computer resources, for how long and when. The positioning of an operating system in the overall computer system is shown in *Figure* 1.

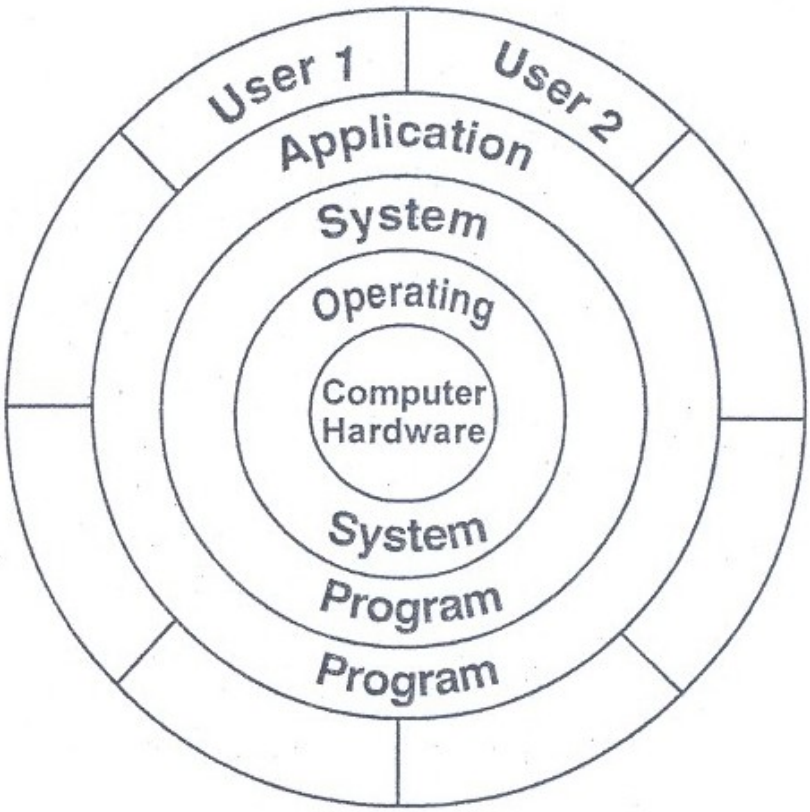


Figure 1: Component of Computer System

From the Figure, it is clear that the operating system directly controls computer hardware resources. Other programs rely on facilities provided by the operating system to gain access to computer system resources. There are two ways one can interact with the operating system:

1) By means of operating System Call in a program
2) Directly by means of Operating System Commands.

System Call: System calls provide the interface to a running program and the operating system. The user program receives operating system services through the set of system calls. Earlier these calls were available in assembly language instructions but now a day these features are supported through high level languages like C, Pascal etc., which replace assembly language for system programming. The use of system calls in C or Pascal programs very much resemble pre-defined functions or subroutine calls.

As an example of how system calls are used, let us consider a simple program to copy data from one file to another. In an interactive system, the following system calls will be generated by the operating system:

Prompt message for inputting two file names and reading it from the terminal.
Open source and destination file.
Prompt error message in case the source file cannot be open because it is protected against access or because the destination file cannot be created since there is already a file with this name.
Read the source file
Write into the destination file
Display status information regarding various Read/Write error conditions. For example, the program may find that the end of the file has been reached or that there was a hardware failure. The write operation may encounter various errors, depending upon the output device (no more disk space, physical end of tape, printer out of paper and so on).
Close both files after the entire file is copied.

As we can observe, a user program makes heavy use of the operating system. All interaction between the program and its environment must occur as the result of requests from the program to the operating system. Operating System Commands: Apart from system calls, users may interact with the operating system directly by means of operating system commands.

For example, if you want to list files or sub-directories in MS-DOS, you invoke dir command. In either case, the operating system acts as an interface between users and the hardware of the computer system. The fundamental goal of computer systems is to solve user problems. Towards this goal computer hardware is designed. Since the bare hardware alone is not very easy to use, programs (software) are developed. These programs require certain common operations; l such as controlling peripheral devices. The command function of controlling and f allocating resources are then brought together into one piece of software; the operating system.
To see what operating systems are and what operating systems do, let us consider how they have evolved over the years. By tracing their evolution, we can identify the common elements of operating systems and examine how and why they have developed as they have.

3.2 Evolution of Operating Systems

An operating system may process its task serially (sequentially) or concurrently (several tasks simultaneously). It means that the resources of the computer system may be dedicated to a single program until its completion or they may be allocated .among several programs in different stages of execution. The feature of the operating system to execute multiple programs interleaved fashion or different time cycles is

called multiprogramming systems. In this section, we will try to trace the evolution of the operating system. In particular, we will describe serial processing, batch processing and multiprogramming.

3.2.1 Serial Processing

Programming in 1's and O's (machine language) was quite common for early computer systems. Instructions and data used to be fed into the computer by means of console switches or perhaps through a hexadecimal keyboard. Programs used to be started by loading the program computer register with the address of the first instruction of a program and its Jesuit (program) used to be examined by the contents of various registers and memory locations of the machine. Therefore, programming in this style caused a low utilisation of both users and machine.

The advent of Input/output devices, such as punched cards, paper tape and language translators (Compiler /Assemblers) brought a significant step in computer system utilisation. Programs started being coded into programming language first changed into object code (binary code) by translator and then automatically loaded into memory by a program called loader. After transferring control to the loaded program, the execution of a program begins and its result gets displayed or printed. Once in memory, the program may be re-run with a different set of input data.

The process of development and preparation of a program in such an environment is slow and cumbersome due to serial processing and numerous manual processing. In a typical sequence first the editor is called to create a source code of user program written in programming language, then the translator is called to convert a source code into binary code and then finally the loader is called to load the executable program into the main memory for execution. If syntax errors are detected, the whole process must be redone from the beginning.

The next development was the replacement of card-decks with standard input/output and some useful library programs, which were further linked with user programs through system software called linker. While there was a definite improvement over machine language approach, the serial mode of operation is obviously not very efficient. This results in low utilization of resources.

3.2.2 Batch Processing

Utilisation of computer resources and improvement in programmer's productivity was still a major problem. During the time that tapes were being mounted or the programmer was operating the console, the CPU was sitting idle.

The next logical step in the evolution of the operating system was to automate the sequencing of operations involved in program execution and in the mechanical aspects of program development. Jobs with similar requirements were batched together and run through the computer as a group. For example, suppose the operator received one FORTRAN program, one COBOL program and another FORTRAN program. If he runs them in that order, he would have to set up for FORTRAN program environment (loading the FORTRAN compiler tapes), then set up COBOL program and finally FORTRAN program again. If he runs the two FORTRAN programs as a batch, however, he could set up only once for FORTRAN thus saving operator's time.

Batching similar jobs brought utilisation of system resources quite a bit. But there were still problems. For example, when a job is stopped, the operator would have to notice that fact by observing the console, determine why the program stopped and then load the card reader or paper tape reader with the next job and restart the computer. During this transition from one job to the next, the CPU sat idle.

To overcome this idle time, a small program called a resident monitor was created which is always resident in the memory. It automatically sequenced one job to another job. The resident monitor acts according to the directives given by a programmer through control cards, which contain information like marking of job's beginnings and endings, commands for loading and executing programs, etc. These commands belong to job control language. These job control language commands are included with user program and data. Here is an example of job control language commands.

$COB - Execute the COBOL compiler
$JOB - First card of a job
$END - Last card of a job
$LOAD - Load program into memory
$RUN - Execute the user program

Figure 2 shows a sample card deck set up for a simple batch system.

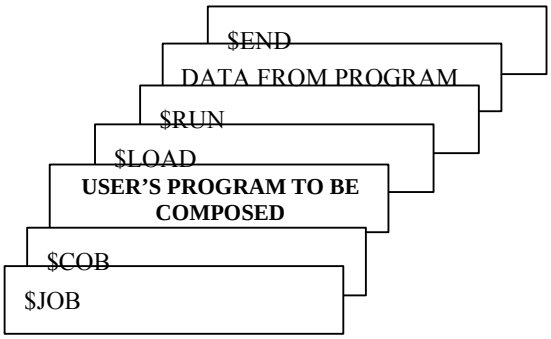


Figure 2: Card Deck for Cobol Program for a Simple Batch System

With sequencing of program execution mostly automated by batch operating system, the speed discrepancy between the fast CPO and the comparatively slow input/output devices such as card readers, printers emerged as a major performance bottleneck. Even a slow CPO works in the microsecond range, with millions of instructions per second. A fast card reader, on the other hand, might read 1200 cards per minute. Thus, the difference in speed between the CPO and its input/output devices may be three orders of magnitude or more.

"The relative slowness of input/output devices can mean that the CPO is often waiting, for input/output. As an example, an assembler or compiler may be able to process 300 or more cards per second. A fast card reader, on the other hand, may be able to read only 1200 cards per minute. This means that assembling or compiling a 1200 card program would require only 4 seconds of CPO time but 60 seconds to read. "Thus, the CPO would be idle for 56 out of 60 seconds or 93.3 per cent of the time. The resulting CPO utilization is only 6.7 per cent. The process is similar for output operations. The problem is that while an input/output is occurring, the CPO is idle, waiting for the input/output to complete; while the CPO is executing, input/output devices are idle.

Over the years, of course, improvements in technology resulted in faster input/output devices. But CPO speed increased even faster. According to Moore's Law, CPO speed is getting doubled every 18 months. Therefore, the need was to increase the throughput and resources utilization by overlapping input/output and processing operations. Channels, peripheral controllers and later dedicated input/output processors brought a major improvement in this direction. DMA (Direct Memory Access) chip which directly transfers the entire block of data from its own buffer to main memory without intervention by1he CPO was a major development. While the CPO is executing, DMA can transfer data between high-speed input/output devices and the main

memory. The CPO requires to be interpreted per block only by DMA. Apart from DMA, there are two other approaches to improving system performance by overlapping input, output and processing. These are called buffering and spooling.

Buffering is a method of overlapping input, output and processing of a single job. The idea is quite simple. After data has been read and the CPU is about to start operating on it, the input device is instructed to begin the next input immediately. The CPU and input device are then both busy. With luck, by the time the CPU is ready for the next data item, the input device will have finished reading it. The CPU can then begin processing the newly read data, while the input device starts to read tile following data. Similarly, this can be done for output. In this case, the CPU creates data that is put into a buffer until an output device can accept it.

If the CPU is, on the average, much faster than an input device, buffering will be of little use. If the CPU is always faster, then it always finds an empty buffer and has to wait for the input device. For output, the CPU can proceed at full speed until, eventually, all system buffers are full. Then the CPU must wait for tile output device. This situation occurs with input/output bound jobs where the amount of input/output device, the speed of execution is controlled by the input/output device, not by the speed of the CPU.

A more sophisticated form of input/output buffering is called SPOOLING' (Simultaneous Peripheral Operation On Line) which essentially uses the hard disk (secondary memory) as a very large buffer *(Figure* 3) for reading and for storing output files.

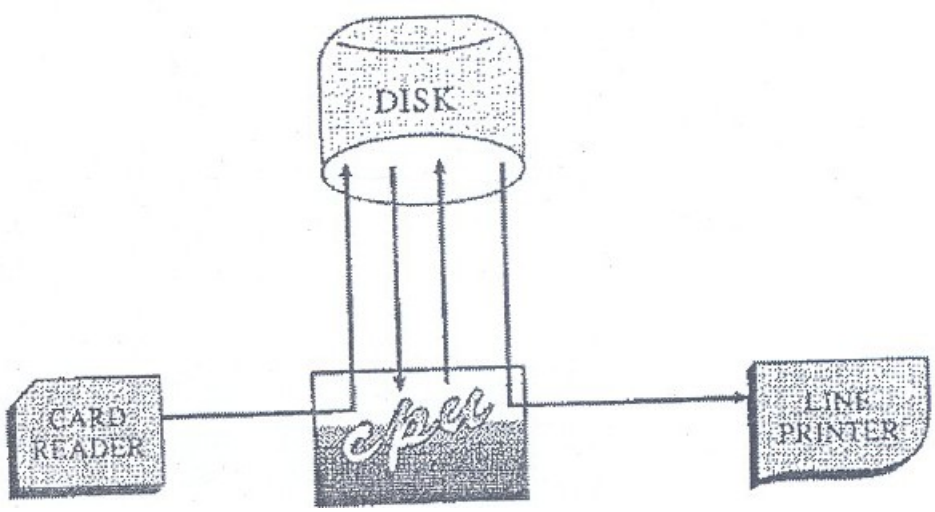


Figure 3: Spooling

Buffering overlaps input, output and processing of a single job whereas Spooling allows CPU to overlap the input of one job with computation and output of other jobs. Therefore this approach is better than suffering. Even in a simple system, the spooler maybe reading the input of one job while printing tile output of a different job.

3.2.3 Multiprogramming

Buffering and Spooling improve system performance by overlapping the input, output and computation of a single job, but both of them have their limitations. A single user cannot always keep CPU or *1/0* devices busy at all times. Multiprogramming offers a more efficient approach to increase system performance. In order to increase the resource utilisation, systems supporting multiprogramming approach allow more than one job (program) to reside in the memory to utilise CPU time at any moment. More number of programs competing for system resources better will mean better resource utilisation.

The idea is implemented as follows. The main memory of a system contains more than one program *(Figure* 4).

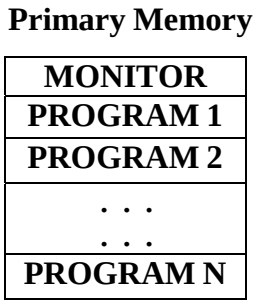


Figure 4: Memory Layout in Multiprogramming Environment

The operating system picks one of the programs and starts executing. During execution of program I it needs some *1/0* operation to complete in a sequential execution environment *(Figure 5a).* The CPO would then sit idle whereas in a multiprogramming system, *(Figure 5b)* the operating system will simply switch over to the next program (program 2).

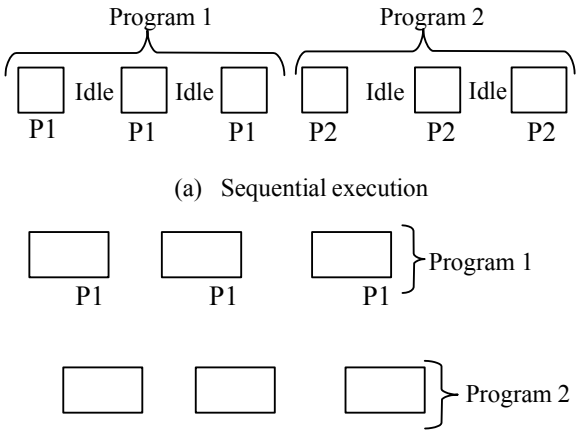


Figure 5: Multiprogramming

When that program needs to wait for some I/O operation, it switches over to program 3 and so on. If there is no other new program left in the main memory, the CPU will pass its control back to the previous programs.

Multiprogramming has traditionally been employed to increase the resources utilisation of a computer system and to support multiple simultaneously interactive users (terminals).

Compared to the operating system, which supports only sequential execution, the multiprogramming system requires some form of CPU and memory management strategies, which will be discussed in the next section.

3:3 Operating System Structure

Since operating system is a very large and complex software, supports a large number of functions. It should be developed as a collection of several smaller modules with carefully defined inputs, outputs and functions rather than a single piece of software. **In** this section, we will examine different operating system structure.

3.3.1 Layered Structure Approach

The operating system architecture based on layered approach consists of a number of layers (levels), each built on top of lower layers. The bottom layer is the hardware; the highest layer is the user interface. The first system constructed in this way was the THE system built by E. W. Dijkestra (1968) and his students. The **THE** system was a simple batch operating system which had 32k of27 bit words.

The system supported 6 layers *(Figure* 6).

5	User Programs
4	Buffering for I/O Devices
3	Device Driver
2	Memory Manager
1	CPU Scheduling
0	Hardware

Figure 6: The layered structure of THE operating system

As shown in *Figure* 6, layer 0 dealt with hardware; the higher layer I handled allocation of jobs to processor. The next layer implemented memory management. Level 3 contained the device driver for the operator's console. By placing it, as well as I/O buffering, at level 4, above memory management, the device buffers could be placed in virtual memory. The I/O buffering was also above the operator's console, so that I/O error conditions could be output to the operator's console.

The main advantages of the layered approach is modularity which helps in debugging and verification of the system easily. The layers are designed in such a way that it uses operation and services only of a layer below it. A higher1ayer need not know how these operations are implemented, only what these operations do. Hence each layer hides implementation details from higher-level layers. Any layer can be debugged without any concern about the rest of the layer.

The major difficulty with the layered approach is definition of a new level i.e. how to differentiate one level from another. Since a layer can use the services of a layer below it, it should be designed carefully. For example, the device driver for secondary memory must be at a lower level than tile memory management routines since memory management requires the ability to use the backing store.

3.3.2 Virtual Machine

It is a concept which creates the illusion of a real machine. It is created by a virtual machine operating system that makes a single real machine appear to be several real machines. This type of situation is analogous to the communication line of Telephone Company, which enables separate and isolated conversations over the same wire(s).

The following figure illustrates this concept.

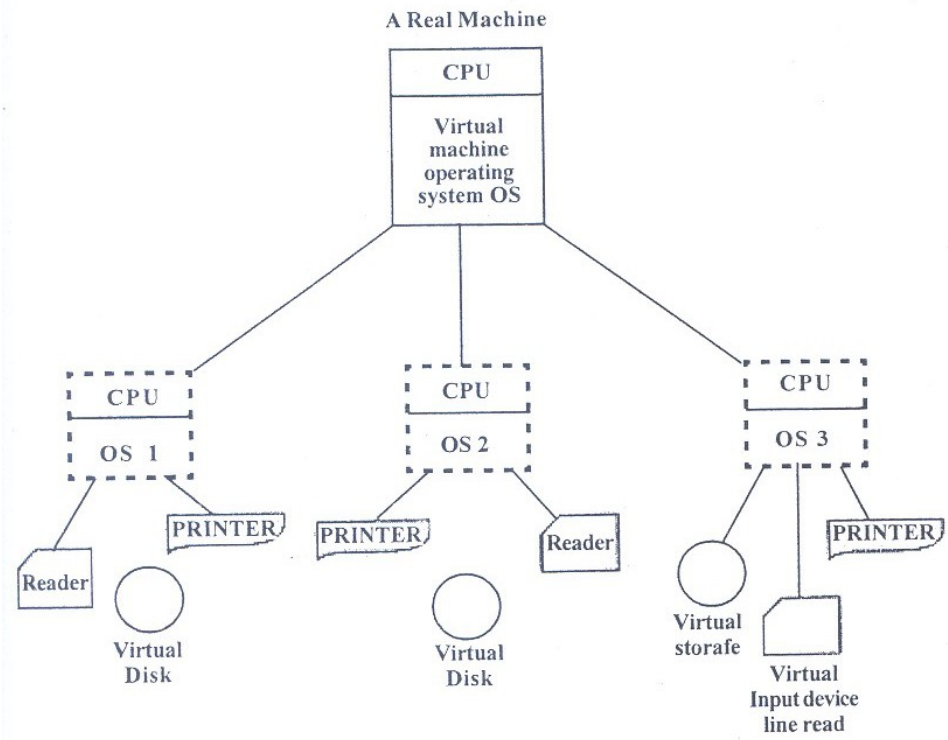


Figure 7: Creation of Several Virtual Machines by a Single Physical Machine

From the user's viewpoint, a virtual machine can be made to appear very similar to an existing real machine or they can be entirely different. An important aspect of this- technique is that each user can run the operating system of his own choice. This fact is depicted by OS, (Operating System I), OS2' OS, etc. as in *Figure* 7.

To understand this concept, let us try to understand the difference between conventional multiprogramming system (Figure 8) and virtual machine multiprogramming *(Figure* 9). In conventional multiprogramming, processes are allocated a portion of the real machine resources. The same machine resources are distributed among several processes.

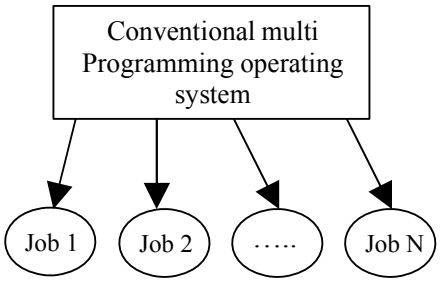


Figure 8: Conventional Multiprogramming

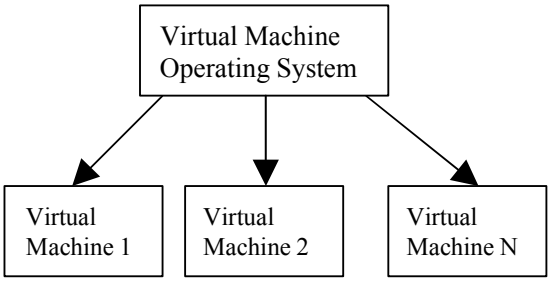


Figure 9: Virtual Machine Multiprogramming

In the virtual multiprogramming system, a single real machine gives the illusion of several virtual machines, each having its own virtual processor, storage and I/O devices possibly with much larger capacities.

The virtual machine has many uses and advantages:

1) Concurrent running of dissimilar operating systems by different users.
2) Elimination of certain conversion problems.
3) Software development: Programs can be developed and debugged for machine configurations that is different from those of host (for example virtual operating system VM/370 can produce virtual 370 that are different from the real 370 such as larger main memory).
4) Security and Privacy: The high degree of separation between independent virtual machines aids in ensuring privacy and security. The most widely used operating system in this category is VMl370. It manages IBM/370 computer and creates the illusion that each of several users has a complete system 370 (including wide range of I/O devices) which can run different operating systems at once, each of them on its own virtual machine.

It is also possible through software to share files existing on physical disk and much information through virtual communication software.

The virtual machines are created by sharing the resources of the physical computer. CPU scheduling can be used to share the CPU and make it appear that users have their own processor. Users are thus given their own virtual machine. They can run on their virtual machines any software desired. The virtual machine software is concerned with multiprogramming multiple virtual machines onto a physical machine but need not consider any other software support from user.

The heart of the system, known as the virtual machine monitor, runs on the bare hardware and does the multiprogramming, providing not one,

but several virtual machines to the next layer up, as shown in *Figure?* However, unlike all other operating systems, these virtual machines are not extended machines, with files and other nice features. Instead, they are exact copies of the bare hardware, including kernel/user mode, I/O, interrupts, and everything else the real machine has.

Because each virtual machine is identical to the true hardware, each one can run any operating system that will run directly on the hardware. In fact, different virtual machines can, and usually do, run different operating systems. Some run one of the descendants of OS/360 for batch processing, while other ones run a simple, single-user, interactive system called CMS (Conversational Monitor System) for time-sharing users.

When a CMS program executes a system call, the call is trapped to the operating system in its own virtual machine, not to VM/370; just as it would if it were running on a real machine instead of a virtual one. CMS then issues the normal hardware I/O instructions for reading its virtual disk or whatever is needed to carry out the call. These I/O instructions are trapped by VM/370, which then performs them as part of its simulation of the real hardware. By making a complete separation of the functions of multiprogramming and providing an extended machine, each of the pieces can be mush simpler and more flexible.

The virtual machine concept has several advantages. Notice that there is complete protection. Each machine is completely isolated from all other virtual machines, so there is no problem with protection. On the other hand, there is no sharing. To provide sharing, two approaches have been implemented. First, it is possible to share a minidisk. This scheme is modeled after a physical shared disk, but implemented by software. With this technique, files can be shared. Second, it is possible to define a network of virtual machines, each of which can send information over the virtual communications network. Again, the network is modeled after physical communication networks, but implemented in software.

Such a virtual machine system is a perfect vehicle for operating systems research and development. Normally changing an operating system is a difficult process. Since operating systems are large and complex programs, it is difficult to be sure that a change in one point does not cause obscure bugs in some other part. This situation can be particularly dangerous because of the power of the operating system. Since the operating system executes in monitor mode, a wrong change in a pointer could cause an error that would destroy the entire file system. Thus, it is necessary to test all changes to the operating system carefully.
But the operating system runs on and controls the entire machine. Therefore, the current system must be stopped and taken out of use,

while changes are made and tested. This is commonly called system development time. Since it makes the system unavailable to users, system development time is often scheduled late at night or on weekends.

A virtual machine system can eliminate much of this problem. System programmers are given their own virtual machine and system development is done on the virtual machine, instead of on a physical machine. The normal system operation seldom need be disrupted for system development.

3.3.3 Client-Server Model

VM/370 gains much in simplicity by moving a large part of the traditional operating system code (implementing the extended machine) into a higher layer itself. VM/370 is still a complex program because stimulating a number of virtual 370s is not that simple (especially if you want to do it efficiently).

A trend in modern operating systems is to take this idea of moving up into higher layers even further, and remove as much as possible from the operating system, leaving a minimal kernel. The usual approach is to implement most of the operating system functions in user processes. To request a service, such as reading a block of a file, a user process (now known as the client process) sends the request to a server process, which then does the work and sends back the answer.

In this model, shown in *Figure 10,* all that the kernel does is to handle the communication between clients and servers. By splitting the operating system up into parts, each part is made to handle one facet of the system, such as file service, process service, and terminal service or memory service. This way, each part becomes small and manageable. Furthermore, because all the servers run as user- mode processes, and not in kernel mode, they do not have direct access to the hardware. Asa consequence, if a bug in the file server is triggered, the file service may crash, but this will not usually bring the whole machine down.

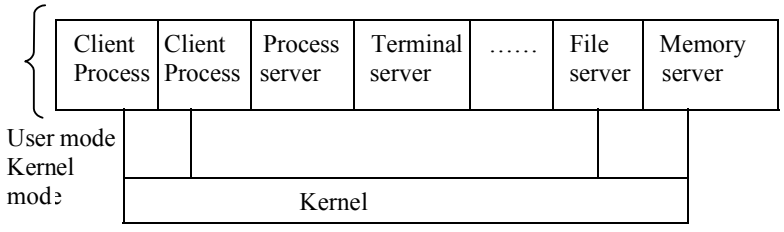


Figure 10: The Client-Server Model

Another advantage of the client-server mode! is its adaptability to use in distributed systems *(Figure11).* If a client communicates with a server

by sending it messages, the client need not know whether the message is handled locally in its own machine, or whether it was sent across a network to a server on a remote machine. As far as the client is concerned, the same thing happens in both cases: a request was sent and a reply came back.

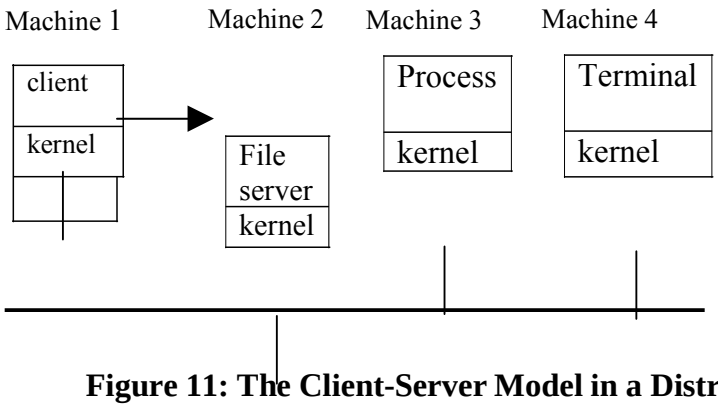


Figure 11: The Client-Server Model in a Distributed System

The picture painted above of a kernel that handles only the transport of messages from clients to servers and back is not completely realistic. Some operating system functions (such as loading commands into the physical I/O device registers) are difficult, if not impossible, to do from user-space programs. There are two ways of dealing with this problem. One way is to have some critical server processes (e.g., I/O device drivers) actually run in kernel mode, with complete access to all the hardware, but still communicate with other processes using the normal message mechanism.

The other way is to build a minimal amount of mechanism into the kernel, but leave the policy decisions up to servers in user space. For example, the kernel might recognize that a message sent to a certain special address means to take the contents of that message and load it into the I/O device registers from some disk, to start a disk read. In this example, the kernel would [jot even inspect the bytes in the message to see if they were valid or meaningful; it would just blindly copy them into the disk's device registers. (Obviously some scheme for limiting such messages to authorized processes only must be used). The split between mechanism and policy is an important concept; it occurs again and again in operating systems in various contexts.

3.3.4 Kernel Approach

Kernel is that part of operating system which directly makes interface with hardware system. Its main functions are:

To provide a mechanism for creation and deletion of processes.

To provide processor scheduling, memory management and I/O management
To provide a mechanism for synchronisation of processes so that processes synchronise their actions.
To provide a mechanism for interprocess communication.

The UNIX operating system is based on kernel approach *(Figure* 12). It consists of two separable parts: (i) Kernel (ii) System Programs.

As shown in the *Figure* 12, the kernel is between system programs and hardware. l The kernel supports the file system, processor scheduling, memory management and other operating system functions through system calls. The UNIX operating system supports a large number of system calls for process management and other operating system functions. Through these system calls, the program utilises the services of the operating system (kernel).

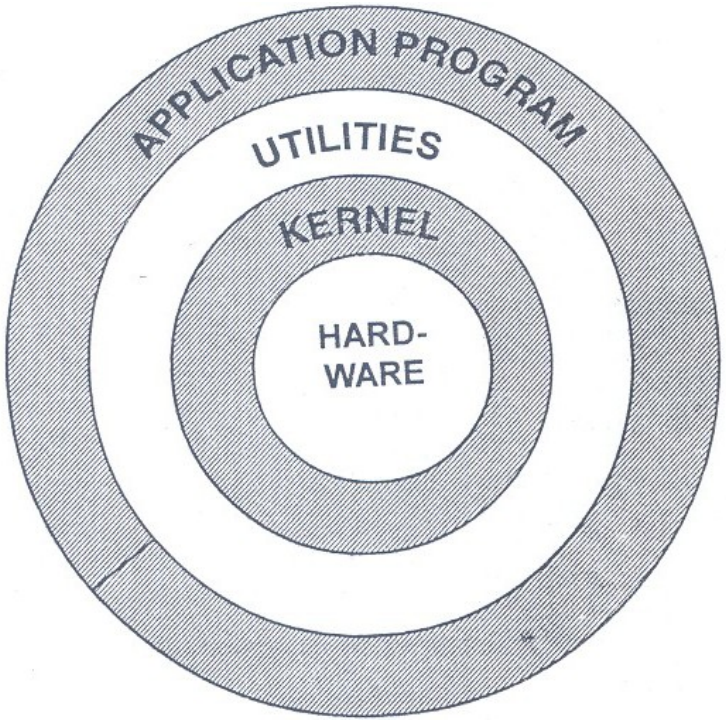


Figure 12: UNIX Operating System Structure

3.4 Classification of Advanced Operating Systems

The drive for advanced operating systems has come from two directions. First, it has come from advances in the architecture of multi-computer systems and is now driven by a wide variety of high-speed architectures. Hardware design of extremely fast parallel and distributed systems is fairly well understood. These architectures offer great potential for

speed up but they also present a substantial challenge to operating system designers. The following *Figure* 13 gives a broad classification of the advanced operating system.

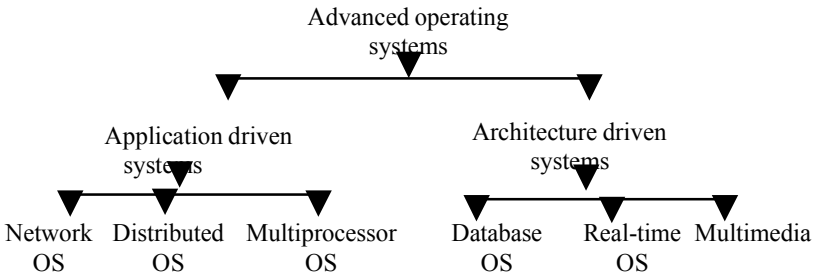


Figure 13: Classification of Advanced OS

A second class advanced operating system is driven by applications. There are several important applications that require special operating system support, as a requirement as well as for efficiency. General-purpose operating systems are too broad in nature and inefficient and fail to provide adequate support for such applications. Three specific applications, namely, multimedia operating systems, database systems and real- time systems, have received considerable attention in the past and the operating system issues for these systems have been extensively examined.

3.4.1 Architecture Driven Operating System

A brief discussion of three operating systems in the category of the Architecture Driven as follows:

Network Operating System

A network operating system is a collection of software and associated protocols that allow a set of autonomous computers, which are interconnected by a computer network, to be used together in a convenient and cost-effective manner. In a network operating system, the users are aware of the existence of multiple computers and can log in to remote machines and copy files from one machine to another machine.

Some of typical characteristics of network operating systems which make it different from a distributed operating system (discussed in the next section) are the followings:

Each computer has its own private operating system instead of running part of a global system wide operating system.

Advanced operating systems
Application driven systems
Architecture driven systems
Network OS
Distributed OS

Multiprocessor OS	Database OS	Real-time OS	Multimedia

Each user normally works on his/her own system: using a different system requires some kind of remote login, instead of having the operating system dynamically allocate processes to CPUs.

Users are typically aware of where each of their files are kept and must move file from one system to another with explicit file transfer commands instead of having file placement managed by the operating system.

The system has little or no fault tolerance; if 5% of the personal computers crash, only: 5% of the users are out of business.

The network operating system offers many capabilities including:

Allowing users to access the various resources of the network hosts.
Controlling access so that only users in the proper authorization are allowed to access particular resources.
Making the use of remote resources appear to be identical to the use of local resources.
Providing up-to-the minute network documentation on-line.

As we said earlier, the key issue that distinguishes a network operating system from a distributed one is how aware the users are of the fact that multiple machines are being f used. The visibility occurs in three primary areas; file system, protection and program execution.

Distributed Operating System

Distributed operating systems are operating systems for a network of autonomous computers connected by a communication network. A distributed operating system: controls and manages the hardware and software resources of a distributed system such that its users view the entire system as a powerful monolithic computer system. When a program is executed in a distributed system, the user is not aware of where the program is executed or of the location of the resources accessed.

The basic issues in the design of a distributed operating system are the same as in a traditional operating system, viz., process synchronisation, deadlocks, scheduling, file systems, interprocess communication, memory and buffer management, failure recovery, etc. However, several idiosyncrasies of a distributed system, namely, the flack of both shared memory and a physical global clock, and unpredictable communication

delays make the design of distributed operating systems much more difficult.

Network operating systems focus on the use of remote services and resources existing on a network of computer systems. Distributed operating systems focus on effective utilization of resources in distributed computing environments.

Distributed systems provide many advantages concerning cost-effectiveness of both computations and resources. The primary advantages are:

Resource sharing
Reliability
Communication
Incremental growth

Resource sharing has been the main motivation for distributed systems. The earliest form of a distributed system was a computer network which enabled the use of specialized hardware and software resources by geographically distant users. Resource sharing continues to be an important aspect of distributed systems today. However, the nature of distribution and sharing of resources has changed due to advances in networking technology. Sharing of resources is now equally meaningful in a local area network (LAN), for example sharing laser printers in the lab.

One aspect of reliability is availability of a resource despite failures in a system. A distributed environment can offer enhanced availability of resources through redundancy of resources and communication paths. For example, availability of a disk resource can be increased by having two or more disks located at different sites in the system. If one disk is unavailable due to a disk or site failure, a program can use some other disk. The availability of a data resource, e.g., a file, can be similarly enhanced by keeping copies of the file at various sites in the system.

Communication between users at different locations is greatly facilitated using a distributed system. There are two important aspects to communication. First, users have unique id's ill a distributed system. Their use ill communication automatically invokes the security mechanisms of the OS, thereby ensuring privacy and authenticity of communication. Second, use of a distributed system also implies continued availability of communication when users migrate to different sites of it system.

Distributed systems are capable of incremental growth, i.e., the capabilities of a system (e.g. its processing power) can be enhanced at a price proportional to the nature and size of the enhancement. A major advantage of this feature is that enhancements need not be planned in advance. This is in contrast to the classical mainframe architectures where enhancements often took the form of up gradation, that is, replacement of subsystems by more powerful ones-hence enhancement costs tended to be disproportionate to the nature and size of an enhancement.

Distributed systems today cover a wide spectrum of computer hardware, software and topological configurations; resource sharing services range from off-line access to real- time access and topologies vary from locally distributed to geographically distributed.

Multiprocessor Operating Systems

A typical multiprocessor system consists of a set of processors that share a set of physical memory blocks over an interconnection network. Thus, a multiprocessor system is a1ightly coupled system where processors share an address space. A multiprocessor operating system controls and manages the hardware and software resources such that users view the entire systems as a powerful uniprocessor system; a user is not aware of the presence of multiple processors and the interconnection network.

The basic issues ill the design of a multiprocessor operating system are the same as ill a traditional operating system. However, tile issues of process synchronisation, task scheduling, memory management, and protection and security become more complex because the main memory is shared by many physical processors.

3.4.2 Application Driven Operating System

A brief discussion of three operating systems in the category or the Architecture Driven OS follows:

Multimedia Operating System

The operating system is the shield of the computer hardware against all software components. It provides a comfortable environment for the execution of programs, and it ensures effective utilisation of the computer hardware. The operating system offers various services related to the essential resources of a computer: CPU, main memory, storage and all input and output devices.

For the processing of audio and video, multimedia application demands that humans perceive these media ill a natural, error-free way. These continuous media data originate at sources like microphones, cameras and files. From these sources, the data are transferred to destinations like loudspeakers, video windows and files located at the same computer or at a remote station. On the way from source to sink, the digital data are processed by at least some type of move, copy or transmit operation. In this data manipulation process there are always many resources which are under the control of the operating system. The integration of discrete and continuous multimedia data demands additional services from many operating system components.

The major aspect in this context is real-time processing of continuous media data. Process management must take into account the timing requirements imposed by the handling of multimedia data. Appropriate scheduling methods should be applied. In contrast to the traditional real-time operating systems, multimedia operating systems also have to consider tasks without hard timing restrictions under the aspect of fairness.

To obey timing requirements, single components are conceived as resources that are reserved prior to execution. This concept of resource reservation has to cover all resources on a data path, i.e. all resources that deal with continuous media. It also may affect parts of the application that process continuous media data. In distributed systems, for example, resource management also comprises network capacity.

The communication and synchronization between single processes must meet the restrictions of real-time requirements and timing relations among different media. The main memory is available as a shared resource to single processes.

In multimedia systems, memory management has to provide access to data with a guaranteed timing delay and efficient data manipulation functions. For instance, physical data copy operations must be avoided due to their negative impact on performance; buffer management operations (such as are known from communication systems) should be used.

Database Operating System

Database systems place special requirements on operating systems. These requirements have their roots in the specific environment that database systems support. A database system must support: the concept of a transaction; operations to store, retrieve, and manipulate a large volume of data efficiently; primitives for concurrency control, and

system failure recovery. To store temporary data and data retrieved from secondary storage, it must have a buffer management scheme.

Real-time Operating System

The main characteristics of the real-time systems are the correctness of the computation. This correctness does not apply to error-free computation, but also on the time in which the result is processed and produced. Therefore, the real-time systems must execute its critical workload in time to prevent failures. Examples of applications requiring support of real-time systems are process control, air traffic control, guidance of missile systems, etc.

Real-time systems also place special requirements on operating system, which have their roots in the specific application that the real-time system is supporting. A distinct feature of real-time systems is that jobs have completion deadlines. A job should be completed before its deadline to be of use (in soft real-time system) or to avert a disaster (in hard real-time systems). The major issue in the design of real-time operating systems is the scheduling of jobs in such a way that a maximum number of jobs satisfy their deadlines. Other issues include designing languages and primitives to effectively prepare and execute a job schedule.

3.5 Characteristics of Modern Operating System

Over the years, there has been a gradual evolution of operating system structure and capabilities. However, in recent years a number of new design elements have been introduced into both new operating systems and new releases of existing operating systems that create a major change in the nature of operating systems. These modem operating systems respond to new developments in hardware and new applications. Among the key hardware drivers are multiprocessor machines, greatly increased machine speed, high speed network attachments, and increasing the size and variety of memory storage devices. In the application arena, multimedia applications, Internet and Web access, and client/server computing have influenced operating system design.

The rate of change in the demands on operating systems requires not just modifications and enhancements to existing architectures but new ways of organising the operating system. A wide range of different approaches and design elements has been tried in both experimental and commercial operating systems, but much of the work fits into the following categories:

Microkernel architecture
Multithreading
Symmetric multiprocessing

Most operating systems, until recently, featured a large monolithic kernel. Most of what is thought of as operating system functionality is provided in these large kernels, including scheduling, file system, networking, device drivers, memory management and more. Typically, a monolithic kernel is implemented as a single process with all elements sharing the same address space. A microkernel architecture assigns only a few essential functions to the kernel, including address spaces, interprocessor communication (IPC), and basic scheduling. Other OS services are provided by processes, sometimes called servers that run in user mode and are treated like any other application by the microkernel. This approach decouples kernel and server development. Servers may be customised to specific application or environment requirements. The micro kernel approach simplifies implementation, provides flexibility, and is well suited to a distributed environment. In essence, a microkernel interacts with local and remote server processes in the same way, facilitating construction of distributed systems.

5.5.1 Microkernel Architecture

Microkernel architecture assigns only a few essential functions to the kernel, including address spaces, interprocess communication (IPC), and basic scheduling. Other as services are provided by processes, sometimes called servers that run in user mode and are treated like any other application by the microkernel. This approach decouples kernel and server development. Servers may be customised to specific application or environment requirements. The micro kernels approach simplifies implementation, provides flexibility, and is well suited to a distributed environment. In essence, a microkernel interacts with local and remote server processes in the same way, facilitating construction of distributed systems.

3.5.2 Multithreading

Multithreading is a technique in which a process executing an application is divided into threads that can run concurrently. We can make the following distinction:

> Threads: A dispatchable unit of work. It includes a processor context (which includes the program counter and stack pointer) and its own data area for a stack (to enable subroutine branching). A thread executes sequentially and is interruptable so that the processor can turn to another thread.

Process: A collection of one or more threads and associated system resources (such as memory containing code and data, open files, and devices). This corresponds closely to the concept of a program in execution. By breaking a single application into multiple threads, the programmer has great control over the modularity of the application and tile timing of application related events.

Multithreading is useful for applications that perform a number of essentially independent tasks that do not need to be serialised. An example is a database server that listens for and processes numerous client requests. With multiple threads running within the same process, switching back and forth among threads involves less processor overhead than a major process switch between different processes.

Threads are also useful for structuring processes that are part of the as kernel as described in subsequent chapters.

3.5.3 Symmetric Multiprocessing

Until recently, virtually all single-user personal computers and workstations contained a single general-purpose microprocessor. As demands for performance increase and as the cost of microprocessors continues to drop, vendors have introduced computers with multiple microprocessors. To achieve greater efficiency and reliability, one technique is to employ symmetric multiprocessing (SMP), a term that refers to a computer hardware architecture and also to the operating system behaviour that reflects that architecture. A symmetric multiprocessor can be defined as a standalone computer system with the following characteristics:

1) There are multiple processors.
2) These processors share the same main memory and I/O facilities, interconnected by a communications bus or other internal connection scheme.
3) All processors can perform the same functions (hence the term Symmetric).

The operating system of an SMP schedules processes or threads across all of the processors. SMP has a number of potential advantages over uniprocessor architecture, including the following:

Performance: If the work to be done by a computer can be organised so that some portions of the work can be done in parallel, then a system with multiple processors will yield greater performance than one with a single processor of the same type.

With multiprogramming, only one process can execute at a time; meanwhile all other processes are waiting for the processor. With multiprocessing, more than one process can be running simultaneously, each on a different processor.

Availability: In a symmetric multiprocessor, because all processors can perform the same functions, the failure of a single processor does not halt the machine. Instead, the system can continue to function at reduced performance.

Incremental growth: A user can enhance the performance of a system by adding an additional processor.

Scaling: Vendors can offer a range of products with different price and performance characteristics based on the number of processors configured in the system.

It is important to note that these are potential rather than guaranteed benefits. The operating system must provide tools and functions to exploit the parallelism in an SMP system.

Multithreading and SMP are often discussed together, but the two are independent facilities. Even on a uniprocessor machine, multithreading is useful for structuring applications and kernel processes. An SMP machine is useful for non-threaded processes, because several processes can run in parallel. However, the two facilities complement each other and can be used effectively together.

An attractive feature of an SMP is that the existence of multiple processors is transparent to the user. The operating system takes care of scheduling of threads or processes on individual processors and of synchronisation among processors. A different problem is to provide the appearance of a single system for a cluster of separate computers -a multi-computer system. In this case, we are dealing with a collection of entities (computers), each with its own main memory, secondary memory, and other I/O modules. A distributed operating system provides the illusion of a single main memory space and a single secondary memory space, plus other unified access facilities, such as a distributed file system. Although clusters are becoming increasingly popular, and there are many cluster products on the market, the state of the art for distributed operating systems lags that of uniprocessor and SMP operating systems.

The most recent innovation in operating system design is the use of object oriented technologies. Object oriented design lends discipline to the process of adding modular extensions to a small kernel. At the

operating system level, an object-based structure enables programmers to customise an operating system without disrupting system integrity. Object orientation also eases the development of distributed tools and a full- blown distributed operating system.

4.0 CONLUSION

This unit has introduced you to the basic principles and concept of operating system. Discussing extensively the evolution of operating system, several design approaches to operating system, classification and characteristics of operating system independent of any OS environment.

5.0 SUMMARY

The operating system is an essential component of system software, which consists of procedures for managing computer resources. Initially computers were operated from the front console. System software such as Assemblers, Loaders and Compilers greatly helped in software development but also required substantial setup time. To reduce the setup time an operator was hired and similar jobs were batched together.

Batch systems allowed automatic job sequencing by a resident monitor and improved the overall utilisation of systems greatly. The computer no longer had to wait for human operations- but CPU utilisation was still low because of slow speed of I/O devices compared to the CPU. A new concept buffering was developed to improve system performance by overlapping the input, output and computation of a single job. Spooling was another new concept in improving the CPU utilisation by overlapping input of one job with .the computation and output of other jobs.

Operating systems are now almost always written in a higher-level language (C, PASCAL, etc.). UNIX was the first operating system developed in C language. This feature improves their implementation, maintenance and portability.

The operating system provides a number of services. At the lowest level, there are system calls, which allow a running program to make a request from the operating system directly. At a higher level, there is a command interpreter, which supports a mechanism for a user to issue a request without writing a program.

In this unit, we began with tracing the evolution of the operating system through serial processing, batch processing and multiprogramming. We

also presented different operating system models: Layered structured, Kernel based, virtual machine system and client server model.

At the end we presented classification and characteristics of emerging operating systems.

6.0 TUTOR-MARKED ASSIGNMENT

1) What is a system call?
2) Define the essential difference between:

   a) Spooling
   b) Buffering

3) What are the basic design issues in the distributed operating systems?
4) What is the basic difference between Network operating systems and Distributed operating systems?

7.0 REFERENCES/FURTHER READINGS

Operating System Concepts by Abraham Silberschatz and James L. Peterson, Addision Wesley.

Operating Systems Design and Implementation by Andrew S. Tanenbaum, Prentice Hall of India.

UNIT 3 INTRODUCTION TO NETWORKING CONCEPT

CONTENTS

1.0 INTRODUCTION

A network can consist of two computers connected together on a desk or it can consist of many Local Area Networks (LANs) connected together to form a Wide Area Network (WAN) across a continent. In simple terms it is an interconnected set of some objects. For decades, we have been familiar with the Radio, Television, Railways, Banks and various other types of networks. In recent years, the computer network, a new form of network is becoming more and more visible in our day-to-day life. A computer network is an interconnected set of autonomous computers.

Autonomous means each of them can function independent of others, i.e., each computer has individual processors. Simply we can say each computer (terminal, node) should not be a dump terminal. The key is that two or more computers are connected together by a medium and are sharing resources. These resources can be files, printers, hard drives, CPU or data. By using a computer network, people can send and receive back information more quickly.

2.0 OBJECTIVES

After going through this unit, you should be able to:

define what a network is?
understand what is the need of a computer network and the applications of networks
list types of networks, topologies and mediums finally
Learn how devices are connected though repeater, bridges, router, Gateway.

3.0 MAIN CONTENT

3.1 Why Computer Networks?

Computer Networks offer a number of advantages to individuals and organization. Some of these are:

a) Communication medium: It offers a powerful communication medium among a group of people widely separated on the earth.
b) Resource Sharing: Resources like files, printers, hard drives, or CPU can be shared through a computer network.
c) Higher Reliability: If one computer is down; its workload can be taken over by tile other computer. So it offers higher reliability than a centralized computing environment.
d) Higher flexibility: A heterogeneous system can be connected in a computer network, by which users get better flexibility.
e) Sealable: Computers and other equipments can be gradually added to satisfy the need of an organisation at different points of time, without changing tile original network.

Applications of Computer Network

a) Electronic Mail (e-mail or Email). The most widely used network application is E-mail, which is forwarding of electronic files to an electronic post office for the, recipient to pick up.

b) Scheduling programs allow people across the network to schedule appointments directly by calling up their fellow worker's schedule and selecting a time!
c) Videotext is the capability having a two-way transmission of picture and sound. Games like distance education lectures, etc. use videotext.
d) Groupware is the latest network application. It allows user groups to share \ documents, schedules databases
e) Teleconferencing allows people in different regions to "attend" meetings using, telephone lines.
f) Automated Banking Machines allow banking transactions to be performed everywhere: at grocery stores, drive-in machines etc.
g) Information Service Providers provide connections to the Internet and other information services.
h) Telecommuting allows employees to perform office work at home by "Remote Access" to the network.
i) Value Added Networks are common carriers such as ERNET, Satyam, VSNL etc. (they can be private or public companies) who provide additional leased line connections to their customers. These can be Frame Relay, ATM (Asynchronous Transfer Mode), X.25, etc.
j) Marketing and sales Marketing professionals use computer network to collect, exchange and analyse data relating to customer needs.

3.2 The Topologies

The topology is the geometric arrangement (either physically or logically) of the linking devices (usually called nodes) and the links, connecting the individual computers or nodes together. Five basic topologies:

1) Bustopology
2) Ringtopology
3) Star topology
4) Mesh topology
5) Combined topologies.

1) The Bus Topology

In the bus topology there is a single bus that carries all the data to the entire network. A bus is a single continuous communication cable to which all the computers are connected. A cable or bus runs throughout the office to which all the workstations are connected. The bus topology is also known as *linear bus.*

When one workstation wants to talk to another the message or signal travels down the bus in both directions. Each one reads the message to see if it matches its address. The bus topology is a passive topology. It means that the computers connected to the bus amplify the signal on the bus.

The main advantage of bus topology is that it is quite easy to set up. Any workstation can be easily moved to another location as bus runs throughout the office. Another benefit of this layout is that if one computer on the bus fails, it does not affect the rest of the traffic on the bus.

A network with bus topology cannot become too big as all the traffic is on a single bus. The entire network can be down only if the bus has a break. The open ends of bus must be terminated to prevent signal bounce. If one or both ends of the bus are not terminated, the whole network can be down.

Disadvantages Include difficult reconfiguration and fault isolation

2) The Ring Topology

In the ring topology all the workstations are connected in the shape of a ring. The ring does not have an end. It is made up of short segments that connect one PC to the next and so on, until all the computers are joined in a circle. The signals travel only in one direction and from one PC to the next until it reaches the appropriate node. It is also difficult to move a workstation or to add more computers to an existing ring.

In ring topology the wiring for a ring could be arranged in a circle throughout a building or a group of buildings. The signal travels in one direction only from one computer to the next. The ring topology is an active topology. Each computer boosts the signal (like a repeater) and passes to the next computer till it reaches the destination computer. A drawback of this topology is that if one computer fails, the entire network is down. However, now some ring networks are so designed that a faulty workstation is automatically bypassed. Another drawback is that the traffic is in only one direction. This topology is not used for a large number of nodes.

3) The Star Topology

In the star topology all the stations are connected to a central computer or hub creating a star configuration. The devices are not directly linked to each other Messages pass from the nodes to the hub, where they are processed or passed along to another node. The hub controls the traffic

on the network. If the hub fails, the entire network becomes inoperative, but if a node fails it does not affect the rest of the traffic on tl1e network. All client/server networks use this topology. But since cable from each node must be connected to a central hub, the length of total wiring required increases very much. A hub can be an active hub or a passive hub. A passive hub simply organizes the wiring and works just like a wiring panel for various connections. It does not need any power connection. An active hub does what a passive hub does, but besides this it regenerates and retransmits the signals the way a repeater does. An active hub needs a power connection.

4) Mesh Topology

In a mesh topology, every node has a dedicated point-to point link to every other node. Simply dedicated means that the links carry traffic only between the two nodes. So mesh topology does not have traffic congestion problems every node has n-1 link, for a fully connected mesh topology having n nodes. So the total number of links will be n (n-I). This also means that every node has (n-l) I/O ports.

Advantages of Mesh topology

a) Use of dedicated links guarantees that each connection can carry its own data load. Thus eliminates the traffic problem.
b) If one link fails, it does not affect the rest of network. This means it is robust.
c) Point to point links makes fault identification and fault isolation easy.
d) Privacy or security is high, since the other link cannot gain access to the dedicated link where the message is travelling.

Disadvantages of Mesh Topology

a) More cabling and I/O ports are required, because every node must be connected to every other node.
b) Cost is very high, because more number of node and cabling required.
c) Installation and reconfiguration is difficult.

5) Combined Topologies

A network does not have to stick with one topology. Any two topologies or all the topologies can be used in a network. For example, a hub may be connected to other hubs using a bus and the work3tations may be connected by a star.

Two main hybrid topologies are:

1) The Star Bus Topology

The star bus topology is a combination of bus and star topologies. In this topology the hubs of many star topology networks are linked together with a linear bus or trunk. For example, we want to link three star topology-networks together. In each network, the nodes are connected to its own hub. Thus we have three hubs. These three hubs are connected by a bus topology.

2) The Star Ring Topology

In this topology the hubs of many star topology networks are connected to another main hub in a star pattern. Thus if we have three star topology networks, then the three hubs of the networks are connected to a fourth hub (main hub) in star pattern.

3.3 Characteristics of the OSI Layers

Thee are primarily two architectural models for designing computer networks. OSI model & TCP/IP model. In this section we shall discuss one such model.

The seven layers of the OSI reference model can be divided into two categories: upper layers and lower layers.

The *upper layers* of the OSI model deal with application issues and generally are implemented only in software. The highest layer, the application layer, is closest to the end user. Both users and application layer processes interact with software applications that contain a communications component. The term upper layer is sometimes used to refer to any layer above another layer in the OSI model.

The *lower layers* of the OSI model handle data transport issues. The physical layer and the data link layer are implemented in hardware and software. The lowest layer, the physical layer, is closest to the physical network medium (the network cabling, for example) and is responsible for actually placing information on the medium.

*Figure I i*Illustrates the division between the upper and lower OSI layers.

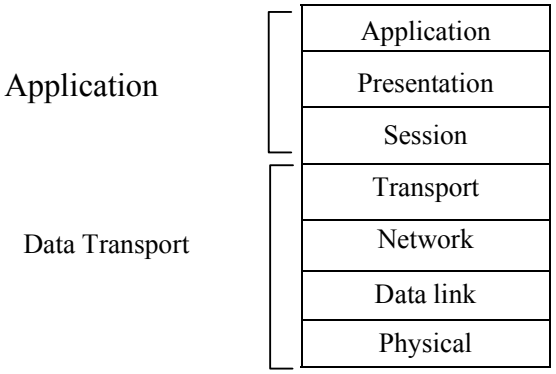


Figure 1: Two Sets of Layers Make Up the OSI Layers

Each layer has well defined functionalities and standard protocols for implementing these functionalities.

3.4 OSI Model and Communication between Systems

Information being transferred from a software application in one computer system to a software application in another must pass through the OSI layers. For example, if a software application in System A has information to transmit to a software application in System B, the application program in System A will pass its information to the application layer (Layer 7) of System A. The application layer then passes the information to the presentation layer (Layer 6), which relays the data to the session layer (Layer 5), and so on down to the physical layer (Layer I). At the physical layer, the information is placed on the physical network medium and is sent across the medium to System B. The physical layer of System B removes the information from the physical medium, and then its physical layer passes the information up to the data link layer (Layer 2), which passes it to the network layer (Layer 3), and so on, until it reaches the application layer (Layer 7) of System B. Finally, the application layer of System B passes the information to the recipient application program to complete the communication process.

3.5 Interaction between OSI Model Layers

A given layer in the OSI model generally communicates with three other OSI layers: the layer directly above it, the layer directly below it, and its peer layer in other networked computer systems. The data link layer in System A, for example, communicates with the network layer of System A, the physical layer of System A, and the data link layer in System B. *Figure* 2 illustrates this example.

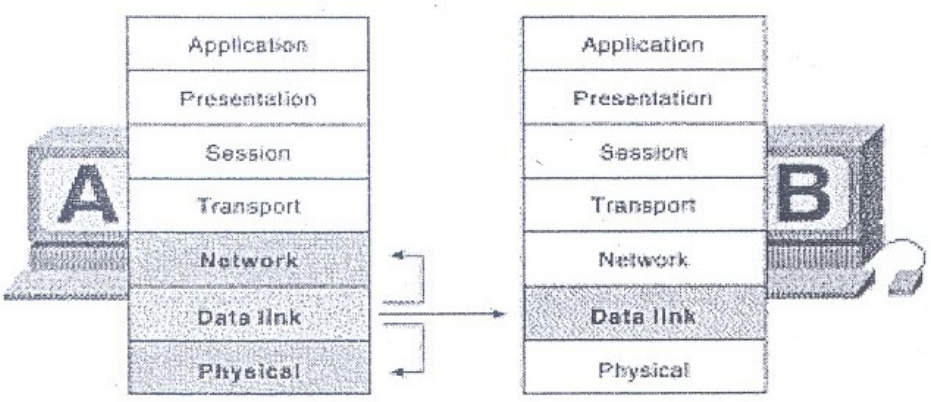


Figure 2: OSI Model Layers Communicate with Other Layers

3.6 Protocols

Just like human beings need to have a common languages to speak to one another, digital devices and computers also need to have common 'languages' to be able to communicate with one another. The binding function of 'common language' in digital communication is performed by communication protocols.

A communication protocol is a set of conventions or rules that must be adhered to by both communicating parties to ensure that information being exchanged between two parties is received and interpreted correctly. Without a protocol, two devices may be connected but not communicating, just as a person speaking Hindi cannot be understood by a person who speaks only Tamil.

A protocol defines the following three aspects of communication.

1) **Syntax:** The format of data being exchanged, character set used, type of error correction used, type of encoding scheme (e.g., signal level) being used. For example, a simple protocol may use first eight bit for address of sender, the second eight bit for address of receiver and the rest of bit for message itself.

2) **Semantics:** Type and order of messages used to ensure reliable and error free information transfer.

3) **Timing:** Define data rate selection and correct timing for various events during data transfer. Simply when data should be sent and how fast they can be sent.

It has been accepted that the complexity of writing communication software can be reduced by adapting the principle of protocol layering. The idea here is to partition communication functions into a vertical set

of layers. Each layer performs a related set of functions. Division of work between layers is done in such a way that they are manageable and provide a logical interface and break point. Each communication layer provides certain services to layers above it and relies on the next lower layer to perform more primitive functions. Each layer hides internal details from other layers. Thus dividing the communication problem into several layers reduces its complexity and makes the work of developing communication software a lot easier and error free.

3.7 Types of Networks

The differences among different types of computer networks are usually based on perspective. For example, computer networks are frequently classified by the geographical area (LAN, MAN, WAN), their topologies (e.g., point to point or broadcast), or the type of communication path they use and the manner in which data are transmitted across this path (e.g., circuit-switched and packet-switched).

Computer networks are classified by the geographical area are:

1) Local Area Networks (LANs)
2) Metropolitan Network (MANs)
3) Wide Area Networks (WANs)

3.7.1 Local Area Networks (LANs)

LANs (local area networks), as shown in *Figure* 3 are privately-owned networks that connect computers and resources together in a building or buildings that are close together. LAN plays an important part in everyday functioning of schools, businesses, and government.

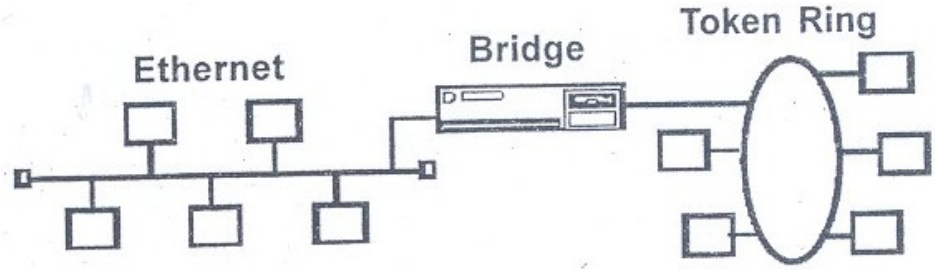


Figure 3: Local Area Network in a building

A *LAN* is a high-speed data network that covers a relatively small geographic area. It typically connects workstations, personal computers, printers, servers, and other devices, so that devices can communicate with each other to share resources. LANs offer computer users many advantages, including shared access to devices and applications, file

exchange between connected users, and communication between users via electronic mail and other applications.

A Local Area Network is a system of computers that share resources such as disk drives, printers, data, CPU power, fax/modem, applications, etc. They usually have distributed processing, which means that there are many desktop computers distributed around the network and that there is no central processor machine (mainframe).

Location: In a building or individual rooms or floors of buildings or nearby buildings. Can be campus wide like a college or university.

LAN Characterization

There are four key areas that characterize a local area network. These are:

1) Transmission Medium
2) Access Method
3) Topology
4) Signaling Techniques

LAN Media-Access's Methods

Media contention occurs when two or more network devices have data to send at the same time. Because multiple devices cannot talk on the network simultaneously, some type of method must be used to allow one device access to the network media at a time. This is done in two main ways: carrier senses multiple accesses collision detects (CSMA/CO) and token passing.

In networks using CSMA/CO technology such as Ethernet, network devices contend for the network media. When a device has data to send, it first listens to see if any other device is currently using the network. If not, it starts sending its data. After finishing its transmission, it listens again to see if a collision occurred. A collision occurs when two devices send data simultaneously. When a collision happens, each device waits a random length of time before resending its data. In most cases, a collision will not occur again between the two devices. Because of this type of network contention, the busier a network becomes, the more collisions occur. This is why performance of Ethernet degrades rapidly as the number of devices on a single network increases.

In token-passing networks such as Token Ring and FDDI, a special network packet called a token is passed around the network from device to device. When a device has data to send, it lust wait until it has the

token and then send its data. When the data transmission is complete, the token is released so that other devices may use the network media. The main advantage of token-passing networks is that they are deterministic. In other words, it is easy to calculate the maximum time that will pass before a device has the oppo1"tunity to send data. This explains the popularity of token- passing networks in some real-time environments such as factories, where machinery must be capable of communicating at determinable intervals.

For CSMA/CO networks, switches segment the network into multiple collision domains. This reduces the number of devices per network segment that must contend for the media. By creating smaller collision domains, the performance of a network can .be increased significantly without requiring addressing changes.

Normally CSMA/CO networks are half-duplex, meaning that while a device sends information, it cannot receive at the same time. While that device is talking, it is incapable of also listening for other traffic. This is much like a walkie-talkie. When one person wants to talk, he presses the transmit button and begins speaking. While he is talking, no one else on the same frequency can talk. When the sending person is finished, he releases the transmit button and the frequency is available to others.

When switches are introduced, full-duplex operation is possible. Full-duplex works much like a telephone-you can listen as well as talk at the same time. When a network device is attached directly to the port of a network switch, the two devices may be capable of operating in full-duplex mode. In full-duplex mode, performance can be increased, but not quite as much as some like to claim.. However, full-duplex operation does increase the throughput of most applications because the network media is no longer shared. Two devices on a full-duplex connection can send data as soon as it is ready.

Token-passing networks such as Token Ring can also benefit from network switches. In large networks, the delay between turns to transmit may be significant because the token is passed around the network.

LAN Transmission Methods

For Transmission, LAN usually broadcast their message to all hosts on the LAN The address in the packet or frame enables the destination to receive the packet, while the rest *of* the hosts ignore the broadcast message.

LAN data transmissions fall into three classifications: unicast, multicast, and broadcast. In each type of transmission, a single packet is sent to one or more nodes.

In a unicast transmission, a single packet is sent from the source to a destination on a network. First, the source node addresses the packet by using the address *of* the destination node. The package is then sent onto the network, and finally, the network passes the packet *to* its destination.

A multicast transmission consists of a single data packet that is copied and sent to a specific subset of nodes on the network. First, the source node addresses the packet by using a multicast address. The packet is then sent into the network, which makes copies of the packet and sends a copy to each node that is part of the multicast address.
A broadcast transmission consists of a single data packet that is copied and sent to all nodes on the network. In these types of transmissions, the source node addresses the packet by using the broadcast address. The packet is then sent on to the network, which makes copies of the packet and sends a copy to every node on the network.

Topology: The most common LAN topologies are bus, ring, and star.

3:7.2 Metropolitan Networks (MANs)

Metropolitan Area Networks (MANs), as shown in *Figure* 4, are networks that connect LANstogether within a city.

Figure 4: MANs use Local Exchange Carriers

The main criterion for a MAN is that the connection between LANs is through a local exchange carrier (the local phone company) .The protocols that are used for MANs are quite different from those used for LANs (except for ATM, which can be used for both under certain conditions). It has been distinguished as a separate type of network; because of the specific standard known as Distributed Queue Double Bus (DQDB) that has been adopted for MAN. The DQDB comprises two unidirectional buses for connecting computers.

A Metropolitan Area Network is a system of LANs connected throughout a city *(Figure* 5) or metropolitan area. MAN can be

considered as a bigger version of a LAN, typically covering a city. It can be either public or privately owned. MANs have the requirement of using telecommunication media such as voice channels or data channels. Branch offices are connected Jo head offices through MANs. Examples of organizations that use MANs are universities and colleges, hotels, and banks.

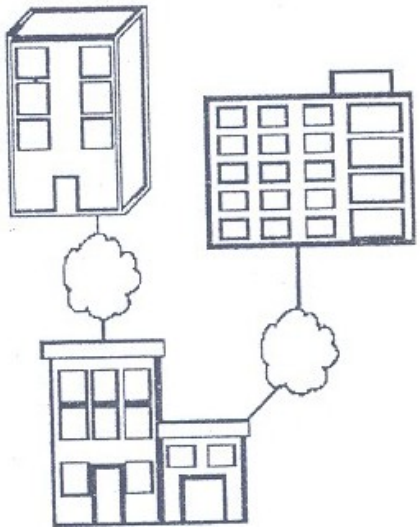


Figure 5: Location: Separate buildings distributed throughout a city

3.7.3 Wide Area Networks (WANs)

Wide Area Networks (WANs) connect LANs together between cities *(Figure* 6).

Communication is usually done through public communication systems such as telephone line, fiber optic cable or wireless technology.

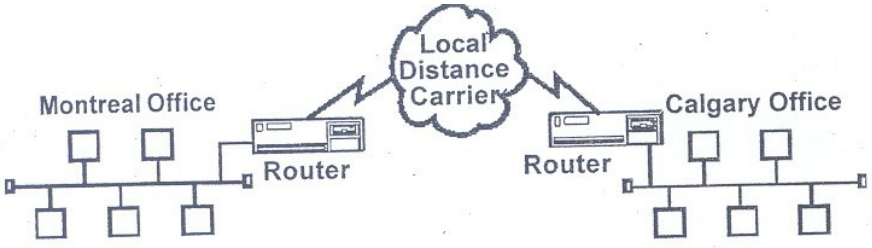


Figure 6: WANs use Long Distance Carriers

A Wide Area Network is a network system connecting cities, countries, or continents together *Figure* 7. WANs are connected together using one of the telecommunications media.

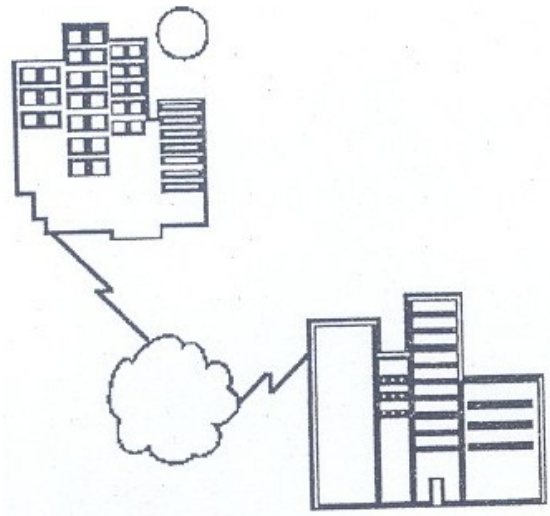


Figure 7: City to City, Cross a Country or Cross a Continent

The main difference between a MAN and a WAN is that the WAN uses Long Distance Carriers. Otherwise the same protocols and equipment are used as in MAN.

Main differences between a LAN and a WAN are given in the following Table 1.

Table l: Difference between LAN and MAN

Wide area Network	Local Area Network
Distance up to thousands of Kilometer Typical data rates between 9.6k to Mbps Higher error rates (I in 10^5) Often use analog circuits from the telephone systems Generally has point-to-point link with topologies mesh and star May be managed by organizations independent of users WAN uses complex protocols and extensive error recovery mechanisms Number of node computers has no theoretical limit and could be very large. The practical limit comes from addressing schemes used to identify individual system on the network and other resource constrains.	Within a local site High band width between 1-16 Mbps Lower error rate (I in 10^9) Use digital signaling over private cables Generally use bus or ring topology Managed by the same common company which owns the computers connected to LAN LAN uses simple protocols and does not employ any retransmission strategy for lost frames Number of host on a LAN is limited (usually up to 1024)

3.8 Medium

Data communication system is made up of following components *(Figure* 8).

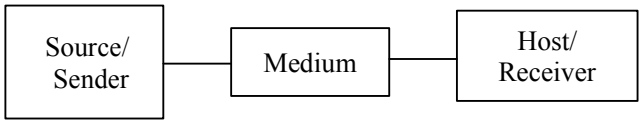


Figure 8: Data Communication-l

Source Sender: Sender can be Terminals, Computers, Mainframes, workstation, telephone hand set, video camera. Main function of sender is to send data (message) to receiver.

The communications stream through which the data is being transmitted. Examples are:

Cables
Microwave Link
Fiber optic Link
Radio Frequencies (RF)
Infrared Wireless

Receiver: The receiver receives the message (data) from sender. Receiver can be Terminals, Computers, Mainframes, workstation, telephone hand set, printer, television and so on.

Protocol: A communication protocol is a set *of* conventions or rules that must be adhered to by both communicating parties to ensure that information being exchanged between two parties is received and interpreted correctly.

Message: Message consists of text, numbers, pictures, sound, or video- or any combination of these to be transmitted from sender to receiver.

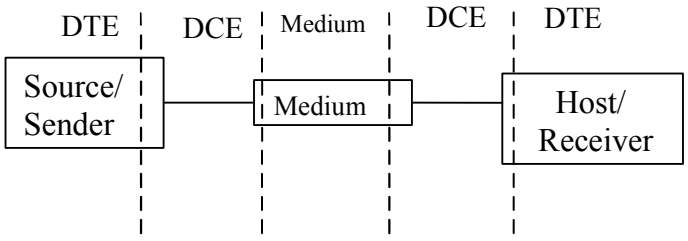


Figure 9: Data Communication-II

DCE: The interface between the Source and the Medium, and the Medium and the Receiver is called the DCE (Data Communication Equipment) *(Figure* 9) and is a physical piece of equipment.

DTE: Data Terminal Equipment is the telecommunications name given to the source and receiver's equipment. It is any device that is a source of or destination for binary digital data.

The DTE generates the data and passes them, through DCE. The DCE takes the generated data by DTE and converts them to an appropriate signal. Then this signal is introduced to telecommunication link. Most commonly used DCE is a modem, discussed in the section.

3.9 Dataflow

Data flow (transmission mode) is the flow of data between two points. There are three types of dataflow *(Figure 10):* simplex, half duplex and full duplex.

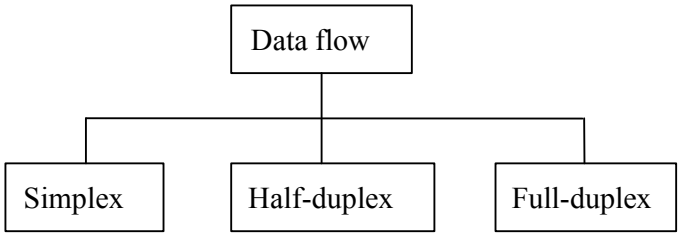


Figure 10: Data Flow

Simplex: data flows in only one direction (Figure 11) on the data communication line (medium). Examples are radio and television broadcasts. They go from the TV station to your home television.

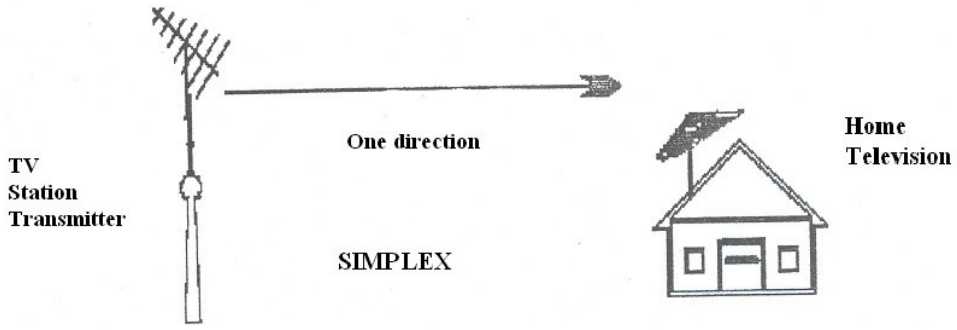


Figure 11: Simplex

Half-Duplex: Data flows in both directions but only one direction at a time *(Figure J* 2) on the data communication \in~. Each of the stations can both transmit and receive. For example; a conversation on walkie-

talkie is a half-duplex data flow. Each person takes turns talking. If both talk at once -nothing occurs!

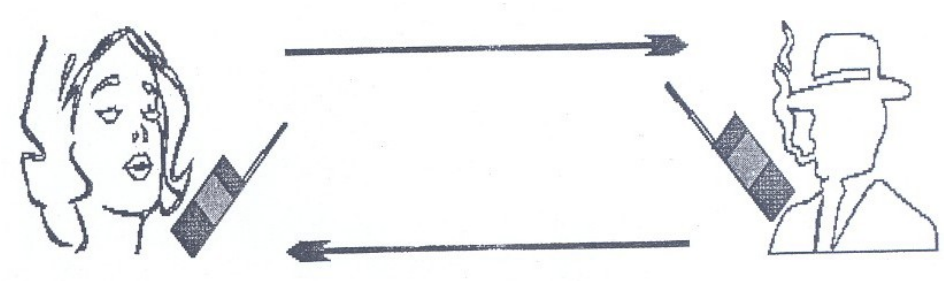


Figure 12: Half-Duplex

Bi-directional but only one direction at a time!

HALF-DUPLEX

Full-Duplex: data flows in both directions at the same time *(Figure J* 3). The system is configured to flow data in both directions.

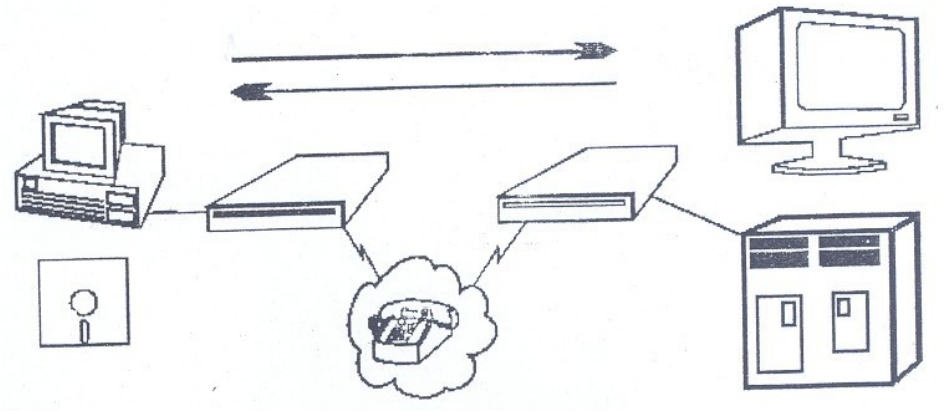


Figure 13: Full Duplex

Bi'-directional both directions simultaneously!

Another example of full duplex is two-way street with traffic flowing in both directions at the same time.

Modems

A modem (MOdulator/DEModulator) connects a terminal/computer (DTE) to the Voice Channel (dial-up line).

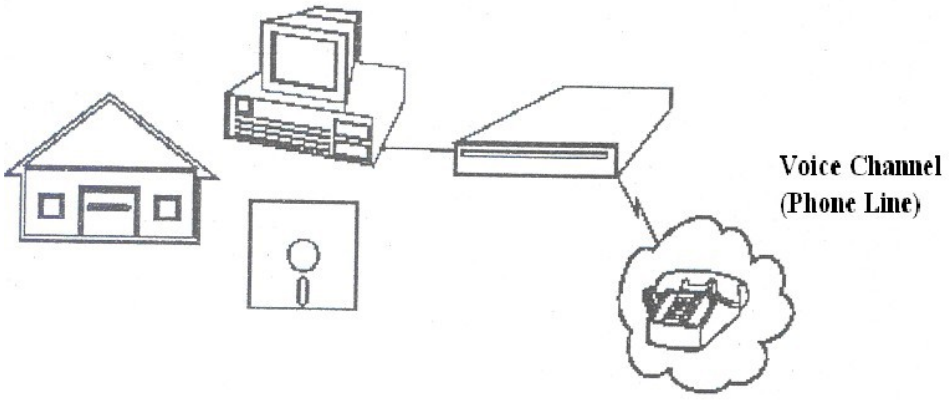


Figure 14: Modems

Basic Definition

The modem (DCE -Data Communication Equipment) is connected between the terminal/computer (DTE -Data Terminal Equipment) and the phone line (voice channel). A modem converts the DTE (Data Terminal Equipment) digital signal to an analog signal {or vice versa) that the voice channel can use.

A modem is connected to the terminal/computer's RS-232 serial port (25 pin male D connector) and the outgoing phone line with an RJII cable connector (the same as on a telephone extension cord). Male connectors have pins, female connectors have sockets.

Digital Connection

The connection between the modem and terminal/computer is a digital connection. A basic connection consists of a Transmit Data (TXD) line, a Receive Data (RXD) line and many hardware handshaking control lines *(Figure* 15).

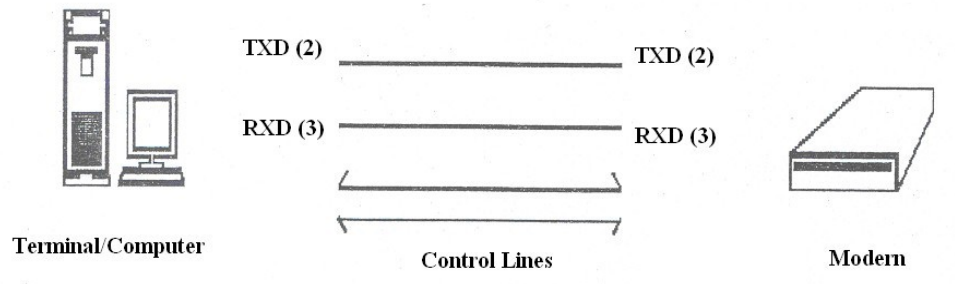


Figure 15: Digital Connection

The control lines determine whose turn it is to talk (modem or terminal), if the terminal/computer is turned on, if the modem is turned on, if there is a connection to another modem, etc.

Analog Connection

The connection between the modem and the outside world (the phone line) is an analog connection *(Figure* 16). The voice channel has a bandwidth of 0-4 kHz but only 300- 3400 Hz is usable for data communications.

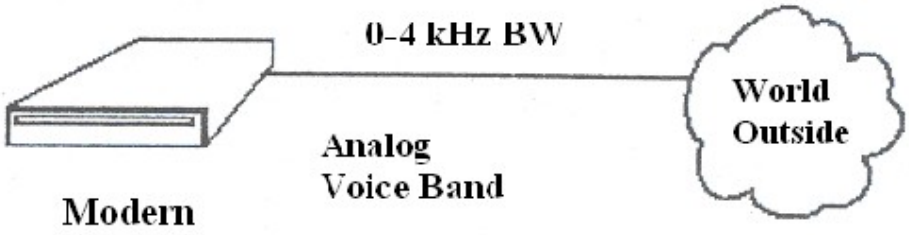


Figure 16: Analog Connection

The modem converts digital information into tones (frequencies) for transmitting through the phone lines.

External/Internal Modems

There are 2 basic physical types of modems: Internal &: External modems. External modems *(figure I7)* sit next to the computer and connect to the serial port using a straight-through serial cable.

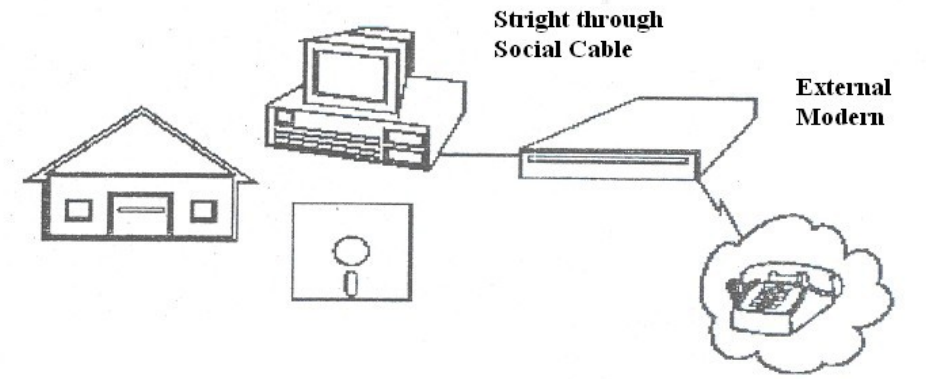


Figure 17: External Modems

Internal modems *(Figure* 18) are a plug-in circuit board that sits inside the computer. It incorporates the serial port on-board. They arc less expensive than external modems because they do not require a case, power supply and serial cable. They appear to the communication programs as if they were an external modem for all practical purposes.

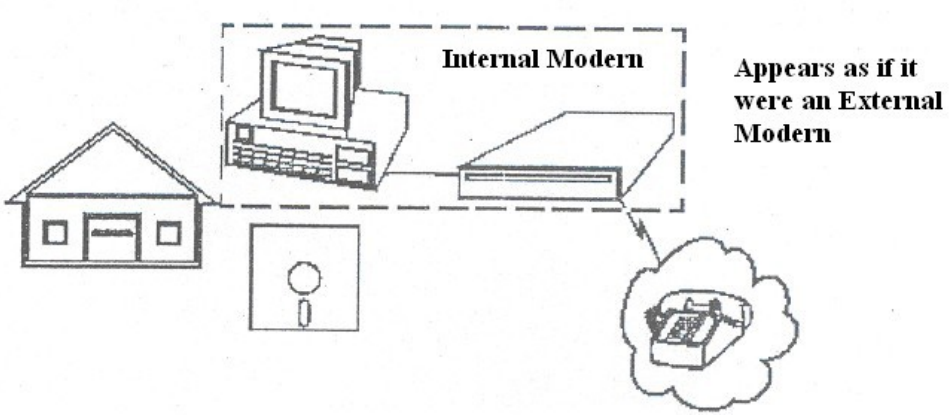


Figure 18: Internal Modems

Modem Types

There are many types of modems, the most common of which are:

1) **Optical Modem:** Uses optical fiber cable instead of wire. The modem converts the digital signal to pulses of light to be transmitted over optical lines (more commonly called a media adapter or transceiver).

2) **Short Haul Modem:** A modem used to transmit data over 30 km or less. Modems we use at home or to connect computers together among different offices in the same building are short haul modems.

3) **Acoustic Modem:** A modem that couples to the telephone handset with what looks like suction cups that contain a speaker and microphones. Used by travelling sales people to connect to hotel phones.

4) **Smart Modem:** A modem with a CPU (microprocessor) on board that uses the Hayes AT command set. This allows auto-answer & dial capability rather than manually dialing & answering.

5) **Digital Modem:** Converts the RS-232 digital signals to digital signals more suitable for transmission. (Also called a media adapter or transceiver).

6) **V.32 Modem:** A milestone modem that uses a 2400-baud modem with 4 bit encoding. This results in a 9600 bps (bits per second) transfer rate.

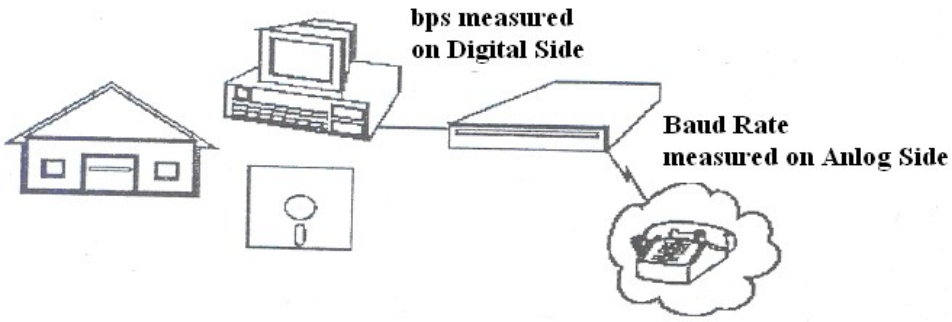


Figure 19: BPS and Baud Rate

Baud (Figure 19) is the speed at which the analog data is changing on the voice r channel and bps is the speed at which the decoded digital data is being transferred.

Features of Modems

1) **Speed:** The speed at which the modem can send data in bps (bits per second). Typical modem speeds are: 300, 600, 1200, 2400, 4800, 9600, 14.4K, 19.2K, 28.8K bps.
2) **Auto Dial/Re Dial:** Smart modems can dial the phone number and auto re dial if a busy signal is received.
3) **Auto Answer:** Most modems have Ring Detect capability and can automatically answer the telephone when an incoming call comes in.
4) **Self-Testing:** Newer modems have self-testing features. They can test the digital connection to the terminal/computer and the analog connection to a remote modem. They can also check the modem's internal electronics.
5) **Voice Over Data:** Voice Over Data modems allow a voice conversation to take place while data is being transmitted. This requires both the source and Networking C destination modems to have this feature.
6) **Synchronous or Asynchronous Transmission:** Newer modems allow a choice of synchronous or asynchronous transmission of data. Normally, modem transmission is asynchronous (we send individual characters with just start and stop bits). Synchronous transmission or packet transmission is used in specific applications.

3.10 Physical Connection

The physical connection determines how many bits (1's or O's) can be transmitted in a single instance of time. If only I bit of information can be transmitted over the data transmission medium at a time then it is considered a serial communication *(Figure 20).*

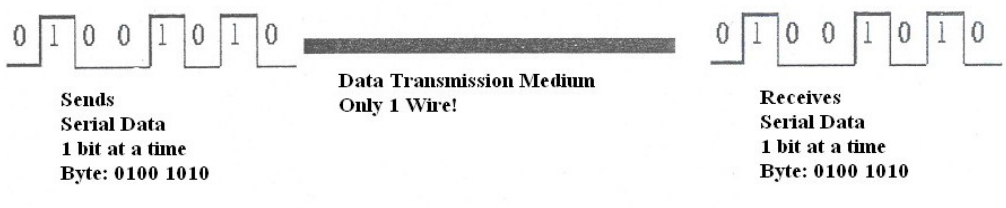


Figure 20: Serial Communication

If more than I bit of information is transmitted over the data transmission medium at a time then it is considered a parallel communication *(Figure* 21). By grouping, we can send data n bits at a time instead of one, through n wires.

