NATIONAL OPEN UNIVERSITY OF NIGERIA

SCHOOL OF SCIENCE AND TECHNOLOGY

COURSE CODE: CIT 723

COURSE TITLE: OPERATING SYSTEM DESIGN AND PROGRAMMING

OPERATING SYSTEM DESIGN AND PROGRAMMING

MODULE ONE OPERATING SYSTEM FUNDAMENTAL

Unit 1: Introduction to operating system

Unit 2: Objectives of Operating System

Unit 3: Graphical User interface

UNIT ONE INTRODUCTION TO OPERATING SYSTEM

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
    3.1 What is Operating System
    3.2 Types of Operating System
    3.3 Views of Operating System
    3.4 Qualities of an Operating System
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about the definition of operating system as well as fundamentals of operating system.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Define an operating system
- Explain types and views of operating system
- Identify the qualities of an operating system

3.0 MAIN CONTENT

3.1 WHAT IS AN OPERATING SYSTEM?

The 1960's definition of an operating system is the software that controls the hardware. However, today, due to microcode we need a better definition. We see operating system as the programs that make the hardware useable. In brief, an operating system is the set of programs that controls computer hardware. It is a program that acts as an intermediary between a user and the computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner.

The operating system must ensure the correct operation of the computer system. To prevent user programs from interfering with the proper operation of the system, the hardware must provide appropriate mechanism to ensure proper behavior.

Some examples of operating systems are UNIX, Mach, MS-DOS, MS-Windows, Windows/NT, Chicago, O/S2, MacOS, VMS, MVS, and VM.

Controlling the computer involves software at several levels. We will differentiate kernel services, library services and application-level services, all of which are part of the operating system. Processes run applications, which are linked together with libraries that perform standard services. The kernel supports the processes by providing a path to the peripheral devices. The kernel responds to service calls from the processes and interrupts from the devices.

Operating system are resource managers. The main resource is computer hardware in the form of processors, storage, input/output, communication devices and data. Some of the operating system functions are: implementing the user interface, sharing hardware among users, allowing users to share data among themselves, preventing users from interfering with one another, scheduling resources among users, facilitating input/output, recovering from errors, accounting for resource usage, facilitating parallel operations, organizing data for secure and rapid access, and handling network communications.

3.2 TYPES OF OPERATING SYSTEM

Within the broad category of operating systems, there are in general four types, classified based on the types of computers they control and the kind of application they support. The broad categories are:

Real-Time Operating System (RTOS): Is commonly used to control machinery, scientific instruments and industrial systems. An RTOS typically has very little user interface capability, and no end-user utilities, since the system will be a "sealed box" when delivered for use.

Single-user, single-tasking: As the name implies, this operating system is designed to manage the computer so that one user can effectively do one thing at a time. The palm OS for palm handheld computers is a good example of a modern single user, single task operating system.

Single-user, Multi-tasking: This is the type of operating system most people use on their desktop and laptop computers today. Microsoft's windows and Apple's MacOS platforms are both examples of operating systems that will let a single user have several programs in operation at the same time. For example, it's entirely possible for a windows user to be writing a note in a word processor while downloading a file from the internet while printing the text of an e-mail message.

Multi-User: A multi-user operating system allows many different users to take the advantages of the computer's resources simultaneously. The operating system must make sure that the requirements of the various users are balanced, and that each of the programs they are using has sufficient and separate resources so that a problem with one user doesn't affect the entire community of users. UNIX, VMS, and mainframe operating systems, such as MVS, are examples of multi-user operating systems.

3.3 VIEWS OF OPERATING SYSTEM

Operating system is a hard term to define. What you consider an operating system depends on your view of the system.

As a scheduler/resource allocator:

The operating system has resources for which it is in charge, responsible for handling them out (and later recovering them). Resources include CPU, memory, I/O devices, and disk space.

As a virtual machine:

Operating system provides a "new" machine. This machine could be the same as the underlying machine. Permits many users to believe they have an entire piece of hardware to themselves.

As a multiplexor:

Allows sharing of resources, and provides protection from interference and provides for a level of cooperation between users.

3.4 QUALITIES OF AN OPERATING SYSTEM

What are the desirable qualities of an operating system? We can discuss them in terms of: Usability, Facilities, Cost, and Adaptability.

USABILITY

1. Robustness: accept all valid input without error, and gracefully handle all invalid inputs
2. Proportionality: Simple, cheap and frequent things are easy. Also, expensive and disastrous things are hard.
3. Forgiving: Errors can be recovered from.
4. Convenient: Not necessary to repeat things, or do awkward procedures to accomplish things.
5. Powerful: Has high level facilities.

FACILITIES

1. Sufficient for intended use
2. Complete: Don't leave out part of a facility
3. Appropriate: Do not use fixed-width field input from terminal.

COST

1. Want low cost and efficient services
2. Good algorithms: Make use of space/time tradeoffs, special hardware.
3. Low overhead: cost of doing nothing should be low. e.g. idle time at a terminal

ADAPTABILITY

1. Tailored to the environment: Support necessary activities. Do not impose unnecessary restrictions. What are the things people do most - make them easy.
2. Changeable over time: Adapt as needs and resources change. e.g. expanding memory and new devices of new user population.
3. Extendible-Extensible: Adding new facilities and features

4.0 CONCLUSION

You have learned about the definition of operating system as well as fundamentals of operating system.

ACTIVITY B

1. State the features of operating system.

5.0 SUMMARY

What you have learned in this unit borders on the definition of operating system as well as fundamentals of operating system.