Data Transmission Medium -8 Wires

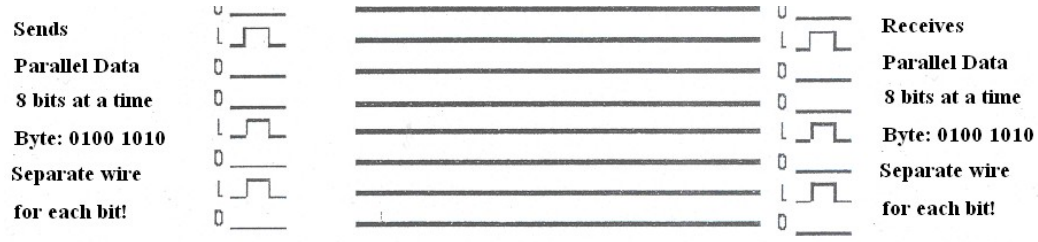


Figure 21: Parallel Communication

Communications	Advantages	Disadvantages
Parallel	Fast Transfer Rates	Short distances only More cost due to more number of lines
Serial	Long Distances	Slow transfer rates Less cost due to only one line required for serial transmission.

3.11 Transmission Media

The transmission media provide the physical path for communication among the nodes. In a computer network, where all the nodes are geometrically interconnected, it is known as its topology.

Transmission media can be broadly categorised in to two types: guided and unguided *(Figure* 22).

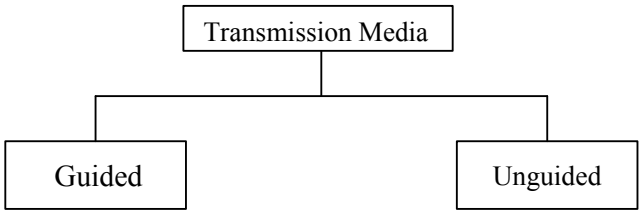


Figure 22: Type of Transmission Media

Guided transmission uses a cabling system that guides the data signals along a specific path. The data signals are bound by the cabling system. Guided media is also known as bound media. "Cabling" is meant in a generic sense, and is not meant to be interpreted as copper wire cabling only. Guided media are commonly used for point-to-point connection. The characteristics of the medium mainly decide the nature and quality of transmission. Guided media are commonly used for LAN application.

Unguided transmission media also called wireless communication consists of a means for the data signals to travel but nothing to guide them along a specific path. The data signals, not bound to a cabling media transport electromagnetic waves without using a physical conductor and are therefore often called unbound media. Unguided media are commonly used for broadcast type communication. Some examples of unguided media are sea water, free space and air. Unguided media are commonly used for WAN application.

Transmission Media Guided

There are 4 basic types of guided media:

a) Open Wire
b) Twisted Pair
c) Coaxial Cable
d) Optical Fiber

Open Wire

Open wire is *(Figure* 23) traditionally used to describe the electrical wire strung along power poles. There is a single wire strung between poles. No shielding or protection from noise interference is used. We are going to extend the traditional definition of open wire to include any data signal path without shielding or protection from noise interference. This can include multi conductor cables or single wires. This medium is susceptible to a large degree of noise and interference and consequently is not acceptable for data transmission except for short distances of less than 20 ft.

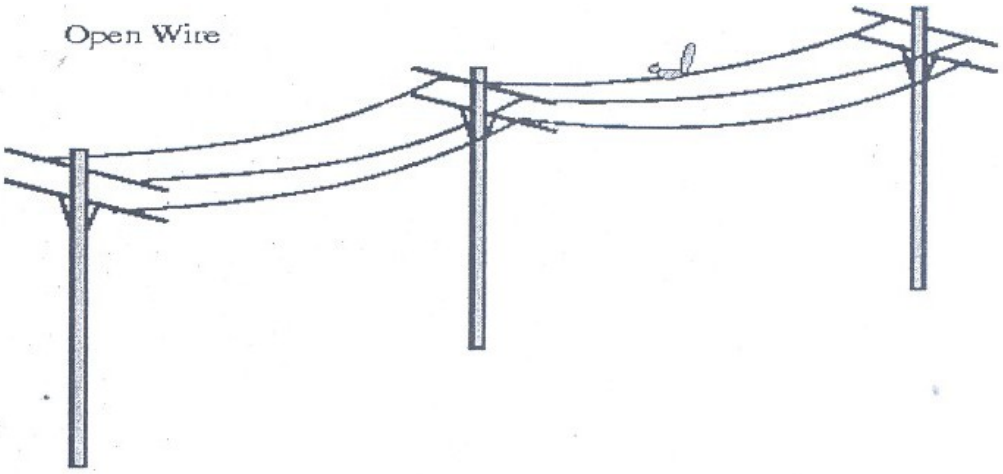


Figure 23: Open Wire

Twisted Pair

The wires in twisted pair *(Figure* 24) cabling are twisted together in pairs. Each pair consists of a wire used for the +ve data signal and a wire used for the -ve data signal. Each pair is twisted together to minimize electromagnetic interference between the pairs. Any noise that appears on I wire of the pair will also occur on the other wire. Because the wires are opposite polarities, they are 180 degrees out of phase (180 degrees - phasor definition of opposite polarity). When the noise appears 011 both wires, it cancels or nulls itself out at the receiving end.

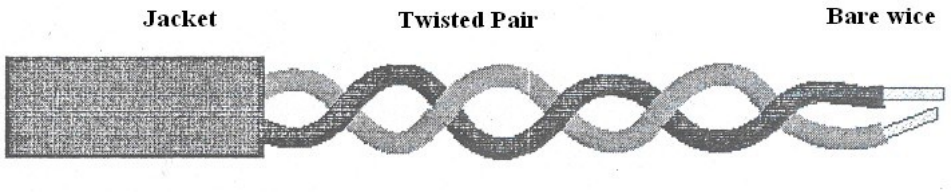


Figure 24: Unshielded Twisted Pair

The degree of reduction in noise interference is determined specifically by the number of turns per foot. Increasing the number of turns per foot reduces the noise interference. To further improve noise rejection, a foil or wire braid "shield" is woven around the twisted pairs. This shield *(Figure* 25) can be woven around individual pairs or around a multi-pair conductor (several pairs).

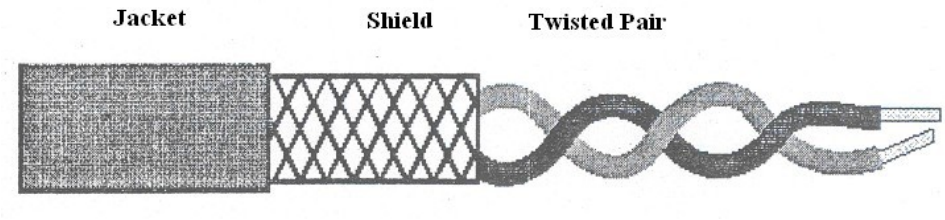


Figure 25: shielded twisted Pair

Shielded Twisted Pair

The twisted pair can be shielded *(Figure* 25) with metallic braid, to reduce the interference Cables with a shield are called shielded twisted pair and are commonly abbreviated STP. Cables without a shield are called unshielded twisted pair or UTP. Twisting the wires together results in characteristic impedance for the cable. Typical impedance for UTP is 100 ohm for Ethernet 10BaseT cable.

UTP or unshielded twisted pair cable is used on Ethernet 10BaseT and can also be used with Token Ring. It uses the RJ line of connectors RJ45, RJ11, etc.)

Use Twisted pair can be used for both analog and digital communication. Twisted pair cables are most effectively used in systems that use a balanced line method of transmission: polar line coding (Manchester Encoding) as opposed to unipolar line coding (TTL logic). Most popular use of twisted pair is in *our* oldest telephone system. It is also used in LAN for point-to point short distance communication

Coaxial Cable

Coaxial cable *(Figure* 26) consists of two conductors. The inner conductor is held inside an insulator with the other conductor woven around it providing a shield. An insulating protective coating called a jacket covers the outer conductor.

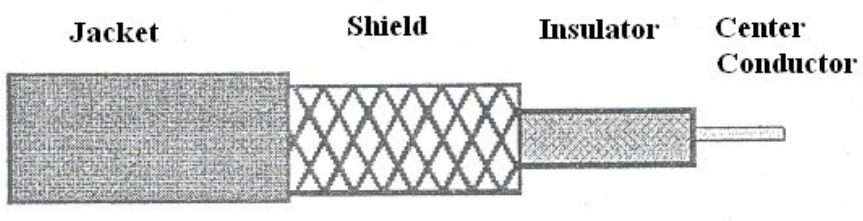


Figure 26: Coaxial Cable

The outer shield protects the inner conductor from outside electrical signals. The distance between the outer conductor (shield) and inner conductor plus the type of material used for insulating the inner conductor determine the cable properties or impedance. Typical impedances **for** coaxial cables are 75 ohms for Cable TV, 50 ohms for Ethernet, Thinnet and Thicknet. The excellent control of the impedance characteristics of the cable allow higher data rates to be transferred than with twisted pair cable.

Optical Fiber

Optical fiber consists of thin glass fibers that can carry information at frequencies in the visible light spectrum and beyond. The typical optical fiber consists of a very narrow strand of glass called the core. Around the core is a concentric layer of glass called the cladding. A typical core diameter is 62.5 microns (I micron = 0" meters). Typically Cladding has a diameter of 25 microns. Coating the cladding is a protective coating consisting of plastic, it is called the Jacket.

Optical fibers work on the principle that the core refracts tile light and the cladding reflects the light. The core refracts the light and guides the light along its path. The cladding reflects any light back into the core and stops light from escaping through it -it bounds the medium.

Advantages of Optical Fiber

Noise immunity: RFI and EM] immune (RFI Radio Frequency Interference, EMI -Electromagnetic Interference) because fiber-optic transmission uses light rather than electricity, noise is not a factor.
Security: cannot tap into cable.
Large Capacity due to BW (bandwidth). .No corrosion.
Longer distances than copper wire.
Smaller and lighter than copper wire.
Faster transmission rate.

Disadvantages of Optical Fiber

Physical vibration will show up as signal noise!
Limited physical arc of cable. Bend it too much and it will break!
Difficult to splice.

The cost of optical fiber is a trade-off between capacity and cost. At higher transmission capacity, it is cheaper than copper. At lower transmission capacity, it is more expensive.

Infrared

Infrared (IR) transmission is another line of sight medium .Infrared technology uses electromagnetic radiation of wave lengths between radio waves and visible light, operation between IOOGHZ and IOOTHZ (terahertz). These frequencies are very high offering nice data transfer rates. We are used to seeing infrared technology utilised for our television or VCR remotes. IR is generally restricted to LAN within or between buildings

Advantages

1) Higher bandwidth means superior throughput to radio
2) Inexpensive to produce
3) No longer limited to tight interroom line-of-sight restrictions

Disadvantage

1) Limited in distance
2) Cannot penetrate physical barriers like walls, ceilings, floors, etc.

3.12 Connecting Devices

As companies grow, so do their networks. When a network outgrows its original design, the network becomes slow and print jobs take longer to be completed. In such cases it is a better idea to segment the existing LAN so that each segment becomes a separated LAN.

We can connect two or more networks together to create larger networks. A LAN (local area network) can be connected to another LAN. A LAN (local area network) can be connected to another WAN (wide area network). The components or devices that are employed to connect two or more networks together are:

1) Repeaters
2) Hubs
3) Bridges
4) Routers
5) Gateways

3.12.1 Repeaters

Repeaters, also called regenerator, are physical hardware devices. They connect two network segments and broadcast packets between them, thus extending your network beyond the maximum length of your cable

segment. They have the primary function to regenerate the electrical signal (shown below):

Reshaping the waveform
Amplifying the waveform
Retiming the signal, to avoid collision on the network

As signal travels along a cable, its strength or amplitude decreases. This is called attenuation. In other words, the signal attenuates as it travels along a cable. This limits the length of a cable used to connect the computers together.

Since signal is a factor in the maximum length of a segment, repeater can regenerate (or amplify) the weak signals so that they can travel additional cable lengths. A repeater has intelligence, so that it takes a weak signal from one cable segment, regenerates it and passes it on to the next segment. Simply we can say that it recreates the bit pattern of the original signal. No more than four repeaters are used to join segments together to keep collision detection working properly. We should not confuse repeater with amplifiers. As the amplifier uses analog signal, it cannot differentiate between original signal and noise, therefore it amplifies both original signal and noise. The repeater does not amplify the original signal, it regenerates the original bit pattern.

Purpose of a Repeater

The purpose of a repeater (Figure 27) is to extend the LAN Segment beyond its physical limits (as defined by the Physical Layer's Standards: e.g. Ethernet is 500m for I OBase5). ALAN Segment is a logical path, such as the logical bus used by all 802.3 Ethernet types. A LAN Segment is given an identification number, called a Segment Number or Network Number, to differentiate it from other segments.

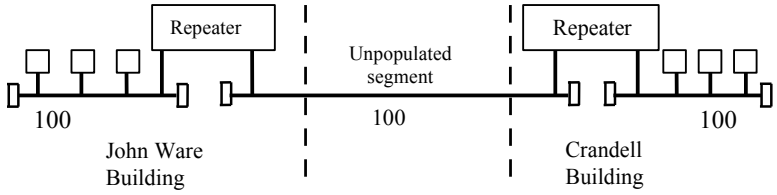


Figure 27: Repeater

Typically, repeaters are used to connect two physically close buildings together (when they are too far apart to just extend the segment). They can be used to connect floors of a building that would normally surpass the maximum allowable segment length. Note: for large extensions, as

in the above example, two Repeaters are required. For shorter extensions, only one Repeater may be required.

Repeater's OSI Operating Layer

Repeaters operate at the OSI Model Physical Layer (Figure 28).

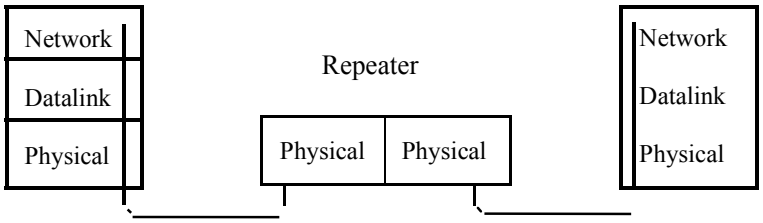


Figure 28: Repeater's Segment-to-Segment Characteristics

A repeater cannot join two cable segments using different access methods. A repeater is not used to connect a segment using *CSMA/CD* access method to a segment using token passing access. Repeaters can join two different physical media, but they must use the same access method. Thus a repeater can have physical connections to join a coaxial cables segment to a fiber optic segment.

Repeaters do not "de-segment" a network. All traffic that appears on one side of the repeater appears on both sides. Repeaters handle only the electrical and physical characteristics of the signal.

Repeaters work only on the same type of Physical Layer: Ethernet-to-Ethernet or Token Ring-to- Token Ring. They can connect 10BaseS to 10BaseT because they both use the same 802.3 MAC layer.

You can run into problems with the transfer rate (I Mbps vs. 10 Mbps) when you connect 1 Base5 to 10BaseT. A repeater cannot connect Token Ring to Ethernet because the Physical Layer is different for each network topology.

Repeater Addressing: MAC Layer and Network Segment

The MAC Layer Address is used to identify the Network Card to the Network. The Repeater is transparent to both sides of the segment and both sides can "see" all the Mac Addresses (regardless of which side they are on). This means that any network traffic on Floor I will also appear on Floor 5, and vice versa *(Figure* 29).

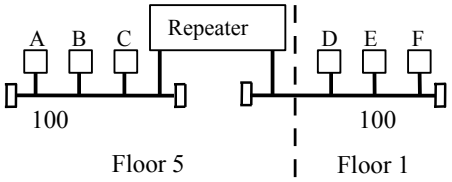


Figure 29: Repeater Addressing

Nodes A & B could be furiously are exchanging files; this network traffic would also appear on Floor I. Repeaters don't provide isolation between segments (there is only one collision domain).

Because Repeaters provide no isolation between segments, and the repeater is transparent to both sides of the segment, both sides of the repeater appear as one long segment. The Network Number, or Segment Number, is the same on both sides of the Repeater.

3.12.2 Hubs

Hubs can also be called either Multi port Repeaters or Concentrators. They expand one Ethernet connection into many. They are physical hardware devices. A hub is similar to a repeater, except that it broadcasts data received by any port to all other ports on the hub.

Some hubs are basic hubs with minimum intelligence (i.e. no microprocessors), Intelligent Hubs can perform basic diagnostics, and test the nodes to see if they are operating correctly. If they are not, the Smart Hubs (or Intelligent Hubs) will remove the node from the network. Some Smart Hubs can be polled and managed remotely.

Purpose of Hubs

Hubs are used to provide a Physical Star Topology *(Figure 30).* The Logical Topology is dependent on the Medium Access Control Protocol. At the center of the star is the Hub, with the network nodes located on the tips of the star.

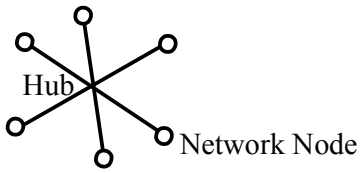


Figure 30: Position of Hub

Star Topology

The Hub is installed in a central wiring closet (Figure 31) with all the cables extending out to the network nodes. The advantage of having a central wiring location is that it is easier to maintain and troubleshoot large networks. All of the network cables come to the central hub. This way, it is especially easy to detect and fix cable problems. You can easily move a workstation in-a star topology- by changing the connection to the hub at the central wiring closet.

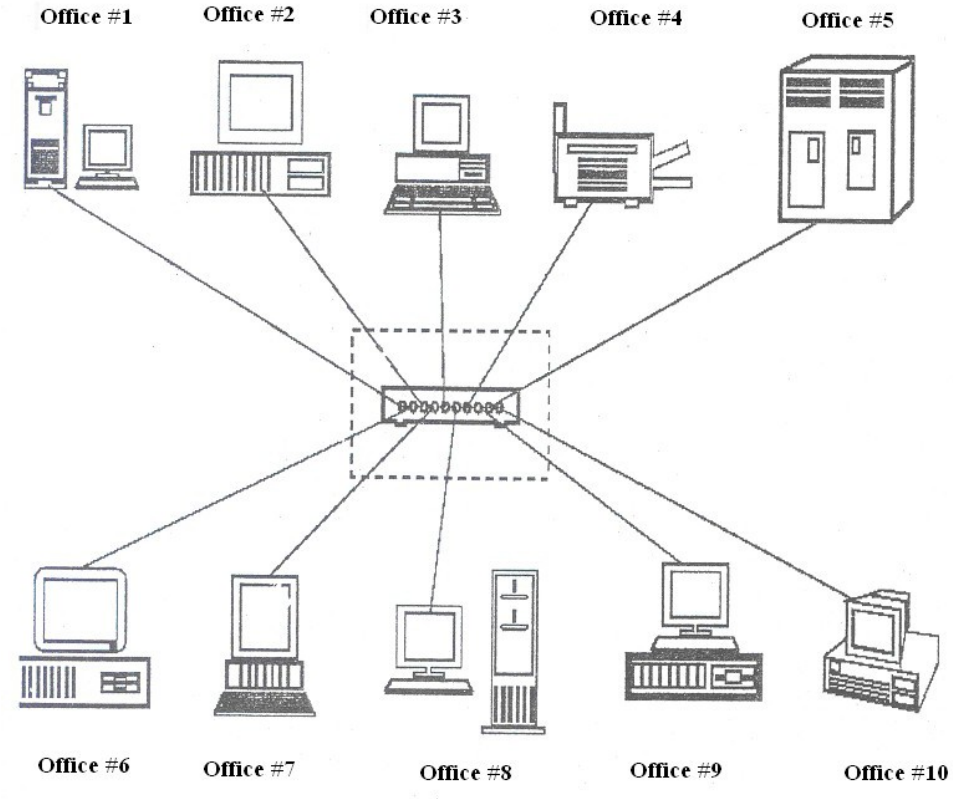


Figure 31: Example of Hub

Hub's OSI Operating Layer

Hubs are multi port repeaters, and as such they obey the same rules as repeaters (See previous section OSI Operating Layer). They operate at the OSI Model Physical Layer.

Hub's Segment-to-Segment Characteristics

To understand the Ethernet segment-to-segment *(Figure* 32) characteristics of a hub, determine how the Ethernet Hubs operate. Logically, they appear as a Bus Topology, and physicaJ1y as a Star Topology. Looking inside an Ethernet Hub, we can see that it consists of

an electronic printed circuit board (which doesn't teJ1 us much). If we form a functional drawing, then we can clearly see how the Physical and Star Topology appears:

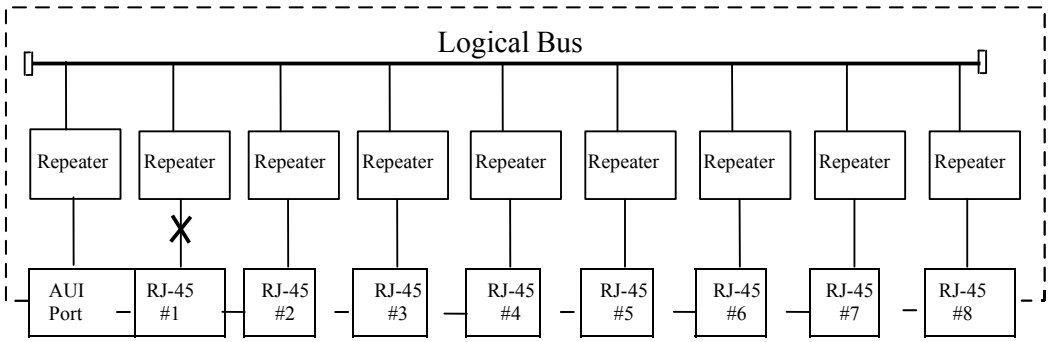


Figure 32: Hub's Segment to Segment

Understanding that inside the Hub is only more repeaters, we can draw the conclusion that aJ1 connections attached to a Hub are on the same Segment (and have the same Segment Number). A single repeater is said to exist from any port to any port, even though it is indicated as a path of 2 repeaters.

Hub's Addressing

Again, because a Hub is just many repeaters in the 'same box, any network traffic between nodes is heard over the complete network. As far as the stations are concerned, they are connected on one long logical bus (wire).

Switching Hubs

Switching hubs *(Figure* 33) are hubs that wiJ1 directly switch ports to each other. They are similar to full duplex hubs, except that they allow dedicated 10 Mbps channels between ports.

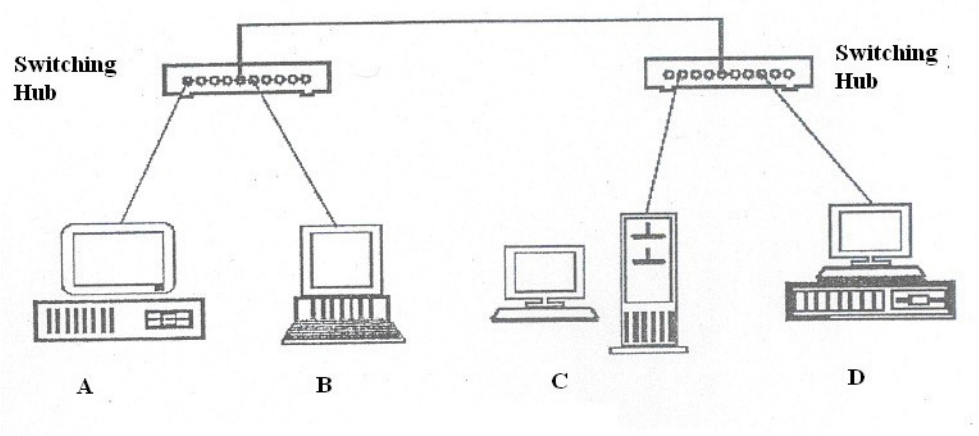


Figure 33: Switching Hub

If A wanted to communicate with B, a dedicated] 0 Mbps connection would be established between the two. If C wanted to communicate with D, another dedicated] 0 Mbps connection would be established.

3.12.3 Bridges

Bridges have all the features of the repeater. Besides regenerating the signals, a bridge can segment (or divide) a network to isolate traffic related problems. A bridge sends the data frames only to the concerned segment, thus preventing excess traffic. A bridge can split an overloaded network into two separate networks, reducing the amount of traffic on each segment and thus making each network more efficient. Just like repeaters, the bridges can be used to link different physical media. Bridges can also be used to connect dissimilar networks like Ethernet system to a Token Ring system. Thus bridges can be used to join networks using CSMA/CD access and token passing access.

Bridges are both hardware and software devices. They can be standalone devices - separate boxes specifically designed for bridging applications- or they can be dedicated PCs (with 2 NICs and bridging software). Most server software will automatically act as a bridge when a second NIC card is installed.

Bridge QSI Operating Layer

Bridges *(Figure* 34) operate on the OSI Model Data Link Layer, while repeaters work at the physical layer. Since bridges work on a higher layer than repeaters, they are more complex than repeaters and cost more than repeaters. They look at the MAC addresses for Ethernet and Token Ring, and determine whether or not to forward--or ignore--a packet.

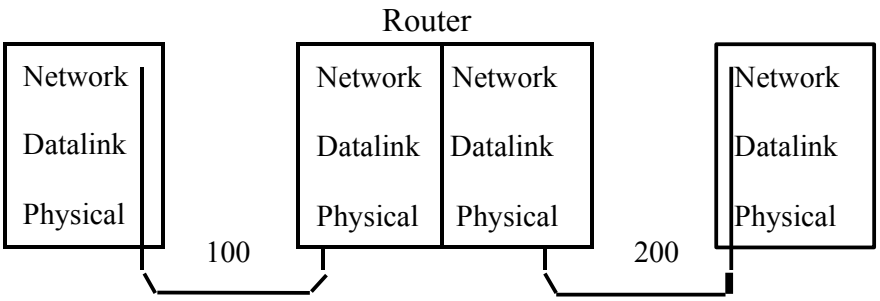


Figure 34: Bridge OSI Operating Layer

Bridges have their own routing tables. Initially the bridge's routing table is empty. As nodes send packets, the source address is copied to the routing table. With this address information, the bridge learns where the computers are situated. When any packet is received by a bridge it reads its source and destination address. If the bridge knows the location of the destination node it forwards the packet to the segment on which the destination node is situated. If it does not know the destination, it forwards the packet to all the segments.

Purposes of a Bridge

The purposes of a Bridge are the following:

Isolates networks by MAC addresses.
Manages network traffic by filtering packets
Translates from one protocol to another

Isolates networks by MAC addresses

For example, you have one segment called Segment 100: it has 50 users (in several departments) using this network segment. The Engineering Dept. is CAD (Computer Aided Design)-oriented, while the Accounting Dept. is into heavy number crunching (year end reports, month end statements, etc.).

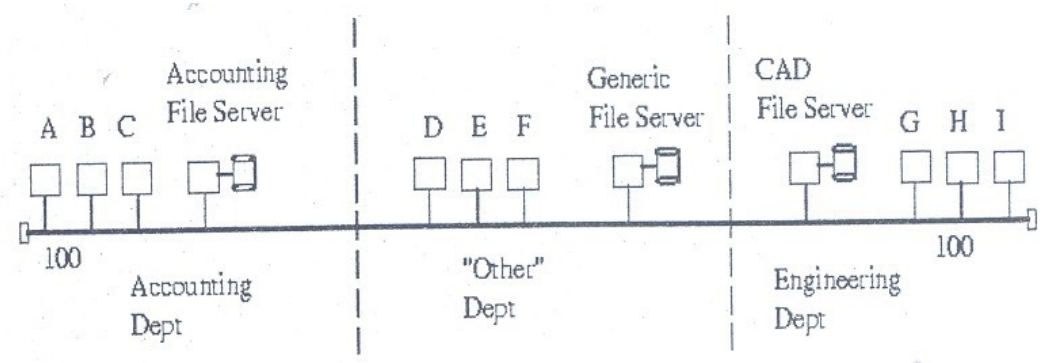


Figure 35: Bridge Example 1

On this network, any traffic between Clients A, B or C and the Accounting File Server (in the Accounting Dept.) will be heard across the Segment 100. Likewise, any traffic between the Engineering Dept. Clients G, H or I (to the CAD File Server) will be heard throughout the Network Segment. The result is that "Other" Department accesses to the Generic File Server are incredibly slow: this is because of the unnecessary traffic that is being generated from other departments (Engineering & .Accounting).

The solution is to use one Bridge *(Figure* 36) *to* isolate the Accounting Dept., and another bridge to isolate the Engineering Department. The Bridges will only allow packets to pass through that are not on the local segment. The bridge will first check its "routing" table to see if the packet is on the local segment. If it is, it will ignore the packet, and not forward it to the remote segment. If Client A sent a packet to the Accounting File Server then Bridge #1 will check its routing table (to see if the Accounting File Server is on the local port). lf it is on the local port, then Bridge #1 will not forward the packet to the other segments.

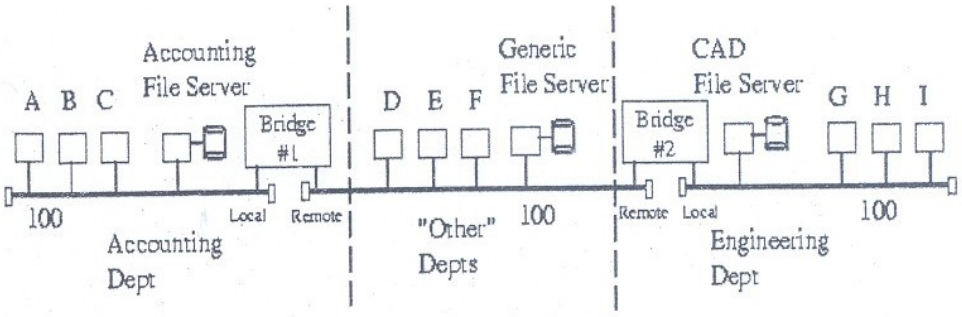


Figure 36: Bridge Example 2

If Client A sent a packet to the Generic File Server, Bridge #1 will again check its routing table to see if the Generic File Server is on the local port. If it is not, then Bridge # I will forward the packet to the remote port.

Note: The terms local and remote ports are arbitrarily chosen to distinguish between the two network ports available on a bridge.

In this manner, the network is segmented, and the local department traffic is isolated from the rest of the network. Overall network bandwidth increases because the Accounting Dept. does not have to fight with the Engineering Dept. (for access to the segment). Each segment has reduced the amount of traffic on it and the result is faster access. Each department still has complete access to the other segments, but only when required.

3.12.4 Routers

A router is a special-purpose computer having a processor (CPU) and memory like any other computer. But unlike any other computer, it has more than one I/O interface that allows it to connect to multiple computer networks.

Routers are both hardware and software devices. Just like bridges, Router can connect network segments and filter and isolate traffic. Unlike a bridge, a router can connect networks that use different technologies, addressing methods, media types, frame formats, and speeds. Routers are used in complex network situations because they provide better traffic management than bridges. A router keeps track of the address of all the segment of a network and can even determine the best path for sending data. Routers do not pass broadcast traffic.

Like bridges, the routers also maintain routing tables in their memories to store information about physical connections on the network. The router examines each packet of data, checks the routing table, and then forwards the packet if necessary. Routers are more inelegant than bridges, as routers can share status and routing information with one another and use this information to bypass slow or malfunctioning connections. Routers do not maintain any state information about the packets; they simply move them along the network. Routers are usually employed by wide area networks using dissimilar addressing schemes and different communication protocols.

Routers do not allow bad data to get passed on to the network. Thus they save networks from broadcast storms.

There are two types of routers -static routers and dynamic routers.

Static routers require an administrator to manually set up and configure the routing table and to specify each route.

Dynamic routers maintain a routing table automatically and require minimal set up and configuration.

Router OSI Operating Layer

Routers operate on the OSI Model's Network Layer as shown in *Figure* 37. The Internet work must use the same Network Layer protocol. Routers allow the transportation of the Network Layer PDU through the Internetwork, even though the Physical and Data Link Frame size and addressing scheme may change.

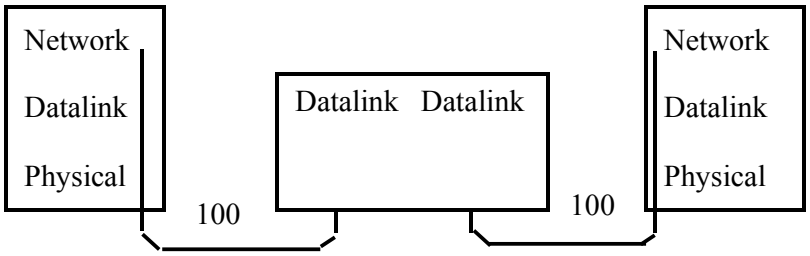


Figure 37: router Segment-to-Segment Characteristics

Routers that only know Novel. IPX (Internet work Packet Exchange) will not forward UNIX's IP (Internet work Packet) PDUs, and vice versa. Routers only see the Network Layer protocol that they have been configured for. This means that a network can have multiple protocols running on it (e.g. spxnpx, TCP/IP, Appletalk, XNS, etc.).

Router Addressing

Routers know the address of all known networks. They maintain a table of pathways between networks and can select an optimal route over which to send data. Routers look only at network address and not at destination node address. Routers talk to other routers, but not to remote computers.

Routers combine the Network Number and the Node Address to make Source and Destination addresses {in routing Network Layer PDUs acrpss a network). Routers have to know the name of the segment that they are on, and the segment name or number where the PDU is going. They also have to know the Node Address: MAC Address for Nove.!l, and the IP address for TCP/IP.

3.12.5 Gateways

A Gateway is the Hardware/Software device that is used to interconnect LANs & WANs.

Gateways are much more complex and powerful than a router. They are slower than a router and are expensive. A Gateway incorporates the functions of routers and bridges, but it can translate instruction set on sending network into corresponding instruction set of the receiving network. Gateways make communication possible between different architectures and environments.

Often, the router that is used to connect a LAN to the Internet will be called a gateway. It will have added capability to direct and filter higher

layer protocols (layer 4 and up) to specific devices (such as Web servers, ftp servers and e-mail servers).

A Gateway links two systems that do not use the same communication protocols, data formatting structures, languages and architecture, which can not be done by a router. Gateways perform protocol and data conversion.

Gateway's OSI Operating Layer

A Gateway operates at the Transport Layer and above and it typically translates each source layer protocol into the appropriate destination layer protocol. Gateways use all the seven layers of the OSI model A mainframe gateway may translate all OSI Model layers. For example, IBM's SNA (System Network Architecture) does not readily conform to the OSI Model, and requires a gateway to translate between the two architectures.

4.0 CONCLUSION

In this unit, you have been introduced to concepts such as network topologies, OSI layers, types of network, transmission media, and various types repeaters, bridges, switches, routers, gateways, etc.)

5.0 SUMMARY

A network allows one to share access to information devices.
Communication protocol is a set of conventions or rules that must be adhered to by both communicating parties to ensure that information being exchanged between the two parties is received and interpreted correctly.
The major criteria to judge a data communication network are: performance Consistency, Reliability, Recovery, Security.
The topology is the geometric arrangement (either physically or logically) of the linking devices (usually called nodes) and the links, connecting the individual computers or nodes together. Different topologies are mesh, Star, ring or combined topology.
Communication between two devices can occur in three transmission modes: simplex, half-duplex or full duplex.
Computer network is classified into three types: LAN, MAN and WAN.
The network of networks is called the Internet.
The most familiar type of DCE is the modem that modulates and demodulates signals.

Guided transmission media use a cabling system that guides the data signals along a specific path.
Unguided transmission media consist of a means for the data signals to travel but nothing to guide them along a specific path.
Repeater is a device that operates at the physical layer, bridge at the data link layer, router at the network layer and Gateway at all seven layers of the OSI model.

The cost of optical fiber is a trade-off between capacity and cost. At higher transmission capacity, it is cheaper than copper. At lower transmission capacity, it is more expensive.

The Bus Topology

The main advantage of bus topology is that it is quite easy to set up. Any Workstation can be easily moved to another location as bus runs throughout the office. Another benefit of this layout is that if one computer on the bus fails, it does not affect the rest of the traffic on the bus.

A network with bus topology cannot become too big as all the traffic is on a single bus. The entire network can be down only if the bus has a break. The open ends of the bus must be terminated to prevent signal bounce. If one or both ends of the bus are not terminated, the whole network can be down.

Disadvantages include difficult reconfiguration and fault isolation

Mesh Topologies

In a mesh topology, every node has a dedicated point-to point link to every other node. Simply dedicated means that the links carry traffic only between the two nodes. So mesh topology does not have traffic congestion problems. Every node has n-l link, for a fully connected mesh topology having n nodes. So total number of links will be n (n-1). This also means that every node has (n-l) I/O ports

Advantages of Mesh topology

1) Use of dedicated links guarantees that each connection can carry-its own data load. This eliminates the traffic problem.
2) If one link fails, it does not affect the rest of network. This means it is robust.
3) Point to point links makes fault identification and fault isolation easy:
4) Privacy or security is high; as any other link cannot gain access to dedicated link where the message is travailing.

Disadvantages of Mesh Topology

1) More cabling and I/O ports are required, because every node must be connected to every other node.
2) Cost is very high, because more number of nodes and cabling required.
3) Installation and reconfiguration is difficult.
4) The complexity of writing communication software can be reduced by adopting the principle or protocol1ayering. The idea here is to partition communication functions into a vertical set of layers. Each layer performs a related set *of* functions. Division of work between layers is done in such a way that they are manageable and provide a logical interface and break point. Each communication layer provides certain services to layers above it and relies on. the next lower layer to perform more primitive functions. Each layer hides internal details from other layers. Thus dividing the communication problem into several layers reduces its complexity and makes the work of developing communication software a lot easier and error free.

6.0 TUTOR-MARKED ASSESSMENT

1) Compare the advantage of fiber over copper wire.
2) Discuss the advantages and disadvantages of Bus & Mesh Topologies.
3) What are the roles of protocols in a computer network?

7.0 REFERENCES/FURTHER READINGS

"Computer Networks, Tanenbaum", Third Edition, Prentice-Hall 1996.

"Data and Computer Communications", William Stallings, Fourth Edition, Macmillan, 1994.

"Data Communication and Networkings". Behrouz A.Forouzan 2nd edition, TMH 2000.

"Internetworking with TCP/IP", Douglas Comer; Volume I, Fourth Edition, Prentice Hall, 2000.

UNIT 4 INTERNETWORKING: CONCEPT, ARCBJTECTURE AND PROTOCOLS

CONTENTS

1.0 INTRODUCTION

An internetwork is a collection of packet-switching and broadcast networks, connected by bridges, switches, or routers which are intermediate networking devices, that functions as a single large network. So all users and devices can communicate, regardless of the network segment to which they are attached. *Figure* 1 illustrates some different kinds of network technologies that can be interconnected by routers and other networking devices to create an internetwork.

In this unit we will explain the concepts, architecture and protocols of internetworking.

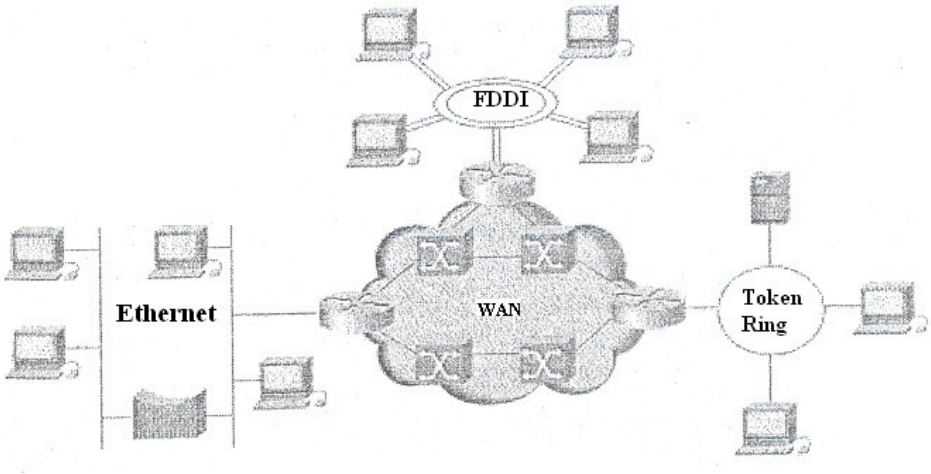


Figure 1: Internetworking

2.0 OBJECTIVES

After going through this unit you should be able to:

define packet switching concept
differentiate between virtual circuit & datagram understand
the functioning of a large number of protocols.

3.0 MAIN CONTENT

3.1 History of Internetworking

Networking allows computers to share information, applications and even hardware devices. For any two computers to communicate with each other, they must follow a set of rules called protocol. In early years of networking, every vendor was concerned only with his own product. Each vendor defined its own standard for communication between its computers. This resulted in many different types of standards for network hardware and software that did not share a common protocol. This meant that different systems could not interact with each other. Two computers using network products of different vendors could not be connected together.

In the late 1970s, the networking community came together in an effort to replace these closed systems with open systems. They wanted that all networking products should be compatible with other vendor products. The International Standards Organisation (ISO) developed Open Systems Interconnection (OSI) reference model for networking the model gives guidelines about how the parts of a network communication system should work together.

The Open System Interconnection (OSI) reference model describes how Information from a software application in one computer moves through a network medium to a software application in another computer. The OSI reference model is a conceptual model composed of seven layers, each specifying particular network functions. The model was developed by the International Standards Organisation (ISO) in 1984, and it is now considered the primary architectural model for inter computer communications. The OSI model divides the tasks involved with moving information between networked computers into seven smaller, more manageable task groups. A task or group of tasks is then assigned to each of the seven OSI layers. Each layer is reasonably self-contained so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers. The following list details the seven layers of the Open System Interconnection (OSI) reference model:

Layer 7-Application
Layer 6-Presentation
Layer 5-Session
Layer 4- Transport
Layer 3-Network
Layer 2-Data link
Layer 1-Physical

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data link
1	Physical

Figure 2: The OSI Reference Mode! Contains Seven Independent Layers

3.2 Packet Switching

Data communication takes place between two devices that are directly connected through some form of transmission medium. It is impractical for two devices to be connected directly. Instead, a network of switching nodes provides a transfer path between two devices. Packet switching involves the breaking up of messages into smaller components called packets. Packets often range in size from about 128 bytes to over 4096 bytes depending on the system involved. Each packet contains source

and destination information, and is treated as an individual message. These mini-messages are received and routed through optimal routes by various nodes on a wide area network. For example a file to be transmitted between two machines maybe broken into many packets that are sent across the network one at a time. The network H/W delivers the packet to the end where the network software reassembles them into a single file again. There are two major types of packets to be switched. They are datagram and virtual circuit.

In the **datagram approach**, each packet is treated independently and may follow a different path through the network. Packets may be re-ordered, dropped or delivered in wrong sequence. The communication protocols will have to provide error recovery and sequencing of packets at the destination.

In-the **virtual circuit** approach, a fixed logical path through the network from "sender to destination is established before any packets are sent. This path remains unchanged for the duration of the connection or session. Although no resources are reserved along the path, packets are buffered at intermediate nodes awaiting transmission.

Thus, a virtual circuit only defines a path for packets to follow without actually reserving dedicated channels along the route as is the case with circuit switching. Virtual circuit may provide a number of services including sequencing, error control it and flow control.

In comparing datagram and virtual circuit switching with other switching technologies, there are several factors to be considered. First of all, packet switching is faster because messages are not stored in their entirety for later retrieval. Each packet is small enough to be stored in a router's machine memory until it can be routed an instant later. Secondly, packet switching allows the avoidance of route failure due to excessive traffic loads. This is accomplished by routing packets along routes that are the most free and clear. Thirdly, packet switching spreads the load of communication across several paths.

The motivations for adopting packet switching are cost and performance. Because multiple machines can share network fewer interconnections are required and cost is kept low. Packet switched networks that span large geographical distances are fundamentally different from those that span short distances. To help characterize the differences in capacity and intended use, packet switched technologies are often divided into three broad categories: Wide Area Network (WAN), Metropolitan Area Network (MAN) and Local Area Network. There are structural protocols for each category. In the next section we will examine one such protocol.

3.3 Internetworking Concepts

We have seen how machines connect to individual networks. The question arises, "How are networks interconnected to form an inter network?" The answer has two parts. Physically, two networks can only be connected by a computer that attaches to both of them. A physical attachment does not provide the interconnection we have in mind, however, because such a connection does not guarantee that the computer will cooperate with other machines that wish to communicate. To have a viable internet, we need computers that are willing to shuffle packets from one network to another. Computers that interconnect two networks and pass packets from one to the other are called internet gateways+ or internet routers.

Consider an example consisting of two physical networks shown in *Figure* 3. In the figure, machine G connects to both network I and network 2. For G to act as a gateway, it must capture packets on network 1 that are bound for machines on network 2, and packets on network 2 that are destined for machines on network 1 and transfer them.

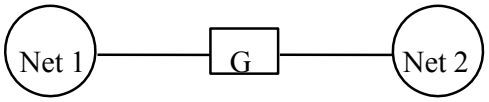


Figure 3: Two networks interconnected by G, a gateway (router)

Interconnection through IP Gateways or Routers

When internet connections become more complex, gateways need to know about the topology of the internet beyond the networks to which they connect. For example, Figure 4 shows three networks interconnected by two gateways.

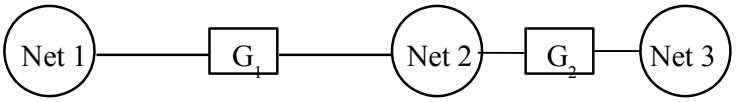


Figure 4: Three networks interconnected by two gateways.

In this example, gateway G must move from network I to network 2 all packets destined for machines on either network 2 or network 3. As the size of the internet expands, the gateway's task of making decisions about where to send packets becomes more complex.

The idea of a gateway seems simple, but it is important because it provides a way to interconnect networks, not just machines. In fact, we have already discovered the principle of interconnection used throughout an internet:

In a TCPIIP internet, computers called gateways provide all interconnections among physical networks.

You might suspect that gateways, which must know how to route packets to their destination, are large machines with enough primary or secondary memory to hold information about every machine in the internet to which they attach. However, gateways used with TCP/IP internets are usually minicomputers; they often have little or no disk storage and limited main memories. The trick to building a small internet gateway lies in the following concept:

Gateways route packets based on destination network, not on destination host.

Lf routing is based on networks, the amount of information that a gateway needs to keep is proportional to the number of networks in the internet, not the number of machines.

3.4 Internet Addresses

In the previous section we defined a TCP/IP internet as a virtual network built by interconnecting several physical networks through gateways which are usually controlled by the internet service providers think of internet as a large network like any other network. The difference of course is that the internet is a virtual structure imagined by its designers and implemented entirely in TCP/IP s/w. Each machine (host) on an internet is assigned a unique 32 bit internet address that is used in all communication with that machine. Conceptually each address is a paid (netid, hosted) where **netid** identifies a network and hosted identifies a host on that network. Because IP addresses comprise both a network as a host on that network they do not specify an individual machine but a connection to a network. For example a getway connecting networks has distinct IP address, one for each network connection.

Unicasting, Broadcasting, and Multicasting

When an IP datagram is sent to an individual IP address, it is called a unicast IP datagram. The process of sending the datagram is called unicasting. Unicasting is used when two IP nodes are communicating with each other.

When an IP datagram is sent to all nodes on a specific network, it is called broadcasting.

There is a third mode of sending an IP datagram called multicasting. In multicasting the IP datagram is delivered to a group of systems

identified by a class D address. The systems that have the same multicast address are said to belong to a multicast group. Members of the multicast group, while being assigned a class D address, must also be assigned an IP address (from the class A, B, or C address group). The multicast group can receive an IP datagram in two ways:

IP datagrams sent directly to their individual IP address (class A, B, C).
IP datagrams sent to their multicast address (class D).

3.5 Configuring IP Addresses

Hosts on a TCP/IP network need to be configured using proper IP addresses. The actual configuration procedure depends on the operating system. The configuration procedure can be classified into the following categories.

Command line based
Menu interface
Graphical User Interface based
Obtained dynamically when host boots from a central server

Table 1: Configuring IP Address and other Parameters for Hosts

Method	**Operating System! Protocol**
Command line	Unix, VMS, Router, devices, and MS-DOS
Menu interface	Router devices, Unix, NetWare servers, and MS-DOS
Graphical User Interface	Microsoft Windows products.
Dynamically assigned	DHCP and BOOTP protocols. Available on almost all major operating system platforms.

Many systems offer a command that can be executed to modify the IP address. In these systems, the command line is often placed in the startup script for the operating system. The menu interface is built using extended line drawing character sets. You are prompted to enter the IP address and other IP parameters. The Graphical User Interface is for systems -such as X-windows and Microsoft's Windows operating systems-that offer a pixel-based graphical view. The dynamically assigned IP address is used in conjunction with BOOTP or DHP protocol. When starting up, a device requests its IP address and other

parameters from a central server that can deliver this information using either the BOOTP or the DHCP protocol.

The IP address information for the host is recorded in a number of places. When the IP address information is entered, it is cached in memory and is available for use by the TCP/IP software. Alternatively, this information can be recorded on a system file in the operating system. IP addresses of oilier-systems can be discovered by consulting a special file, usually called the "hosts" file, or by using the DNS protocol. The DNS service is typically used to determine a host's IP address given its symbolic DNS name. In some systems, proprietary protocols can be used to discover an IP address on a network. An example of this is the WINS service used in Microsoft's operating systems.

You must consult your operating system manuals for actual details on configuring IP addresses. The following sidebar provides examples of configuring IP addresses for most Unix implementations, and for the Windows NT operating System.

Imagine that you are configuration network interface to IP address 144.19.74.102, subnet mask 255.255.0.0, and directed broadcast address 144.19.255.255.

For a Unix Configuration

1) Log on as root user
2) Run the following command:

If configeth 0144.19.74.102 netmask255.255.0.0. Broadcast 144.19.255.255. Replace eth0 with the Unix logical name of the network interface. For Windows NT

1) Log on as Administrator user.
2) Select the following:

   Start
   Settings
   Control Panel .Network
   Select the *TCP/IP* protocol
   Select Properties.

3) You will see a dialog box in which you can enter the IP address and subnet mask.

Windows NT, by default, uses the appropriate directed broadcast address of I 44. I 9.255.255 based on the subnet mask that you specify.

3.6 TCP/IP

Perhaps no other protocols designed to work above the Data Link and Physical OSI layers are as popular as TCP/IP. That's primarily because this global protocol suite has been used by and continually promulgated by thousands of government and educational institutions world-wide.

The Transmission Control Pr9tocol (TCP) and the Internet Protocol (IP) are commonly referred to collectively as TCP/IP. TCP represents a transport layer protocol that provides end to end reliable transmission. To do so, TCP includes such functions as flow control, error control, and exchange of status information. In comparison, IP represents a connection less-mode network layer protocol designed to route message between networks. In addition to TCP, the Internet suite specifies an optional connection less -mode layer 4 transport protocol known as the User Datagram Protocol (UDP).UDP is used for transaction -based applications, such as transmission of network management information when transmission efficiency is more important than reliability.

Figure 5 illustrates the layering structure of the TCP/IP protocol suite to include a few of the application services included in the suite.

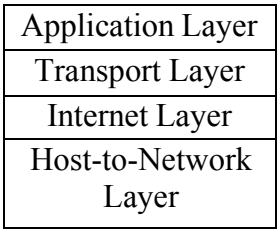


Figure 5: TCP/IP reference model

The advantage of TCP/IP for a network operating system is simple: Interconnectivity is possible for any type of operating system and hardware platform that you might want to add.

TCP/IP is not a single protocol but a set of more than a dozen protocols. Each protocol within the TCP/IP family is dedicated to a different task. All the protocols that make up TCP/IP use the primary components of TCPIIP to send packets of data.

Transmission Control Protocol and Internet Protocol are two of the primary protocols in the TCPIIP family. The different protocols and services that make up the TCPIIP family can be grouped according to their purposes. The groups and their protocols are the following:

Transport: These protocols control the movement of data between two machines.

TCP (Transmission Control Protocol): A connection-based service, meaning that the sending and receiving machines communicate with each other through a stream of messages. TCP has message delivery assurance routines incorporated into it.

3.7 Additional TCP/IP-Related Protocols

There are several additional protocols designed to assist TCP and IP. Since routing is so important on a packet-switched network like the Internet, specialized protocols have been designed to assist in this function. Special protocols for determining addressing on the Internet have also been devised. Additionally, some additional protocols may be involved in error-checking and flow control, just to name a few. Let's explore some of these additional protocols that are included in the TCP/IP suite of protocols.

File Transfer Protocol (**FTP)** allows the transfer of copies of files between one node and another. FTP is not hardware-dependent so its services can function just about anywhere. Using this utility to copy data is typically referred to as "FTPing" a file.

Network Filing System (**NFS)** was developed by Sun Microsystems Inc. It provides shared access to files in a very transparent and integrated way. This protocol is discussed in more detail a little later.

Remote Terminal Emulation (**TELNET)** allows users to communicate with diverse hosts. The TELNET application provides terminal-type access to PCs.

User Datagram Protocol (UDP) is a bare-bones rapid transmission protocol -that uses IP packets to deliver data with no reliability features like connections and ACKs. The forte of UDP is speed, not reliability. It is used in NFS.

Simple Mail Transfer Protocol (SMTP) is the middleman that uses UDP to move data around from one internet work host to another. Applications run on both hosts that make use of SMTP.

Internet Control Message Protocol (ICMP) offers flow control and error- detection to the unreliable delivery method of IP. It provides a facility for routers and gateways on the net to

communicate with a source if there is a problem. It also provides a mechanism for determining if a destination cannot be reached.

Routing Information Protocol (RIP) provides information for routing devices about pathways and number of hops to achieve them. RIP was popularized by its use in a Berkeley UNIX application called "Routed". RIP is ideal for smaller networks, but considered impractical for larger internetworks.

Address Resolution Protocol & Reverse Address Resolution Protocol (ARP & RARP) are special protocols to allow TCP/IP to interact in environments such as Ethernet. ARP maps TCP/IP addresses to Ethernet Data Link layer addresses. RARP maps the Ethernet Data Link layer address to the TCP/IP address.

That's an overview of some of the better-known additional protocols.

3.8 Application Layer Protocols

The first generation of Internet applications from the late 1960s and early 1970s addressed the need to transfer files between computers, to have remote access to computing resources, and to support communication between users at different hosts. The applications, which met these needs: FTP, Telnet, HTTP, SNMP and DNS, were the predominant applications in the first decades of the Internet. The emergence of the World Wide Web in the early 1990squickly made web browsing the most popular Internet application. Recently, applications for streaming audio and video, telephony over the Internet, and peer-to-peer file sharing have again changed the landscape of Internet applications.

This section discusses knowledge about how applications interact with the Internet and how applications take advantage of the Internet to provide network services to end- users.

3.8.1 File Transfer Protocol

The File Transfer Protocol (FTP) for copying files between computer systems is one of the oldest application-layer protocols and was created before the *TCP/IP* protocol suite. FTP is a client-server protocol where an FTP client accesses an FTP server. The FTP client authenticates itself with a user name and a password to establish an FTP session. A successful authentication results in the establishment of an FTP session, where the FTP client can download ("get") and upload ("put") files and file lists. When transferring files, FTP respects the ownership and access privileges of files.

Most hosts have a utility program, generally also called FTP, which provides an interactive command line interface to run an FTP session. .FTP clients can also be integrated in other applications, such as web browsers. Anonymous FTP is a special form of file transfer service that allows public access to files at an FTP server. An FTP client can establish an anonymous FTP session by supplying the user name "anonymous" and an arbitrary password (The FTP server usually requests to supply an email address as password).

FTP employs TCP as its transport protocol, thereby ensuring a reliable transfer of transmitted data. Two TCP connections are established for each FTP session, called control connection and data connection. The control connection is used for commands from the client and messages from the server. The data connection is used for the transport of files. FTP server uses the well-known TCP port 2] for the control connection, and the well-known TCP port 20 for the data connection, and an FTP client selects available ephemeral port numbers. The control connection is established at the beginning of the FTP session and stays up for the entire lifetime of the session. The control connection is used by the FTP client for authentication, for setting various session parameters, and for commands to download or upload files. The data connection is opened and closed for each transfer of a file or file list. As soon as a file or a file list has been transferred, the data connection is closed. If there is another file transfer, the data connection is re-opened. By default, the data connection is established upon request by the FTP server. The FTP client starts a TCP server that waits for a connection on an ephemeral port, and sends the port number of this port on the control connection to the FTP server. After the FTP server receives the port, it can request a data connection to the FTP client. A security concern with FTP is that the user name and the password submitted by the FTP client on the control connection at the beginning of an FTP session are not encrypted. Therefore, anyone with the ability to capture traffic from an FTP client can obtain the user name and password used by an FTP client.

The FTP client sends commands to the FTP server, and the FTP server responds with a three-digit response code and an explaining text message. The commands from the FTP client as well as the responses from the FTP server are transmitted as ASCII

Characters.13 The end of a client command and the end of a server response is represented by an end-of-line sequence, which consists of the ASCII special characters Carriage Return (ASCII 10) followed by Line Feed (ASCII 13). When the TCP connection to TCP port 21 of the FTP server is established, the FTP server sends a message that it is ready to interact off a new FTP session. Then, the user supplies a user name and a password. If the authentication is successful, the user sends the IP

address and port number of its ephemeral port for the data connection. The IP address and port number is sent in dotted-decimal notation, where the first four numbers specify the IP address and the last two numbers specify the port number. By default, files are transmitted as text files using ASCII characters. However, the FTP client can change this default so that files are transmitted bit-by-bit without any interpretation. When the file transfer has been completed, the FTP server sends a message to the FTP client. At this 13 ASCII (American Standard Code for Information Interchange) is an encoding format for textual data, which represents an alphanumeric character or special character by seven bits. Application-layer protocols transmit each ASCII character in a byte (octet) with the most significant bit set to zero. The FTP client can download or upload another file, or end the FTP sessions by issuing the command "Quit".

3.8.2 Trivial File Transfer Protocol (TFTP)

The **Trivial File Transfer Protocol** (TFTP) is a minimal protocol for transferring files without authentication and no separation of control information and data as in FTP. Therefore TFTP must not be used on computer where sensitive /confidential information is stored. TFTP is frequently used by devices without permanent storage for copying an initial memory image (bootstrap) from a remote server when the devices are powered on. Due to the lack of any security features, the use of TFTP is generally restricted.

TFTP uses the unreliable transport protocol *UDP* for data transport, whereas FTP uses TCP. Each TFTP message is carried in a separate *UDP* datagram. The first two bytes of a TFTP message specify the type of message, which can be a request to download a file, request to upload a file, a data message, or an acknowledgement or error message. A TFTP session is initiated when a TFTP client sends a request to upload or download a file from an ephemeral *UDP* port to the (well-known) UDP port 69 of a TFTP server. When the request is received the TFTP server picks an ephemeral UDP port of its own and uses this port to communicate with the TFTP client. Thus, both client and server communicate using ephemeral ports.

Since UDP does not recover lost or corrupted data, TFTP is responsible for maintaining the integrity of the data exchange. TFTP transfers data in blocks of 512 bytes. Each block is assigned a 2-byte long sequence number and is transmitted in a separate UDP datagram. A block must be acknowledged before the next block can be sent. When an acknowledgment is not received before a timer expires, the block is retransmitted.

3.8.3 Telnet

Telnet is a remote login protocol for executing commands on a remote host. The Telnet protocol runs in a client-server mode and uses the TCP protocol for data transmission. A client initiates a Telnet session by contacting a Telnet server at a remote host. Recently, the use of Tel net in public networks has been discouraged since Telnet does not offer good protection against third parties that can observe ("snoop") traffic between a Telnet client and a Telnet server. At the Telnet client, a character that is typed on the keyboard is not displayed on the monitor, but, instead is encoded as an ASCII character and transmitted to a remote Telnet server. At the server, the ASCII character is interpreted as if a user had typed the character on the keyboard of the remote machine. If the keystroke results in any output, this output is encoded as (ASCII) text and sent to the Telnet client, which displays it on its monitor. The output can be just the (echo of the) typed character or it can be the output of a command that was executed at the remote Telnet server.

Telnet uses a single TCP connection for communications. The Telnet server uses the well known TCP port 23 and the Telnet client uses an ephemeral TCP port. After establishing the TCP connection, the Telnet client and server negotiate a set of parameters for the Telnet session, including terminal type, line speed, if typed characters should be echoed to the client or not, and so on. Unless a Telnet session is explicitly configured not to do so, Telnet client sends one TCP segment for each typed character. Telnet provides some independence from the differences of hardware and software at hosts by mapping input and output to a virtual device, called Network Virtual Terminal (NVT) or pseudo terminal. The Telnet client and Telnet server map the input from a keyboard and the output to a monitor to the ASCII character set. Thus, before a character is sent over the network it is encoded as ASCII character. Also, when an ASCII character is received on the TCP connection, the receiving host interprets the character and translates it into its local character format.

3.8.4 Remote Login

Rlogin is an alternative remote login application program for hosts that run the Unix operating system. Rlogin takes advantage of the fact that both the client and server run a similar operating system, and, for this reason, is simpler than Telnet. Rsh is an application program for the execution of a single command on a remote Unix host. There is also an application program for file transfers between Unix hosts, called rcp. However, this group of applications has poor security features, and is, therefore, often disabled.

The Secure **Shell** suite of protocols provides application layer services for remote login and file transfer services, similar to FTP, Telnet, rlogin, rsh, and rcp, but ensures secure encrypted communications between untrusted hosts. **All** components of Secure, Shell provide authentication confidentiality, and integrity of data, using a variety of encryption and authentication algorithms, and protect against common attacks on the security or integrity of communications between hosts.

3.8.5 Electronic Mail (Email)

Electronic Mail (email) is the primary Internet service for exchanging messages between users at different hosts (computer) on the Internet. The exchange of email messages is asynchronous, meaning that the transmission and retrieval of all email messages can occur at different times.

A user on a host prepares an email on a mail preparation program, called a user agent. Examples of user agents on Unix operating systems are mail, xmail, elm, or pine 14. Email messages are written in plain text and users can add non-text files to an email message. The user agent passes the email to a mail server, where the message is queued for transmission, the user agent and the mail server are on the same host, but it is also possible that they are on different hosts. The mail server uses the application- layer protocol SMTP (Simple Mail Transfer Protocol) to transmit the email message to the mail server of the receiver of the email. The sending mail server uses DNS, the domain name service, to locate the correct remote mail server. In addition to translating host names into IP addresses, DNS also provides the IP addresses of mail servers. For the email, the mail server at Host A queries DNS for the IP address of the mail server that is responsible for the domain of the email receiver. Once the IP address of the remote mail server is obtained, the sending mail server starts an SMTP client and initiates a TCP connection to the SMTP server of the remote mail server at the well-known TCP port 25. As soon as the TCP connection is established, the SMTP client and server exchange SMTP commands and transfer the email message. These are user agents for hosts with a UNIX operating system. Commands and the email message are transmitted as plain text using ASCII characters. If an email message contains parts that are not text files, these parts are converted to ASCII characters before transmission. As in FTP, all messages are transmitted as ASCII characters, using the end-of-line sequence to indicate the end of a command. The SMTP client issues commands to the server, and the server responds to each command with a three-digit response code and explaining text. After the TCP connection is established, the remote mail server sends a brief message.

An email message may traverse multiple SMTP servers before reaching its destination. For example, on most networks a single host is dedicated to handle all outgoing email messages for all hosts of the network. In this case, all hosts forward their outgoing email message to the dedicated mail server, which relays the emails to the proper destinations. Also, in many networks, the receiving mail server does not have access to the receiver's mailbox, and relays incoming messages to the mail server on a host where the receiver's mailbox resides. When a mail server relays an email message, it adds lines to the header of an email message. These lines can be used by the receiver of all email messages to trace the path of an incoming email. A mail server adds incoming emails to the mailbox of a user, and assumes that the user has access to the mailbox. Often, however, a user is not permanently connected to its mail server. In such situations, the user can employ mail access protocols to retrieve mail messages from their mailboxes at a remote host. Currently, two popular mail access protocols are in wide use: Post Office Protocol Version 3 (POP3) and Internet Mail Access Protocol (IMAP). Mail access protocols are generally integrated as a component of the user agent.

Exchange of POP3 messages between a POP3 client and a POP3 server. An exchange of POP3 commands between a POP3 client and a POP3 server. All commands and messages are in plain ASCII text, and lines are separated by an end-offline sequence (CR and LF). The POP3 server acknowledges each command from the client.

For outgoing messages, the user agent at Host X still uses SMTP. The SMTP client of the user agent at Host X connects to the SMTP server of Host A. Here, Host A acts as a relay and forwards the message to the mail server of the email receiver. Alternatively, the SMTP client at Host X can directly connect to the mail server of the receiver.

Browser-based Emails

Browser based emails allow user to access emails through web browser. The user agent (the web browser) is used HTTP (not POP/IMAP) for the interaction between the browser and email server. When a user wants to send or view the received mail, the browser sends all such commands in the form of HTTP

Example of browser based emails are rediffmail, Gmail, etc.

Multipurpose Internet Mail extensions (MIME)
Traditional email systems are text based. *Multipurpose Internet Mail extensions (MIME)* system extends the basic email system by permitting users to send binary file, e.g., multimedia file, any other arbitrary format

3.9 World Wide Web (www or Web)

The World Wide Web (WWW or Web) emerged in the early 1990s as a new application for access to content stored on Internet hosts. Within a few years, the Web became the most popular Internet application, and traffic on the Internet was dominated by Web applications. The Web is a distributed hypertext system, which is implemented as a client-server application.

A Web client program, called a Web browser, retrieves and displays documents from a Web server. The documents, often referred to as web pages, are formatted using the Hypertext Markup Language (HTML). HTML documents are text files that contain HTML tags, which describe how text should be displayed in the user interface of a Web browser. A hyperlink is special type of tag. It is a reference to another document, which may be located at a different Web server. When displayed in a browser, hyperlinks can be activated with a mouse click.

When activated, the browser retrieves the document that is referenced in the hyperlink. A Web browser makes it convenient to access a variety of documents at different web servers, which refer to each other by hyperlinks, thus, providing users with a feeling of navigating a global database of documents. On the Web, the location of a document is expressed in terms of a Uniform Resource Locator (URL). A URL specifies a unique location for a document on the web. It can reference an HTML document, but also any other file that can be accessed with a Web browser. An example of a URL is http://www.ignou.ac.in/index.html, which specifies that an HTML document with name *index.htm!* can be accessed via the protocol *H1TP from a host with the name www.ignou.ac.in*

There is no notion of sessions, as, for example, in Telnet and FTP.

In older versions of HTTP, which are still in use today, an HTTP client initiates one TCP connection for each request to the HTTP server. When a single client issues multiple requests to the HTTP server, the number of TCP connections between the HTTP client and HTTP server can grow large. To reduce the number of TCP connections, HTTP/1.1 the current version of HTTP, permits multiple HTTP requests and responses on the same TCP connection, leaves the TCP connection open after request has been served. The HTTP client does not need to wait until a request is completed before issuing new requests. As in many other application layer protocols of the internet, HTTP messages are transmitted as ASCII text, using the end-of-line character sequence to .indicate the end of a message. The most common HTTP messages sent by a client are requests for HTML or other documents. The server

responds to such a request either with a message that contains the requested document or, if the request cannot be satisfied, with an error code. Let us now discuss a request and a response between an HTTP client and an HTTP. Suppose a user has typed the URL is http://www.ignou.ac.in/index.html in a web browser.

When the URL is typed, the web browser starts an HTTP client, which establishes a r TCP connection to port 80 of the HTTP server of host www.ignou;ac.in

3.10 Domain Name System

The Internet Protocol address is a 32- bit integer. If somebody wants to send a message it is necessary to include the destination address, but people prefer to assign machines pronounceable, easily remembered names (host names). For this reason the Domain Name System is used. These logical names also allow independence from knowing the physical location of a host; A host may be moved to a different network, while the users continue to use the same logical name. The idea behind domain names is simple: Rather than forcing people to memorize IP numbers, why not give them cryptic names to remember instead?

Domain Name System maps a name to an IP address and conversely an address to a name. Initially when the size of the Internet was small all machines used to maintain a host.txt file, which was passed on, incase of Updates. This host.txt file was centrally managed but looking at the present Internet scenario this does not seem to be a feasible option due to the following reasons:

Size of the file will be very large, scalability also becomes an issue

This is centrally managed but since the Internet has distributed management, the management of name space should also be of distributed nature.

There can be inconsistent results for queries.

Because of the centralized management as the frequency of lookups increases the time for reply can be very large.

The Domain Name System (DNS) is a distributed database used by TCP/IP applications to map between hostnames and IP addresses, and to provide electronic mail routing information. Each site (university department, campus, company, or department within a company, for example) maintains its own database of information and runs a server program that other systems across the Internet can query. The DNS provides the protocol, which allows clients and servers to communicate with each other.

The system accesses the DNS through a resolver. The resolver gets the hostname and returns the IP address or gets an IP address *(Figure* 6) and looks up a hostname. As we can see in *Figure* 6 the resolver returns the IP address before asking the TCP to open a connection or sending a datagram using UDP.

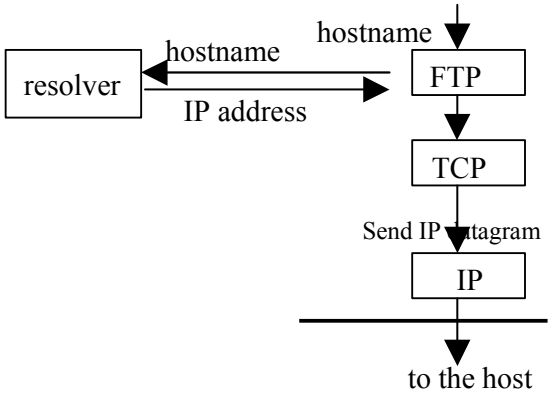


Figure 6: DNS working scheme

DNS Design Goals

The primary goal is a consistent name space, which will be used for referring to resources. Names should not be required to contain network identifiers, addresses, routes, or similar information as part of the name. Name space should be maintained in a distributed manner, with local caching to improve performance. Mechanisms for creating and deleting names; these should also be distributed.

The costs of implementing such a facility dictate that it is generally useful, and not restricted to a single application. We should be able to use names to retrieve host addresses, mailbox data and other as yet undetermined information. All data associated with a name is tagged with a type, and queries can be limited to a single type.

The name space should be useful in dissimilar networks and applications. In short design goal of DNS:

1) Distributed ownership: Since the Internet has distributed ownership; the ownership of name space should also be of distributed nature.

2) Have no obvious size limits for names, name components data associated with a name, etc.
3) DNS protocol should be independent of the network topology.
4) OS/Architecture independent

Design Principles

Hierarchy: The name space as well as management space should be hierarchical. The name space can be represented as a tree with the root label as a nu.11 string. The domain name system uses a hierarchical naming scheme known as domain names, which is similar to the Unix file system tree. The root of the DNS tree is a special node with a null label. The name of each node (except root) has to be up to 63 characters. The domain name of any node in the tree is the list of labels, starting at that node, working up to the root, using a period ("dot") to separate the labels (individual sections of a name might represent sites or a group, but the domain system simply calls each section a label). Thus, the domain name "ignou.ac.in" contains three labels: "ignou", "ac", and "in". Any suffix of a label in a domain name is also called a domain. In the above example the lowest level domain is "ignou.ac.in" (the domain name for the IGNOU in India), the second level domain is "ac.in" (the domain name for Academic organizations of India), and the top-level domain (for this name) is "in" (the domain name for India). The node in is the second level node (after root) *(Figure* 7).

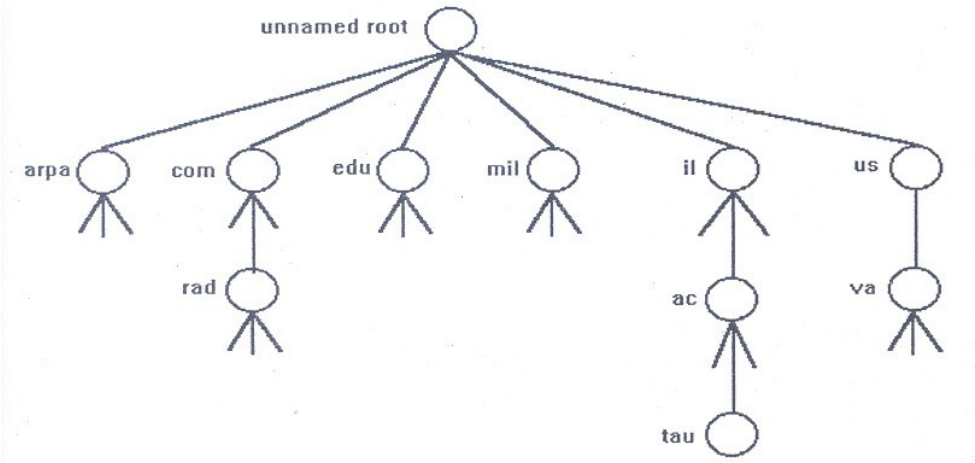


Figure 7: Hierarchical Organisation of the DNS

Caching: A fundamental property of the DNS is caching. That is, when a name server receives information about a mapping, it caches that information. Thus a later query for the same mapping can use the cached result. The DNS uses the caching for optimizing search cost. Caching is required as otherwise there will be:

long time for lookup
congestion at the root server

Every server has a cache for recently used names as well as a record of where the mapping information for that name was obtained. When a client asks the server to resolve certain name the server does as follows:

1) Check if it has authority for the name. If yes, the server doesn't need caching information.
2) If not, the server checks its cache whether the name has been resolved recently. If yes, the server reports the caching information to its clients.

We can examine the cache when the server cashed the information once, but didn't change it. Since information about a particular name can change, the server may have incorrect information in its caching table. The Time to Live (TTL) value is used to decide when to age information. Whenever an authority responds to a request, it includes a TTL value in the response, which specifies how long it guarantees the binding to remain.

DNS Architecture

NAME 'SERVERS are server programs, which hold information about the domain tree's structure and set information. A name server may cache structure or set information about any part of the domain tree, but in general a particular name server has complete information about a subset of the domain space, and pointers to other name servers that can be used to lead to information from any part of the domain tree.

REVOLVERS are programs that extract information from name servers in response to client requests. Revolvers must be able to access at least one name server and use that name server's information to answer a query directly.

Data in the DNS consists of Resource Records. There exists a data type for each record. It is of the form (A, MX) where A is the 32-bit IP address, MX is a 16-bit value along with a host name which acts as the mail exchange for the domain. DNS can be used for both forward lookup (host name to IP address) and reverse lookup (JP address to host name). Name space has an entire subtree for reverse mapping e.g. INADDR.ARPA for reverse lookup

DNS Zones

The zone *(Figure* 8) is a subtree of the DNS that is administered separately. Whenever a new system is installed in a zone, the DNS administrator for tile zone allocates a name and an IP address for the new system and enters these into the name server's database. Within a zone DNS service for subsidiary zones may be delegated along with a subsidiary domain. A name server can support multiple zones.

Zones 'are contiguous regions of the name space, where each can be forked into sub zones. Each of these sub zones can have its independent management.

For example, the IGNOU zone has 2-sub zones cse and lib, which have their own management.

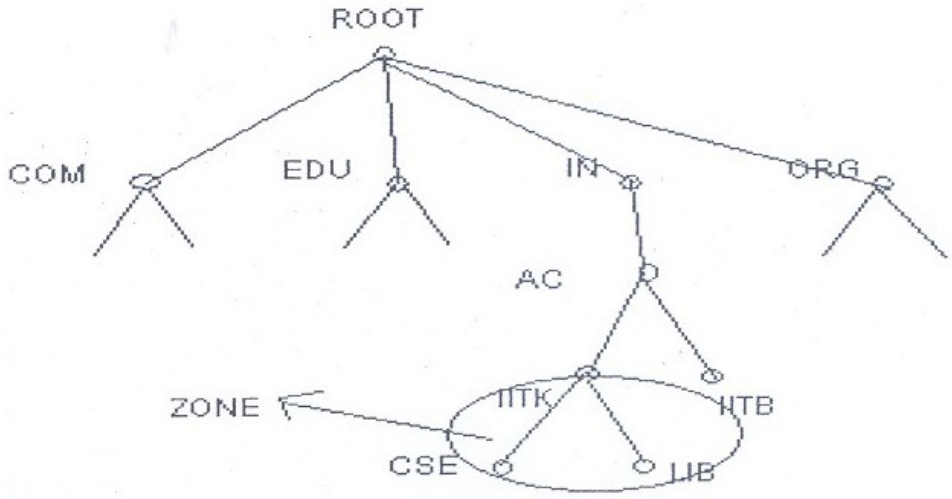


Figure 8: DNS Zones

A name server can support multiple zones; several sub zones can use the same name server. Name servers have pointers among each other.

Remarks on DNS

DNS uses datagram based access, although a DNS query requires reliability, TCP is not used, as it is a query response mechanism.
Root server is replicated for improved reliability.
Address resolution is done recursively. Any DNS server will pass requests it cannot handle to a higher-level server and so on until either the request can be handled or until the root of the DNS name space is reached.

DNS for System Break-In

DNS is highly vulnerable to attacks and spoofing. An intruder can intercept virtually all requests to translate names to IF addresses, and supply the address of a subverted machine instead; this would allow the intruder to spy on all traffic, and build a nice collection of passwords if desired.

IP spoofing attacks can be prevented to an extent. Ssh provides an improved type of authentication. The server has a list of host keys stored in /etc/ssb_known_host, and additionally each user has host keys in $HOME/.ssh/known_hosts. Ssh uses the name servers to obtain the canonical name of the client host, looks for its public key in its known host files, and requires the client to prove that it knows the private host key. This prevents IP and routing spoofing attacks.

Rlogin and rsh permit ordinary users to extend trust to remote host/user combinations. In that case, individual users, rather than an entire system, may be targeted by source routing attacks. The information required for this attack are the target hostname, trusted hostname and the user name, which are obtained by the "finger" command.

Attack is done as below:

In spoofing a host or application to mimic the actions of another. The attacker pretends to be an innocent host by following IF addresses in network packets. rlogin service can use this method to mimic a TCP connection from another host by guessing TCP sequence numbers.

These attacks can be prevented by:

Prevent datagram routing with invalid source addresses.
Introduce unpredictability into connection control mechanisms, such as TCP sequence numbers and the allocation of dynamic port addresses.
.Letting rsh/rlogin to do forward loop along with the reverse lookup.

Allowing to do forward lookup creates a problem called" poisoning the cache" where the attacker sends an unsolicited record along with the PTR record (PTR-a pointer to another part of the domain name space). This attack can be subverted by rejecting with the record, which arrives along with the PTR record.

3.11 SNMP and UDP

Simple Network Management Protocol (SNMP)

SNMP *(Figure* 9) is the simple network management protocol. It is used by network management frameworks to manage and monitor network devices, such as hubs and routers. Some computer systems also respond to SNMP queries

SNMP is not actually a protocol: it's a client server application that runs on the UDP (User Datagram Protocol) service of the *TCP/IP* protocol suite to manage the network. Network management means to ensure that network is up and running, taking corrective measures and performing maintenance activities. It was developed to be an efficient means of sending network management information over UDP, using Ports 161(SNMP) and 162 (SNMP TRAP).

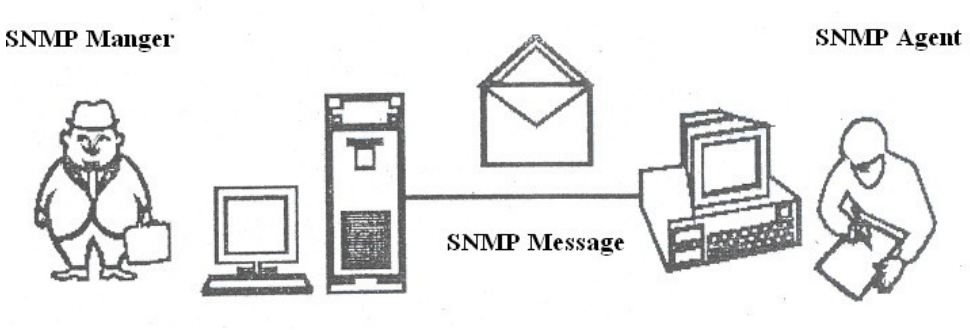


Figure 9: SNMP

Network management system contains two primary elements: a manager and agents. The Manager is the console through which the network administrator performs network management functions. Agents are the entities that interface to tile actual device being managed. Bridges, Hubs, Routers or network servers are examples of managed devices that contain managed objects. These managed objects might be hardware, configuration parameters, performance statistics, and so on, that directly relate to the current operation of the device in question. These objects are arranged in what is known as a virtual information database, called a management information base, also called MIB. SNMP allows managers and agents to communicate for the purpose of accessing these objects.

The model of network management architecture looks like *(Figure 10)* this:

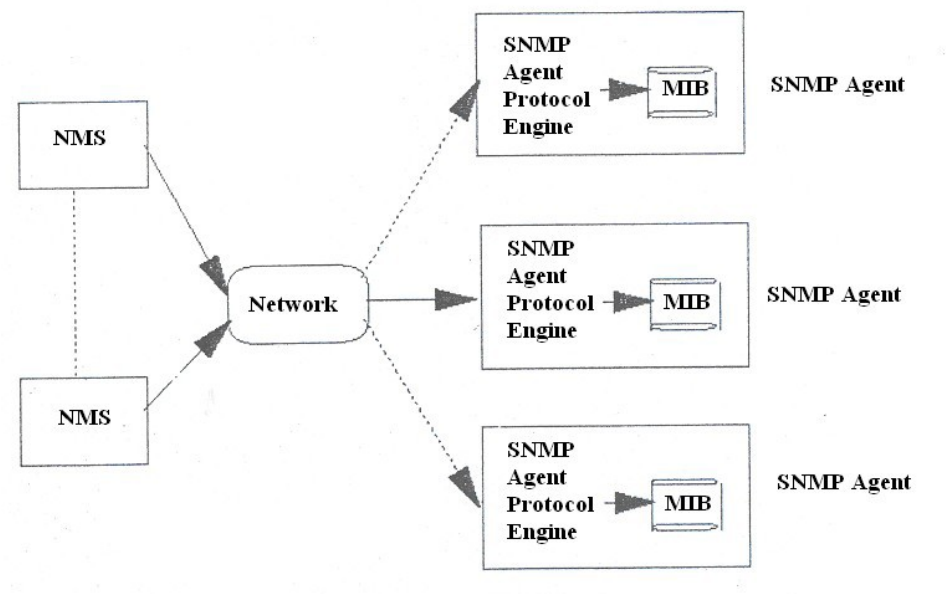


Figure 10: SNMP Architecture

A typical agent usually:

Implements full SNMP protocol.
Stores and retrieves management data as defined by the Management
Information Base.
Can asynchronously signal an event to the manager.
Can be a proxy for some non-SNMP manageable network node.

A typical manager usually:

Implemented as a Network Management Station (the NMS)
Implements full SNMP Protocol

Able to:

- Query agents
- Get responses from agents
- Set variables in agents
- Acknowledge asynchronous events from agents

SNMP uses the *User Datagram Protocol* (UDP) as the transport protocol for passing data between managers and agents. UDP was chosen over the *Transmission " Control Protocol* (TCP) because it is connection less; that is, no end-to-end connection is made between the agent and the NMS when *datagrams* (packets) are sent back and forth. This aspect of UDP makes it unreliable, since there is no acknowledgment of lost datagrams at the protocol level. It's up to the

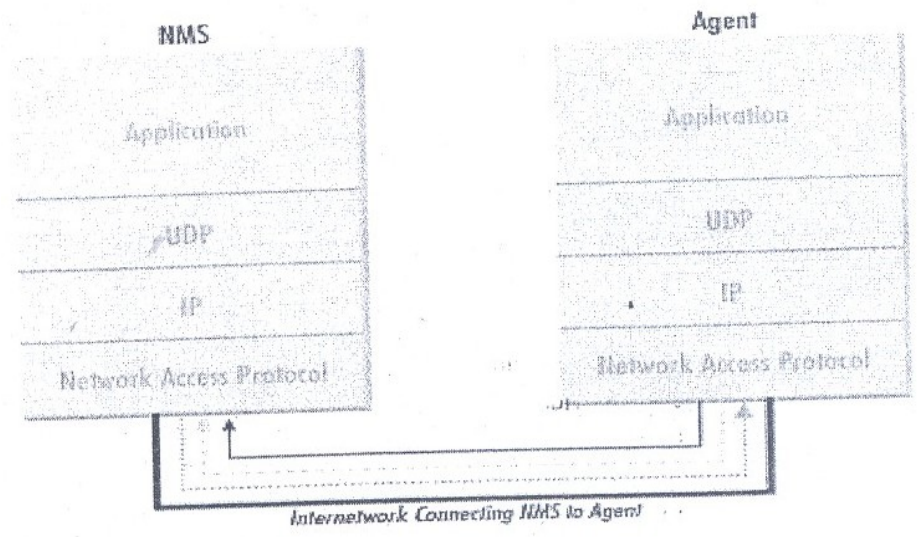
SNMP application to determine if datagrams are lost and retransmit them if it so desires. This is typically accomplished with a simple timeout. The NMS sends a UDP request to an agent and waits for a response. The length of time the NMS waits depends on how it's configured. If the timeout is reached and the NMS has not heard back from the agent, it assumes the packet was lost and retransmits the request. The number of times the NMS retransmits; packets IS also configurable.

At least as far as regular information requests are concerned, the unreliable nature of UDP isn't a real problem. At worst, the management station issues a request and never receives a response. For traps, the situation is somewhat different. If an agent sends a trap and the trap never arrives, the NMS has no way of knowing that it was .I ever sent. The agent doesn't even know that it needs to resend the trap, because the NMS is not required to send a response back to the agent acknowledging receipt of f the trap.

The upside to the unreliable nature of UDP is that it requires low overhead, so the impact on your network's performance is reduced. SNMP has been implemented over TCP, but this is more for special-case situations in which someone is developing an agent for a proprietary piece of equipment. In a heavily congested and managed network, SNMP over TCP is a bad idea. It's also worth realizing that TCP isn't magic, and that SNMP is designed for working with networks that are in trouble-if your network never failed, you wouldn't need to monitor it. When a network is failing, a protocol that tries to get the data through but gives up if it can't is almost certainly a better design choice than a protocol that will flood the network with retransmissions in its attempt to achieve reliability.

SNMP uses the UDP port 161 for sending and receiving requests, and port 162 for receiving traps from managed devices. Every device that implements SNMP must use these port numbers as the defaults, but some vendors allow you to change the default ports in the agent's configuration. If these defaults are changed, the NMS must be made aware of the changes so it can query the device on the correct ports.

Figure 11 shows the TCP/IP protocol suite, which is the basis for all TCP/IP communication. Today; any device that wishes to communicate on the Internet (e.g., Windows NT systems; Unix servers, Cisco routers, etc.) must use this protocol suite. This model is often referred to as a protocol stack, since each layer uses the information from the layer directly below it and provides a service to the layer directly above it.



KEY

——— Trap sent to port 162 on the NMS

——— SNMP request sent from the NMS to the agent on port 161

——— Response to SNMP request sent from the agent to port 161 on the NMS

Figure 11: TCP/IP communication model and SNMP

When either an NMS or an agent wishes to perform an SNMP function (e.g., a request or trap), the following events occur in the protocol stack:

Application

First, the actual SNMP application (NMS or agent) decides what it is going to do. For example, it can send an SNMP request to an agent, send a response to an SNMP request (this would be sent from the agent), or send a trap to an NMS. The application layer provides services to an end user, such as operator requesting status information for a port on an Ethernet switch.