6.0 TUTOR MARKED ASSIGNMENT

1. What is an operating system?
2. Explain briefly the types of operating system
3. Discuss the qualities of an operating system

7.0 REFERENCES/FUTHER READINGS

1. Lecture notes on operating system by Jelena Mamcenko, Vilinus Gediminas Technical University, 2010.
2. Dictionary of Computing, Fourth Ed. (Oxford: Oxford University Press, 1996).
3. History of Operating Systems by Ayman Moumina, 2001
4. A short introduction to operating system by Mark Burgess, 2002.
5. Operating system handbook by Bob Ducharme- McGraw-Hill, 1994.

UNIT TWO

OBJECTIVES OF OPERATING SYSTEM

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
  3.1 History of Operating System
  3.2 Objectives of Operating System
  3.3 Operating System Services
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about the history of operating system, the objectives of operating system and operating system services.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Describe the generations of computers
- Explain the objectives of operating system
- Explain operating system services

3.0 MAIN CONTENT

3.1 HISTORY OF OPERATING SYSTEMS

Historically, operating systems have been highly related to the computer architecture. It is good idea to study the history of operating system from the architecture of the computers on which they run.

FIRST GENERATION

The earliest electronic digital computers had no operating systems. Machines of the time were so primitive that programs were often entered one bit at a time on rows of mechanical switches (plug boards). Programming languages were unknown (not even assembly languages). Operating systems were unheard of.

SECOND GENERATION

By the early 1950's, the routine has improved somewhat with the introduction of punch cards. The system of the 50's generally ran one job at a time. These were called single-stream batch processing systems because programs and data were submitted in groups or batches.

THIRD GENERATION

The systems of the 1960's were also batch processing systems, but they were able to take better advantage of the computer's resources by running several jobs at once. So, operating system designers developed the concept of multiprogramming in which several jobs are in main memory at once, a processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral device in use. For example, on the system with no multiprogramming, when the current job paused to wait for other I/O operation to complete, the CPU simply sat idle until the I/O is finished. The solution for this problem that evolved was to partition the memory into several pieces, with a different job in each partition. While one job was waiting for I/O to complete, another job could be using the CPU.

Another major feature in the third generation operating system was the technique called spooling (simultaneous peripheral operations online). In spooling, a high-speed device like a disk interposed between a running program and a low speed device involved with the program in input/output. Instead of writing directly to a printer, for example, outputs are written to the disk. Programs can run to completion faster, and other programs can be initiated sooner when the printer becomes available, the output may be printed.

Another feature present in this generation was time sharing technique, a variant of multiprogramming technique, in which each user has an on-line (i.e. directly connected), and terminal. Because the user is present and interacting with the computer, the computer system must respond quickly to user requests, otherwise user productivity could suffer. Timesharing systems were developed to multi-program large number of simultaneous interactive users.

FORTH GENERATION

With the development of LSI (Large scale integration) circuits, chips, operating system entered into the personal computer and workstation age. Microprocessor technology evolved to the point that it becomes possible to build desktop computers as powerful as the mainframes of the 1970s.

3.2 OBJECTIVES OF OPERATING SYSTEM

Modern operating systems generally have the following three major goals. Operating systems generally accomplish these goals by running processes in low privilege and providing service calls that invoke the operating system kernel in high-privilege state.

- To hide details of hardware by creating abstraction
  An abstraction is software that hides low level details and provides a set of higher level functions. An operating system transforms the physical world of devices, instructions, memory, and time into virtual world that is the result of abstraction built by the operating system. These are several reasons for abstraction. Firstly, the code needed to control peripheral devices is not standardized. Operating systems provide subroutines called a device driver that performs operation on behalf of programs for example, input/output operations. Secondly, the operating system introduces new functions as it abstracts the hardware. For instance, operating system introduces the file abstraction so that programs do not have to deal with disks. Thirdly, the operating system transforms the computer hardware into multiple virtual computers, each belonging to a different program. Each program that is running is called a process. Each process views the hardware through the lens of abstraction and lastly, the operating system can enforce security through abstraction.
- To allocate resources to processes (Mange resources)

  An operating system controls how processes (the active agents) may access resources (passive entities).

- Provide a pleasant and effective user interface
  The user interacts with the operating system through the user interface and usually interested in the “look and feel” of the operating system. The most important components of the user interface are the command interpreter, the file system, on-line help, and application integration. The recent trend has been toward increasingly integrated graphical user interfaces that encompass the activities of multiple processes on networks of computers.

One can view operating systems from two points of views: Resource Manager and Extended Machines. From resource manager point of view, Operating systems manage the different parts of the system efficiently and from extended machine point of view, Operating systems provide a virtual machine to users, that is, more convenient to use. The structural Operating system can be designed as a monolithic system, a hierarchy of layers, a virtual machine system, an exo-kernel, or using the client-server model. The basic concepts of Operating systems are processes, memory management, I/O management, the file systems, and security.

3.3 OPERATING SYSTEM SERVICES

The operating system provides certain services to programs and to the users of those programs in order to make the programming task easier.

Basically the functions of an operating system are:

1. Program execution
2. I/O operations
3. File system manipulation
4. Error detection
5. Communication

PROGRAM EXECUTION

The purpose of a computer system is to allow the user to execute programs. So the operating system provides an environment where the user can conveniently run programs. The user does not have to worry about the memory allocation or multitasking or anything. These things are taken care of by the operating systems.

Running a program involves the allocating and de-allocating memory, CPU scheduling in case of multi-process. These functions cannot be given to user level programs. So user-level programs cannot help the user to run programs independently without the help from the operating systems.

I/O OPERATIONS

Each program requires an input and produces output. This involves the use of I/O. the operating system hides the user the details of underlying hardware for the the I/O. all the user see is that the I/O has been performed without any details. So the operating system by providing I/O makes it convenient for the user to run programs.

For efficiency and protection, users cannot control I/O so this service cannot be provided by user-level programs.