UDP

The next layer, UDP, allows two hosts to communicate with one another. The UDP header contains, among other things, the destination port of the device to which it is sending the request or trap. The destination port will either be 161 (query) or 162 (trap).

IP

The IP layer tries to deliver the SNMP packet to its intended destination, as specified by its IP address.

Medium Access Control (MAC)

The final event that must occur for an SNMP packet to reach its destination is for it to be handed off to the physical network, where it can be routed to its final destination. The MAC layer is comprised of the actual hardware and device drivers that put your data onto a physical piece of wire, such as an Ethernet card. The MAC layer also is responsible for receiving packets from the physical network and sending them back up the protocol stack so they can be processed by the application layer (SNMP, in this case).

4.0 CONCLUSION

In this concluding unit of module 1, you have been taken through the concept of packet switching, and internetworking such as internet addresses, TCP/IP, domain name system (DNS) and its architecture and zones, SNMP and UDP.

5.0 SUMMARY

This unit introduced the building blocks on which internet works are built. Internet works are complex systems that, when viewed as a whole, are too much to understand. Only by breaking the network down into the conceptual pieces can it be easily understood. As you read and experience internet works, try to think of them in terms of OS I layers and conceptual pieces.

Understanding the interaction between various layers and protocols makes designing, configuring, and diagnosing internetworks possible. Without understanding of the building blocks, you cannot understand the interaction between them.

Assigning Internet address to the nodes on the network is a very common task you will perform when building a TCP/IP network. Two 1Jjpes of IP addresses exist: those for IP version 4 and those for IP version 6: IP addresses must be unique on a network. in special cases, it is possible to introduce non-unique addresses, called private addresses.

IP addresses posses a certain structure; they are divided into a network number (netid) and a host number (hostid) portion. This chapter examines the strengths and weaknesses of this scheme. Not all IP addresses can be assigned as a unique identification for network connections. Only class A, B, and C addresses are assignable to individual network connections. Class D is used for IP multicasting. In addition, there are several special addresses used for broadcasting, loop back addresses, and special circumstances. Besides IP & TCP we looked at several other protocols.

6.0 TUTOR-MARKED ASSIGNMENT

1) What is a packet?
2) Which layer of the OSI model has been divided into two sublayers, and what?
3) Although UDP is connection less, what is the benefit of UDP?
4) Describe how a TCP message gets from one, place to another in step sequence?
5) What are the various categories of TCP/IP Well-known Services?
6) What is standard HTTP port?
7) Can a server will work as an FTP server and a webserver?
8) Multiple choice

i) *TCP/IP* is the main protocol used by computers on the Internet

   a) TRUE
   b) FALSE

ii) If I wanted to login to a host computer via the Internet, I would use

   a) telnet
   b) traceroute
   c) snmp
   d) ftp

iii) To determine if another computer was reachable (alive), the utility I could use is

   a) ping
   b) nslookup
   c) telnet
   d) ftp

iv) My computer is very slow. I wish to take advantage of a much faster computer on the network and have my program run there. Which of the following would I use?

   a) ftp
   b) snmp
   c) rsh
   d) ping

9) I need to print a document on a printer attached to a UNIX server. Which, 1 command would I use?

a) telnet
b) ping
c) rsh
d) Ipr

10) I wish to gain some statistics from a network server about the number of data packets being sent and received. Which of the following should I use?

a) snmp
b) telnet
c) rsh
d) 1pr

7.0 REFERENCES/FURTHER READINGS

"Internetworking with TCPIIP. Douglas Comer, Volume 1, Fourth Edition", Prentice Hall, 2000.

Computer Networking. Kurose and Ross, Second Edition, Addison-Wesley, 2003 (optional, in RBR).

"Computer Networks: A Systems Approach", Peterson & Davies, Morgan Kaufmann, Second Edition, 2000 W.

"TCPI/P Illustrated', volume I, The Protocols, Richard Stevens, Addison- Wesley, 1994.

"Internetworking with TCPI/P", Douglas Comer, Volume 3, BSD Socket Version, Prentice Hall, 1993.

"Computer Networks", Tanenbaum, Third Edition, Prentice-Hall 1996.

Data Networks, Dimitri Bertsekas and Robert Gallagher, Second Edition, Prentice-Hall 1992.

"OS/ A *Model for Computer Communication Standards",* Uyless Black, Prentice Hall, 1991.

"Data and Computer Communications", William Stallings, Fourth Edition, Macmillan, 1994.

MODULE 2 LINUX OPERATION SYSTEM

UNIT 1 INTRODUCTION TO LINUX OPERATING SYSTEM

CONTENTS

1.0 INTRODUCTION

Today's trends are towards development of free and platform independent software. Java popularised this concept in the field of programming language and now it is being done by Linux in the operating system.

Linux started out as a Unix variant to run on an IBM PC platform but with a major difference- its source code was freely available under the auspices of Free Software Foundation (FSF). Due to this, it quickly positioned itself as an alternative to other Unix workstations such as those offered by Sun Microsystem, Compaq and Sillicon Graphics. Due to high quality designing of its kernel qualities such as stability, modularity and easy configurability -it is now dominating tile corporate world significantly. For example, major banks, investment houses, retail establishments, educational institutions, etc., use it.

2.0 OBJECTIVES

After going through this unit you should be able to:

list the features of Linux
list the drawbacks of Linux
understand the process of thread management
understand the memory management features.

3.0 MAIN CONTENT

3.1 Features of Linux

Linux has several strong features. In this section we will discuss and explain them.

Linux is Inexpensive

The first benefit of Linux is cost. All versions of Linux may be freely downloaded from the web. If you don't want to download, prepackaged versions of Linux may be purchased online. In addition, the software may be legally shared with your friends. In addition, when the time comes to upgrade the operating system, the Linux upgrade would be free.

In addition to being inexpensive, Linux can run on the old system. Its products can run on Intel 386 microprocessors, which were popular in the late 1980s. The server has never slowed down despite increased use.

Linux is Fast

Linux runs respectably well on old computers, and it is even faster on newer, more powerful computers. This is because Linux programs are very efficient and lean. They use as few resources as possible, and unlike Windows, Linux programs use little, if any, graphics. Graphics can slow a system's response time, making it slower than it truly is. Linux may not be pretty, but it is fast.

Linux is Stable

The Linux code is well written. This both increases the speed at which Linux runs and improves the stability of the operating system. Linux is next to impossible to crash. If an application crashes, you can simply remove the program from memory to restart your computer. In older versions of Windows, a crashing program had the potential to take down the entire computer. This is one of the reasons why Linux is used on

many web servers where stability is crucial. With Linux, web-hosting providers can guarantee 99.9 percent uptime.

Open-Source Software

Finally, Linux has open-source software. This means that users can read the source code and modify it as needed. This probably means little to the average user of the final version of a Linux kernel. However, during development, "beta" releases of the kernel are available to developers who will download the code and test it thoroughly. When possible, they will find any problems and correct the code. This process helps to ensure that the final release of the kernel is as well written as possible.

3.2 Drawbacks of Linux

Even though Unix and Linux operating systems are widely used on corporate servers, l web sites, and large-scale networking environments, we still won't find many people r using it on their desktop computers or workstations at home. There are several reasons for this.

Security

Because code is distributed with the Linux software, programmers are free to explore how the system works -for good or bad. Many security loopholes have been reported in the literature. Although most system vulnerabilities are detected before the product is released, clever programmers can still discover new ones.

Lack of Support

No system is 100 percent secure. Even Microsoft products have security vulnerabilities. However, Microsoft products do have extensive documentation and support. Microsoft releases service pack and updates frequently to fix discovered vulnerabilities.

Support and documentation for Linux can be spotty at best. A customer who downloads Linux from a server may receive only an electronic manual and access to online help pages.

Limited Software Selection Choice

People purchase computers to run software. Users of Windows computers have many software titles to choose from. Linux users have software in every category but are often limited in their choices. For example, consider Internet browsers. The two most popular browsers are Netscape Navigator and Microsoft Internet Explorer. Both are available

for Linux, but only Netscape has created a Linux version of its latest browser, version 6.32. Internet Explorer's most recent Linux version is 5, despite the fact that Windows XP ships with Internet Explorer version 6.

You are also limited in your choice of word processors. The most popular word processor is Microsoft Word. Chances are that every computer in your school uses Word. But Word is not available for Linux. Your best choice is **StarOffice**, a suite of applications that contains a word processor. However, StarOffice, although a very nice product, is not Word. A proficient Word user would have to learn some new skills to use StarOffice.

Limited Hardware Support

Just as not all popular software run on Linux, not all hardware products work with Linux. Red Hat and the other Linux vendors work very hard to support the more common devices. They provide drivers for hardware devices. A driver is a small program that allows the operating system to communicate with the peripheral devices. Having the correct driver is crucial. If you have a new or unusual device, you may be out of luck. For instance, many branded companies have not written a Linux Compatible driver.

Complexity

Linux, like its predecessor Unix, assumes that you know what you are doing, and it assumes that you know the consequences of every command you type. In contrast, Windows XP asks you to verify everything, and then shows you a pretty animation to confirm the action.

For a beginning user, Linux can be frightening to use; entering the wrong command can have serious consequences. It doesn't help that Linux is also case sensitive, so you must enter the commands in lowercase, and be careful to use the correct case for each subcommand you use with a command. Upper -and lowercases are often different actions.

3.3 Components of Linux

In this section we will take up the various component of Linux Operating System.

3.3.1 Memory Management Subsystem

Linux is made up of a number of functionally separate pieces that, together, comprise the operating system. One obvious part of Linux is the kernel itself; but even that would be useless without libraries or shells. In this section we will discuss the various components of Linux kernel.

One of the basic objectives of any operating system is to make one feel that there is a large amount of memory although it is having a small physical memory. This apparently large memory is known as virtual memory. The system divides the memory into easily handled pages (logical unit) and swaps these pages onto a hard disk as the system runs.

The memory management subsystem is one of the most important parts of the operating system. Since the early days of computing, there has been a need for more memory than exists physically in a system. Strategies have been developed to overcome this limitation and the most successful of these is virtual memory. Virtual memory makes the system appear to have more memory than it actually has by sharing it between competing processes as they need it.

Virtual memory does more than just make your computer's memory go further. The memory management subsystem includes:

Large Address Spaces: The operating system makes the system appear as if it has a larger amount of memory than it actually has. The virtual memory can be many times larger than the physical memory in the system.

Protection: Each process in the system has its own virtual address space. These virtual address spaces are completely separate from each other and so a process running one application cannot affect another. Also, the hardware virtual memory mechanisms allow areas of memory to be protected against writing. This protects code and data from being overwritten by rogue applications.

Memory Mapping: Memory mapping is used to map image and data files into a processes address space. In memory mapping, the contents of a file are linked directly into the virtual address space of a process.

Fair Physical Memory Allocation: The memory management' subsystem allows each running process in the system a fair share of the physical memory of the system.

Shared Virtual Memory: Although virtual memory allows processes to have separate (virtual) addresses spaces, there are times when you need processes to share memory. For example, there could be several processes in the system running concurrently and simultaneously depending upon the number of processors residing in the system but might be using the common file, e.g., C-amplifier.

Therefore, it is better to have only one copy in physical memory and all pf the processes running sharing it. Dynamic libraries are another common example of executing code shared between several processes.

Another example of shared memory is that it can also be used as an Inter Process Communication (IPC) mechanism, with two or more processes exchanging information via memory common to all of them. Linux supports the Unix™ System V shared memory IPC.

An Abstract Model of Virtual Memory

Before considering the methods that Linux uses to support virtual memory it is useful to consider an abstract model which is applicable to a large number of systems.

As the processor executes a program it reads an instruction from memory and decodes it. In decoding the instruction it may need to fetch or store the contents of a location in memory. The processor then executes the instruction and moves onto the next instruction in the program. In this way the processor is always accessing memory either to fetch instructions or to fetch and stored data.

In a virtual memory system all of these addresses are virtual addresses and not physical addresses. These virtual addresses are converted into physical addresses by the processor through a mapping scheme using a set of tables maintained by the operating system.

To make this translation easier, virtual and physical memory are divided into handy sized chunks called pages. These pages are all of the same size. They need not be, but if they were not the system would be very hard to administer. The size of a page varies from one system to another. Each of these pages is given a unique number; the frame number (FN). In this paged model, a virtual address comprises two parts; virtual page frame number (VPFN) and offset within the frame. Each time the processor encounters a virtual address it must extract the virtual page frame number and the offset. The processor must translate the virtual page frame number into a physical one \address of RA~) and then access the location at the correct offset into that physical page. To do this the processor uses page tables. The size of a page table varies from one

process to another and the number of page tables depends upon the number of processes residing in a system.

Figure 1 shows the virtual addresses spaces of two processes, process PI and process Pl, each with its own page tables. These page tables map each processes virtual pages into physical pages in memory. This shows that process P_1's virtual page frame number 0 is mapped into memory in physical page frame number 1 and that process P_2's virtual page frame number 1 is mapped into physical page frame number 4. Entry in the theoretical page table contains the following information:

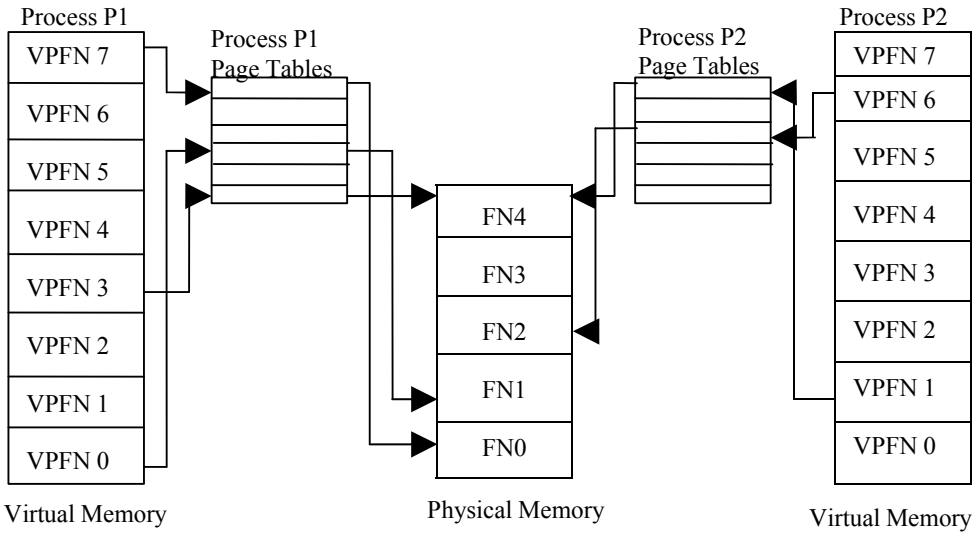


Figure 1: Abstract model of Virtual to Physical address mapping

The processor uses the virtual page frame number as an index into the processes page table to retrieve its page table entry. If the page table entry at that offset is valid, the processor takes the physical page frame number from this entry. If the entry is invalid, the process has accessed a non-existent area of its virtual memory. In this case, the processor cannot resolve the address and must pass control to the operating system so that it can fix things up.

In case the required page frame is not found, the processor generates a page fault and then the required page is brought from the hard disk.

Mapping of virtual address to physical address can be done in any order. For example, in process P, virtual page frame number 0 is mapped to physical page frame number 1 whereas virtual page frame number 7 is mapped to physical page frame number 0 even though it is higher in virtual memory than virtual page frame number 0. This demonstrates an interesting byproduct of virtual memory; the pages of virtual memory do not have to be present in physical memory in any particular order.

Linux shares many of the characteristics of the memory management schemes of other Unix implementations but has its own unique features. Overall, the Linux memory management scheme is quite complex. Here, we give a brief overview.

Linux Virtual Memory

Linux makes use of a three-level page table structure, consisting of the following types of tables (each individual table is the size of one page):

> **Page Directory**: An active process has a single page directory that is the size of one page. Each entry in the page directory points to one page of the page middle directory. The page directory must be in main memory for an active process.
>
> **Page Middle Directory:** The page middle directory may span multiple pages. Each entry in the page middle directory points to one page in the page table.
>
> **Page Table:** The page table may also span multiple pages. Each page table entry refers to one virtual page of the process.

To use this three-level page table structure, a virtual address in Linux is viewed as consisting of four fields. The leftmost (most significant) field is used as an index into the page directory. The next field serves as an index into the page middle directory.

The third field serves as an index into the page table. The fourth field gives the offset within the selected page of memory.

Page Allocation

To enhance the efficiency of reading in and writing out pages to and from main memory, Linux defines a mechanism for dealing with contiguous blocks of pages mapped into contiguous blocks of page frames. For this purpose, the buddy system is used. The kernel maintains a list of contiguous page frame groups of fixed size; a group may consist of I, 2, 4, 16, or 32 page frames. As pages are allocated and deallocated in main memory, the available groups are split and merged using the buddy algorithm.

Page Replacement Algorithm

The Linux page replacement algorithm is based on the clock algorithm in which a use bit and a modify bit are associated with each page in main memory. In the Linux scheme, the use bit is replaced with an 8-bit

age variable. Each time that a page is accessed, the age variable is incremented. In the background, Linux periodically sweeps through the global page pool and decrements the age variable for each page as it rotates through all the pages in main memory. A page with an age of 0 is an "old" page that has not been referenced in some time and is the best candidate for replacement. The larger the value of age, the more frequently a page has been used in recent times and the less eligible it is for replacement. Thus, the Linux algorithm is a form of least frequently used policy.

3:3.2 Linux Process and Thread Management

Processes carry out tasks within the operating system. A program is a set of machine code instructions and data stored in an executable image on disk and is, as such, a passive entity; a process can be thought of as a computer program in running state. It is a dynamic entity, constantly changing as the machine code instructions are executed by the processor. As well as the program's instructions and data, the process also includes the program counter and all of the CPU's registers as well as the process stacks containing temporary data such as routine parameters, return addresses and saved variables. Linux is a multiprocessing operating system which can support many processes running in parallel. Processes are separate tasks each with their own rights and responsibilities and also running in their own address spaces. If one process crashes it will not cause another process in the system to crash. Each individual process runs in its own virtual address space and is not capable of interacting with another process except through secure mechanisms to be managed by kernel.

The most precious resource in the system is the CPU, usually there is only one except in a multi-processors based system. Linux is a multiprocessing operating system; its objective is to have a process running on each CPU in the system at all times, to maximize CPU utilization. If there are more processes than CPUs (and there usually are), the rest of~ processes must wait before a CPU becomes free until they can be run. In a multiprocessing system many processes are kept in memory at the same time. Whenever a process has to wait, the operating system takes the CPO away from that process and gives it to another, more deserving process. It is the scheduler which chooses which is the most appropriate process to run next and Linux uses a number of scheduling strategies to ensure fairness.

Linux supports a number of different executable file formats, ELF (Executably and linkable format) is one, Java is another and these must be managed transparently.

Data Structure for Linux Processes

So that Linux can manage the processes in the system, each process is represented by a task-struct data structure (task and process are terms that Linuxuses Interchangeably). The task vector is an array of pointers to every task-struct data structure in the system. This means that the maximum number of processes in the system is limited by the size of the task vector; by default it has 512 entries. As processes are created, a new task-struct is allocated from system memory and added into the task vector. To make it easy to find, the current, running process is pointed to by the current pointer.

As well as the normal type of process, Linux supports real time processes. These processes have to react very quickly to external events (hence the term "real time") and they are treated differently from normal user processes by the scheduler. Although the task-struct data structure is quite large and complex, its fields can be divided into a number of functional areas:

State: As a process executes its changes state according to its circumstances. Linux processes have the following states:

Running: The process is either running (it is the current process in the system) or it is ready to run (it is waiting to be assigned to one of the system's CPUs).

Waiting: The process is waiting for an event or for a resource. Linux differentiates between two types of waiting process; interruptible and uninterruptible. Interruptible waiting processes can be interrupted, by signals whereas uninterruptible waiting processes are waiting directly on hardware conditions and cannot be interrupted under any circumstances.

Stopped: The process has been stopped, usually by receiving a signal. A process that is being debugged can be in a stopped state.

Zombie: This is a halted process which, for some reason, still has a task-struct data structure in the task vector. It is what it sounds like, a dead process.

Scheduling Information: The scheduler needs this information in order to fairly decide which process in the system most deserves to run,

Identifiers: Every process in the system has a process identifier. The process identifier is not an index into the task vector, it is simply a number. Each process also has User and group identifiers, these are used to control this processes access to the files and devices in the system.

Inter-Process Communication (IPC): Linux supports the classic Unix™ IPC mechanisms of signals, pipes and semaphores and also the System V IPC mechanisms of shared memory, semaphores and message queues to allow processes to communicate with each other and with the kernel to coordinate their activities.

Links: In a Linux system no process is independent of any other process. Every process in the system, except the initial process has a parent process. In Unix operating system the initial process is known as init. New processes are created; they are copied, or rather cloned from previous processes. Every task-struct representing a process keeps pointers to its parent process and to its siblings (those processes with the same parent process) as well as to its own child processes.

Times and Timers: The kernel keeps track of a process creation time as well as the CPU time that it consumes during its lifetime. Each clock tick, the kernel updates the amount of time in jiffies that the current process has spent in system and in user mode. Linux also supports process specific interval timers, processes can use system calls to set up timers to send signals to themselves when the timers expire. These timers can be single-shot or periodic timers.

File System: Processes can open and close files as they include pointers to any files opened by this process.

Virtual Memory: Most processes have some virtual memory (kernel threads and daemons do not) and the Linux kernel must track how that virtual memory is mapped onto the system's physical memory.

Processor Specific Context: A process could be thOUgl1t of as the sum total of the system's current state. Whenever a process is running it is using the processor's registers, stacks and so on. This is the processes context and, when a process is suspended, all of that CPU specific context must be saved in the task-struct for the process. When a process is restarted by the scheduler its context is restored from here.

Linux Threads

A new process is created in Linux by copying the attributes of the current process. A new process can be cloned so that it shares resources, such as files, signal handlers, and virtual memory. When the two processes share the same virtual memory, they function as threads within a single process. However, no separate type of data structure is defined for a thread. Thus, Linux makes no distinction between a thread and a process.

3.3.3 File Management Subsystem

In Linux, as it is for Unix, the separate file systems that the system may use are not accessed by device identifiers {such as a drive number or a drive name) but instead they are combined into a single hierarchical tree structure that represents the file system as a single entity. Linux adds each new file system into this single file system tree as they are mounted onto a mount directory, for example *I* mnt/cdrom. One of the most important features of Linux is its support for many different file systems. This makes it very flexible and well able to coexist with other operating systems. The most popular file system for Linux is the EXT2 file system and this is the file system supported by most of the Linux distributions.

A file system gives the user a sensible view of files and directories held on the hard disks of the system regardless of the file system type or the characteristics of the underlying physical device. Linux transparently supports many different file systems (for example MS-DOS and EXT2) and presents all of the mounted files and file systems as one integrated virtual file system. So, in general, users and processes do not need to know what sort of file system that any file is part of, they just use them.

The block device drivers hide the differences between the physical block device types (for example, IDE and SCSI) and, so far as each file system is concerned, the physical devices are just linear collections of blocks of data. The block sizes may vary between devices, for example 512 bytes is common for floppy devices whereas 1024 bytes is common, Air IDE devices and, again, this is hidden from the users of the .system. An EXT2 file system looks the same no matter what device holds it.

3.3.4 Device Drivers

Device drivers make up the major part of the Linux kernel. Like other parts of the operating system, they operate in a highly privileged environment and can cause disaster if they get things wrong. Device drivers control the interaction between the operating system and the peripheral devices that they are controlling. For example, the file system makes use of a general block device interface when writing blocks to a disk. The driver takes care of the details and makes device specific things happen. Device drivers are specific to the controller chip that they are driving.

4.0 CONCLUSION

This unit has introduced you to Linux operating system: its features, drawbacks and components.

It has also discussed to details how memory management, file management and device drivers are handled in this operating system.

5.0 SUMMARY

Linux is a Unix like operating system, with the major difference that its source code W is freely available. In this unit we have described the strong features of the operating: system and also highlighted its drawbacks. Linux like any other operating system is made up of a number of functionally separate precis. The kernel is the most important component of the system. It deals Memory Manager, File Manager, Process Manager and I/O Manager. Device Drivers make up tile major part of the Linux kernel. They control the interaction between the Operating system and peripheral devices. One of the main objectives of Memory Management is to make available to the process a large amount of Memory although it is having a small physical memory through virtual memory concept. Although virtual memory allows processes to have separate address space it also provides shared space among many processes; The Unit also describes tile process and Thread Management.

6.0 TUTOR-MARKED ASSIGNMENT

1) What are the features of Memory Management subsystem?
2) What are the different states of a Linux operating system?
3) What is the purpose of a file system?

7.0 REFERENCES/FURTHER READINGS

Operating systems, 4th Edition, William Stallings, PHI.

UNIT 2 LINUX COMMANDS AND UTILITIES

CONTENTS

1.0 INTRODUCTION

This unit introduces you to Red Hat Linux 9 (hereinafter referred to as Linux) and tells you how to start working on your Linux computer. A few elementary commands are all you need to get the feel of what working in a Linux environment is like. The attempt is to let you see enough Linux features to allow you to walk up to a computer running Linux, login and start working on it, However, in this block we will not touch upon the design of Linux. This unit is oriented towards acquainting you with the richness of the system and making you comfortable with the Linux environment. Apart from the academic, theoretical and sociological aspects of tile development of Linux, it is undoubtedly a rich, open and otherwise convenient operating system that gives you a bewildering array of tools to be productive. Unlike the earlier versions of UNIX, Linux is not open to the charge of being arcane and difficult to use. It is growing in popularity all over the world and is all set to enter tile mainstream of corporate computing. If you are accustomed to an operating system like Microsoft Windows, then you might find Linux a bit different in terms of took and feel and as far as the commands go. However, these days the actual operating system you use is less at the forefront than it used to be, say, a decade ago, unless

you are a software developer. The focus has now shifted more and more to the user and tile facilities that the system can give him.

2.0 OBJECTIVES

After going through this unit you should be able:

to start and carry on a login session under Linux
to change your account password
to understand basic Linux concepts like the hierarchical directory structure
to understand the various types of files under Linux
to close a login session.

3.0 MAIN CONTENT

3.1 Entering the Machine

We will now start learning how to use a Linux computer. This unit will talk about the basic steps involved in logging 011 to a system running Linux and also what you can do once you have gained ingress. But remember that you cannot learn Linux by reading this unit or even this block. That might at best allow you some familiarity with the terminology used and might tell you something about its organisation. You might every come to know something about its features and tile tools available under it. But you: will not be able to work expertly on a Linux computer or feel comfol1able in a Linux environment, much less be productive in it. You will not be able to appreciate the power and beauty of Linux, nor marvel at the collaborative approach that produced it. The only way to learn Linux is by working on a real Linux machine. This unit, this block and other supplementary reading material, together with tile Linux documentation and the many excellent books on the subject, will be a valuable aid in your voyage of discovery. So you must gain access to a working Linux machine arid try out whatever t you feel like. Do not be afraid of exploring or making mistakes.

Whenever you learn about a command or any other feature, do not hesitate to tryout all its variations. Do not confine yourself to only what is mentioned here. This block is of necessity brutally brief and can only serve as an incomplete introduction. Use any other material to which you have access and experiment to your heart's content. You will learn as much from your mistakes and from seeing unexpected outcomes as from things you do by tile book.

3.1.1 User Names and Groups

Every Linux user is given a name when she is allowed access to a Linux system. This is also called an account, as in commercial arrangements an account is kept of tile usage of the machine by each user. The user name need not have any relation to tile actual name of tile user, though it quite often is some abbreviation of the name. For example, a person called Ram Kumar might be given a name kumarr on a Linux system. Here the account name is formed by taking tile surname (abbreviated if it is too long) and the first letter of the first name. It is quite possible for a person to have more than one account on a single machine (in a different name), especially if the person uses the machine in more than one capacity.

Another way of making account names is based on tile role being played by the person. For example, there can be several people working on some programming projects. Ram Kumar could be working on two projects with different teams. In such a case he might have an account name like cryptO2 for his cryptography project, while for his natural language processing project his account might be nlpO4. Such an arrangement helps to keep people in teams part of the same group. One of the motivations for Linux was to allow easy sharing of information, consistent with the needs of security and privacy. So Linux allows account names to be grouped under a common group name. All users belonging to the same group can share group privileges. If Ram Kumar leaves the organization and Zafar Khan takes his place, he might be assigned to continue work on cryptO2 while somebody else might be assigned to nlp04.

Some user names are reserved by Linux for its use, for example, bin and uucp. So you cannot use those names for yourself. There is also a special kind of user on every Linux system who has all possible access rights on the system. This user is called the superuser, the system administrator or simply root because that is the user name conventionally allotted to her. For administrative convenience large systems can have more than one superuser account. The superuser is the one who can create new user accounts, shutdown the system and perform other maintenance tasks.

You would by now be wondering why everybody cannot access the machine as root. The reason is that when you are granted access to a machine you are given a password as well as an account or user name. You are free to choose your password though there are usually certain constraints imposed in the interests of security. So you cannot enter the system as root unless you know the root password. On MY well maintained installation the root password is guarded very carefully, as

public knowledge of this would jeopardize the security of the installation.

While root can access all user files and override any system protection meant for mere mortals, nobody can figure out what your password is. However, root can change or remove your password. A user can also change her password though there are usually some constraints on this too. There can be a minimum period before which you cannot change your password. After a certain maximum period you might be forced to change your password. There can also be constraints on what your password can be. There can be rules that force you to have a minimum password length, include special characters, disallow your previous five passwords and so on.

All the above constraints are meant to make your password difficult to guess. This is needed to make it hard for anyone else to masquerade as you and gain unauthorized access to the machine. Computer security is a matter of great concern to all of us as we become more and more dependent on them for performing our day-to-day tasks. Remember that on the machine your identity is determined by your user name and the machine cannot usually distinguish between physical individuals. However, for high security applications such as military work, access to a machine might be through the use of biometric methods like retinal scans or fingerprints. For most daily applications, however, passwords are still the only means of authentication.

SELF ASSESSMENT EXERCISE 1

1) Can more than one person use the same user account on a Linux system?
2) Can there be more than one account with the same name on a Linux system?
3) Can more than one account have the same password?

3.1.2 Logging In

You will now see how to enter a Linux system so that you can start to use its facilities. This process of gaining access to a system is called logging in. For this you must have a valid user account on the machine and also know your password. In an organisational environment, this set up would have been performed for you by the system administrator of the machine when you were granted permission to use it. In this block we will assume you are working in some organisation. If you are running Linux on a personal machine, then you would have to do all the set up activity yourself, or get somebody to do it for you. Of course, it does mean that you can grant yourself all authority and permissions on

the machine, something that is usually not possible in an organisational context.

There are different ways of connecting to a Linux machine. You could go to the console of a Linux machine, such as a personal computer (PC), start it up and log in. You could be working on a Microsoft Windows machine or even another Linux machine and could connect over the network (whether local or a wide area network) to a Linux machine. Having once gained access you have all the facilities allowed to you by the administrator. There are only a few situations where physical access to the machine makes a difference. In this block we will not dwell on those matters.

When you see the console of your Linux machine or connect to it over the network, you see a message like:

IGNOU Linux machine
Kernel 2.4.20-8 on an i686
Login:

The actual message on these lines could be different or absent depending on the installation. It could be longer, shorter or even be absent. This does not affect anything else that you do in any way.

You should now type in your user name and press the RETURN or ENTER key. In most cases you have to press the RETURN key, sometimes labelled as ENTER, to register what you have typed. You will also find that as you type on the console you will be able to see whatever you have typed. This is because Linux usually echoes whatever you type on the terminal. So your screen should now look like this:

IGNOU Linux machine
Kernel 2.4.20-8 on an i686
login: kumarr
Password:

You must type in your user name, also called the login name, exactly as allocated by your system administrator. This is because Linux is case sensitive, that is, it distinguishes between lower and upper case letters. In this respect it differs from operating systems like VMS or Microsoft Windows. So be careful of small and capital letters when working on Linux.

When Linux asks you for your password, key it in carefully. Notice that your password is not echoed as you type. In fact, the cursor does not

move at all. This is to prevent somebody from reading your password over your shoulder, as that would enable that person to masquerade as you by logging into the computer in your name and using it.

Linux now checks whether you are a valid user and whether you entered the correct password. If there is any mistake you get a message saying:

Login incorrect
Login:

This means you can try to login again. There can be other reasons why you might not be able to login even though you are a valid user and did not make any typing mistake. The messages you get in those situations will however be different.

Why be so pessimistic? Let us assume you have managed to login successfully. The system may then display some messages and finally give you a sign that it is now ready to obey your commands. The messages you see depend on how the system has been configured or set up by the system administrator and by you. So you might not even see any messages. However, usually there is a message indicating when you logged in last. This is useful because if the date and time differ from what you remember about your last login, it could mean that somebody else is using your account.

Let us now look at some of the other common types of messages you see on most systems as you login. These usually give some information about the system like the space available on tile machine, news about the system and whether you have any mail. The news is called the message of the day and appears whenever you login. The message:

You have mail.

Means someone has sent you mail using the user-to-user communication facilities available in Linux.

After the login messages you see a prompt, which is the sign that Linux is ready for your commands. The prompt can be changed to whatever you like but the default prompt also depends on what shell you have been assigned. The usual default in Linux is the Bourne again shell, called bash, which is normally set to have the following prompt:

[kumarr@linuxkumarr]$

This is the prompt we will use throughout the block unless some other prompt is explicitly called for. When you see the prompt on your

terminal it usually means that Linux has finished executing the last command you gave it and is ready for your next command. Here kumarr is your login name, Linux is the name of the machine and the home refers to the directory in which you are located after you login. This is usually your home directory.

There can be limits on the number of attempts, say five, that you can make at logging in. The action taken depends on the installation but can be alerting the system administrator or deactivating the terminal, perhaps for a short time only. So you should be careful not to make too many typing mistakes. In particular be careful not to forget or mistype your password and avoid passwords with certain characters like#.

Now you are still at your console that is black and white. If you want to use the graphical features of Linux you need to get the X Window system up. For this you need to issue the command:

[kumarr@linux kumarr]$ startx_

Whereupon the X Window system will start up and you will see a coloured screen with a bar containing several icons at the bottom. The background and the colour of the screen as well as the icons and facilities available in the tray depend on the configuration of Linux that has been performed during installation or later. You will usually want to work in the X Window system rather than directly on the console as you can then use the mouse and other graphical facilities available in Linux. At this point you are presented with your desktop.

Once here you can open up a terminal window by right clicking the mouse and selecting the appropriate option. You can have as many windows as you like and you can be doing different tasks indifferent windows. It is not that only terminal windows can be opened up. You can click on the icons on the desktop or in the tray and run any applications, such as a browser, office productivity tools and so on.

You can have several desktops. In your tray you will see four rectangles that represent four available desktops to you. All the windows and applications that you are running are associated with the desktop in which you open them. If you want a clean slate where you are doing some other related tasks, you can click on another rectangle and go to that desktop. This is very convenient if you want to do groups of tasks and do not want to clutter up one desktop with too many unrelated windows.

3.1.3 Correcting Typing Mistakes

Many of us are not professional typists and we make a lot of mistakes while typing. In any case all of us are human beings and are prone to error. Whether you are a one or two finger expert or know touch typewriting, you are going to mistype your commands some time or the other. What do you do when you want to find out in a session whether you have any Sundays left in the month? Normally you would use the cal command thus:

Jkumarr@linux kumarr]$ cal

Suppose now that by mistake you type:

[kuma!I@linuxkumarr]$ csl

After you press the RETURN key Linux will say:

-bash: csl: command not found

If you are lucky and a command csl does not exist. If it does it will be executed and you could well be in deep trouble depending 011 what csl does.

You would therefore do better to cancel your command or correct your mistake.
These actions can be accomplished by using the kill and erase characters respectively. The kill character cancels the entire line you typed while the erase character erases or rubs out the last character.

Usually the erase character will be the character AH. This means that you have to type H while holding down the CONTROL key, often abbreviated to Ctrl. This key is normally located on both sides of the keyboard near the Shift keys. In this block we will use the convention of writing AH to mean Ctrl-H and you must be careful not to confuse this with the two separate characters A (circumflex) and H. The backspace key generates the sequence AH on the keyboard and you can use it to delete the character preceding the cursor.

But Linux offers you other facilities to make typing easier. What if you want to have typed 8 characters and want to change the 5th? Just use the left arrow key to go to the character you want to change, press the Delete key and type in the correct character. You can use the right arrow key to go to the right in a command. If you want to repeat a command you issued 3 commands earlier, you can use the up arrow key. Pressing it once brings up the last command you issued, so press it thrice to get the command you issued 3 commands earlier. Then use the RETURN

key as usual to invoke the command, that is, to ask the computer to execute it. You can likewise use the down arrow key to go forward through your command history.

What is more, once you have reached one of your old commands you can edit it by using the left and right arrow keys together with the backspace or delete keys. This is useful if you want to rerun a command with some other options. For example,

{kumarr@linuxkumarr]$ cal2004_

Will give you the calendar for the year 2004. If after issuing a few more commands you want to see the calendar for the year 2003, you do not need to type in the full command again. Just use the up arrow keys to locate the command and to change the 4 in 2004 to 3. Then hit the RETURN key. What a boon for typing challenged people like this author!

You can use the command:

[kumarr@linuxkumarr]$ history_

to see the previous commands that you entered. To run a command entered a long time ago, instead of using the up arrow key, you can say *!* <*no*> where *no* stands for the command number. To run a command *n* commands previously, say

[kumarr@linux kumarr]$!-n

3.1.4 Format of Linux Commands

We will now be at the general format of Linux commands and take the opportunity to study some simple commands. Let us gCbback to the bash shell and the cal command that we mentioned earlier.

[kumarr@.linux kumarr]$ cal_

This gives the calendar for the current month and year (of course this will depend on what the system date has been set to -that is what the computer believes the current month and year to be), and you can use it to, say, find out how many Sundays are left in the month. Another simple command is

[kumarr@linuxkumarr]$ date_

Wed Oct 13 09:20:59 IST2004

which displays the current system date and time. You will realize that the computer has no way of knowing what the current date and time really are, so what it can tell you is whatever it thinks is the current time and date. This can be set by the system administrator to be almost anything, but in most installations, care is taken to see that the date is set correctly. This is because today computers are in general networked with other computers and many system utilities depend on the correct date and time. In Linux when we say date, we mean both the date and time, which is why the output shown above includes th~ time as well.

Notice that the time zone is part of the output. This is significant when you are on a network spanning time zones.

Another simple one word command is:

[kumarr@linuxkumarr]S who_

kumarr ttyl Oct 13 08:41

kumarr pts/O Oct 13 08:54(:0.0)

khanztty2 Oct 13 07:39

This tells you the names of the users currently logged in to the system, their terminal numbers and the date they logged in. You will find that you will always be listed as one of the users, since you usually run the commands only when you are logged in to the machine. There is another form of the who command that you can now try out.

[kumarr@linuxkumarr]S who am i

kumarr pts/O Oct 1308:55 (:0.0)

This time we have given the arguments am i to the actual command who. The result is similar to that obtained earlier, but now you are the only user listed. This command has the effect of telling you the login name of the user currently logged in at that terminal, the number of the terminal and the date the user logged in. Other users of the system at that time are not listed.

Arguments to the command are separated from the command by one or more spaces. It might seem silly to ask the computer who you are, but if the previous user has not terminated his session, you can find who it was by this command. But you would do well never to leave your terminal unattended while you are logged in, as it would be a security lapse. Actually Linux also provides the command:

[kumarr@linuxkumarr]$ who are you

kumarr pts/O Oct 13 08:56 (:0.0)

which is synonymous with who am i, but sounds much more intelligent.

You have now seen the general format of Linux commands, which comprises the
basic command followed by zero or more arguments. The command and the various arguments are separated by one or more spaces and the whole sequence is terminated by the new line character, which is produced when the ENTER key is pressed.

You can enter more than one command on the same line by separating the commands from one another with semicolons like this:

[kumarr@linuxkumarr]Sdate;who

Thu Oct 14 00:08:28 IST 2004 kumarr

ttyl Oct 13 23:51

kumarr pts/0 Oct 1323:59 (:0.0)

The commands are executed one after the other in the order they were specified on
the command line. After the last command is over you get the prompt again.

Arguments to commands should not contain spaces otherwise the different words of the argument would be interpreted as different arguments by the computer. If for some reason the argument needs to contain a space, you must enclose the argument in double quotes (") or in single quotes (‘).

Most arguments to commands are filenames (discussed later in this unit), options or expressions. All of these could occur in the same command. The exact order in which these arguments are listed can depend on the command and should be ascertained by examining the documentation for that command. Usually options immediately follow the command with the expressions and filenames coming next.

You will see details of such cases later when we study more complex commands than f the ones we have looked at so far.
If an argument itself contains quotes of one kind you can enclose it in quotes of the, other kind. Thus:

[kumarr@linux kumarr]$ grep -n "Ram Kumar's Salary" employee payroll

looks for the expression Ram Kumar's Salary in the files employee and payroll and prints the line numbers of the lines in which the expression is found.

Sometimes the bash shell places restrictions on the use of certain characters because it interprets them in some special way. To use these characters in arguments, you have to use quotes. The details of bash, the shell, are discussed in Unit 4.

3.1.5 Changing Your Password

You saw earlier that your password was the only way *of* preventing somebody from using your account *on* the system. Without it anybody who knew your login name could walkup to the machine and start using your account. This would be really serious in the case *of* the superuser or root.

When you are first given your account you are told what your password is. This is usually some default convention used by the system administrator, though not a good practice. It should be a different password generated for each new user, otherwise you are inviti1)g a security breach. You are then invited to change your password when you first log in. This can be done with the command:

[kumarr@linux kumarr]$ passwd_

Changing password for user kumarr.

Changing password for kumarr

(current) UNIX password:

password: Authentication token manipulation error

Notice that unlike the commands you saw so far, the password command is interactive. It asks you to enter some information rather than doing all the work by itself. The first thing it asks for is your current {or old) password. This is to make sure that somebody else cannot change your password while you have left your terminal unattended. If the wrong password is entered here, you get a message as shown above and you get back the prompt. You can then of course try again:

[kumar!:@linuxkumarr]$passwd

Changing password *for* user kumarr.

Changing password *for* kumarr

(current) UNIX password:

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

If you now enter the old password correctly you are asked to type in your new password. After that you are asked to enter it again. If you type two different things here, the system tells you that they do not match and asks you to try again. If you keep getting mismatches, the command terminates with the message passwd: Authentication information cannot be recovered

The number of retries allowed is configurable by the system administrator and is usually kept at 3. This is because if you are unable to change your password you are unlikely to be able to enter it correctly later to login. Also notice that none of the passwords is echoed. Your system will probably have restrictions *on* what passwords you can choose. The password should not be too short or too long to remember. You should change it periodically so that if someone has been using your account by laying hands *on* your password, they cannot continue to do so indefinitely.

If you are wondering how the passwords are stored *on* the machine such that even the superuser cannot find out what your password is, the answer is that Linux encrypts your password before storing it. This means that what is stored *on* the computer bears no resemblance at all to what you typed in as your password. When you try to login the next time, Linux again encrypts the password you type in and compares it with what has been stored. If the two are the same, you are allowed to login, otherwise your attempt is blocked. No ordinary user can even read the encrypted password-only the superuser can do so. So no one can find out what your actual password is -at least, not easily.

In Linux, you cannot change somebody else's password, even if you know what it is, from within your account. If you try to do so, you are told:

passwd: Only root can specify a user name.

How then can root change a user's password without already knowing it? Ah! When the user executing the passwd command is root or the superuser, Linux does not ask it to supply the old password. That is why root can set your password to anything without knowing what it is currently.

Since your password is the only way of protecting your account, you must take care to choose passwords well, that is, choose one which cannot be easily guessed. As a general rule, do not write down your password anywhere and let it be locked up in your 1'- head. Your Linux installation will probably enforce some rules on what you can set the password to be, or even the intervals at which you can change it.

Here is a short excerpt from the Linux manual entry for the passwd command about choosing passwords.

Don't write down your password -memorise it. In particular, don't write it down and leave it anywhere, and don't place it inan unencrypted file! Use unrelated passwords for systems controlled by different organisations. Don't give or share your password, in particular to someone claiming to be from computer support or a vendor. Don't let anyone watch you enter your Password. Don't enter your password to a computer you don't trust. Use the password for a limited time and change it periodically.

Your password should be hard to guess, and so you should not use information about yourself that is easily available. This includes your name that of your family or anything to do with your vehicle, credit card and so on. Also avoid dictionary words. It would be reasonably safe to use a combination of upper and lower case letters and special characters, with a length of at least 8 characters.

SELF ASSESSENT EXERCISE 2

1) Can you run the passwd command again and set your password to what it already is?
2) Can a friend (not the superuser) help if you have forgotten your password?
3) Find out any restriction in force at your installation on what you can set the password to be.
4) What does Linux have to say about choosing passwords? Find out from the documentation.

3.1.6 Characters with Special Meaning

Some characters are interpreted in a special way by the shell. These meanings will be discussed in detail in the next unit of this block, but that apart, there are certain characters you will find to be useful.

For example, suppose you start a command that takes a long time to execute and you change your mind and do not want to wait for the command to finish. You can abandon or break a command in between by pressing the BREAK character, which is set to ^C. Again, consider a command which produces a lot of screen output. This could happen if you were typing out a long file, for example. The output will probably be dumped on your terminal far too fast for you to read. To stop output on the screen temporarily, hit AS. You can press any other key to restart the output.

A social character that you can use to erase a command line that you have typed is ^U. This is useful when you just want to start over rather than correcting some small mistake.

Another special character can be used to terminate your login session. While you can do so using the exit command, you could also try the special character AD. This indicates to the shell that there will not be any more input from you. So Linux logs you out and again displays the login message on the console for another user, or you again, to start another login session. If you are in the X-Window mode and are in a graphical terminal window, the window is closed if you logout.

3.1.7 Linux Documentation

Linux comes with copious documentation that is mostly available on line. You should, as a user, learn how to use the Linux manuals and other resources. While we will not be able to discuss this topic in detail here, you will have to acquire this skill if you want to obtain a good understanding of Linux. This is because in this block we do not have -the space to consider any but the most basic commands, and even those only briefly. We will not even be able to consider all the options available with many of the commands that we do discuss. The only way for you to master them will be by consulting the documentation.

Usually the documentation will have been installed on your machine when Linux was set up. In that case you can look up the manual entry for a command by using the man command. For example, to learn more about the who command than what we have talked of, saying:

HYPERLlNK "mailto:kumarr@linux" kumarr@linux kumarr]$ man who

You can similarly learn more about the date, color any other command. So to learn more about the man command itself say:

[kumarr@linux kumarr]$ man man

For many commands you can use the info commandio get more complete information. There are several other resources available, for example you can look at the URL:

https://www.redhat.com/docs/

that has a host of documentation on Red Hat Linux. You can use a search engine to look for Linux documentation on the Internet, to find user groups and so on. These days it is less common to use printed manuals, primarily because they tend to get outdated so quickly! Besides user level documentation, there are many resources that you can find for system administrators and programmers, as you get to a more advanced level in Linux with practice and experience. So use this block as a quick introduction to get your feet wet and then move onto the more detailed documentation available.

SELF ASSESSMENT EXERCISE 3

Look up the manual entries for all the commands we have studied so far. What do you feel about the number of options available With each command?

3.2 The File System

In this section we will explore the file and directory structures of Linux. Just as a paper file is something into which you can put papers and bunch a group of papers together, a Linux file is something into which you can put data. A file has a name, and this name is the property of the file rather than the data present in it at any given time. It is Possible to change the data in a file. This act does not affect the name of the file. Thus Linux commands can be made to operate on the data in a file as a group.

A file usually exists on the hard disk(s) of the computer. This will be the case when you are logged in to the machine and are engaged in a session. The actual areas of the hard disk used by a file can change as the file is increased and decreased in size. As you will see later, the size

of a file in Linux has a precise technical meaning, and the size of a file does not necessarily tell you the actual amount of data in it.

Linux has three kinds of files -ordinary, directory and special. You have already got an idea of what ordinary files are. Special files will be discussed in Unit 5 of this block.

Directory files contain information about other files, including directories or special files. A directory groups its contents together hierarchically under itself, and a directory within a directory is called a subdirectory of the directory at the higher level, also called the parent directory. Thus a Linux file system is like an inverted tree of directories, starting at a root and going down to an arbitrary depth of hierarchically arranged levels.

We will now look at some of the files in Linux, learn how to navigate the file structure and how to make use of it.

3.2.1 Current Directory

Every user who is given an account on a Linux system is also given a directory where he reaches on logging in. This directory is also called the home directory. The current, working or current working directory is the directory in which you are currently located. On logging in, your current directory is normally your home directory" You can find out what your current directory is at any time by using the command:

```
[kumarr@linux kumarr]$ pwd_

/home/kumarr
```

This means that your current directory is called kumarr and is located under the directory horn': Which is in turn located under the root directory. Of course the actual home directory you are allotted will depend on the installation. By the way pwd is one of the few Linux commands that do not take any arguments or options.

The output that pwd displays is called the full pathname of your current working directory. This is also known as the complete or absolute pathname, that is, the pathname starting from root. You can refer to your directory by just saying kumarr. But this is not unambiguous as there can be another directory called kumarr under some other directory as well. But no two directories or files on the same Linux machine call have the same complete or full pathname. The various components of the path are separated from one another by slashes (' / ').

We have not yet talked of what a valid filename can be. Actually in Linux there are no restrictions and a filename can have any characters and any length. The same rules apply to directories as well. In practice it is best to avoid certain characters in filenames because they have special meaning10 the shell.

3.2.2 Looking at the Directory Contents

We will now see how to look at the contents of a directory. The command is:

{HYPERLINK "mailto:kumarr@linux"kumarr@linux kumarr]$ 1s

This gives you a listing of all files in the current directory. If you have just been allotted your account and are logging in for the first time, you will be in your home directory and that directory will be empty, that is, there will be no files in it. Over the years the ls command has accumulated a lot of options and it takes some time and experimentation to understand them all. The first option we look at is:

[kumarr@linux kumarr]$ls -a

Here a means all. This is your first taste of Linux options, so look at the command carefully. The command 1s is followed by at least one space after which the hyphen or minus sign introduces the option letter. The -a option tells Linux to list all files including those that are "hidden". Hidden files are those that start with the '.' character. Unless the -a option is used, Is never lists such files in its output. The output of 1s is always sorted in some order, the default order being alphabetical. This sort order can be altered by other options to Is which we will take up later. This is why the file (actually a directory) '.' is listed before '..' in the output.

.	.emacs	.gtkrc-I.2-gnome2	.netscape6
..	.esd_auth	.ICEauthority	.openoffice
abc	_files	ignou	.qt
abc\d	.fonts.cache-1	.kde	.recently-used
abc d e f	.gconf	.mailcap	.rhn-applet.conf
.bash_history	.gconfd	.mcop	.sversionrc
.bash_logout	.gnome	.metacity	.user60.rdb
.bash yrofile	.gnome2	.mime.types	.Xauthority
.bashrc		.gnome_private	.rnozilla
.chromium		.gnome-desktop	.nautilus
.chromium-score		.gtkrc	.netscape

You will usually find that directories are listed in a different colour such as blue. This helps you locate them quickly. The '.' refers to the current directory and '..' to its parent. These are pronounced dot and dot dot respectively. In-this case '.' refers to / home/kumarr and '..' to /home. The directory 'j' or root is its own parent. This output is of course not very interesting because your home is devoid of files created by you and you do not yet know how to create any. The only files you see are hidden files made by the software itself.

To get around this, let us look at some other directory. You can get the listing of any directory by supplying its name as an argument to 1s. Thus to look at the directory listing of the root directory, use the command:

[kumarr@linuxkumarr]$ls /

bin dev home lib misc opt root tfipboot usr

boot etc ini1rd lost+found mnt proc sbin tmp var

Here the output is sorted column wise. We must caution you that it is quite likely that you will see a different listing than the one shown here. It is self evident that the listing will depend completely on the machine you are working on. However there are some files that will surely exist on the root directory of a working installation. The above directories are such files. To sort the output according to rows, from left to right, say:

[kumarr@linux kumarr]$ls -x /_

As you have seen the Is command lists several columns in its output. This is easy to change. If you want I column in the output, say:

[kumarr@Iinux kumarr]$ls-1

bin_
boot
dev
etc
...

Another variation of the command is the -C option that might produce the same output as the command without an option. If that happens it means that the actual command has been configured to use the -C option by default.

Besides using the colour of the file to identify directories, you can use the -p option or the -F option to append a '/' to every file that is a

directory. The -F option also appends a ...' to every file that is an executable file, that is, a command. Check this out and understand how the -F and -p options differ.

[kumarr@linuxkumarr]$ls -Cp /

bin/ dev/ home/ lib/ misc/ opt/ root/ tfipboot/ usr/

boot/ etc/ initrd/ lost+found/ mnt/ proc/ sbin/ tmp/ var/

If you repeat this command on the bin directory; you will see something different.

[kumarr@linux kumarr]$ls -Cp /bin

arch df hostname nice su

ash dmesg igawk nisdomainname@ sync
ash.static dnsdomainname@ ipcalc pgawk tar
aumix-minimal doexec kbd_mode ping tcsh

You see some files in a different colour. These are executable files, or commands that you can run. You can use the -F option to verify that they are so because each of them will have a * appended to the name in the listing.

When you have a very large directory, you might want to use an option of is that gives a very compact listing.

[kumarr@linuxkumarr]$ls-m/bin

arch, ash, ash.static, au!\llx-minimal, awk, basename, bash, bash2, bsh, cat, chgrp, chmod, chown, cp, cpio, csh, cut, date, dd, df, dmesg, dnsdomainname

The colours of the entries are as usual but each entry is separated from the other by a comma.

The above examples show you that the root directory consists of both directories and ordinary files. In Linux everything is considered to be a file. The directories here, or anywhere else, can themselves contain other directories to any depth. To see the contents of /home, you can say:

[kumarr@linux kumarr] $ Is -xp /home

kumarr/khanz/

You might surmise that you are seeing the home directories of users who have accounts on the machine. You also see your own home

directory here. You can tell that they are directories by the colour as well as the t that is appended to the name. But there is one thing that you need to note. When you had logged in and checked your current directory, the result had been:

[kumarr@linuxkumarr]$pwd_

/home/kumarr

which is different from what you see here. Why is this so? We have seen in the last section that the pwd command tells us the full, absolute pathname of the current working directory. When we look at the contents of /home, kumarr is merely one of the directories under it and is shown as such. To get the complete pathname we must specify the preceding portion which is /home. Thus the full or complete pathname is t home/kumarr.

If you look at the directory listing of /usr, you will find a bin directory under it too. By now you will have understood that this bin directory is different from the one you saw under the root directory. The first has the full pathname /usr/bin, while the second has the full pathname /bin. You should now look at the contents of the other directories and try specifying their complete pathnames, You can also try 10Qking:llt their contents by specifying relative path names. We will look at complete and relative pathnames again in the next section. You would do well to understand pathnames, relative and absolute, thoroughly as that will be useful in navigating around the directory tree.

But let us now get back to our friend the 1s command. One of the most useful and often used options is -1, which gives the so called long listing of the directories asked for.

[kuma!:r@linuxkumarr]$ Is-I

total 24

-rw-rw-r-	I	kumarr	kumarr 4 Oct 1501:18 abc
-rw-rw-r-	I	kumarr	kumarr 5 Oct 1501:17 abc\d
-rw-rw-r-	I	kumarr	kumarr 5 Oct 1501:17 abc d e *f*
drwxr-xr-x	2	*kumarr*	*kumarr* 4096 Aug 21 20:33 _files
drwxrwxr-x	3	kumarr	kumarr 4096 Oct 918:01 ignou
-rw-rw-r-	I	kumarr	kumarr 27 Oct 11 22:51 xx

Now this is a complicated looking output, so let us try and understand the meaning of this listing. The first column of the output tells you whether the file is a directory or not. A '-' means that it is an ordinary file whi1e'a directory has a'd' in that position. So you now know

another way of telling whether a file is a directory, apart from the-p or -F options and the colour of the listing that you have already looked at. The other 9 columns in the first field tell you about the permissions on that file. We will look at these in detail in a later section.

The next field in the output is a number indicating the number of links to the file. For a file this shows the number of names it has. In Linux the same physical data may have several names, although it must have at least one. Eacl1 name is a link to the file. Usually ordinary files have only one link, but if there are more it does not mean that there are that many copies of the data in the file. There is only one physical copy of the data that can be referenced using any of its names. In the case of directories the number of links tells you the number of subdirectories that it has.

The third field of the output shows the owner of the file. Root and bin are names reserved by Linux for its use as we have seen earlier. In some cases you might see a number like 207 instead of the user name.

The next field is the group name and in certain situations can be a number in the display. The user is a part of the group shown here.

The fifth field if the sire of the file in bytes. You already know that the size of a file in Linux has a precise meaning which is unrelated to the amount of data in it. However, do not be alarmed because in most cases the intuitive meaning of sire does hold good and the figures you see usually do represent the number of bytes of data in the file in question. For large files the sire is difficult to comprehend when represented in bytes. For such cases, you can use the -h option that gives the sire with a K, M or G following the number for kilo, mega and giga bytes respectively. These are true kB, MB or GB as the multiplier used is 1024. Often we colloquially use K to mean] 000 times. If you want the sire represented with a multiplier of 1000, use the -si option. The sire is then shown as kB, MB or GB after the number to help you distinguish which multiplier has been used.

The next item of information is the date the file was last modified, and in the end the name of the file is shown.

With tile long listing of Is, you can find out many useful things about the file. Try looking at the directory long listing of the various system and other directories on your machine.]n the course of this exploration, when you look at directories like /bin, Is bin and /usr/bin, you will see some familiar names such as who, pwd or Is. Thus 1s is itself to be found in the /bin directory. The three directories we just mentioned are the ones where most of the binaries or executables of the system

commands are to be found. There are some to be found under /etc. as well.

We will now look briefly at three other options to the 1s command. When a directory is given as an argument to 1s you get to see the contents of the directory. But suppose you want to check the permissions on a directory, say /home/kumarr. If you try:

[kumarr@linuxkumarr]$ls -I /home/kumarr_

you will see nothing of what you need because Is tries to list the contents of the directory and at present there is no file in your home directory. In the examples above, we had created some files in the home directory of kumarr so as to be able to show you some output, and these files were then deleted. So to see the permissions you could say:

[kumarr@linuxkumarr]$ls-l/home

whereupon kumarr would be one of the entries. But this is awkward as you will have to wade through potentially several entries before you can locate the one you are interested in. The answer to this is:

[kumarriWlinuxkumarr]$ls -Id /home/kumarr_

drwx- 19 kumarr kumarr 4096 Oct 16]6:48 /home/kumarr

which lists /home/kumarr as a directory and shows all the information about it.

By now you would have also realised that subdirectories are normally shown as single entries and any files inside them are not shown. To look at the contents of a directory and recursively of all subdirectories within it, use -R.

[kumarr@linux kumarr]$ls -IR/usr

will show the contents of /usr and also recursively of every subdirectory inside it, down to ordinary files. Thus using:

[kumarr@linux kumarr]$Is -IR /

you can see each and every file and directory on your system, though not hidden files.

Another option you might need sometimes is the -r option, for reverse. This reverses the sort order of files displayed by Is. You can try this with any option, such as:

[kumarr@linux kumarr]$ls -1Rr /

So far you have only given directories as arguments to Is, but you can use ordinary files as well. It then lists only that file if it exists. Moreover you can give all Y number of files 0; directories as arguments to Is and it will list whichever ones exist. You can also use wild cards here. The '?' character matches any single character, while the '*' matches any number of characters except a leading '.'.

If you feel out of breath after looking at these options, there are many more we have not looked at! You are encouraged to look up the documentation for 1s and experiment with them. Many Linux commands have zillions of options -getting used to them all requires time and effort. But you will find that you soon get to know the options you use often. It is probably best, when learning a new comman4, to concentrate on a few useful looking options only. As you use them frequently, you will get to know them well. Then you can spend some time deepening your knowledge of the command by trying out the other options.

Most beginners get overwhelmed by the large number of options and do not know where to start or when to stop. You will have to work out a method that suits you. Maybe you are the type who likes to learn everything about a command at one go. But many people, including the author, find that building on a solid foundation of already known options is easiest.

SELF ASSESSMENT EXERCISE 5

1) Read up on and try out the other options to Is. What is the output of Is -1m? Of Is -ml? Which option takes precedence? What is the result of Is -d?
2) If your system has the -x or -C option set by default, how can you get the standard Is listing?
3) How can you control whether you see different file types in different colours?
4) Find out how to sort the output on time rather than alphabetically.
5) Since you can have filenames that are of arbitrary length, would you prefer to store information in the filename or in the file?

3.2.3 Absolute and Relative Pathnames

You saw in the last section how pathnames could be relative or absolute. Since the Linux file system is logically structured like an inverted tree, it is important to understand how to specify pathnames. Both methods can be used and in Linux it does not matter which approach you use in identifying the file you mean, as long as you are careful about specifying it correctly. However there are situations where one or the other approach is more convenient. So you should take the trouble to assimilate the concept and learn how to navigate around the system with felicity. Let us look at a typical directory hierarchy on a Linux machine.

At the top is the root directory, under which are directories such as bin, boot, dev, home, lib, opt, home, etc, usr and so on. Under /dev you would have some 18 directories besides the ordinary files. Similarly /etc might have 63 odd directories under it and /home would have the home directories of different users. /usr could have about 13 directories that include bin, etc. and others.

The exact layout of the directory hierarchy on your machine is likely to be different. We will soon be looking at some of the important directories and files on a Linux system. For the moment, though, you would do well to just concentrate on learning how to move around. You already understand what is meant by the current directory. This is the directory in which you are located at any given time. If you issue the command Is, it is1he files in the current directory that are brought up for you to see. If you login as kumarr, you will probably end up in /home/kumarr when you get your prompt unless things have been arranged otherwise.

Now suppose /home/kumarr has a directory called nip that has a file called augcfg.C. Suppose you want to seethe size of this file alone. For this you need to use the 1s command and provide the filename as an argument to it. .In Linux you can provide a path name (relative or absolute) as an argument to a command wherever you could otherwise provide a bare filename. So you now have three ways of accomplishing what you want (we will assume that you have the required permissions-this will, in fact, be the usual situation) to do.

Let us first use an absolute path name. So you have to specify the filename starting from root or '/'. Thus your command needs to be:

[kumarr@linux kumarr]$ls -I /home/kumarr/nlp/augcfg.C_

You have already used this method in the last section. The second way is to use a relative path name, where you specify the pathname relative

to where we are currently: Here you only need to recall that '..' stands for the parent directory of the current directory. So if you are at /home/khanz, you can say:

[kumarr@linuxkumarr]$ls -1../kumarr/nlp/augcfg.C

The '..' takes you one level up, that is, to /home. From there you continue naming the file as before. Of course, you could have used the following rather convoluted way:

[kumarr@linuxkumarr]$ls-1 ../../home/kumarr/nlp/augcfg.C

This is inefficient because you implicitly to root before naming the file. The first '..' takes you to /home and the second one level higher, to / or root itself. Then you begin your descent until you reach the file you desire. Here it would have been better to use an absolute path name instead of this, for then you would not have had to use two steps to reach root.

Usually a filename is specified by the method that results in the shortest possible specification of the name, though of course that is not at all necessary. This depends on whether the filename is closer to you or to the root directory. If you are located in *I* home/khanz and want to specify a file in the directory /home/kumarr, it is easier to say. /kumarr rather than /home/kumarr.

There is a third way of looking at the size ofaugcfg.C. For this you will have to learn a new command, cd, which lets you change the current directory. This command can be given an argument which is your intended destination and it then changes your directory to what you asked, provided you have the appropriate permissions. And how do you specify your desired destination directory? By specifying, the pathname, of course. The pathname can be specified, as you would have undoubtedly guessed now, either as a relative or absolute pathname. So you can say from /home/khanz

[kumarr@linuxkumarr]$cd/home/kumarr/nlp _

or

[kumarr@linuxkumarr]$cd ../kumarr/nlp

and then look at the size by:

[kumarr@linux kumarr]$ls -I augcfg.C

This really amounts to specifying the filename relative to /home/kumarr/ nlp, the current directory. In general when you specify a bare filename you are specifying the filename relative to the current working directory. So the command above is really a shorter way of saying:

[kumarr@linuxkumarr]$ls ./augcfg.C

One form of the cd command can be very convenient if you have wandered far off your home directory and want to return there, especially if your home directory happens to far away from the root directory. This is:

[kumarr@linuxkumarr]$cd

without any arguments. It always brings you back to your home directory irrespective of where you are, even if you were already there to start with.

SELF ASSESSMENT EXERCISE 6

1) Go to the root directory and then try to go to its parent with 'cd ...'. What happens? What do you conclude?
2) What happens if you try to cd to a non-existent directory?
3) Can there be a file under a directory with the same name? Why?

3.2.4 Some Linux Directories and Files

It will be interesting and useful to now get acquainted with the Linux system directory structure. We will look at the layout and contents of the Linux system directories and understand how the various system files are grouped under directories. We will also learn about the functions of some of the system files. The typical Linux directory structure is as described in the earlier section.

We again emphasize that only some of the system directories are shown here. Your machine could have a somewhat different organization. How will you find out the directory tree for your Linuxsystem? You can do this by exploring the files on your machine.

As you have already seen, the I bin directory contains some of the Linux system commands and utilities. These include some of the commands that you have learnt so far, such as Is and pwd. You can look at the long listing of this directory and note the information provided. Look at the sizes to get an idea of the size of executables on your machine. These will depend, among other things, on the architecture of your computer.

The /dev directory contains device special files concerned with hardware devices like printers, mice, audio devices, storage devices such as floppy drives and CD-ROM drives and so on. You will learn more about these files and the /dev directory later in this block.

The /etc directory, as the name suggests, has several miscellaneous files and directories. It contains many files and commands that are reserved for the use of the system administrator. Many of the system defaults are set up using these files. Ordinary users cannot execute these commands or use these files. For example, look at /etc/issue or /etc/motd. The first contains the text that is displayed at your login prompt before you have logged in, while the second file (for message of the day) contains the text that you see just after you login. These files can be blank, and sometimes are. The file /etc/group has the names and group numbers of all the groups in the installation. The /etc/passwd file contains the login name of each user, his user identification number, his home directory, his default shell and sometimes some commentary. In Linux the encrypted password of each user is stored in a separate file called /etc/shadow that cannot be read by ordinary users. So you cannot see even another user's encrypted password.

The /lib directory contains system libraries that are used with compilers and shared libraries that are needed at run time for executing commands and running executables.

The /sbin directory contains some standalone commands and utilities used during installation. These are of interest to system administrators and those who need to install and maintain the system.

The /tmp directory contains temporary work files that might be created by utilities and commands when they run. It provides work space to such commands. This directory is cleared out periodically on many installations. In any case you should assume that any file in the /trnp directory can be erased without warning. So you should not store any useful files here and put them under your home directory only.

The directory /usr/bin contains useful and important commands and utilities for users. There is no sharp distinction between the commands in /bin and here, though. Our old friend, the cat command, is to be found here. /usr/include contains header files that are useful in writing C or C++ programs.

An interesting directory is /usr/games that traditionally contained text games in older Unix installations. Today it could contain some more sophisticated games, such as chromium or maelstrom.

The directory /usr/local/bin is often used as a repository of commands used locally and frequently developed by local talent. Such commands are often those that are found useful and convenient in that installation.

The /mnt directory is used to mount different devices such as cdroms to make them part of the directory tree.

While looking at these directories, you might have noticed that files under /usr/include often end in ".h" and that there are many files ending in ".so" in /lib. Although Linux places no restrictions on the characters possible in filenames, there are some conventions followed in some cases. Such files are often referred to as '.h' files, '.so' files and so on, or sometimes simply as h or so files. The Linux commands or utilities might enforce restrictions on the names of these files although Linux itself does not do so. Thus C program files end in ".c", C++ program files end in ".C", assembler source files end in ".s", and so on.

There is a useful command, file, to determine the type of a file. This takes any number of files as arguments and tries to determine the type of each. Although it is not foolproof and is open to deceit, it usually does a good job.

SELF ASSESSMENT EXERCISE 7

1) Look up the Linux documentation for the various utilities above and find out which of them enforce file naming conventions.
2) Run the file command on various kinds of files from the various directories you have seen and see if their types are reported correctly.

4.0 CONCLUSION

This unit has taken you through how to start working on Linux machine. The unit is oriented towards familiarization with the richness of the system. It introduced to concepts like hierarchical directory structure how to change your account password and how to close a login session, etc.

5.0 SUMMARY

In this unit we have started at the beginning and looked at many basic Linux commands. However, there are many useful commands that we have not been able to examine. You will need to refer to the documentation and learn these. By now you would know enough about Linux to conduct a session with ease. In the units to follow we will examine some more commands and look at some utilities in slight detail

6.0 TUTOR-MARKED ASSIGNMENT

1) What happens if you type in your login name all in upper case?
2) Try logging in by using a friend's account (do not ask him for the password). Are you able to get the prompt?
3) Try logging in using an account which does not exist on your system (confirm this from your system administrator). Is there any difference in the computer's response from that in the last exercise? Why do you think this is so?
4) Try using your mouse just after logging in at the console when you are in text mode. What happens? Are you able to perform any operation with the mouse?
5) What are the characters you need to use to correct typing mistakes while logging in? How do you correct typing mistakes while entering the password?
6) How many previous commands can you invoke by using the arrow keys?
7) How will you enter a '\' in the command that you invoke?
8) How can you get the calendar for some other month in some other year?
9) Get the calendar for the 1752 and look at it. It anything the matter?
10) Find out how to set the system date. Why do you think only the upper user it allowed to do this?
11) Study the ‘who’ command and use it to find the date the machine was started up,
12) Open p terminal windows and applications on all four desktops and use the ‘who’ command. What do you see? Are you considered to be the same user on all desktops?
13) The commands you have learnt so far produce only a small amount of screen output. How will you produce output which does not fit into one screen, using only the commands you have learnt so far?
14) How much control does the AS command allow you over the output? Can you read a long file in sequence easily by this method?
15) What would happen if your home directory did not exist and you tried to login?
16) How would you put a space into a filename?

7.0 REFERENCES/FURTHER READINGS

There are a host of resources available for further reading on the subject of Red Hat Linux version 9.0.

http://www.redhatcom/docs/manuals/linux

htt://w.linux.org gives among other information, a list of good books on Red Hat Lihux.

Consider joining a good linux mailing list, e.g.

UNIT 3 LINUX UTILITIES AND EDITOR

CONTENTS

1.0 INTRODUCTION

In this unit, you will start to delve deeper into Linux and learn some more useful commands. You will also see how to combine commands together to perform useful tasks for which there might not be any single command available. You will learn to write simple programs in the bash shell that will let you write your own Linux commands. You will also see how to use the graphical user interface of Linux to perform many tasks without issuing any commands on the command line. Finally, you will look at the editor, vi, available in Linux for you to edit text files.

Because of the large amount of material to be covered, we will have to be brief. However, we shall try to illustrate important concepts with realistic examples. You will then need to practice whatever you learn on a Linux computer so that you understand the variations and nuances of the commands.

2.0 OBJECTIVES

After studying this unit, you should be able to:

use some simple and important Linux commands
understand the concept of standard input, standard output and standard error
be able to use filters and pipes to connect commands together
write simple shell scripts to produce your own commands
use the graphical user interface to perform tasks without needing the command line
use the programmer's editor iv.

3.0 MAIN CONTENT

3.1 Some Useful Commands

In the last unit, you have seen how to use some basic Linux commands like 1s, cd and pwd. We will now look at some more useful commands that are commonly required. Linux has a large number of commands some of which are useful for system administrators, 5,9IDe for software developers and so on. Here we will consider only general-purpose commands that any category of user needs. Also remember that we will look at only a few most commonly used options of the commands. For a full descripti6nyou should refer to the documentation for the command.

File Manipulation Commands

cat

You have so far learnt how to see directory listings that tell you the contents of a directory in Linux. Usually there will be several files and perhaps some other directories listed. But how do you see the contents of a file? This is done with the cat command.

[kumarr@linuxkumarr] cat first_file

prints out the contents of first_file on the screen.

Remember that Linux is not concerned with what kind of file first_file is. If it is
an executable file or a file produced by using some editor that does formatting, then the output will most likely not be intelligible to a human reader. In any case, the file will probably be printed out so fast that you will not be able to see anything but the last screenful. So it is actually useful only if you want to see what is there in a small text file. There are other forms of the command that you will study a bit later in this unit.

For now, try the following command

[kumarr@linuxkumarr] cat first_file second_file

You will find that the contents of both the files are printed on the screen one after the other without any kind of gap in between. Actually cat stands for concatenate, that is, join the files together. You can give any number of arguments to the cat command and it will print them out on the screen in the sequence you have specified. You will see later how to use this facility to advantage.

tail

Sometimes you might only be interested in something at the end of a text file. You could use ca t and see what lies at the end, but if the file is a big one, you could be waiting for quite a while before the gobbledygook on the screen stops and you can make sense of what you see. For such a situation, you can use the tail command. It shows what is in the last part of a file.

[kumarr@linux kumarr] tail first_file

This prints out the last 10 lines of first file. What if you need to see something that is in the 14"' line from the end? You can say

[kumarr@linuxkumarr] tail -14 first file

You can use any other number instead of 14 that you need. So to see just the last three lines, say

[kumarr@linuxkumarr] tail -3 first file

Like cat, you can give multiple file names as arguments; the last part of each is printed out with a header showing the name of the file that follows. Some other useful options are +n to start printing from the nth line of the file instead of the beginning, unlike cat. You can also use -cn to print the last few bytes instead of lines, with n being the number of bytes you want to see.

cmp

This command lets you compare two files and determine whether their contents are the same or different. If they are the same, the command says nothing. Otherwise it prints out the first byte for both files where there is a difference between the two. There is also the -I option to print out all differences.

[kumarr@linux kumarr] crop first_file second_file

Remember that if one file is a part of the other, they are still considered to be different files.

diff

This command compares two files line by line and reports the differences between them. Unlike the mechanical comparison of the cmp command, diff makes a much more intelligent comparison. It

indicates the differences between the files in three ways a, d and c. These stand for lines which have been added, deleted and changed between the two files. The symbol "<,'refers to the first file and ',>" to the second file.

kumarr@linux kumarr] diff first file second file

There is also the -b option to diff that ignores white space, and the -B option that ignores blank lines.

wc

This command counts the number of characters, words and lines in a text file. You can give the command any number of arguments, whereupon it performs the count for each file and gives the total on the last line. It takes the -c option to report only characters, the -w option to report only words and the -I option to report only lines.

[kumarr@linuxkumarr] wc first_file

You can also combine options together, such as -cl to report characters as well as lines.

sort

This is a useful command that lets you print out the contents of a file in sorted fashion. You can choose the delimiter that separates fields and the default is the space character. It produces output on the screen by default but you can place the output into a file using the -f option.

[kumarr@linuxkumarr] sort unsorted_file

There are several options to sort on fields, to use numeric order, to choose the collating -sequence and so on.

There are many other useful commands that manipulate text files. The tr command can be used to translate or change characters in a file. The split command breaks up large text files into a number of files with a fixed number of lines each. The cut command can be used to produce vertical sections of a text file. To get formatted output on the screen, rather than the plain dump of the file that cat gives; you can use the pr command.

3.2 Permission Modes and Standard ES

We will now see what file permissions are and how to change the permission modes of files and directories. So far we have had occasion to look at various commands, many of which have to do with files of various types. The permissions of files can make a great deal of difference to the way the commands behave. What are these?

You have already seen in Unit 2 that the long form of the 1s command, the one with the-I option, tells you about the permissions of the file.

[kumarr@linux kumarr] Is -1

total 32

-rw-rw-r-	1 kumarr	users	4 Oct 15 01:18	abc
-rw-rw-r-	1 kumarr	users	4 Oct 16 17:31	abcd
-rw-rw-r-	1 kumarr	users	5 Oct 15 01:17	abc\d
-.rw-'rw-r-	1 kumarr	users	5 Oct 15 01: 17	abc d e f
drwxr-xr-x	2 kumarr	users	4096 Aug 21.20:33	_file
drwxrwxr-x	3 kumarr	users	4096 Oct 9 18:01	ignou
-rw-rw-r-	1 kumarr	users	27 Oct 11 22:51	xx.
-rw-rw-r-	1 kumarr	users	75 Nov 2 22:1	xx.c

The first column of the first field has a "d" in it if the file is a directory. So ignou and _fi1es are both directories. For ordinary files, the first column of the first field is a hyphen. The next 9 columns specify the file permissions. The user community in Linux is divided into three categories -owner, group and others. The owner is tile person who first creates the file. Several users at an installation can be made part of a user group. Such a facility is useful in keeping working on the same project or department categorised together. All group members form the second category of users. Finally, the rest of the community is lumped under others, which are users who are neither the owner nor part of the same group. ln the example shown above, the owner of the files and directories is kumarr and he belongs to the group users. Other users might also belong to the same group, though it is quite possible for kumarr to be the only one who is part of the group.

Having studied this characterisation of Linux users, you can now begin to understand the permission modes. Every file has three possible modes of access -read, write and execute, represented in the directory listing by r, w and x respectively. A file to which you have read access can be read by you, which means you can look at its contents by using any method like the cat command. If you have write access to a file, you can alter its contents. Execute permission is relevant in the case of executable files

like those that we have seen in/bin, or for shell scripts that we will study later in this unit. You can run or execute a file only if you have executed permission on it. Execute permission does not mean anything in the case of text files, nor actually for any file that is not executable, except for directories where this permission has another meaning.

Now you know enough to understand the permission information. The nine columns are divided into three parts -owner, group and others -of three columns each. The permissions are specified in the order read, write and execute. If permission is available, the corresponding letter is shown, while the absence of permission is indicated by a hyphen. So rwx means the category concerned can read tile file, can write to it or alter its contents and can execute the program which the file contains. Similarly the permission r-x means that tile file can be read or executed but not altered or written to, because write permission is absent, r- means the file can only be read, and -x means it can only b~ executed. -- means that there are no permissions available and the file cannot be accessed at all. However, you can still see an ordinary file like this listed in a directory listing taken in the usual way. So you see that the permission columns in the listing shown above mean that for directories the owner has read, write and execute permission, whereas other members of his group have read and execute permission but no write permission. Similarly others, that is, users who are not part of the owner's group, also have read and execute but no write permission. Thus if you want to have all the permissions on a file, while denying group members write permission and allowing others only execute permission, the permission modes should be rwxr-xr-x.

With this knowledge, you should again look at tile Linux system files and study the owners and permission bits for each. You will find that all users have executed permission on the files containing system commands. This is obviously necessary otherwise you could not use those commands. If possible, ask your system administrator to remove your execute permission on the 1s command for a short while and then try to see your directory listing. If you are at a large installation where this kind of thing might not be possible, you can either wait until you know more about Linux to be able to see the effects of the absence of permissions, or ask somebody to make a copy of a system command in your directory, remove execute permissions on it and try to run it from your directory. Here you must take care not to run the system version of that command.

So far we have talked only about ordinary files and what file permissions mean in their case. But directories are also files, as you have seen earlier. Do the permissions all have the same meaning where

directories are concerned? If so, what is it like to execute a directory? Let us delve a bit into this and find out some answers.
First of all you must understand that directories are special kinds of files, which contain information about other files, the permission bits have somewhat different meanings than what a hasty guess would suggest. Before we even look at what read permission for a "directory means, try running the cat command on a directory and see what output you get. Linux will report an error and tell you that you were trying to cat a directory.

[kumarr@linuxkumarr] cat ignou

cat: ignou: Is a directory

Actually a directory contains information on the files it contains, such as their names. Knowing this, it should be easy to deduce what read permission for a directory could mean. If you can read a directory you can see what files are in it, which means you can do an Is on the directory. In the absence of read permission you will not be able to look at the directory listing for that directory.

What about write permission? If you have write permission on a file you can change its contents. In the same way, having write permission on a directory allows you to change its contents. What does that mean? Creating, renaming or removing files would mean altering the contents of that directory. So it follows that having write permission on a directory enables you to create, rename or delete files in that directory.

Beginners often find it a bit difficult to grasp this point though it is not really hard to underst3l1d. You should remember that having write permission on an ordinary file allows you to change the contents of the file but does not allow you to delete or rename the file. That is possible only if you have write permission on the directory containing the file. Later we will see how this could lead to a security lapse in some situations.

Lastly, coming to execute permission you would agree that there is no way you could execute a directory corresponding to the normal sense that the operation refers to for an ordinary file. For a directory this permission bit determines whether you can cd to the directory or can copy files from that directory (this only if you have read permission on the directory as well). This permission is often called search permission. To be able to cd to any directory you must have search permission on every component of the absolute pathname of the directory. If search permission is absent on any component, all files and directories on that component and below it become inaccessible.

Apart from these permissions there are some other permission modes that you will come across. We will take these up in the unit on system administration. For now it will be sufficient to know that some other permission modes exist so that you do not get taken aback if you find characters other than r, w or x in the permission modes of a directory listing.

Changing Permission Modes

We will now see how to change the permission modes of files and directories. The action of a command can differ greatly depending on permissions. For example, the ca t command will usually type a file on the terminal but will refuse to do so if the user does not have permission to read the file.

How do we change the permission modes of a file? The command to do so is chmod. It can be used only by the user who created the file or by the superuser. The owner is the user who created the file first, by any means such as by using a text editor or by copying an existing file.

Note that having all permissions on a file does not amount to owning it. If you can read somebody else's file, it does not mean that you can prevent others from doing the same, but you can establish such protection for a file of your own. Also do not get confused between the original file and a copy you might have created, having only read access to the source file. You can change your copy in any fashion you wish, but you will not be able to alter the original.

There are two forms in which you can use the chmod command. Let us look at the absolute method first, as it is slightly easier to understand and use. In this the permission mode desired for the files is given to the command in an octal notation that we will explain shortly. The mode of the file then gets changed to what was asked irrespective of the permissions before the command was run. The form of the command is thus

[kumarr@linux kumarr] chmode mode file

where filename is a list of one or more files whose permission modes are to be set to mode. If the permission bits are mode to start with there is no effect on the file or files after running the chmod command.

You know that file permissions are specified by 9 columns, for example rwxr-xr-x or rw-r-r-. In the absolute method the presence of permission is indicated by a 1 and its absence by a 0. The resulting 9 bit binary

number is then converted to octal. This octal number is what has to be specified as tile mode for the chmod command.

You know that a binary number can be easily converted to octal by making groups of 3 bits starting from the right. Now convert each group into octal as if it were a single number. The resulting string of octal digits is the number in octal. Thus
rwxr-xr-x

can be written in binary as 111101101 after replacing each permission by a I and each hyphen by a O. This binary number can be written as 111 101 101 after grouping the bits in threes. The octal form of the number is thus 751. So to convert a file to this mode say

[kumarr@linux kumarr] chmod 755 progfile

This will give progfile the permissions rwxr-xr-x. Likewise rw-r-r- in octal is 644. So you can provide these permissions to your file motd by saying

[kumarr@linux kumarr] chmod 644 motd

Instead of one file you can set the permissions on several files at the same time (all to the same value) by listing the files after the mode. So

[kumarr@linux kumarr] chmod 600 motd passwd

will make their permissions rw----. Thus you can read these files or change them whereas nobody else (except root) can even read them. So

[kumarr@linux kumarr] chmod 0 passwd

will mean nobody has any permissions on the file passwd and even you will not be able to read a file of your own with such a set of permissions. However, you can change the permissions any time since you own your file. Also, the super user can change the permissions of any file.

Let us now look at the symbolic method of telling chmod the mode. Here the permission types are, as always, r, wand x. In addition, there is a set of characters which specify the target of the actions. The target can be u (users), g (group), 0 (others) or a (all of these). The actions are + to add a permission, = to set it absolutely and -to remove a permission. So you can say (there must be no spaces in the mode argument)

[kumarr@linux kumarr] chmod a+x progfile

to allow everybody to run progfile, irrespective of the earlier execute permissions on it. However, this is different from saying

[kumarr@linux kumarr] chmod 111 progfile

because this would remove read or write permission for everybody, whereas in the earlier case, those permissions would have been left untouched. If the owner had read, write and execute permission, he would retain it. If the group earlier had read and execute permission it would continue to have that privilege. If others had no permission, they would acquire execute permission. One can remove read and write permissions for others by saying

[kumarr@linux kumarr] chmod o-rw progfile

One can specify absolute permissions by saying

[kumarr@linux kumarr] chmod u=rwx,g=rx, o=x progfile

Here the different target categories are given different permissions on progfile by separating them with commas. No spaces should be present in the argument, otherwise only the first part will be taken as the desired mode. The portion after the space will be treated as a filename, which has to be assigned those permissions.

You might find the numerical way easier to use. However, if you do not want to alter some permission bits, there is no straightforward alternative to using the symbolic mode. For example, if you want to deny others any permission on a file but do not want to alter your own or your group's permissions, you can say

[kumarr@linux kumarr] chmod o-rwx progfile

But to achieve the same result using the absolute method you would have to first determines the existing permissions. Suppose the value is 644. You will now have to say

[kumarr@linux kumarr] chmod 640 progfile

If the initial permissions were 666, you **need** to say

[kumarr@linux kumarr] chmod 660 progfile

Instead. If using the symbolic method, you would not need to worry. The command you need to give remains the same in both cases since the 0 action leaves the u and g permissions intact.

3.3 Pipes, Filters and Redirection

By now you have seen quite a few Linux commands, and you must have observed that many commands produce or can produce output on the terminal screen. Likewise many commands can take input from the keyboard. Actually, these commands have been written to accept input from a standard input file and to produce output in a standard output file. Usually these files are set to the keyboard and the terminal screen respectively. Let us look at this in somewhat more detail by studying some examples.

Standard Output

If you make a list of commands you have learnt so far you will find that many of them produce some output. For instance let us say

[kumarr@linuxkumarr] ca1

which prints the calendar for the current month and year on the screen. In practice there are very few commands designed to produce output on the screen specifically. The programs are written to produce output on what is called the standard output, and Linux sets the standard output to be the screen by default. That is how the output happens to appear on the terminal.

The shell, which interprets all your commands and passes them onto the Linux kernel for execution, has a facility to alter the standard output. In other words, you can define a file, rather than the screen, to be your standard output. (Actually the terminal screen is also a file as far as Linux is concerned.) To do this you need to say

kumarr@linux kumarr] ca1 > ca1fi1e

There can be zero or more spaces before and after the sign. This sign indicates that the standard output of the command preceding it should go to the file specified to its right rather than to the terminal screen. This is called redirecting the standard output. In the current case the calendar for the current month will be placed in the file calfile. You can verify this by

[kumarr@linuxkumarr] cat ca1fi1e

although you could have redirected this output as well.

[kumarr@linuxkumarr] cat ca1fi1e > catfi1e

Is that not a way of copying calfile to catfile? Note that the file to which output gets redirected gets overwritten if it already exists. You can verify this easily by

[kumarr@linuxkumarr] ls -1 > ca1file

and now examining the contents of ca1fi1e.