FILE SYSTEM MANIPULATION

The output of a program may need to be written into new files or input taken from some files. The operating systems provide this service. The user does not have to worry about secondary storage management. User gives a command for reading or writing to a file and sees his task accomplished. Thus operating systems can make it easier for user programs to accomplish their task.

This service involves secondary storage management. The speed of I/O that depends on secondary storage management is critical to the speed of many programs and hence I believe it is best relegated to the operating systems to manage it than giving individual users the control of it. It is not difficult for the user-level programs to provide these services but for above mention reasons it is best if this service is left with the operating system.

ERROR DETECTION

An error in one part of the system may cause malfunctioning of the complete system. To avoid such a situation the operating system constantly monitors the system for detecting the errors. This relieves the user of the worry of errors propagating to various part of the system and causing malfunctioning.

This service cannot be allowed to be handled by user programs because it involves monitoring and in cases altering area of memory or de-allocation of memory for a faulty process or may be relinquishing the CPU of a process that goes into an infinite loop. These tasks are too critical to be handed over to user programs. A user program if given the privileges can interfere with the correct (normal) operation of the operating systems.

COMMUNICATIONS

There are instances where processes need to communicate with each other to exchange information. It may be between processes running on the same computer or running on different computers. By providing this service the operating system relieves the user of the worry of passing messages between processes. In case where the messages need to be passed to processes on the other computers through a network it can be done by the user programs. The user programs may be customized to the specifics of the hardware through which the message transits and provides the service interface to the operating system.

4.0 CONCLUSION

You have learned about the history of operating system, the objectives of operating system and operating system services.

ACTIVITY B

1. Briefly describe the generations of computers
2. Describe the operating system services

5.0 SUMMARY

What you have learned in this unit borders on the Generation of computers and objectives of operating system as well as operating system services.

6.0 TUTOR MARKED ASSIGNMENT

1. What are the basic objectives of an operating system?

REFERENCES/FUTHER READINGS

1. Lecture notes on operating system by Jelena Mamcenko, Vilinus Gediminas Technical University, 2010.
2. Dictionary of Computing, Fourth Ed. (Oxford: Oxford University Press, 1996).
3. History of Operating Systems by Ayman Moumina, 2001
4. A short introduction to operating system by Mark Burgess, 2002.
5. Operating system handbook by Bob Ducharme- McGraw-Hill, 1994.

UNIT THREE

GRAPHICAL USER INTERFACE

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
    3.1 What is GUI?
    3.2 Elements Of Graphical User Interface
    3.3 User Interface And Interaction Design
    3.4 Comparison To Other Interfaces
    3.5 Three-Dimensional User Interface
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about what GUI is and its element. Also, you will learn user interface and interaction design, comparison with other interfaces as well as 3-dimensional user interface.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Define GUI
- State the elements of GUI
- Explain User interface and interaction design
- Compare GUI with other interfaces
- Describe 3D GUI

3.0 MAIN CONTENT

3.1 What is GUI?

In computing, a graphical user interface (GUI, sometimes pronounced gooey) is a type of user interface that allows users to interact with electronic devices with images rather than text commands. GUIs can be used in computers, hand-held devices such as MP3 players, portable media players or gaming devices, household appliances and office equipment. A GUI represents the information and actions available to a user through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. The actions are usually performed through direct manipulation of the graphical elements.

The term GUI is historically restricted to the scope of two-dimensional display screens with display resolutions able to describe generic information, in the tradition of the computer science research at the PARC (Palo Alto Research Center). The term GUI earlier might have been applicable to other high-resolution types of interfaces that are non-generic, such as video games, or not restricted to flat screens, like volumetric displays.

3.2 Element of Graphical User Interface

A GUI uses a combination of technologies and devices to provide a platform the user can interact with, for the tasks of gathering and producing information.

A series of elements conforming to a visual language have evolved to represent information stored in computers. This makes it easier for people with few computer skills to work with and use computer software. The most common combination of such elements in GUIs is the WIMP ("window, icon, menu, and pointing device") paradigm, especially in personal computers.

The WIMP style of interaction uses a physical input device to control the position of a cursor and presents information organized in windows and represented with icons. Available commands are compiled together in menus, and actions are performed making gestures with the pointing device. A window manager facilitates the interactions between windows, applications, and the windowing system. The windowing system handles hardware devices such as pointing devices and graphics hardware, as well as the positioning of the cursor.

In personal computers all these elements are modeled through a desktop metaphor, to produce a simulation called a desktop environment in which the display represents a desktop, upon which documents and folders of documents can be placed. Window managers and other software combine to simulate the desktop environment with varying degrees of realism.

Desktop Graphics

Both Windows and Macintosh systems are based on Graphical User Interface or GUI, which simply means that the interface uses graphics or pictures to help the user navigate and access programs. When you first turn on a new computer, most of the screen will be plain blue or blue with a logo or design. This background graphic is called Wallpaper. It is essentially a backdrop for your work area. The graphic can be changed to a different pattern or even a photo or picture by accessing "Display" in the Control Panel.

Another important graphic feature that you will find on a desktop is an icon. Icons are small pictures that are linked to programs. Double-clicking on the icon runs the program or accesses the file and right-clicking accesses a menu offering options, actions and properties. Certain icons are a permanent fixture on the desktop. The user can put other icons on the desktop that will quickly access programs or files - like a shortcut. Icons can be moved around on the desktop by clicking and dragging them.

One of the most important icons on the desktop is My Computer, which accesses drives, printers, the Control Panel and other system applications. The Control Panel gives the user access to the computer system and many support applications, such as "Add/Remove Programs" and "Accessibility Options". From the Control Panel, you can access hardware settings for the keyboard, mouse, printers and modem; as well as settings for the monitor display and sound.

Another important icon that you should know about is the Recycle Bin. It has the same purpose that a real trash can does - you put things in it that you no longer need or want. Anytime you delete a file or folder, it goes into the Recycle Bin where it stays until the bin is emptied. Double-clicking on the icon will open a window that will show you what is stored in the Recycle Bin. Just like in real life, things sometimes get thrown away by accident and have to be rescued. The Recycle Bin lets you do the same thing. If you delete something you shouldn't have, you can find it in the Recycle Bin and restore it to its proper place. When the Recycle Bin is emptied, everything in it is permanently deleted. Never put anything in the Recycle Bin or empty the Recycle Bin without permission!

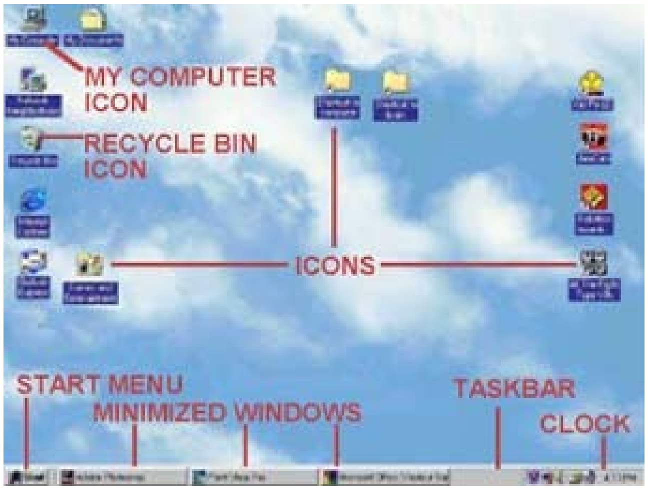


Fig. 1.1: Desktop Window Environment

The Start Menu and Taskbar

At the edge of the screen (usually the bottom edge), you will see a long, thin bar with a box labeled "Start" on one end and a clock on the other end. This is the taskbar - another graphic tool that helps you to access programs and files. You may see icons on the taskbar, too. These are called "Quick Launch" icons that allow one-click access to frequently used programs.

If you click on the "Start" button, a box called a menu will appear. The menu will contain several words. Some words will have arrows next to them that access other menus. Listed below are the basic Start-menu options:

- Programs - accesses installed programs.
- Favorites - accesses book-marked web pages.
- Documents - accesses the most recently opened documents.
- Settings - accesses system applications such as Control Panel, printers, taskbar and Start Menu options.
- Search- searches for specific files or folders.
- Help - offers helpful topics for computer use.
- Run - user can input commands to run specific programs.
- Log Off - allows a password-protected user to log off and another to log on.
- Shut Down - shuts down or restarts the computer.

The Start Menu can be personalized by adding and removing programs, files and folders.

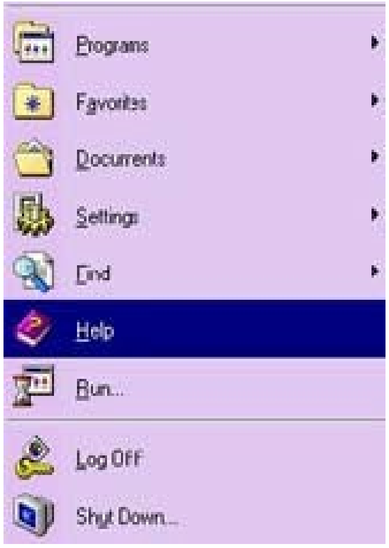


Fig. 1.2: Window Task Bar

Windows (not the operating system)

Many programs and applications run within windows or boxes that can be opened, minimized, resized and closed. At the top of each window, you will see a title bar that contains the title of the program or folder. To the right of the title bar are three square icons or buttons. The button on the far right has an "X" on it and closes the window (which also closes the program). The middle button will have one or two small boxes on it - this is the resize button. Resizing allows the user to make the window full-screen or smaller.

The button on the left has a small line on it - this is the minimize button. When a window is open, you will see a rectangular button on the taskbar that displays the windows title and maybe an icon. Minimizing the window clears it from the screen, but keeps the program running - all you will see of a minimized window is the button on the taskbar. A raised button indicates a minimized or inactive window, and a depressed button indicates an open or active window. Minimizing a window is helpful if the user is temporarily not using the program, but plans to return to it soon. To restore a minimized window, simply click on the button on the taskbar. Also, it is sometimes possible to have several windows open at once and lined up in a cascade, one in front of another. In this case, the active window will always be in the front.

In the Windows operating system, each window contains its own menu. Found just under the title bar, the menu contains several words that will access drop-down menus of options and actions. The menus vary from one program to another, but here are some basic examples:

- File menu contains file items like new, open, close, and print.
- Edit menu contains editing items like undo, cut, copy, paste and clear.
- View menu contains items for changing how things appear on the screen.
- Help menu contains items to access tutorials or helpful information.

Under the menu, you will often find a toolbar - a bar of icons or options that allow you to perform specific operations within the program.

In the main body of the window, you may find lists of folders or files or you may be given a workspace for typing, drawing or other activities. On the right side of the window, you may see a scroll bar. The scroll bar appears when the window contains more information that can fit on the screen. Moving the scroll bar up or down allows the user to see all of the information in the window.