There is another operator, which appends output to the file specified rather than overwriting it. This is achieved by

[kumarr@linux kumarr] ca1 06 1994 > > ca1fi1e

Now calfile will contain the calendars for the current month as well as for June 1994. Compare this with

[kumarr@linux kumarr] ca1 06 1994 > ca1fi1e which leaves only the calendar for June 1994 in ca1fi1e.

Thus the» sign is safer to use because it never destroys any data, but this operation will keep adding to the file, and it can sometimes be difficult to make out what part of the output was produced by your last command and which portion is the outcome of previous redirections or was simply the original content of the file.

Standard Input

Just as many commands produce output on the screen, some commands take input from the keyboard although most take input from files. Look at an aspect of the cat command you have not studied so far.

[kumarr@linuxkumarr] cat

The result of this command is deafening silence. The uninitiated might wait several minutes before aborting the command, thinking there is something wrong because the system does not appear to be doing anything at all. The truth is that ca t can take its input both from the standard input as well as from a file. However, the output is always produced on the standard output. If any filenames are specified they are used as the input but if none is mentioned the input is taken from the standard input. There are also some commands that take input only from the standard input.

In the present case no filename has been specified and cat is waiting for input from the standard input, the keyboard here. So if you type

something cat writes it out to the standard output and the effect is that of echoing your input.

A foolish consistency is the hobgoblin of little minds -Emerson

A foolish consistency is the hobgoblin of little minds-Emerson

If you want to put an end to your misery you can terminate your input file by saying ^d, thereby causing cat to finish and present you with your prompt. To redirect standard input, say

[kumarr@linuxkumarr] cat < catfilesrc

whereupon cat will print the contents on the screen. This is just the same as

[kumarr@linuxkumarr] cat catfilesrc

because cat can take its input from a file as well. So to copy this file to catfiletarget *F* you can say

[kumarr@linuxkumarr] cat < catfilesrc > catfiletarget

or

[kumarr@linux kumarr] cat catfilesrc > catfiletarget

Thus you can redirect both standard input and standard output in the same command. Some commands do not take input from the standard input. In such cases redirection of the input is not possible, as with the 1s or who commands.

Remember that redirection is a facility provided by the shell, not by the command.

The command being run does not know or care what it& standard input and output are connected to, and it continues to use them. So the command has to be designed to take input from the standard input if redirection of input is to be possible. Thus you cannot say

[kumarr@linuxkumarr] cp < cpsrcfile

because cp does not take its input from the standard input. Similarly, output redirection is not possible unless the command is designed to write to its standard output.

Standard Error

So far we have seen the effect of redirecting the output of some commands that completed successfully. Let us look at this a bit more closely. For example, if there is- no command like gah, say

[kumarr@linuxkumarr] gah > gahfile

If you do so you will find that you get a protest message from Linux on the terminal but that gahfile is empty. Similarly

[kumarr@linuxkumarr] Is -1 gah > 1sfile

produces a message on the terminal but nothing in lsfile. Why does the redirection fail? After all the command did produce output.

The reason is that there is a third standard file in Linux, called the standard error.

Linux utilities and programs are usually designed to provide error messages in case there is something wrong and the program is not able to proceed as expected. Such messages are often referred to as diagnostic output because they can help the user diagnose the reason for failure. This kind of output is usually written to the standard error file. Usually the standard error is also connected to the terminal by default, but like the standard input or output, the standard error can also be redirected. To do this in the bash shell, say

[kumarr@linuxkumarr] gah 2> gahfile

This will place the standard error in gahfile. How does one place both the standard output and standard error in the same file? For this, say

[kumarr@linux kumarr] Is -1 gahfile yy > Isfile 2>&1

To redirect the standard error and standard output to different files, say

[kumarr@linux kumarr] Is -1 gahfile yy> Isfile

2>lserrfile

Filters

A filter is a command which can take its input from the standard input and can produce output on the standard output. Having the capability to read from or write to files is not a disqualification. So 1s is not a filter

because it does not read from the standard input but cat is one because it can do so (although it can read from a file as well) and also writes to the standard output.

You can think of a filter as a "device" placed between the standard input and the standard output which filters the standard input before placing it on the standard output. In the case of cat there is no filtering action at all, but a command like grep does perform some weeding action on its output.

The standard output of a command can serve as the standard input of another. Several commands can be chained together like this. Such an arrangement is called a pipeline. Pipelines are one of the big strengths of Linux, because they often enable us to group several existing commands quickly to perform a task for which there is no command directly available.

A major design goal of Linux was to have an operating system which allowed easy sharing of data and programs, and allowed people to build on the work of others instead of having to do things from scratch. The facility of pipelining helps meet this goal because you can piece together commands written by different people to achieve your objective rather than wasting your time on doing things which have already been done. Let us take a simple example.

Suppose you want to find out how many of the files in a directory are directories rather than ordinary files. It would have been wonderful if there had been an option to Is which did this job, but since that is not the case we will have to try something else. One-way .is to look at the directory listing with Is -p and count lines, which end in /. Such a visual method is tedious and prone to error, especially if there are many files in the directory. So let us try to make Linux do this for us. How about the following?

```
[kumarr@linux kumarr]ls -1 -p > tmp

[kumarr@linux kumarr]grep -c '/$' tmp
```

We first get the listing in a temporary file tmp and then count the number of occurrences of / at the end of a line in tmp using the grep command. The result will be available on the standard output. While this method will work it has a few disadvantages. One is that it is slow because an intermediate file has to be created. Secondly we cannot start the grep command before the 1 s finishes. Also if we run many commands like this we will be left with temporary files, which we will

have to meticulously delete lest they clutter up our directory listing and otherwise waste disk space. So we can use a pipeline like this

[kumarr@linux kumarr] Is -1 -p I grep -c '/$'

The '|' symbol is the pipe character. It means that the standard output of 1s -p is passed to grep. The act of connecting the standard output of a command to the standard input of another is also referred to as piping the output of the first command to the second. Here no temporary files need to be created or cleaned up by the user as Linux itself takes care of the details. Also the speed improves because the subsequent commands can start as soon as some data is available to them.

A command like Is which does not take its input from the standard input can only be the first command in a pipeline. Similarly a command which does not write to the standard output can only be tile last command in a pipeline. Also, it is the user's responsibility to see that each command receives input in a form which it can meaningfully transform, otherwise the results will be gibberish. Thus do not pass data files other than text files to grep because grep works only with text files, with lines delimited by the newline character.

3.4 Shell Scripts

The shell in Linux is a wonderful entity that serves us in various ways. It is started up automatically every time you login to the system. The shell sets up your environment when you start off on the machine. It is the shell that lets you run different commands without having to type the full path name to them, even when they do not exist in the current directory. The shell expands wildcard characters, thus saving you laborious typing. It gives you the ability to run previously run commands without having to run the full command again. It is tile shell that does input, output and error redirection.

You can use the shell as a programming language. It has all the usual language constructs like sequencing, looping, decisions, variables, functions and parameters.

Here we will take a very brief look at bash, the shell commonly used in Linux. A shell itself is a program and there can be many different shells available. Linux also has a shell called tcsh that has a C like syntax and is an enhanced version of the Unix C shell. Bash is the Bourne again shell and is compatible with the Unix Bourne shell. To discuss the capabilities and features of bash to some extent would require an entire block in itself. What we will try to do here is introduce some basic features and refer you to the documentation or to other books on the

subject for more detail. To become comfortable with shell programming, you will need to practice a 10t, j!1st as you would need to do for any other programming language.

After writing some shell programs you will realize that some Linux commands like sed that earlier seemed to you to be of limited utility are actually very useful. Shell programs are often called shell scripts.

A Linux machine can be thought of as being composed of several layers. At the lowest layer is the hardware which does all the physical tasks and without which there would be 1)0 computer and no Linux. Above that is the Linux kernel, which is the core of the operating system and does memory management, device handling and all the other mundane tasks needed to make the hardware easily usable by us. The Linux commands and utilities come next. At the top is the shell which Call be considered to be the outermost layer and which enables us to run tile utilities and other Linux commands. You can also construct higher application layers of your own which run above the shell. However, despite the layering that we have talked of, the shell is itself a program like any other.

Wild Cards

Wild cards characters are characters which can stand for characters other than themselves, somewhat like a joker in a pack of cards (though, unlike wild card characters, a joker has no intrinsic meaning by itself). A judicious use of wild card characters can make many commands easy to issue by saving a lot of typing and .preliminary research. Suppose you have been writing a series of programs for enciphering text. You have been calling them cph01.C, cph02.C, cph03.C and so on. You suddenly realize that you have been doing this in the directory ~khanz/crypt, while actually these programs are for a particular project and you would like them in the directory ~khanz/crypt/knapsack. All you have to do to rectify the situation is to move your programs to the correct directory after creating it. So you can start off

[kumarr@linux kumarr] cd ~khanz/crypt

[kumarr@linux kumarr] mkdir knapsack

[kumarr@linux kumarr] my cph0l. C knapsack

[kumarr@linuxkumarr] my cph02. C knapsack

[kumarr@linuxkumarr] my cph03. C knapsack

Soon you get sick of typing almost the same thing again and again, even if you just use the up arrow key and keep changing the required characters in the filename. Moreover, when do you stop? If you have

used a naming convention whereby the filenames have numbers from 01 onwards, you could stop as soon as the mv command reports that the file does not exist. But this method is hardly a rigorous one. What if there are gaps in the sequence, and cph08.C does not exist but you have other programs going up to chp39.C? So you would have to keep looking at the directory listing first, and also from time to time in the process, just to be sure. With many such files, the method is tedious and prone to error.

There is much less effort if you use wildcards. After making the subdirectory, just say

[kumarr@linux kumarr] mv cph?? C knapsack

The? is a wildcard character that can stand for any single character including itself. So cph?? C expands to cph followed by any two characters and then by .C. Also remember the? stands for exactly one character. Therefore a filename like cphl .C will not be matched by the command given above. To match that filename, you would need to say cph? .C. SO the command given will leave any files from cphl .C to cph9/. C in the original directory.

What do we do if our naming convention for files starts the filenames with cph but allows any characters after that, with of course .C at the end? Does it mean we have to give several mv commands with one, two, three, and many more? characters? That would be quite tedious again. Also, after how many? characters could we stop, knowing that there is no specific limit on the number of characters in a filename? The answer to that is another wildcard character, the *. It can expand to any number of any other characters. So all you have to do is to issue the command

[kumarr@linux kumarr] mv cph * .C knapsack

However, the * does not expand filenames starting with a leading. character. Also, in bash, a filename like yy* .c is not expanded unless there is at least one filename around to which it expands. So if you do not have a filename starting with yy and you issue a command like
[kumarr@linux kumarr] vi yy*. c

what the vi editor will create is a file called yy* .c. If a file like yyl .c already exists, then you can create yy * .c by escaping the * with a backs lash, so that it is taken literally.

[kumarr@linuxkumarr] vi yy\ *. c

You can create a file called yy? .c in the same way, though it might be a good idea to avoid such characters in filenames to prevent confusion.

Simple Shell Programs

Let us now look at a simple shell program. Suppose you are kumarr and are working on your cryptography project in a directory ~/prj / crypt /pkc/ src, where your source files are located. But sometimes you are looking at documentation in another location, or are also working on some other project. To look at the files in this directory you have to issue a longish command, which can be cumbersome if you have to d9 it often. What you can do instead is to create your own shell program, say in a file called wd. This you can do using any editor. The file wd should contain

Is ~/prj/crypt/pkc/src

Now you need to make wd executable by setting its permissions to 755. Otherwise you would need to invoke it by saying

[kumarr@linuxkumarr] bash wd

But if you just said wd, it would not work because when bash is given some command to execute, it looks for the commands in different directories in a fixed sequence. Typically this sequence is /usr/local/bin, followed by /bin and/usr/bin, but this can be controlled in a way we will describe shortly. Since wd is not in any of those directories, Linux complains. So you can say -fwd to take care of this problem.

What we have now done is created a command of our own that anybody can use. Currently it allows others to look at our directory, something we might not really care to do, but it nevertheless illustrates the point. And we have created the command by taking an already available command and using it in a different way. This is at the core of the Linux philosophy of building on the work of others.

Many installations have locally useful commands available in /usr/local/ bin. You could also be contributing to this repository of useful commands by and by, and you should first explore to see whether the task you want to do is not already accomplished by using these commands. However, before you can release shell scripts for general use, you will need to make them robust, which is something we have not looked at in the current case. If you have some commands that you feel you need but are not fit for general release, you can place them in your own bin directory like -/bin.

Suppose the system default form of the 1s command at your installation is not what you find useful; you would like it to be 1s -1. One way is to create a new command like Is that contains the line /bin/1s -1. Note that if you put only 1s -1 in your private command, it might keep calling itself indefinitely and so will not work. Also your program will not be able to use arguments because we have not given that ability.

Variables

Variables can be defined and used in bash like in any other programming language. For example, to set the value of a variable vehicle to "bus", you can say

```
[kumarr@liilUxkumarr] export vehic1e=bus
```

To see the value of a variable you can say

```
[kumarr@linuxkumarr] echo $vehic
bus
```

If the variable has not been defined, nothing is printed. You can also print several
variables together or print literals using the echo command.

```
[kumarr@linuxkumarr] echo $vehic.le and car
bus and car
```

To set the value of a variable to the output of a command, give the command in
backquotes.

```
[kumarr@linux kumarr] export mydir: 'pwd'
[kumarr@linuxkumarr] echo $mydir /home/kumarr
```

To let your command take arguments, you can refer to them as $1, $2, $3 and so on up to $9. $0 stands for the command itself and $10 wound be interpreted as $1 followed by O. If you want to use all arguments then say $*. Similarly, $# stands for the number of arguments. So if you write a command ech that echoes only the first and fifth arguments, it would have

```
[kumarr@linux kumarr] cat ech
/bin/echo $1 $5
```

Now you would find the other arguments would be ignored. The shell has several inbuilt variables of its own that you can look at by using the

env command, to show the environment. You would be able to recognize several of them such as HISTSIZE for the number of commands that are kept in the history buffer, LOGNAME for your login name, SHELL for the shell you are using, HOME for your home directory and so on.

Programming Constructs

The shell scripts we have seen so far have been nothing but a sequence of commands that we could have anyway issued at the prompt itself. Shell programming would not have been of much use in that case. What makes it powerful are the programming constructs available such as loops and decisions. Let us first look at a program mkupper that converts the contents of its arguments to upper case.

```
[kumarr@linuxkumarr]catmkupper
/ for i in $1 $2 $3
do
 tr ' [a-z]' , t~-Z]' < $i > $i. up
done
```

This converts the contents of up to three files to upper case and places them in a file of the same name with a .up suffix.

Besides the for loop, you can use the while or until loops with their usual meanings. For example, the script above could be written

[kumarr@linux kumarr]. cat mkupper

```
while test $# -gt 0
do
tr '[a-z]' , [A-Z}' < $1 > $l.up
shift
done
```

Here suppose there are 25 arguments to the mkupper command. The conditions in the while loop tests whether there is an argument available. If so, the contents of the file are converted to upper case. After each argument has been dealt with, it is discarded and the next argument becomes the first argument. This is done by the shift command in the rnkupper file. Finally when there are no arguments 1eft,the test fails and

the loop terminates. You can achieve a similar effect using the until loop, given below

```
if test $# -eq 0
        then echo 'No files to translate'
        exit
else
        until test $# -eq 0
        do
                tr '[a-z]' '[A-Z]' < $1 > $l.up shift
        done
fi
```

The test operation can be used to check various conditions, such as if a variable equals a number. You can also use the other relational operators with the keywords 1e, gt, 1t, ge and ne. Other tests that can be performed are -w or -r for checking if the filename is writeable or readable, -d to check if it is a directory and - f to see if it is an ordinary file. In the script above we print an error message and exit if there are no arguments to the file. We could also have chosen to exit silently in such a case, as we did with the while loop example. Or we could wait for input from the standard input. The example also shows how to make a decision using an if statement. The statement is terminated with the fi keyword. The else part is optional. We can also use the exit keyword to break out of the shell script. Several else parts can be present in an if statement, they are then introduced with el if. Let us take an example script called countargs that counts the number of arguments and prints the result.

```
if test $# -eq 0

then echo “No arguments”

elif test $# -eq 1

then echo “Only one argument’

elif test $# -eq 2

then echo "Two arguments”

else

echo “Many arguments”

fi
```

Within a loop you can use the break and continue statements to exit the loop or to go back to the beginning of the loop respectively. Comments are introduced by the # character. Anything after this on a line is treated as a comment. In a shell script, like in any program, it is important and

recommended practice to provide comments *j* that explains the working of the program.

You can also use the case statement in place of multiple if. ..e1if statements when it is the same condition that has to be checked each time. For example, let us write a shell script that checks the first argument to decide the operation to be performed and then performs the operation on the second argument. The case is terminated with an esac statement.

```
if test $# -ne 2
then echo “Usage: $0 operation files”
exit
fi
case $1 in
upper) tr '[a-z]' '[A-Z]' < $2 > $2. up; ;
lower) tr '[A-Z]' '[a-z]' < $2> $2.lw;;
*) echo "Invalid operation specified";;
esac
```

You must bear in mind that the options in the case statement have to be specified in the right order. If we give the *) option first, then it would match every case and the script would do nothing.

You can also pass input from the user to a shell script. For this use the read command. Take the following simple script that prints out the directory listing as desired by the user. It has a friendlier interface than the Linux command 1s.

```
echo "1 for long listing"
echo "2 for stream list"
echo "3 for single column list"
read x
case $x in
1) 1s -1 $*;;
2) 1s -m $*;;
3) ls -1 $*;;
*) echo “InvalJ.d choice”
esac
```

In the read statement you can assign values to several variables. The first value goes to the first variable, the second to the second variable and so on. If there are more values provided by the user than there are variables, the extra values go to the last variable. If there are few values, the remaining variables do not get any values. Values are delimited by spaces and a newline character causes the assignment to happen. So if you say

[kumarr@linuxkumarr] read x y z

abcd

[kumarr@linuxkumarr] echo $x

a

[kumarr@linuxkumarr] echo $y

b

[kumarr@linuxkumarr) echo $z

cd

You can do simple arithmetic in the shell with the expr command. The operators and operands have to be delimited by spaces. When you use the * for multiplication, you have to escape it with a\. lest the * be expanded to all available filenames.

[kumarr@linuxkumarr] expr 2 + 3

5

[kumarr@linuxkumarr) expr 18 / 3

6

[kumarr@linux kumarr) expr 4 \ * 5

20

[kumarr@linux kumarr) expr 7 -5

2

To help debug shell scripts you can use the -y or -x options that give verbose output. The -x option precedes each command with a + sign. Thus

[kumarr@linux kumarr] bash good1s –v

will print each command as it is executed.

SELF ASSESSMENT EXERCISE 1

Try the command?

echo*

Write files does it neglect to furnish you with? How can you get all filenames?

2) Write a shell script that prints out the contents of some fixed file in upper case.
3) Write a shell script that prints out a list of every unique word contained in the file in alphabetical order?

3.5 Graphical User Interface

Unlike the early versions of Unix, Linux has a good graphical user interface (GUI). This makes it different from the cryptic command line interface that proved daunting to most lay users who ventured to try Unix. Most of the common operations can be performed graphically and it is not necessary to use the command line. However, the power that the command line gives is still available to power users.

When you login you are presented with a text based screen, but thereafter you can have the system set up so that you reach the graphical user interface mode. This can be done by issuing the command

[kumarr@linux kumarr] startx

whereupon the X-window system is started up and you reach Gill mode. You have a default of 4 desktops that are available to you. Each of these can be arranged differently as you wish. To start up a terminal session, just right click the mouse on the desktop and select the "New Terminal" option. You will get a terminal window where you can use the mouse as well for editing, such as copy and paste. These options are available by right clicking the mouse or by choosing the Edit option on the menu bar in the terminal window.

The bottom tray contains some useful icons that will depend on what software packages have been installed. The leftmost icon is the red hat, the logo of the version of Linux that we are studying here. It has a small arrow to its right, clicking which brings up a list of options such as Accessories, Games, Graphics, Internet, Office and so on. Many of these options have further suboptions that you can select with your mouse.

The tray typically would contain icons for invoking the browser, email, the Open Office Writer, Presentation software and the spreadsheet, a printer manager and the icons for the desktops. To invoke any application, simply click its icon once in the bottom tray. This saves you having to know the name of the program for each application. When you move your mouse over any icon, a small explanation of the icon appears in a yellow box. It tells you the name of the program and what it does. This mouseover makes it easy for you to identify the correct icon for the program you want to run so that you do not have to rely on your memory to locate it.

If you right click on any of these icons, you get a menu where you have options to look at its properties or obtain help on the application. You can also remove the icon from the tray, but then if you want to run the application you will have to locate it in the directory structure. You can also move the icon around in the tray.

At the very left you have an icon in the shape of a red hat with an arrow pointing upwards. Clicking on this brings you to the main menu that has several options and suboptions. For instance, you can go to your home folder using one of the options. You can also copy and move files using mouse commands, or simply drag and drop files from one folder to another if you want to move them.

While in the folder window, you can right click the mouse and create a new folder. You can also rename folders or files or delete them.

3.6 Editor

Linux is rich in text manipulation and document preparation facilities. Here we take a brief look at vi, a line editor that is useful for creating and modifying text files. For example, if you want to be writing shell scripts or other programs, you will need an editor so that the program file can be created.

Text editors are different from the commands you have studied so far because you will be able to change existing files directly by using them. You need not be writing out the file to be changed to another file. Moreover, editors are interactive in that you need to be telling them what to do, command by command. This is unlike the commands you have seen so far, where you give the command once and it then does its job and terminates. Editors can be line editors or screen editors. Line editors work on a line at a time. Two line editors available in Linux are ed and ex. In contrast, screen editors present you with a screen of text and you can move around there, making changes as you want. So screen editors are more powerful and easier to use. A screen editor available in

Linux is vi. Again because of lack of space we will only be able to look at a few basic features of vi. It is a programmer's editor that is not too easy to learn, though.

Unlike some other editors, vi does not automatically create a backup copy of the file being edited. Although editing does occur on a copy of the file, this copy is in the tmp directory that gets deleted when you save the file. The actual editing is done in a buffer in memory and the changes are written to the file only when you tell it to do so. You can therefore easily abandon a session that has gone badly wrong. But the same feature can be a problem in case of a system crash, though vi will try to recover as much as possible of the file that was then being edited.

It is difficult to describe an interactive command on paper and so vi is best understood by trying out the commands on a terminal. To start up vi and edit a file called linuxdoc, you say

[kumarr@linux kumarr] vi 1 inuxdoc

At this the screen gets cleared, the file gets read into the edit buffer, the first portion of the window to the buffer appears on the screen and the cursor is at the first character of the 1ine that you were editing when you last saved the file. This presupposes that vi knows how to deal with your terminal type, a task that would have been ensured by your system administrator when she set up your machine. You can resize your window at anytime you wish without affecting your file. The bottom line of the screen shows the name of the file and its size in characters and lines. You can also see the current cursor position at the right of this status line. At times when you give commands to vi, the Status bar disappears and instead you see the command that you are issuing. We will shortly look at some of these commands.

The text that you enter in vi is organised in lines. The editor is not a word processor and works only in non-document mode. So you need to explicitly tell vi when a line has ended. Otherwise it will continue to add text to the line until it reaches *its* limit for the length of a line. How do such long lines look on the screen? They are wrapped onto the next physical line on the screen if the width of the screen has been reached. If you increase the width of the window you can see the line getting redrawn.

Similarly if you reduce the width of the window you will be able to see the line being rearranged on the screen. While by mere visual inspection you cannot make out whether multiple lines are actually the same line or are separate, you can easily find out, say by resizing the screen or going to the end of the line by the $ command.

You can start up vi with more than one filename, and you can even give wildcards.

[kumarr@linux kumarr] vi linuxdoc xx xx. c

or

[kumarr@linux kumarr] vi *. c

In such a case the files are presented to you one by one. After you are done with the first file, you are presented with the next one unless you choose to exit the whole operation, in which case the rest of the files are not presented to you. In this respect vi differes from other word processors where you can only bring up one file at a time. In fact you can start up vi without any filename at all. If you do that you will be presented with a blank screen and can then start entering commands. You can give the file a name when you save it.

Another option with which you can start up vi is the -x option that allows you to encrypt the file. You have to supply a key (you need to retype it to confirm) and when you save the file it is encrypted with that key. To now read the file you must supply the same key, without which it will not be intelligible. While not a foolproof method, it will certainly safeguard you against the casual busybody. One more option to vi is to open it in readonly mode with -R. You can navigate and examine the file as you wish but cannot make accidental changes as you have to use the force options to commands that alter the file. This mode can also be called up by using the command view instead of vi.

The -r mode of vi tries its best to recover from system crashes. If you want to start editing from a line other than the first, you can use the +n option where n is the line number you want to be at. To go to the last line of the file, just omit the number and simply say +.

From the number of different ways in which you can start it up, you would have had some idea of the kinds of commands available and the bewildering array of options that must be supported by vi. Now that you know how to enter vi, let us also learn how to come out of it. You can save the file with: w and force a save, say when you are in readonly mode, by: w!. Similarly: q will quit the file without saving changes and :q! will quit without trying to save any changes. To save and quit you can also say: x or simply ZZ.

Navigating Around the File

When you open up a file in vi you will find the cursor at the line you asked for during the invocation. Unlike a word processor, vi starts up in command mode. It does not start inserting the text you enter. So any key you press is taken to be a command. Vi has many commands, and just about any key is likely to be taken as one. Once a command is given, subsequent keys pressed will pertain to the command until you exit the command. If you press a key that is not a valid command in command mode, nothing will happen and vi will emit a beep.

For example, to move around the file, you can use the arrow keys. If your terminal type is not properly set and you have some problem with them, you can use j, k, I and h for down, up, right and left respectively. The commands I or h will work only within the line. At the last or first character respectively of the line, they have no effect. Similarly pressing k at the first line or j at the last line of the file has no effect. To scroll forward half a screen, use AD and to scroll half a screen back, use

^U. Similarly, ^F and ^B will take you forward and backward one whole screenful respectively. If you are at the bottom of the screen and you press j or the down arrow key, the display scrolls up by one line and your cursor continues to remain at the last line. To scroll forward one line without changing the line at which the cursor is, use ^Y. Likewise, ^E will scroll backward one line without moving the cursor. The commands 0 and $ take you to the beginning and end of the current line respectively.

Operations on the window can blank out the status line, which you can always get back by ^G. To move forward one word, say wand use b to move backward one word. flut unlike the character at a time commands, this works throughout the file, which means that pressing b at the first word or a line will take you to the last word of the previous line. The command will not have any effect if you are at the first word of the file. Similarly w will have no effect at the end of the file. Note that b will bring you to the beginning of the current word if you are not already there and to the beginning of the previous word otherwise. Similarly e brings you to the end of the next word if you are at the end of the current one, and to the end of the current word itself otherwise. These lower case commands define a word in a way that is similar to the definition of an identifier in programming languages like C. This is very convenient while programming, as vi is really a programmer's editor. If you want to consider a word to be a sequence of characters delimited by spaces, you can use the upper case equivalents W, B and E.

These commands all take counts. This means 20Wwilltake you 20 words forward and 90b will take you 90 words backward, with the appropriate definition of word.

The % command takes you to the matching opening or closing (, { or [character. To move forward a sentence, use (and) to move a sentence back.

Similarly [and] move forward and backward one paragraph. These too can be preceded by a number that specifies the count.

Adding, Deleting and Changing Text

By now you have a good idea of the kind of commands vi takes. With your feet wet, it will not be hard to understand the commands for actually changing text. First let us see how to insert text. You can do this by using the i command. This puts you into insert mode. In this mode, whatever you type becomes part of the file. The text is added starting from before the current position of the cursor. To come out of insert mode back into command mode, you can say ESC or ^ [. You can also use I to insel1 text at the beginning of the current line. Similarly, the a command adds text after the current position of the cursor, and A adds text at the end of the current line. To open up a new line under the current line, type 0 and use 0 if you want to open up a line before the current line. To add a control character to the file, type ^V followed by the character. The s command deletes the current character and puts into insert mode, while S deletes all text on the current line and puts you into insert mode.

Deleting text is similarly straightforward. The x command deletes the character at the current cursor position but does not move the cursor. At the end of the line, the command brings the previous character under the cursor. So pressing x repeatedly anywhere on a line will finally leave you with a blank line. The X command deletes characters to the left of the current cursor position and on reaching the beginning of the line no more characters are deleted.

To delete more than one character use the d command. This has to be followed by another letter to indicate what is to be deleted. So dw deletes a word. db deletes a word in the backward direction and dW and dB delete a word in the forward and backward directions according to the respective definitions. Word deletions happen across line boundaries. When deleting words on the next or previous line, the two lines are joined together. To delete the rest of a sentence from the current position, use d (, and d) will do the same up to the beginning of

the sentence. The same can be done for paragraphs as well with the d{ or d} commands.

All the above deletion commands can be preceded by a number to indicate how many times they should be performed. To delete a line completely, say dd, or use D to delete the rest of the line from the current cursor position. Similarly d^ will delete a 1ine from the beginning till the current cursor position. These commands will not remove the newline character. To delete a block of lines, give the range preceded by a colon, as in: 4, 10d to delete from the 4th to the 10th line (both inclusive). Here you can use a .to indicate the current line and $ to indicate the end of the file. So: $d will delete the rest of the file starting from the current line. You can also use the + t sign to indicate the number of lines to delete, as in: 15, + 8d to delete 8 lines starting from the 15th line.

Internally vi maintains the line number of each line in the edit buffer. You can display line numbers by saying: set number or: se nu. All commands that begin with a colon have to be followed by the ENTER key before they will take effect.

Changing text is done in much the same way. The r command will replace the current character with whatever you type next, while R places you in replace mode. Then each character you type will replace the next character until you press ESC. To replace a fixed number of characters by the same character, precede r with the count, such as 23r. The ~command changes the case of each character that is a letter. It too can be preceded by a count.

For changing blocks of text, use the c command followed by what you want to change, such as cw for a word, cw to use the other definition of a word, c$ to change text upto the end of the line and c^ to change text upto the beginning of the line. To change the complete line on which the cursor is located, use cc. You can precede cw, cW, cb, cB or cc by a count to indicate how many words or lines are to be changed. All c commands place you in insert mode that you need to exit in the usual way with ESCor ^ [.

The .command can be used to repeat the last action. To join the next line to the current one by deleting the newline character between them use the J command.

Searching for, Copying and Moving Text

leI us now see how to search for text. The f command searches for the next character you give on the same line. This can be preceded by a

count, so saying 3fx will look for the 3ro x from the current cursor position. Similarly F looks for the character in the backward direction. A; continues the search for the same character in the same direction, while a, continues to search for the same character in the reverse direction. Both these commands work only on the current line, but the; or, commands can be used on a different line after repositioning the cursor.

To search for any text, precede it by a / character. It place the cursor at the beginning of the first occurrence of the text pattern. To go to the next occurrence, say n and say N to find the same pattern in the reverse direction. If the pattern, say hello, is not found, the message

Pattern not found: hello

is displayed on the status line at the bottom of the window. You can start the search in the backward direction from the current curson position by preceding the text with? Both the / and? will wrap around the file end or beginning. So after reaching the end, the search will continue from the beginning of the file. You can use the d command to delete text from the current cursor position to an occurrence of a search string, say hello, by saying d/hello.

You can bookmark a position in the file. m followed by any letter. So mx will bookmark the current cursor position. Now you can reach that position from anywhere in the file by saying 'x. To reach the beginning of the line containing the bookmark x, you can say 'x. Bookmarks are valid only for that edit session.

You can undo the effect of most commands by the u command, but repeating it merely brings back the previous situation. So the command works like a toggle. However, you can use U to undo all changes on a line if you have not moved away from it. Deleted text goes into an unnamed buffer, from where it can be retrieved by using the p command and placed after the current cursor position. The P command places it before the current cursor position. You can use these commands repeatedly to place the contents of the buffer at different points in the file. Thus to take care of a transposition error involving two characters, say xp at the first character.

The y command yanks a block of text into the unnamed buffer without deleting it. So you can use yw or yy (also Y) to yank a word or a line. These can be preceded by a count. You can then place the text wherever you want by first going to that point and then using p or P.

Finally, you can use any of 26 named buffers by preceding the command with “followed by the buffer name, which can be any lower case letter of the alphabet. Thus "m4yy will yank 4 lines and place them in the named buffer m. To put back the text at some point in the file, just go to that point and say "mp. You can append text to a named buffer by using the same buffer name but in upper case.

This completes our very quick overview of the vi editor. Linux also has available an office suite that includes a full fledged word processor suitable for editing text documents, while vi is useful for editing programs.

Linux has a full screen windows based editor, gedit that creates text files. But it does not offer the weal!!1 of commands and features that VI does, in spite of tile fact that it is windows based. For a programmer, it might yet be better to use vi.

4.0 CONCLUSION

After learning to change your account password and close a login session in the last unit, this unit goes on to explain how to combine comments together to perform some useful functions. you have also been introduced to the concepts of standard input, standard output and standard error as well as how to write a shell script to produce your own comments.

5.0 SUMMARY

In this chapter we have covered a lot of ground and the treatment of the topics has had to be brief. We have looked at some text manipulation commands that are frequently useful. We studied the meaning of permission modes for files that you see in the directory long listing, and also saw how to change them for files that we own. Next we saw how we can construct our own commands easily by joining together filters with pipes. This also brought us to the concept of the standard input, standard output and standard error. We then looked at how to write simple shell scripts in the bash shell. The graphical facilities available in Linux were then touched upon followed by a brief discussion of the programmer's editor vi.

6.0 TUTOR-MARKED ASSIGNMENT

1) Try using the ca t command on a directory. What do you see?
2) What happens if you try to cat a non-existent file?
3) The crop command norn1ally reports files as different if they differ in any way. Suppose you have two files that differ at the beginning but their last portions are the san1e. How can you

ascertain using cmp that two files are the same after a certain offset?

4) Are you able to use wc on a binary file? Are the results meaningful?
5) Try executing a directory on which you do not have search permission, and another on which you do have it. What happens?
6) Can you look at the listing or a directory if you do not have search permission on it? Why? ,.
7) Can you remove files form a directory if you lack permission on them?
8) Which of the commands you have learnt so far are filters?
9) How will you count tile number of all files in a directory?
10) Look up the fee command. How do you think it can be useful?

7.0 REFERENCES/FURTHER READINGS

There are a host of resources available for further reading on tile subject of Red Hat Linux version 9.0.

http://www.redhat.com/docs/manuals/linux

http://www.linux.or£ gives among other information, a list of good books on Red Hat Linux.

Consider joining a good Linux mailing list.

UNIT 4 USER TO USER COMMUNICATION

CONTENTS

1.0 INTRODUCTION

In this unit, you will learn how to communicate with other users in Linux. Such communication can be offline or online. Online communication can be done using the write command. While instant messaging applications are commonplace these days, they are not specific to Linux and are available for most popular operating systems. We will take a quick tour of an instant messaging application from one of the popular providers to understand some of the facilities that are available. Electronic mail is one of the offline applications available in Linux and we will look at the Ximian Evolution e- mail client that also allows you to schedule your engagements. The Apache webserver is a very popular one, and you will learn the elements of configuring it to enable your computer to serve out web pages. You will also see how to configure some other network services such as the domain name service and a network file system server. Again, because of the lack of space, we will be able to look at only the basic concepts and features and will not be able to delve into the details of these matters. There are also some other services that we will not be able to touch upon in this short chapter.

2.0 OBJECTIVES

After going through this unit you should be able to do the following tasks. The configurations that you will be able to perform will be at a basic level and you will not become an expert.

communicate with other users on the same machine with the write command
use an instant messaging application to chat with another user on the network
use the Ximian Evolution e-mail client to send and receive mail
perform simple configuration of the Apache webserver on your machine
configure your machine as a domain name server
configure your machine to be a network file system server.

3.1 On-Line Communication

In Linux there are two ways of communicating with other users online. The first is available if the other user is logged in to the same machine. Whenever this is so the user will be a terminal on the machine to which you can send messages using the write command, even if that user has connected to your machine from another machine over the network. While the facilities are rudimentary, write is quick and simple to use. If you require more sophisticated features, you can use an instant messaging application from any of the popular providers, or use the gaim instant messenger that comes with Linux.

The write command

This command allows you to send a message to another user logged onto the same machine at the same time. It is thus a means of online communication because the recipient obtains the message immediately. There are other means of communication that we will look at that are offline in the sense that you send the message and the other user might not pay any attention to them if he so desires. Of course, even with the write command, the other user might not pay attention to the message that you send. The same is true for an instant messaging application as well. So the distinction between what we are calling online and offline methods is just that in the former, the other user or users are expected to be available at that time -they mayor may not choose to respond. If they do decide to engage in the communication, they are available at the other end and you can then conduct the conversation.

You should be careful while using write as it can be very annoying to receive a message which clutters up your screen while you are

concentrating hard trying to do some piece of work. Sometimes you have some precious output on the screen which you have obtained after some effort and a write message appears and causes that output to scroll off. If you are working in a terminal window you can scroll back to see your output, but it can still be annoying to be disturbed. Or you are working in an editor and the message garbles your screen. There are many such situations in which you would rather not receive any message on your screen. Again, some people are touchier than others on this issue. So you should be careful and use write only when you have something urgent to say, or when you are sure that the other Party will not mind. In fact, one of the most irritating times to receive a write message is when you are already replying to somebody else's write message. You might even want to arrange with some friends for such a situation to happen and see how it feels.

One thing you could do is to find out using the ps command what the intended recipient is doing. If your party is in the shell, it might not be so inconvenient for him. On the other hand, if he is in vi or some application package, then it might be best not to bother the person. Then again, the urgency of your situation needs to be taken into account. Another possibility is to send the user a brief write message asking whether it is all right to send more. If the recipient does not want to be bothered at that time, she can tell you so. All said and done, you should be aware of the fact that write can be a rude command to use.

After this long sermon, we can come to the command itself, which should be a breeze -for you after all of the two units of Linux experience that you have had. Let us find out some of the users on the system currently.

[kumarr@linux kumarr]$ who

ramk tty1Dec 10 20:47
W kumarr pts/0 Dec 10 22:59 (:0.0)
khanz tty2 Dec 10 21:29
pramod tty3 Dec 10 22: 11

Suppose you, kumarr, want to write to khanz and ask him about some program he had promised to give you. So you say:

[kumarr@linux kumarr]$write khanz
Can r have that matrix inversion program please?
I need it badly to test out my work.
^D

[kumarr@linux kumarr]$

We get back the prompt after sending the message. Notice a few things about write. One is that after issuing the command there is no indication at all that you should proceed with your message, because there is neither prompt on the screen nor any automatic acknowledgement which comes from the other end. The command expects input from the standard input and faithfully transmits it to the destination.

Therefore after issuing the command you can type in your message, which can be as long as you like. When you are finished, you should type the end of file character to signal the end of input. This character is usually ^D. On doing this you will get back the prompt. There is no indication whether the other party has received the message or not. Also the message should not be longer than a screenful or the other person might have difficulty reading it, even though you can scroll back in a graphical terminal window in Linux.

Now what about the other end of the channel? Let us zip across to khanz's terminal and see what he has got, omitting the secret work he was busy with when you butted in.

Message from kumarr@linux on pts/O at 23:08 ...

I need it badly to test out my work.

EOF

SO he did got your message, but what about a reply? Well, you did not see anything on your screen because he has not had a chance to answer yet. Let us see his reply now;

[kumarr@linux **kumarr]$** write kumarr

I was downloading it but the line got cut midway.

will give it when I have it.

Have patience or get it yourself.

^D

and that is exactly what you receive on your terminal except for the letters EOF in place of the ^D.

So we now have seen a straightforward case of exchanging urgent messages with write. Often such messages would have gone by mail. The write command is more commonly used for carrying on an online

conversation with others who are also logged in at the time, rather than just sending a message. Let us see how to do this.

Using the information from the who command earlier, let us say we try to converse with pramod.

[kumarr@linux kumarr]$ write pramod

Instead of immediately writing out your message, you can wait to see if pramod is in a mood to respond. After a minute or so you have not had a reply and so you just come out with ^D. He is too busy to talk to you, or maybe he just does not want to be disturbed and has arranged matters that way. This is getting to be irritating now, so let us try somebody else.

[kumarr@linux kumarr]$ write ramk

Message from kumarr@linux on tty1 at 11:44 ...

At last somebody is ready to talk to you! This means that rarnk has something like this on his screen.

Message from kumarr@linux on pts/O at 11:43 ...

Now you are both all set to carry on a conversation, because both have issued a write command for writing to each other and can consequently write to each other's terminal. So go ahead and pour out your woes to your only friend in the installation.

I wanted a program so badly but khan has not yet got it.

Which program? I am sure he must have tried

don't know what to do...

What is happening here? This can be disconcerting for beginners, but there is nothing mysterious about it. The communication is asynchronous and full duplex. Both sides can transmit and receive at the same time, and unless you wait until the other side has finished, there can always arise opportunities for confusion. What you need is a protocol to be adhered to so that the screen does not get cluttered up and cause bemusement. The thing to understand here is that there is no way of knowing when the other party has finished unless the protocol is set up and observed. This is because after every linefeed character the message is sent to the other side and there is nothing to restrict a message to one screen line. So the other party might respond before you have finished, especially if you have also entered a linefeed in your message.

There are many local conventions you could come up with for this, but let us see a protocol where we use ga at the end of each message and terminate the conversation with a bye. Now the previous conversation will be more intelligible to both parties.

I wanted a program badly but khan has not yet got it.

ga

Which program?

ga

I don't know what to do.
ga

I am sure he must have tried.

Bye

EQF

bye

^D

This was when everything was favourable. Sometimes you will have the following spot of bad luck.

[kumarr@linux kumarr]$ write ramk

write: ramk is not logged in

[kumarr@linux kumarr]$

This response is self explanatory. If you try to write to a user who is not logged in you will get this response from Linux. Since write is an online command, it is not enough for the user to be a valid user on that installation. The user must actually be logged in to the system at the time you try to write to him. You can always make sure of that by doing a who first. If the person you want to write to is not logged in, you can save yourself the trouble of trying to write, knowing you will fail.

Then there are the times when you check out the users of the system and find something like this:

[kumarr@linux kumarr]$who

ramk	tty1	Dec 10 20:47
kumarr	pts/0	Dec 10 22:59 (:0.0)
khanz	tty2	Dec 10 21:29
pramod	tty3	Dec 10 22:11
shyama	tty4	Dec 10 23:46
shyama	tty5	Dec 11 01:29

Now if you try to write to shyama you find

[kumarr@linux kumarr]$ write shyama

write: shyama is logged in more than once; writing to tty4

If that user is logged in onto more than one place, you will find that Linux automatically writes to the lowest numbered terminal into which that person is logged in. That is usually fine, but if you want to specifically write to the person at some other terminal, you can do so by giving the terminal number of your recepient as well with the command itself. Thus

[kumarr@linux kumarr]$ write shyama tty5

will write to shyama at the terminal tty5 rather than the default terminal oftty4. So you see that you write to a terminal, not to a user. It follows that you can write to yourself if you wish.

Are you at the mercy of others to be able to work peacefully? Not entirely, because there is a command in Linux to prevent other users from being able to write to your terminal. .In Linux every device is a file and you could use the chmod command to turn off write permission for others to that file. Then users will not be able to Write to your terminal and consequently will not be able to disturb you while you are working. But to use write you would need to know which device file corresponds to your terminal. An easier way is to use the mesg command.

[kumarr@linux kumarr]$ mesg n

Now anybody trying to write to you will get a message like this

write: kumarr has messages disabled on pts/0

If you just want to check the status of write permission on your terminal, just issue the mesg command without any arguments. To restore write permission, you can use the y argument to mesg. Any denial of write

permission lasts only for the duration of your login session. You can have a different write status on some other terminal to which you are logged in. You need to be aware of the fact that the superuser can always write to your terminal, irrespective of mesg status.

You should use the mesg command only when you are doing something important and do not want to be interrupted. Just as it is rude to unnecessarily bother a person, it is also rude to shut your doors to others trying to reach you, perhaps with something important.

The write command gets its input from the standard input and so you can always write the contents of a file to a

[kumarr@linux kumarr]$ write ramk < mymessage

Here the recipient will see the contents of tile file. So the message should be short. lest it be hard to read off the screen. This session will not be interactive, because it will end after the file has been sent off and the party at the other end will get an EOF. Another facility you have is to send a message to all users logged in at the time. This can be done with the wall command.

[kumarr@linux kumarr]$wall

Don't any of you guys want lunch

^D

[kumarr@linux kumarr]$

Broadcast message from kumarr (pts/0) (Sun Dec 12 21:38:50 2004):

Don't any of you guys want lunch?

EOF

Here even the sender gets the message. An ordinary user cannot be sure that all those logged in will get the message because some users might have disabled the receipt of messages. So this command is suitable for the superuser only. It is usually used to send important messages pertaining to the installation to the users.

Instant Messaging Applications

The write command that we looked at is rather rudimentary in that it does not offer very much by way of features or convenience, but it is

useful if you want to communicate quick and fast with no set up required. For something that is much more powerful, you can use any of the instant messaging applications available today. With these applications you can communicate online with any user anywhere in tile world if both of you are connected to the Internet, You can also hold conferences where more' than two users can participate.

The act of using an instant messaging application is often referred to as chatting. This has become a very popular means of communication today. While not specific to Linux, chat applications are commonly used and will be described here briefly. Such applications are available from various providers but the one we will describe here is the Yahoo chat, called Yahoo messenger. This should not be construed as any endorsement of this particular application by the author or by IGNOU. This example has been used because you will not be able to appreciate the features of such applications unless we discuss them with reference to some specific application.

The Yahoo messenger application has many good features that make it useful and popular. It is available free of charge from the Yahoo website and is provided under different operating systems including Red Hat Linux. Once the file has been downloaded, it needs to be installed on your machine. This act requires root permission and will therefore not be discussed in this unit. We assume that your system administrator has done this for you. You can set up things in various ways but we will also assume here that you have to start up the application yourself whenever you want to chat with someb9dy or want to set up a conference. For this you just have to issue the command:

[kumarr@linux kumarr)$ /usr /bin/ymessenger

If you have set up your path properly in your login environment, you can omit tile / usr /bin and simply say ymessenger after entering into graphics mode and creating a terminal window. This brings up two windows that read Yahoo Messenger and Login. To be able to use the application you must already have a Yahoo id. If you do not have one, you could click on the button in the Login window that says "Get a Yahoo! ID". In the Login window you can now login by providing your id and password. The password is not echoed on the screen as you type and instead, you see an asterisk (*) for every' character that you type. This is to provide some security because then somebody looking over your shoulder will not be able to make out your password easily.

There are also two checkboxes available in the Login window. The first one will tell the application to remember your id and password the next time you login. This can be very convenient, but there are two points to

consider here. First, you might well forget your password if you do not use it for long. That can be inconvenient, though you can provide some information that you gave while signing up for the id and obtain a new password. That process takes time, however. Secondly, it is certainly most unwise to use that option unless you are working on a personal machine that you do not share with others, or at least with those you cannot trust. On a public computer, you must never set this option. Otherwise somebody could masquerade as you and perhaps perform objectionable acts in your name.

The second feature is to be able to login under invisible mode. When you enter the application, you will not be visible to others as having logged in. This will prevent others from trying to converse with you as they will not know you are online. We will see later that you can also put yourself into invisible mode after logging in normally.

The other Messenger window is blank in the beginning. While the Login window is visible, the Messenger window will not respond to your input (except for window manager operations that allow you to resize or close the window). It also shows in a status bar at the bottom that you are Not Connected.

Once you provide a valid id and password the login window closes, the application connects to the internet and shows this in the status bar at the bottom of the Messenger window. It will also state that you are available, unless you have logged in invisible mode. You will initially not have any friends on the application so the middle portion of the window will be blank.

To be able to make use of the application, you need to have people you can chat with. These are called friends. So let us now add some friends to our list. To do this, click on the icon labelled Add, to open up a window that says" Add Friend".

The Add Friend window shows you four steps that you need to take to add a friend. You must know that person's id to be able to do so. Next, you have to decide what group that friend is to belong to. The application allows you to choose different kinds of groups into which you can classify your friends or contacts. You have the groups "Work" and "Personal" available to you by default. You can create a new group if you want to do so.

The ability to create groups is very useful as you can classify your contacts appropriately. For example, you could set up a group like "Family" where you put only your family members and relatives.

Business or workplace contacts could be in the group called "Work" and so on.

In the next step you can enter the identity under which you want to add your friend. This will default to your login id.

In the last step you can enter a short message that your friend will see when she next logs in to the application, thereby telling her that she is now on your friend list. She could then choose to likewise add you to her friend list under the appropriate group. After this you can click on the "Add Friend" button to add the person to your list of friends. Now the person is displayed in the Messenger window under the appropriate group as your friend.

You can change your status by right clicking on the status bar at the bottom of the Messenger window. You get a list of status messages that you can choose from.
These include "Busy", "Not At Home" and so on. When people who have you on your friends list login, they will see that status message against your name. Friends who are online at the time are displayed in bold, the others are displayed in ordinary type. Under invisible mode, your friend list is displayed in italics.

Now suppose your friend is online. You can now begin a chat session with that person by clicking on the Message icon in the Messenger window. This brings up another window that shows up the name of the friend in the title bar at the top. Below that are two rows of menu options. This is followed below by a text area where you can see the messages that have been exchanged in that session. Below that is another area where you can type in your message.

For this message area there are several options available to you. You can set your messages to bold, italics or underline foTn1at by clicking on the B, I or U buttons. That changes the style of the text that you type subsequently in the message area. To stop bold, italics or underline, just click on the button again -it acts as a toggle. The recipient at the other end will see the message with the proper style. You can select more than one modification to the text style at the same time, such as bold and underline simultaneously.

Besides, there are options to change the font and the size of the characters you type. There is a drop down list available for both of these that include a bewildering array of choices.

That is not all! You can also change the colour of the text that you type to any of some preset colours. As if that were not enough, you can create

any custom colour you want to by specifying the RGB or HSB values of the colour. And then there are the different emoticons you can send by clicking on the smileys icon. There are several of them to display all kinds of feelings or emotions.

Once you have constructed your message, you can press the enter key or click the send button to have your recipient see what you typed. This is visible at your end
also. Messages sent and received can be distinguished both by the id of the person as well as the colour.

When your conversation is over, or at any point that you wish, you can save the transcript of the entire discourse in a file on your disk. You could also cut and paste the conversation into a file in your word processor. If a person is not responding, you can "buzz" your friend to draw attention using the "Friend -> Buzz Friend" option in the menu bar. This will send an audible sound to the other end (will work only if the other party has their speakers on) as well as shaking the chat window of your correspondent.

There are several other options available and we will not be able to talk of all of them here. But one very useful and important facility is that of conferencing. Here you can invite a third party, and a fourth and more, to join in the conversation. Here all of you get to see what all of the others are saying. Thus you can conduct a meeting on- line. They are of course free to join your conference or to decline.

You can send a file to your recipient by using the "File -> Send" option or simply use the "Send File" option. You are then directed to another window where you can indicate the details of the file to be sent. With this you can also send a message to your friend. One more feature is the ability to ignore a user. You will then no longer receive any messages from them.

As your friends come online, you will receive a notification telling you about the event. Similarly when you come online, any users who have you on their friends list will be advised about the fact.

There are several preferences you can set to customize the look and feel of the application to your taste. These can be accessed by using the "File -> Preferences menu option. You can, among other things, change the privacy options, decide the colour scheme used or set the alerts that you will receive.

So you see that an Instant Messaging application is much more sophisticated than the rudimentary write command that we saw earlier.

However, for a quick and short conversation, you could still use write. These days you can also conduct a voice chat with your friends, effectively using your computer as a telephone. This would require a microphone and speakers at both ends and a reasonably good Internet connection. You can also conduct a voice conference.

3.2 Off-Line Communication

Let us now explore the facilities available in Linux for offline communication. This term is used here to refer to a method of communication where the other party is not necessarily on-line at the time. The recipient sees your message only the next time that she logs in. After looking at the array of features available in an instant messaging application, you might wonder what good it would be to use any other method! But consider some of its limitations and disadvantages.

The other party has to be available at the same time as you are. This can be inconvenient across different time zones.

Unless you keep yourself focussed, the conversation can tend to ramble and go on an on. This means more time spent on the chat proper.

If you are using text mode, then typing things out quickly can be difficult and tiresome unless you are a trained typist.

Infrastructural issues are important. If either side has a power or network outage, the conversation cannot happen.

Lengthy files cannot be sent, or even if they are sent, there is no chance for the recipient to peruse them.

We can get around many of these limitations by using an offline mode of communication such as electronic mail. While some of the good features of instant messaging might not be available, there are some advantages to it.

Communication is asynchronous. You send your message and the other party can reply at leisure. 1fthey are not available, tile message resides in their mailbox until they are ready to look at it.

File attachments can be sent and the other party can reply after examining tile file.

Typing speed is not much of an issue as there is no one impatiently waiting for your reply at the other end.

Infrastructural bottlenecks are of less significance. The message can be sent or received as and when resources become available.

At the same time, if both parties are online, e-mail can be almost like an online means of communication as people exchange e-mail messages rapidly one after the other. We will therefore look at the e-mail facilities available in Linux. Here we will not concern ourselves with the mechanisms of setting up an e-mail system, which is the job of the system administrator or the network specialist at your installation. We assume that has been done and we will only look at an e-mail client to explore the features and facilities that it offers.

Linux has a rudimentary e-mail client in the mail command. This is a text oriented facility that is difficult to use, especially in today's scenario where we are all accustomed to graphical interfaces. It works at the command line level and allows us to send, receive and otherwise manage e-mail messages. But here we will look at a more sophisticated e-mail client -Evolution from Ximian. To be able to use it, it needs to be set up on your machine. Here we assume that the basic set up has been done for you and that you are ready to use it as your primary e-mail client.

To start up the client, click on the little arrow near the Red Hat on your bottom tray. This brings up a menu of choices from which you need to choose "Internet -> Evolution Email". This takes you to the main window panel on your screen. This window has several parts that are briefly described below. You will find that there are many ways of reaching a particular option. For example, to see your inbox, you can click on the Inbox link on the right hand panel in the summary page, or choose Inbox from the left hand panel, called the Shortcut bar, itself. We will here look briefly at some of the main features of the client.

The client has features such as helping you keep track of your appointments and tasks that you need to do. Here we will not dwell on those aspects as we would like to focus only on the communication aspects of the client that are made available through the e-mail related facilities.

Menu Bar

At the top is a menu bar with the options File, View, Actions, Tools and Help. Each *or* these in turn has several options. The bar is context sensitive. For example, when you' are viewing the summary, you will *not* see the Edit button. Likewise, the options that appear under the View or Actions button depend on where you are currently.

The View option lets you choose the appearance of the window. You can choose to see the Shortcut bar on the left or to hide it. You can also choose to see or hide a Folder bar that shows all the mail folders that you have. This appears to the right of the Shortcut bar.

The Actions option lets you send or receive your messages through your e-mail server that has been configured for your client. If you choose this option then all messages that have arrived on the server will be transmitted to your client and will appear in your Inbox. Also any messages in your Outbox will be sent out.

The Tools option lets you change the Settings of the client. This will allow a host of preferences to be set. These include the settings related to mail messages, message composition, servers, contact lists and others. You can also configure the synchronization of your computer with a Palm Pilot.

The Help option gives you a full description of how to configure and use the Evolution e-mail client. You should refer to it for all the details that we will not be able to discuss here.

The File option lets you create a new mail message, set up an appointment or add a task. You can also work with folders, import external files, print or work offline. Many of these choices can also be used through other navigational paths.

Shortcut bar

This bar appears on the leftmost part of the window, unless you have chosen to hide it. It has several choices.

The Summary shows you some weather information in the centre pane, while the right hand pane shows the number of messages In your outbox and Inbox. You also see any appointments and tasks that you have left. All of these are links that take you to the respective feature. So clicking on the "Inbox" will take you to your inbox.

Clicking on Inbox takes you straightaway to your Inbox, where you can see a list of the messages that you have received with the sender, subject and date. You see the total number of messages and the number of new messages as well.

There are also shortcuts to your appointments calendar and todo list.

You can maintain you Contact list by clicking on Contacts.

If you right click on the Shortcut bar, you get a menu whereby you can change the appearance of the bar. You can now-

Hide the Shortcut bar, whereupon you can get more space for the actual option that you are at currently. So if you are in your inbox and you hide the bar, the space that becomes available is now used for your .Inbox itself. You can .get back the bar by clicking on the View option in the menu bar at the top of the window and selecting the Shortcut bar.

Choose to see large or small icons in the Shortcut bar according to taste. Small icons take up little space and give a compact appearance.

You can create another Shortcut group where you can put in your own shortcuts to folders that you choose, such as a folder that you have created.

You can change the names of groups or remove groups that you have created.

Inbox

The inbox shows you at the top statistics on the number of new and total messages that you have. By clicking on a column header such as "From", "Subject" or "Date", you can sort the list in ascending or descending order, or even remove the column. You can also add columns of your own by selecting one of the available choices and you can place the column wherever you like.

When you click on a message summary you can see the actual message in the bottom half of the window. You can now perform the usual actions of replying to the message, l deleting it or forwarding it to another set of recipients.

You can search through your messages based on several criteria, such as

What the subject does or does not contain

What the body does or does not contain

What the sender's name contains

What the recipient's name contains

Your own custom criteria that you can build up using the options provided

Composing Mail

One way of beginning to compose mail is to select Actions -> Compose New Message when in the Inbox. This brings up the "Compose a Message" window that lets you enter.

The "To" or recipient information
Information on those to whom you want to copy the message, using the "Cc" field.

You can choose to send a "blind" copy by using the "Bcc" field. Other recipients will not be able to see that the mail has gone to the Bcc addresses, but they will be able to see the other addresses to which the mail has been Cc'd. To be able to use the Bcc field, you might have to choose the View option on the top menu bar and check the Bcc option in it.

For the above three fields, you can bring up your contact list by clicking on the To, Cc or Bcc buttons. Now you can select by name the contacts to whom you want to send the mail. When the list appears, you can place contacts in any of the three fields that you need. This way you do not have to remember any e-mail addresses, nor do you have to type them in.

The subject line. Although this is not compulsory, it is good to enter a subject so that the recipient can make out what the message is about when it sits in her inbox. The subject should be descriptive enough so that this can be done.

The Format option on the menu bar in the Compose windows has many features.

You can now start entering the message itself. Here again you have many features available. To begin with, the message can be in plain text or in HTML. In either case, you can make the text bold, underline it, put it in italics or use strikethrough mode.

You can choose the font size for your message.

You can align the text that is entered -left, centered or right alignments are possible. You can change the colour of the text from among several preset choices, or you can even create your own custom colour.

You can create bulleted or numbered lists, with choices for different kinds of numbering such as Roman or alphabetical.

You can indent your text as desired.

While you are entering your message, you have several facilities available through the "Edit" option on the menu bar.

You can undo an action that you have performed by mistake, such as changing the indent or the font colour.

If you find you have undone something by mistake, you can redo that same action. You can cut or copy and paste a block of text from another file to the message window or vice versa. The cut option will delete the text from the original location, while the copy option will leave the original as it is.

You can find and replace text in your message.
You can spell check the text that you have entered in your message.

Besides, you can add an attachment to your message. This attachment can be any file that you like to put in. Various emoticons, or smileys, can be inserted into the message body as well. Apart from this, you can insert images that you have, hypertext links, ruler lines, text and html files directly into tile body of the message. Having done all this and composed your message, you can now use the "Send" icon on tile tool bar, just below the menu bar, to send your message off to the recipients.

If you wish, you can save your message as a file on the disk using the "File" option in the menu bar. You can also save your message in your Drafts folder for editing and sending later. This is useful when you are not able to complete an elaborate message in one session and want to continue to work on it later.

Folders

To be more organised, you can store your mail in various folders. The default set of mail folders consists of Inbox, Outbox, Drafts, Sent and Trash, besides which there is a folder for your contacts. There are also folders for your calendar and todo list, but we do not look at them here.

The Inbox stores all messages that you have received. New messages that you have not read are to be seen in bold. The Outbox stores messages that you have composed and that you have asked to be sent. They remain here until they have been sent off successfully. If, for example, the network link is down, the message might remain here for some time till the link is restored. The Drafts folder contains messages that you have composed and chosen to save instead of sending off. As we saw before, this could be because you need to work on the message

some more in a later session. The Sent folder contains messages that have been successfully sent off. They are moved here from the Outbox when this happens.

You can delete a message from any of these folders if you do not want them lying around. Such messages are not removed permanently at that time but are instead moved to the Trash folder. This gives you a chance to undo the delete if you realize you have made a mistake, or if you change your mind. To remove a message for good, you need to delete it from the Trash folder.

You can also create your own folders by using the File -> New -> Folder option.
You will be asked to specify under which folder you want this to be created. You might find it useful to create new folders to organise your messages better. For example, instead of letting all your incoming mail clutter up your Inbox, you might like to store it in folders organized by sender name or organisation.

Contacts

You can create a list of contacts with various details like their name, organisation, title, contact details with e-mail, postal address, phone numbers and so on. This can have other notes on them together with their birthday, spouse's name, anniversary, web page and other information too.

You can remove contacts and edit their information as well. This contact list can then be used while sending mail, as already described. You do not have to remember e-mail addresses and can just use the name. You can search for contacts using various fields as well.

Besides what we have described above, you can set a host of preferences for the look and feel of your client. With this wealth of features, it is a real pleasure composing and sending e-mail on your Linux machine. This is a big advance over the rudimentary facilities offered by the mail command that is inbuilt in Linux.

3.3 Apache Server Settings

The Apache webserver is a very popular webserver used by many websites around the world. Here we will take a brief look at its facilities and see how you can perform basic configuration of the webserver. While this will not make you an expert, it will give you an introduction and start you off.

In Linux, Apache version 2.0 is available as your webserver. This package can be installed while installing Linux. If it is not already available, then you can install the package that is called httpd using the command

[root@linux root]# rpm -i httpd

The configuration of Apache is specified in the configuration file that lies in / etc/ httpd/coni/httpd.conf. You could make entries into this file by hand if you wish. When you install the package, a default version of this file is put in automatically. However, hand configuring this file is only for experts, and you should not try it until you are thoroughly familiar with it.

You can start the http server by issuing the command

[root@linux root]# httpd -k start

If you now startup your browser and point to http://localhost, you should see the Apache test page. This indicates that the Apache webserver has been configured properly and that it is working correctly. If there is something wrong with the configuration you will get an error message.
To be able to work with the webserver, you will need superuser or root access to your machine. Ordinary users cannot configure the webserver or stop or start it. This is because the impact of a reconfiguration is installation wide and is not confined to the User who makes the change. Also it requires access to facilities that are accessible only to the superuser.

What if you are not an expert user and are not very familiar with the webserver? How do you begin to configure it? One way is to use the graphical HTTP Configuration Tool that comes with Linux. This gives you a user friendly interface and saves you from having to know any complex commands. To be able to use this tool, apart from superuser or root access to the machine, you need to have the X-Window system running. This needs setting up by your administrator, and after logging in, you can invoke it using the command

[root@linux root]# startx

You need to understand that if you use the HTTP Configuration Tool, you cannot configure the webserver by hand. This is because the tool generates the file after you exit the tool. Any changes you have made are therefore not preserved. Also there could be some modules that you have added to your webserver. The tool only supports modules that are made available with Linux. If you use any other modules from other sources, they cannot be configured using this tool.

To start the tool click on the red hat icon with a little upward pointing arrow. This is the Main Menu Button for Red Hat Linux. So you need to say Main Menu Button-> System Settings -> Server Settings -> HTTP Server. If you wish, you can also start up the tool from the command line at the root prompt.

[root@linux root]# redhat-config-httpd

This brings up the main window of the tool, shown in *Figure* 1. You are currently at the Main tab where you enter the basic settings for the webserver.

Main

In the Server Name, you should enter the complete or fully qualified domain name of the machine on which you are doing the configuration. This could be something like www.mycompany.com, or www.mycompany.co.in. If you do not give a valid name, when the webserver starts up, it tries to obtain the name from the IP address of the machine. The name of the machine that you give does not have to be the same as this one.

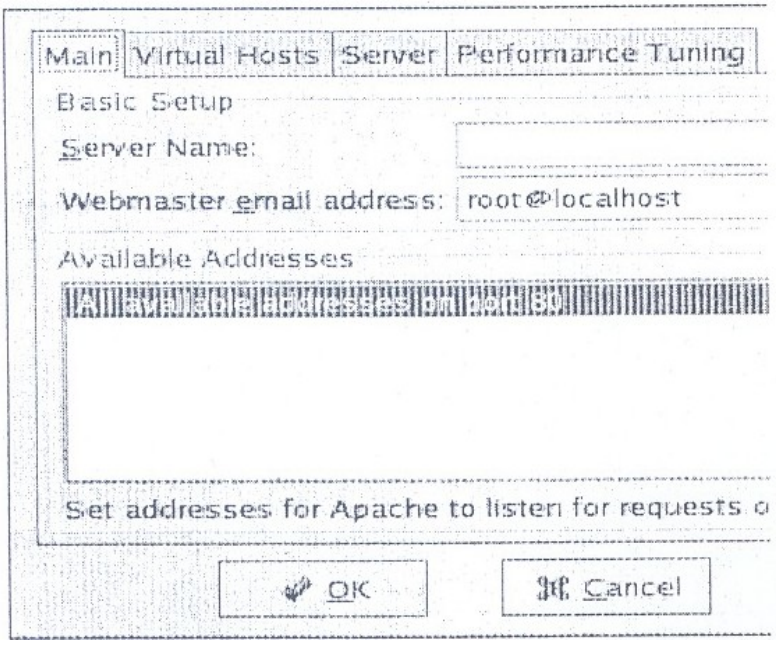


Figure 1: Main window of HTTP Configuration Tool

The next field that you enter is the e-mail address of the webmaster. It defaults to root@localhost as can be seen in the figure. You can configure the webserver's error pages to contain an e-mail address. This

can then be used by users to send e-mail to report errors. You can make the address something like

webmaster@mycompany.com.

Next you can set up the ports on which the webserver should listen for requests. The default is port 80 for non-secure communication. You can add an address by clicking on the Add button at the right. This brings up another window where you can either choose to Listen to all addresses or specify an IP address. You can also specify the port number at which you want to listen to requests from that address. You will have to make a separate entry for every combination of IP address and port number. It is better not to use domain names because of the possibility of DNS lookup failures.

You can edit or delete an entry by selecting it and then choosing the appropriate option from the buttons on the right. Note here that while the edit option lets you cancel your edit and seeks confirmation from the OK button before your entries take effect, the delete button immediately deletes the entry without further ado or warning.

Virtual Hosts

This completes your basic settings. Now you should go to the next tab that is "Virtual Hosts". On clicking here you will see the virtual hosts listed by name and address. Initially both will say "Default Virtual Host". In the lower portion of the window is a button labelled "Edit Default Settings". You need to click on this to bring up another window titled "Virtual Host Properties". On the left of this window you will be on the option for "Site Configuration". On the right you see entries for the Directory Search Page and the Error Page. You should not need to change these settings in the beginning. However, the tool does give you options to Add, Edit or Delete these entries.

The Directory Search Page is the page where the webserver looks for content to serve out when a user points a browser at the server's URL without specifying any directory or with a directory name ending in a *I.* These are searched for in the order that they appear here.

The Error Page is used to gracefully handle errors. The message in the footer is what 'the user sees in case of an error. If you click on the Edit button you can change this behaviour. The user can be redirected to any URL that you choose in such a situation, and this can be an internal or external one. For an internal URL choose the File option. Let us say you want to show the user a message that you have stored in a file called badrequest.html whenever he makes an invalid request. In the Error

Page, select Bad Request and click on Edit. In the behaviour, from the drop down list, choose File and enter the name of the file. This file must be under the Document Root directory.

The last menu option here is for the Default Error Page Footer. Here you can choose the default, no footer or the default together with the e-mail address of the person maintaining the website that was specified during the basic settings.

Having done with Site Configuration, you can move on to the Logging menu. Here you can configure the Transfer log and the Error log. The transfer log logs all connection attempts to the webserver, with their IP address, date and time and the file that the client is trying to access. The default for this is the file /var / log /httpd/access_log. Similarly the error log holds all errors that the webserver encounters and the default for this is the file called error log in the same directory.

You can change these default log files by entering either an absolute pathname or a pathname relative to the server root, You could also choose a custom log format by entering a string that defines the format, but we will not go into this here.

The log level determines the amount of logging that goes on and you can set to one of the eight possible levels, from Emergency that logs the least amount, to debug level that puts in a large amount of information. The default is to log only errors or more severe problems.

The last option here is the Reverse DNS Lookup, which you should leave at the "No Reverse Lookup". You can also do a Reverse Lookup or a Double Reverse Lookup, but this is not recommended because of the load it would place on your server and the Internet in general.

The next menu option under Virtual Hosts is Environment Variables. Here you can control the environment variables that are passed onto Common Gateway Interface (CGI) or Server Side Include (SSI) pages. This has three options -to set, to pass and to prevent an environment from being passed to the CGI script.

In the "Set for CGI Scripts" section, you can choose the Add button to add an environment variable to be sent to CGI scripts. This brings up a small window called "Environment Variable" where you can enter the name of the variable and its value. Then choose OK to add it to the list. Similarly in the "Pass to CGI Scripts" section, you can choose the Add button to bring up a window where you can enter the name the environment variable. This passes to any CGI scripts the value of that variable when the webserver was started. Click on OK to add this to

the .list. Finally, in the "Unset for CGI Scripts" section, you can enter the name of variables that you do not want to PIISS to CGI scripts. For all the three cases, you can also edit and delete entries already made.

The last menu option here is Directories. Here you can set various options for different directories. At the top are options that are the default for all directories not mentioned in the lower portion of the window. You can edit these default options by using the Edit button the right. In the bottom part of the window you can set options for specific directories. For this click on tile Add button at the right.

You now see a "Directory Options" window with various sections. In tile Order section, you can either allow all machines to access the directory, process a deny list first or process an allow list first. In the Deny List and Allow List sections, you can either select all hosts or specify hosts based on matching with a partially specified domain name, an IP address or a subnet. You can also put in the directory to which these options apply. Moreover, you can choose to allow directives in a .htpaccess file to take precedence. There are also some options that you can set. These same options can also be set for the default directories. These include allowing CG.I scripts to be executed, allowing server side includes and allowing tile following of symbolic links. You can also edit and delete directories that have options already set.

You would recollect that we have done all the above using the "Edit Default Settings" button, which means that what we did applies to all virtual hosts. Different virtual hosts can be hosted on the same physical machine, and they can have different IP addresses, ports or names. We will now see how to manage virtual hosts in Apache.

For every virtual host that you add, you can configure directories, environment variables, logging and site options the same way as described earlier. To add a host, click on the Add button which will bring you to the "General Options" button. You will now see a window where you can specify the name of the host, its document root, and the e-mail address for the webmaster. In the Host Information section, you can choose the type of host, which can be default, IP based or Name based. You should have only one default virtual host. For IP based virtual hosts you have to specify the IP address. This can be in the form IP: port if you want to specify the port as well, and you can specify multiple addresses by separating them with a space. You can likewise configure name based hosts by putting in the information that is asked for. Name based virtual hosts cannot work with SSL and can work only with a non-secure webserver. You can set up SSL options by choosing the SSL menu option for your virtual host. This requires that your

webserver have been set up to allow SSL support, and will require you to allow access through port 443 in the basic settings.

Server

The next tab is "Server" and it allows you to configure some basic settings for your webserver. You should not normally have to alter these, at least the lock file and the PID file. The Core dump directory is where the server dumps core if it crashes.

The Apache webserver must not be run as root as that would expose many security holes in your system. The user for the server is specified in the User section, and by default the user is Apache. Here you can also enter the group to which that user belongs. This too is Apache by default. The user id is what determines access to the system resources. Any file that cannot be accessed by this user cannot be accessible to the server and will return an error. Also any CGI scripts run are run with the user id of this user. So you must take care to see that any scripts that are not to be run by the outside world cannot be accessed by the webserver user. Also any files that you do not want to allow others to read should not be accessible to the webserver user.

You must ensure that the core dump directory permissions allow the Apache user to write to it. If not, then it will not be possible to write core to the disk in case of a crash **Performance Tuning**

This is tile last tab in the HTTP Configuration Tool. The window that it opens up has a few options that we look at here. In this case too the default options are likely to be appropriate choices for most situations.

The "Max Number of Connections" is the upper limit on the number of simultaneous client requests that can be handled by the webserver. This needs to be below 256 and is 150 by default. For a higher value, above 256, you will have to compile the Apache webserver again with the proper options – not recommended for beginners. If there are more client requests they will be refused unless connections become available.

The "Connection Timeout" setting sets the number of seconds for which the webserver will wait for a response to a communication request. The default value, as you can see, is 300 seconds. You can also set the maximum number of requests that are allowed per connection or allow any number of requests. If you allow persistent connections, the server will keep a connection open even after a request has been serviced. The time for which it will do so can be specified in "Timeout for next Connection". This should not be set too high as it could slow down the webserver as it waits for another request from clients.

With this we have looked at how to perform the basic configuration for the Apache webserver. There are of course many features that we have not touched upon, but after reading the material here you should be able to at least start off.

3.4 Network Server Settings

In this section we will look at how to set up your machine to serve as a Domain Name server and as a Network File server. A domain name server helps convert domain names to IP addresses so that human beings do not have to remember incomprehensible strings of numbers and can instead work with names that they find much more intelligible. You will need to have sufficient understanding of the domain name service and the program, bind that provides the service.

3.4.1 Domain Name Server

It is certainly possible to configure bind by hand, but that requires a good understanding of the format of the configuration files. If you are not an expert, the Bind Configuration Tool provides a graphical, user friendly way of performing basic configuration of the DNS service. However, any complex configuration will not be possible through this tool.

Like other server configurations, setting up the DNS service also requires you to have root or superuser access on your machine. Moreover, to run the tool, you need to be I working in the X-Window system so that graphical facilities are available. You can f start the tool through the command line at the root prompt, though.

[root@linux root]# redhat-config-bind

To start the tool from your Gnome desktop, click on the red hat icon with a little upward pointing arrow. This is the Main Menu Button for Red Hat Linux. So you need to say Main Menu Button -> System Settings -> Server Settings -> Domain Name Service. This will bring up the main window of the tool, as shown in *Figure* 2.

The configuration file for bind is /etc/named. conf. If you are going to use the configuration tool, do not edit the file by hand as all changes you make will be lost when the tool writes out to the file when it is closed. Unlike the HTTP Configuration Tool, though, you can also put in your own configuration that you make by hand, by placing it in the file /etc/named.custom.

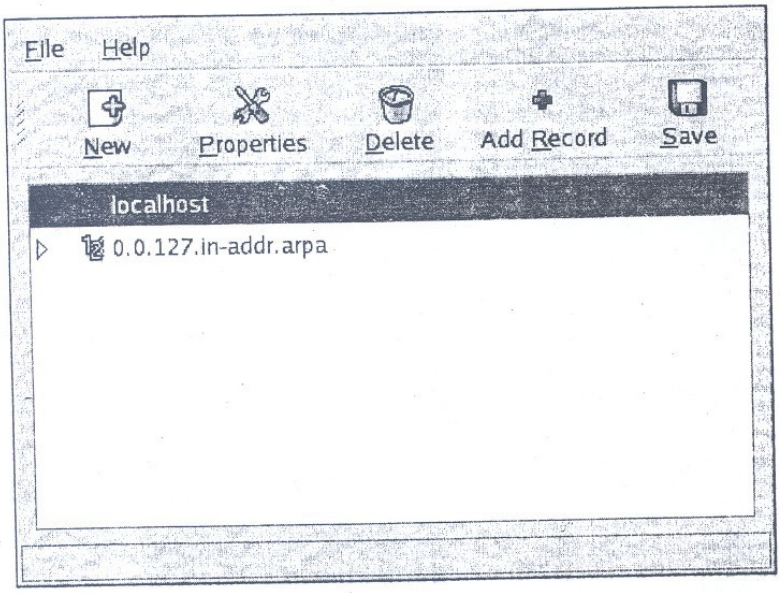


Figure 2: Bind Configuration Tool

You can work with the zone files and can add, delete or edit zones, which can be a forward master, a reverse master or slaves. To save your changes you need to use the menu option File ->Save or just use the save button. If you want to quit without making changes, use File -> Quit. The changes you made are reloaded by named so that they take effect right away.

You can add zones by clicking on the New button in the main window. A window titled "Select a zone type" appears where you can select from Forward, Reverse or Slave.

Forward Master

A window for "Name to IP Translations", shown in Figl1re 3, appears with the name of the zone that you just entered. The zone file name is the same as that of the zone, but with a .zone suffix. This filename is relative to the /var /named directory. In the contact field you can enter the e-mail address of the primary contact for the domain. The default, root@localhost, is probably not the best choice.

Next you need to enter the name of server that is authoritative for this domain. This is used to create the Start of Authority record for the domain. You must specify a primary name server and enter a record for it. The serial number has to be incremented each time there is any change to the configuration file, so that slave servers will update their own files. Usually it is set to the date of the file followed by a number,

such as 20041226001. To change the serial number from the default of 1, click on the set button and enter it.

The button for "Time Settings..." lets you change the different life cycle times that you have to specify in a DNS file. These are the time to refresh, retry, expire and tl1e minimum time to live (TTL). The values are to be entered in seconds, and here the defaults that appear, 28800, 7200, 604800 and 86400, are reasonable enough that you can leave them alone. However, depending on the characteristics of your site, you might have to alter these values.

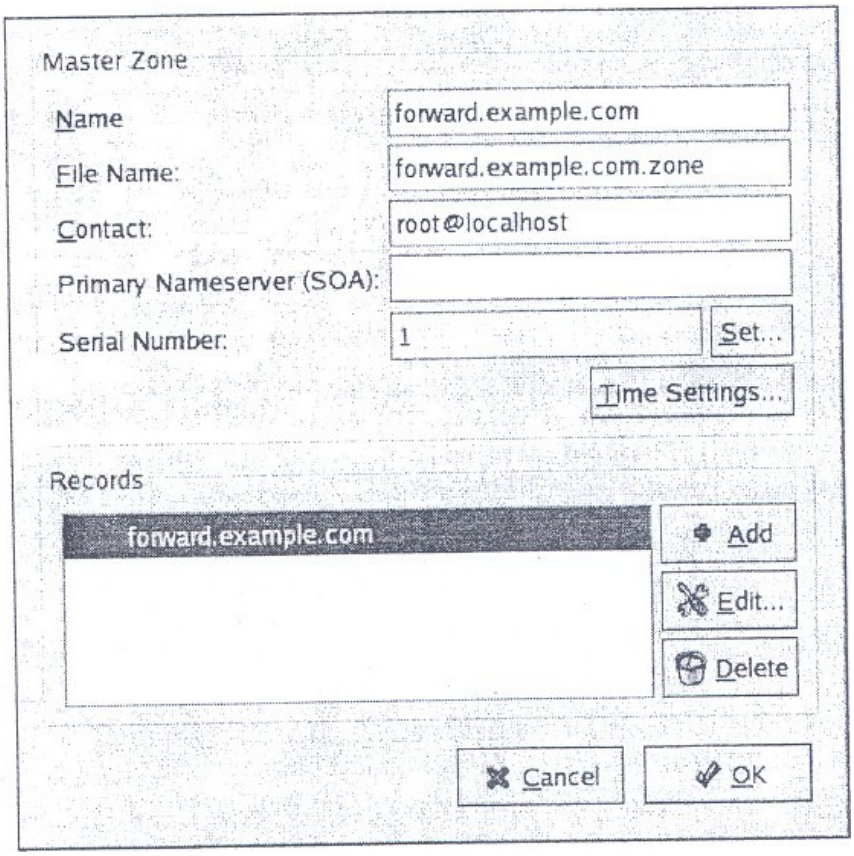


Figure 3: Forward Master Configuration

Finally you can add records for hosts, aliases and nameservers by clicking on the Add button in the Records section. You must add an entry for a nameserver. When you are done you can save the configuration.

Reverse Master

When you choose to add a reverse master zone, you have to enter the first three octets of a valid IP address. The tool performs the necessary

checks to see that the address entered is of a valid format. After you click OK, you will see a new window titled "IP to Name Translations" that shows several fields for you to enter. The first is the IP address field that is the same as the IP address that you just entered. The name of the zone is constructed from this by putting the octets in reverse order and appending ".in-addr.arpa". The name of the zone file is the same as this with the suffix being ".zone". The contact field contains the e-mail address for the primary contact for the zone. As in the case of a forward master, you have to enter the name of the server that is authoritative for the domain together with the serial number. This is automatically incremented by the tool and can also be set manually by using the Set button.

You have to again add the zone life cycle information and provide at least one name server for the zone. The portion here that is different from the forward zone is the reverse lookup table of that gives the hostname for each IP address in the zone. When you click on the Add button you are prompted to add the IP address and the complete hostname (ending with a period) for hosts in the zone.

At the end of this you can save the configuration or quit. If you save, the named service will take cognizance of the changes and they will take effect. The window for Reverse Master configuration is shown in *Figure* 4.

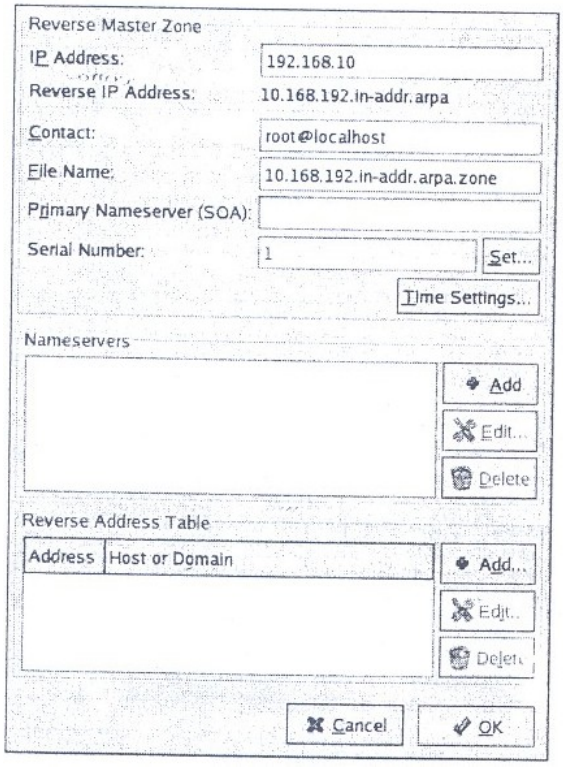


Figure 4: Reverse Master Configuration Window

Slave Zones

You can also add a secondary master or a slave zone to your configuration. These serve as backups to the primary master for the zone. Here you have to add information on the nameservers from which the slave is to pick up its data, together with the name of the DNS database file. This name is specified relative to the /var/ named directory. The nameservers are specified as IP addresses.

3.4.2 Network File Server

You can configure a Linux machine to work with a Network File System (NFS), where files on other machines on the network can be made available as if they were local files. A Linux machine can work as an NFS client, whereby it accesses files on the network. You can also configure your Linux machine as an NFS server, whereby you can let other machines access files on yours. In this section we will look at how this can be done.

As we have seen for the webserver and DNS server cases, although you can construct an NFS configuration file by hand, Linux comes with a tool to ease the task. This is the NFS Server Configuration Tool. It requires superuser or root access to use the tool. Being graphical, you must have the X-Window system running to be able to use the tool. But you can still start up the tool from the command line by issuing the following command at the root prompt.

(root@linux root]# redhat-config-nfs

The NFS Server Configuration Tool both reads from and writes to the configuration file /etc/exports, and so you can modify the configuration file by hand after using the tool. If you use the tool again later, it will understand and recognize your changes, provided you did the configuration correctly with the proper syntax. The main window of the tool is shown in *Figure* 5 below:

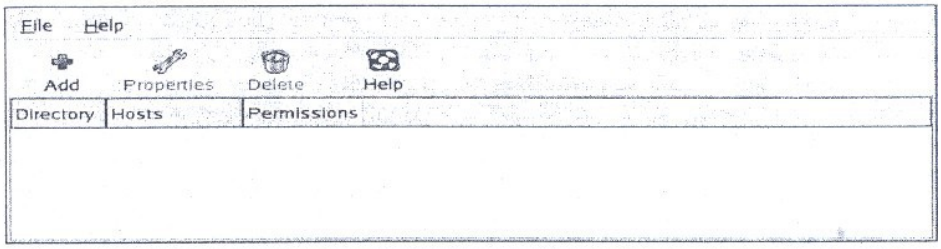


Figure 5: NFS Server Configuration Tool Main Window

To share a directory, called adding an NFS share, you need to click on the Add button above. This brings up a window with the title "Add NFS Share" that has three tabs. The "Basic" tab allows you to specify a directory and a radio button lets you decide whether you want to allow read-write or read only access to others on it. You also have to specify the machines or hosts that are to be allowed access to that directory. This can be done by:

Giving a fully qualified domain name. This should be something your machine can resolve to an IP address.
Giving an IP address.
Giving a host name, again your machine should be able to resolve this to an IP address.
Giving a group of machines by specifying them as a domain name or host name with wildcards. You can use a * for matching any number of characters except a period, and a? to match any single character.
Giving an IP network by specifying the network and a / followed by the number of bits in the netmask, or by specifying the netmask itself.

Doing the above makes the directory accessible to the host or hosts with permissions as desired.

The "General Options" tab has five options as described below:

If you want to allow ordinary users to be able to start the NFS service and allow shares, you have to allow the service to be started on ports higher than 1024. This does make the service less secure because the share does not require the concurrence of the administrator.

You can decide to allow insecure file locking.

You can decide to disable subtree checking. This is useful if you have exported an entire file system, because your server will no longer check to see whether a file requested by a client is in the directory that has been shared.

You can choose to force synchronization of writes immediately.

You can choose to disable synchronization of write options, where the server first writes out to disk the changes caused by a request before replying to it.

The "User Access" tab has tile following options that you can set.

> You can allow the superuser of a client machine root privileges on your machine. This is a big security risk and should be used only if necessary. Otherwise, by default, even the root user of the client is treated as an anonymous user on your machine.
>
> You can map all users on the client to the anonymous user on your machine. If you choose this option, you can set the user id and group id of the anonymous user.

You can now click on the OK button to save the configuration you have made. Of course you can add as many directories as you wish to share. You can also edit directory properties by selecting it and choosing the "Properties" button in the main window. This button is initially greyed out when there are no directories shared. Similarly you can delete a directory by selecting it and choosing the "Delete" button. Whether you add, edit or delete a directory, the configuration takes effect immediately after you save it. This is done by generating the new /etc/exports file and restarting the NFS server daemon.

4.0 CONCLUSION

In this unit you have learnt how to connect with other users in Linux through write command and emailing. It has also introduce you to how to perform simple configuration of the Apache web server and how to configure a machine as a domain name server and a network file server.

5.0 SUMMARY

This brings us to the end of this unit. Here we have looked at two broad themes- communicating with other users and setting up your machine as a server for different kinds of services.

In user communication, we have looked at online and offline communication. We have seen how to use the write command to quickly communicate with other users on your machine. It is a simple command that provides only a rudimentary communication facility. We then looked at an instant messaging application from Yahoo! That provides many sophisticated features and allows us to talk to anyone else connected to the Internet. The only demand this places on us is that it needs a good connection to tile Internet, while the write command has no such constraints.

For offline communication we looked at the Evolution e-mail client from Ximian. This allows us to communicate asynchronously with users

who might not be online at the same time that we are. It has a wealth of features that make the task of sending and receiving mail easy and we therefore did not consider the inbuilt mail command in Linux.