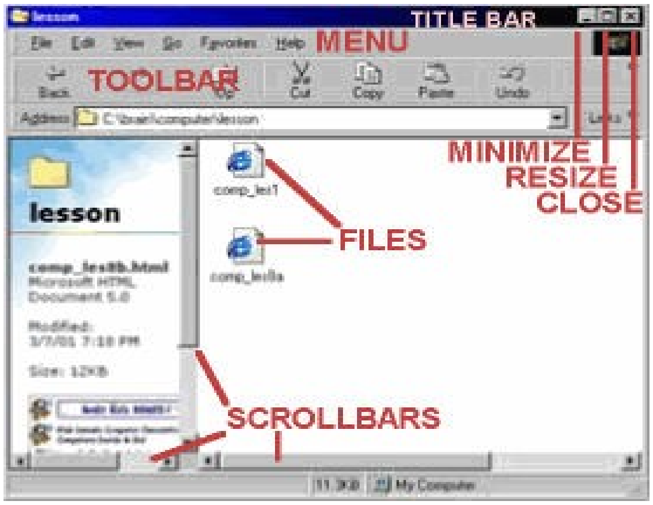


Fig. 1.3: Window Environment

Post-WIMP Interface

Smaller mobile devices such as PDAs and smart-phones typically use the WIMP elements with different unifying metaphors, due to constraints in space and available input devices. Applications for which WIMP is not well suited may use newer interaction techniques, collectively named as post-WIMP user interfaces.

As of 2011, some touch-screen-based operating systems such as Android and Apple's iOS (iPhone) use the class of GUIs named post-WIMP. These support styles of interaction using more than one finger in contact with a display, which allows actions such as pinching and rotating, which are unsupported by one pointer and mouse.

Post-WIMP include 3D compositing window managers such as Compiz, Desktop Window Manager, and LG3D. Some post-WIMP interfaces may be better suited for applications which model immersive 3D environments, such as Google Earth.

3.3 USER INTERFACE AND INTERACTION DESIGN

Designing the visual composition and temporal behavior of GUI is an important part of software application programming. Its goal is to enhance the efficiency and ease of use for the underlying logical design of a stored program, a design discipline known as usability. Methods of user-centered design are used to ensure that the visual language introduced in the design is well tailored to the tasks it must perform.

Typically, the user interacts with information by manipulating visual widgets that allow for interactions appropriate to the kind of data they hold. The widgets of a well-designed interface are selected to support the actions necessary to achieve the goals of the user. A Model-view-controller allows for a flexible structure in which the interface is independent from and indirectly linked to application functionality, so the GUI can be easily customized. This allows the user to select or design a different skin at will, and eases the designer's work to change the interface as the user needs evolve. Nevertheless, good user interface design relates to the user, not the system architecture.

The visible graphical interface features of an application are sometimes referred to as "chrome". Larger widgets, such as windows, usually provide a frame or container for the main presentation content such as a web page, email message or drawing. Smaller ones usually act as a user-input tool.

A GUI may be designed for the rigorous requirements of a vertical market. This is known as an "application specific graphical user interface." Among early application specific GUIs was Gene Mosher's 1986 Point of Sale touch-screen GUI. Other examples of an application specific GUIs are:

- Self-service checkouts used in a retail store
- Automated teller machines (ATM)
- Airline self-ticketing and check-in
- Information kiosks in a public space, like a train station or a museum
- Monitors or control screens in an embedded industrial application which employ a real time operating system (RTOS).

The latest cell phones and handheld game systems also employ application specific touch-screen GUIs. Newer automobiles use GUIs in their navigation systems and touch screen multimedia centers.

3.4 COMPARISON TO OTHER INTERFACES

GUIs were introduced in reaction to the steep learning curve of command-line interfaces (CLI), which require commands to be typed on the keyboard. Since the commands available in command line interfaces can be numerous, complicated operations can be completed using a short sequence of words and symbols. This allows for greater efficiency and productivity once many commands are learned, but reaching this level takes some time because the command words are not easily discoverable and not mnemonic. WIMPs ("window, icon, menu, pointing device"), on the other hand, present the user with numerous widgets that represent and can trigger some of the system's available commands.

WIMPs extensively use modes as the meaning of all keys and clicks on specific positions on the screen are redefined all the time. Command line interfaces use modes only in limited forms, such as the current directory and environment variables.

Most modern operating systems provide both a GUI and some level of a CLI, although the GUIs usually receive more attention. The GUI is usually WIMP-based, although occasionally other metaphors surface, such as those used in Microsoft Bob, 3dwm or File System Visualizer (FSV).

Applications may also provide both interfaces, and when they do the GUI is usually a WIMP wrapper around the command-line version. This is especially common with applications designed for Unix-like operating systems. The latter used to be implemented first because it allowed the developers to focus exclusively on their product's functionality without bothering about interface details such as designing icons and placing buttons. Designing programs this way also allows users to run the program non-interactively, such as in a shell script.

3.5 THREE-DIMENSIONAL USER INTERFACES

For typical computer display, three-dimensional is a misnomer—their displays are two-dimensional. Semantically, however, most graphical user interfaces use three dimensions - in addition to height and width, they offer a third dimension of layering or stacking screen elements over one another. This may be represented visually on screen through an illusionary transparent effect, which offers the advantage that information in background windows may still be read, if not interacted with. Or the environment may simply hide the background information, possibly making the distinction apparent by drawing a drop shadow effect over it.

Some environments use the methods of 3D graphics to project virtual three dimensional user interface objects onto the screen. As the processing power of computer graphics hardware increases, this becomes less of an obstacle to a smooth user experience.

Motivation

Three-dimensional GUIs are quite common in science fiction literature and movies, such as in Jurassic Park, which features Silicon Graphics' three-dimensional file manager, "File system navigator", an actual file manager that never got much widespread use as the user interface for a Unix computer. In fiction, three-dimensional user interfaces are often immersible environments like William Gibson's Cyberspace or Neal Stephenson's Metaverse.

Three-dimensional graphics are currently mostly used in computer games, art and computer-aided design (CAD). There have been several attempts at making three-dimensional desktop environments like Sun's Project Looking Glass or SphereXP from Sphere Inc. A three-dimensional computing environment could possibly be used for collaborative work. For example, scientists could study three-dimensional models of molecules in a virtual reality environment, or engineers could work on assembling a three-dimensional model of an airplane. This is a goal of the Croquet project and Project Looking Glass.