We then looked at setting up three different kinds of services. First was the Apache webserver that is available in Linux and is one of the most widely used webservers. We saw how to use the configuration tool provided with Linux to configure this webserver.

Next we saw how to configure our machine as a DNS server using the configuration tool available in Linux. Finally, we saw how to set up our machine so as to make it an NFS server.

All the server settings were discussed with reference to the user friendly, graphical configuration tools available in Linux.

6.0 TUTOR-MARKED ASSIGNMENT

1) What do you think are the advantages and disadvantages of electronic communication compared to:

   a) Postal communication
   b) A telephone conversation

2) Can you think of a couple of other ways of sending output to another terminal other than the ones covered here?
3) What happens if a user logs out after you have started writing to him?
4) If you wanted to send some message requiring some preparation, how would you use to write?
5) What is meant by being in invisible mode in Yahoo Messenger?
6) How do you go about chatting with a friend?
7) When would you want to buzz a friend?
8) What are the different status massages that you can show others?
9) How cab you save a transcript of the conversation you have had with a friend?
10) What are some of the advantages of e-mail over instant messaging?
11) Why would you not want to use the inbuilt mail command for managing your e-mail?
12) What is the utility of having different folders set up in your mail client?
13) Why would you want to set up an address book?

7.0 REFERENCES/FURTHER READINGS

There are a host of resources available for further reading on the subject of Red Hat Linux version 9.0.

http://www.redhat.corn/docs/manuals/linux.

httl!://www.linux.org Gives among Other Information, a List of Good Books on Red Hat Linux.

Consider Joining a Good Linux Mailing List.

UNIT 5 UNIX SYSTEM ADMINISTRATION

CONTENTS

1.0 INTRODUCTION

In the previous units you have been introduced to Linux and have got an idea of its features and the facilities available in it. In this unit we will look at how to administer a Linux system and take care of it so that you can m use of its power. This would mean being able to install Linux on a machine a configuring and setting it up so that you could start working on it. It also requires; maintaining the machine subsequently, such as by adding user accounts and keeping your data safe by backing it up.

The activities described in this unit are mostly performed as the superuser. So you have to be very careful and understand the commands you issue tll0roughly, because Linux does not conduct many checks on what the super user asks it to do. As an ordinary user you could not modify a file if you did not have permission, but the superuser can change any file irrespective of its permission settings. This might seem like a convenience, and it is indeed so. But it also means that you have the potential to cause severe damage to the installation through one mistakenly issued command.

2.0 OBJECTIVES

After completing this unit, you should be able to have an understanding of basic system administration tasks and be ab1e to perform basic installation, configuration and maintenance of a Linux system. Some of the abilities you should have acquired are:

understand what is meant by system administration
understand the responsibilities of tile system administrator
install Linux on a personal computer
understand how to boot a Linux system and what goes on while starting it up
how to add and maintain user accounts
understand, character and block special files
know how you can back up data from the system and restore it when required.

3.0 MAIN CONTENT

3.1 System Administration

You have started working on a Linux system and have learnt some of the basic facilities provided by the operating system. When you began your exploration of Linux, you were given a user account on a machine. For this to have happened, what are the actions that somebody would have performed for you? Let us try to work backwards and figure this out.

First of all, somebody would have had to load Linux on the machine that you are working on. This would mean having to know how the operating system is to be loaded on to your machine. Secondly, having got the machine working under Linux, somebody would have to set up user accounts and give them permission to log in to the machine and work on it. There are several other such activities that one can think of.

Configuring the system to connect to the local area network (LAN)

Adding new hardware devices to the system, such as hard disks, a printer or a CD-ROM drive. Taking backups of the important system and configuration files.

Booting up and shutting down the system as needed. Restoring the system to format working in case of a crash, by using the backups.

Fixing software problems that might arise, such as a corrupted system file. Addressing issues such as performance tuning if the system is

slow. Making sure that the system is secure from the assault of unscrupulous persons. Installing operating system upgrades and patches as required. Installing and configuring any new software that has been acquired by the organisation. Providing required network services such as e-mail, Internet connectivity and other services as appropriate. Helping ordinary users in trouble, such as those who might have forgotten their password.

Being able to do the above requires much more knowledge of the working of Linux than you have obtained so far. It also requires experience with various commands, tools and utilities. The tasks listed above are not the only ones that need to be performed for ordinary users to make use of the system and are only indicative.

Moreover, most of these responsibilities can be performed only by persons having superuser or root access to the system. Ordinary users, without root permission, cannot run many of the required commands. Because of this, these tasks have to be performed by experienced individuals, and with great care. A moment of carelessness could result in a wrong command being issued or an incorrect option being used, with the potential of catastrophic damage to the installation and to the work of several users who use that machine.

The activities we have described above are the domain of a role called the system administrator. The person who performs these tasks is called the system administrator Depending on the size of the organisation, the responsibilities of a system administrator can be discharged by one person or by a group dedicated to this work. These people need to have adequate experience and training or knowledge about the working of Linux, and need to keep themselves updated with the latest happenings to keep their r knowledge current.

In a larger organisation, the work of a system administrator can be divided into various specialities, with a different person or group looking after each such speciality. These can be:

A Network Administrator to look after the LAN, Internet connectivity and network services such as e-mail.

A Security group to ensure that the system is secure from hackers and other disreputable individuals.

A Hardware support group to keep hardware in fine fettle and to ensure it works well with your Linux system.

Sometimes you can have a Systems Group with different individuals or sub-groups f looking at these functions. The exact structure and division of responsibility will depend heavily on the kind of organisation, the number of users there and the nature of work performed. In some companies, all installations might be performed by a different group or might be part of the computer supplier's work itself. A software j development group in a financial services company might have much more need for security and so that aspect might receive more emphasis in such an establishment.

Any installation will tend to find the need for performing certain tasks again and again, or there might be a need to do certain things for which there are no specific commands or utilities in Linux. The system administrator therefore has occasion to create and use shell scripts that will help her do all this easily. She should therefore be skilled at shell programming. Such locally developed programs are often kept in a directory such as /user/local/bin for general use.

So we see that the system administrator is responsible for installing and maintaining the Linux installation. The specific nature of duties will depend on the organisation, but there are a few tasks that are common. The system administrator will therefore need to have adequate skill and experience in those tasks. In addition, he will often be faced with problems that have not occurred before. He should therefore be good at problem solving and innovation. Some of the qualities and training that he needs to have are:

Knowledge of the structure of Linux and its usage
Tools and utilities with their various options
Shell programming
Diagnosing the causes of problems.

These are therefore the qualities you have to imbibe as you take on the task of system administration. We will now look at how to perform some of the tasks of the system administrator.

3.2 Installing Linux

One of the basic tasks that you need to perform first of all, before you can do anything else, is to install Linux on your system. There are many ways in which you can do this and we will look at the main methods here. As always, because of the limited amount of space that we have here, we will concentrate on the main points. For more details you should refer to the documentation that Red Hat Linux makes available to you.

The installation process can be broken down into the following steps:

Choosing an installation method
Choosing an installation class .Pre-installation checks
Actual installation
Configuration and set up

In some cases you might find it expedient to perform an upgrade instead of a full blown installation. This is something you could decide while choosing the installation method. There are also different things to be taken care of when installing Linux on different kinds of computers. For large machines that are more powerful than the typical microcomputer, the vendor of the machine would provide the Linux installation and support mechanism. Here we are looking at installing Linux on an IBM compatible personal computer. If your PC is of another type, such as an Apple Macintosh or a laptop or an older IBM compatible, there could be additional intricacies involved that we will not be able to cover here. You would need to make sure that Red Hat Linux will work on that system.

So now let us assume that we have a computer on which Linux will work and begin the steps for performing the installation.

3.2.1 Choosing an Installation Method

You have to first decide how you are going to be performing the Linux installation, as there are many methods available. One straightforward way is to do a local installation using a Linux distribution purchased from Red Hat. In that case you get a CD-ROM that you can used to boot your machine with, the rest of the installation can then be continued using the Linux CD-ROMs provided. For simplicity, we are assuming here that your computer has an IDE CD-ROM drive and that there is no other operating system installed on it. This means that your computer is blank, with nothing else on it and your Linux installation is going to be the first time anything has been put on it. If you have some other hardware, then there will be an additional driver diskette that you will need to have and that you will have to insert at the appropriate time. You can boot the Linux installation program from a floppy also.

Some other ways of installing Linux are:

Using an FTP server that has the Linux images on it
Using an HTTP server that has the Linux images on it
Using an NFS server that has the Linux images on it.

In all the above cases, you will need a network driver floppy diskette or a PCMCIA driver diskette.

3.2.2 Choosing an Installation Class

There are different classes or types of Linux installations that you can choose from. They are appropriate in different situations and you can decide the one to use based or the intended end use of the system. Remember that these types really determine what kind of software components are installed and there is no fundamental difference between them. They are really a short, predetermined set of components or packages. The classes are briefly described here.

Personal Desktop

This is the simplest installation that brings in the least number of software packages. It is useful if the machine is going to be used by an end user, such as at home, or for office productivity. It will also install the GUI desktop environment with the X- Window system. This is useful for new users who are just getting familiar with Linux and will not want to perform advanced tasks.

Such an installation needs 1.8 GB of free space. During the installation you can still choose to install additional packages that are not part of the default installation. So what you really put on the machine is still up to you. But it does save you from the tedium of choosing every single package to put in.

This installation will use up to twice your RAM as disk space for your swap partition. The root partition will hold all the system files and user files. The size of the boot partition will be 100MB.

Workstation

A workstation installation is useful for professional software developers. It puts in a graphical user interface together with basic software development tools and more system administration utilities. It will use up to 2.2 GB of disk space. The other characteristics are the same as the Personal Desktop installation. Both of these can take up any other software packages that you want to install as you go through the process. Remember that if you choose to put in more packages, the disk space required will be correspondingly greater.

Server

If you are going to use your machine as a server then you would perhaps not need much configuration to be done on it. In such a case you can de a server installation without even putting in a graphical user interface. You would then need just 850 MB to 1.5 GB of space depending on

how much functionality you plan to put in. Should. you decide to put in the GUI, the space required would increase correspondingly. The other characteristics of the default server installation are the same as for the other classes.

Custom

A custom installation is the most flexible option as it does not have any predetermined packages that will be put in. You have to decide which packages you want as you proceed. This therefore requires an understanding of tile packages that are available and their dependencies. It is meant for advanced users who know exactly what they want and need the freedom to decide. The disk space required varies from 475 MB to 5.0 GB, depending on what you choose to put in.

All the above classes of installations have the same partitions if you choose automatic partitioning. Also, the disk space requirements mentioned above assume you are installing only the English language version. Should you choose other languages as well, the space requirements will go up in each of the above cases.

Upgrade

It is also possible to install Linux on a system that already has some earlier version of the same operating system. The old Linux version should be 6.2 or later, because that is when the rpm method of package installation was introduced. If that is the case, you do not have to go through all the steps needed in a fresh installation. The advantage is that all the data that you have will be preserved in an upgrade. Your partitions will not be altered and only the appropriate software packages will be upgraded. So you do not have to go through the exercise of backing up your data and restoring it after the installation has been completed.

3.2.3 Pre-Installation Checks

Before commencing your installation, it would be politic to gamer information about your machine. You first need to see how much hard disk space you have. That could have an impact on the kind of installation you can perform. However, in these days of large hard disks of80 GB and above, it is unlikely to be a constraint. Yet it is best to make sure.

When we refer to the disk space, it means unpartitioned disk space that is not currently being used by another operating system because every partition on the disk behaves like a separate disk drive. It is quite

possible to install Linux on a machine that is currently running some other version of Linux or of an operating system like Microsoft Windows of some flavour, such as Windows 2000 or Windows 98.

You next need to find about more about the type and make of the hardware on your machine, as this is information you will have to provide as you go ahead with the installation process. For this, you will need to refer to the documentation provided by the manufacturer of your machine. If you have an assembled machine, refer to the documentation that came with the computer components that you put together. In many cases, your existing operating system can give you information about your hardware. For example, in Windows 2000 you can look at your Device Manager tab under the System icon in tile Settings to find out about the hardware and the specific type and make. While this is no substitute for actually finding out about your hardware physically, it is usually sufficient for you to be able to perform the installation.

Here is a list of some of the hardware that you should get more information about, to be able to carry out your Linux installation without interruption.

> The hard disk -is it an IDE or a SCSI? What is its capacity and is there any volume label? What are the partitions you will want to create and what will be their sizes? For example, do you want to have /tmp or /home as separate partitions or will everything be in /root?
>
> Similarly you need to know about your CD-ROM drive. Is it an IDE or a SCSI type?
>
> Which is your hard disk controller? If using a SCSI adapter, who is the manufacturer and what is the model number?
>
> What kind of mouse do you have? Here you must know the mouse type, such as serial, bus or USB and the protocol, such as generic. Also note the number of buttons, which are commonly 2 or 3. Does it have a vertical or horizontal scroll wheel?
>
> What kind of monitor do you have? Here note the manufacturer and the model number for reference.
>
> What are the characteristics of your display adapter? This is especially important if you are going to install the Graphical User Interface. Remember the manufacturer and the model number as well as the amount of video RAM used.

Find out about your sound card. For this you again need to know the manufacturer and the model number. In addition, note down the chipset used.

Do not forget the amount of RAM in your computer.

Besides the above, you will need to have some more information for your computer to be able to access tile network. First you have to choose a name for your system. This is a personal choice and has to be unique within a domain. Organisations have their own schemes for this. Some use the names of rivers, mountains, sages or places. Others leave it to the system owner while still others use mundane numbering schemes such as marketing001. You will have to follow the organisation's policy here. Make sure you know the organisation's Internet domain name. This is typically derived from the organisation's name but could be different. For example, for IGNOU, the domain name is ignou.ac.in. In larger companies, the Internet domain name is further subdivided and if so, you have to know the domain name for your subdivision in the organization. This could be something like marketing.mycompany.com.

Next, make note of your network interface card. Again, here you must know the manufacturer and the model number.

Find out the IP address allotted to your machine. This could be a static IP address decided by the organisation. If it is allocated dynamically by DHCP, you should be aware of the DHCP server address that does this allocation. In many cases, while the address is allocated by a DHCP server, it is permanent in that it does not change from one boot to the next. In addition, you need to know the network mask for your machine.

In case your machine will boot off a machine on the network, you need to supply that machine's IP address.

You also need to know the IP address of the default gateway for your network that connects it to the outside world.

Lastly, you need to note down the IP address of the DNS server that your machine will use to resolve hostnames to IP addresses and vice versa. There could be more than one nameserver, in which case you need to know the IP addresses of those machines as well.

Once you have all of the above information, you can proceed to the actual installation of Linux on your machine. You should see that all of this is noted down on a piece of paper for ready reference as you go ahead with the process.

3.2.4 Installation

We will now see how to install Linux on your machine using the Linux bootable CD-ROM. As we have already observed, there are several ways of installing Linux. Here we are assuming a simple scenario where you are installing Linux from a Red Hat distribution from a CD-ROM on a computer with IDE drives. There is no other operating system running on the computer, so we do not have to take into consideration the issues involved in creating a dual boot system.

As a beginner to installation, you would prefer to use the graphical user interface with a mouse for your work. Of course, the keyboard can also be used for working with the GUI. Later, when you become more experienced, you can look at using a command line interface and performing more intricate installations. These can include using text mode installation, booting from a different source, controlling memory usage, using kernel options and so on.

To begin loading Linux, insert your bootable CD-ROM into the drive and turn on the f machine. Here you will need to make sure that your system is configured to boot from the CD-ROM. For this you might have to change your BIOS settings to alter the boot order. If this is taken care of, your machine will display a prompt.

Boot

At this point you can either do nothing and the boot process will start automatically after some time. Alternatively, to begin right away, you can press the ENTER key. If you do this, you would be ignoring the various boot options that you are presented with by the program. That is just what you would probably want to do as you learn about the process of loading Linux.

Once the booting process starts, you should find that your hardware is automatically detected. If this does not happen, you will have to do the installation in expert mode. But here we assume that things go well and you are able to proceed. You will now get the boot loader screen, where you should select the option to install from CD-ROM.

The system will first try to determine the type of your CD-ROM drive. If yours is an IDE type drive, the program should detect it. In case you have a SCSI drive, you will be asked to cho6'"sea SCSI driver. If your drive is an IDE but the program is not able to detect it, you will need to refer to the Linux installation documentation or seek expert help on tile matter.

If all goes well, you will come to a screen that says Language Selection. Here you have to specify the language that you would like to use during the installation process. There is a wide choice here, which includes Chinese, Japanese, French, Italian and many more. Let us assume that you want to work with English, in which case you should select it and then click on the Next button to continue.

The next screen that you get prompts you to select your keyboard layout type that you want to use. Here you would probably want to use the U.S. English layout. This is the QWERTY keyboard that we are familiar with and that has the $ sign over the number 4 in the top row. So select this layout using the mouse. You can also change your keyboard layout after the installation is over by using the keyboard configuration tool.

This brings us to the next screen, which is where we specify the kind of mouse that we have. Here there are many choices, and unlike the previous two cases, it is not at once apparent what type you are likely to have. The research on your hardware that you

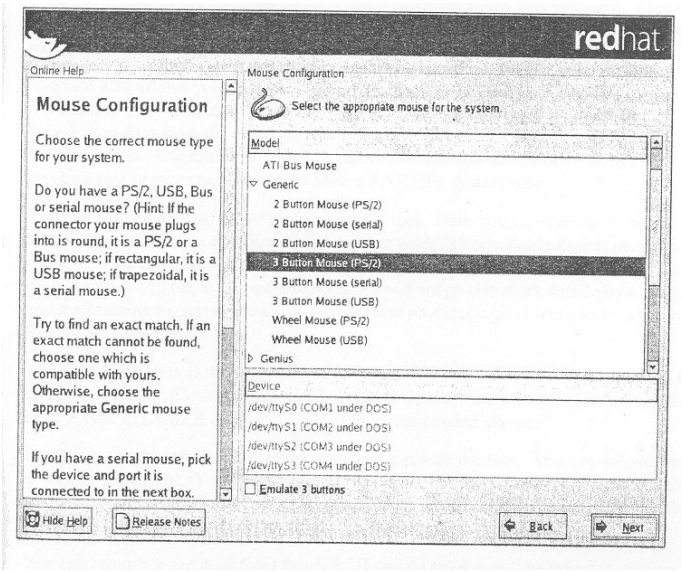


Figure 1: The Mouse Selection Screen

would have done in the previous section is what will now come in useful. You need to choose from among the many mouse types supported. Since you already know the kind of interface your mouse uses (PS/2, AT, serial, USB) and the number of buttons that it has, you will be easily to make the appropriate selection. *Figure 1* shows this screen to make things clearer for you.

What does one do when one cannot find an entry that seems to match our mouse? You can try the Generic type. If your mouse has a scroll wheel, you can make the selection as applicable, by choosing USB *or*

PS/2. Another selection that you have to make is whether to emulate three buttons when your mouse has only two. This can make it easier to use a graphical user interface. The emulation is done by considering the middle button to be substituted by both the left and right buttons. So if you press both of them simultaneously, it is equivalent to pressing the middle button.

After completing the installation, you can make your mouse left handed. This is more convenient for left handers as it corresponds to their natural hand. Here the functions of the left and right mouse buttons are interchanged. So the right button becomes the main button.

Finally, if your mouse is a serial mouse, you will have to specify the device *or* port (such as COMI *or* COM2) that your mouse connects to. Now having answered all the questions related to your mouse, you can click on the Next button and proceed to the next screen.

The installation program can detect a previous version of Linux that is already present on your system. In such a case, you will now come to a special screen that will prompt you and ask whether you want to upgrade your existing installation. Should you choose to upgrade, all your existing data will remain safe. None of your partitions will be changed and only the required files will be altered and overwritten. This option will work for Linux versions 6.2 and later. You will have the option to customise the installation and to choose the packages you want upgraded.

However, it might be that you simply want to throwaway your old installation and begin again from scratch. This time your data might *or* might not be preserved, depending on what kind of partitions you make in your fresh installation. If they are the same as the existing ones, your data will be safe. Should you choose a different partitioning scheme, then your data will be erased and will no longer be accessible after the process is over.

Once you have made your choice, you need to click on the "Next" button to move on to the next screen. This is where you choose your installation class *or* type. You already know about the differences between the Personal, Workstation, Server and Custom installations. If your machine will be used by developers in an office environment, the Workstation type might be best for you. After making your selection, go to the next part.

Now you come to an important screen where you have to partition your hard disks.

This can be a daunting task for a beginner, but since you area system administrator, it should be easy for you. Linux comes with a tool called Disk Druid to help you do this. Whenever, you perform an installation, you should be prepared for the possibility of losing all data on the disk. Therefore, it is important to back up all your important data before commencing the operation. In this discussion, we have not so far dwelt on the point because we assume you are installing on a new machine that does not have anything on it.

When using Disk Druid, you will have full control over the partitions. You will be able to choose the partition sizes and the types of the file systems you want to create on them. You will be able to decide where they will be mounted when Linux runs. However, if you do not want to go through this, you can ask the program to perform automatic partitioning. This does not give you much control but is a good option if you do not want to change anything *or* are doing your first Linux installation.

Even automatic partitioning will allow you to choose what kind of existing partitions are to be deleted from your disk. These choices are presented to you on the next screen. You can decide that you want to delete all old Linux partitions, *or* all existing partitions that might have other operating systems or file system types installed on them. Instead, you can also choose to preserve all existing partitions and use whatever free space is available on the hard disk. Since you might have more than one hard disk, the screen allows you to choose the one on which you want to create these partitions. All the partitioning choices will apply only to the disk that you have selected.

You can then review the partition selections that will apply and confirm or decide to modify t1tem. Here you will be working with the Disk Druid utility, as you would have had you chosen to partition manually. At this point you will be faced with some detail, as you have to decide where you want Linux to be installed. You would have already made these choices while preparing for the installation.

The Disk Druid shows you the hard disk and its model number at the top of the screen. You also see the number of cylinders, heads and sectors that collectively describe the drive's geometry. You should make sure that this is in tune with what you already know about your hard disk. The tool then shows the device file corresponding to the device, the point in the directory hierarchy at which it will be mounted, the type of the file system it will contain whether the partition will be formatted, its size and the start and end cylinders for the partition. This is shown for every partition.

Let us take a brief look at the different kinds of file systems that are available to you. A swap partition is used to increase the amount of virtual memory that your Linux system can use, as it supplements the RAM. Whenever required, pages are swapped from RAM to the swap partition on the disk and are loaded back when possible. Usually the size of the swap partition is made twice the RAM that you have. Linux has a file system called VFAT that is compatible with Microsoft's FAT file system and supports its long filenames.

The tool also allows you to create RAID partitions for redundancy that makes for more safety of your data, because if one partition develops an error, you can still access your data from the other. However, in this first shot at Linux installation, we will not venture into configuring RAID partitions. But this can be done by first creating two or more partitions that have a RAID file system type.

You can also create logical volumes using the tool. Here you can use one or more partitions on the same or different hard disks to work like one single logical partition or volume. This can allow you to have volumes that are larger than any single hard disk that you have. Again, in this introduction, we will not go into more detail about creating and maintaining logical volumes. You can create physical logical volumes by selecting that as your file system type.

The ext2 file system is the basic Unix file system and ext3 is the journaling version of ext2. Journaling allows you to quickly recover from system crashes. The default file system type in Linux is ext3, and is also the recommended choice.

Let us now look at the operations we can perform with the tool. You can add, remove or modify partitions. You also have a reset button that lets you restore the initial state of the disk, the situation that prevailed at the time you started the session. If you reset, the changes that you might have asked for in the session will be lost.

You can remove a partition from the disk. If you do so, you will be asked to confirm, as this has the potential to cause loss of data if it was an existing partition that was in use.

When you choose to add a partition, you have to provide information about it. The first thing is the mount point. To help you, there is a pull down menu that you can use to select the appropriate value here. For example, the boot partition should be mounted on /boot and the root partition on /. Then you need to enter the type of the file that you want the partition to contain. The next selection is where you specify the hard disks that can contain the partition. If you do not select a disk, then your

partition will not be created on that disk. You would usually let the tool decide where to place die partition.

You now need to enter the size of the partition. You can either enter a fixed size in MB, or choose to allow the partition grow from a base size to an upper limit as required. You can even choose to let the partition grow to fill up the entire available space on the disk. Next you can decide to let this be one of the primary,. or first four, partitions on the hard disk. If not, it would become a logical partition. The last option while adding a partition is specifying whether you want to check for bad blocks while formatting. While it is a good thing to do, checking for bad blocks can take a fair amount of time, especially with the large capacity hard disks of today.

Similarly; when you select a partition for editing, you can change its attributes such as the file system type or size. If the partition information has been already written to the disk, you can only change the mount point of the partition. Should you want to change other attributes, the only way is to remove the partition and create it again with the new attributes.

Once you have done partitioning your hard disks, you need to move ahead to the next installation screen that lets you install your boot loader. A boot loader is needed so that it can load Linux. Whenever the computer is started up, its BIOS load the program l that is installed on the Master Boot Record (MBR) of the hard disk. This program then loads the operating system. You can als01nstall the boot loader on the first sector of your boot partition, in which case the boot loader on the MBR will need to be asked to start up your boot loader. So if your disk has only Linux as the operating system, I' you should place your boot loader on the MBR.

Linux has two boot loaders called Grub and Lilo. Grub is powerful and can load Linux as well as other operating systems. Lilo is also a good boot loader and can boot Linux from several different sources. You will also have to choose which operating system to boot from by default, in case you have another already installed on your hard disk. Then when you boot your machine, you will have to choose the other manually if it is not the default.

You need to remember that if you do not install a boot loader, you will be able to boot Linux only through a boot diskette. So you should install one unless you have some special reason for not doing so. This could be because you already have your favourite boot loader, such as a third party commercial offering like Partition Magic, installed on the hard disk. After the boot loader installation screen, the next screen you reach

will depend on whether you have a network interface card on your computer. If so, you will reach the network screen. In an organisational set up, this is the most likely situation. So let us now look at the configuration to be done here.

If you were installing off a network, then you would have already used a network driver diskette to be able to work. But if you started installing using, say, a CD-ROM, then you can now configure your network cards. You have to select from the devices; that are detected during installation and decide whether they should be activated automatically at boot time. For each device, you need to edit its properties, namely, its IP address and netmask. If these will be supplied b);' a DHCP server, you can specify this here. These values will depend on what IS used in your organisation.

You can now enter the hostname and let it be detected automatically by a DHCP server. These decisions you would have already made as part of your pre-installation preparation. Now you also need to put in the IP addresses of the gateway for your network that connect it to the outside world and the IP addresses of up to 3 DNS servers for it. If you are using DHCP, these will be provided automatically and you need to enter this information only if you are providing the IP address of the machine manually.

It is quite all right to provide this information manually for one network interface card and to set these values for another card through DHCP. Once you are done with the network configuration you can select the Next button to move on to the next screen where you will set up your firewall.

In these days of networks, your computer is vulnerable to all manner of attacks by persons so inclined. A firewall offers protection against intruders over the network. Well configured, it reduces the chances of an attack greatly. You can choose from three levels of predetermined security policies. The first is the option of not using a firewall at all. While it might seem strange after what we have just said, this can be a good choice if you are sure your machine is going to be on a trusted network, such as a corporate network protected by other firewalls. Also, you might be planning to perform a more elaborate firewall configuration later.

The next level of security that you can choose is medium. This disables access from the outside to reserved ports, that is, up to port 1023. So services like HTT, FTP and so on will not work. Also barred are NFS servers and clients and X-Window display access to remote machines. The X font server is also disallowed.

The highest level of security bars all but DNS and DHCP. If the pre-configured firewall rules seem too restrictive or are not what you want, you can always perform additional customization. One thing that is possible is to declare some devices as trusted, whereupon no firewall checking will be done for them. You could do this on a multi-homed machine, where some or the networks are trusted. So if you are on the Internet through one interface and are connected to your company network through another, you can decide to declare the company network as trusted. You can also choose to allow access to additional ports beyond those allowed by your security level. Some of these are listed as service options that you can select through a checkbox, for instance mail and FTP. Besides, you can allow any arbitrary ports by using the notation.

port; protocol

such as 1111: tcp.

Having secured your system, you can select the Next button to move on to the succeeding screen where you can set your language options. Linux allows you to work in multiple natural languages. As you might imagine, you have to choose at least one language to work with. If you choose only one language, it is the default as well. 1fyou choose more than one language, you need to choose one language that will be your default.

The language that you chose to use for installation at the beginning of the installation process might not be the same as the language you choose to install for use thereafter. However, that is the language selected by default. If they are different, then after the installation is over, you will be able to use only that language. For example, if you use English (VSA) as the language for installation, and select French (France) as the language in this screen, then your installation will continue in English. Once the installation is over, you will be able to use only French.

You should choose all the languages that you might want to use during normal working. But it might be better to avoid choosing languages that you are not likely to use, because of the extra disk space that will be taken up by putting them in.

You are now at the tail end of the installation process. The next item to be configured is the time zone. Here you can choose to set your system clock to Universal Time Coordinated (VTC) that is based on the 00 longitude that passes through Greenwich. The time that you will then see will be based on UTC and the time zone that you enter.

To select the time zone, you have two methods. You can enter the offset from UTC for your location. So for India choose the option for +5:30, as Indian Standard Time is 5 hours 30 minutes ahead of UTC. For countries that use daylight saving time, you can set that too.

The second method is to set the time zone interactively, based on your location. You see a map of the world with many important cities marked out with yellow dots. The place you select will be shown with a red X.

Once the time zone has been set, it is time to set up your root password. This is an important password, as the root or superuser has complete control over the system. Restrictions and permission settings that apply to ordinary user accounts are no bar for the root user. So this password must be set with care. Make sure that it is not easy to guess. You have to enter the password twice and both the entries must match for the installation to move forward. If you are going to be administering this system, you must preserve the password carefully. But even then, do not use this account for ordinary work, because a small mistake as root could damage your installation. Use it only for administrative work on the machine.

The next screen is for setting up authentication information. You need not really do anything here unless you are going to set up network passwords. But you could choose the "Enable shadow passwords" option. This keeps your password more secure because the actual password is stored in the file /etc/shadow instead of /etc/passwd. The shadow file is readable only by the superuser, unlike the passwd file that is readable by all.

After this you come to another important screen where you configure the packages to be installed on your machine. This is where choosing an installation class is useful. Depending on what has been chosen, a list of package groups appears. You have the option of accepting the recommended packages for installation or of deciding on them yourself. Each group of packages can have optional packages that can be installed depending on your preference, besides base packages that are installed with that group by default.

When you choose to customise the packages to be installed, you have to select the checkbox next to the package group. Then you can click on the details link to obtain a list of all packages that constitute the group. Here you can select the individual packages that you want to have. After selecting the package groups, you can see all the packages that will be installed on your machine. This can be done in an alphabetical listing or as a tree structure under each group. Certain required packages needed to run Linux cannot be selected or deselected as they do not appear in

the package listings. They are always installed. Information about a package can be had by clicking on it.

What you need to be aware of is that some packages might be dependent on others in order to work properly. You need not worry about what these dependencies are as the install program finds that out automatically for you. After you have completed your package selection, you get a list of such uI1resolved dependencies. You are then given the option of resolving these dependencies by including the dependent packages, ignoring the dependencies or simply discarding the packages that have the dependencies. Remember that if you choose to ignore the dependencies, the packages with the dependencies might not work as expected. Of course, if there are no unresolved dependencies to start with, you will not see this screen.

You can see the amount of disk space that will be required by your selection, so if there are constraints you can adjust your package selection. However, in these days of large disks, it is not likely to be a problem.

Well, that has been a lot of choosing and decision making! Now it is time to sit back and let the machine do some work for a change. When you go to the next screen, you are at a point where you have to make one final decision about whether to go ahead with the installation as you have chosen. This is the last chance you have to abort safely, because if you go ahead here, the partition information will be written to disk and the installation will begin. Even so, to abort the installation, you need to reboot the computer.

Once you click on the Next button here, the install progran1 will start installing Linux and all the selected packages according to the options that you have provided. There is nothing for you to do but to wait and watch the installation proceed. To relieve the monotony and to reassure you that progress is indeed being made, you can see a bar and other screen messages that give the latest situation regarding the install. As you can imagine, the time taken here will depend on how fast your computer is and what packages you have chosen for installation.

Presently your installation will be completed. You will then be asked whether you want to create a boot diskette. You should create one, because it will enable you to boot should anything happen to your boot loader. You can create the diskette even after the installation is over and you have started working on it. For that matter, you can perform most of the steps in the installation such as configuring devices, language selection, package selection, configuration of services, firewalling and so on at any time even after the initial installation is over. The only

requirement is that you know the root password as only the superuser can perform such maintenance.

If you want, you can now configure the X-Window system server that is needed to get your graphical user interface. You have to choose the video card that your machine has from the list provided. If you know your card characteristics well, you can select the unlisted card and configure it, but in general, if your card does not appear in the program's list, it is not supported by the X-Window system. After selecting the card, you have to specify the amount of memory it has. Try to give the correct value and consult the card documentation if needed. While specifying more memory will not harm anything, it can mean that you have trouble with your X server.

The next screen lets you configure your monitor. You will be presented with a list of monitors from which you should choose your display, or the one closest to it. If your monitor does not appear on the list, choose an appropriate Generic monitor. Here you have to be careful that you do not select a monitor that has capabilities beyond yours. Your monitor can be damaged permanently if the selected monitor has horizontal or vertical synchronization frequencies beyond those of your display. Your monitor could get overclocked and be destroyed in such a case.

Now decide your colour depth and display resolution. You can also decide whether after booting you want to land up in graphics or text mode. In the latter case you will be presented with a command prompt from which you can issue any Linux commands that you wish. It might be better to set up the machine to get into graphics mode directly. Finally you can see your efforts bearing fruit. When you select the Next button on this screen, you will see the long awaited message that tells you the il1stallationis complete. You will now be asked to reboot the machine. You should remove your CD-ROM or diskette from which you had booted the machine for installation, otherwise the computer will boot again from that device and not from your hard disk.

3.3 Booting the System

Now you are ready to boot up the machine from your hard disk, using your freshly installed Linux. You can reset the computer or power it off and on again. After the machine's preliminary power on checks are over, you will come to a screen where you have the choice of booting the default operating system. Let us assume here that you have configured Linux to be your default system.

If you have another operating system such as Microsoft Windows 2000 loaded on your r machine, you would have the choice of selecting it to

boot from. If you choose to take no action, the default operating system will be booted up anyway after the timeout period has elapsed.

Once Linux starts to boot, you will see several checks being performed, at the end of I' which you are presented with the prompt to login. This assumes you have not chosen, to start up the graphical user interface, in which case you see a graphical login screen rather than a text based terminal.

When you boot up Linux for the first time there is some configuration you might like to do, such as setting the system date. You will be presented with a Setup Agent that will lead you on through such tasks.

Booting is so called because the job involves something akin to lifting oneself up by one's bootstraps, that is, from a system which is off you need to have a system running a complex operating system like Linux. This task is done in stages, with a very simple L program in the computer's ROM that runs and loads a boot loader program on say, the t hard disk, which is then able to load the whole of the Linux kernel.

Once the Linux kernel is loaded, it brings the system state to the run level that is specified in the file /etc/inittab. In this state you have the system in multiuser mode with NFS services available, if configured. This is the default state for a Linux system. If you want to perform some maintenance on the machine you would do it in single user mode that is run level 1. In this state only the console is available for logging in and other users cannot come in.

The tasks to be done when entering a particular run level are determined through a file called /etc/rc.d/rcn. d, where n is the run level. This gives a list of scripts that are run, and consists chiefly of services that are started up.

Together with booting up, you need to know how to shut down the system safely. When in operation, the file system is buffered in RAM, meaning that it is stored in the RAM and is then flushed to the disk fr9m time to time. Thus the file system on the hard disk is not always in synchronization with that in RAM. So abruptly powering off the computer can badly damage the file system, sometimes beyond repair. To power off a computer running Linux, use the shutdown command, which really is a user friendly front end to the init command.

You can look at the manual entry for shutdown to see all of its many options. There are only two that you are likely to use frequently. To initiate a shutdown immediately, say:

[root@root]# shutdown -h now

You can also give an absolute time in the form hh:mm, such as 23: 45 to shutdown the computer at that time. More likely you want to specify the amount of time to wait by specifying +m for the number of minutes you want to give as the grace period. The keyword now is equivalent to +0.

Sometimes you want to not just shutdown but to reboot as well. For this y the -r option to the command.

[root@linux root]# shutdown -r now

This command will perform all the necessary work to shutdown your machine in an orderly manner.

3.4 Maintaining User Accounts

You have seen in the second unit how users can login into the system and how passwords are used to prevent unauthorised access. You also saw how one or more users could be associated with a particular group. When you studied the long listing provided by tile -1 option to the 1s command, you saw how the user name and group were also given for each file in the listing.

You also know that an ordinary user cannot help you if you have forgotten your password, and that only the superuser call set somebody's password without already knowing what it is. One of the duties of the system administrator is in fact maintaining user accounts. This activity encompasses creating new accounts, deleting accounts of users who are no longer permitted to access the site and helping people who might have forgotten their passwords. On many large installations the system administrator will run regular checks on user passwords to make sure that they cannot be guessed easily.

The Linux utility for setting passwords does perform some checks for the quality of the password, but do not rely on them completely. Some of the messages it can give are:

BAD PASSWORD: it is based on your username
BAD PASSWORD: it is too short
BAD PASSWORD: it does not contain enough DIFFERENT characters
BAD PASSWORD: it is too simplistic/systematic

These give you an idea of the kind of checks that are done, but the system administrator can also help here by setting a minimum and maximum time between permitted password changes. The check for the

minimum time is based on tile premise that an intruder will want to change the password as soon as he gains access. The user can also be given a warning some time before the password is due to expire and passwords not changed for a given time even after they expire can be taken as inactive. You should choose passwords that are not easy to guess and should change them periodically.

With all this background, let us see just how user accounts are maintained. This information is kept in a file called /etc/passwd. This file looks like this:

[root@linux root]# cat /etc/passwd

root:x:0:0:root:/root:/bin/bash

bin:x:l:l:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

…

kumarr:x:500:500::/home/kumarr:/bin/bash

…

You should have been able to decipher some of this file. It contains a one line entry for every account. The different fields in this line are separated by a colon (:). The first field is the login name of the user. This is the name by which the user will be known on the system. The second field simply contains an x when you are using a shadow password file. Because the file is readable by everybody, placing the password, even though encrypted, in the shadow file gives you added security. The /etc/shadow file is readable only by root. So an intruder will not be able to easily read even the encrypted password. Look at the shadow file entries for the accounts shown above:

[root@linux root]# cat /etc/shadow

root:1oFtzPwDL$sw6qnXSqCTZeo5doxfpis0;12554:0:99999:7::

bin:*:12554:0:99999:7:: :

daemon:*:12554:0:99999:7:::

…

kumarr:1ByVJnCdk$HtTL8AIRNrX9hLj6dq0Ir.:12704:0:99999:7::

…

The first field is the user's login name followed by the encrypted password. It can be up to 34 characters long depending on the algorithm used. A. in the password field means a user cannot login using that name because it will not match any password. But the same effect can be achieved by locking the account using the -I option to the passwd command. This does not apply to the superuser because in his case the password check is not performed at all.

Coming back to the passwd file, the third field is the user identification number, or user id or uid, The Linux operating system works in terms of the user id rather than the name for most purposes. Any user with an id of 0 is a superuser, and ids below 100 are usually reserved for the use of Linux. Ordinary users get ids from 100 onwards. Now we will discover an interesting fact. Suppose you login as an ordinary user, kumarr; who has a user id of 500. Create a file /home/kumarr/checking, logout and login as root, and look its long listing.

[root@linux root] # cd /home/kumarr
[root@linux root]# 1s -1 checking

-rw-rw-r- 1 kumarr kumarr 665 Nov 29 21: 13
checking

Now edit the pa88wd file carefully and change kumarr in the first field to smith. Look at the listing of the file checking again.

[root@linux root]# 1s -1 checking
-rw-rw-r- 1 smith kumarr 665 Nov 29 21:13
checking

you will see that 1s now reports smith as the owner. This shows that the 1s command uses the passwd file to translate the user id to the user name. If you delete the account of kumarr by removing his entry in the passwd file, you will see

[root@linux root]# 1s -1 checking

rw-rw-r- 1 500 500 665 Nov 29 21:13 .
checking

This is because now 18 is notable to find out who the owner is and therefore reports e user id instead. Do put back kumarr's account under his own name now!

The fourth field IS the group identification number or group id or gid. The file /etc/group contains the group names corresponding to the group ids, and also contains information on which users belong to which

group. This file is also used by the 1s command to translate group ids to group names, just as it does with user names. The fifth field is a comment field containing up to 30 characters. The sixth held gives the home directory of the account and thus determines where he will reach then he logs in. The last field gives the shell he will be presented with. If the command there is not executable, the user will not be able to log in. The default shell /bin/ sh, used when the field is empty.

Managing User Accounts

You now need to know as a system administrator how to add new user accounts, remove or inactivate them and change their characteristics. This is quite simply done Linux. You need to go to the start button which is the red hat icon with an upward pointing arrow, usually to the left of your tray. Then choose the appropriate options as shown here.

cart -> System Settings -> Users and Groups

You are not the superuser, you will be prompted to enter the superuser password. You then reach the Red Hat User Manager screen. Here you will see a list of user accounts. This does not include system related accounts. You have buttons to add an account, at which you will have to enter the relevant information for the user such as his group and initial password. You also have an option to manage groups, where you can change the groups to which a user belongs.

To change the attributes of an account, you need to select it and click the Properties button. This brings up a User Properties screen where you have four tabs. Using these you can change all information about the user, including setting his password and changing the groups he belongs to.

You can also delete an account if you need to, or you can just lock it and disable access to the account by any ordinary user.

3.5 File Systems and Special Files

So far we have looked at two kinds of files, directories and ordinary files. Linux looks at everything as a file. So all hardware devices like terminals, tape drives, floppy drives, CD-ROM or DVD-ROM drives, printers and scanners are considered to be files in Linux. What this means is that there is a filename associated with each device and you can read from or write to these files just as you would to an ordinary file. There is a device driver in the kernel for every device supported and the name of the file for the device points to it within the system. Thus

when you perform an operation on such a device special file the device driver takes over.
There are two kinds of special files, character and block special, which are associated with character oriented devices and block oriented devices respectively. A tape is a character oriented device while a hard disk can be both character and block oriented. Thus you will find that both kinds of special files exist for your hard disk on the system, but that there are only character special files for any tape drives. Actually there are also pipe special files but we will not consider them here.

The bridge between the physical device name (the filename) and the driver for the device is located in the /dev directory. Let us look at a long listing of this directory.

[root@linux root]# cd /dev

[root@linux root]# 1s-1

total 228

crw------	1 root	root	10, 10 Jan 30 2003	adbmouse
crw-r--r--	1 root	root	10, 175 Jan 30 2003	agpgart
crw,-----	1root	root	10, 4 Jan 30 2003	amigamouse
crw------	1root.	root	10, 7 Jan 30 2003	amigamousel
crw------	1milind	root	10, 134Jan30 2003	apm_bios
drwxr-xr-x	2 root	root	4096 May 16 2004	ataraid
crw------	1root	root	10, 5 Jan 30 2003	atarimouse
crw------	1root	root	10, 3 Jan 30 2003	atibm
crw------	1 root	root	10, 3 Jan 30 2003	atimouse
crw------	1 milind	root	14, 4 Jan 30 2003	audio
crw------	1 milind	root	14, 20 Jan 30 2003	audiol

and many more such lines. Your listing might look somewhat different in terms of the devices and other fields here. But let us look at some of the main features of this listing.

The first thing you must have noticed is that the first character of the permission modes is no longer a hyphen (-), but is c or b. This indicates whether the device file is character or block special. The other permission modes have their usual meanings, except that you cannot execute a device no matter how disgusting its behaviour might be! So execute permission has no significance here.

Instead of the file size, you find two numbers separated by a comma. These are called the major and minor device numbers. A major number stands for a class of device like a terminal or a mouse, while the minor number gives the number of that kind of device. Thus different terminals might have minor device numbers 0 (for tty0), 1 (for ttyl) and soon.

An entry for a device can be added by the mknod command. Suppose your terminal major device number is 4 and you have 32 entries in the /dev directory from 0 to 31. You will not be able to activate another terminal even if the driver can support it unless the device entry is made

[root@root]# cd /dev

[root@root]# /bin/mknod tty 32 c 4 32

The arguments to the command are the device name by which It will be known, the type (c for character and b for block) followed by the major and minor numbers. A device entry can be removed as usual with the rm command.

The /dev directory in Linux is itself organised into several directories like / dev / raw for the hard disk as a raw device, /dev / ida for the hard disk as a block device and so on. This is for convenience because of the large number of devices that are supported. You must remember that merely making a device entry is not enough. The Linux kernel must have support for that device. If you purchase a new device like a DVD-ROM it will come with a driver that you can install on your system. This will put the device driver for your device in the kernel so that you are able to use the device. The device entry is thus a link to the device driver for the device.

We will now look at hard disks and see how they are utilised in a Linux system. A file system is the complete hierarchy of directories and files starting from its root. A small device like a floppy diskette cannot hold many files but a large (80 GB) hard disk can easily hold several file systems. It is common to partition a large physical hard disk into several logical disks, on each of which a file system can then be created. This is done by the mkfs command. You have already seen about partitioning disks while installing Linux on your machine.

When you create a file system on a partition, you should use up all the space in the partition because otherwise the extra space is simply wasted. The device is named as a character device before the file system is made. When it has a file system on it, it is used as a block device. Although you could still read a disk with a file system as a character device, it would be very difficult for you to interpret the data stored on it

unless you know intimately the structure imposed by the file system. Creating a file system is naturally done only in system maintenance mode.

What happens when a file system is created on a disk? The disk could previously be accessed only as a character device, but now a structure is imposed on this. You can now take advantage of the fact that you can access blocks on the disk randomly without having to go through all the previous blocks. So you can now look at the disk quickly. The file system structure consists of the boot block, the super block, the inodes and the data blocks.

The zeroeth block of every file system is reserved for storing booting information. It does not have any significance as far as the file system itself is concerned. It is the first block, called the super block, which contains information about the file system. Some of these items are the size of the file system, its name, the number of blocks kept apart for inodes, the list of in odes and the list of free blocks.

The index node or the inode blocks vary in number depending on the size of the file system and can actually be specified by the user while creating the file system. This number is the same as given in the super block. There is one inode for every file or directory in the file system and it contains information about the file such as the number of links to the file, the permission modes and the block numbers occupied by the file. The first 10 such numbers are called direct blocks because they directly hold information about the data blocks. The eleventh number holds a block address and that block holds the addresses of the actual blocks. There are more levels of indirection available so that you can store very large files in your file system. You cannot have more files and directories in the file system than the number of in odes. The remaining blocks in the file system contain the actual data in the files.

When the computer is booted, these file systems are not immediately accessible to users. This is because each file system has to be attached to the main file system. To do this you have to use the mount command.

[root@linux root]# /bin/mount fdevfhda4 front

The first argument to the command is the device file associated with that file system. The second argument is the name of a directory. The mount command associates the logical device containing the file system with the directory. Now /mnt becomes the root of the directory hierarchy on the device. So if you had a directory called khanz on /dev fhda4, you can now access it as /mnt/khanz. The directory /mnt should preferably be empty before you mount some file system onto it, because while the file

system is mounted you cannot access anything that was there on /mnt previously. To break the link between the file system and the directory say.

[root@linux root]# /bin/umount /dev/hda4

or simply

[root@linux root]# /bin/umount /mnt

Now you can no longer get at the files in that file system. The various file systems are usually mounted automatically when the system reaches the multiuser state through commands in the /etc/rc shell script. If you issue the mount command by itself, it will list all the file systems currently mounted and the directories to which they are attached.

If somebody is working on a file system, it would be disastrous to unmount it because the file system would then become inconsistent when the device suddenly vanished. That is why you cannot unmount a file system unless no user or program is accessing it. The umount command will tell you that the device is busy and will not take any action.

In the booting process, a special root file system is mounted on the root directory (/). The major system directories like bin, etc and dev are under this directory. Since this file system is always in use, you cannot unmount or mount it yourself. All the other file systems that you mount are below / in the hierarchy, as you know. Sometimes /tmp is also mounted as a separate file system.

You have seen the intricate structure of the file system. This structure is subject to disruption due to many reasons. For example a block on the free list might also appear attached to a file, or a block might appear on neither. An inode might appear duplicated or a file might seem to be under no directory. Such situations potentially mean the loss of data. Fortunately the file system contains a fair amount of redundant t information, and this enables a program to check whether it is consistent. If it is not, it is usually possible to repair the damage. The Linux system comes with a program called /sbin/fsck that performs a file system consistency check.

This program is run before mounting the various file systems because if the system is inconsistent to begin with it will get worse as users work on it. The fsck program looks for file systems to check in a file called /etc/ fstab. Also, the fsck command is actually a front end to file system specific consistency check programs.

The file system check is not done every time the system reboots and is performed only if the system was shutdown abruptly leaving the file system in an inconsistent state. However after a certain number of reboots the check is forced anyway. The ext 3 file system that is the default in Linux allows journalling, which means that the check, does not depend on the size of the file system but only on the size of the journal.

Every file system should contain a directory called lost + found, and this should be the first operation on it when the file system is made. Whenever fsck finds a file that is not linked to any directory, it places it in this directory. The program works in several phases and each phase performs some different check. You can set options to it that put it in interactive mode, where it expects a user response before taking any corrective action on a file system it is trying to repair.

3.6 Backups and Restoration

At a large installation, users do not usually have their own backups and the system administrator is responsible for the integrity of the system. On smaller installations and certainly on your own personal computer, you will be responsible for your own data and program files. So it is useful for every user to be able to backup data and know .how to restore it if needed. As a computer professional you do not need to be told how important backups are.

There can be many different backup strategies and tools that you can use. We can classify backups into full backups, incremental backups and differential backups. A full backup is a complete backup of an installation or a part of its file system. If there is a complete crash, you can restore user data from this. But the operating system configuration files are also something that one produces over a period of time and with considerable effort. So you can consider a backup scheme for them as well. How often you should take backups depends on how much you are prepared to lose.

A full backup is easy to restore from, but given the large disk capacities of today, the amount of time taken to make a full backup can be non-trivial. So you can think of a full backup at periodic intervals with daily or more frequent incremental backups, where only files that have changed since the last backup are copied. But it can be difficult to locate a needed file in an incremental backup, so the concept of differential backups was introduced. Here you backup all files that have changed since the last full backup. So the amount of backing up required is in between that of a full backup and an incremental backup.

It is important to consider what the backup media should be as well. Very often it is a tape of some kind, but a USB disk or a network backup to a data centre can also be considered. You have to look at the reliability of a backup and make sure that you do not use media beyond their useful life. A backup is pointless if you cannot restore from it when you need to. So make sure to test your backups periodically. You cannot afford a single failure in a backup. You can also consider taking multiple backups each time.

Depending on the criticality of your business, you might keep different copies of your backup at different locations. But whatever your company characteristics, you have to make sure you have at least one backup offsite that is geographically sufficiently distant from your site.

There are several different kinds of backup tools. Some are sophisticated commercial offerings while at the other end you have the basic Linux commands such as tar and cpio. Both of them work quite well and you can decide what you will use.

The tar command allows you to take a backup of all or selected files in a directory hierarchy onto tape, floppy disk or the hard disk itself. Tar knows about directories and links, and maintains headers, checksums and file permissions and owners. To take a backup of all files under /home/khanz

[khanz@linux khanz]$ tar cvbf 40/dev/rmt0/home/khanz

Let us look at the meanings of the various letters after the tar command. The c stands for create, and it causes tar to create the backup. Remember that anything previously present on the backup medium gets erased thereby (unless it is a file on the hard disk). The v is the verbose option of tar, and makes it chatter about what it is backing up. The b is the blocking factor and defaults to 20. You then specify the file or backup medium where the backup is to be taken followed by the directories to backup at the end. Everything under the directories you mention is backed up. You can also specify individual files if you want to.

To look at the contents of a tar file, you can use the t option.

[khanz@linux khanz]$ tar tvf /dev/rmt0

For extracting, you use the x option. One thing you should be careful about is the use of absolute pathnames during the backing up. When you restore from such a backup, it will restore to that same pathname. So it might be better to use relative path names while creating a backup

archive as you have the liberty of placing it wherever you want while restoring.

You can also use the z option to compress the archive, thereby saving a fair amount of space. The extent of the saving will depend on the kind of files that are being backed up.

Another useful command for creating and restoring from archives is cpio. It reads the list of files from the standard output and copies them to wherever you specify. For generating the list of filenames, you can use a program like find.

[khanz@linux khanz]$ find/home/khanz/| cpio-o>/dev/rmt0

You can use the many options of the find command to choose files that satisfy specific conditions so that only those files are backed up.

4.0 CONCLUSION

In this final unit of Module 2, you have been taken through Linux administration by starting with the basic definition of system administration and then on to the responsibilities of a system administration. It also taught you how to boot a Linux system, how to add and maintain user accounts and how you can backup data from the system and restore it when required.

5.0 SUMMARY

This has again been a long unit wherein we have tried to cover a lot of ground in a very small amount of space. You saw what the responsibilities of a system administrator were and how the exact composition of such a group depends on the organization characteristics. The major topic of this chapter was Linux installatiol1 and you saw how to perform a simple Linux installation on a machine. You also were introduced to file systems and how the system was booted, with some idea of run levels. As a system administrator, you would need to maintain user accounts and you saw how to add, delete or change user account information including their passwords. Finally we briefly looked at tile important topic oftaki11g backups and restori11g from them using the tar command.

This block would now have given you a fair idea of the Linux operating system al1d its capabilities. It should have prepared you to read up the large amount of documentation available on the subject and to explore the various resources devoted to Linux on the Internet.

6.0 TUTOR-MARKED ASSIGNMENT

1) List some of the duties of a system administrator.
2) Turn off the computer abruptly when you and other are working on it. See how a file system consistency check is performed. Is it possible that some data gets lost.
3) Find how to force a file system consistency check.
4) Remove the device file for a terminal and see what happens. Also try renaming the file and see the consequences.
5) Add a blank line to the beginning of the password file. What happens? What if there is a blank line in the middle?
6) Try umounting a file system when users are working on it. What happens and why?
7) You have backed up a directory containing several files onto 12 floppies using tar. When you try to restore these after a crash, the 4th floppy is found to be corrupted. How much data do you lose?
8) Why is it useful to take differential backups?

7.0 REFERENCES/FURTHER READINGS

There are a host of resources available for further reading on the subject of Red Hat Linux version 9.0.

httn://w.redhat.com/docs/manuals/linux

htt://w.linux.org gives among other information, a list of good books on Red Hat Linux.

Consider joining a good Linux mailing list.

MODULE 3 WINDOWS 2000

UNIT 1 WINDOWS 2000 NETWORKING

CONTENTS

1.0 INTRODUCTION

Windows 2000 is a network operating system with built-in support for peer-to- peer and client-server networking. The focus of a network operating system (NOS) is on use of remote services and resources existing inn networked computer system. In distributed operating system, the focus is on effective utilisation of resources in distributed computing environment.

Windows 2000 consists of 4 separate products.

Windows 2000 Professional
Windows 2000 Server
Windows 2000 Advanced Server
Windows 2000 Data Center Server

Following is a list of features for *Windows 2000* network support:

It has an integrated **support for network protocol** like TCP/IP and IPX/SPX.
It supports dial up networking (that facilitates mobile users to connect to a computer that is running on *Windows 2000* platform).
Windows2000 server incorporates **Microsoft's Internet Information Server (IIS)** that is a secure web server to host Internet.
It supports a set of security features which were not there in earlier versions of windows.

In this unit we will explore the issues related to networking support in Windows 2000- operating system.

2.0 OBJECTIVES

After going through this unit you should be able to:

describe Windows 2000 operating system architecture
describe peer-to-peer networking support in Windows 2000
describe Windows 2000 domains
identify protocols supported by Windows 2000
distinguish between FAT16 and FAT 32 file systems
how to shareholders m Windows 2000
describe Distributed File System
describe support of network printing in Windows 2000 environment.

3.0 MAIN CONTENT

3.1 Windows 2000 Operating System Architecture

CISC (Complex Instruction Set Computer) is a computer with a large number of instructions (complex) and constructs. Most of the NOS are based on CISC whereas early 80s designers recommended RISC (Reduced Instruction Set Computers) with simple instructions.
Windows 2000 is a portable operating system that is meant for CISC based machines (Complex Instruction Set Computing). CISC is a processor technology which is represented by a large set of instructions

with variable formats. Major processor families used in the design of modern computer system are RISC, superscalar VLIW (Very Large Instruction Word), superpipelined, vector and symbolic processors. Windows 2000 is always pre-emptive, which means that the high priority process gets executed first then compared to the low priority process. A complete Windows architecture is given in *Figure I.*

Windows 2000 system is made of layers: It works in two modes:

User Mode
Kernel Mode

User Mode is responsible for providing insulation of end users from kernel mode. *Windows 2000* user mode API subsystems are responsible for execution for different supporting system applications like win32 and POSIX. These subsystems have their own API's (Application Programming Interface) System data and hardware is accessible to kernel mode layer of *Windows200V.* Operating system itself runs in the kernel mode. Environment subsystems run in user mode. The lowest two layers nearest to the hardware use the kernel and Hardware Abstraction Layer (HAL) that is written in C and assembly language. Upper layers are written in C and are machine independent layers. Most of the drivers in Windows2000 are written in C or C++.

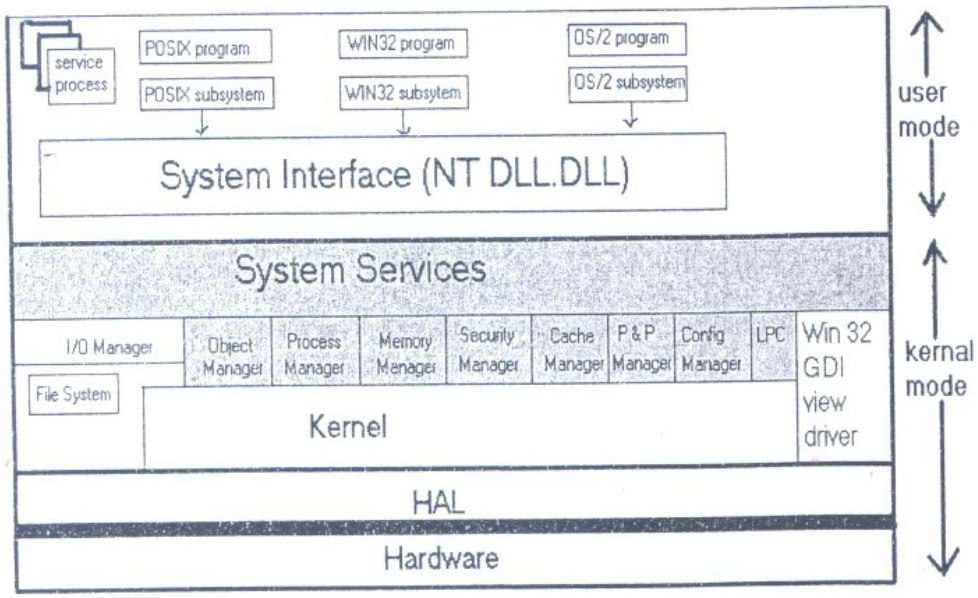


Figure 1: Windows 2000 layered architecture

HAL (Hardware Abstractor Layer)

The aim of HAL is to present the rest of the Operating System with an abstract view of hardware devices. It isolates the OS from platform specific H/W differences. The HAL makes each components such as bus system DMA controller, computer, interrupt controller, system timer & memory module look the same to the kernel.

Kernel

The aim of the kernel is to make the rest of the Operating System machine independent, hiding all the low-level details. Accessing the hardware using HAL Kernel is responsible for generating higher-level abstractions.

Kernel also includes the code for thread scheduling. It also provides low-level support to two internal objects -control objects and dispatcher objects. The shaded area is the executive. The entire executive area is written in C language and is architecturally independent and can be easily ported to machine.

It consists of the following objects:

Object Manager
I/O Manager
Process Manager
Memory Manager
Security Manager
Cache Manager
Windows/graphics manager
Local Procedure Call Manager.

1. **Object Manager:** Creates, manages, and deletes W2K Executive objects and abstract data types that are used to represent resources such as processes, threads, and synchronisation objects. It enforces uniform rules for retaining, naming, and setting the security of objects. The object manager also creates object handles, which consist of access control information and a pointer to the object. W2K objects are discussed later in this section.

2. **I/O Manager:** Provides a framework through which I/O devices are accessible to applications, and is responsible for dispatching to the appropriate device drivers for further processing. The I/O manager implements all the W2K I/O APIs and enforces security and naming for devices and file systems (using the object manager).

3. **Process Manager:** Creates and deletes objects and tracks process and thread objects.

4. **Memory Manager:** Maps virtual addresses in the process's address space to physical pages in the computer's memory.

5. **Security Manager:** Enforces access-validation and audit-generation rules. The W2K object-oriented model allows for a consistent and uniform view of security, right down to the fundamental entities that make up the Executive. Thus, W2K uses the same routines for access validation and for audit checks for all protected objects, including files, processes, address spaces and I/O devices.

6. **Cache Manager:** Improves the performance of file-based I/O by causing recently referenced disk data to reside in main memory for quick access, and by deferring disk writes by holding the updates in memory for a short time before sending them to the disk.

7. **Windows/Graphic Manager:** Creates window oriented screen interface and manages the graphic device.

8. **Local Procedure Call Manager:** Enforces a client/server relationship within a subsystem in a manner similar to remote procedure call facility used for distributed application.

Minimum Hardware Requirements for *Windows 2000* are 32-bit Pentium .133 MHz processor, 128 MB RAM, 500 MB or more of disk space to setup Windows 2000.

3.1.1 Peer-To-Peer Network

MS Windows 2000 is an ideal Operating System for peer-ta-peer networking. In a peer-ta-peer network, computers work independently, providing various services like:

Each computer can have its own separate user accounts.
Sharing of resources (folders, printers etc.) is possible.
Each computer IS responsible for managing its security.
Easy set up for the network.

On a peer-to-peer network, workstations communicate with one another through their own operating systems. Files, folders, printers, and the contents of entire disk drives can be made available on one computer for others to access.

Here is a simple peer-to-peer network *(Figure* 2)

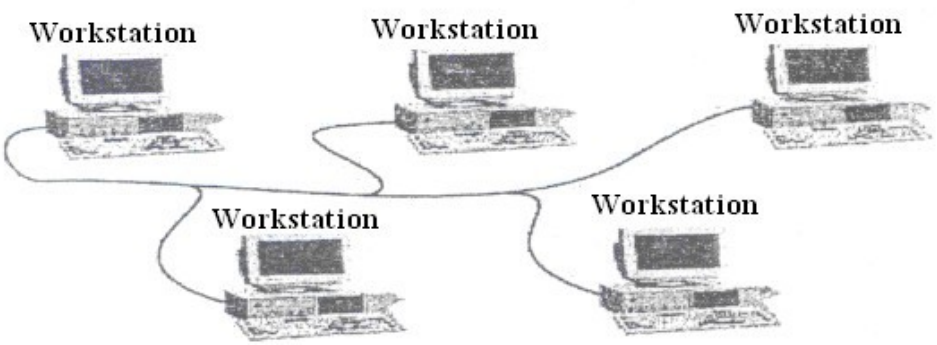


Figure 2: Peer-to-peer Network

3.1.2 Domains

A domain is a collection of accounts representing network computer uses, and group of users all maintained in a control security database for care of administration.

In Windows 2000, *domain* is a collection of computers where a server computer referred to as a *Domain controller* is responsible for the management of security for the entire network. This type of logical grouping is desirable for corporate application. Computers of a domain network have local user accounts, but are dependent on a centralised information store called as Active Directory Service. Thus Active Directory in Windows 2000 provides a centralised control.

Domains add several interesting features to Windows 2000 functionality.

Centralised storage of user information.

Each domain has domain controller associated with it. In Windows NT, domain controllers are either BDC or primary domain controller. In Windows 2000 there is only one type of domain controller.

Extension of the existing network becomes easy.

In Windows 2000 Active Directory unites namespace of internet with window NT directory services since Windows 2000 domain naming uses DNS (Domain Name System).

What is DNS, conceptually, the internet is divided into several domains (e.g., gov, edu, com, net, etc.), where each domain covers many hosts. Each domain is partitioned into several domains and these are further

partitioned. The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing the naming scheme. It is primarily used for mapp1ng host names and e-mail destinations to IP addresses.

While creating a Windows 2000 domain, the DNS should be executing and properly configured on the corresponding machine. If in case, DNS is not running, on creation of a domain controller, it is automatically installed later. Thus domain provides Windows 2000 with a grouping mechanism where not only accounts but also network resources are grouped under a single domain name.

Joining a Domain

Windows 2000 has **"Join a Computer to the Domain"** pem1ission for those computers that wish to be a part of Domain. By obtaining this permission, an account is created for that computer. It is like a class of objects, where all the objects of that class are of the same type. The objects type may vary from users to computers. Active Directory Service provides a hierarchy to various resources stored in domain. A Domain has information about the objects it contains. It provides the network with a secure boundary.

3.1.3 Network Protocols

Protocol refers to a set of rules that facilitate communication across a network. A network application does not directly interact with the underlying network hardware; rather it interacts with protocol software that follows the rules of a protocol.

Windows 2000 include support for following different protocols through various layers:

1. TCP/IP
2. IPX/SPX
3. Net BIOS Enhanced user Interface (NETBEUI)
4. Data link control (DLC)
5. ATM (Asynchronous Transfer Mode)
6. AppleTalk
7. Infrared Data Association (IrDA).

Default protocol in Windows 2000 is TCP/IP. Since TCP/IP is the universal protocol of the Internet, thus enabling access to Internet resources. TCP/IP facilitates communication over a network that is otherwise a collection of computers with different architecture and

operating systems. Two commonly used Windows 2000 troubleshooting utilities for TCP/IP are Ping and IpConfig.

The **advantages** of TCP/IP are:

Designed for routing. (IP) and end-to-end data delivery (TCP)
Is the most used protocol of Internet
Compatible with standard networking tools.
Facilitates communication among diverse networks and network operating systems.
Enables the use of DHCP and WINS.
Compatible with Microsoft Windows Sockets.

AIM is a protocol that is able to provide voice, data and video services across wide area networks.

NWLink is MS equivalent of Novell Netware.

IPX/SPX (Internet packet exchange/sequenced packet exchange)

Only TCP/IP is accessible to Windows 2000 networking running Active Directory Services, not NWlink or NetBEUI.

NetBEUI is a kind of legacy protocol that is used to provide accessibility with existing network (small network) that is already using NT.

Appletalk: Apple Computer Corporation developed this protocol suite and is included in Windows 2000 so as to provide compatibility with Apple Machinist clients. In addition because of Appletalk Windows 2000 CJU1 function as a router and a dial up server.

DLC (Data Link Control) protocol originally developed for IBM mainframes is required for printers and other peripheral devices installed on a network.

Ir DA is a collection of bi-directional wireless infrared based protocols (that spans a short range). It facilitates communication among multiple device types like camera, printers and PCs.

IP Addressing

The IP address format is called the dotted decimal notation address. It is 32 bits long and contains four fields, consisting of decimal values representing 8-bit binary octets. For example, an IP address might be 198.60.204.2.

A unicast transmission is one in which one packet is sent from a server to each client that requests a file or an application.

A multicast means that the server is able to treat all the clients as a group and send one packet per transmission that reaches all the clients. It saves the bandwidth of a channel.

A **subnet mask** is used to divide a network into subnetworks to meet addressing requirements with limited availability of address.

Static and Dynamic Addressing

Each server and workstation needs a unique IP address, either specified at the computer or obtained from a server that assigns temporary IP addresses.

Static addressing involves assigning a dotted decimal address that is each workstation's permanent, unique IP address.

Dynamic addressing automatically assigns an IP address to a computer each time it is logged on.

Dynamic addressing method uses the **Dynamic Host Configuration Protocol (DHCP),** which is supported by Windows 2000 Server for dynamic addressing. It provides an enhancement to TCP/IP. DHCP in addition to permanent addresses assigned to computers that run a server it automatically allocates an address. Yet a DHCP does not assign an address permanently, rather it specifies a lease for the address use.

In Windows 2000, configuring TCP/IP using DHCP has many advantages. On Windows 2000 servers that provide Internet communication, when one is configured as a DHCP server, Windows Internet Naming Service (WINS) is also installed so that the Windows 2000 Server is both a DHCP and a WINS server.

Domain Name System provides automated mapping between computer names and IF address. Conversion of a domain name into an equivalent IP address is referred to as name resolution, and domain name is said to be resolved to an address.

3.1.3 File Services

Windows 2000 provides read and write support for NTFS, FAT 16 and FAT 32 file systems. FAT is designed for small disks and simple folder structure. Windows 2000 supports both FAT 16 and FAT 32 file system and FAT is designed for small disks and simple folder structure.
A FAT 16 partition is divided into 512 byte sectors and disks have files in dusters in the default cluster size dependent on partition size and can range from 8 sectors to 128 sectors. FAT 32 can support partition up to

2047 GB in size. The major advantage of FAT 32 over FAT 16 is larger partition sizes.

NTFS (NT File System)

Windows2000 supports a new version of NIFS, i.e., NTFS version 5.0. This new version of NTFS is better than in terms of reliability and better performance.

NTFS 5.0 includes the following features:

All of the new features of Windows 2000 Active Directory Services.
Storage features like reparse points.
Features for Software Management.
Enhanced security features for servers, which provides an authentication mechanism to users before they can actually gain access to network resources.
It supports CDfs;

The fundamental unit of disk allocation in NTFS is cluster that comprises multiple sectors.

Disk Storage Types:

In Windows 2000 two kinds of disk storage are possible:

Basic Storage
Dynamic storage.

Disk should be initialised with a storage type before data could be stored on it. Either of the two storage types can be used on one disk. But in a system with multiple disks both storage types can be used. Basic disk storage is the default storage type for Windows 2000. All disks are basic until converted to dynamic. Disks can be managed on local and remote networks. Only Windows 2000 has support for DYI1arnjc storage, which can be resized unlike basic storage type.

Basic disk is divided into partitions. Disk partition can be primary or extended and they function as disks in their own entirety.

Dynamic disk is divided into volumes. Volumes can be simple, spanned, mirrored, striped or RAID-5. Only computers running Windows 2000 can access dynamic disks.

File Replication Service (FRS)

Another file service feature supported by Windows 2000 is File Replication Service (FRS). It is so configured that it automatically starts on all domain controllers and manually on all standalone sectors. Its automatic file replication service is responsible for the copying and maintenance of files across network.

Two kinds of replications are possible:

Intrasite Replication
Intersite Replication.

Sites are subnets comprising well-connected computers. Any portion of the network, subnet, is a site.

3.1.4 Shared Folders

The mechanism by which resources across a network are accessible is referred to as sharing. Only those users who have been granted access to the shared folders can use - files of a shared folder. By default any user who logs on to a computer has access to the shared folders on that computer.

A shared folder data may range from personal to corporate data. A shared folder Permissions may vary depending upon the kind of data a folder contains read, change, and full control permissions.

Shared folders permissions exhibit the following features:

They provide a security boundary not detailed security, since shared folders permissions hold true for the entire folder and not to individual files.
On a FAT system it is the only way **to secure network**.
Full control is the default permission for a shared folder.

Permissions for a shared folder may be granted or denied to users or to groups. Also if a user is denied permission for a shared folder then even if she/he is member of a group that is granted shared access permissions for the folder, she/he cannot access the folder.

Sharing a folder

When a folder is shared it can be given a share name, comments can be added to it for the description of the contents of the folder etc.

3.1.5 Distributed File System

Another Windows 2000 file service is *Distributed File System;*
It is an efficient and easy way to access shared folders across the network; Files are arranged in a hierarchy in DFs. It is a logical tree structure, comprising DFs root and DFs links. In DFs resources from various locations, servers are shared in DFs root.

Features of Windows 2000 DFs:

Facilitates network administration
Simplifies network navigation
Provides a hierarchical logical organisation for shared folders across different computers on a network.

Two types of DFs roots can be implemented on Windows 2000 Servers:

Standalone DFs roots
Domain DFs roots.

In standalone DFs roots, DFs is stored on a single computer. It has no support for fault tolerance in case the computer that stores the DFs topology fails. Domain DFs root writes the DFs topology to Active Directory. It supports file duplication in case *of* failure. Here DFs links point to multiple copies of the same shared folder. When changes are made to a DFs link that is a part of a domain DFs root, the changes are automatically reflected to other members also.

3.1.6 Print Services

Windows 2000 has support for networking printing. Thus, it facilitates printing *from* any computer in the network. Also printer can be managed from any computer by having just a web browser installed on that computer. Using *Windows 2000* various components with different Operating Systems/platforms can send jobs for printing.

For network printing basic requirements are:

Sufficient memory (RAM)
Sufficient disk space
A server computer.

Remote network printing, non-remote local printing and non-remote network printing is supported by *Windows 2000* networking printing. TCP/IP is the default network .protocol for *Windows 2000* in use by many network-printing devices. Printers on a network can be shared if

printing jobs are more on the network that an unshared printer is unable to handle.

In order to share an unshared printer on a network

In the Printers Windows.

Click the properties dialog box and then click on sharing tab.

This sharing tab acts as an interface for sharing a printer, on the network.

Managing Printing Jobs

Windows 2000 facilitates job management that primarily involves restarting, resuming, pausing and canceling printing jobs if a problem arises while printing.

Another interesting feature in Windows 2000 is that the user manages print job by setting printing priorities and printing time, provided the user has been granted manage Documents permission for the desired printer.

In a network *Windows 2000* facilitates managing network printers even with Web browsers. Thus eliminating the need for having installed *Windows 2000* on every computer.

Role of a Printer Driver

In a network some computers cannot access the printer installed over the network. This is due to the fact that printer may by attached to a computer that is not having *Windows 2000* installed on it.

Since *Windows 2000* has all the required printer drivers installed within it and printer drivers are responsible for the creation of special printer file that carries requisite information the printer needs. *Windows 2000* always keep the drivers up-to-date.

3.2 Using the Mapped Drive

Windows 2000 allows the user to assign a drive letter to a share network resource **that may be a printer, folder or a drive using the mapped drive. A file server or workstation shares a mapped folder or drive on the network.**

By default, Windows attempts to reconnect any mapped drives the next time user logs on. If you do not want this to happen, click to clear the Reconnect at Logon check box.

By default, you are connected to the other computer with the logon details that you are currently using. If you want to use other credentials, click Connect using a different user name, and then type the appropriate user name and password to connect to this network resource.

The mapped drive that we create is visible in the Folders, in Windows Explorer, along with all the other drives on our computer. Files in the shared folder can be accessed with any program on our computer by using the mapped drive letter.

To Assign (Map) a Drive Letter to a Network Computer or Folder

1. Click **Start,** point to **Programs**, and then click **Windows Explorer**.
2. On the **Tools** menu, click **Map Network Drive**.
3. In Path, type the path to the resource you want. For example: \\computername\foldernarne.
   If a password is required, Windows prompts you.

Notes

You can also right-click **My Computer** or **Network Neighbourhood**, and then click **Map Network Drive**.
To map to a computer or folder you have used recently, click the arrow to the right of **Path**, and then click the resource you want

3.2.1 Printing a Mapped Drive

Once the letter has been assigned to a drive, after selecting a file from the drive, pull down the File menu, choose the Print option. Also by right click any document icon and choose Print.

3.2.2 Disconnecting a Mapped Drive

To disconnect a mapped drive

Click **Start**, point to Programs, and then click Windows Explorer.

1. On the Tools menu, click Disconnect Network Drive.
2. In Drive, click the resource that you want to remove, and then click OK.

Note

You can also right-click My Computer or Network Neighbourhood, and then click Disconnect Network Drive.

3.2.3 Viewing Directory Information

From the My Computer window:

1. In order to view the contents of a drive double click on a "drive icon.
2. Then select a folder within that drive and double click on it and keep moving down till the desired folder is found.

3.2.4 Creating a Shared Folder

To specify a path

1. Type the drive letter followed by a colon (:) and back slash (\). See the examples in Note.

2. Type the names of the folders and subfolders that contain the file, typing backslashes before each folder name.

3. Type the name of the file. A backslash should precede the file name.

If you use file names that contain spaces or exceed eight characters in length, enclose the path in quotation marks.

Note

You can specify a path from within a program, from Run, or from the MS-DOS prompt:

To specify the location of Disk Defragmenter, which is located on drive C in the Windows folder, type:
c:\windows\defrag.exe
To specify the location of a document named List.doc, which is located in the II folder within the Events folder on drive C, type:
c:\events\ll \list.doc
To specify the location of a bitmap named Canyon, which is located in a shared folder named Scenic on a computer named Pictures, type:
.\\pictures\scenic\canyon.bmp

Or, map the shared folder to a drive (for example, drive D), and then type: d:\canyon.bmp

3.2.5 Logging off a Client

When one is finished working on a shared computer on a network, or when one wishes to log as another user:

1. Press Ctrl+ Alt+Del and choose log off option
2. Click Yes when asked if currently an application is running, thus giving the user -an opportunity to save any open files.

When the entire process is complete, the machine is available for a new logon.

Note: Windows 2000 may restrict a user from logging on even if the user is entering: the correct password.

Such a situation arises when the user has two accounts with the same name but with different passwords one on the network and other on the local computer. And the user may have selected the wrong location.

The *solution* to this problem is that click on the option button to make sure the correct location is selected in the log on list.

3.3 A Few Important Facts about Windows 2000 Usages

1. If your computer is on a local network but you have a local account on your computer that gives you permission to make changes, log off from the network and then log on again to the local computer. Enter your user name and password for the local computer account, and enter the computer name in the Domain box. You would be able to make changes to the computer that you were not allowed to make when you were connected to the network.

2. Folder windows are the gateways to your files and documents. Users' folder windows can display all information and these windows can be customised.

3. Windows 2000 uses a single logon system -when you logon to a domain using an authorised user name and password you unlock access to all resources on the network.

4. The difference between logging-off and locking the computer is that Logging- off closes all progran1rne and data files. In order to resume work you need to 'logon again and restart the entire programme. By locking the computers, however, you keep

running programme and memory. When you return, by entering your password, you can resume work.

5. It is not possible to change the letters assigned to the drive that contain systems files or boot files. Also assigning the drive path instead of a letter works only if two conditions are true. First, the drive that contains the path you want to use must be formatted with NTFS, FAT 16 and FAT 32 will not work. Second, the folder path must be empty.

6. If your computer is part of domain, any member of the domain Administrator group is a member of Administrator group on your computer automatically.

4.0 CONCLUSION

In this first unit of module 3, has taken you through the architecture of windows 2000 OS and the different types of networking protocols. It has also described to you the concepts of distributed file system and how a network printer can be managed.

5.0 SUMMARY

Windows 2000 consists of a family of four products namely Windows 2000 Professional, Windows 2000 Server, Windows 2000 Advanced Server, and Windows 2000 Data Center Server. It is an object based operating system, supports networking and provides centralised control of data. It supports both types of networks: workgroups and domains. An important feature of Window§ 2000 is its Active Directory Service that not only allows removing, adding or relocating users and resources but also completely segregates the physical structure of domain from its logical structure. And thus presents a layer of abstraction. Windows 2000 includes support for following different protocols TCP/IP, IPX/SPX, Net BIOS Enhanced user Interface (NETBEUI), Data link control (DLC), ATM (Asynchronous Transfer Mode), AppleTalk, Infrared Data Association (IrDA), Default protocol in Windows 2000 is TCP/IP. Since TCP/IP is the universal protocol of the Internet, thus enabling access to Internet resources. TCP/IP allows communication over a network that is otherwise a collection of computers with different architecture and operating systems. Two commonly used troubleshooting Windows 2000 utilities for TCP/IP are Ping and IpConfig. Windows 2000 supports two versions of FAT file systems: FAT 16 and FAT 32.It also supports NTFS. FAT originally was designed for small disks; FAT 16 can support partition up to 4 GB in size, while FAT 32 can support up to 2047 GB size partitions. The process by which resources across a network are accessible is referred to

as sharing. Windows 2000 has support for shared folders. A shared folder data may range from personal to corporate data. Shared folders Permissions vary depending upon the kind of data a folder contains read, change, and full control permissions.

Distributed file system is an advanced file service in Windows 2000. Files are arranged in a tree hierarchy in DFs. It is a logical tree structure, comprising DFs root and DFs links. In DFs resources from various locations servers are shared in DFs root. Two types of DFs roots exist -standalone DFs root and domain DFs root. Windows 2000 has support for networking printing. It supports printing from any computer in the network. Printer can be managed from any computer by having just a web browser installed on that computer. Windows 2000 being a network operating system supports sharing of all the resources across the network.

Heads On

1 Your campus has installed an additional network and you are the network administrator for the new network. Your job is to configure the DNS naming scheme for both the networks new and the existing one. Both the networks are installed in different departments and operate independently but there needs to be a communication between the two. How should the DNS be configured between the two networks?

2. Try CMD rather than COMMAND to open Windows 2000 command line arguments.

3. Just like shared files and folders, try to hide the shared printer on Windows 2000 environment.

4. What will happen if instead of using a screen saver you try to lock your computer?

6.0 TUTOR-MARKED ASSIGNMENT

1) What is the purpose of a directory service in Windows 2000?
2) In what mode does the console run?
3) How does a domain differ from a workgroup?
4) In a multi-user environment while printing, how can the possibility of a user ending up with a wrong document be avoided?
5) Can a single document be redirected on a network?
6) When do DCHP clients try to renew their leases?

7) Can moving and copying files and folders between disk volumes change their compression state?
8) What type of data is replicated by FRS?
9) What is the default permission when a partition is formulated with NIPS?

7.0 REFERENCES/FURTHER READINGS

www.microsoft,comwebsiteforadetaileddescription of Windows2000 Environment.

White Paper for Distributed File Systems at www.microsoft.com.

"Operating System Concepts", Silberschartz, Galvin and Gagne, Sixth Edition, John Wiley & Sons.

UNIT 2 MANAGING WINDOWS 2000 SERVER

CONTENTS

In the previous unit we examined the structure and basic networking support of Windows 2000. In this unit we will explain how to manage Windows 2000 server.

By default Windows 2000 restricts most system management features to specially privileged users called administrators. Unlike Windows 95 and Windows 98, most of the management tasks cannot be performed until the user is logged on using the administrator's account. As a part of the setup process, Windows 2000 creates a built- in account called Administrator and requires that the user may enter a password for that. Windows 2000 also creates a built-in group called administrator. Any user who is a member of this group can perform management tasks as well.

2.0 OBJECTIVES

After studying this unit you should be able to:

describe Windows 2000 client & Server architecture
log onto the network
browse network resources
access network resources using My Network Places
map a drive letter to a network resource (a folder or a shared folder)
use Windows explorer.

3.0 MAIN CONTENT

3.1 Using Windows 2000 Server and Client

Microsoft Windows 2000 Server is a more robust network operating system than Windows 95 or 98. A server is a single computer that provides extensive multi-user access to network resources.

Here is a diagram of a server-based network *(Figure* 1).

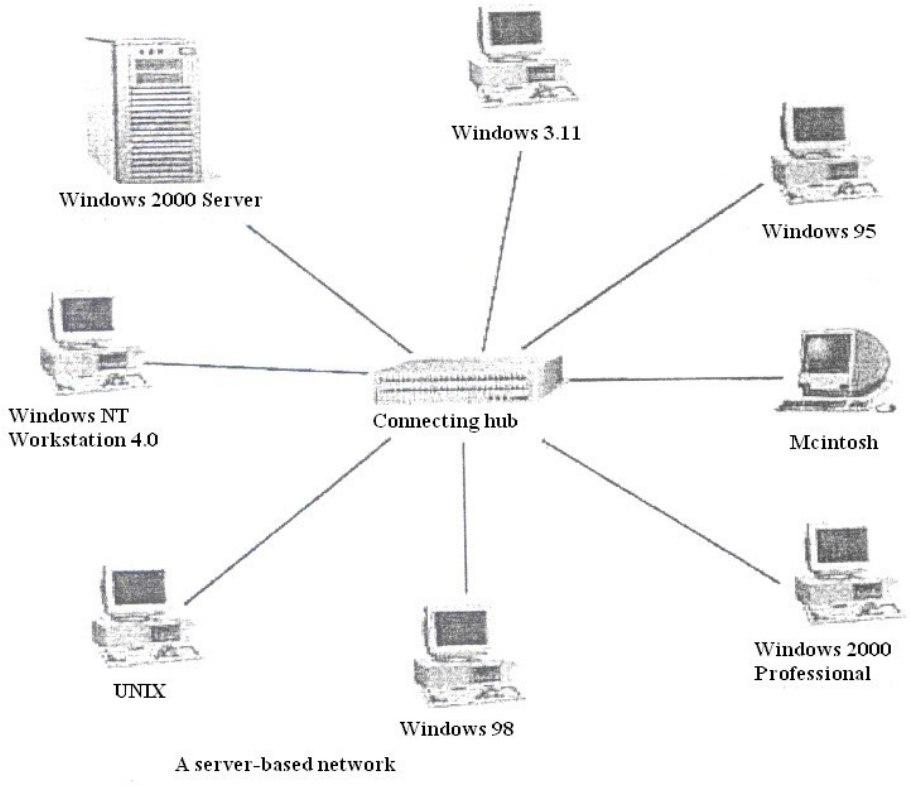


Figure 1: A Server-based network

Windows 2000 Server can provide the following advantages:

Sharing of files among member computers.
Sharing of printers and other resources.
Centralised control and administration of resources.
The server administrator can save time when installing software upgrades.
Software applications can be shared among members of a client sever network.
All computers can be backed up more easily.

Windows 2000 Server and Windows 2000 Professional Compared

The basic server version is called Windows 2000 Server, and Windows 2000 Professional is designed for workstations.

Windows 2000 Server offer services including:

Virtually unlimited numbers of users simultaneously (optimally for 10 users).
Active directory management.
Effective network management.
Web-based management services.
Network-wide security management.
Remote network access.
Application services management.
Network printer management through the Active directory.

3.2 Logging onto the Network

In Windows 2000 environment a user can log on either as a **local user** or a **domain user.**

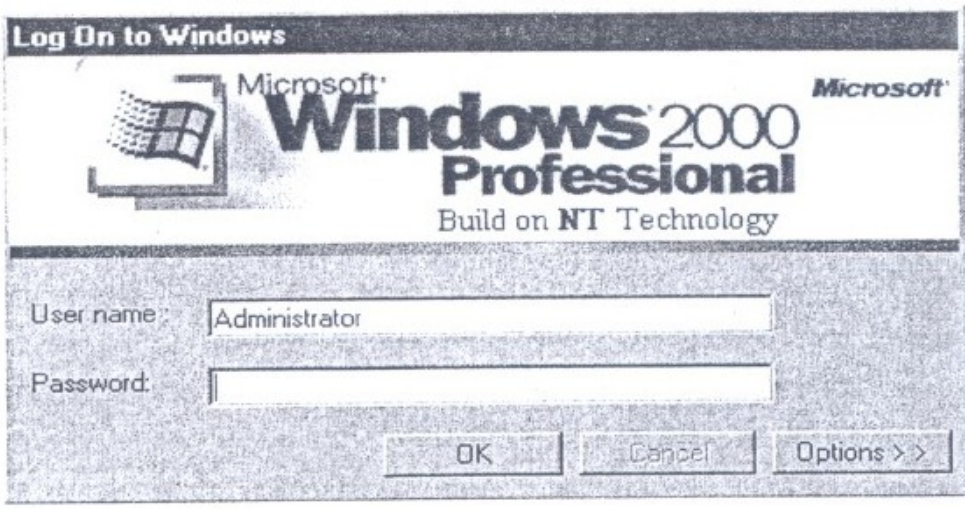


Figure 2: Logging on the Window Screen

Domain User Account permits the user to log on to the domain and allows him to access resources on the network whereas a local user can log on to a local computer to be able to access resources on that machine.