Technologies

The use of three-dimensional graphics has become increasingly common in mainstream operating systems, from creating attractive interfaces—eye candy— to functional purposes only possible using three dimensions. For example, user switching is represented by rotating a cube whose faces are each user's workspace, and window management is represented via a Rolodex-style flipping mechanism in Windows Vista (see Windows Flip 3D). In both cases, the operating system transforms windows on-the-fly while continuing to update the content of those windows.

Interfaces for the X Window System have also implemented advanced three-dimensional user interfaces through compositing window managers such as Beryl, Compiz and KWin using the AIGLX or XGL architectures, allowing for the usage of OpenGL to animate the user's interactions with the desktop.

Another branch in the three-dimensional desktop environment is the three-dimensional GUIs that take the desktop metaphor a step further, like the BumpTop, where a user can manipulate documents and windows as if they were "real world" documents, with realistic movement and physics.

The Zooming User Interface (ZUI) is a related technology that promises to deliver the representation benefits of 3D environments without their usability drawbacks of orientation problems and hidden objects. It is a logical advancement on the GUI, blending some three-dimensional movement with two-dimensional or "2.5D" vector objects.

4.0 CONCLUSION

You would have learned about what GUI is and its element, user interface and interaction design, comparison with other interfaces as well as 3-dimensional user interface.

5.0 SUMMARY

You have learned the definition, elements and function of a graphical user interface. Also, you have studied the differences between GUI and other interfaces.

ACTIVITY B

1. State the elements of GUI

6.0 TUTOR MARKED ASSIGNMENT

1. What is graphical user interface?

7.0 REFERENCES/FUTHER READINGS

1. Lecture notes on operating system by Jelena Mamcenko, Vilinus Gediminas Technical University, 2010.
2. Dictionary of Computing, Fourth Ed. (Oxford: Oxford University Press, 1996).
3. History of Operating Systems by Ayman Moumina, 2001
4. A short introduction to operating system by Mark Burgess, 2002.
5. Operating system handbook by Bob Ducharme- McGraw-Hill, 1994.

MODULE TWO

INTERNETWORKING

Unit 1: Introduction to networking concept

Unit 2: User to User Communication

Unit 3: Network Architecture

Unit 4: Networking Protocols

UNIT ONE

INTRODUCTION TO NETWORKING CONCEPT

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
    3.1 Definition

3.2 Types of Network
3.3 Peer-to-Peer Network Model
3.4 Purpose of a Network
3.5 Benefits of Networking

4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about the definition of network, types of network, purpose of network and benefits of networking.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Define a network and internetwork
- Explain types and purpose of network
- Explain benefits of networking

3.0 MAIN CONTENTS

3.1 DEFINITION

A network can be defined as a group of computers and other devices connected in some ways so as to be able to exchange data. Each of the devices on the network can be thought of as a node; each node has a unique address.
Addresses are numeric quantities that are easy for computers to work with, but not for humans to remember.
Example: 204.160.241.98
Some networks also provide names that humans can more easily remember than numbers.
Example: www.javasoft.com, corresponding to the above numeric address.

3.2 TYPES OF NETWORKS

There are two principle kinds of networks: Wide Area Networks (WANs) and Local Area Networks (LANs).

WANs

- Cover cities, countries, and continents.

- Based on packet switching technology
- Examples of WAN technology: Asynchronous Transfer Mode (ATM), Integrated Services Digital Network (ISDN)

LANs

- Cover buildings or a set of closely related buildings.
- Examples of LAN technology: Ethernet, Token Ring, and Fiber Distributed Data Interconnect (FDDI).

Ethernet LANs: Based on a bus topology and broadcast communication

Token ring LANs: Based on ring topology

FDDI LANs: Use optical fibers and an improved token ring mechanism based on two rings flowing in opposite directions.

3.3. PEER-TO-PEER NETWORK MODEL

Peer-to-peer network

A network where any computer can communicate with other networked computers on an equal or peer-like basis without going through an intermediary, such as a server or host. This is often used in very small organizations, such as a two to ten person office.

Advantages of Peer-to-Peer Networking

- A group of computers can share files, folders, and printers
- Peer-to-peer networking is easy to set up
- Supports using workgroups (Microsoft workgroup is a number of users who share drive and printer resources in an independent peer-to-peer relationship.)

Disadvantages of Peer-to-Peer Networking

- It offers only moderate network security
- No centralized storage or account management
- Not effective for complex network management
- Not optimized for simultaneous access by over 9 or 10 computers

Server-Based Network Model

This is a model in which access to the network, to resources, and the management of resources is accomplished through one or more servers. It is used particularly in medium and large organizations.

Advantages of the Server-Based Model

- It provides extensive multiuser access to resources
- It is Ideal for coordinated server and network management
- It provides robust security to network resources
- It contributes to fast network performance

Disadvantages of the Server- Based Model

- Generally requires more advanced planning than peer-to-peer networking
- Can be more complex to set up than peer-to-peer networking

3.4 PURPOSE OF A NETWORK

The general purpose of a network is to transmit information between two or more devices. This usually consists of one system sending a request for information to another system, which then acts upon the request and returns some sort of information back to the requesting system. Sometimes these systems are computers, and sometimes not; they could also be printers, bank teller machines, or telephones. Sometimes these systems are on the same piece of wire, and sometimes they are located on different continents, connected via the Internet or some other global network.

In order to successfully deliver information between the devices on a network, several steps must occur:

1. The originating system has to package the information in a manner which both systems understand.

2. The sender must then deliver the package to the destination, using techniques that are commonly understood by the systems and the network alike (these packaging and delivery functions are defined as "protocols").

3. The destination system, upon receiving the package, must check for any errors which may have incurred during transmission.

4. It must then unpack the package, and pass the data to a local application for processing.

5. If any information is to be returned to the first system, the above process must be repeated.

Although this process is oversimplified somewhat, it describes the basic operation of most communication technologies. Where things start to get complicated is in the differences between the various technologies and products that provide these functions.

Since most network technologies are designed for a specific use, they tend to be highly- optimized for specific environments. This optimization results in specific benefits (and liabilities) that are a direct result of the design goals. For example, modem cables and printer cables are extremely different entities, as are the mechanisms used to provide services across them, although both provide "network" services.

Modems typically use serial cables, which are designed to transmit data one bit a time. While slow, the one-bit-at-a-time design is necessary for devices like modems that rely on the historically noisy public telephone network for data transmission. Every bit needs to be verified for accuracy, so they are sent as single units.

Conversely, printers are typically attached directly to a PC and do not encounter much network interference. This allows for the use of parallel communication cables which are able to transmit multiple bits of information simultaneously. Because they do not need to conduct much error checking, they can transmit much more information simultaneously. Figure 2.1 below illustrates the difference between networks based on serial cables and parallel cables:

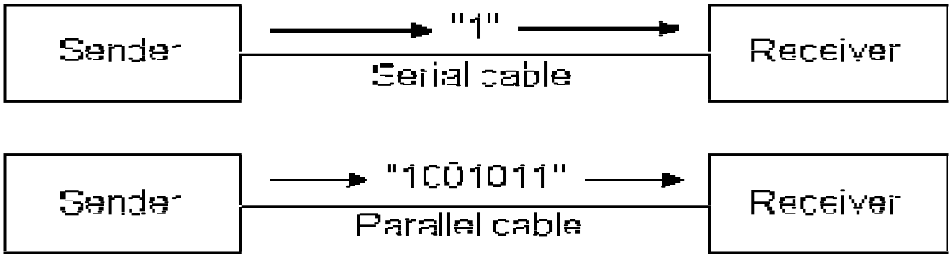


Figure 2.1: Serial cables send data one bit at a time, while parallel cables send data eight (or more) bits at a time.

Parallel cables are much faster than serial cables. However, the applications and services which use serial cables are much more robust than those that use parallel cables, because they have to be able to deal with more signaling noise and errors.

3.5 BENEFITS OF NETWORKING

- File sharing: Network file sharing between computers gives you more flexibility than using floppy drives or Zip drives. Not only share photos, music files, and documents, you can also use a home network to save copies of all of your important data on a different computer. Backups are one of the most critical yet overlooked tasks in home networking.
- Printer / peripheral sharing: Once a home network is in place, it's easy to then set up all of the computers to share a single printer. No longer will you need to bounce from one

system or another just to print out an email message. Other computer peripherals can be shared similarly such as network scanners, Web cams, and CD burners.

- Internet connection sharing: Using a home network, multiple family members can access the Internet simultaneously without having to pay the Internet Service Provider (ISP) for multiple accounts.
- Multi-player games: Many popular home computer games support LAN mode where friends and family can play together, if they have their computers networked.
- Internet telephone service: So-called Voice over IP (VoIP) services allows you to make and receive phone calls through your home network across the Internet, saving you money.
- Home entertainment: Newer home entertainment products such as digital video recorders (DVRs) and video game consoles now support either wired or wireless home networking. Having these products integrated into your network enables online Internet gaming, video sharing and other advanced features

4.0 CONCLUSION

You would have learned about the definition of network, types of network, purpose of network and benefits of networking.

5.0 SUMMARY

You have studied the various types of network, the purpose of a network and the benefits of networking.

ACTIVITY B

1. Briefly explain the benefits of networking

6.0 TUTOR MARKED ASSIGNMENT

1. Differentiate between LAN and WAN

7.0 REFERENCES/FUTHER READINGS

1. Introduction to Computer Networks by William Stallings 9th Edition, Publisher – Prentice Hall, ISBN: 0131392050.
2. Understanding and Designing Computer Networks by Graham King. Publisher – Edward Amold, ISBN: 0-340-61419-6.
3. Network Design by Mani Subramanian. Publisher-Addison-Wesley, ISBN: 0-201-35742-9.

UNIT TWO

USER TO USER COMMUNICATION

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
    3.1 On-Line Communication
    3.2 Off-Line Communication
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

When UNIX was being developed, one of the avowed aims was to provide users with an environment which facilitated sharing of information. This implied good facilities for communication between users.

Unlike the other commands which you can learn by solitary effort, the communication features are best mastered by working with a partner to whom you can send practice messages. Of course this is not an essential requirement, and you could, if necessary, learn even the communication commands all by yourself.

2.0 OBJECTIVE

At the end of this unit you shou1d be able to:

- Explain on-line communication
- Explain off-line communication

3.0 MAIN CONTENT

3.1 ON-LINE COMMUNICATION

We will first look at the write command, which allows you to send a message to another user logged in onto the same machine at the same time. This is thus a means of on-line communication because the recipient obtains the message immediately. There are three other means of communication in UNIX that we will consider in later sections, and two of them are off line in that you send the message and the intended recipient might not pay any attention to it if he so desires.

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 confusion. The thing to understand here is there is no way of knowing when the other party has finished unless the protocol is set up and observed. This is because every character you type goes to the other side, and there is nothing which restricts a message to one line.