Local User Account: Local user accounts are not replicated to domain controller; rather they are created in local machine security database.

By default a user has access to a domain via any other computer in that domain if it is a domain member. Then there are groups, which is a collection of user accounts. Individual users can be members of more than one group.

Windows 2000 supports two types of groups:

Security Groups
Distribution Groups.

Security groups are responsible for assigning access permission for resources.

Distribution groups are used for non-security related functions.

Now the actual log on procedure to enter the domain:

Windows 2000 by default assumes that the user wishes to log on as a local user. However in a networked environment in an organisational set up it asks for both user name as well as domain name.

In the Log On to Windows Dialog box as shown in *Figure* 2 in the user name box type

Username_+@domainname :

Example: userl@)domain2

Where user I is the user name domain2 is the domain name as shown in *Figure* 3 and *Figure* 4 click on it. An expanded dialog box appears and then choose the domain from log on to: list box but remember if user name is entered with @ symbol in the user name box then options box will be grayed out at the end when the user wants to log on as another user press ctrl+aIt+del and choose log of option then logging off would be conflrD1ed by displaying the yes. Click on it to confirm logging off. In the process windows shuts down all applications that are currently executing. After this the machine is available for log on. Windows 2000 has a built-in administrator account.

The following windows *Figure* 3 and *Figure* 4 adds new user, **user1** to the existing network:

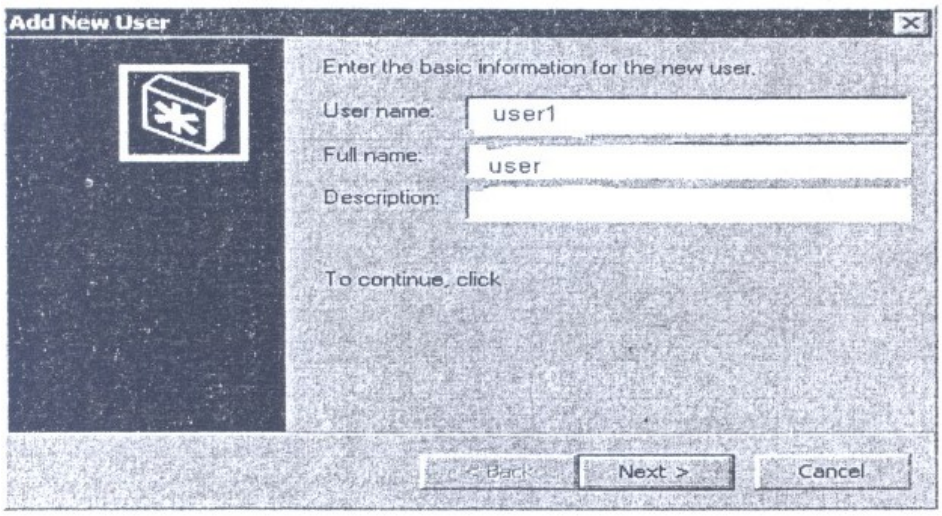


Figure 3: User Basic Information Screen

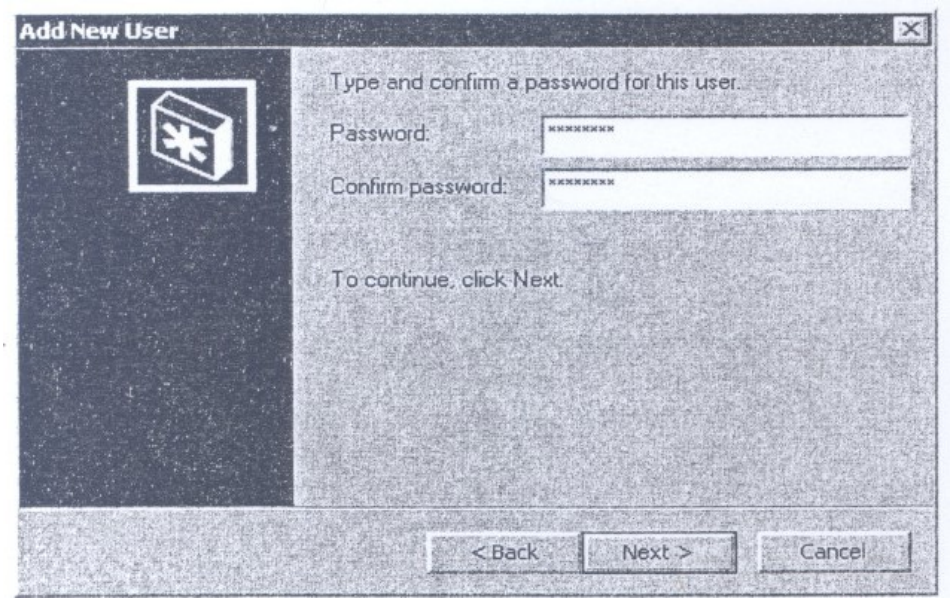


Figure 4: Add User Password Screen

Windows 2000 also supports built-in group accounts.

1. While creating a new user account with the wizard's help, select the Standard user option. With this option the user's account becomes a member of the power user group. Users of this group can participate in installation as shown in *Figure* 5 and *figure* 6.

2. If the user selects Restricted User option, user account becomes member of built-in user's group.

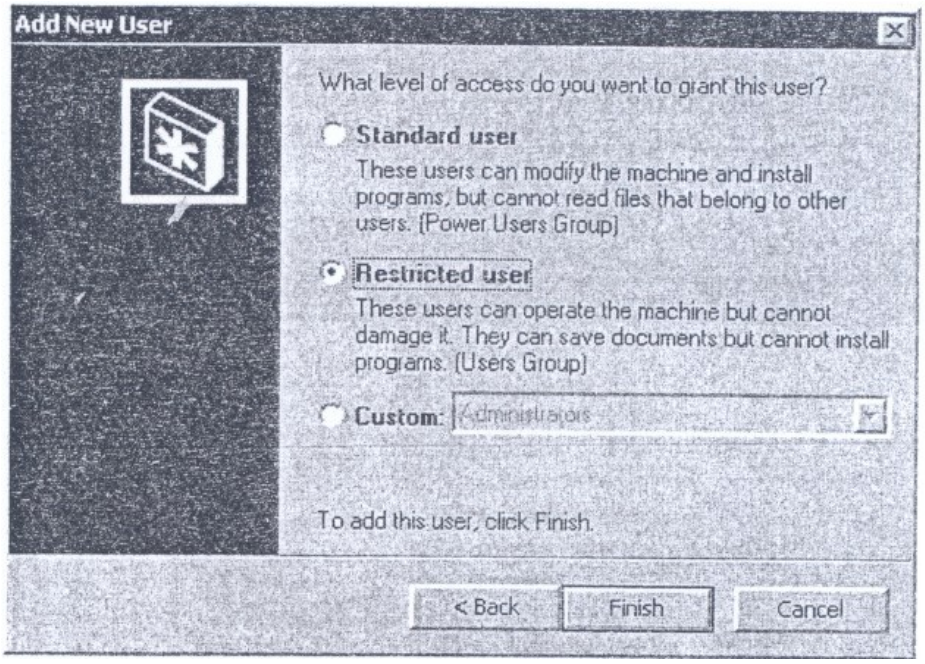


Figure 5: Grant Access Level Screen

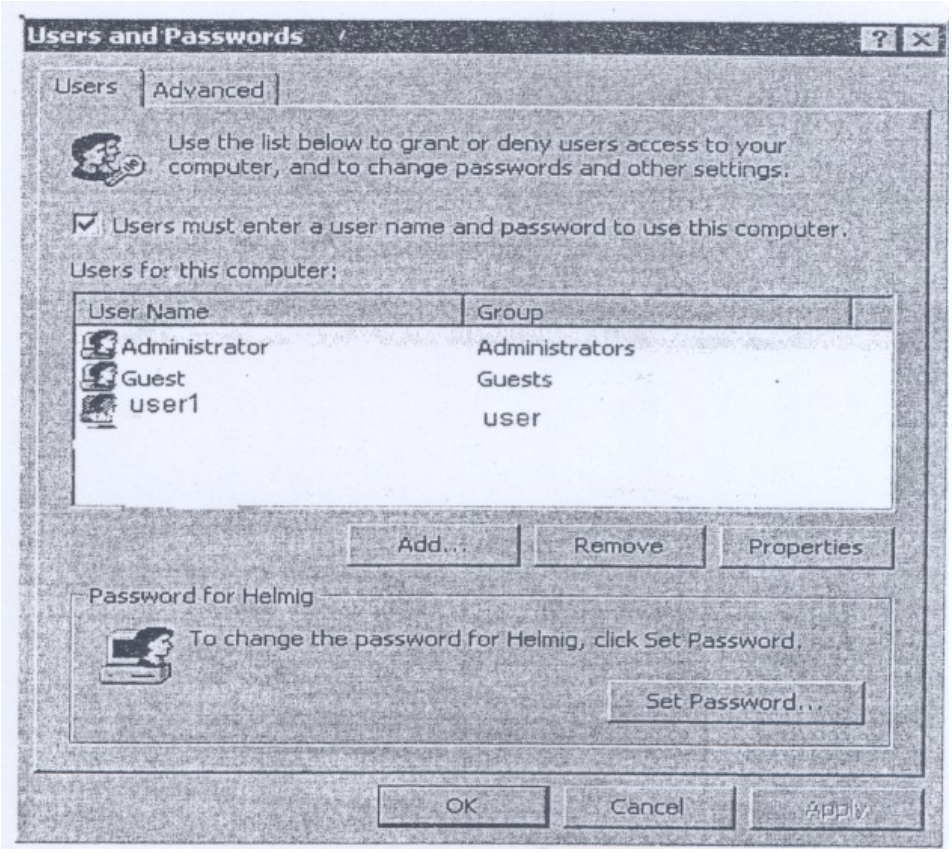


Figure 6: Change User Password Screen

You can view in detail the list of groups with each right/privilege:

The following set of windows add members to a group

While creating a new group, users can be added immediately to become a member of the group. But users can be added later to become a member of a group as well as in *Figure* 7 and *Figure* 8.

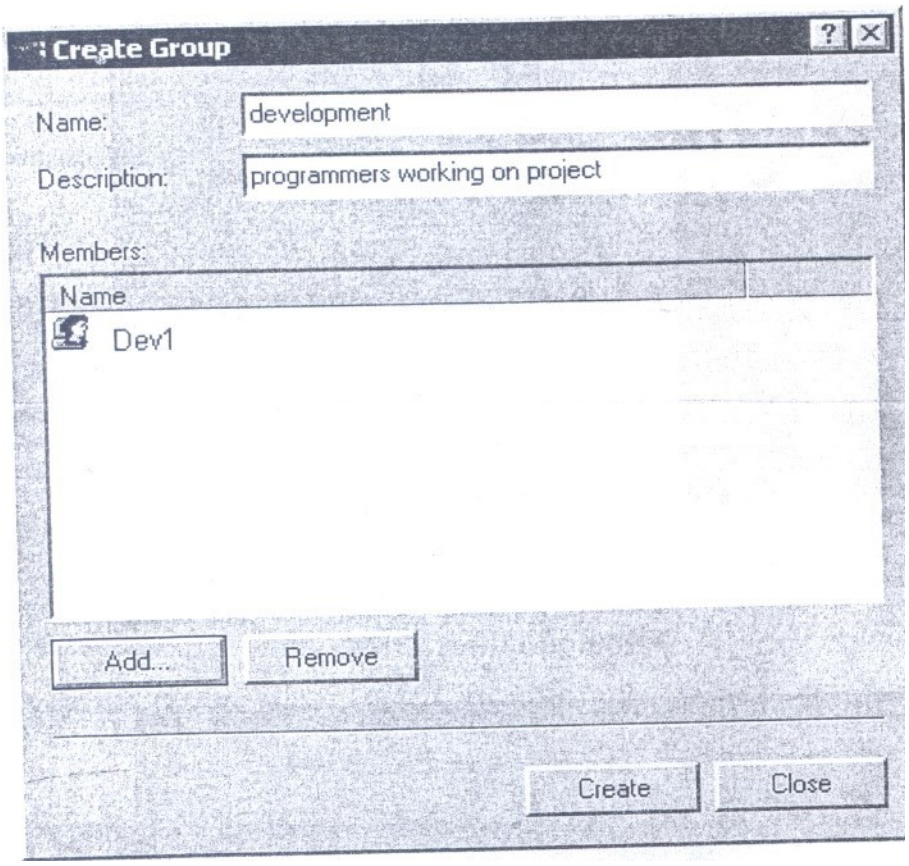


Figure 7: Create Group User Screen

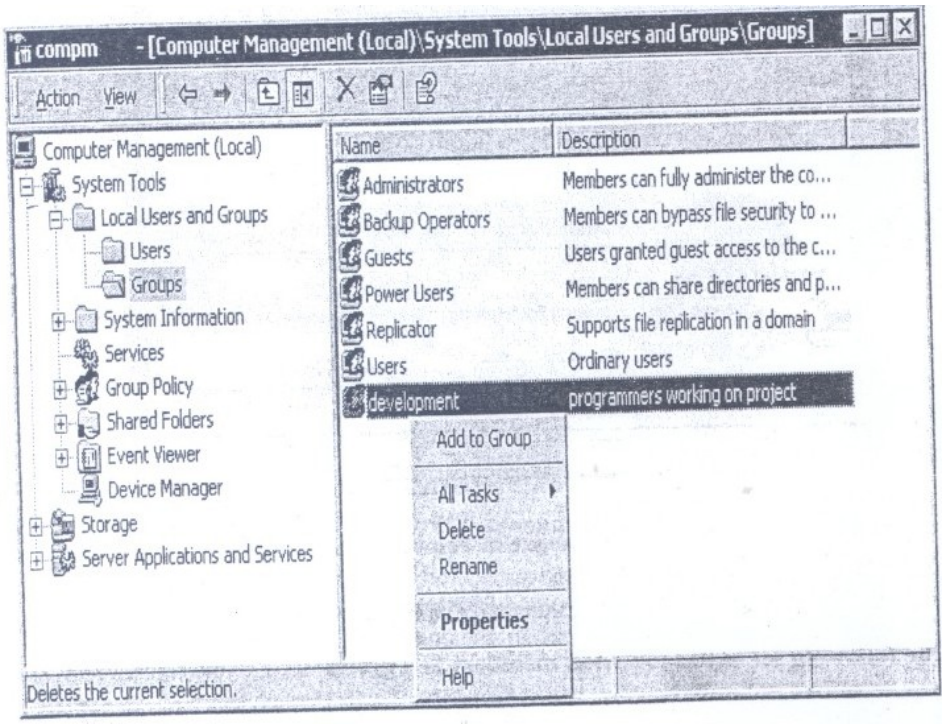


Figure 8: User Addition Screen

But to see in detail the permission/rights/privileges of a group, you need to "drill down" in the *"Group -Policies"* 4 levels down as in *Figure* 9.

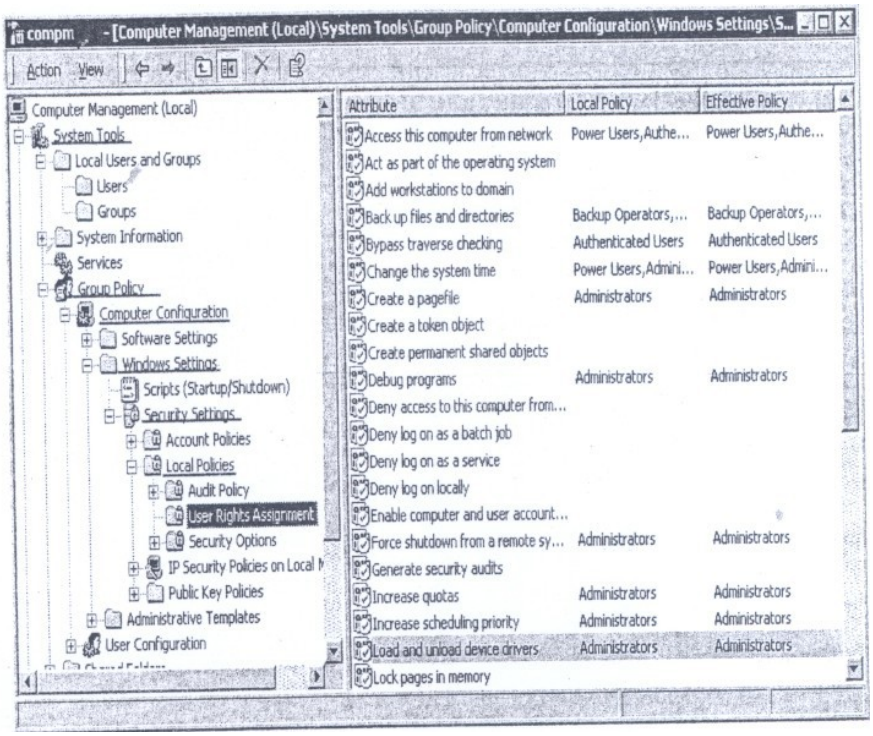


Figure 9: Detailed Permission Screen

For example, "regular users" do not have the right/permission/privilege to make backups.

To enable another group (one of the predefined or our own-defined groups) to have a right/privilege (like: make a backup), you need to add the group to the list: as shown in *Figure 10.*

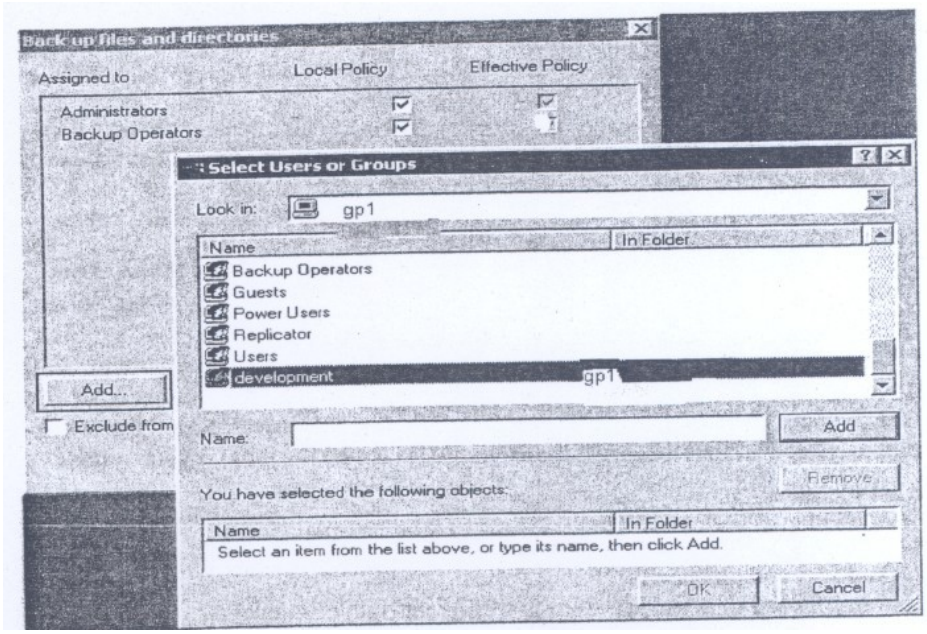


Figure 10: Printages/Permission Screen

3.3 Browsing Network Resources

When installed, Windows 2000 creates a set of folders to store program and data files. Windows folders and subfolders correspond to DOS directories and subdirectories but system folders do not.

Some of the system folders are:

1. Desktop
2. My Documents
3. My Computer
4. My Network Places
5. Recycle Bin
6. Internet Explorer

Descriptions of these folders are given below:

Desktop 

The desktop includes:

My Documents, My Computer, and My Network Places system folders. Here Files and folders can be saved and created.

If the user creates folders, save files on the desktop -then these are stored in Desktop render user's own user profile.

My Documents 

This icon is a short-cut of the actual folder that the user uses for data files.

My Computer 

This is responsible for the displaying of:

All local drives
Shared network drives
Mapped drives
Control Panel icon

This is a completely virtual folder, i.e., no file can be created or saved in this. My Computer folder IS a system folder.

My Network Places 

This is another virtual folder; it is responsible for providing access to all the network resources. Here you find the list of rights/privileges for all the jobs on your system.

Job list includes:

Accessing this computer from the Network
Backup files and directories
Restore files and directories (yes, it is a different right/privilege)
Load and unload device drivers -> Configure hardware, reserved for Administrators.

You can view in detail the list of groups with each right/privilege of networked computers. It provides the San1e functionality as was provided by the Network Neighbourhood in Windows 95/98.

Recycle Bin 

This folder is used 10 store files that are temporarily deleted from the system and has options for permanent deletion or restoring of files to their original locations.

Internet Explorer 

Viewing Folders as Web Pages

Windows 2000 provides an opportunity to display each folder as a web page.

This feature can be activated/deactivated for all folders using the option Web view on General tab of the folder options dialog box.

If you check enable web content in folders then info pane is available at all times for all folders.

If a use windows classic folder is selected then only a simple list of icons can be viewed without web content.

Four special attributes are associated with every file and folder for controlled access

On new files that are created by users these four attributes are always off. These special attributes are:

1. System

2. Archive
3. Read Only
4. Hidden

Windows Explorer

It is an all purpose system utility, it lets the user organise files in folders, allows for searching for documents and also data editing.

Windows explorer supports two views:

1. Single folder view
2. Two-pane explorer view.

Using the single folder view the contents of the current drive or folder can be viewed, whereas using two-pane explorer view all the drives, folders and resources on the user's computer and the network can be viewed in a tree structure. Two-pane view is also possible.

Arranging Files and Folders

Contents of folder window can be sorted by name, type, size or date. To sort files within a folder, pull down view menu and choose arrange Icons and choose any among the following options:

a. By name
b. By type
c. By size
d. By date

Even the width of folder panes can be changed by pointing to the vertical dividing line between the panes. When the mouse pointer changes to a two-headed arrow, click and drag.

3.4 Accessing Network Resources

My Network places are the system folder. It includes icons for all those computers that are part of the network in our domain (Servers and workstation). In Windows 95 Network Neighborhood was there. Windows 2000 has *Computer Near Me* which is similar to network neighborhood of Windows 95.

1. The most convenient way to gain access to or to manage files folders that are stored on another computer on the network is using My Network Places folder.

2. But if the shared folder is in another domain, a user name and a password is required to access the machine's resources. Thus the easiest way to find shared resources on your network is via My Network Places.

In My Network Places Folder

1. Double click on Entire Network icon, then choose Search for the shared resource on a network.

2. After entering the name of the computer that contains the shared resource in computer Name Box, click Search now. Also on double clicking Microsoft windows Network icon you get to see all other computers and domains on your network.
3. Another icon is Computers Near Me icon. This icon is available only if the network is a workgroup not a domain.

Step wise short cut

Double clicking icons on MY Network Places can be a tedious task, on large networks Windows 2000 provide a mechanism by which shortcuts can be created.

In order to create a short cut on .MY Network £laces folder double click on add network place icon on Add Network places folder.

For shared computer use \\computer-name. Shows all shares that are available on a given computer:

1. FTTP server -shortcuts, user can browser for files on a server. Using FTP use ftp://server_name
2. Web folder (HTTP Server) -lets the user save files directly on web server. Using http://server_name
3. Shared folder or drive: use \\computer_name\share_name

Following windows *(Figure* 11) describe My Network places:

As shown in *Figure* 11 From *"My Network Places",* we find the equivalent of *"Network Neighborhood"* as *"Computers Near Me".(Note:* if you make a logon to a domain server, there will be no *"Computer Near Me"* displayed).

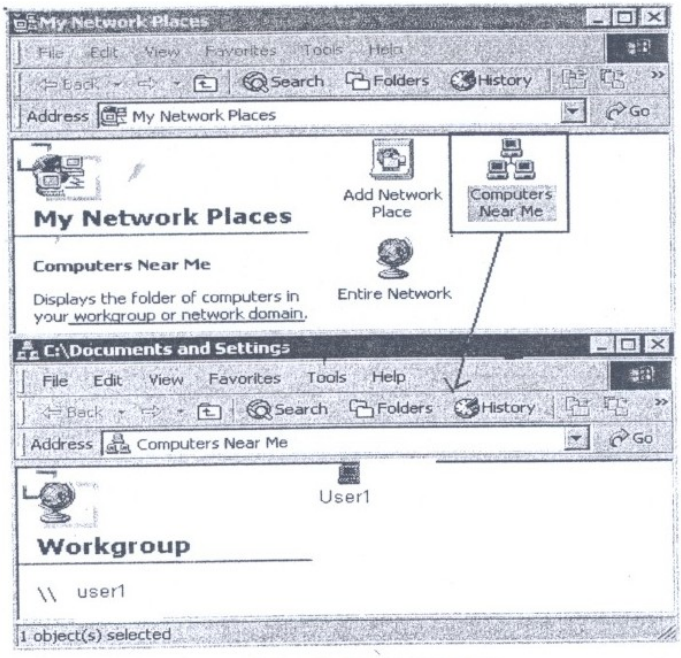


Figure 11: My Network Places

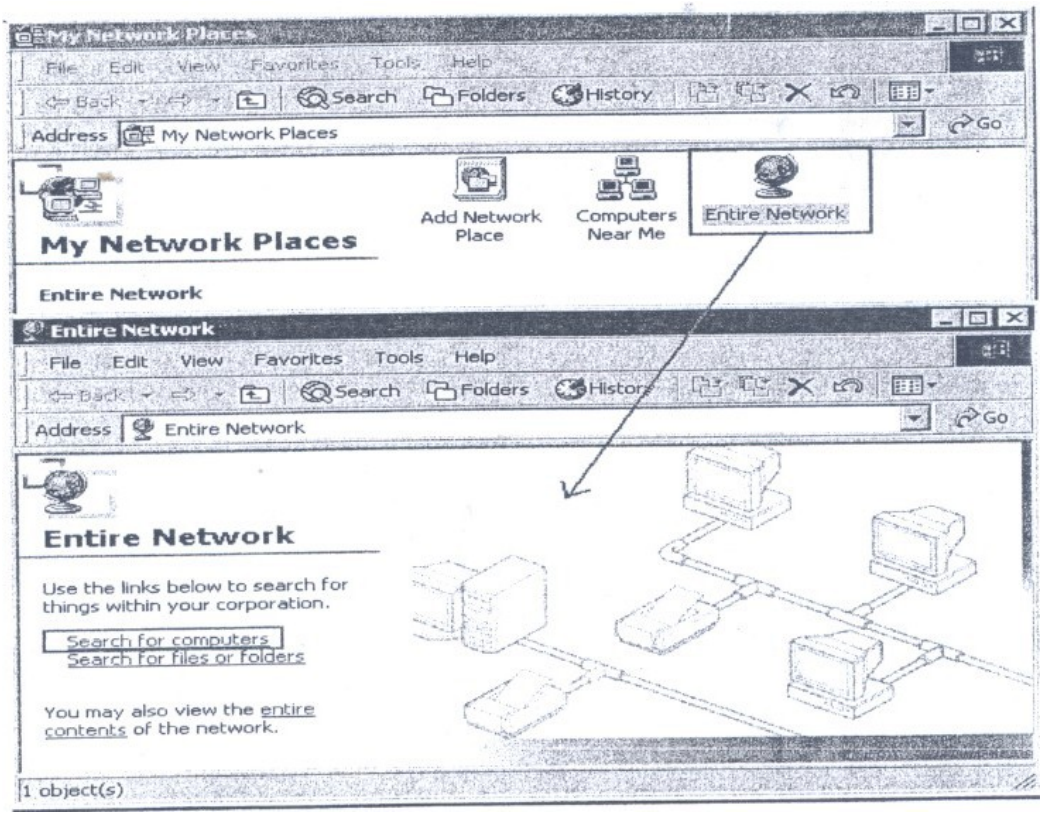


Figure 12: My Network Places Screen

Our search results give location, which is the workgroup only (as shown in Figure 13).

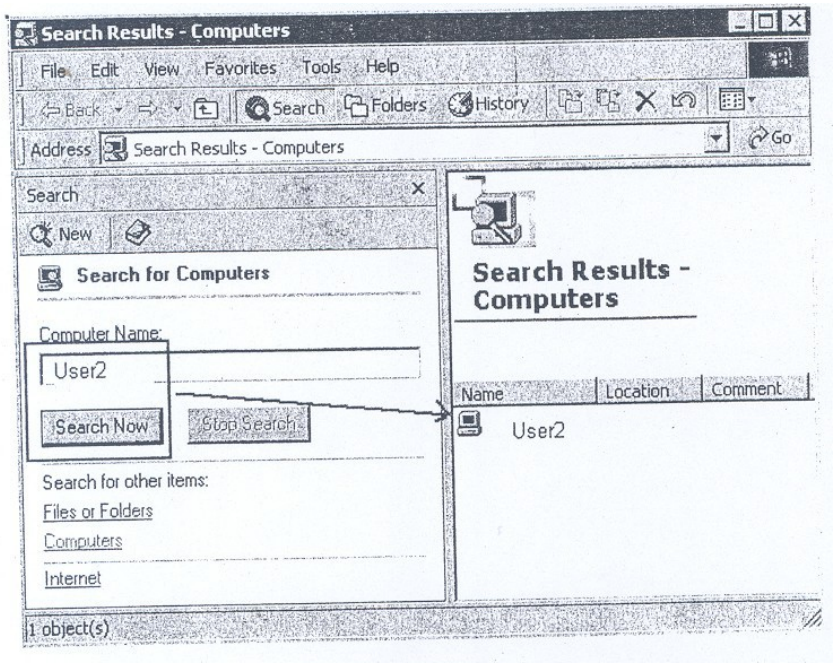


Figure 13: Search Result Screen

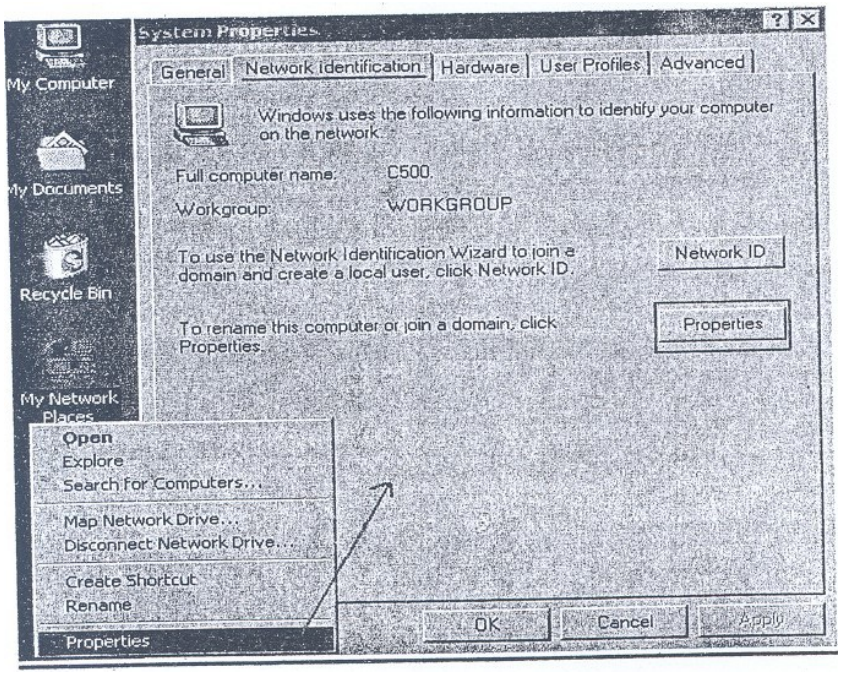


Figure 14: System Properties Screen

The following window is the place to define or make changes (as shown in *Figure* 15)

Computer name
Member of Domain or Workgroup
Domain/Workgroup Name.

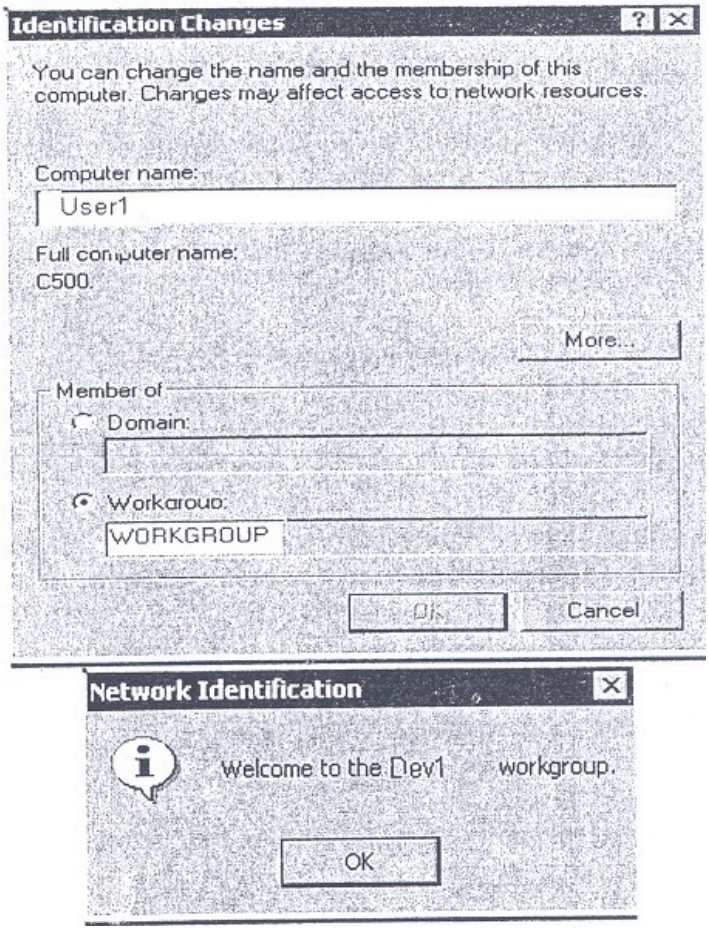


Figure 15: Change/Network Identification Screen

The workgroup, to which your system belongs, is defined in the Properties of "My Computer", Tab: "Network Identification". By default, the name of the workgroup is *"WORKGROUP"*

To implement the change click on the button" *Properties"* as shown in *Figure 1*6.

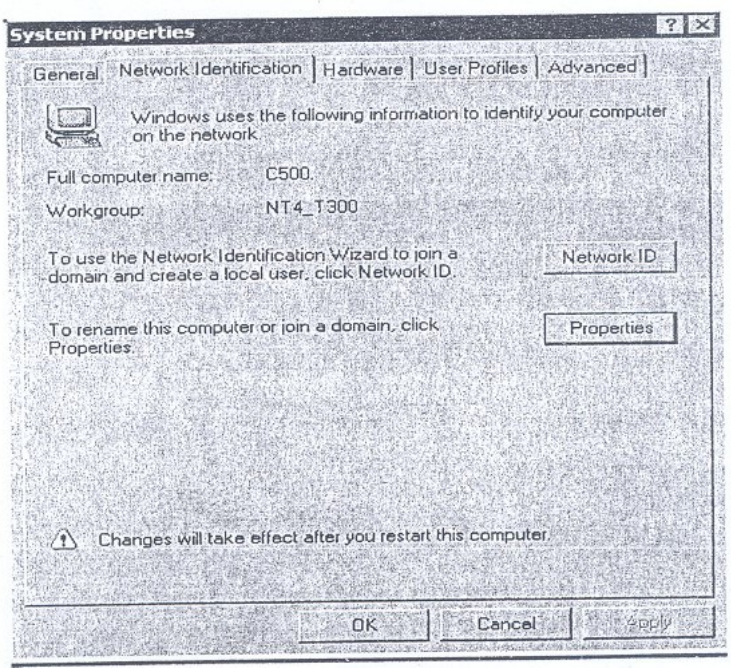


Figure 16: System Properties Screen

3.5 Mapping a Folder

(Using Windows 2000 user can map a drive letter to network resources - a printer, a drive, a folder. After mapping a drive letter shared resources can be treated as if they were on a local drive).

To map a network drive, right-click on the network share-name *(NOT* on *the Computer and not* on *any folder inside the share)* and select "Map Network Drive" Select the drive character to be used, decide on whether to "Reconnect at Logon" (If yes select the check box or else leave it).

Select the drive character to be used as shown In *Figure* 17.

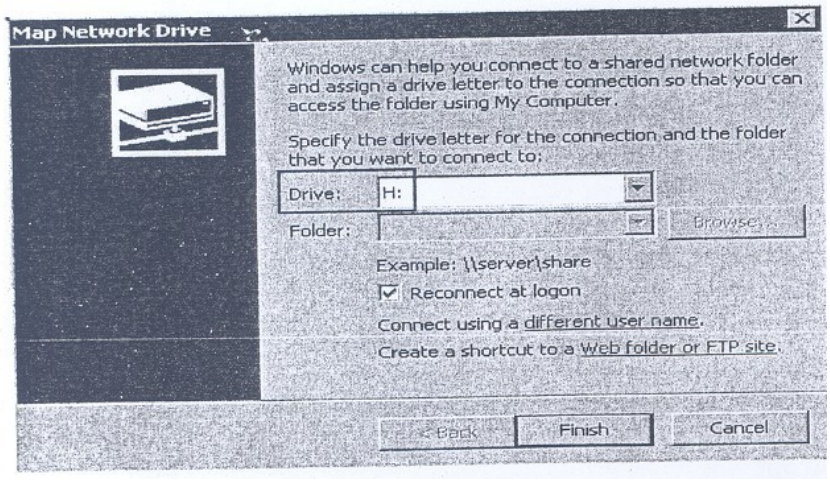


Figure 17: Drive Character Selection Screen

Mapping a network resource to a drive letter from explorer window *(Figure* 18)

1. From pull down Tools menu, choose Map Network Drive.
2. In the drive box, select the drive letter.
3. Write the name of the shared resource in the Folder box.
4. Click fi1sh after you are done with all the steps.

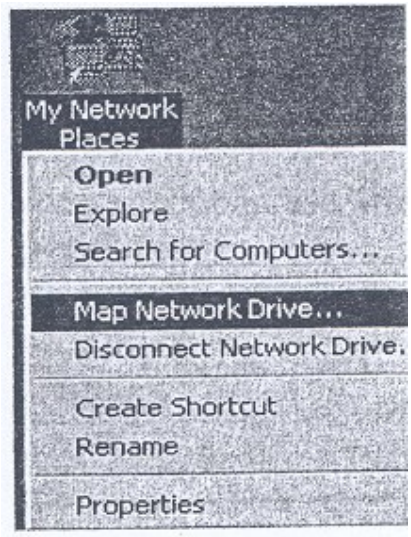


Figure 18: My Network Places

Following these steps can disconnect a mapped drive:

1. In any explorer window, in the Tools menu, choose Disconnect Network Drive. Since window displays a list of all currently mapped drives (as shown in *Figure* 21).
2. In the My computer window, choose mapped drive icon, right click on it, choose Disconnect from the shortcut menu.

Note: In order to assign a mapped drive to different drive, letter drive needs to be disconnected and then remapped to a new drive letter.

4.0 CONCLUSION

In this unit, you have been taught the two types of groups supported by the OS and how to browse and access network resources, based on windows 2000 server

5.0 SUMMARY

After reading this unit you are able to use Windows 2000 server and client. You can log-on to the network, browse through network resources and access network resources using My Network Places. This

unit describes the method of Mapping a folders. It also explains how to Map Shared Folders m to devices. There after reading this unit a user can access network resources (files, devices, printers) etc.

6.0 TUTOR-MARKED ASSIGNMENT

Fill up the blanks:

1) ---------- includes icons for all networked computers (servers and workstations) in a workgroup or domain in Windows 2000.
2) By default Windows 2000 assumes that user wants *to* log in using ---------- account._
3) In order to lock a local computer press ctrl+alt+del and click ---------- option
4) *To* share a folder, by default Windows 2000 uses name of the ---------- as the name of the share.
5) If the user is a member of administrator group on a local computer, the user can see and manage all share folders from a central location in ----------._
6) A user name in windows 2000 can be ---------- character long.
7) Contents of folder window can be sorted out by ----------regardless of the view, i.e., chosen by the user.
8) For access control every file and folder has four special attributes ----------,----------,----------,----------.

7.0 REFERENCES/FURTHER READINGS

www.microsoft,comwebsiteforadetaileddescription of Windows2000 Environment

White paper for distributed file systems at www.microsoft.com.

"Operating System Concepts", Silberschartz, Galvin and Gagne, Sixth Edition, John Wiley & Sons.

UNIT 3 ADVANCED WINDOWS 2000 NETWORKING

CONTENTS

1.0 INTRODUCTION

Windows 2000 provides an efficient networking environment. Domains, workgroups and trusted relationships describe the logical structure of Windows 2000. The physical structure of the domain hierarchy is completely segregated from the logical structure. As described in this unit, logical structure is made up of objects, Organisation Units (Ous), domains, trees and forests. The physical structure of the domain hierarchy is mainly composed of domain controllers and sites. A user account gives the user the ability to log on to the network or to a local machine. Everybody who regularly uses the network should have a network' account. Group policies further refine the user management in Windows 2000.

Also discussed in this unit is auditing. Lastly, there is RRAS (Routing and Remote Access Screen) which is a very important feature of Windows 2000 that lets remote access possible and is a tool for maintaining network security in Windows 2000.

2.0 OBJECTIVES

After going through this unit you should be able to:

describe Windows 2000 domains, workgroups and trusted relationships
manage efficiently user accounts in Windows 2000 networking environment

describe policies, auditing, active directory service in Windows 2000
describe remote access in Windows 2000.

3.0 MAIN CONTENT

3.1 Windows 2000 Domains, Workgroups & Trusted Relationships

In the following section we will introduce concepts of domains, workshops and trusted relationships.

3.1.1 Concept of Domains

A *Windows 2000* domain is a logical collection of network computers that share a centralised directory database referred to as Active Directory Service. In a domain this centralised information directory resides on a computer called domain controller. In Windows 2000 domain controllers are peers only.

Thus Windows 2000 domains provide the following advantages:

They provide extensibility features to existing networks.
Domains provide centralised control of all user information.
Thus domain can be referred to as the basic unit that is used for network growth and security in Windows 2000 network.

Usually one or more domain controllers are associated with a domain. In Windows 2000 Server a domain controller is the computer that is responsible for storing an entire copy of domain directory. In Windows 2000 it is the Windows 2000 Active Directory service that divides an organisation's network logically and physically. Logical structuring facilitates the finding by a user of a resource by name not by its physical location.

Logical structure of a domain comprises:

Objects
Organisation Units (OU)
Domains
Trees
Forests

Physical Structure of a domain comprises:

Domain controllers
Sites

Objects: A distinct named network resource can be referred to as an object. This object comprises certain related attributes. As an example, for an object printer, the attribute list may include printer name, make, etc. Similar objects can be grouped into classes.

Organisational Units: This is a container object. Container objects are objects that are residing within other objects. The purpose of an organisational unit is to organise the objects of a domain into logical administrative groups.

Domains: The basic unit of Active Directory Service is a domain. It is also referred to as a partition of an Active Directory Service. It is the domain only that is responsible for containing all network objects within it. It also serves as a security boundary to its objects. None of the security policies and settings, such as administrative rights, ACLs, ACE (Access Control Entries) can cross from one domain to another.

Trees: In order to support global sharing of resources trees are required. In a tree one or more Windows 2000 domains are arranged in a hierarchy. Thus by joining multiple domains in a hierarchy a large namespace can be constructed, which can further avoid name conflicts. All domains that are a part of a tree, or that share a tree can share information and resources. A domain tree has only one directory. As long as the user has the appropriate permissions he can use the resources of other domains in a tree. All domains in a tree share a common schema, which is a layout, a formal definition of all objects.

The central repository of information about objects in a tree or forest is called a **global catalog.** All domains belonging to a single tree share a global catalog. Domains in a tree also share a common namespace.

Forest: One or more trees can be grouped into a forest.

A forest comprises:

One or more trees
A common schema
It serves transitions trust relationships between trees.
Different namespaces between these trees.
A global catalog that contains the list of all objects in the forest.

Different users while accessing user objects must be aware of the domain name.

3.1.2 Trust Relationships

A trust relationship refers to a link between two such domains, where one domain is referred to as the trusting domain and other as the trusted domain. Trusting domain lets the trusted domain logon.

User accounts and groups that are defined for a trusted domain can access trusting domain resource even though those accounts are not present in trusting domain directory database.

A **Kerberos** (a security algorithm) transitive trust refers to a relationship type where:

Domain I trusts Domain II,
Domain II trusts Domain III,
Domain I trusts Domain III.

So a domain joining a tree acquires trust relationships of every domain in the tree. In Windows NT and earlier versions, there used to be only one-way trust relationships among domains.

Physical Structure of an Active Directory Service is responsible for affecting efficiency of replication in domain controllers.

Domain Controllers contains a copy of domain database. Wherever an update in the directory takes place, Windows 2000 automatically replicates the change to all other domain controllers in a domain. In a domain having multiple domains controller's directory information is replicated from time to time.

Only those computers running Windows 2000 Server, Advanced Server, or Data Center server can become domain controllers.

Sites are groupings of IP subnets (ranges). For example, one site can be 192.168.20.0/24 to 192.168.30.0/24

3.1.3 Building Domains

A computer can join Windows 2000 domain only after an account has been created in or added to the domain database. For that, a user must have the *Join A Computer* to the Domain permission.

By default, permission is granted to Administrator Members, Domain Administrator or Members of Administrators, Account Operators and Domain Administrator groups.
To join a domain a computer account for that computer should have been created in advance or it may be created during the installation process by selecting the check box 'Create a Computer Account in the Domain'.

3.2 User Administration

This section discusses user account administration. For a user to log onto a Windows 2000 network, a user account must be created. It is unique to every user and includes a user name and a password for authentication. A user can logon as a local user and a domain user as well. Thus by having an account a user has access to all network resources. As discussed in previous sections, in the Windows 2000 operating system two kinds of user accounts can be created:

Domain account
Local account

User account Administration includes setting up user profiles and name directories and modifying existing user accounts.

The next section discusses Group Account Administration.

Existing User Accounts Modification

Many different kinds of modifications are required with user accounts. These modifications may be required because of organisational or personal changes. An instance is whenever a new employee joins, the company may want to modify an existing account and give access to the new employee. Also, personal profiles may need to be updated at times.

Modification may include the following:

Renaming
Erasing
Disabling
Deleting User Accounts

1) To Rename a user Account: Normally renaming an account is done so that all access services to an account remain intact. When an account that has been created for a particular user is to be assigned to another user, all permissions, rights, properties set for that account are retained.

2) To Enable/Disable a user account: A user account is disabled when it is not needed for some time but would be accessed after a certain period of time. It is a situation when a user temporarily disables the account and needs access to it after a fixed period of time.

3) To Delete a user account: When a user no longer needs it, it is deleted.

Use Active Directory Users and Computers Snap-In,

Modify properties. To Reset the User Password:

1) Open Active Directory Users and Computers Snap-In and select the user object.
2) Activate the Action menu, click Reset Password. In the Reset Password dialog box, enter a password and select.

User must change password at next logon to force the user to change his or her password the next time1hat the user logs on.

Managing User Profiles

A user profile contains all data pertaining to a user. It also contains current desktop settings; all connected networked computers and all mapped drives. Modifying desktop settings can modify a user profile. It is created the first time when a user logs on to a computer.

When you log on to a network computer in Windows 2000 environment you get individual desktop settings and connections.

Windows 2000 supports Roaming User Profiles (RUPs), for users who work on more than one computer. A user set up a RUP on a network server and it is available to all the computers or the domain network. It is copied to client computer from Windows 2000 server when a user logs on. Thus, unlike user profile, with a Roaming User Profile the user always gets his individual desktop settings. Also a local user profile is on single client computer only.

Home Folder: A home folder is one that is provided to the user in addition to my documents folder to store personal data. It is not included in RISP (Routing and Remote Access Screen).

Group Accounts Administration

User accounts can be collected together. Such collections are called as groups. The grouping simplifies administration as new access permissions are assigned to a group rather than to individual accounts. All user accounts belonging to that group have access privileges. Moreover user(s) can belong to multiple groups.

In Windows 2000 environment there are two kinds of groups, Security groups and Distribution groups.

Windows 2000 has 4 built-in groups:

Global groups
Domain Local groups
Local groups
System groups.

Common types of user accounts are contained in groups. The group scope is responsible for membership of a group. Active Directory Users and Computers Snap- in are used to create a user group in a domain.

Group Policy

A group policy primarily comprises configuration settings that determine the layout of an object and its successors (children) objects. Group policies provide for controlling the programs, desktop settings, and network. In a network, group policies are normally set for the dColi1ain. Policy administrators administer group policies.

Types of Group Policies:

Scripts: let the policy administrator specify applications and batch files to run at specified times.
Software settings execute the applications. These policies can automate application installation.
Security Settings are responsible for restricting user access to files etc.
Remote Installation Services (RIS).
While executing client installation wizard, it controls RIS installation options.
Folder Redirection facilitates movement of Windows 2000 folders from their default user profile location to a place where they can be managed centrally.
Administration Templates consist of registry based group policies for managing registry settings, etc.

GPO (Group Policy Objects)

These objects contain configuration settings for group policies. Information is stored in two ways in a GPO:

1) In containers
2) In Templates

Creation of GPOs takes place before group policies. Group Policies can be modified using:

1) Group Policy snap-in or
2) Using Active Directory Users and templates snap-in.

Only administrators, creator owner or a user with access to GPO can edit a group policy.

Auditing

Windows 2000 auditing is a facility responsible for security. It is responsible for tracking user activities, keeps a check on them. Windows 2000 maintains a security log. User events are written onto their security log. All the events related actions are entered onto security log. An audit entry in security log not only comprises action that takes place, but also the user and success or failure of the event and when the action occurred. Thus whatever event takes place in Windows 2000, Security Log has an entry for the same.

An audit group policy is configured for all domain controllers in a domain. Auditing is assigned to parent container and it passes it down the hierarchy to the child containers. However, if explicitly a child container is assigned a group policy then child container group overrides parent container settings.

To plan an audit policy, computers must identify on which auditing is to be applied. By default, auditing option is turned off.

Only certain specific events can be audited on computers:

User logging on and off.
User accounts and group changes.
Changes to Active Directory Objects.
Files access.
Shutting down Windows 2000 Server
Restarting Windows 2000 Server.

3.3 Remote Access

Windows 2000 remote access mechanism lets remote clients connect to corporate networks or to the .Internet. Windows 2000 supports two kinds of remote access connection methods *(Figure* 1).

Dial up remote access
VPN (Virtual Private Network) remote access.

VPN provides a *secure* network connection between two remote machines.

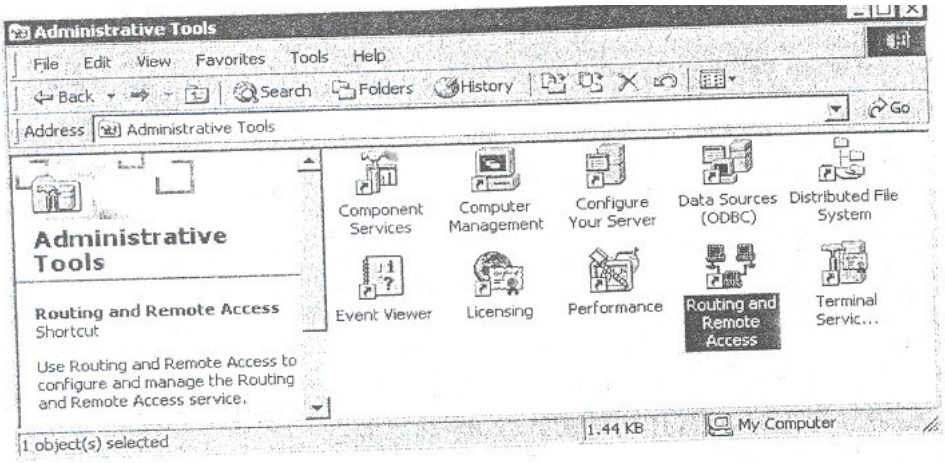


Figure 1: Dial up Remote Access screen

With **dial up** remote access, a remote access client uses telecommunication infrastructure to create a temporary physical structure to create a temporary network or a virtual network.

Right click on the server icon and select "configure and Enable Routing and Remote Access" as shown in *Figure* 2.

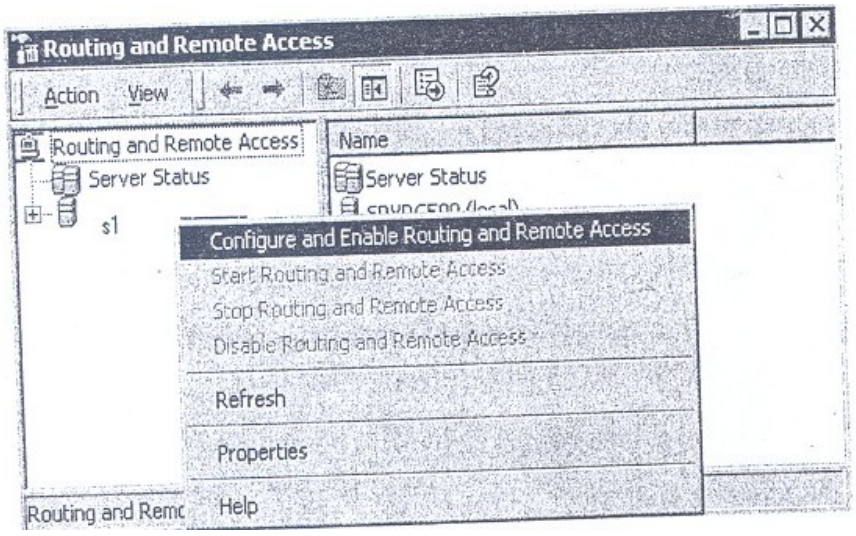


Figure 2: Renting and Remote Access Screen

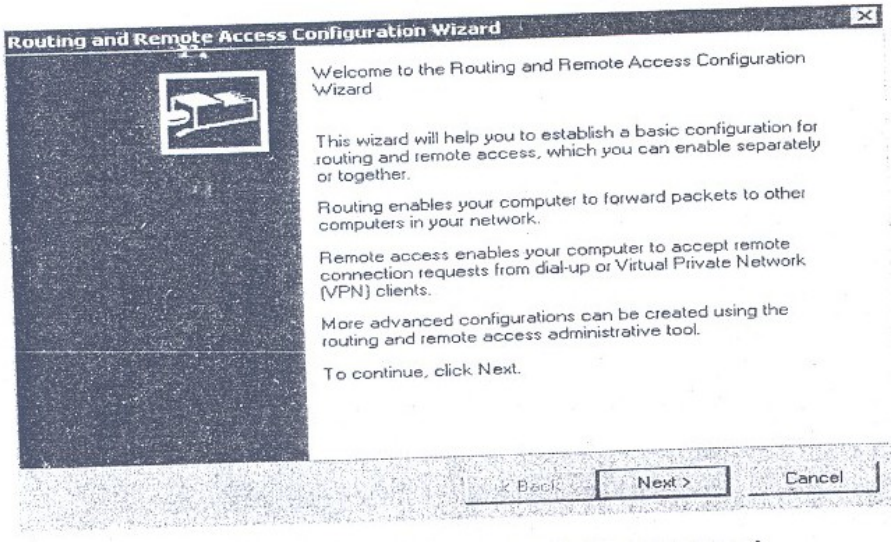


Figure 3: Routing and remote Access Configuration wizard

Using this, all devices for remote access can be enabled and the following screen appears *(Figure* 3 (a)).

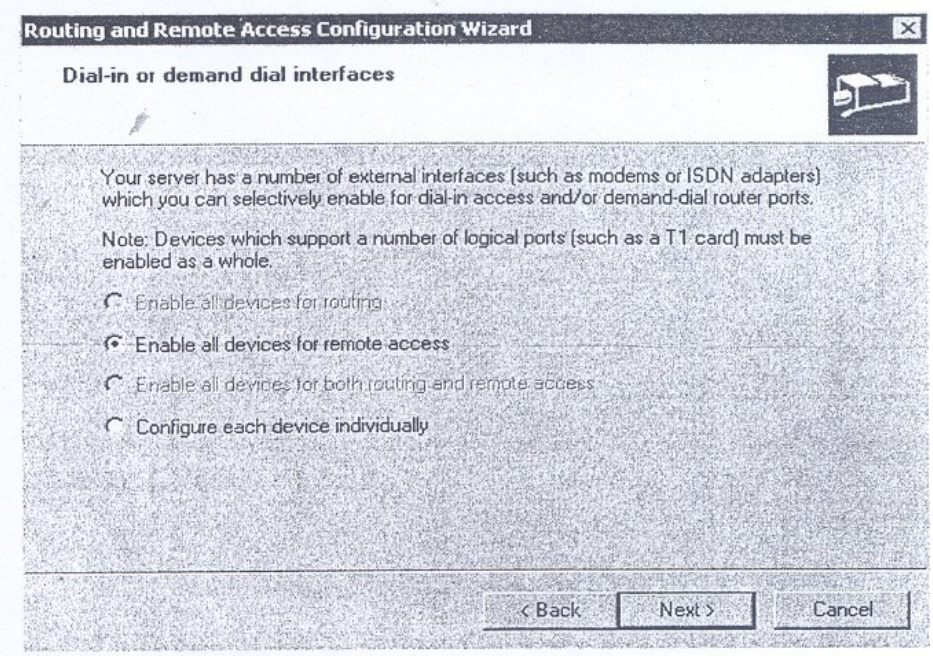


Figure 3(a): Routing and remote Access Configuration wizard

For security reasons use the following option as shown in *Figure 3(b):*

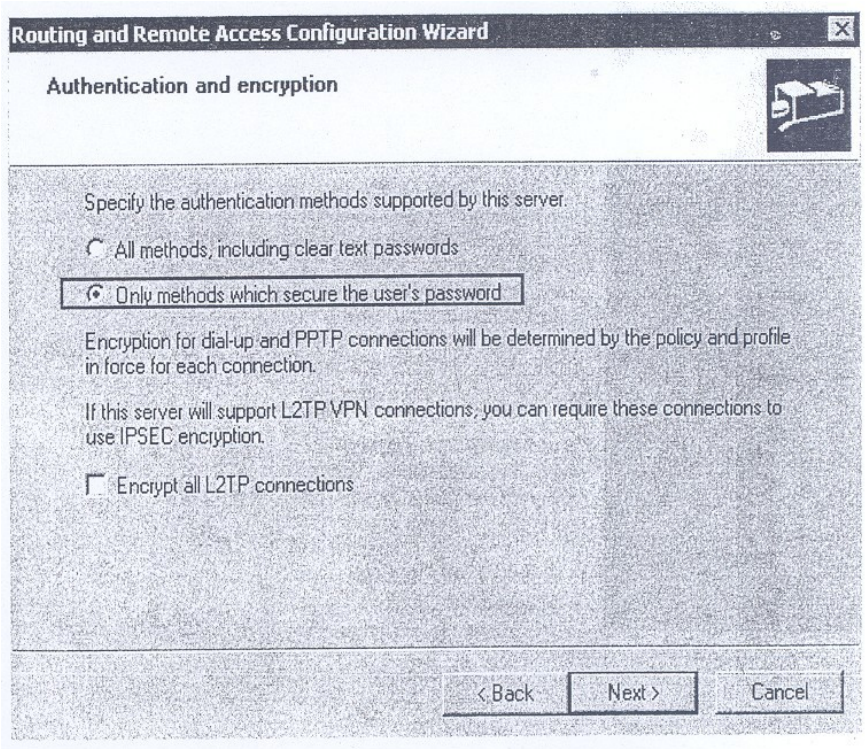


Figure 3(b): Routing and remote Access Configuration wizard

If you have TCP/IP then write TCP/IP as shown below in *Figure 3c.*

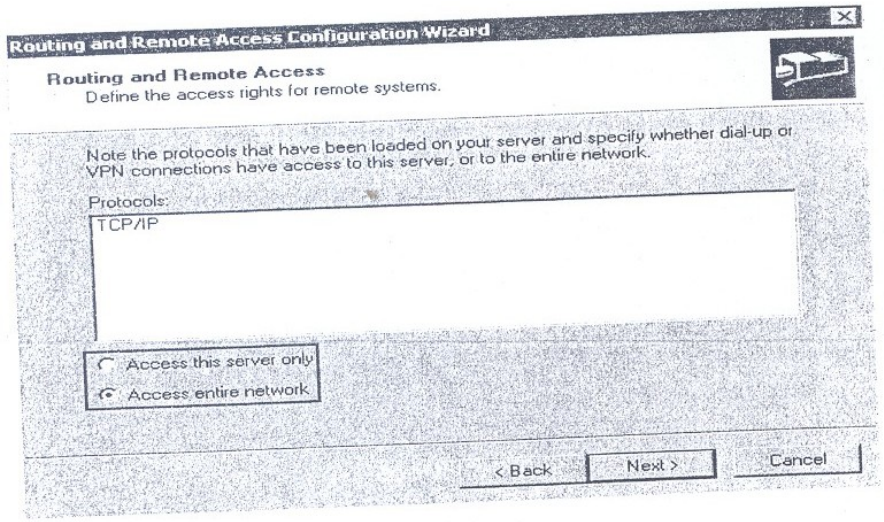


Figure 3 (c): Routing and remote access configuration wizard

Once all the requisites are complete then the following wizards *(Figure 3(c), 3(d)* and *3(1)* appear:

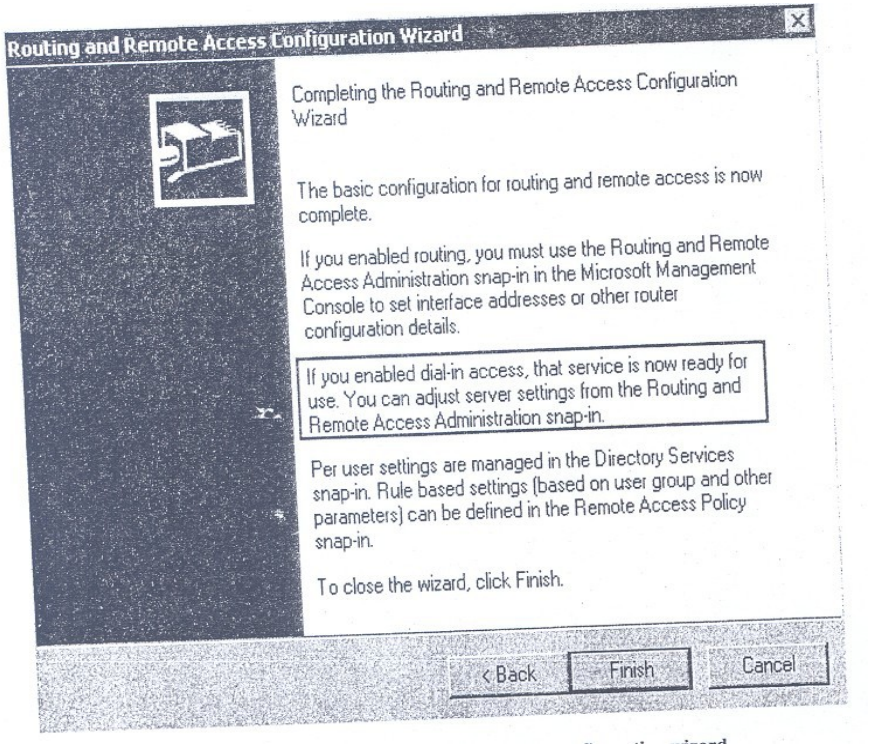


Figure 3(d): Routing and remote Access Configuration wizard

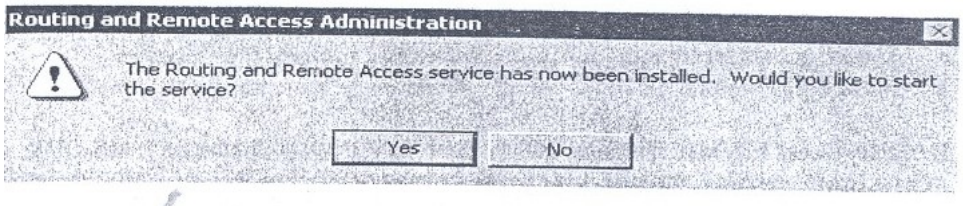


Figure 3(e): Routing and remote Access Configuration (RRAS) wizard

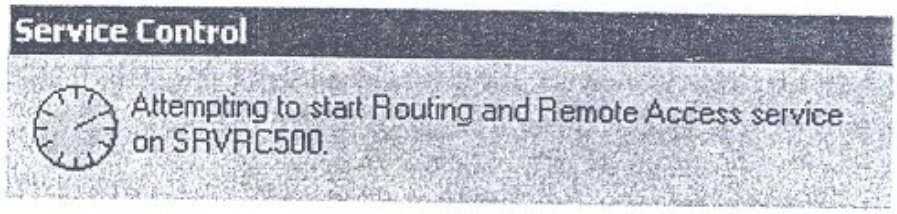


Figure 3(f): Routing and remote Access Configuration wizard

After this screen RRAS is now configured and the contents can be viewed s figure 4:

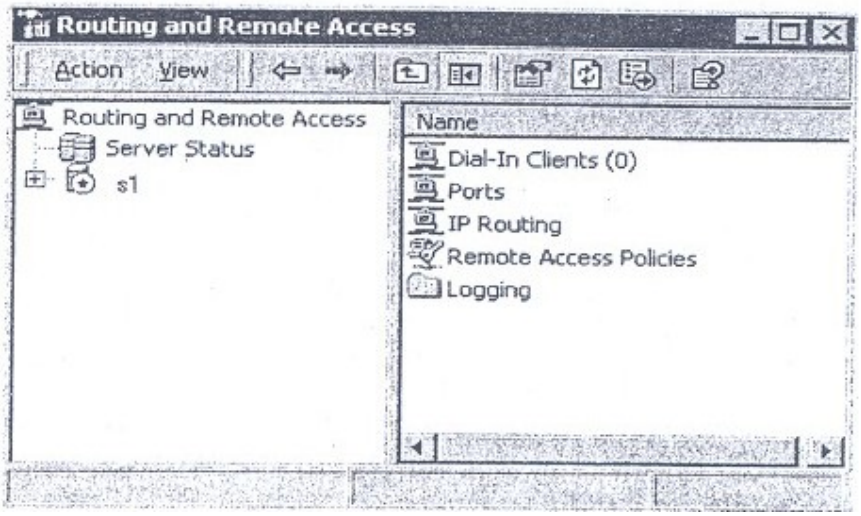


Figure 4: Routing and Remote Access

By default, Windows 2000 creates automatically 5 PPTP and 5 L2TP port for incoming VPN-connections (figure 5)

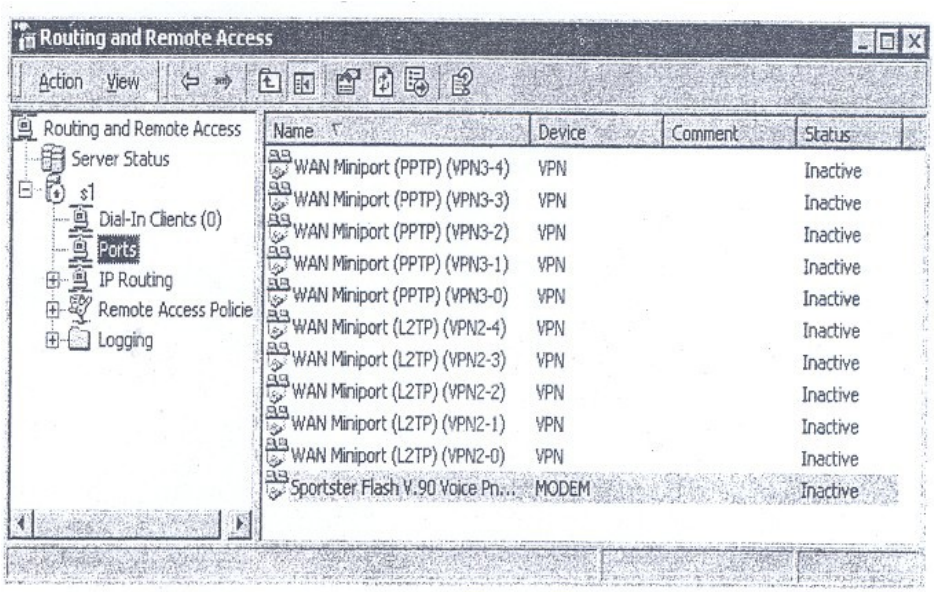


Figure 5: Remote Access Connection Screen

With VPN remote access -remote access of a VPN client uses an IP network to create virtual point-to-point connection with a RAS server acting as a VPN server. A dial up Remote Access Connection consist of:

Remote access client
Remote access server
WAN infrastructure.

Remote Access Clients: Windows 2000, Win NT, WIN 98, Windows 95, MS-DOS; MS LAN Manger are remote access clients that can connect to Windows2000 remote access server. Third party clients like UNIX and Apple Macintosh too can connect to windows 2000 remote access server.

Remote Access Server: Windows 2000 server accepts requests from client's connections and forwards it to other clients or to the network.
WAN Infrastructure depends upon the type of connection being made. There are various networks like:

PSTN (Public switched telephone network
ISDN (Integrated services digital network
X.25 (UY -T Protocol based WAN)

Windows 2000 support three types of Remote Access protocols PPP, SLIP and asynchronous NetBEUI, also TCP/IP, IPX, AppleTalk.

Windows 2000 remote Access provides a variety of security features like:

User Authentication
Mutual authentication
Data encryption
Call back
Caller id
Remote access account lock out.

Remote Access Management involves managing users, addresses, accesses and authentication.

Virtual private network is an extension of private network that involves encapsulation, encryption, authentication to links across shared or private networks. A VPN mimics the properties of a dedicated Private network through Internet; allowing data transfer between two computers in a network. Corporate offices can use two different methods to connect to a network over the Internet:
Using dedicated lines or dial up lines VPN uses tunneling to transfer data in a VPN. Tunneling is a secure method of using an internetwork infrastructure to transfer a payload.

A tunneling protocol comprises tunnel maintenance protocol and tunnel data transfer protocols. Two basic types of are:

1) Voluntary tunnels
2) Compulsory tunnels.

Protocols used by WIN 2000 for VPN are PPTP (Print to print tunnel Protocol), UTP (Layer 2 Transfer Protocol), IPSec (IP security), IP-IP.

VPN management involves managing user addresses, servers' access, authentication, and encryption. Troubleshooting VPN involves checking connectivity, remote access connection establishment, routing, IPSec.
Windows 2000 provides a set of RRAS tools:

Routing And Remote Access Snap In enables RRAS, management of routing interfaces, IPX routing configuration, creation of static IP address pool, configuring remote access policies. This is available from Administrative Tools folder.

Net Shell Command: Windows 2000 Netshell command is a command line and scripting utility. It is named Netsh.exe and is installed in % systemroot %\system32 when a Window 2000 is installed.

4.0 CONCLUSION

This unit has taken you through advanced windows 2000 networking. You have learnt policies, auditing, active directory services and how to use remote access in windows 2000.

5.0 SUMMARY

This unit highlights working of a domain, workgroups and trusted relationships in a Windows 2000 network. Windows 2000 provides a secure network environment for efficient resource sharing. Logical structure of domain hierarchy comprises objects, organisational units, domains, trees and forests. Domain controller and sites make up the physical structure of a domain. Many types of group policies exist, software settings, scripts, security settings, folder redirection etc. Group policies are a set of configuration settings that apply to one or more objects in the directory store. The structure of a group policy is made up of group policy objects, templates and containers. Group objects must be created before the creation of group policies.

Auditing is the process of tracking both user and Windows 2000 events. Windows 2000 writes the events to the security log on each computer. An audit entry contains information about the event that occurred, user responsible for performing that event, success and failure of that action. Another interesting feature in Windows 2000 is RRAS that lets the remote access possible.

6.0 TUTOR-MARKED ASSIGNMENT

1) Give the default order of group policy implementation through Active Directory service hierarchy.
2) Is it possible to set u encryption on a compressed folder?
3) When should security groups be used instead of distribution groups?
4) If the domain mode is switched over from mixed mode to native mode, what are the implications?
5) If a remote access client wants to connect to RAS server but connection is not allowed how will this error be solved?
6) Write the purpose of VPN and name VPN technologies supported by Windows 2000?

7.0 REFERENCES/FURTHER READINGS

www.microsoft.com/ windows2000 OS living

"Operating system concepts" Silberschatz, Galvin & Gagne Sixth Edition, John Wiley and Sons.

UNIT 4 WINDOWS XP NETWORKING

CONTENTS

1.0 INTRODUCTION

Windows XP is a network operating system. Microsoft introduced Windows XP so that it can be used in small networks as well as in networks spanning a large area. Windows XP comes with Windows XP Home Edition and Windows XP Professional. Home Edition supports workgroup networking but does not support domain networking. Windows XP also supports most of the networking features that were there in Windows 2000. Our objective in this unit is to highlight the features of Windows XP professional edition.

2.0 OBJECTIVES

After going through this unit you should be able to describe:

Windows XP networking features
file sharing features in Windows XP
folder sharing in Windows XP
disk sharing features in Windows XP
file Encryption in Windows XP
offline features in Windows XP.

3.0 MAIN CONTENT

3.1 Introduction to Windows XP Networking

In this subsection we will take up some standard protocols supported by Windows XP system.

3.1.1 TCP/IP Protocol Setting for Windows XP

TCP/IP Protocol is a suit of protocols that provides a set of vast networking capabilities. In Windows networking environments TCP/IP is the default protocol for both user group and domains. Windows XP has many built in features for configuring and monitoring TCP/IP.

Configuring IP settings in Windows XP:

TCP/IP protocol suite is the default installation on all Windows XP systems.

To access TCP/IP properties:

1. Initially log as administrator
2. Open network Connections:
   From windows XP start menu, choose connect to
3. Right click local area connection icon, choose properties from shortcut menu.
4. On the general tab, select Internet Protocol (TCP/IP) and click properties.

The *Internet Protocol (TCP/IP)* Properties dialog box opens. Through this dialog box the computer can be configured to use static or dynamic addressing.

A new feature in Windows XP is Alternate IP Configuration tab in *Internet Protocol (TCP/IP)* Properties dialog box.

It allows an automatically assigned:

IP addresses if a DHCP server is available.
Static IP configuration when a DHCP server is not available.

Thus this option enables the user to connect to two different networks and get address assigned.

3.1.2 To Select a Network Protocol

Click the network protocol that you wish to work on (as shown in *Figures* 1 & 2):

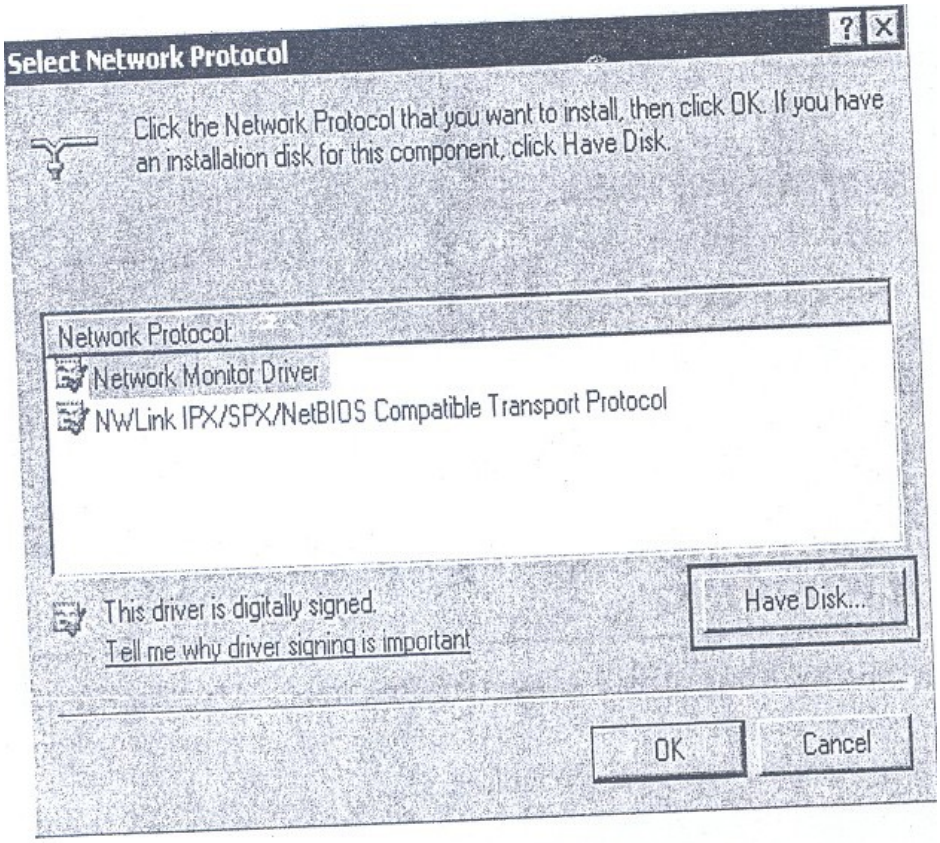


Figure 1: Network Selection Screen

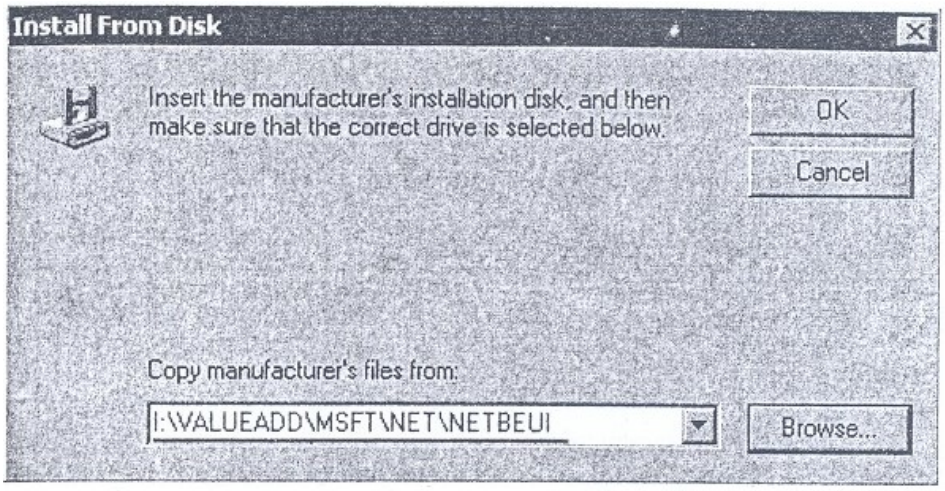


Figure 2: Installation Screen

If we right click on My Network Places to display network properties, this window (figure 3) appears on the screen,

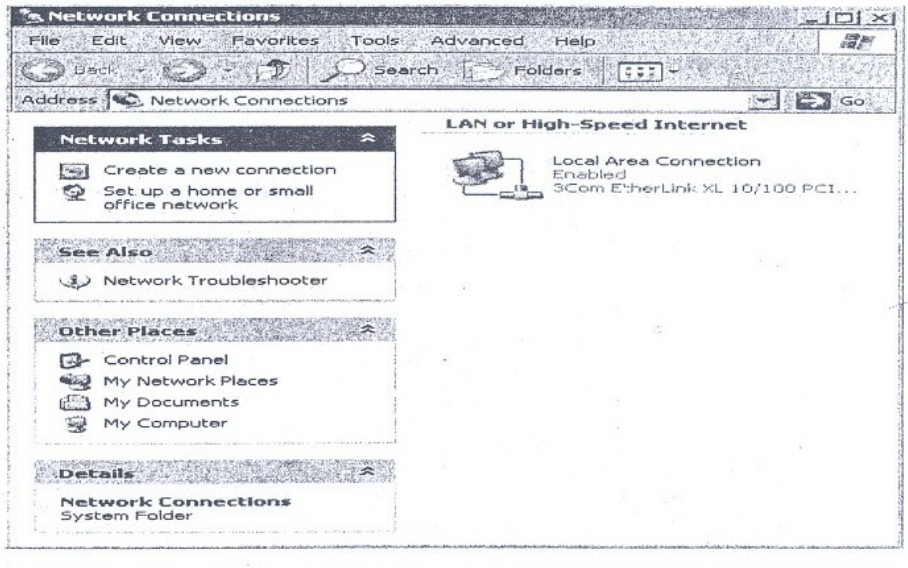


Figure 3: Network Connection Screen

Then the following windows (figure 4) for LAN connection properties appear:

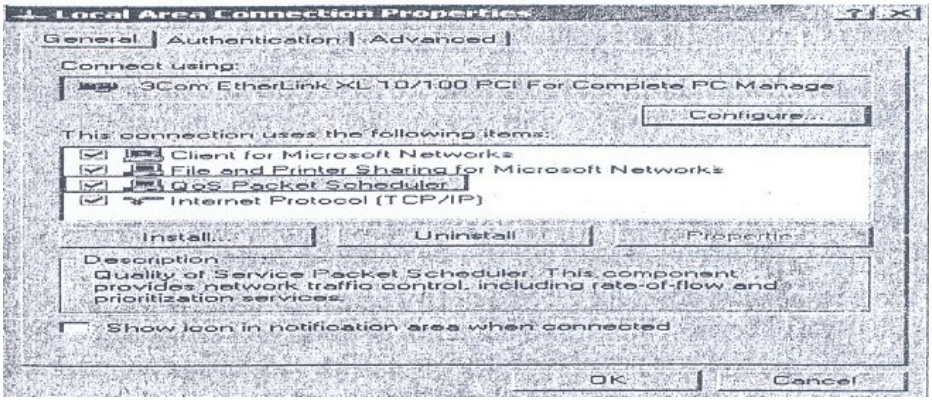


Figure 4: LAN Connection Properties Screen

For authenticated network access the following screen *(Figure* 5) is used.

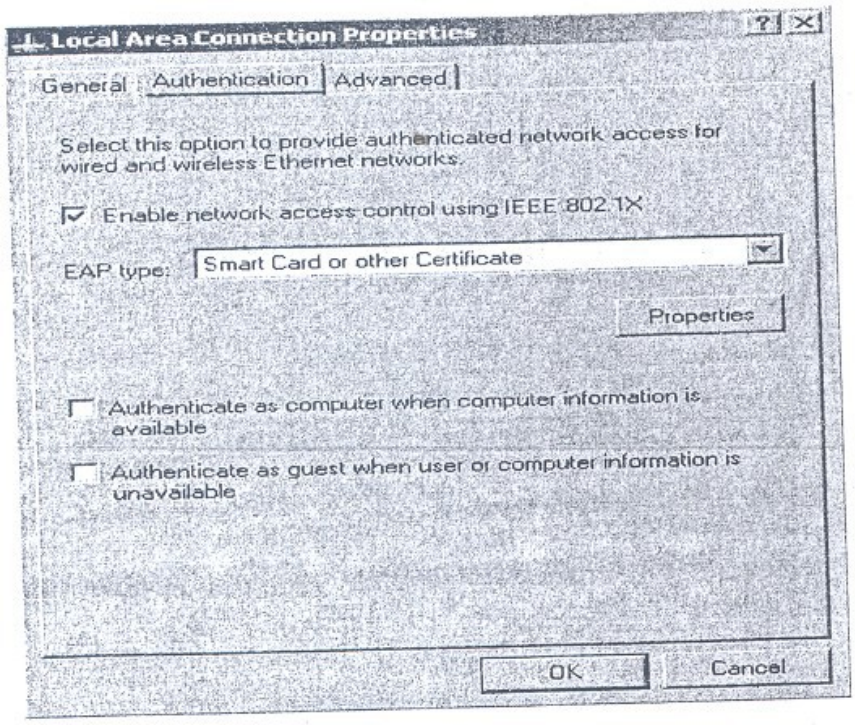


Figure 5: LAN Properties Screen.

For selecting network components that you wish to install on your network use the following screens *(Figure* 6):

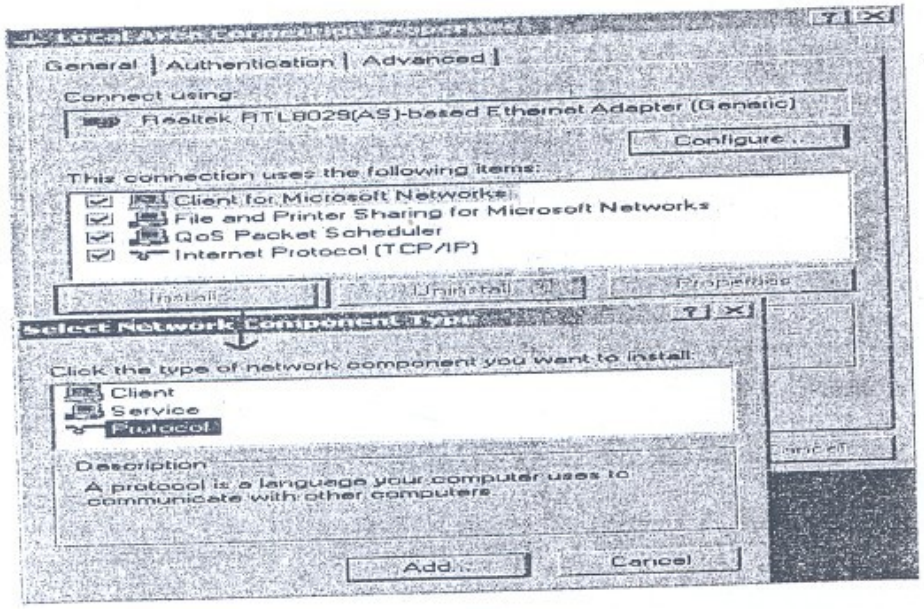


Figure 6: Network Component(s) Selection Screen

The NetBEUI Protocol is not available in Windows XP:

Support for the NetBIOS Extended User Interface (NetBEUI) network protocol has been discontinued in Windows XP. This protocol is not available for installation in Windows XP.

If you upgrade from a previous version of Microsoft Windows with NetBEUI installed, the Compatibility Wizard displays the following message:

The currently installed driver for the NETBEUI Transport Protocol is not compatible with Microsoft Windows XP and will be uninstalled during the upgrade. This protocol is removed from this new version of Windows as shown in *Figure* 7.

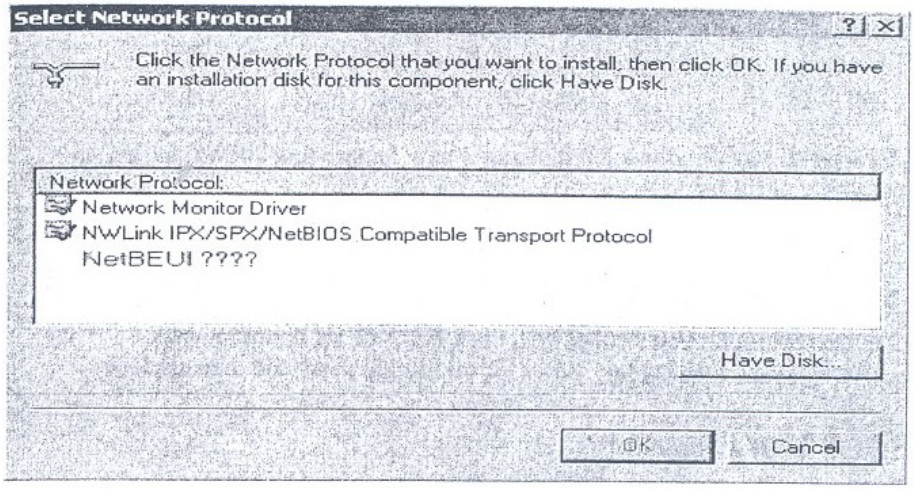


Figure 7: Network Protocol Selection Screen

For more information about this driver, visit the manufacturers Web site at http://www.microsoft.com. Web addresses can change, so you may be unable to connect to this Web site.

For a list of protocols supported by Windows XP, see the Microsoft Windows Whishtis Protocols Compatibility List at the Microsoft Web site.

3.1.3 Virtual Private Networks and Remote Networking

Windows supports Virtual private networks connection to access machines remotely. A VPN connection lets one system connect securely to another machine over the network. A VPN is an extension of a private network that comprises links across shared or public networks. But here in VPN, local network data is encrypted and is secure (referred to as

tunneling), for security considerations. For VPN connection either use Point to Point (PPTP) or Layer 2 tunneling protocol (L2TP).

3.2 Windows XP in File Systems

File Systems manage the way in which system resources are shared. All network file sharing are based on it. By default NTFS is the file system for fixed storage in Windows XP.

To connect a drive to NTFS, follow these steps:

1) Choose Start, Run, Type cmd and click Ok.
2) At command prompt, type converts C:\FS: NTFS where C is the letter of the drive.
3) Press enter to run the command.

Note: If aI1y of the files on a disk volume are open then volume won't be converted.

File Encryption

Windows XP Professional lets the user encrypt any of the files or folders using EFS. The user can still use that file or folder but no one else will be able to access it, if that file is not shared.

To encrypt a file or folder:

1. Right click the file and choose properties.
2. On the General tab, click the Advanced option.
3. In the Advanced Attributes dialog box, select Encrypt contents to secure data and click OK.

This EFS service in Windows XP includes a new feature that allows sharing an encrypted file or fo1der.

3.3 Sharing Network Resources in Windows XP

In the subsection we will describe the process of sharing files, folders and devices in Windows XP.

3.3.1 Sharing Files in a Windows XP

By default, Windows XP computers that do not belong a domain use a new feature called Simple File Sharing.

New Feature in Windows XP:

Simple File sharing makes NTFS permissions easy for users to manage.

While sharing a resource with simple file sharing enable others users have read only access to the file. Also Full Control can be given to the users.

But Windows XP computers that belong to a domain cannot use simple File sharing.

3.3.2 Sharing Folders in Windows XP

To share a folder with Simple File Sharing enabled, you first need to ensure that the folder does not currently reside in a private folder. If the folder does, it is either removed from the parent folder or to another location (as in *Figure 8(a)).*

To share the folder, follow these steps:

1) Right click the folder that user wishes to share. Choose sharing and security.
2) On the sharing tab, select share this folder on the network; give a name for the folder in the share name box.

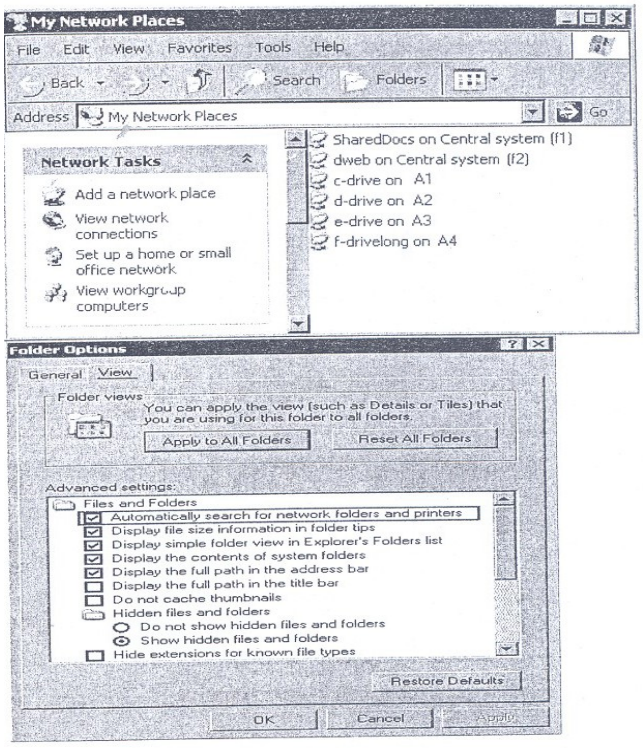


Figure 8(a): Sharing Folder Screen(s)

Following *Figure 8(b)* is a list *of shares* (shares refer to shared resources over the network) on the network: if the permission for sharing has not been granted then a dialog box appears as it is shown in *Figure* 8 *(b).*

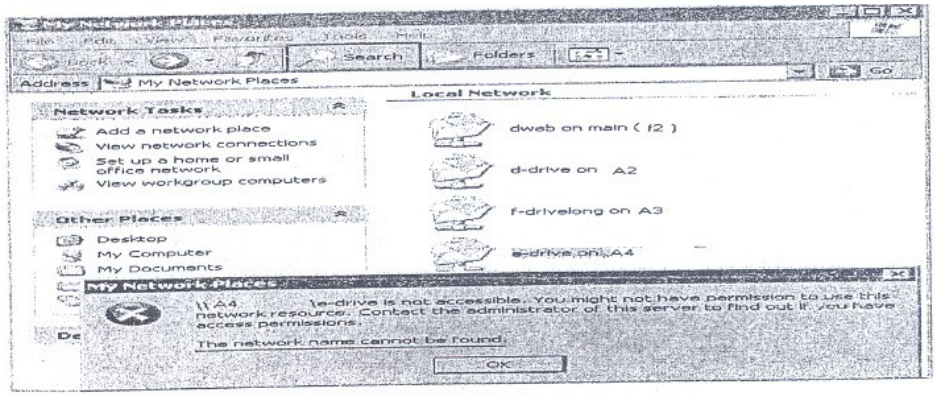


Figure 8(b): Drive Mapping Screen

The following screen (figure 8(c) & (figure 9) share a give folder on the network.

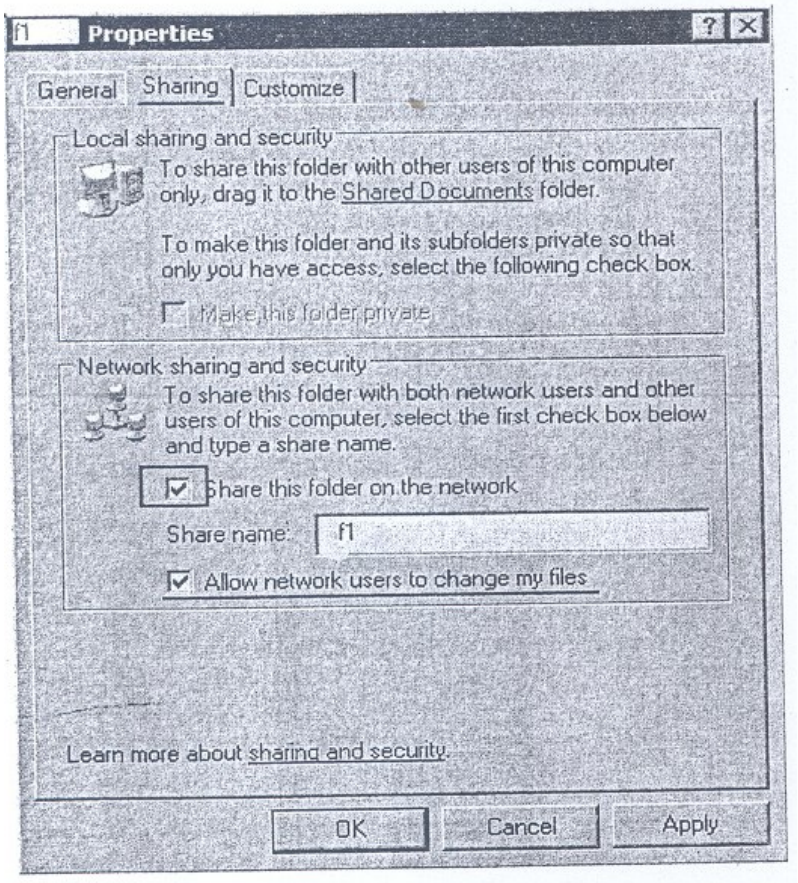


Figure 8(c): Resource Properties Screen

When this folder n is now shared using the "Simple File Sharing" then also the security settings are modified. .Thus the option -Allow Network Users to change my files is enabled and users will have full control to edit and delete files. But if you want users to be able to read your files only, clear this check box.

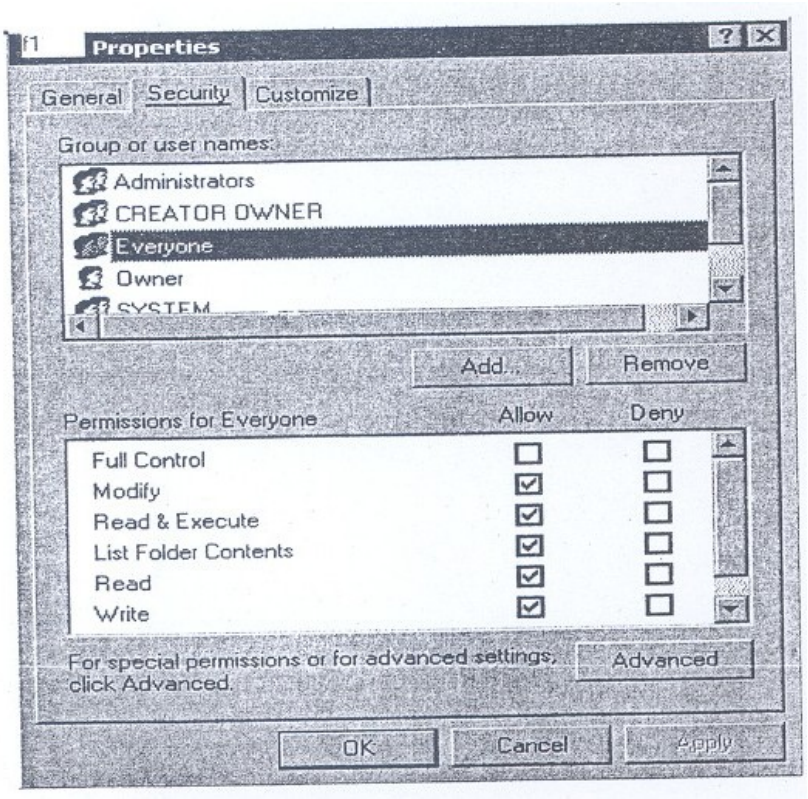


Figure 9: Advanced Properties Screen

3.3.3 Sharing Drives in Windows XP

To share a drive *(Figure 10),*

1. Right click the drive letter that the user wishes to share.
2. Choose sharing and Security.

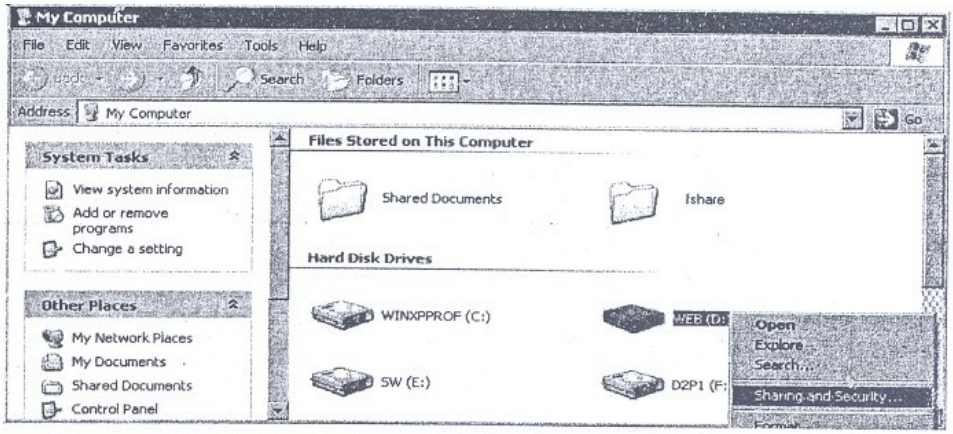


Figure 10: Drive Sharing Screen

Windows XP lets the user handle security issues *(Figure* 11)

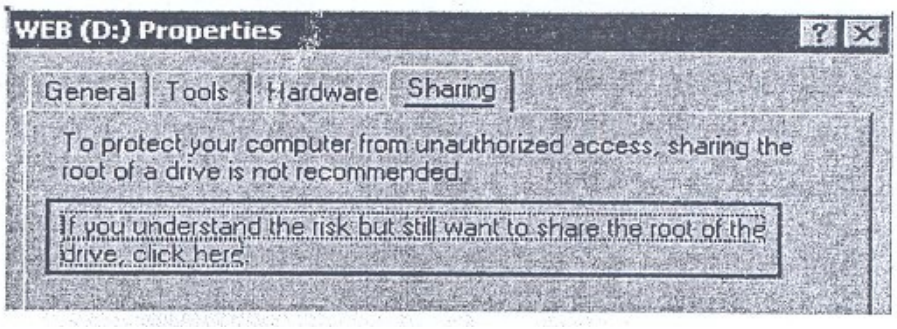


Figure 11: Web (D :) Properties

1) Select the desired folder from the share.
2) Right click on the folder and select "Sharing and Security".
   or
3) On the left side select "Share this Folder" *(Figures* 12 *and* 13).

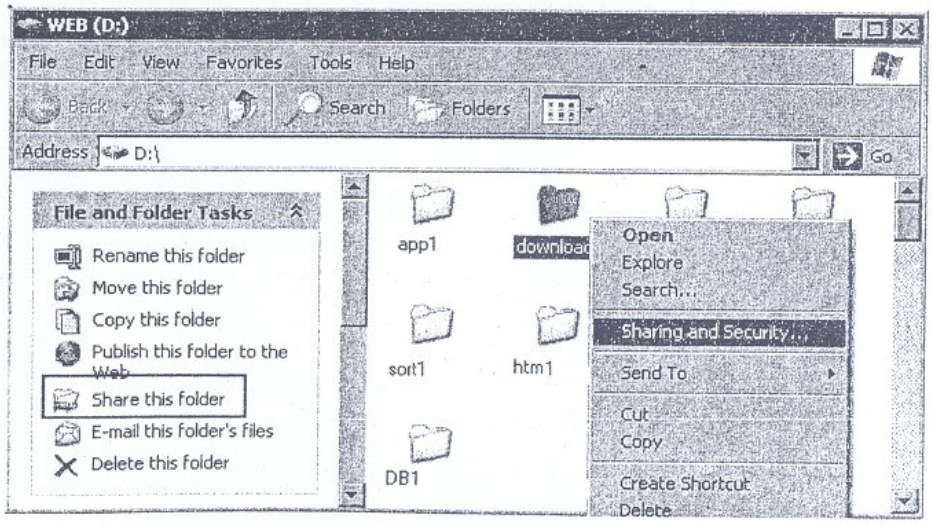


Figure 12: Web (D :)

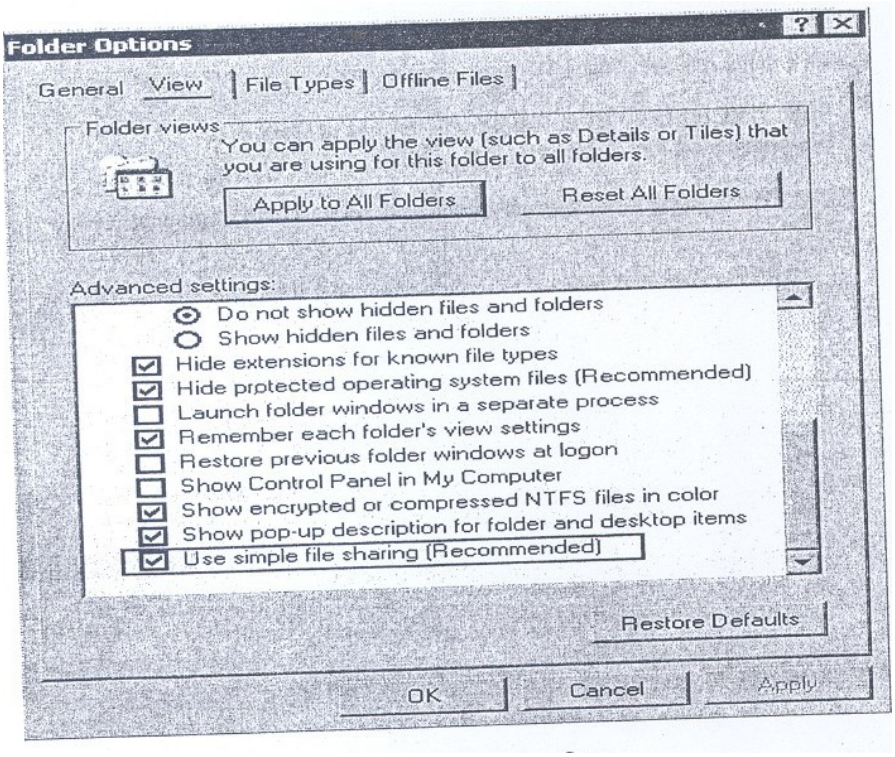


Figure 13: Folder Option Screen

The process of sharing a disk is identical *(Figure* 14) to the procedure used on Windows NT4 and Windows 2000.

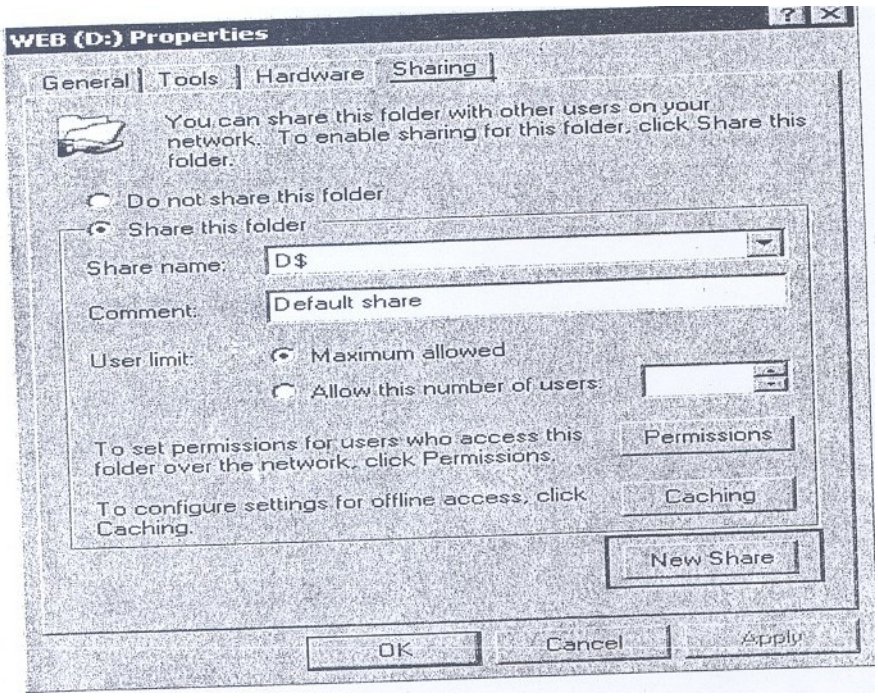


Figure 14: Web (D :) Properties

Enter the name of the share, as to be used on the network and as to be displayed in the Network Neighborhood as given in the above screen *(Figure* 15).

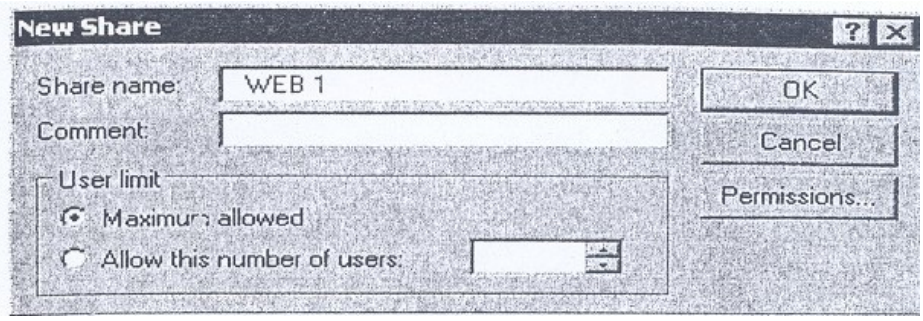


Figure 15: New Share Screen

By default all users in a network have access for a share, Even this group can be reduced *(Figure* 16).

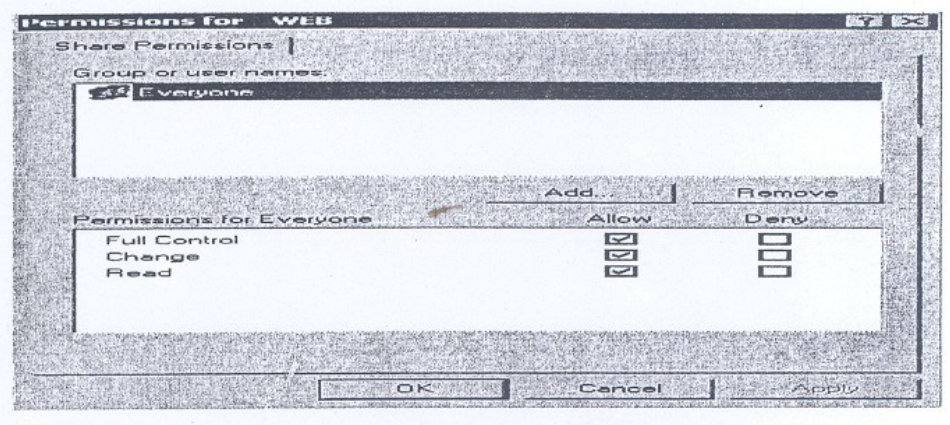


Figure 16: Shore Permission Screen

To view/modify the permissions or to remove the sharing you can select the share names from the drop down list as shown in *Figure* 17:

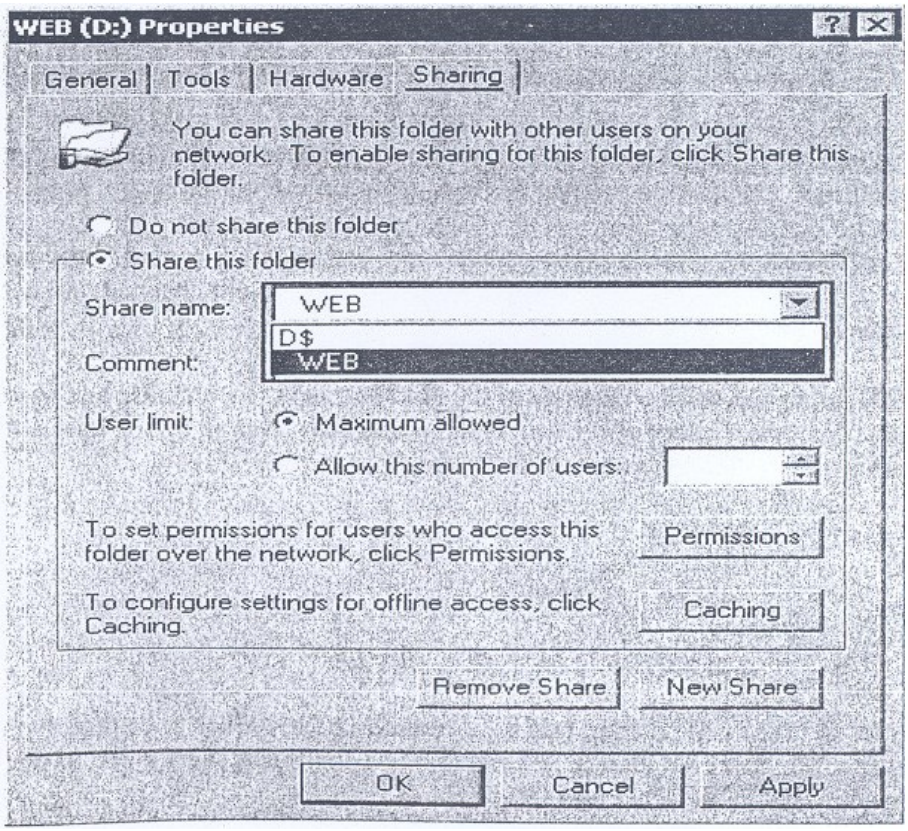


Figure 17: Share Permission Screen 2

Then the following screen *Figure* 18 shows Files and the Hard disk drives on the shared network.

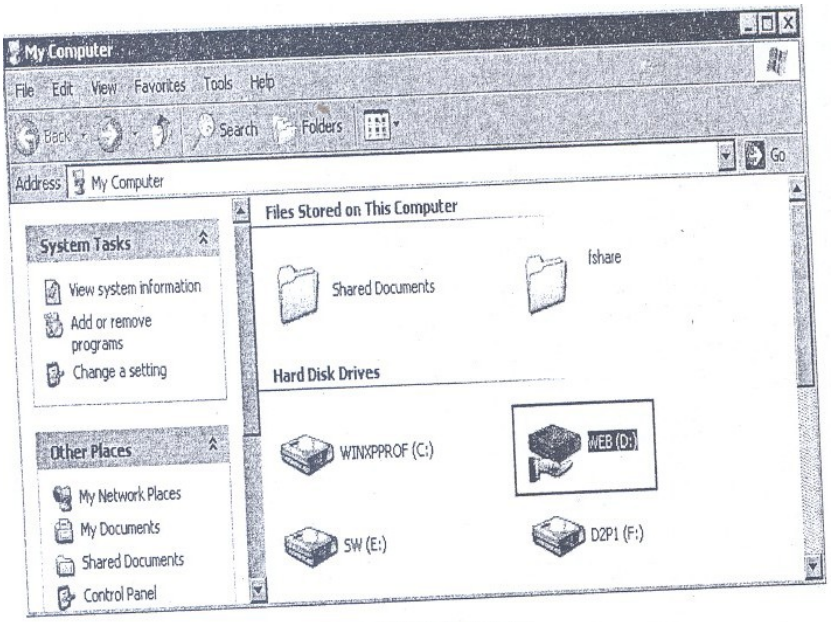


Figure 18: My Computer

While sharing a drive system warning is invoked that sharing an entire drive is not recommended but sharing the entire drive is recommended for several situations as well. But under circumstances, access should be given to everyone group (as shown in *Figure* 16), but doing so makes the shared drive highly vulnerable.

3.4 Enabling Offline File Features

Offline file allows users to keep copies of network files on a local machine. When say not connected to network, the user can use cached copy. When the user again reconnects to the network, offline file is synchronized with the online copy.

If changes have been made to the offline copy, offline file is copied to the network copy. If network version is changed but offline copy has not changed, the online copy is copied over the users' offline versions.

If both online and offline versions of the file have changed, a dialog box appears that lets the user select either of the two versions and also gives an option to retain both the versions of different filenames with the same name.

This feature is useful for:

1. Users working on a network.
2. Mobile Users
3. Users with an unreliable network connection

In order to make a file offline **fast user switching** feature has to be disabled first. This new Windows XP feature lets one or more additional users logon to the local computer without the other users logging off

This **fast user switching** option is to be turned off first before making a network fi1e offline.

1. From Start, Control Panel chooses open folder option.
2. Select offline file option
3. Here select drive, ok.

While working with offline files (in Windows XP environment) following options can be set:

1. **Synchronize all offline files when logging on:** If the users choose this option it synchronizes all files as the user logs on to the network.

2. **Synchronize all offline files before logging off:** This is by default i.e. before logging off all files is synchronized. This option makes sure that all users' files are synchronized before logging off from the networks. For most users, this option is the best while working with offline files.

3. **Display a reminder every x minutes**: A balloon reminder appears in the notification areas, when the user is working offline. By default, this message appears every hour. This time interval can be adjusted.

4. **Create an offline File shortcut on the Desktop**: In order to make shortcut to the Offline files folder on your desktop you can easily access any offline fi1es.

5. **Encrypt offline files to secure data:** This option facilitates the encryption of files on the local hard disk.

6. **Amount of disk space to use for temporary offline files:** This option lets the user control the amount of disk space that is allocated for temporary offline files.

4.0 CONCLUSION

This unit has introduced you to windows XP deliberations. Having gone through this unit, it is expected you will be able to describe windows XP networking features, file sharing features, folder sharing, file encryption, disk sharing features and offline features.

5.0 SUMMARY

Windows XP provides networking features that are capable of supporting a wide range of networks. In this unit Windows XP networking has been discussed, since TCP/IP is the de facto protocol for the Internet so it is also considered the favoured protocol for Windows XP machines. Windows XP does not support NetBEUI. File sharing; diskshal1ng folder sharing is very much similar to Windows and Windows 2000 environment. Also supported with this network operating system is file Encryption. Offline features are very useful for mobile users. And a window XP does support many offline features.

6.0 TUTOR-MARKED ASSIGNMENT

1) Right click the encrypted file and choose properties.
2) On the General tab, click advanced button, then click details button in Advanced Attributes Dialog Box.
3) In encryption Details Dialog box, click the Add for multiple users.
4) In select user's dialog box select the additional users and then click ok.
5) ---------- allows users to keep copies of network files on a local machine.
6) By Default windows XP computers contain ---------- file-sharing feature.
7) By Default ---------- is the file system for fixed storage in Windows XP.
8) Command line option in any environment lets the user interact with the ----------
9) Which operating systems support NTFS file system? Two computers are connected using a Local Area Network; Machine A is running on a 98 second Edition with FAT 32 file system. Machine B is running on XP Pro with NTFS file system. Will the Machine A be able to view and access files on XP which are shared. Assume ideal situations with no group policies. Also answer. If not why?
10) Mrs. Smith had Windows XP Pro on her Office desktop. She had some critical data on his computer as password protected and secure. Due to some error, she called a technician, who did a parallel installation of Windows XP on different folder and removed the initial installation of XP. Will Mrs. Smith still be able to access the shared File/Folders (assuming no recovery systems installed)? If not, can you enable it? Also what difference had it been if we had FA T32 or XP Home Edition and why?
11) Mr. Smith wants to computerize his office. He has a Medium Scale business with plans of growing in near future. What type of operating system and network structure would you design for them?

7.0 REFERENCES/FURTHER READINGS

www. microsoft.com.

Survey of Operating System. John Holcombe & Charles Holcombe, Tala McGraw Hill.

MODULE 4 SECURITY AND MANAGEMENT

UNIT 1 SECURITY CONCEPTS

CONTENTS

1.0 INTRODUCTION

Computer Security can be defined as technological and managerial procedures applied to computer systems to ensure the availability, integrity, and confidentiality of the information managed by the computer. It means the protection of Integrity, Availability and

Confidentiality of Computer Assets and Services from associated Threats and vulnerabilities.

Security is divided into two categories; (a) computer security and (b) network security. In generic terms, computer security is the process of securing a single, standalone computer; while network security is the process of securing an entire network of computers.

a) **Computer Security:** Technology and managerial procedures applied to computer systems to ensure the availability, integrity, and confidentiality of the data managed by the computer.

b) **Network Security:** Protection of networks and their services from unauthorised modification, destruction, or disclosure and provision of assurance that the network performs its critical functions correctly and there are no harmful side effects.

The major weaknesses in a computer system pertain to hardware, software, and data. However, other components of the computer systems may also be targeted.

2.0 OBJECTIVES

After going through this unit you should be able to:

understand the threats to computer security
understand what causes these threats
understand the various security techniques.

3.0 MAIN CONTENT

3.1 Goals of Computer Security

The goals of computer security are integrity, confidentiality, and availability of the information managed by the computer system. The relationship among the three is shown in *Figure* 1.

3.1.1 Integrity

The data Integrity in computer security deals with the knowledge that data has not been modified. Data Integrity is related to data accuracy, but integrity and accuracy are not the same. For example, if information is entered incorrectly, it will remain incorrect. So, it is possible to have Data Integrity without Data Accuracy.

Integrity means preventing unauthorised modification. To preserve the integrity of an item means that the item is unmodified, precise, accurate, modified in an acceptable way by authorised people, or consistent.

3.1.2 Confidentiality

Confidentiality means preventing unauthorised access; it ensures that only the authorised person accesses the computer system. Not all data available on the computer falls in the category of confidential data. There is data that can be made public and there is data that is considered sensitive. It is this critical or sensitive data that will require confidentiality. Data confidentiality cannot be enforced unless data integrity is present. The following items could require data confidentiality: credit card files, medical records, personnel data, mission-critical data, and R&D data etc.

3.1.3 Availability

There is no point in making the computer system so secure that no users can access the data they need to perform their jobs effectively.

The system should be accessible to authorised persons at appropriate times.

A computer system is available if:

The response time is acceptable
There is a fair allocation of resources
Fault tolerance exists
It is user friendly
Concurrency control and deadlock management exists. Terms like concurrency control and deadlock management will be discussed in the operating system course.

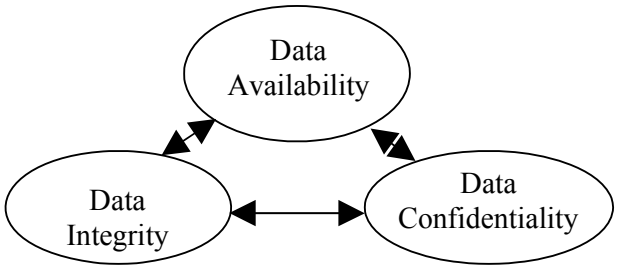


Figure 1: Relationship between Confidentiality, Integrity, and Availability

3.2 Security Problem and Requirements

Protection of information has been a major challenge since the beginning of the computer age. The computer security problem has grown with the computer industry; the computer itself was not really part of the security problem or its solution.

Connecting computers introduces a need for communication security (often utilising cryptography) to prevent the possibility of an attack. Connecting computers gives them greater accessibility, which increases computer security problems.

Computer security attempts to ensure the confidentiality, integrity, and availability of the computing system's components. The principal components of a computing system subject to attacks are: hardware, software and data. These three components and the communications among them is tile basis of computer vulnerabilities.

Attackers can devise attacks that exploit these vulnerabilities. There are basically four kinds of attacks on computing systems: interception, interruption, modification, and fabrication. These terms will be explained later.

One of the ways to identify security problems is by means of risk analysis. Risk analysis involves determining:

What you need to protect,
What you need to protect it from,
And how to protect it.

It is the process of examining all of your risks, and ranking those risks by level of severity.

There are three major steps in risk analysis, namely:

Identifying the assets (what are you protecting)
Identifying the threats (against what)
Identifying impact.

3.2.1 Identifying the Assets

List all the things that are subject to security threats. These include:

Hardware: CPUs, boards, keyboards, terminals, workstations, personal computers, printers, disk drives, communication lines, terminal servers, routers, hubs, gateways, servers, modems, etc.

Software: source programs, object programs, utilities, diagnostic programs, operating systems, communications program, firewall software, IDS (Intrusion Detection System) software etc.

Data: during execution, store on-line, archive off-line, backup, audit logs, databases, in transit over communication media etc.

People: user, people needed to run systems.

Documentation: on programs, hardware, systems, local administrative procedures.

Supplies: paper, forms, ribbons, floppy diskettes, magnetic media.

Based on the above, asset inventory can be created with the following component for it each asset:

Designated owner
General support system or critical/major application
Physical/logical location.

3.2.2 Identifying the Threats

Once the assets requiring protection are identified, it is necessary to identify threats to these assets. The threats can be then evaluated to determine what potential for loss exists.

There are two basic type of threats: accidental threats and intentional threats.

Accidental threats can lead to exposure of confidential information or causing an illegal system state to occur due to modification of information. An intentional threat is an action performed by an entity with the intention to violate security. And this includes destruction, modification, fabrication, interruption or interception of data.

In general, threats to an asset should be considered in terms of the availability, confidentially and integrity of the asset. The possible threats to a computer system can be.

Unauthorised Access
Disclosure of information
Denial of service.

3.2.3 Identifying the Impact

After identifying the assets and threats, the impact of security attack should be assessed. The process includes the following tasks.

Identifying the vulnerabilities of the system;
Analysing the possibility of threats to exploit these vulnerabilities;
Assessing the consequences of each threat;
Estimating the cost of each attack;
Estimating the cost of potential counter measure, and
Selecting the optimum and cost effective security system.

The consequence of a threat materialised in an organisation could result in one or more impacts. For example, an impact can be:

Infringement of privacy
Financial loss
Disruption of activities.

3.3 Threats and Vulnerabilities

With the rise of multiprogramming, the several aspects of a computing system requiring protection are system software, memory, sharable I/O devices such as disk, printers, tape drivers, shared programs/procedures, networks, shared data, flies, and execution environment. A threat is a set of instances that has the capability 9f causing loss or harm to the computer system. There are many threats to a computer system and can be (a) Human initiated, (b) Computer initiated, and (c) Natural disasters like flood or earthquake.

A threat can be accidental or deliberate and the various types of security breaches can be classified as (a) interruption, (b) interception, (c) modification and (d) fabrication.

Interruption: An asset of the system becomes lost, unavailable, or unusable.

- Malicious destruction of a hardware device
- Deletion of program or data file
- Malfunctioning of an Operating system.

Interception: Some unauthorised entity can gain access to a computer asset. This unauthorised entity can be a person, a program, or a computer system.

- Illicit copying of program or data files
- Wiretapping to obtain data.

Modification: Some unauthorised party not only accesses but also tampers with the computer asset.

- Change in the values in the database
- Alter a program
- Modify data being transmitted electronically
- Modification in hardware.

Fabrication: Some unauthorised party creates a fabrication of counterfeit object of a system. The intruder may put spurious transaction in the computer system or modify the existing database.

An attacker needs three things (1) method, (2) opportunity and (c) motive.

A method: It comprises the tools, skills, knowledge etc.
Opportunity: Opportunity means the right time and right access to perform the attack.
Motive: Motive is the reason to carry out the attack.

A **threat** can be blocked by control of vulnerability. We can use a control as a protective measure. A control can be in action, device, procedure and technique that limit or eliminate vulnerability.

A computer system has three valuable components as pointed out earlier: hardware, software and data. Vulnerability is a weakness in the system. This weakness may be exploited by threats causing loss/damage or harm to the system. Vulnerability does not cause any harm until exploited. It can be a weakness in: (a) Procedures, (b) Design and (c) Implementation.

The various vulnerability examples are: insufficient security training, lack of security l' awareness, inadequate recruitment procedures, insufficient preventive maintenance, lack of identification and authentication mechanisms, transfer of password in readable form (clear text), unprotected public network connections, poor password management, well-known flaws in the software, unsupervised work by external staff, no security policy, exposed/unprotected communication lines, poor cable joint, inadequate system management, no audit-trail, wrong allocation of access rights or permissions, lack of documentation and dialup connections, etc.

The computing system vulnerabilities are:

Software vulnerabilities: software vulnerability can be due to interruption, interception, modification, or fabrication. The examples of software vulnerabilities are: (a) destroyed/deleted software, (b) stolen or pirated software, (c) unexpected behaviour and flaws, (d) non-malicious program errors, (e) altered (but still run) software.

Hardware vulnerabilities: hardware vulnerability is caused due to interruption (denial of service), modification, fabrication (substitution) and interception (theft).

Data vulnerabilities: Data vulnerability is caused by interruption (results in loss of data), interception of data, modification of data and fabrication of data.

Human vulnerabilities: The various human generated vulnerabilities are break-ins, .virus generation, security violation, inadequate training.

3.4 User Authentication

Authentication in a computer system uses any of three qualities to authenticate the user:

Something the user knows, like password, PIN numbers; pass phrases, a secret handshake etc,
Something the user has: Identity badges, physical keys, a driver's license, or a uniform.
Something the user is: This is based on the physical characteristic of the user (Biometrics), such as a finger print, face recognition, voice recognition etc.

Two or more methods can be combined for more solid authentication; for example, an identity card and PIN combination.

The computer system needs a system in place to be sure that only authorised users have access to its resources. On the computer system, one of the critical areas of security is who has access to what.

There are two types of access control that can be implemented:

Mandatory Access Control (MAC): MAC is an access control policy that supports a system with highly secret or sensitive information. Government agencies typically use a MAC.
Discretionary Access Control (DAC) : DAC is an access control policy that uses the identity of the user or group that they belong

to allow authorised access. It is discretionary in that the administrator can control who has access, to what and what type of access will they have, such as create or write, read, update, or delete.

Authentication occurs when a user provides the requested information to an authentication verification authority. The traditional method of authentication is to provide a password.

To increase the level of reliability, biometric authentication can be introduced. The user is not only identified digitally, but by their physical characteristics such as fingerprint scan, iris scans or hand geometry.

Authentication Token: It is a portable device used for authenticating a user. The tokens are devices that operate by using systems such as:

Hardware Tokens

Challenge and response: It is an authentication technique using a calculator type of token that contains identical security keys or algorithms as Access Server, which sends an unpredictable challenge to the user, who compu1es a response using their authentication response token.

Time-based challenge response Token: The Time-based Token utilises an authentication method where the security token and server use an identical algorithm. To gain access, the user takes the code generated by the token and adds his or her user name and PIN to create a pass code. The pass code is combined with a seed value and the current time, encrypted with an algorithm and sent to the server. The server authenticates the user by generating its own version of the valid code by accessing the pre-registered PIN and using the same seed value and algorithm for validation.

Software Token

If an organisation does not wish to purchase hardware tokens, it may opt for a software type instead. A software token is an authentication process using portable devices such as a Palm Pilot, Palm PC, or wireless telephone to carry the embedded software.

3.5 Security System and Facilities

Security controls should be installed and maintained on each computer system or computer node to prevent unauthorised users from gaining entry to the information system and to prevent unauthorised access to data.

System software and resources should be accessible after being authenticated by access control system.

3.5.1 System Access Control

Access to information system resources like memory, storage devices etc. sensitive utilities and data resources and programme files shall be controlled and restricted on "need-to-use" basis.

The access control software or operating system should be providing features to restrict access to the system and data resources. The use of common passwords such as "administrator" or "president" or "game", etc, to protect access to the system and data resources should be avoided.

Guidelines and procedures governing access authorisation shall be developed, documented and implemented.

Each user shall be assigned a unique user ID

Stored passwords shall be encrypted using internationally proven encryption techniques to prevent unauthorised access.

Automatic time-out for terminal inactivity should be implemented.

Audit trail of security sensitive access and actions shall be logged.

Audit trails must be protected against modification or deletion.

Activities of all remote users shall be logged and monitored closely.

The startup and shutdown procedure of the security software must be automated.

Sensitive operating system files must be protected using proven tools and techniques.

3.5.2 Password Management

Certain minimum quality standards for password shall be enforced. The following, control features shall be implemented for passwords:

Minimum of 8 characters without leading or trailing blanks;

Shall be different from existing passwords;

To be changed at least once every 90 days and for sensitive systems it should be changed every 30 days;
Should not be shared, displayed or printed;
Password retries should be limited to a maximum of 3 attempted logons after which the user ID shall then be revoked for sensitive systems;
Passwords, which are easy to guess, should be avoided;
Password shall always be of encrypted form to avoid disclosure, and
All passwords must be resistant to dictionary attacks and all known password racking algorithms.

3.5.3 Privileged User Management

The following points must be taken into account while granting privilege to users.

Privileges shall be granted only on a need-to-use basis.
Login available only from console.
Audit log should be maintained.

3.5.4 User Account Management

Procedures for user account management should be established to control access to application and data. It should include:

Should be an authorised user.
A written statement of access rights should be given to all users.
A formal record of all registered users shall be maintained.
Access rights of users who have been transferred, or left the organisation, shall be removed immediately.
A periodic check/review shall be carried out for redundant user accounts and access right that is no longer required.
Redundant user accounts should not be reissued to another user.

3.5.5 Data and Resource Protection

All information shall be assigned an owner responsible for integrity of data and resource. This will help in protection of data and resources to a great extent. And this assignment of responsibility should be formal and top management must supervise the whole process of allocation of responsibilities.

3.5.6 Sensitive System Protection

Security token/smart cards/bio-metric technologies such as iris recognition, finger print verification technologies, etc, shall be used to complement the usage of password to access the computer system.

Encryption should be used to protect the integrity and confidentiality of sensitive data. In this unit we will discuss various techniques used in the protection of sensitive computer systems and networks.

Data Backup and Off-Site Retention

Backup procedures shall be documented, scheduled and monitored.

Up-to-date backup of critical items shall be maintained. These items include: data files, utilities/programmes, databases, operating system code, encryption keys, documentation, full/incremental backup frequencies as per schedule.

Firewall

The firewall is the first line of defense for any computer system or network. All packets that enter the network should come through this point. A modern firewall is a system of applications and hardware working together. A sophisticated firewall performs a combination of packet filtering, network address translation (NAT), and proxy services. These applications are depicted in *Figures* 2, 3 and 4 respectively.

Firewalls have two general methods of implementing security for a network. Although variations between these two exist, most modifications belongs to one or the other of the following:

- packet filtering and
- proxy server (Application Gateway)

Packet Filtering were designed to look at header information of the packet. Packet Filtering, shown in *Figure* 2, was the first type of firewall used by many organisations to protect their network. The general method of implementing a packet filter was to use a router. These routers had the ability to either permit or deny packets based on simple rules.

Proxy Servers use software to intercept network traffic that is destined for a given application. The proxy server, shown in *Figure* 3, recognizes the request, and on behalf of the client makes the request to the server.

In this, the internal client never makes a direct connection to the external server. Instead, the proxy functions as man- in-the-middle and speaks to both the client and server, relaying the message back and forth. The addition of proxy server capabilities added to the firewalls created a much more solid security product than a pure packet filter. Proxy software can make .decisions based on more than the header information of a packet.

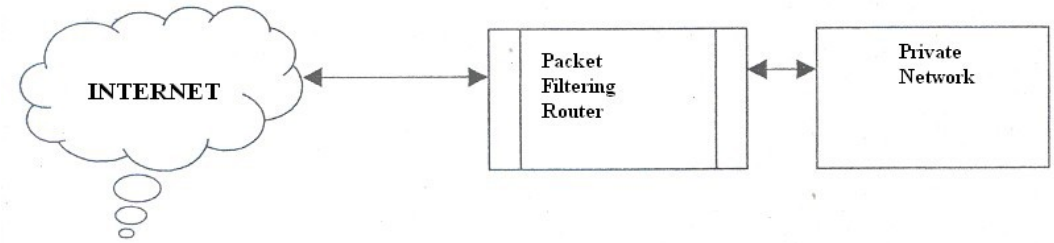


Figure 2: Packet Filtering Router

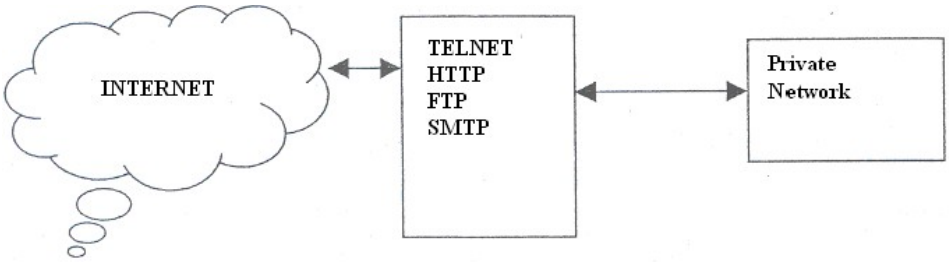


Figure 3: Application level gateway or Proxy server

A firewall can have a negative impact on the network by blocking access to the desired resources. This is due to improper configuration of a firewall that makes the desired resource unavailable. Additionally, if an ordinary PC has been configured to be the firewall (a multi-homed computer) it may not have the internal speed to perform all the functions of the firewall fast enough, resulting in increased latency.

Encryption

Central to all security mechanism
Confidentiality of data
Some protocols rely on encryption to ensure availability of resources.

The encryption process as a whole is taking data that is plain text (readable form), and using a mathematical technique to make the text unreadable. The receiver then performs a similar technique to decrypt the message. The process of encrypton and decrypton.is shown in *Figure* 4.

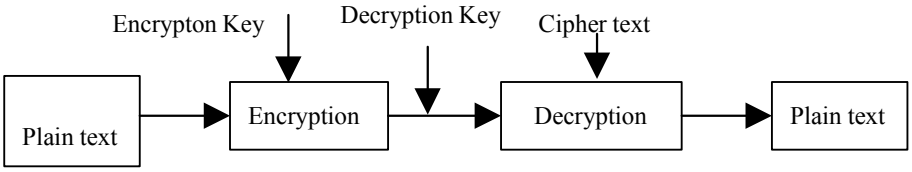


Figure 4: Encryption Decryption Mechanism

The performance hit is much more obvious in encryption. If the data packets are encrypted, the information that must be transmitted is bigger, and more bandwidth will be required. Additionally there will be more overheads on devices for performing encryption and decryption.

The computer that is asked to perform encryption and decryption must be able to handle extra workload.

Intrusion Detection System (IDS)

Intrusion Detection Systems are a combination of hardware and software systems that monitor and collect information and analyse it to detect attacks or intrusions. Some IDSs can automatically respond to an intrusion based on collected library of attack signatures. IDSs uses software based scanners, such as an Internet scanner, for vulnerability analysis.

Intrusion detection software builds patterns of normal system usage; triggering an alarm any time when abnormal patterns occur.

What IDS can do?

By using various techniques it attempt to detection of intrusion into a computer or network by observation of actions, security logs, or audit data.
Detection of break-ins or attempts via software systems that operate on logs or alert information.
Cannot stop crime, only prevent and provide evidence for investigations.

Software Controls

Internal program controls
OS controls
Development controls.

Hardware Controls

Locks or blocks limiting access
Hardware or smart card based encryption
Devices for user's authentication
Mechanism to control access to storage media.

Policies

The security policies and procedures must be properly implemented to ensure their proper use.

Physical Controls

Easy to implement, effective and less costly
Include locks on doors, guards at entry/exit points
Backup copies of critical software and data
Access Control
Media Control
Precautions against water and fire damage
Air conditioning
Physical site planning that minimizes the risk of natural disasters.

System Security

International Security Standards: Most computer vendors nowadays adopt international standards into building security facilities into their system.

Computer Virus

Computer should be equipped with updated virus protection and detection software.

Virus detection software *must* check storage drives both internal and external to the system on a regular basis.

All diskettes and software shall be screened and verified by virus scanner r software before being loaded onto the computer system.

Personnel Security

Personnel security is everything involving employees, who are potentialeleme1jts of breaches of security.

Hiring them Training
them Monitoring them
Handling their departure

Why personnel Security?

Most of the Security breaches are caused by people only like, break-ins, virus generation etc.
Statistics reveal that the most common perpetrators of significant computer crime are the legitimate users of the computer system.
Some studies show that over 80% of incidents are due to internal users.

Auditing

Auditing is a tedious process and requires a good eye for details.

Track everyone who logs on and off the computer system.
Audit movement 9fcriticalfiles, attempted deletion or access to mission-critical data.
Some of the common red flags to watch for in auditing are- multiple bad logon attempts, or same account trying to log in from many locations at the same time, and attempted shutdown of critical servers.

It is due to time-consuming process of reading the logs that many companies avoid auditing of log files. The organisation that takes the log files for granted will end up as one that is unable to inform the legal agencies that an incident has really happened

3.6 Cryptography

A cryptosystem is an algorithm, plus all possible plaintexts, cipher texts, and keys.

A cryptographic algorithm, also called a cipher, is the mathematical function used for encryption (E) and decryption (0). The key is a large number. The range of possible values of the key is called the key space. Both encryption and decryption use this key space.

$E_k[M] = C$
$D_k[C]=M$
or. $D_k[E_k[M)] = M$

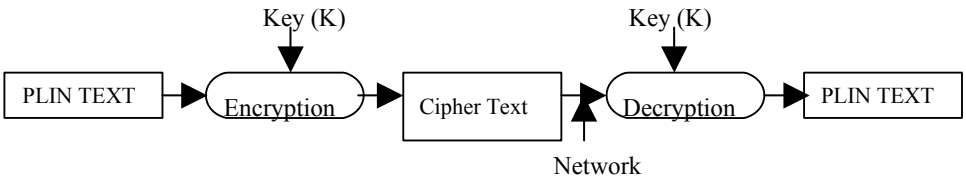


Figure 5: Encryption & Decryption using same key

Sometimes algorithms use a different encryption and decryption key. The encryption key K I is different from decryption key K2 (Figure 6).

$E_{k1}[M] = C$
$D_{k2}[C] = M$
Or $D_{k2}[E_{k1}[M]] = M$

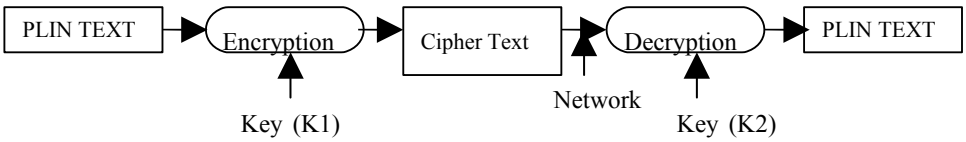


Figure 6: Encryption & Decryption using different keys

There are two general types of key-based algorithms: symmetric and public-key (asymmetric algorithm). The universally accepted modern method of electronic authentication is the one based on asymmetric cryptosystems. This is also known as public key cryptography, and is the basis for creating digital signatures. However rapid advancements and technological changes are challenging the supremacy of digital signatures as the only method of electronic authentication. Biometrics and dynamic signature analysis, among other technologies, are expected to be equally important in the years to come. It is also expected that some of the biometric techniques may prove to be more reliable and less susceptible to compromise than digital signature. In view of the pace of technological development, no single technology may prevail or a long time as the sole means of electronic authentication.

3.7 Intrusion Detection

ID stands for Intrusion Detection, which is the art of detecting inappropriate, incorrect, or anomalous activity. ID systems that operate on a host to detect malicious activity on that host are called host-based ID systems, and ID systems that operate on network data flows are called network-based ID systems.

Sometimes, a distinction is made between misuse and intrusion detection. The term 1 intrusion is used to describe attacks from the

outside; whereas, misuse is used to describe an attack that originates from inside the organisation's network. However, most people don't draw such distinctions.

The most common approaches to ID are statistical anomaly detection and pattern- matching detection.

Increased usage and consequent exposure have led to tI1t need to develop security components for the Web interface architecture. One such component is Intrusion Detection system. Intrusion Detection systems are however complex to implement, especially on large networks, because they generate 'Vast quantities of data and require significant configuration and management. IDS's come in many forms and implementation models. Some rule-based system relies on preset rules. Anomaly-based systems generate their own baseline overtime by building a database of recorded network usage. When network usage moves outside of the developed pattern, the IDS sounds an alarm.

In addition, IDS can be either host or network based or a combination thereof. A host based IDS is installed on and looks for potential malicious authority on a specific computer. A network based IDS records network traffic and scans for suspicious activity using sensors and agents installed throughout a network often through a tap off of a hub or a switch with a spanner port. It looks for malicious commandos, repeated failed login attempts, traffic peaking at odd hours or other evidence of possible mischief.

3.8 Computer-Security Classifications

The "Trusted Computer System Evaluation Criteria (TCSEC or orange book)" is the most widely accepted standard in the industry. The TCSEC model was developed based on a hierarchical model of security classifications.

The classes of systems recognised under the TCSEC are as follows. They are represented in the order of increasing desirability from computer system security point of view.

Class D (Minimal Protection)

A system with a Class D rating does not have to pass any tests to be rated as a class D system.

Class C1 (Discretionary Security Protection)

For a system to have C I level security, it must provide a separation of users from data. Discretionary access controls need to be available to

allow a user to limit access to data. Usersi11ust be identified and authenticated.

Class C2 (Controlled Access Protection)

For a system to have C2 level security, a user must be able to protect data so that it is available to only one user at a time. An audit trail that tracks access and attempted access to objects, such as files, must be kept. Further C2 security requires that all the residual data generated in temporary memory or register is erased.

Class B1 (Labeled Security Protection)

Systems at the B I level of security must have mandatory access control capabilities. Mandatory access controls limit access to objects based on the sensitivity of the information contained in the objects and formal authorisation of subjects to access information. The subject and objects that are controlled must be individually labeled with a security level. Labels must include both hierarchical security level such as "unclassified", "secret", and "top secret", and categories. Discretionary access control must also be present.

Class B2 (Structured Protection)

For a computer system to meet the B2 level of security there must be a formal security model. Covert channels used to transmit data must be constrained. There must be a verifiable top-level design, and testing must confirm that this design has been implemented. A security officer is designated to implement access control policies.

Class B3 (Security Domains)

The security of systems at B3 level is based on a complete and conceptually simple model. The capability of specifying access protection for each object, and specifying allowed subjects, the access allowed for each, and disallowed subjects must be included. A reference monitor for accessing user's requests and allows or disallows access based on access control policies, must be implemented. The system must be tamper proof and highly resistant to penetration. Auditing must be available for detection of security violations.

Class AI (Verified Design)

The capabilities of a class AI system are identical to those of a B3 system. However, the formal model for a class AI system must be formally verified as secure.

4.0 CONCLUSION

This unit has exposed you to the goals of computer security, security problems and requirements, threats and vulnerabilities, user authentication, security system and facilities, intrusion detection, cryptography and classification of computer – security. It expected that having some through this unit you can extensively discuss each of these.

5.0 SUMMARY

Computer security attempts to ensure the integrity, confidentiality, and availability of computer system. Computer systems are subject to attacks: hardware, software, and data. These three components and communication equipment associated with the computer constitute the basis of computer security vulnerabilities. Further, the people and systems interested in compromising a system can devise attacks that exploit the vulnerabilities. Four kinds of attacks on a computer system-interception, interruption, modification, and fabrication-have been discussed.

Controls can be applied at the level of the data, the programs, the system, the physical devices, the communication lines, the environment, and the personnel.

6.0 TUTOR-MARKED ASSIGNMENT

1) What are the goals of Computer Security?
2) Justify the following statement:

   "There is no confidentiality without integrity"

3) Identify computer assets in your organisation.
4) Identify threats to assets listed in progress] above.
5) Identify the impact of security attack listed in 2 above.
6) Distinguish between vulnerability and threat.
7) List any three recent computer security failures.
8) Do you currently apply any computer security control measures? If so, what? Against what attacks are you trying 10 protect?
9) Discuss various security systems and facilities.
10) What is computer-security classification?
11) What do you understand by symmetric and asymmetric cryptography?

7.0 REFERENCES/FURTHER READINGS

http://www.mit.gov.in/it-bill.aspInfonnation Technology Act 2000, India.

Cryptography and Network Security, Principles and Practice, William Stallings -SE, PE.

RSA Security's Official Guide to Cryptography, Steve Burnett and Stephen Paine -RSA Press.

http://www.cca.gov.in Controller of Certifying Authorities, Web Site.

Security in Computer, Charles P. Pfleeger and Shari Lawrence Pfleeger, Third Edition, Pearson Education.

UNIT 2 COMPUTER SECURITY

CONTENTS

1.0 INTRODUCTION

In the previous unit we described threats to computer security, what are the reasons for these threats and various security techniques. In this unit we will provide you specific guidelines for establishing a secure Microsoft Windows 2000. This includes hardening operating system, File System, Local Security, various services, default accounts, network services etc.

2.0 OBJECTIVES

After going through this unit you will be able to secure:

operating System
application Code
file System
local Security
services
default Accounts like guest and administrator
network services etc.

3.0 MAIN CONTENT

3.1 Hardening Operating System and Application Code

The first step towards hardening is to make sure that your OS and Applications are up-to-date with service packs and hotfixes.

Microsoft periodically distributes large updates to its as in the form of Service Packs. 1 Service Packs include all the major and minor fixes up. Service Packs should be used r in a test setup before being pushed into production due to the possibility of hidden or undetected bugs. If a test system is not available, wait a week or two after the release of a Service Pack, and monitor Microsoft Website for potential bug reports.

Microsoft also distributes intermediate updates to their operating systems in the form of Hotfix. These updates are usually small and address a single problem. Hotfixes can be released within hours of discovering a particular bug or vulnerability.

It is important to be aware that Service Pack and Hotfixes are Dot just applicable to Operating Systems. Individual applications have their own Service Pack and Hotfix requirements. The total security of the system requires attention to both operating system and application levels.

The process of discovering the appropriate Service Pack and hotfixes has been t automated since the release of Windows 2000. The following steps describe the f automated process of discovering and installing Service Packs and hotfixes to a Window 2000 system.

Open IE (Internet Explorer)
Go to Tools → Windows Update
When asked if you trust Microsoft, say Yes.

Windows update will take some time to analyze your system. You will then be prompted with a listing of Service Packs or Hotfixes for your system. Additionally the following websites provide the necessary information for manual updates.

Security Bulletins: http://www.microsoft.com/technetisecuritv/

Service Pack:
http/w.microsoft.com/windows2000/downloads/servicepacks/

Hotfixes:http://www.microsoft.com/windows2000/downloads/critical/

Microsoft Windows Security: http://www.Microsoft.com/securitv

3.2 Hardening File System Security

The second step is to make sure that your hard drive partitions are formatted with NTFS (NT File System). This file system is more secure than FAT or FAT32 schemes.

Step 1: Check Your Hard Drive Partitions

Log in as Administrator ' .Double click on My Computer
Right Click on each Hard Drive and Choose properties
General Tab will identify the File system type.

Step 2: Converting FAT or FAT32 Partitions to NTFS

Go to Start →RUN
Type cmd and click OK
At command prompt issue the following command convert drive /FS:NTFS/V
Hit return to run the command
Reboot the system.

3.3 Hardening Local Security Policies

The third step is to modify the default local security policy. While many system attacks take advantage of software inadequacy, many also make use of user accounts. To prevent such sort of vulnerability, "Policies" or rules define what sort of account/password "behaviour" is appropriate, what type of auditing is required. The configuration of user account policies is inadequate or disabled in a default installation.
Account policies answer the following:

How often do I need to change my password?
How long or how complex does my password need to be?

Auditing policies determine what kind of security transactions are recorded in the Security Event Log. By default, not is retained in the Security Event Log, so any attempt to compromise a system goes completely unrecorded. Logging events is critical for analysis in the aftermath of an intrusion incident.

The options given below can be set using the Local Security Policy editor on each individual computer. Nevertheless, Group Policy Configurations may override any changes made at the local level.

Local Security Policy Editor Tool

Go to Start → Programs → Administrative Tools → Local Security Policy
Expand Account Policies by clicking the + box
Select the appropriate category
Double-click the individual policy setting to make the appropriate changes for tile following.
Password Policy
Account Lockout Policy
Audit Policy
User Right Management
Security Options
When all settings have been configured, close the policy editor.

EVENT VIEWER

It is important to frequently check the Event Viewer to review log files for possible security concerns. You can access the Event Viewer by:

Go to Start → Programs→ Administrative Tools, →Event Viewer
Go to Start → Programs → Administrative Tools → Local Security Policy .Expand Account Policies by clicking the + box
Select the appropriate category
Double-click the individual policy setting to make the appropriate changes for the following:
Password Policy
Account Lockout Policy
Audit Policy
User Right Management
Security Options
When all settings have been configured, close the policy editor.

3.4 Hardening Services

The fourth step you take is to remove programs and services that are not required or needed. The more the number of applications that are installed on your system, the greater the risk of one of them containing a bug or security flaw.

The following is the list of services that can be disabled:

Alerter: This service makes it possible for Windows 2000 computers to "alert" each other of problems. This feature is generally unused.
Clipbook: The Clipbook Service is used to transfer clipboard information from one computer to another. This IS generally used In Terminal Services.
Fax Service: The Fax Service sends and receives faxes. It is generally unused.
Messenger: The messenger service works in conjunction with alerter service arid does
NetMeeting Remote Desktop Sharing: Net Meeting users have the option to share their desktops, and allow other NetMeeting users to control their workstation.
Telnet: The Telnet service allows a remote user to connect to a machine using command prompt._

3.5 Hardening Default Accounts

The fifth step is to change the default configuration of the administrator and guest account. In general, a prospective user must have a login name and password to access a Windows 2000 system. The default installation creates an Administrator and Guest account. By changing these accounts name, system security is greatly enhanced.

Steps: Configuring Administrator Account

Login as Administrator
Go to Start → Programs Administrative Tools → Computer management
Open Local Users and Groups
Click on the User Folder
Right-click the Administrator Account, and choose to rename it.
Make it a non- obvious name.
Right click this renamed Administrator account and select "set password."

The Guest account is disabled in Windows 2000 by default. Enabling the guest account allows anonymous users to access the system. If you share a folder, the default permission is that Everyone has full control. Since the Guest account is included in "Everyone", system security is compromised. A standard practice is to always remove the share permissions from "Everyone" and add them to "Authenticated Users."

Steps: Configuring: the Guest account

Login as Administrator
Go to Start → Programs Administrative Tools → Computer management
Open Local Users and Groups
Click on the User Folder
Right-click the Guest Account, and choose to rename it. Make it a non-obvious name.
Right click this renamed Administrator account and select "set password."

3.6 Hardening Network Activity

Next step is to install a host based antivirus solution and firewall/intrusion detection system. This step will provide an added level and you can configure TCP/UDP ports that can be accessed. This step is to ensure that undesired communications are not occurring on ports.

3.6.1 Malicious Code Type of Malicious Codes

Viruses
Worms
Trojan Horses
Back doors/Trap Doors
Logic Bombs
Bacteria/Rabbit

A. Viruses

A true virus is a sequence of code that is inserted into other executable code, so that when the regular program is run, the viral code is also executed. Viruses modify other programs on a computer, inserting copies of them.

Different Types of Viruses

Boot Sector Viruses: They infect either the DOS boot sector or the master boot records of the disk and execute during booting.

File Infectors: They attach themselves to executable files. These viruses are activated when the program is run.

Macro Viruses: They come attached with documents with macro (built in program). When the document is opened the viruses are activated.
Multipartite Viruses: They combine boot sector with file infector.

Polymorphic viruses: They alter themselves when they replicate so that anti-virus software looking for specific patterns known as signature, will not find them.

B. Worm

Worms are programs that can execute independently and travel from machine to machine across network connections.
They create a copy of themselves. This self-replication spreads worms like a floop in the networks causing slowdown and even breakdown of network communication services.

C. Trojan Horses

It is a code that appears to be innocent and useful but it also contains a hidden and unintended function that presents a security risk. It does not replicate but it can steal passwords, delete data, format hard disks or cause other problems.

D. Back Doors/Trap Doors

These are codes written into applications to grant special access to programs bypassing normal methods of authentication.
This special code used by programmers during debugging can be present in released version, either unintentionally or intentionally, and is a security risk.

E. Logic Bombs

Logic bombs are programmed threats that lie dormant in commonly used software for an extended period of time until they are triggered when some pre-conditions are met like a particular day etc. Logic bombs come embedded with some programs.

F. Bacteria/Rabbit

These codes do not damage files. Their purpose is to deny access to the resources by consuming all processor capability/memory/disk space by self replicating.

Damage Caused by Malicious Codes

The damage ranges from merely annoying to catastrophic (loss of data services, disclosure of information).

Loss of reputation or legal consequences for software firm if s/he inadvertently ships software containing any malicious code.

Who Creates or Writes Virus Code

Disgruntled employees
Spies
Experimenters
Publicity Hounds .Political activists

Steps for Protecting Your System from Viruses

Be careful about installing new software.
Never install binaries obtained from untrustworthy sources.
When installing new software, install it first on a non-critical system and test for bugs.
Periodically review all system start-up and configuration files for changes.
Turn off the automatic open on receipt feature from your email software.
Before opening any attachments first scan it using updated anti-virus software.
Regularly update anti-virus software engine and data files.
Select "Hide File Extension" option.
While opening any .doc file attachment using word disable macro.
Turn off visual basic scripting.
When not in use turn off the workstation or disconnect it from the network.
Take regular backup of critical data and system files.

3.6.2 Firewall

A firewall is a safeguard one can use to control access between a trusted and a less trusted on. A firewall is that:

Enforces strong authentication for users who wish to establish connection inbound or outbound.

Associates data streams that are allowed to pass through the firewall with previously authenticated users.

A firewall is a collection of hardware, software and security policy.

Without firewall, a site is more exposed to TCP/IP vulnerabilities, attacks from internet, and OS vulnerabilities.
Due to increased number of hosts in a network, it is difficult to achieve host security through imposition of control on individual hosts.
An intermediate system can be plugged between the private LAN (trusted network) and the public network (untrusted network).
All traffic in and out of the trusted network can be enforced to pass through this intermediate system.
This intermediate system is a good place to collect information about system and network use or misuse.
This intermediate system is known as firewall.

Why Firewall?

Protection from Vulnerable Services

- Filtering inherently insecure services like NFS/NIS.
- Routing based attacks

Controlled Access to Site System

- Prevent outside access except some special service like E-mail or HTTP

Concentrated Security

- All security measures like one time password and authentication software can be at the firewall as opposed to each host.

Enhanced Privacy

- Services like "finger" which displays information about user like last login, whether they have read e-mail etc., can be blocked.
- IP addresses of the site can be shielded from outside world by blocking DNS service.

Logging Statistics on Network use or Misuse

- All incoming and outgoing traffic from the Internet can be logged to provide statistics about the network usage. These statistics will provide the adequacy of control of firewall on network.

Policy Enforcement

- Provides means for implementing and enforcing a network control.

Limitations of Firewall

Restricted Access to Desirable Services

- It may block servict1s like TELNET, FTP, NF:S, etc. which user wants.
- Some network topologies require major restructuring from implementation of firewall.

Large Potential Back Door

- If modem access is permitted, attacker could effectively jump around the firewall.

Little Protection From Insider Attack

- Firewalls are generally designed to prevent outsider's attack.
- Cannot prevent an insider from copying data, etc.

Other Issues

- Firewall does not provide protection against users downloading virus- Infected program from Internet or from E-mail attachments.
- Potential bottleneck in throughput
- Firewall, if compromised, will be a disaster.

Primary Aspects

The primary aspects of a firewall are:

Firewall policy
Packet filters
Application Gateway
Advanced authentication mechanism.

Firewall Policy

The firewall policy directly influences the design, installation and use of the firewall system.

Higher Level Policy: The Higher level policy addresses the services that will be allowed or explicitly denied from/to the restricted network.

It is a subset of overall organisation's policy on security of its information assets.

It focuses on Internet specific issues and outside network access (dial-in policy, PPP connections, etc.).
It should be drafted before the implementation oft\1e firewall.
It should maintain a reasonable balance between protecting the network from known risks while still providing Internet access to the users.
Its implementation depends on the capabilities and limitations of the Firewall System.

Example

No inbound access from Internet but allow outbound access from the network.
Allow access from the Internet to selected systems like Web Server, Email Server, etc.
Allow some users access from the Internet to selected servers but after strong authentication.

Lower level Policy: The Low level policy describes how the Firewall actually goes about restricting access and filtering the services that are defined in the Higher-level policy.

The Lower level policy is specific to the Firewall and defines to implement the "Service Access Policy' already approved in Higher level Policy.
Generally implements one of the two basic design policies:

- Permit any service unless it is specifically denied
- Deny any service unless it is explicitly permitted. This option is stronger and safer but difficult to implement.

Packet Filter or Packet Filtering Gateways

One type of firewall is the packet filtering firewall. In a packet filtering firewall, the firewall examines five characteristics of a packet.

Source IP address
Source port
Destination IP address
Destination port
IP protocol (TCP or UDP)

Based upon rules configured into the firewall, the packet will be allowed through, rejected, or dropped. If the firewall rejects the packet, it sends a message back to the sender letting him know that the packet was rejected. If the packet was dropped, the firewall simply does not respond

to the packet. The sender must wait for the communications to time out. Dropping packets instead of rejecting them greatly increases the time required to scan your network. Packet filtering firewalls operate on Layer 3 of the OSI model, the Network Layer. Routers are a very common form of packet filtering firewall.

A packet filter rule consists of two parts: An Action Field (BLOCK or DENY) and a Selection criteria (PERMIT or ALLOW).

Example: Sample Basic Packet Filters rule set.

SI. No.	Protocol	Source Address	Destination Address	Source Port	Destination Port	Action	Description
1	TCP	Any	192.168.200.3	>1023	80	Permit	Allow inbound HTTP access to the host having IP address 192.168.200.3
2	TCP	Any	192.168.200.4	>1023	21	Permit	Allow inbound FTP control channel to the host having IP address 192.168.200.4
3	TCP	Any	192.168.200.4	Any	20	Permit	Allow FTP data channel to this host
4	UDP	Any	Any	53	>1023	Permit	Permit all inbound DNS resolution
5	Any	Any	Any	Any	Any	Deny	Cleanup rule blocking all traffic not included above.

Problems with Packet Filters

Packet filtering rules are complex to specify and difficult to test thoroughly.

Exception to packet filtering rules sometimes can be unmanageable.

Some packet filtering routers do not filter on the TCP/UDP source port, which can make the filtering rule set more complex and can open up "holes" in the filtering scheme.

Problem of IP Fragmentation: If fragmentation of IP packet occurs only the first fragment keeps the TCP/UDP header information of the original packet, which is necessary to make filtering decision. Some packet filters may apply rules on the first fragmented piece, which is not serious for inbound traffic. For outbound traffic, even if the first fragmented piece is dropped other may go out leaving a serious security threat.

Stateful Packet Filtering

An improved form of the packet filtering firewall is a packet filtering firewall with a stateful inspection engine. With this enhancement, the firewall 'remembers' conversations between systems. It is then necessary to fully examine only the first packet of a conversation.

A stateful inspection peeks into the payload of data of the IP packets and takes out the required information on which the filtering can be done. A stateful inspection maintains the state information about the past IP packets.

> For robust security, a firewall must track and control the flow of communication passing through it.
>
> For TCP/IP based services, firewall must obtain information from all communication layers.

State information, derived from past communications and other applications, are an essential factor in making the decision.

State Information

> Communication information from all layers in the packet.
> Communication derived from previous communications (Example: The outgoing "Port" command of an FTP session could be saved so that an incoming FTP data connection can be verified against it).
> Application derived state from other application. (Example: A previously authenticated user would be allowed access through the firewall for authorized services only).

Application Proxy Firewall

Another type of firewall is the application-proxy firewall. In a proxying firewall, every packet is stopped at the firewall. The packet is then examined and compared to the rules configured into the firewall. If the packet passes the examinations, it is re-created and sent out. Because each packet is destroyed and re-created, there is a potential that an application-proxy firewall can prevent unknown attacks based upon weaknesses in the TCP/IP protocol suite that would not be prevented by a packet filtering firewall. The drawback is that a separate application-proxy must be written for each application type being proxied. You need an HTTP proxy for web traffic, an FTP proxy for file transfers, a Gopher proxy for Gopher traffic, etc... Application-proxy firewalls operate on Layer 7 of the OSI model, the Application Layer.

Application Gateway Firewall

Application-gateway firewalls also operate on Layer 7 of the OSI model. Application- gateway firewalls exist for only a few network applications. A typical application- gateway firewall is a system where you must telnet to one system in order to telnet again to a system outside of the network.

Gateway interconnects one network to another for a specific application.
Gateway used in firewall configuration is an Application Level Gateway or a Proxy Server.
The function of application Gateway is application specific. If an application Gateway contains proxies for FTP and TELNET, then only those traffics will be allowed and other services are completely blocked.
Imposition of an application gateway breaks the conventional client/server model as each communication requires two connections one from the client and the other from the firewall to the server.

The Internet community often uses the term Bastion Host to refer to an exposed firewall system that hosts an application gateway.

Advantages of Application gateways:

Information Hiding: The application gateway is the only host whose name is made known to the outside systems.
Robust Authentication and Logging: All traffic can be pre-authenticated and logged to monitor the effectiveness of security policy.
Less Complex Filtering Rule: The packet filtering router needs only to allow traffic destined for the application gateway and reject the rest.

3.7 Fault Tolerant System

A Fault tolerant system is designed by using redundant hardware (hard disk, disk controller, server as a whole) to protect the system in the event of hardware failure. There are various techniques to do that:

SFT (System Fault Tolerance) Techniques

Disk Mirroring: Data is writer in two separate disks, which are effectively mirror images of the each other. The disk mirroring technique is depicted in *Figure* 1.

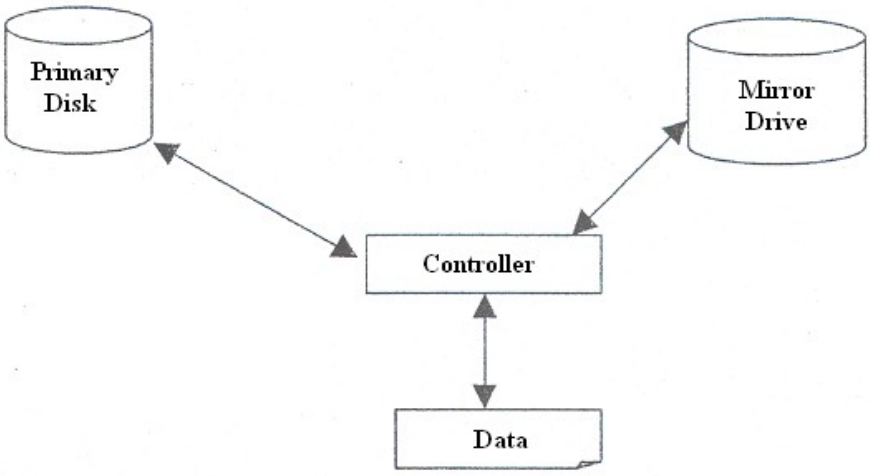


Figure 1: Disk Mirroring

Disk Duplexing: Disk duplexing, shown in *Figure* 2, implements separate controller for each disk.

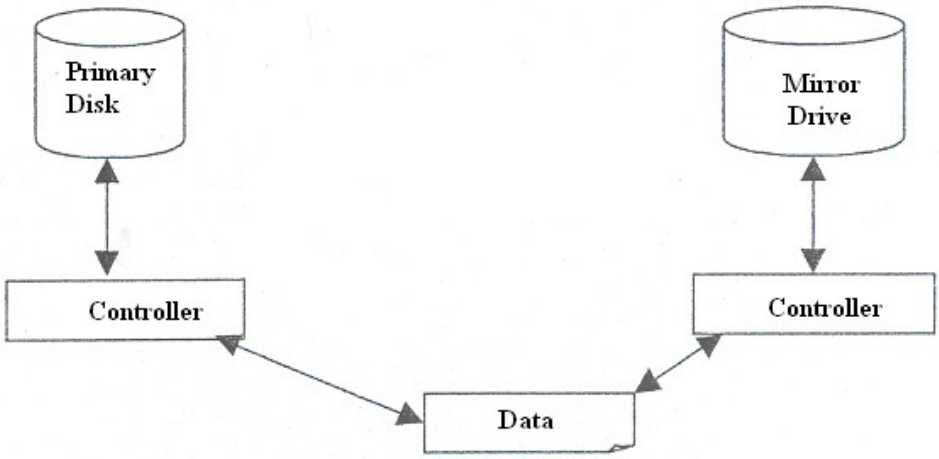


Figure 2: Disk Duplexing

RAID

The term RAID (Redundant Array of Independent Disks) was first coined by a research group at University of California, Berkeley, to describe a collection of disk drives (disk array), which can:

Collectively act as a single storage system
Tolerate the failure of a drive without losing data.
Function independently of each other.

The RAID advisory board defines RAID levels and the most common levels are numbered from 0 to 6, shown in *Figure:* 3. where each level corresponds to a specific type of fault tolerance.

RAID Level	Fault Tolerance
Level 0	Striping without parity
Level 1	Mirroring / duplexing
Level 2	Striping with ECC (Error Correction Code)
Level 3	Striping with a dedicated parity disk
Level 4	Independent data disks with shared parity disk
Level 5	Independent data disks with distributed parity blocks (striping with parity)
Level 6	Second parity

Figure 3: RAID Levels

Striping without Parity

Disk striping is a technique where data is divided into 64K blocks and spread in a fixed order among all the disks in the array. Because it provides no redundancy, this method cannot be said to be a true RAID implementation. If any partition in the set fails, all data is lost. It is used to improve performance by spreading disk I/O over multiple drives.

This strategy requires between 2 and 32 hard disks. It provides the best performance when used with multiple disk controllers. The technique is shown below in *Figure* 4.

Mirroring / Duplexing

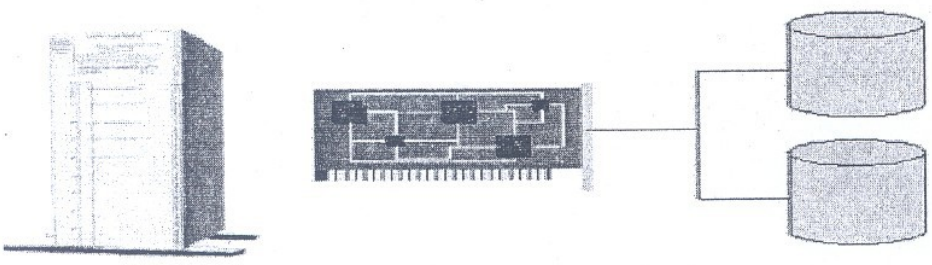


Figure 4: Drive Mirroring

Mirroring requires two hard disks and a single disk controller. It takes place at the partition level and any partition, including the boot/system partitions, can be mirrored. This strategy is the simplest way of protecting a single disk against failure.

In terms of cost per megabyte, disk mirroring is more expensive than other forms of fault tolerance because disk-space utilisation is only 50 percent. However, for peer-to- peer and modest server based LANs, disk mirroring usually has a lower entry cost because it requires only two disks. Stripe sets with parity (RAID levels) require three or more.

Data is written simultaneously to both partitions/disks.

Duplexing is simply a mirrored pair with an additional disk controller on the second drive. This reduces channel traffic and potentially improves performance. Duplexing is intended to protect against controller failures as well as media failures.

Striping with parity

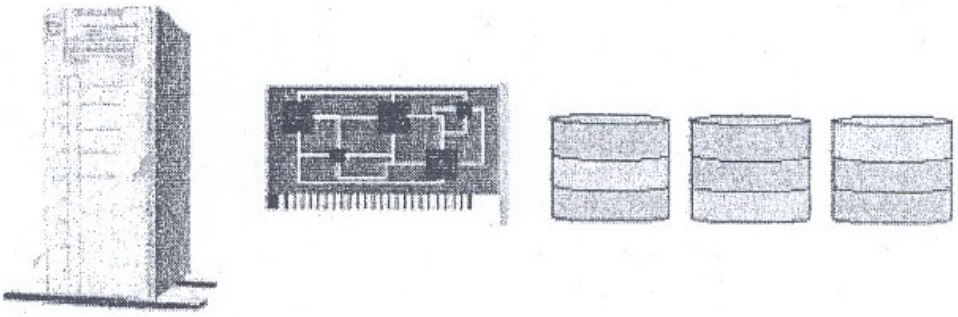


Figure 5: Striping with Party (or RAID 5)

Striping with parity (RAID 5) depicted in *Figure* 5, is the most common strategy for new fault tolerance designs. It differs from other levels in that it writes the parity information across all the disks in the array. The data and parity information are managed so that the two are always on different disks. If a single drive fails, enough information is spread across the remaining disks to allow the data to be completely reconstructed.

Stripe sets with parity offer the best performance for read operations. However, when a disk has failed, the read performance is degraded by the need to recover the data using the parity information. Also, al1 normal write operations require three times ~s much memory due to the parity calculation.

Striping with parity requires a minimum of three drives and up to thirty-two drives are supported. All partitions except the boot/system partition can be part of a stripe set.

The parity stripe block is used to reconstruct data for a failed physical disk. A parity stripe block exists for each stripe (row) across the disk. RAID 4 stores the parity stripe block on one physical disk, while RAID 5 distributes parity evenly across each of the disks in the stripe set.

Implementing RAM

It is possible to implement RAID using either hardware or software.

Hardware Solutions

Some vendors implement RAID level 5 data protection directly into hardware, as with disk array controller cards. Because these methods do not require software drivers, they generally offer performance improvements. In addition, some hardware implementations allow you to replace a failed drive without shutting down the system, The disadvantages of a hardware implementation are that they can be very expensive and may lock you into a single vendor solution.

SCSI control1ers can be purchased with dual interfaces and built-in logic to implement a hardware-level RAID system. This can be used with any operating system, even if the operating system itself is not RAID-aware.

Software Solutions

Both Windows NT Server and NetWare provide be option to set up software fault tolerance using standard disks and control1ers.

Mirroring Versus Stripe Sets with Parity

Implementing a fault tolerance strategy will require some trade-off depending on the level of protection required. The major differences between disk mirroring and striping with parity are performance and cost.

Overall, **disk mirroring** offers better I/O performance and has the advantage of being able to mirror the boot/system partition. Because mirroring utilises only 50% of available disk space, it tends to be more expensive in cost per megabyte. As hard-disk prices decrease, these costs will become less significant.

Disk striping with parity offers better read performance than mirroring, especially with multiple controllers. This is because the data is split among multiple drives. However, the need to calculate parity information requires more system memory and can slow down performance considerably. The cost per megabyte is much lower with striping because the disk utilisation is much greater.

Clustering

It is a collection of computers, which work together like a single system. If a computer in the cluster crashes other surviving computers can serve the client request.

A combination of clustering and disk mirroring can be used to provide a very secure system, in addition to maintaining integrity and high availability it gives scalability.

3.8 Backup and Ups

Why Backup?

The backup is required to recover valuable data and to restore system in the event of disaster due to:

User/System-staff error
Hardware/Software failure
Crackers/Malicious code
Theft
Natural Disaster
Archival of information

Types of Backup

Complete or Full Backup

- Every file on the source disk is copied.
- It clears the archive bits of the all the files of the source disk.
- Slowest but most comprehensive.
- Restoring from full backup is straightforward.

Incremental Backup

- Copies only those files for which the archive bit is set.
- Clears the archive bit after backup.
- Saves backup time and backup media.
- Restoration has to be done first from the full backup tapes from the incremental backup tapes in order of creation.

Differential Backup

- It is only the backup of the files, modified since the last full or incremental backup.
- It does not alter the archive bit setting.
- Takes more space than incremental backup.
- Restoration is simple, restore from the full backup and the latest differential backup.

CASE STUDY: Windows 2000 Backup Strategies

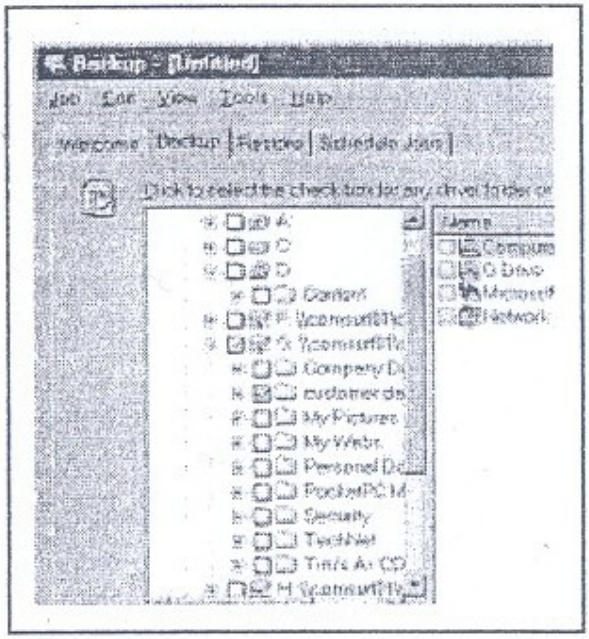


Figure 6: Windows 2000 Backup

One of the most important operations in a network system is the creation of a secure backup. Typically, backups take place using a tape system that has the advantage of high capacity, relatively low cost and portability. When you click on backup option, screen as displayed in *Figure* 6 will be presented to the user.

Backup Methods

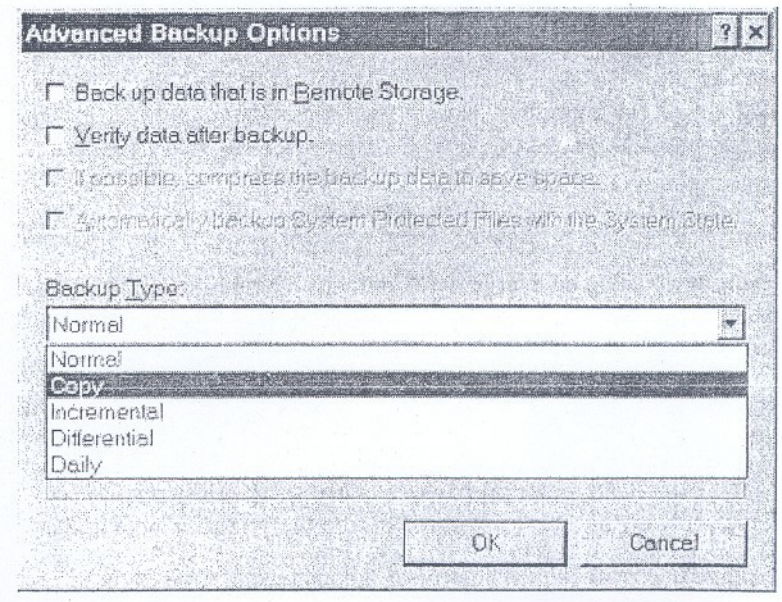


Figure 7: Choosing the Backup type

A backup may be performed using one of three methods as shown in *Figure* 7:

Full
Incremental
Differential

A full backup includes all selected files and directories while incremental and differential backups check the status of the archive attribute before including a file.

The archive attribute is set whenever a file is modified. This allows backup software to determine which files have been changed, and therefore need to be copied.

The criteria for determining which method to use is based on the time it takes to restore versus the time it takes to back up.

Assuming a backup is performed every working day, an incremental backup only includes files changed during that day, while a differential backup includes all files changed since the last full backup.

Incremental backups save backup time but can be more time-consuming when the system must be restored. The system must be restored from the last full backup set and then from each incremental backup that has subsequently occurred. A differential backup system only involves two tape sets when restore is required.

Table 1 summarises the three different backup types:

Table 1: Three different backup types

Type of backup	Data that will be backed up	Time for backup / restore	State of archive attribute
Full	All selected data regardless of when it has previously been backed up	High/low (one tape set)	Cleared
Incremental	New files and files modified since the last backup	Low/high (multiple tape sets)	Cleared
Differential	All data modified since the last full backup	Moderate/ moderate (no more than 2 tape sets)	Not Cleared

Doing a full everyday backup on a large network takes a long time. A typical strategy for a complex network would be a full weekly backup followed by an incremental or differential backup at the end of each day.

The advantage of using a **full daily backup** is that only one tape set is required to restore the system.

The advantage of an incremental backup is that it takes less time to back up but several tape sets may need to be restored before the system is operational.

The advantage of a differential backup is the balance of time for both restoring and backing up.

Example: Power Management in Windows 2000 as shown in *Figure* 8.

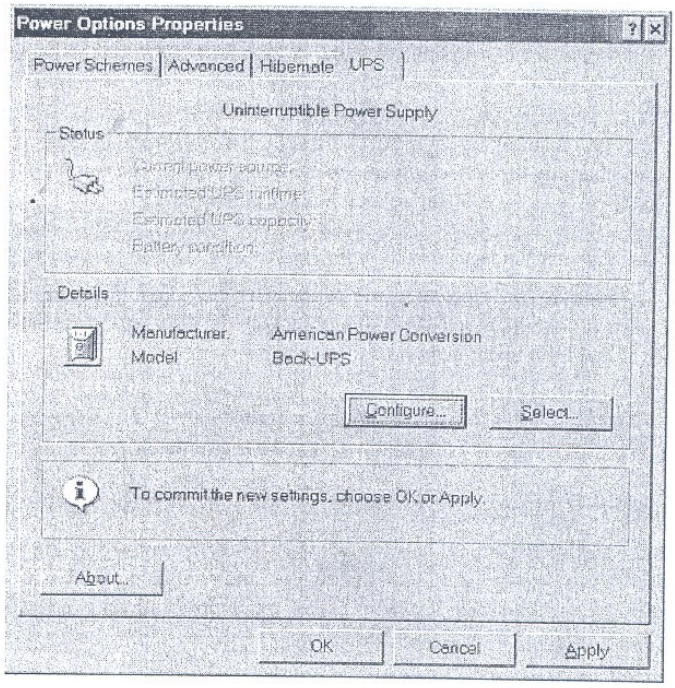


Figure 8: Selecting your UPS in Windows

UPS (Uninterruptible Power Supplies)

UPS (Uninterruptible Power Supplies) provide an alternative AC power supply in the event of power failure and also eliminate the effects of power surges and spikes.

Generally a UPS comprises the following:

A bank of batteries and associated charging circuit.
A DC-to-AC converter to generate AC voltage from batteries.
A switch over circuit to allow the UPS to take over from the (failed) supply.
Spike and surge protection circuitry.

Most UPS fall into one of the following categories:

Offline UPS

An offline UPS keeps the batteries charged all the time but does not operate the inverter until the power fails and the inverter starts and is switched into the power circuit.

Offline UPS are cheaper to build and do not dissipate as much heat as the online varieties but they have one drawback -switchover time.

It takes a small amount of time for an offline UPS to detect a power failure, start the inverter and switch it into the power circuit. This delay can be just a few milliseconds and is not usually 'noticed' by the equipment to which it. is connected. However, this is not always the case and some equipment "ill not work properly with an offline UPS.

Online UPS

An online UPS is constantly supplying power from the batteries and inverter, while at the same time, charging the batteries from the incoming supply. The benefit of this design is that there is no switchover delay when the power fails.

Choosing a UPS

Choosing the right type of UPS is relatively straightforward. The following guidelines assist the choice but should be used in conjunction with the information available from the equipment and UPS manufacturers.

Offline or Online

Check the type of UPS that is suitable for the equipment to be protected.

Power Rating

The maximum power rating (and hence cost of a UPS is determined by the battery specification and the power handling of the inverter and other circuitry. Each UPS is rated according to the maximum VA (power) they can supply without overloading.

To find out the required VA rating of a UPS

= Sum (Watt Used by Each Device) * 1.6

Operational Time

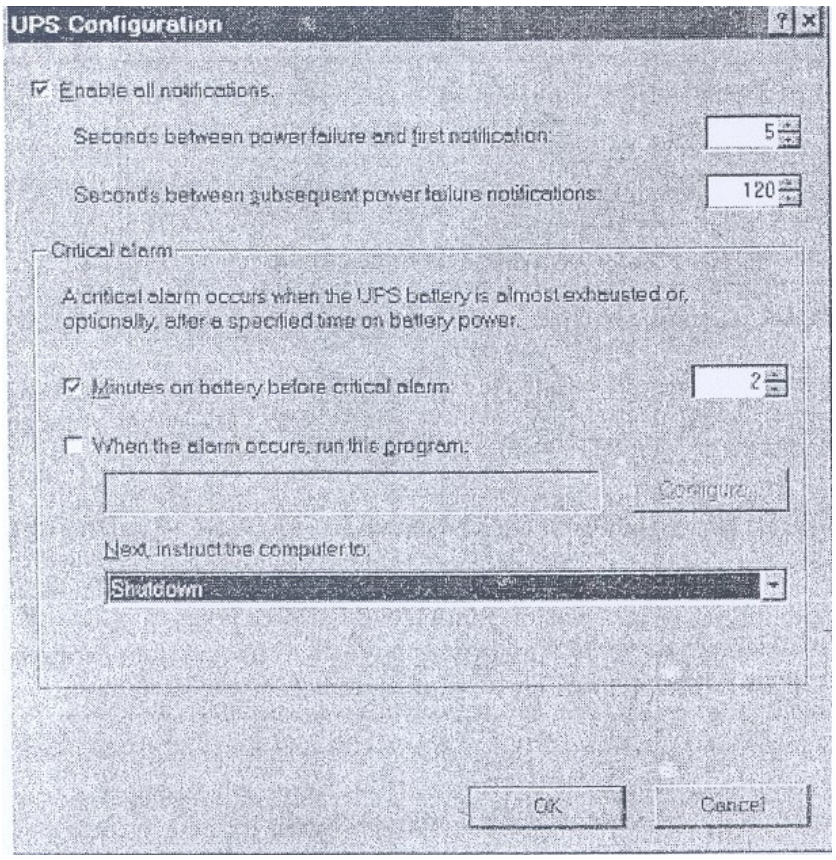


Figure 9: Configuring the UPS

The number of batteries within the UPS determines the amount of time for which it can generate and supply power (the 'up time'). Most vital computer systems require UPS power for at least five minutes. This gives the time needed for correct shut down in the event of a general power failure. The various options for configuring your UPS are shown in *Figure* 9.

Additional Considerations when Choosing a UPS.

UPS Monitoring

Some UPS's can be connected to their host system via a serial port or an add on card; the UPS can then alert the host system when there is a power failure or an impending problem such as 'battery power low'.

Network Monitoring

Some UPS can communicate with monitoring software such as SNMP (the Single Network Management Protocol) via a network connection.

4.0 CONCLUSION

In this unit, you have learnt specific guidelines for establishing secured Microsoft windows 2000 which includes security of the OS, file system, local security policies, various services, default accounts, network services, etc. you have also been taken through discussions about malicious codes and various security techniques like firewall, fault tolerance system, backups and UPS.

5.0 SUMMARY

With proper setting and hardening Operating System, Application Code, File System, Services, Network Service, Default Accounts, Virus Protection, and Proper backup strategies, we can secure our Windows 2000 System from known vulnerabilities and attacks. However, to counter new attacks and vulnerabilities, it is desired that the latest security measures should be implemented under expert guidance.

6.0 TUTOR-MARKED ASSIGNMENT

1) Describe the strategy for hardening your windows 2000 operating system.
2) List the steps for discovering and installing services packs and hotfixes to windows 2000 system.
3) Fill in the blanks:

   a) The first step towards hardening is to make sure that your OS and Applications are up-to-date with -------- and -------
   b) Service Packs should be used in a --------- before being pushed into production due to the possibility of hidden or undetected bugs.
   c) The total security of the system requires attention to both ---------- and ----------

4) What steps will you take for hardening your Windows file system?
5) List the steps for converting a FAT files system to NTFS file system.
6) List the steps for hardening default accounts (Guest and Administrator accounts).
7) List different types of malicious code.
8) List advantages and limitations of firewall.
9) Expand the following:

a) RAID b) UPS

10) Describe backup strategies for your system.

11) Wow will you select a UPS for your system.
12) Discuss and compare existing virus protection tools.

7.0 REFERENCES/FURTHER READINGS

Security Bulletins: http://www.microsoft.com/technet/security/

Service Pack:
http://www.microssoft.corn/windows2000/downloads/servicepacks/

Hotfixes :httl!://www.microsoft.com/windQws2000/downloadsicritical/

Microsoft Windows Security: http://www.Microsoft.com/security

UNIT 3 SECURITY AND MANAGEMENT-I

CONTENTS

10 INTRODUCTION

In this unit we will discuss the concepts and configuration required to secure Microsoft Windows computers and also examine everything from the foundational principles of Windows NT Security Management, up to the advanced issues of securing Windows 2000 machines running Active Directory. The unit address is abroad sweep of concepts of Windows Management Architecture and security related issues: Main Issues in Windows Security; Windows Resource Management; Windows 2000 Operating Systems.

Section 3 of this unit deals with "Main Issues in Windows Security and Management" and it covers the following areas; physical security management, logon security management, user/groups management, Windows NT domain model, domain controllers.

Section 4 of this unit deals with Windows resource security management and it covers areas like; files and folder management, files/folder permissions, printer management and Registry Management.

The most important, section 5, deals with the management of Windows 2000 operating system; Windows 2000 features, active directory, logical structure, physical structure, Windows 2000 DNS, Group Policy etc.

2.0 OBJECTIVES

After going through unit, you should be able to:

learn the management of Windows NT system
examine the fundamentals of the Management of Windows 2000 system.

The objectives of this unit include:

examine the various issues of Management of Windows NT 4.0.
study and manage Windows NT 4.0 Resources
examine the Windows 2000 Infrastructure.

3.0 MAIN CONTENT

3.1 Main Issues in Windows Security Management

In this section we will point on main issues in windows security management.

3.1.1 Physical Security Management

The main problem or issue of computer security is unauthorized physical access to a secure computer system and it is breech of computer security. If a computer is in a public area it should not contain any sensitive data.

The following steps should be taken to improve physical or local security: computer BIOS must have a password, and computer should be configured to boot from hard drive and not through floppy or any other external media. In Windows NT Server provides options to control local access or right to log on locally and this adds another layer of security on computer.

3.1.2 Logon Security Management

When a user logs on to a Windows NT machine, he is presented with an onscreen message or notice. This message must clearly state the intended use of the computer system. It is suggested that the banner should not have a greeting, or a welcome message. The main steps for creating a user account are given below:

Creating a User Account

1) Log on as an Administrator.
2) Navigate to: User Manager for Domains.
3) Select User → New User.
4) Type user name in the Username Field.
5) Ype your first name in the Password Field. Please note that passwords are case sensitive.
6) Type the exact same password in the Confirm Password Field.
7) Add this user to the Administrators Group.

Steps for Logon Security

1) Creating a Logon Warning Message
2) In the run option, type Regedit to open the Registry editor.
3) In the Registry Editor navigate to: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogo.
4) Double-Click the Legal Notice Caption value.
5) In the Value Data entry field, Type Unauthorized Access Warning!! Then press OK
6) Double-click the legal Notice Text value.
7) Type in the Value Data Entry Field.

   "Unauthorized access to this system is punishable by a fine of Rs 7,500 and/or one year in prison. Use of this system indicates that you have read and agree to this warning".

8) Press OK.
9) Close the Registry Editor.
10) Log off and verify the changes.

In addition to logon security management, you can eliminate the name of the previous user logged onto the system. If the last username is not eliminated, then an intruder or hacker can simply look at the screen, or press [Ctrl][Alt][Del] to find out the previous user. By getting a valid username, the intruder has acquired half of what is required to gain access to the system.

3.1.3 Users and Groups Management

In Windows NT every unique user of the system has a unique user account and this account is provided a Security Identifier or SID at the time when account is used. Windows provides multiple levels of user account with the most powerful user account the Administrator (Or Domain Administrator in a domain environment). The Administrator account has the power to manage all the settings on each system and as a result this is the account that must be properly secured. It is suggested that the Administrator account should not be used for day-to-day work at the network. Network administrators should create a separate account for daily routine activities and the Administrator account should be used only when it is absolutely required.

Permissions for resources can be set for individual users but this is not the most efficient way to manage the security of files and folders. It is for this reason that it is necessary to manage the permissions of resources. The function of groups is to assign users who have similar requirements for the use of resources. In this way you will be able to define access to the group rather than to the individual user.

3.1.4 Managing Local and Global Groups

The Administrator can manage the groups in two ways. The two options are Local Groups and Global Groups. Local Groups apply to a single computer and are used to control access to resources on the local computer. Global Groups apply to an entire domain, or group of computers.

You can combine groups together, local and global. But the only allowed combination is to put a Global Group into a Local Group. This is accomplished by adding new computers or members to the Local Group, and from the list selecting a Global Group as the member.

3.1.5 Managing User Accounts

When securing the Windows System, the standards regarding user passwords should be followed. It must be ensured that users are not using weak or easy to guess passwords and there should be no user accounts that have a blank password, and none that have a password that is the same as the username.

Please Note: Windows 95/98 and Windows NT support 14 character passwords and remember this as it may be required for backwards compatibility if you are using Windows 2000.

The Windows System provides the required help to an administrator for managing t passwords. In User Manager for Domains (or User Manager on workstations or standalone servers), the administrator can define Account Policies. These account policies provide various options such as: how long a password is good, how long the password must be, and how many failed attempts will cause the account to lockout, often set to 3. It is necessary to have the password change often for high security and for the system to remember passwords, preventing users from using the same password over and over again. The following steps should be followed for defining the account policies.

Steps for Defining Account Policies for Disabling Last Username Option

1) Disabling the Last Username option
2) In the run box. type Regedit to open the Registry Editor.
3) Navigate to: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ Windows NT\Current Version\Winlogon
4) From the drop-down menus choose Edit> New> String Value.
5) Type DontDisplayLastUsername in the Name Field
6) Double-click on your new string.
7) Type in the Value Data entry field, and press OK to close the Edit dialog box.
8) Close the Registry Editor.

Log off and back on again (no need to restart) to verify the changes. Verify that the last user name no longer appears in the logon box.

1) Log on to your Windows NT Server as Administrator.
2) Navigate to: User Manager for Domains.
3) From the drop-down menus select Policies> Account.
4) Halfway down the page select the Account Lockout radio button.
5) Modify the Account Lockout settings to the following:

    a. Lockout: after 5 bad attempts.
    b. Reset count: after 100 minutes
    c. Lockout
    d. Duration: Forever (until admin unlocks).

6) Close the Account policy dialog box by pressing OK.
7) Log off as Administrator and try these changes.
8) Log back of as Administrator.
9) Navigate to: User Manager for Domains.
10) Double-click on the user you used above.

11) Verify that the Account Locked Out radio button is checked, and uncheck it. Then press OK.
12) Close User Manager.

It is also necessary to secure the Guest account and this account should never be used in a secure environment. The guest account can be locked down by the following steps:

Rename the guest account to a difficult -to -guess account name.
Remove the guest account description.
Set a very complex 14-character password.
Change 1ogon hours to never.
Change the logon to option to a Workstation that is not active.

The concept of user accounts and groups provides an efficient way to manage access to resources, but to define the network itself a larger concept, called the Window NT Domain model, is available.

3.1.6 Windows NT Domain Management

The model of Windows NT Security allows you to control many users, groups, and computers by using a boundary known as a Domain. A server called the Primary Domain Controller (PDC) controls a Domain and there can be only one PDC per domain. But there can be a number of Backup Domain Controllers (BDC) to assist PDC, This Domain model allows for thousands of computers and users under a single management option. When a user logs on to a domain, he is able to access all the computers in the Jogged domain, with the security of those computers dictating the actual level of permission to objects.

This model also provides for a Single Sign On (SSO) to all resources, that is the user is not required to provide credentials for each computer that s/he wishes to access. While the domain model is useful, it does have limitations: (a) a very large domain would be hard to manage efficiently, and (b) users who are very far apart physically may find a more efficient network experience to have one domain per location.

Regardless of the reason, in order for-the network to expand, more than one domain is required. To maintain the SSO across multiple domains a method called trust relationship is used. A trust relationship is an administrative link between two domains. A domain that trusts another domain is called a TRUSTING domain and the other domain is called TRUSTED domain. The TRUSTED domain or Accounts domain holds the user accounts and the TRUSTING domain or RESOURCE domain holds the resources. The trust is only one-way, meaning that if domain A trusts domain B, then domain B does not have to trust domain A. In

order to have trust in both directions, two one-way trusts relationship needs to be created. There are four basic domain models in Windows NT 4.0: (1) the single domain model (no trust created), (2) single master (one Accounts domains (A), one or more Resource domains (R), (3) multiple masters (two or more Accounts domains one or more Resource domain's) shown in *Figure* 2, and (4) complete trust (all domains have direct trusts to all other domain) shown in *Figure* 3.

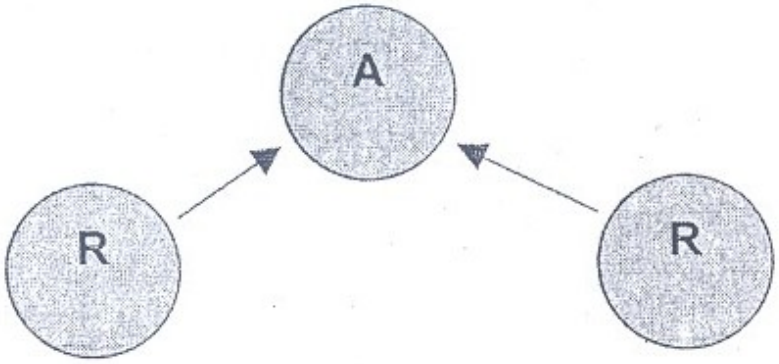


Figure 1: Single Master

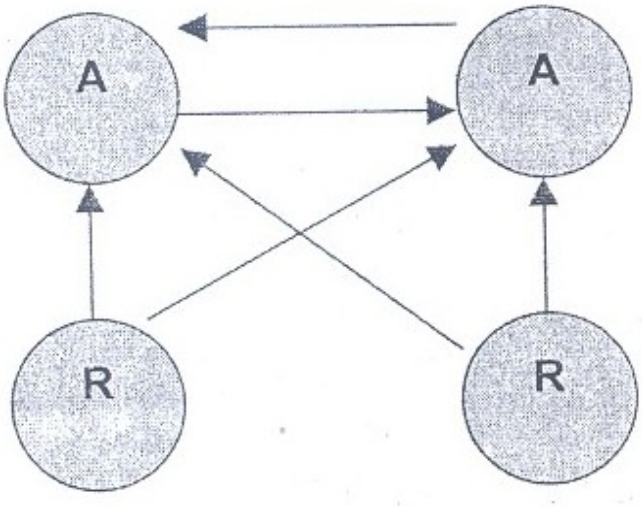


Figure 2: Multiple Master

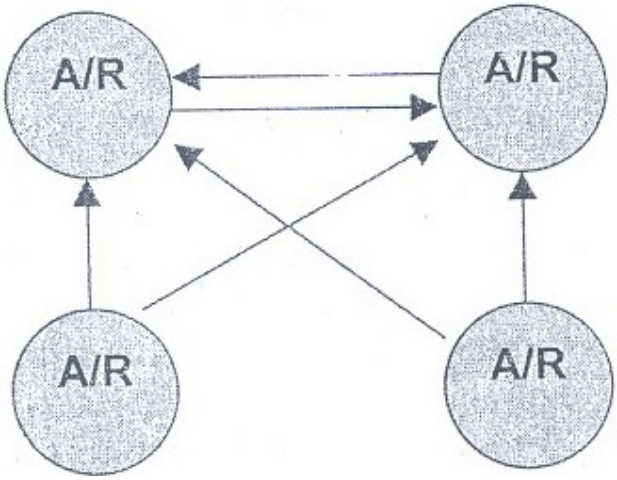


Figure 3: Complete Trust

There is a fifth type of domain structure, but it is not an official model. This type is of a hybrid or mixed layout, shown in *Figure* 4 where the trust structure has no specific pattern.. In this layout there are some Resource domains as well as some Account domains, spread throughout the network.

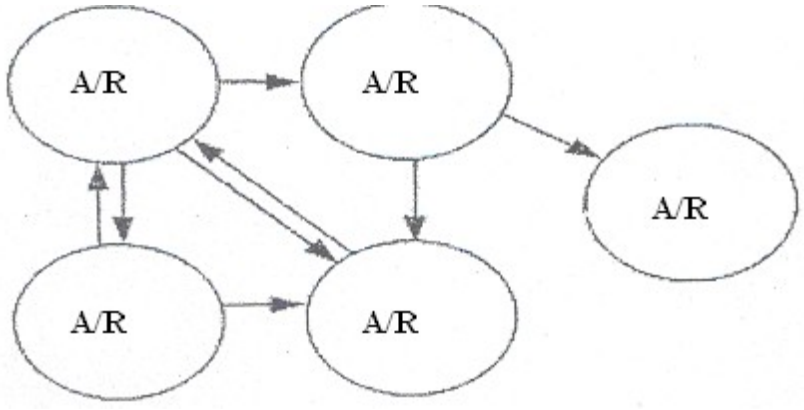


Figure 4: Hybrid or Mixed Layout

SELF ASSESSMENT EXERCISE 1

1) Create a user account "Testuser" and create a logon warning message:

   "Unauthorised access to this system is punishable by a fine of Rs 10,000 and / or one year of imprisonment. Use of this system indicates that you have read and agree to this warning."

2) List the steps for disabling the Last username option.

3.2 Domain Controller

Windows Server organises groups of computers into domains so that all the machines in a particular domain can share a common database and security policy. Domain controllers are systems that run NT Server and share the centra.1ised directory database that contains user account and security information for a particular domain. When users log on to a particular domain account, the domain controllers authenticate the users username and password, against the information stored in the directory database.

When you perform NT Server installation, you must designate the role that servers will play in a domain. Three choices are available for this role: PDC, BPC, and member server (i.e., a standalone server).

3.2.1 The Primary Domain Controller (PDC)

The first Windows NT Server in the domain is configured as a primary domain controller (PDC). The User Manager for .Domains utility is used to maintain user and group information for the domain using the domain security database on the primary controller.

3.2.2 Backup Domain Controllers (BDC)

BDC (Backup Domain Controllers) are the other server after one server has been configured as PDC. BDC stores a copy of the database on the PDC, which is updated periodically to distribute changes made to the main database on the PDC. Such BDC have many advantages:

If the PDC stops functioning due to a hardware failure, one of the BDC can be promoted to the primary role. Such arrangement provides fault tolerance in the network.

PDC provides helps in authenticating network logons. When a user logs on to a domain, the logon request can be handled by any PDC or BDC. This provides an automatic mechanism for load distribution and improves logon performance and it is highly useful in domains with large numbers of users.

3.3 Windows Resources Security Management

Files and Folders Management

The following paragraphs explain the management of Windows resources. In Windows there are two levels of security, share level and file level. Share level security is for controlling user access to a resource

that has been made available to the network, and functions with any file system on the NT machine. File level security is for controlling user access to an individual file locally on a machine, and functions only on the NTFS file system on an NT machine. The share level permissions on a folder (it cannot be set on a file) have four permissions to choose: No Access, Read, Change, and Full Control. This provides power to an administrator to control access to the shared resource from the minimum of No Access, through to the maximum of Full Control.

The share level permission is helpful in many situations, but when you require further control, or wish to secure resources on the local hard drive, you must use file security. The file level security requires that you must use NTFS file system. When permission is set at the file level, the "Everyone group" has Full Control by default.

The share level permission could only be applied to a folder; NTFS permissions may be applied to either a folder or a file. Just as you are allowed to set permissions by each user but you can do so by setting the permission for a group to save time and effort, you will normally set permissions by folder, not file, to save time and effort. Setting the file level security one file at a time can take too long on a file server with thousands of files available over the network.

The NTFS file system provides the following permission for resource:

List-Allows a user to view a field or subdirectory name, but not read the contents of any file or subdirectory.

No Access -Removes all access rights, and will override any other permission a user may have to this object

Read -Allows user List permissions, with the added right of reading tile contents of a file or subdirectory, and run applications.

Add -Allows a user the right to add files and subdirectories.

Read and Add -Allows a user Read permissions, with the added right of adding files and subdirectories to the directory.

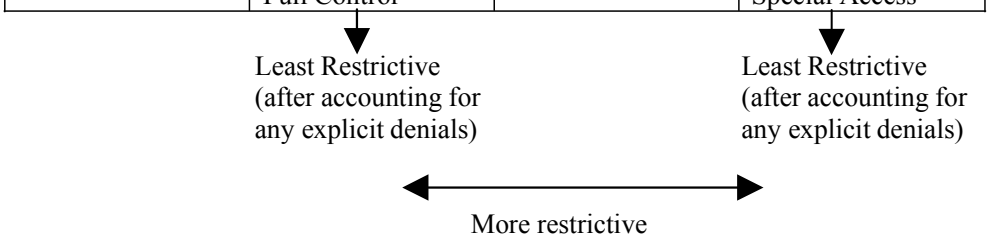
Change -Allows user Read & Add permissions, with the added right of changing data in a file or subdirectory and deleting files and/or subdirectories.

Full Control-Allows user Change permissions, with the added rights of changing permissions on files and subdirectories and taking ownership of files and Subdirectories.

Special Access -This permission allows a user to be given access as in *Table* 1.

Table 1: Assigning Permission for resource

Over the network	Share permissions	NTFS permission local to the machine	
Access Control List	Corresponding Access Control Entry	Access Control List	Corresponding Access Control Entry
User or Group	No Access	User or Group	No Access
"		"	List holidays
	Read	"	Add
		"	Add & Read
		"	Change
"	Change	"	Full Control
"	Full Control	"	Special Access



3.4 Registry Management

In older versions of Windows, the Operating System was controlled by multiple files, such as: autoexec.bat, Config.sys, system.ini, and win.ini. In Windows NT, the configuration of the Operating System is stored in what is known as the Registry.

The Registry consists of values, keys, subtree, and hive.

Values -These contain the information that is stored as part of the Registry. Each value contains three distinct parts: (I) a data type, (2) a name, and (3) a configuration parameter (this contain the actual information).

Keys (and Sub keys) -These contain the actual Subkeys and values.
Subtree -These are tile highest-level Keys of the Registry. There are five Subtrees in Windows.
Hive -These are a set of keys, subkeys, and values of the Registry. Each one is stored in its own file in the %systemroot"/o\ System32\Config.

Regedit.exe and regedit32.exe are the two utilities that can be used to manage the REGISTRY. While Regedit.exe provides the ability to view the entire Registry in a single tree, Regedt32.exe on the other hand, allows for managing of individual keys.

3.4.1 Removing Registry Access

The first step to secure Registry is to try to prevent unauthorised users from accessing the Registry. To do this, the operating system files should be installed on an NTFS partition and change the permissions on both tile Regedit.exe and the Regedt32.exe so that only members of the Administrators group have Full Control.

3.4.2 Managing Individual Keys

In Registry you can secure individual areas of the Registry as necessary. This option is available in Regedt32.exe which permits you to selectively secure the various keys by using the Security Permissions option. Although the details of securing each key are beyond the scope of this unit, the process is identical to that of securing file resources. You must determine the proper level of access for each key, based on your requirement, and limit permissions accordingly.

3.4.3 Audit Registry Access

After locking down the Registry as per your requirement, you need to make sure that the auditing of critical components of Registry is turned on. This option will help in tracking who accessed the Registry, from where, and when. In order to audit the Registry, the first step is to enable auditing for the computer itself.

The steps for enabling auditing are given below:

1. Logon as Administrators.
2. Go to User Manage, for Domains
3. In the Policies menu, select Audit
4. Select Audit These Events to enable these audit choices. Select Failure for the File and Object Access event.
5. Choose OK, and close User Manager for Domains.

There are several options, but for the minimum of registry audits, the Failure for the File and Object Access event is all that is required. Once auditing is turned on for the system itself, you can enable auditing of the Registry. The steps for enabling the auditing registry access is given below:

Steps for enabling the auditing of Registry Access:

1. Log as Administrato,
2. Run Regedt32.exe
3. Select the '\Hkey_Local_Machine' Tree
4. Select the Security, Auditing menu option.
5. Add the specific users and/or groups you wish to audit.

6. Choose OK once you have selected all the users and/or groups you wish to add, and confirm your selection.

Some of the Audit events you may wish to use are listed below:

Write DAC -This audit logs events that try to determine who has access to the key.
Read Control- This audit logs events that try to determine the owner of a key.
Delete -This audit logs events that try to delete a key from the Registry.

If you select auditing on all keys for all users this may result in performance hit 0\1 the system as it tries to track all these events. Therefore, you should only audit the events you specifically wish to audit. You may view the audited events in the Event Viewer under the Security Log. Events that are audited in the Registry will identify the user, computer, and the event that was audited.

3.5 Printer Management

Managing files and folders properly on a Windows machine is just the beginning of setting up the computer's security. Another aspect of computer security is printer management. In Microsoft terminology tile printer is a software component, and the hardware device is called the print device. This section will cover this software component in the computer.

Printer permissions are generally overlooked, but in fact it should be taken seriously. If someone has recently purchased an expensive colour laser print device, it should not be used for general print jobs. Print resources are generally the most misused resources in an organisation.

The following four permissions can be set for printers in Windows environment: access, (2) print, (3) manage documents, and (4) full control.

1. No Access -User cannot print to this device or connect to its print queue.
2. Print-User can print documents and manage submitted print jobs, if the owner of those jobs,
3. Manage Documents -Allows a user to manage print jobs, including pausing, restarting, resuming, and deleting queued documents.
4. Full Control-Allows a user to create, manage, and delete printers, as well as all the control of the Manage Documents permission.

The location of the print spooler should not be overlooked. If print documents are sent to the hard drive for processing, and are waiting to be printed, the security of those locations is a big issue. By default this location is in the % systemroot%/systeni32/ spoool folder and, by default that folder has a permission of Everyone Full Control. So, if you have resources that are secured on an NTFS partition, and they are spooled to a FAT folder with lax security, this may become a security breach. You can modify the security spooler location by using "advanced tab" of print server properties.

3.6 Managing Windows 2000 Operating System

In the sub-section we will focus on how to manage windows operating system.

3.6.1 Windows 2000 Features

In Windows 2000 you can create workgroup for multiple to share resources with one another. The workgroup is referred to as peer-to-peer networking, since every machine is equal.
In Windows 2000, a local security database is a list of authorised user accounts and resource access data located on each local computer.

The major advancement in the design of a Windows 2000 is new domain model instead of multiple models of Windows NT 4.0. In this new model you still group computers together, but they are controlled differently. In a Windows 2000 domain, you group together computers who share a central directory database. This directory database contains user accounts, security information, service information, and more for the entire domain. Active Directory information includes how each object will interact with other objects in the directory. The Active Directory may start out as a small listing and grow to hold thousands to millions of object listings. This directory forms the database for Active Directory and Active Directory is then known as the Window 2000 directory service. In Active Directory no machine is designated as PDC or BDC instead every system is simply called a Domain Controller. In addition to the information mentioned earlier, the Active Directory holds the information regarding access control. When a user logs on to the network, s/he is authenticated by information that has been stored in the Active Directory. When a user attempts to access an object, the information required to authorise such access is also stored in the Active Directory, and is called the Discretionary Access Control List (DACL). Active Directory objects themselves can be organised into what are known as classes. Classes represent a logical grouping of objects at the discretion of the administrator. Object class examples are: user accounts, computers, omams, groups, an organisation Units (OUs). You also have

the ability to create containers, which can hold other .objects. Windows 2000 domain is not bounded by location or network configuration; it may be with in a LAN or far apart over a WAN.

3.7 Active Directory

Active Directory contains several critical components; these components are logical in nature and have no boundaries. These components are domains, forests, trees, and organisational units (OU). The components of Active Directory that are more physical in nature are the domain controllers and sites, the physical IP subnets of the network. The functionality of Active Directory separates the logical from the physical network structure.

3.7.1 Logical Structure

Active Directory has the ability to build a logical network that mirrors the logical structure of the organisation. As logical structure is more intuitive to users they are able to find and identify resources by logical name, without having to have any knowledge of the physical layout of the network.

The main component behind the structure of Active Directory is the Domain.-Active Directory consists of at least, but not limited to, one domain. Microsoft has termed the objects stored inside a domain as interesting objects. These interesting objects are defined as those objects which a user requires in the course of doing their job function. Examples of interesting objects could be printers, databases, email addresses, other users, and more. Each domain holds information about all the objects in the domain, and only those objects that belong to the domain. Domains are allowed to span one or more physical locations.

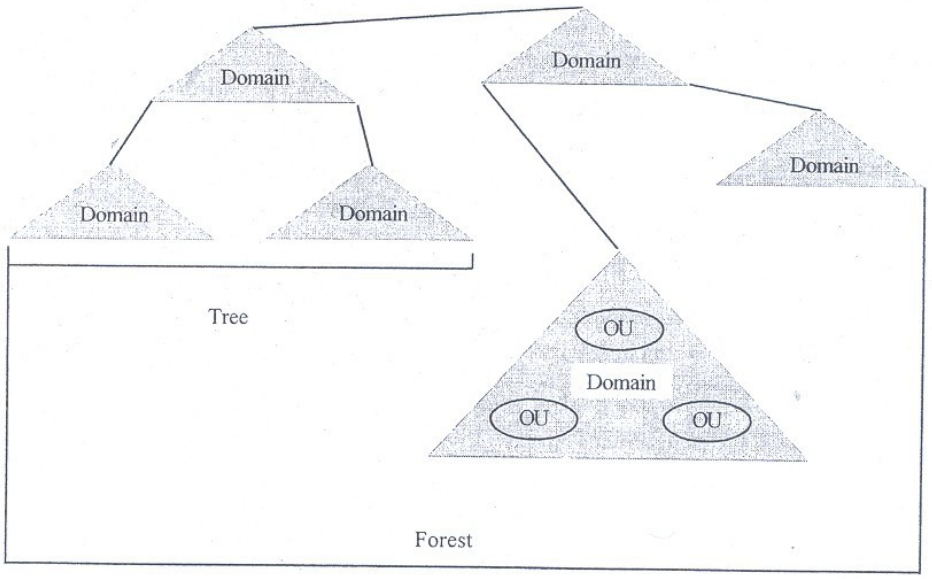


Figure 5: Logical network layout of a Windows 2000 Active Directory

The domain is used as a boundary by which security controls can be implemented. The Access Control List (ACL) is used to regulate specific access to domain objects, such as shared folders, for defined users. The ACL contains the permissions that are used to grant or deny access for an object, such as a user or group to another object, such as a file, folder, or printer. In the domain it can be called Organisational Units or OUs. An OU is a local holder that is used to further mirror the logical structure of the organisation. An au can contain users, groups, files, folders, printers, and even other OUs from the same domain. Every domain in the network can have a unique OU configuration; as there is no dependency on other domains. Permissions can be granted to an OU as desired. It is possible to assign permissions to each au, but not required. If there is a permission that you wish all OUs to use in the network, you may assign it to the parent at! of the domain, as the default structure is to allow child objects to inherit permissions from their parent within the Active Directory.

A new concept in Windows 2000 is that of forests and trees *(Figure* 5). A tree is a logical structure created by the network design team of one or more Windows 2000 domains that share a name space. The domains fall in a hierarchical structure and follow the DNS naming standards. A forest in the Windows 2000 Active Directory structure is a collection of completely independent domain trees. These independent trees are tied together with a trust. Each tree in the forest maintains its DNS name system, and there is no requirement for any similar names pace from one tree to another. Each domain functions on its own, but the logical

connection of the forest enables organisation wide communication on the network. The implementation of Trust in a Windows 2000 Active Directory network is different from Windows NT 4.0. In Windows 2000 all trusts between domains are called two-way transitive trusts. These trusts, based on Kerberos v5 (a security technique) are created automatically when a new domain is added to the tree. The domain that started the tree is considered the root domain, and each subsequent domain's root will form a Two-Way Transitive trust upon joining.

In the event that older Windows machines are on the network, such as a Windows NT 4.0 machine, a specific trust can ~e created. This is called an explicit one-way trust such as and it is non-transitive and in this way a Windows 2000 network, running Active Directory, can have communications with an older Windows NT 4.0 Domain. You also have the option of manually creating trusts such as this, so as to connect two Windows 2000 domains that are far down the trees of different forests to improve communication speed.

3.7.2 Physical Structure

The majority of the design and implementation of the Active Directory network is on the logical side, but the physical side must be equally addressed. The main components of the physical side of Active Directory are sites and the domain controllers.

The site, as defined by Microsoft, "is a combination of one or more Internet Protocol (IP) subnets connected by a highly reliable and fast link to localise as much network traffic as possible." A fast link is reached when the connection speed is at least 512 Kbps. Therefore, the Site is designed to mirror the physical structure of a network, and mayor may not be made up of different IP subnets.

Remember that the domain is designed to mirror the logical needs of the network, and apply that same logic to designing a network using physical aspects. There is no correlation between the site and the domain. It is possible to have multiple domains in a Site, and it is possible to have multiple sites for one domain. A site is also not part of the DNS namespace, which means that when browsing/exploring the directory, you will see user and computer accounts managed by domain and/or OU, but not by site. A site contains only computer objects, and objects relevant to the connection and replication from one site to another.

The other physical component of Active Directory is the actual Domain Controllers (DC) and these machines, which must be running Windows 2000 Server, each have an exact replica of the domain directory. When a

change is made on a DC that has an effect on the Active Directory, all other DCs will receive this replicated change. Because any domain controller can authenticate a user to tile network, each controller is required to have this directory. Therefore, each DC stores a copy of Active Directory information that is relevant to that domain. Each DC replicates changes, at admin-defined intervals, to all the other DCs to ensure a consistent view of the network at all time? Each DC replicates critical changes to all the other DCs immediately and each DC is able to authenticate user logon requests.

3.8 Windows 2000 DNS Management

For the Active Directory to function, DNS must be running for the network. The implementation of the; DNS namespace will form the foundation on which the Active f Directory namespace is created.

A new feature of Windows 2000 is Dynamic DNS (DDNS) which allows clients to receive their IP addresses automatically via a DHCP server and registered with the network. With a DDNS server, the client's machine will automatically communicate with the server, announcing its name and address combination, and will update its DNS information without user information. The advantages of running DNS in a network is the ability to eliminate other protocols and services that may be running to locate resources. For example, the Windows Internet Name Service (WINS) of Windows NT 4.0 is not required, and the use of Net BEUI (Net BIOS Extended User Interface) as a communication protocol is no longer required.

3.9 Managing Group Policy

The final component of the Windows 2000 infrastructure is group policy. A group policy is a logical grouping of user and computer settings that can be inter-connected to computers, domains, OUs, and sites in order to manage a user's desktop environment. For example, a Group Policy is a method of removing objects from the Start Menu.

Group policy consists of GPO (Group Policy Object) and the GPO is then responsible for controlling the application of the policy to Active Directory objects. Once a GPO is configured, it is applied to the AD (Active Directory) object as assigned, and by default the policy will affect all the computers that are in the AD object. The policy can be implemented on all the computers or apply filter how the policy will be implemented for computers and users. The filtering will use Access Control Lists (ACLs), as prepared by you.

Some of the rules for applying a GPO are as follows: a GPO may be associated with more than one domain, a GPO may be associated with more than one OU, A domain may be associated with more than One GPO, and an OU may be associated with more than one GPO. In this section, you have noticed that you are allowed the maximum flexibility in GPO Implementation. However, you are getting into my Implementation, you must take a step back and look into the GPO itself in more detail.

Policies Options

To configure a GPO open Group Policy Editor via the Microsoft Management Console (MMC). In Group Policy Editor you are provided two options; Users Setting, and Computer Setting. In this you will be able to create the GPO as per your requirements.

In the Computer Settings directory you have the option to manage the behaviour of the operating system, account policies, IP security policies, etc. The options will be effective once the computer gets restarted.)

The User Settings directory gives the option to manage behaviour that is unique to the user, such as Desktop settings, Control Panel settings, Start Menu settings, etc. These options will be effective once the user logs on to the computer.

Once you create and edit a GPO, it must be enforced to have any impact on the .network and there can be GPOs on Sites, Domains, and OUs. The order of implementation is critical to proper GPO deployment.

The first GPO that is processed is the called Local GPO. Every Windows 2000 computer has a GPO stored locally. However, it is not practical to implement custom configurations on each machine in the network, so often administrators move right past the Local GPO.
After the processing of the Local GPO, the Site GPO is implemented. Since there can be multiple GPOs for one site, it is the administrator's job to define the order of .implementation by configuring the Site Properties. After processing tile Site GPO, the .Domain GPO is implemented. Just as there can be multiple GPOs for a Site, there can be multiple GPOs for a Domain, so the administrator must take care to define: the order .of implementation in this case also.

The last GPO to be processed is the OU. As in the other implementations, more than one GPO maybe present for the OU, and as such the administrator is required 10 of; properly plan and implement the GPOs as Per the requirements.

In every section with more than one GPO, the place to make the modifications to the order is in the properties of the Site, Domain, or OU (the only exception being the Local GPO When in the properties of the Sit!:, for example, the GPOs are listed, and the option to move them up or down is present, the system will process the GPOs with the highest on the list having the highest priority, taking precedence over GPOs that are lower down on the list.

The implementation order of the GPOs is critical for the security and management of a Windows 2000 network. By seeing at the implementation order, you can identify that if a Site GPO were to define a password age of 45 days, and a Domain GPO were to define a password age of 30 days, that the final password age would be 30 days, as C that GPO was processed last.

4.0 CONCLUSION

After studying this unit, you would have become conversant within the concepts and configure required for network account management of Microsoft windows computers.

In addition, it has taken you through how to examine everything from the foundational principles Windows NT security, as well as discussed the advanced issues of securing windows 2000 machine running Active Directory.

The unit has a broad sweep of the concepts of windows architecture, Network account management and security related issues.

5.0 SUMMARY

This unit describes the broad concepts of Windows Architecture Management and security related issues: Main Issues in Windows Security Management; Windows Resource Management; Windows 2000 Operating Systems. Windows Security specially focuses on Windows NT Management and it covers the areas such as physical security management, logon management, user/groups management, Windows NT domain management, domain controllers. Windows resource management includes t areas like: files and folder management, files/folder permissions, printer management, and Registry management. Further, the unit also discusses about the improvement that has been taken up in Windows Architecture, Management with the Management Windows 2000 operating system; Windows 2000 features, active directory, logical structure, physical structure, Windows 2000 DNS management, Group Policy etc. This unit provides detailed concepts and configuration required for management of Microsoft Windows

computers and you will be able to examine everything from the foundational principles of Windows NT Management, up to the advanced issues of securing Windows 2000 machines running Active Directory.

6.0 TUTOR-MARKED ASSIGNMENT

1) Fill in the blanks:

   a. The model of Windows NT allows you to control many users, groups, and computers by using a boundary known as a ----------
   b. There can be only ---------- PDC per domain. But there can be a number of to assist PDC.
   c. Domain model provides for a ---------- to all resources.
   d. A domain that trusts another domain is called a ----------domain and the other domain is called ---------- domain.
   e. If PDC stops functioning due to hardware failure, one of the BDC can be promoted to ----------
   f. PDC provides help in ---------- network logons.

2) What are limitations of the domain model?
3) What do you understand by PDC and BDC?
4) What is Active Directory?
5) How will you secure guest account?
6) What do you understand by Windows 2000 DNS?

7.0 REFERENCES/FURTHER READINGS

Cryptography and Network Security, Principles and Practice, William Stallings -SE, PE.

Security in *Computer;* Charles P. Ptleeger and Shari Lawrence Ptleeger, Third Edition, Pearson Education.

Windows 2000 Commands by Aleen Frisch.

Microsoft Web Site ht!ll ://www.rnicrosoft.com.

UNIT 4 SECURITY AND MANAGEMENT-II

CONTENTS

1.0 INTRODUCTION

This unit will introduce you to the concepts and configuration required for the management Microsoft Windows computers and you will be able to examine everything from the foundational principles of Windows 2000 security, up to the advanced issues of securing windows 2000 machines running Active Directory.

This unit covers in detail the various security methods that can be implemented in windows 2000 architecture. The unit addresses management of Windows 2000 system: Authentication (section 3); users and group security (section 4); resource security (section 5); windows network security (section 6); and encrypting file system (section 7). The section 3 of this unit deals with windows 2000 user authentication management and it covers the following areas; Subsystems components, and Kerberos.

The section 4 of this unit deals with the users and group security management and it covers the topics like; configuring user's accounts, windows 2000 groups (default group type, local groups, global groups, group policies etc.), security configuration tools, and configuration management and analysis tools.

The section 5 deals with resource security management and it covers the areas like; files and folder management, files/folder permissions, inheritance and propagation, moving data and permissions, shared resources, null session, printer management, and Registry management.

Section 6 the most important deals with tl1e network management; NAT, ICS, RRAS, RAS, IAS, and IPSec are covered in this section.

The section 7 deals with encrypting file system (EFS), data recovery and EFS cryptography.

2.0 OBJECTIVES

After going through this unit you should be able to:

learn windows 2000 authentication
user and group management
resource management
EFS Management
windows network management.

Objectives of this unit are: Examine the basics of user authentication in Windows 2000, learn to manage User and Group Options in Windows

2000, manage and configure security options on Windows 2000 Resources, Examine the methods or management network communications in Windows 2000 Examine and configure EFS on Windows 2000.

3.0 MAIN CONTENT

3.1 User Authentication Management

Despite all the advancements and new components of Windows 2000, a user must be authenticated to access resources on the network. Windows 2000 can use the following for authentication: Kerberos, NTLM, RADIUS, SSL, Smart Cards, and more.

Windows 2000 uses the Security Support Interface (SPPI) to allow for these metl10ds of authentication. The SPPI functions as a interface between the user applications, such as the Web Browser, and the authentication method, such as NTLM or Kerberos. An application developer need not create an application for each type of authentication possible, but create one that can communicate with SPPI.

Although the SPPI plays an important function in the authentication of users with no options for configuration or management involved in the SPPI, it simply performs its job of connecting authentication requests to the authentication provided by the system.

The administrator is involved more with Security Architecture of Windows 2000, which comprises of parts of both the Operating System and Active Directory. For example, in the Active Directory are the stored account information and policy settings, while in the Operating System is the security process that is and information regarding trusts to and from other areas of the network

If the Windows 2000 is installed in mixed mode, means that there can be both Windows NT 4.0 BDCs and Windows 2000 domain controllers present. This allows for maximum communication options over the network, but it is not the most secure environment. The reason behind this is an issue is that an older networking server, called LAN Manager, used the LAN Manager (LM) protocol for authentication and this protocol has weak security. Windows 9x and NT accepted LM authentication, and this is where the weakness lies and a password can be broken into 7-character pieces and cracked individually. Therefore, even though a l4-character password was implemented, the program that is trying to crack the password is cracking two 7- character blocks at once. The implementation of LM in Windows NT requires the system to not only accept LM authentication, but to store a copy of the LM version

of the password in the Registry. Attackers will go after the LM password since they will almost match the NT password.

Microsoft addressed the issue with the default NTLM was to develop and release NTLMv2. There were several increases in the security provided by implementing NTLMv2; the key, or password, was now a 128-bitvalue, which will take much longer to crack, and MD5 (Message Digest 5) was used to verify the integrity of messages. In order for Windows NT 4.0 machines to implement NTLMv2 they must use Service Pack 4 or greater. If all your clients support NTLMv2, you may configure your Windows 2000 clients to do so also. This may be defined by creating a GPO for an OU that holds all the machines that must use NTLMv2. Then configure the response type, as per your network, in the Security Options, under Computer Configuration, in the Group Policy Editor.

3.1.1 Subsystem Components Management

The logon information is stored in the local Registry on a stand-alone machine or a machine that is part of the workgroup. The Windows 2000 logon process is the same as the Windows NT 4.0 logon process for a stand-alone machine and the Registry stores the user account data in the Security Accounts Manager (SAM). The SAM is used in NT 4.0 to store all user account information, and in Windows 2000 is what is used to store local user account information. But, if a Windows 2000 Server is promoted to be a Domain Controller, the local SAM is no longer accessible. The process for when a user tries to access a local resource is as follows: (I) the user account info is given to the Local Security Authority (LSA). The LSA is what creates the access tokens, provides an interactive environment for user authentication, controls the local security policy, and sends authentication requests to NTLM or Kerberos, as required, (2) the LSA gives the authentication request to NT LAN Manager (NTLM), and (3) the user request for the resource is validated by the Security Reference Monitor{SRM). The SRM performs the actual checks on user permissions to access objects.

3.1.2 Kerberos Management

In Windows 2000 no action is required to implement Kerberos. Kerberos will be used by default to authenticate network clients (with Windows 2000) logging onto a Windows 2000 domain.

Kerberos is an IETF standard used for authentication and the Massachusetts. Institute of Technology (MIT) developed it during the 1980s. It is considered to be a secure method, and has been implemented in Operating Systems before the Windows 2000 implementation. There

is a bit of controversy in the method used in Windows systems as it varies slightly from the standard created by MIT, However, it should be noted that Windows 2000 is able to intemperate with non-Windows 2000 machines running Kerberos.

When a user the log on process by entering his credentials, Windows will contact an Active Directory domain controller, and locate the Kerberos Key Distribution Center (KDC). An Authentication Server (AS) performs the actual authentication. The KDC responds by issuing a Ticket Granting Ticket (TGT) to the authenticated user. The TGT contains identification information about this user to various servers on the network, and is used to gain further access in the network.

After the user account has been authenticated, the TGT is used to request further Kerberos tickets in order to access network services. The machine that provides the tickets for the network resources to the authenticated client is known as a Ticket Granting Server (TGS).

The benefits to end-users of a network running Kerberos are that a Single Sign On (SSO) will be maintained and the users are not required to authenticate with each resource they wish to access in the network, and since Trusts in Windows 2000 are transitive, once a user logs on to one domain user, s/he will have access to the other domains of the network. Another key benefit of Kerberos is that it has a mechanism for verifying the identity of the user, not just authentication. This means that in a Kerberos network, if a message says it came from User X, you can be very confident it did indeed come from User X.

3.2 Users and Group Management

In earlier sections we examined the infrastructure of Windows 2000, including the concepts relating to the Group Policy. The following sections build off those foundational issues, introducing users and groups into the network.

3.2.1 Configuring User Accounts

The focal point of Windows system is the users and without users being able to access the network, there is no point in having a network. There are two basic types of user accounts that maybe created in Windows 2000, domain and local. A domain user account has the ability to log on to the network and access authorized resources throughout the domain. A local user account has the ability to log on to a specific computer and access authorized resources on that computer.

The default accounts in Windows 2000 server are tile Guest and Administrator. Securing the Guest account and Administrator should happen right away. These steps are as follows: (I) remove the description, (2) disable all logon hours, (3) create a very complex password, (4) and allow the account to only access the network from a nonexistent machine.

3.2.2 Creating Domain User Accounts

The steps for creating domain user accounts are:

a. Open the management console MMC.
b. Open or add the Active Directory Users and Computer Snap-In.
c. Expand domain listing, to view the console tree.
d. In the Action down menu, select option New User.
e. Create new users, userl, user2, user3, user4, etc.

3.2.3 Managing Logon Hours

Once you have created several users, the next step is to restrict logon hours. That means restricting the hours in which a user can logon to the server. The steps are:

a. Open the Active Directory Users and MMC Snap-in
b. Expand domain listing, to view console tree.
c. Select user folder.
d. Double click user1.
e. In the Property Window, choose Account Tab, and select the Logon Hours Option.
f. Limit user I so that this account can log on to the network during l0 AM to 5 PM during week days (i.e. Monday to Friday).
g. Press OK to close the Logon Hours dialog box.
h. Again press OK to close the Userl Property Window.

3.2.4 Managing Expiry Date for a User Account

You can further control the access to network resources by setting a limit or expiry date for a user account.

a. Open Active Directory Users and Computers MMC Snap-In.
b. Expand domain listing to view the console tree. c. Select user folder.
d. Double click user3 and in property window select the Account tab.
e. In the Account Expires option and select End of option and enter a expiry date.
f. Press OK.

3.2.5 Windows 2000 Groups Management

While working with Windows 2000 you will most likely want to implement and configure a full Active Directory structure, to gain all the benefits afforded by doing so. However, when you first install a windows 2000 server, it is nothing more than a stand alone server, not even part of a domain, Jet alone a domain controller.

Once the machine has become a domain controller (by running DCPROMO), as the administrator there are several groups-for you to manage. These groups include the Domain Administrators and Domain Users.

There are two group types, a Security Group and a Distribution group. The Distribution Group is used to manage lists, such as email lists.

3.2.6 Default Group Types

On Windows NT 4.0 that groups can be either Global or Local, in Windows 2000 this concepts is expanded. In Windows 2000 the group types are: (1) Domain Local, (2) Computer Local, (3) Global, and (4) Universal.

Domain Local Group is one that may have members from any domain in the network. These groups are only created on Domain controllers, and can be used to provide resource access throughout the domain. The Computer Local group is used provides access to resources on the local machine only, and cannot be created on a Domain Controller.

Global Group is one that combines users who often share network resources use and access needs. Global groups may contain members from the domain in which the group was created.

Universal Groups are used in a multi-domain environment where groups of users from different domains have similar resource use and access needs. To implement Universal groups, the network must be running in Native mode, meaning only Windows 2000 computers.

It is also possible to combine groups together, such as Global Groups in Universal Groups. There may be a resource you are trying to control; in this case a Universal group will work for controlling access across the network. You may also place Universal Groups in Domain Local Groups, and control access 10 the resource by placing permissions on the Domain Local Group.

These groups can be used for controlling access to resources; both allowing and denying permissions based on your security needs. If you are trying to secure the computer, user, and network environments, you will use Group Policies, as discussed in the previous sections.

Group Policies Management

Two of the issues that must be discussed are the options associated with Policy Inheritance and Overrides. The Group Policy Objects are implemented in the following order: Local GPO, Site GPO, Domain GPO, and OU GPO. And when there is multiple GPOs assigned 10 an object such as a Domain that the highest GPO on the list takes priority over the rest of the list. You can change the order of implementation on this list by simply choosing a GPO and pressing the Up or down button to re-order the list as you desire. However, you may need to have further control than what the Up and Down option provides you.

Policy Inheritance

Policy Inheritance is the name of the process of a user or computer inheriting the final policy configuration from multiple policies, depending on where the object may be in the Active Directory hierarchy and configured GPOs. To track the policies that may be implemented as a user logs onto a computer, use the following list: (1) a Computer Policy is enabled when the computer is first turned on, (2) a User Policy is applied, (3). When a user logs onto the system, (4) the Local GPO is applied, (4) the site GPO is ~ applied, (5) the Domain GPO is applied, and (6) the OU GPO is applied.

It is not uncommon for Sites, Domains, and OUs to have more than one GPO configured. It is also not uncommon then for there to be conflicting settings in locations throughout the policies.

No Override

One of the methods for you to manage a GPO implementation is through the No Override option and this option is available on any Site, Domain, or OU GPO. When this option selected, this option means that none of the policy settings in this GPO can be overridden. In the event that more than one GPO is set to No Override, the highest GPO takes priority

Block Inheritance

The other choice for managing policy implementation is cal1ed Block Policy inheritance and this choice is also available to any Site, Domain, or OU GPO. This option means that any policy that is higher will not be

inherited. Enabling this option wil1 ensure that the settings of the current GPO will be implemented and not the policies of a higher priority policy.

Block Inheritance and No Override options must be used with proper care and if used with incomplete planning can cause serious disruptions to the overal1 policies that are implemented throughout the organization.

4.3.7 Security Configuration Management Tools

In Windows 2000, there are with a variety of tools and resources for the configuration and management of security options on both individual computers, and the network itself. These tools include The Security Template Snap-In, The Security Configuration and Analysis Snap-In, and Secedit.exe. Secedit.exe is a command line tool that can be used for analyzing the security of computers in a domain.

Security Templates

The task of configuring al1 the options in the GPO can be quite complex at times. To help with defining how the security should be configured for given situations, Microsoft has included Security Templates that can be used in the Group Policy Editor. These templates are in files and can be opened with a text-editor, for viewing.

Templates are stored in the % system root%\security\templates. These templates can be applied to a GPO, and any user or computer that is controlled by that GPO will implement the security template. A template itself is a set of pre-configured options and Microsoft has included a ful1 set of templates designed to cover most of the standard scenarios that are possible. User can use the default templates as-is, or modify them to suit his requirements. In addition to modifying a template, a user can create his oven template from scratch.

Predefined Security Templates

The list of common Security Templates are given below:

BASICDC.LNF -used to configures default Domain Controller security settings.
BASICSY.LNF -used to configures default Server security settings.
BASICWK.LN -used to configure default Workstation security settings.
COMPATWS.INF -used to configures compatible Workstation or Server
security settings.

SECUREDC.INF- This template configures secure Domain Controller security settings.
SECUREWS.LNF -This template configures secure Workstation security settings.
HISEDC.LNF -This template configures highly secure Domain Controller security settings.
HISECWS.INF -This template configures highly secure Workstation security settings.
SETUP SECUR1TY.INF -This template configures out of the box default security settings.

There are several general security levels in the templates: Basic, Compatible, Secure, and Highly Secure. The following sections define the general purpose and function of each of the security levels.

Basic Templates (BASIC*.INF): These templates allow for an administrator to reverse an earlier implementation of a security configuration and configure Windows 2000 security settings that are not related to user rights.

Compatible Templates (COMPAT*.INF) are often only run in a mixed environment. This template configures the system so that local Power Users have security settings that are compatible with Windows NT 4.0 users.

Secure Templates (SECURE*.INF) configure security settings for the entire system, but not on files, folders, and Registry keys.

Highly Secure template (HISEC*.INF) is used to secure network communications on Windows 2000 computers and it allows for the highest level of protection on traffic sent to and from Windows 2000 machines. This template requires that a computer configured to use a HISEC template can only communicate with another Windows 2000 computer

Dedicated Domain Controller (DEDICADC.INF) is used to secure a machine running as a Domain Controller. The reason you may wish to implement this template is that by default the security on a DC is designed to allow for legacy applications, and as such is not as secure as it could be. If your DC is not required to run any of these programs, it is suggested that the Dedicated DC template be implemented.

The final predefined template we will discuss is one that is very important in today's world, but is not included with the other preconfigured templates –the :HISECWEB.INF template.

Microsoftat:http;//microsoft.com/default.aspx?scid=kb;en-us;Q316347& this template is discussed in the Microsoft article: “IIS 5: HiSecWeb Potential Risks and the IIS lookdown Tool (Q316347)".The implementation of the HISECWEB.INF template is a requirement for any US 5.0 Web Server that wishes to be locked down.

HISSECWEB.INF is designed to configure an US 5.0 machine running the WWW service. Although not in the list of default templates, this can be found and downloaded for free, directly from.

Analysing Password Security Policy of Templates

Open MMC management console, and select Add/Remove Snap-In option. Press Add and add the Security Templates Snap-In. Expand and review the password policy of following templates: Hisecdc, Basicsv, etc.

It is evident from above that the security templates provide a range of configuration. And if default or available templates does not quit fit to your needs, you can simply, create a new template altogether.

Creating a Custom Template

a. Open MMC and select Add/Remove Snap-Ins.
b. Click on Add button, and add Security Templates Snap-In.
c. View all templates by expanding Security Templates.
d. Right Click Directory Location (e.g. :\Winnt\Security\Ternplates) and press New Template.
e. Enter template name: Custom Template.
f. Enter Description: Template for highly secure passwords.
g. Press OK
h. Apply following configuration settings to Custom Template.

Password History 30 passwords
Maximum Password age of 15 days and Minimum password age of3 days.
Minimum password length l2 characters.
Account LOCkout duration 0 minutes
Account Lockout Threshold of4 invalid Logon attempts.
Reset Account Lockout Counter after 70 minutes.
Right Click and press Save.

Advance Security Management- Through Security Configuration and Analysis Snap-In Tool

After creating the policy and making changes in the predefined templates, the templates are applied to the network. As mentioned earlier, templates can be applied (also called Imported) to GPOs and importing a template to a GPO is a straightforward procedure, and uses a tool called Security Configuration and Analysis Snap-In.

The Security Configuration and Analysis Snap-In is another of the advances in security management provided by Windows 2000. Through this tool, you are able to implement templates and configure the security of your system. In addition to implementation, this tool allows for a complete security analysis oftheoperati/1g system.

This tool is compares the security settings of a template to the current configuration of the operating system. During this analysis, this tool will highlight items that are in compliance with the settings with a green checkmark, and highlight those items that are not in compliance with a red X. Implementing the security configuration with an analysis tool is a time consuming process.

Steps are:

a. Open MMC and select Add/Remove Snap-Ins.
b. Pres$ Add button, and add Security Configuration and Analysis Snap-Ins.
c. Right Click Security Configuration and Analysis Snap-Ins and select open database.
d. Open Password_Check.sdb.
e. Select your earlier created Custom Template, and press open.
f. Right Click Security Configuration and Analysis Snap-In and choose Analyze Computer Now and press OK.
g. Right Click Security Configuration and Analysis Snap_ins and examine whether or not your system is up to policies in respect of passwords.

Implementing a Template

a. Open MMC, Right Click Security Configuration and Analysis Snap_ins, and click on Configure Computer Now.
b. Press OK. This process will take several minutes and no message will be displayed.
c. Run the analysis again to confirm the configuration.

3.3 Resource Management

In this section we are highlighting resource management related issues.

3.3.1 Files and Folder Management

Windows NT 4.0 had the ability to work with only FAT and NTFS file systems, Windows 2000 can also work with FAT32. Further, NTFS should be used for Windows Security Options. NTFS in Windows 2000, technically called NTFS version 5, is required as an administrator wishes to use Active Directory, Domains, and the advanced file security that is provided. Further, the addition of file encryption and disk quotas require NTFS. It is suggested that all partitions that are still running FAT or FAT32 be converted to NTFS in order to effectively secure Windows 2000 resources. If you need to convert a partition to NTFS, the command is (using the C:\ drive as the example): convert volume IFS: NTFS /C. Any new partitions either created or converted to NTFS will, by default, allow everyone group Full Control access. As this includes the Guest and Anonymous accounts, strict security must be implemented before user accounts, which are able to access the system, are added.

In Windows 2000 some additional steps have been added to prevent users from making changes to the system files of Windows itself. Those changes are to hide the folders in the Winnt folder and the System32 folder by default. However, a quick click on the Show File option and all is revealed. There is a built-in mechanism that is working to keep system files from being modified, called the Windows File Protection (WFP) system, and its job is to ensure that system files installed during the setup of Windows are not deleted or overwritten. Only files that have been digitally signed by Microsoft will be able to make these changes.

3.3.2 Files and Folder Permissions

To view permissions, Right-click the object, Select properties, and view the information on the Security tab. One can view more detailed data in advanced option. File permissions are different in Windows 2000 over NT 4.0. Some of the File Permissions available are defined in the following list:

> **Traverse Folder/Execute File:** The Traverse Folder (Applied to folders only) permission manages a user's ability to move "through" a folder to reach other files and f9lders, regardless of the permissions on the folder. The Execute File (Applied to files only) permission manages a user's ability to run program files.

List Folder/Read Data: The List Folder (Applied to folders only) pennission manages a user's ability to view file names and folder names. The Read Data permission manages a user's ability to read files. (Applied to files only).

Create Folders/Append Data: The Create Folders (Applied to folders only) permission manages a user's ability to create folders within a folder. The Append Data (Applied to files only) permission manages a user's ability to make changes to the end of a file.

Create Files/Write Data: The Create Files (Applied to folders only) permission manages a user's ability to create files within a folder. The Write Data (Applied to files only) permission manages a user's ability to modify and/or overwrite a file.

Delete: This permission manages a user's ability to delete a file or a folder.

Read Permissions: This permission manages a user's ability to read the permissions of a file or a folder.

Change Permissions: This permission manages a user's ability to change the permissions of a file or a folder.

Take Ownership: This permission manages a user's ability to take ownership of a file or folder.

Read Attributes: This permission manages a user's ability to read the attributes of a file or folder.

Write Attributes: This permission manages a user's ability to modify the attributes of a file or folder.

	permissionRead (Display Data, attributes, owner,	Executive (run or execute the file or files in the folder)	Write (to the folder or to the file or chance the file attribute)	Delete (the directory or file)	Change Permission (i.e., the permission to change permissions)	Take Ownership	
FOLDER SECURITY	R	X	W	D	P	O	FILE SECURITY
No Access							No Access
List Folder.	*	*					
Read.	* *	* *					Read
Add		*	*				
Add &Read	*	*	*				
Change	*	*	*	*	*	*	Change
Full Contro	*	*	*	*	*	*	Full Control
Special Access	? ?	? ?	? ?	? ?	? ?	? ?	Special Access

Figure 1: Files and Folder Permissions

These permissions alone are not considered to allow or deny access; the administrator must define on each object. It is not required to specify each of these unique permissions when securing resources. User will most likely use the defined permissions of: Full Control, Modify, Read and Execute, List Folder Contents, Read, and Write. The specific abilities of each of these Permissions are defined in the chart shown in *Figure* 1.

When you apply the Read permission, for example, to a folder, the folder gets the List f Folder / Read Data, Read Attributes, and Read Extended Attributes. NTFS file permissions are similar, with the difference of no List Folder Contents as an option, as the permissions are applying to a file.

3.3.3 Inheritances and Propagation

When a user creates a new file, this new file will inherit the permissions of its parent folder, or parent partition on a root level folder. Therefore, if a parent folder is set: Everyone Modify, the file you create in that folder will have everyone modify as its permissions. User can alter this

behaviour, create a folder apply the permissions to the ;; This Folder Only option, which means that new data created in the folder will not inherit the permissions of the folder and new objects will inherit the permission that is set one level higher. Therefore, if you have a folder D:\Secure\Self, and this folder has had permissions applied to it only, when you create a file D:\Secure\Self\test.txt, this file will inherit its permissions from the D:\Secure object.

User can also block the inheritance of permissions by clearing the Allow Inheritable Permissions from Parent to Propagate to this Object option on the Security tab of the f Properties windows for an object. When you clear this option, you will be presented with three options: (I) Copy the permissions that this object has inherited, (2) Remove all permissions except for those that have been specifically applied, and (3) Cancel the operation and keep the permissions as they were.

The process of configuring/setting permissions in Windows 2000 is similar to that of Windows NT 4.0, with the exception that you will specifically allow or deny access. If you wish to give a user or a group what was called No Access in Windows NT, you would, in Windows 2000, give that user or group Deny to the Full Control permission.

The attacker can get around your NTFS security, if they are able to get physical access to the computer by using MS-DOS etc. User may think that using DOS will not have an effect on any files that are on an NTFS partition, and that DOS will not even be able to recognise the NTFS partition. In most situations this is true; however there are tools and utilities on the market that are designed to access NTFS from DOS. One of the most common of these tools is simply called NTFSDOS.

Steps for assigning permissions

a. Open Windows Explorer, and select any NTFS partition.
b. Create a new folder, called protected folder and right click this folder and select properties.
c. Select Security tab and clear the Allow inheritable permissions from parent to propagate to this folder, and choose copy option.
d. Add the user3 and give this account Deny- Full Control permission.
e. Add the user2 and give this account Allow- Modify permission.
f. Add the user4 and give this account Allow- Read and Execute permission.
g. Press Advance button and select User I and press View/Edit.
h. Modify security settings to Apply onto: This folder only (i.e., protected folder).

3.3.4 Moving Data and Permission

When data files are moved from one folder to another, what will happen to the security permissions that were set to secure these files. When files that are secured on an NTFS partition, how their security settings may be altered if those files are moved. In other words, if a file is defined as having everyone -allow -Read & Execute permission what will happen to those permissions if the file is moved to another folder? The rules in Windows 2000 regarding copying and moving files are the same as they were in Windows NT 4.0 and by default, a file will keep the permissions that are assigned to it when moving the file to another folder on the same NTFS partition. If the file is moved to another NTFS partition the file will inherit the permissions of the destination folder or partition. If a file is copied to any location, it will inherit the permissions of the destination folder or partition.

3.3.5 Shared Resources Management

Windows 2000 is designed to provide extensive network services; the security of resources via the network must be a high priority. The normal users of the system are not granted the permission to create shares on their local machines. Only Administrators and Power Users have this right to do so.

Three permissions are available for a shared folder, which may be applied to a user or a group; (I) Full Control, (2) Change, and (3) Read. These permissions are independent of the permissions set using NTFS security options. Windows 2000 uses both NTFS permissions and Share level permissions to decide the access a user will have to an object. When there are conflicting levels of permission for an object. Windows will determine the least restrictive permission both for the NTFS security and the share security. It will then compare those two permissions and the more restrictive of the two will be the resultant permission for the User. The exception to this rule is if a user has been given the Deny -Full Control permission, this takes precedence over the other permissions.

3.3.6 The NULL Session

For a system to provide shared resources it must communicate with the network and this communication is done via anonymous connections from system to system. If the system is not connected to Internet, this may not present a problem, but if the machine is directly connected 10 the Internet,-this operation may allow an attacker to learn about the inside network without authorisation.

This is called a NULL session connection, and is when an attacker connects as the anonymous logon. User should disable the NULL session and this can be done via any of the Security Templates. The steps for this are as follows: (I) Open anyone of the security templates in the MMC, (2) Navigate to Local Policies, (3) Navigate to security Options, and (4) Set the Additional Restrictions for Anonymous Connections to No Access Without Explicit Anonymous Permissions'.

3.3.7 Registry Management

The Windows 2000 Registry stores the configuration data for the computer, and as such is obviously a critical item to secure properly. The Registry in Windows 2000 can be directly updated with the tools like Regedit.exe and Regedt32.exe. As mentioned earlier it is recommended that Regedt32.exe be used as permissions can be applied to individual keys as you see fit. When setting the primary permissions in the Registry, however, you only have Read and Full Control to choose from.

The following lists are the permissions that are available for Registry: Query Value -Ask for and receive the value of a Key

Set Value -Change a Key Value
Create Subkey -Create a Subkey
Enumerate Subkey -List the Subkey
Notify -Set Auditing
Create Link -Link this Key to some other Key
Write DAC -Change Permissions
Read Control -Find the Owner of a Key
Write Owner -Change Ownership of a Key
Delete –Delete the Key.

The permission "full Control" is equivalent to all permissions listed above and the "Read" permission is equivalent to the Query Value.

3.3.8 Default Registry Configurations

There are systems in place to protect the Registry by default. Administrator SYSTEM account should have Full Control to all areas of the Registry. Power users are given permission to create subkeys in the HKEY _LOCAL- MACHINE\SOFTWARE\ key, which allows them to install new software packages. Power users then have Full Control over the subkeys they create, as does the CREATOR OWNER Account. The extent of control for power users does not expand into all areas of the Registry. For example, in the Hardware hive of the Registry power users

are not on the list to set permissions, by default. While making changes to areas of the Registry, be sure to have planned out the changes very carefully, as unintended actions can happen very easily and quickly.

Steps for Configuring Registry Permissions are given below:

a. Logon as Administrator.
b. Open Regedt32
c. Select HKEY _LOCAL_MACHINE
d. Expand SAM and leave the Greyed out SAM selected, choose Security from drop-down option and select permissions.
e. In this give Administration Full Control permission.
f. Expand SAM and notice that user and account information is now visible.

3.3.9 Registry Backup Management

To Secure the Registry, a backup strategy for the organization should be implemented.

There are several met\1ods in which to backup the Registry; the first of these is to go through the Registry itself to save Subkeys/files, use the Microsoft Backup program. The Microsoft Backup utility can create a full backup of the System State, which includes the Registry configuration information. The storage option for backups is critical and a compromised system state backup is dangerous. The main files to secure, in regards to Registry Backup, is in the Operating System files, and stored in the % system root"/o\repair folder are the settings that must be secured. This folder contains the Registry configuration information that is needed in the event the system needs to be repaired.

Steps for Saving the Registry Information are given below:

a. Open Regedt32 and select software subkey of HKEY _LOCAL_MACHINE.
b. From drop-down option select Save Key.
c. Create a folder Reg_keys_folder in NTFS partition and create soft_I as the file name and press save and close the Registry Editor.
d. Again go to Reg_keys_folder, right click and select security tab. Configure the security such that only user3 has Full Control, and remove any access to any other user account or group.

3.3.10 Printer Security Management

Printer Security in Windows 2000 provides three permissions: Print, Manage Printers, and Manage Documents. The Print option is the default level of security provided to users. This means they are provided the right to print, pause, resume, restart, and cancel documents they have submitted to a printer. To provide more control to a user, you can give them the permission of Manage Documents. With this level of permission, they are able to get the right to pause, resume, restart, and cancel all documents that have been submitted to this printer. You can also give Manage Printer permission and this level of permissions means they are given the right to share the printer, change printer permissions, change printer properties, and delete printers.

More control still over the printer can be acquired through advanced setting of printers. In the advanced settings of a printer, you can define the hours in which the printer is available. If the printer is to be used during only business hours, there is no reason to have the hours of the printer state it may be used 24x7. This type of control helps to keep the device used for official purposes only. You should secure the spooler that holds print jobs waiting to print and if the spooler is left at the default, it is in the % system root %, allows Everyone Full Control. This location should be moved to a secure NTKS location and should be managed individually.

3.4 Windows 2000 Network- Security and Management

3.4.1 NAT and ICS

In the previous section all of the security systems and methods are for securing operating system and data OJ) physical hard disk. This security system is of no use if an attacker is able to sniff network packets.

Network Address Translation (NAT), is used to mask internal IP addresses with the IP address of the external Internet connection. Networks require NAT in their security, policies to add an additional security "layer" between the Internet and the intranet. NAT functions by taking a request from an internal client and making that request to the Internet on behalf of the internal client. In this configuration clients on the internal network, on local LAN, are not required to have a public IP address, thus conserving public IP addresses. The internal clients can be provided with an IP address from the private network blocks. Private IP addresses are not routed on the Internet and the address ranges are:

Private IP Addresses
10.0.0.0-10.255.255.255
172.16.0.0-172.31.255.255
192.168.0.0-192.168.255.255

However, Microsoft has designated a range for private addressing

169.254.0.0-
169.254.255.255.
NAT is an integral part of Routing and Remote Access Services (RRAS), as well as part of Internet Connection Sharing (ICS). The version of NAT used by ICS is scaled down form the full version, and does not allow for: the level of configuration that the RRAS NAT allows. ICS is for a small office or for a home network, where there is one Internet connection that is to be shared by the entire network. All users connect l via a single interface, usually connected via a modern, DSL, or cable access point. 2-

3.4.2 RRAS, RADIUS, and IAS

The Windows 2000 RRAS is made of several components, including: (1) Network Address Translation (NAT), (2) Routing protocols (RIP, OSPF), (3) VPN support (L2TP and PPTP), and (4) Demote Authentication Dial-In Service (RADIUS).

The Remote Access Server of RRAS allows for PPP connections and accomplish required authentication. For authentication, RRAS can use the Remote Authentication Dial-In User Service (RADIUS), or Windows Authentication. If RRAS is using RADIUS, when a user request for authentication is made to the RRAS server, the dial-in credentials are passed to the RADIUS server. The RADIUS server then performs the authentication and authorisation to access for the client to access the network.

The Remote Access Policy is controlled via the Internet Access Server (IAS), which is the Microsoft version of RADIUS. The RRAS server itself does not control the Remote Access Policy. The IAS performs several functions for remote users of the network, including authentication, authorization, auditing, and accounting to those users who connect to the network via dial-up and VPN connections. For authentication, IAS allows for great flexibility, accepting PAP, CHAP; MS-CHAP, and EAR EAP is , Extensible Authentication Protocol, and is used in conjunction with technologies such as: Smart Cards; Token Cards, and One-time passwords.

3.4.3 IPSec

IPSec is a framework for ensuring secure private communications over IP networks. IPSec provides security for transmission of critical and sensitive information over unprotected networks such as the Internet.

IPSec VPNs use the services defined within IPSec to ensure confidentiality, Integrity, and authenticity of data communications over the public network, like Internet. IPSec operates at the network layer, protecting and authenticating IP packets between participating IPSec devices. The IPSec provides the following network security services.

Data Confidentiality -The IPSec sender can encrypt packets before transmitting them across a network.

Data Integrity -The receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.

Data Origin Authentication -The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.

Anti-Replay -The IPSec receiver can detect and reject replayed packet.

In Windows 2000, you have two options for IPSec implementation, Transport Mode, and L2TP Tunnel Mode. Transport mode is designed for securing communication, between nodes 011 an internal network. L2TP Tunnel Mode is designed for securing communications between two networks.

IPSec Features

Two high level features of IPSec are the Authentication Header (AH) and the Encapsulated Security Payload (ESP). The AH is used to provide data communication with both integrity checking and source authentication and ESP is used to provide confidentiality. When using IPSec to secure communication, both the sender and the receiver (and only those two) know the security key used. Once authenticated, the receiver knows that the communication in-fact comes from the sender, and that the data has not been modified.

Since IPSec is works at the IP layer, it is able to secure communications with multiple protocols, including TCP, UDP, and ICMP. From a user viewpoint, the implementation of IPSec is transparent; the user is not required to modify user's environment in any way to use IPSec.

Windows 2000 IPSec Components

The Windows 2000 implementation of IPSec uses three components; (I) IPSec Policy Agent Service, (2) Internet Key Exchange (IKE), and Security Associations (SA). The IPSec Policy Agent Service gets the IPSec policy as configured in Active Directory, or the Registry, and provides that information to the~. Every Windows 2000 machine runs

the IPSec Policy Agent Service, and the policy is pulled when the system starts as Active Directory settings are applied.

The IKE manages Security Associations (SA) and creates and manages the actual authentication keys that are used to secure the communications. This happens in two distinct steps; (I) in the first step is the establishment of a secure authenticated channel of communication, and (2) the second step the Security Associations are determined. The as are used to specify both the security protocol and the key that will be implemented.

IPSec Implementation Options

The configuration may be applied in Active Directory or directly to the Registry. IPSec policies may be applied to computers, domains, OUs, or other GPOs in the Active Directory. The IPSec options are in Group Policy, under Security Settings.

There exist three policy options that are predefined for IPSec implementations. They are: Client (Respond Only), Server (Request Security), and Server (Require Security).

Client (Respond only) -As per this policy the secure communications are not secured most of the time. Computers with this policy respond to a request for secure communication by using a default response. If a client needs to access a secured server, it can use normal communications.

Server (Request Security) -Communication must be secured most of the time, and will allow unsecured communications from non IPSec-computers. It will request IPSec from the client first, and open a secured communication channel is tile client can respond securely.

Server (Require Security) -This policy states that communicatiol1 must always be secured al1d all traffic must use IPSec or it will not be accepted, and the connection will be dropped.

3.5 Encrypting File System Management

In this section we will discuss about the encryption of file system.

3.5.1 Encrypting File System (EFS)

The main benefits of personal computers are that it provides you the flexibility to boot into multiple Operating Systems for desired use. But this flexibility poses great difficulty in the world of security. In addition to the security risks of multiple Operating Systems, there are security

risks introduced with the use of laptop computers. Laptops often get stolen or misplaced, and the data on that computer is vulnerable to compromise as soon as the location of the laptop is changed. With NTFS security you are able to solve the issues of security to a certain extent. As detailed there are tools available to access data even properly secured on an NTFS partition.
The concept of encryption has been introduced to solve this problem. Data encryption works to make the files on the computer only useful to the authorised owner of the data. Some of these methods provide a password for each encrypted file, which while effective, is not practical for large volumes of files. Another method is to use a key to unlock each file that has been encrypted, with only one user holding the key and Microsoft's EFS uses this approach. EFS use "public key cryptography" for encryption decryption of data. Public key cryptography is the use of two keys, one performs encryption and another performs decryption. The keys are keys are mathematically related. The files are encrypted by DES encryption algorithm in EFS. EFS supports file encryption for both on a local hard drive and on a remote file server. But, any files encrypted on the remote server will be transmitted over the network in clear-text by default. So, the file is decrypted at the file server, and then sent to the user. In order to maintain the high level of security, a mechanism should be implemented to secure the network traffic, such as IPSec.

The implementation of EFS works directly with NTFS and data can only be encrypted on an NTFS partition. EFS can encrypt any temp files created along with the original, and the keys are stored in the kernel using non-paged memory, so they are never vulnerable to attackers.

3.5.2 EFS and Users Management

One of good or bad point of EFS is that its use does not require, any administrative effort and keys are created automatically, if the user does not already have a public key pair to use. Files and Folders are encrypted on a single file or single folder basis, each with a unique encryption key and as they are encrypted uniquely, if you move an encrypted file to an unencrypted folder on the same partition, the file will remain encrypted. If you copy an encrypted file to a location that allows for encryption, the file will remain encrypted.

The EFS is a very transparent in use and user may have encryption enabled without aware of it.

3.5.3 Data Recovery Management

EFS designed to be implemented by a user, and is designed to be transparent; it can be used where it was not initially intended. EFS allow for Recovery Agents and the default Recovery Agent is the Administrator. These agents have configured public keys that are used to enable file recovery process. But, the system is designed in such a way that only the file recovery is possible and the recovery agent cannot learn about the user's private key.

Data Recovery for those companies and organisations that have the requirement of accessing data if an employee leaves, or the encryption key is lost.

The policy for implementing Data Recovery is defined at a Domain Controller. And this policy will be enforced on every computer in that domain. In case EFS is implemented on a machine that is not part of a domain, the system, will automatically generate and save Recovery Keys.

3.5.4 EFS Cryptography Management

As mentioned in the previous sections EFS uses public key cryptography, based on 1he DES encryption algorithm. Data is encrypted by what is called a File Encryption Key (FEK), which is randomly generated key. The FEK itself is then encrypted using a public key, which creates a list of encrypted FEKs. The list is then stored with the encrypted file in a special attribute called the Data Decryption Field (DDF). When a user needs to decrypt the file, he or she will use the private key that was part of the key pair. User performs encryption from the command line, or from Explorer. In Explorer, the option to encrypt is under the advanced option on the properties Window. When using the command line version, the command is cipher, with ale switch for encryption and aid switch for decryption.

4.0 CONCLUSION

This concluding unit of this course has introduced you to the concepts of management and configuration required for the management of Microsoft windows computers.

Now try to examine everything from the foundational principles of Windows 2000 security up to the advanced issues of management of windows 2000 machines running Active Directory.

5.0 SUMMARY

This unit covers in detail the various security and management issues that can be implemented in windows 2000 architecture. The unit address

broad sweep of security and management related issues: User Authentication Management- users and group management; resource management; windows network management; and encrypting file system management. Windows 2000 authentication covers the Subsystems components, and Kerberos.

The users and group security in unit covers the topics like; configuring users accounts, windows 2000 groups (default group types, local groups, global groups, group policies etc), security configuration tools, and configuration and analysis tools. In unit covered the resource management in detail and it covers the areas like; files and folder management, files/folder permissions, inheritance and propagation, moving data and permissions, shared resources, null session, printer management, and Registry management.

The network management has been covered in detail in this unit and various network security methods like NAT, ICS,RRAS, RAS, IAS, and IPSec are covered. The unit also talks about the EFS (Encrypting File System) management of Windows 2000 systems and it covers topics like data recovery and EFS cryptography. This unit introduced the management of configuration required to secure Microsoft Windows Computer Systems and now you will be able to examine everything from the foundation principles of Windows 2000 security and management, up to the advanced issues of securing windows 2000 running Active Directories.

6.0 TUTOR-MARKED ASSIGNMENT

1) State True or False

   a. Kerberos is an IETF standard used for privacy.
   b. SPPI is Server Support Interface.
   c. The SPPI functions as an interface between the user applications, such as the Web Browser, and the authentication method, such as NTLM or Kerberos.
   d. MD5 is Mirror Domain 5.

2) Name various methods of authentication available in windows operating system.
3) Describe Kerberos management in windows operating system.

4) a) Create three domain user accounts: trainee l, trainee 2, Trainee 3
   b) Limit trainee1 so that this account can logon to the network during 10 AM to 5 PM during week days (i.e., Monday to Friday).

c) Set expiry date for trainee3 from 3 days from the today's date.

5) Describe policy inheritance.
6) Create security template with following parameters.

Password History 30 passwords
Maximum Password age of 15 days and Minimum password age of 3 days.
Minimum password length 12 characters.
Account Lockout duration 0 minutes
Account Lockout Threshold of 4 invalid Logon attempts.
Reset Account Lockout Counter after 70 minutes

7) Expand the following:

a. RADIUS
b. NAT
c. ICS
d. RRAS

8) What do you understand by VPN? Discuss IPSec security.
9) Discuss in detail EFS (Encrypting File System)
10) What do you understand by a null session? How null session can be disabled?

7.0 REFERENCES/FURTHER READINGS

Windows 2000 Professional Resource Kit, Microsoft Press.

Cryptography and Network Security, Principles and Practice, SE, PE., William Stallings .

Security in Computer; Charles P. P fleeger and Shari Lawrence Pfleeger, Third Edition, Pearson Education.

Windows 2000 Commands by Aleen Frisch.

Microsoft Web Site http://www.microsoft.com.