In UNIX every device is a file and it has permissions associated with it like any other file. We will see more about this in the next unit on system administration. Here it is sufficient to understand that normally when you login, your terminal device file has permissions such that all users can write to your terminal. If you turn off write permission for other users then, nobody will be able to write to your terminal using the write command, and consequently will not be able to disturb you while you are working. You can do this using the chmod command, but you would then need to know more about device files, like what the file name is and where it is located.
A simple way to turn off write permission for other users is to say mesg n

3.2 OFF-LINE COMMUNICATION

Let us now look at two commands which allow UNIX users to communicate in off-line mode. This means that the users will not be able to talk or converse, but a message sent by one will be sent to the other, and the recipient can then decide whether he wants to look at it and maybe even act on it if needed.
You all must have heard about electronic mail or e-mail, as it is usually called. In fact, many computer professionals now refer to e-mail as mail and to conventional mail as paper mail.

Today if you are onto some international network like the Internet, you can send mail to far off places like say, the United States, and if your partner wants to respond you could have the reply the next day.

We will confine ourselves to sending electronic mail to other users on the same machine.
There are advantages and disadvantages to using mail, as opposed to using write. The problem with mail is that you cannot carry on a conversation with your counterpart at the other end. So if there is some small, urgent message to be sent and which the other party must see at once, you need to use write.

But this situation is not common as compared to the times when you just want to send a message to the other party. You either do not need a reply or you can wait for one.
Sometimes your message is a long one, much longer than can conveniently be digested during a conversation with write. These are the times when mail is very useful. Then again with write the other user has to be logged in at that moment if you want to communicate, while with mail you can send a message to any user who is registered on that system, irrespective of whether he is logged in at that time or not.

A message sent to a user by using mail gets stored in a mailbox allocated to that user, and stored somewhere in the file system. The user gets this message the next time he logs in
“You have mail”. As long as there is some mail in your mailbox you will get this message every time you login. You should therefore look at your mai1 and dispose it off while it is recent. It is not obligatory to look at your mail and UNIX does not compel you to do so. If you neglect to read your mail it might go stale. That is a good reason to inspect your mail regularly. You can delete all or part of your mail without reading it, if you wish.

You should use mail to communicate with people in your installation and elsewhere. It has many advantages over a phone call, paper mail or casual conversation. Since it amounts to writing down what you want to say, it ensures that you will be systematic, unlike a verbal exchange. Paper mail can be unreliable. It can get misplaced or might reach after a long-time, and the recipient has to be present at his usual place of work to be able to see it. A phone call might not find the other party on the line or available at that time.
Electronic mail has none of these difficulties. The message reaches quickly and will not get misplaced. The other party does not have to be there at that time. He will get the message whenever he decides to look at his mailbox. The message need not be brief unlike the situation with a telephone answering machine. So you can see that the person need not be at his usual place of work at all. He can look up his mail from anywhere in the world if he can connect to his machine. That is why electronic mail is so popular now and it will soon' become commonplace.
A disadvantage of electronic mail as described here is the lack of privacy. On the system, the super user can always look at anybody else's mail and you might not feel comfortable with this. You could encrypt your mail after you save it, but the super user can, look at it before you do so. One possible solution is to use a public key cryptography mechanism and interface it to a mail program. Such schemes are already available in the public domain.

Let us now look at another communication command available in UNIX. This is again, like wall, a one to many or broadcast kind of command, though not quite, because the recipients have the choice of deciding whether to look at the message or not. The command is called news and is typically used to convey information about the local system or installation.

4.0 CONCLUSION

You would have learned about Communicate on-line with other users on your machine using write and Communicate off-line with other users with the help of mail and news.

5.0 SUMMARY

You have learned about online communication and offline communication.

ACTIVITY B

1. What are the advantages and disadvantages of using a mail command in user to user communication?

6.0 TUTOR MARKED ASSIGNMENT

1. What are the advantages and disadvantages of using a write command in user to user communication?

7.0 REFERENCES/FUTHER READINGS

1. Introduction to Computer Networks by William Stallings 9th Edition, Publisher – Prentice Hall, ISBN: 0131392050.
2. Understanding and Designing Computer Networks by Graham King. Publisher – Edward Amold, ISBN: 0-340-61419-6.
3. Network Design by Mani Subramanian. Publisher-Addison-Wesley, ISBN: 0-201-35742-9.
4. Practical UNIX & Internet Security, by Simson Garfinkel and Gene Spafford, 2nd Edition, O’Reilly, 1996.

UNIT THREE

NETWORK ARCHITECTURE

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
3.1 Definition
3.2 OSI Model
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about the definition of network architecture and the OSI Model.

2.0 OBJECTIVES

- Define network architecture
- Explain the OSI Model

3.0 MAIN CONTENTS

3.1 DEFINITION

Network architecture is the design of a communications network. It is a framework for the specification of a network's physical components and their functional organization and configuration, its operational principles and procedures, as well as data formats used in its operation.

In telecommunication, the specification of a network architecture may also include a detailed description of products and services delivered via a communications network, as well as detailed rate and billing structures under which services are compensated.

The network architecture of the Internet is predominantly expressed by its use of the Internet Protocol Suite, rather than a specific model for interconnecting networks or nodes in the network, or the usage of specific types of hardware links.

3.2 OSI MODEL

The Open Systems Interconnection model (OSI model) is a product of the Open Systems Interconnection effort at the International Organization for Standardization. It is a way of sub-dividing a communications system into smaller parts called layers. A layer is a collection of similar functions that provide services to the layer above it and receives services from the layer below it. On each layer, an instance provides services to the instances at the layer above and requests service from the layer below. Figure 2.2 below shows the layers of the OSI Reference Model:

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

Figure 2.2: The seven layers of the OSI Reference Model.

1. The physical layer is concerned with the physical wiring used to connect different systems together on the network. Examples include the serial and parallel cables mentioned earlier, Ethernet and Token Ring cabling, telephone cables, and even the specific connectors and jacks used by these cabling systems. Without strictly standardized definitions for the cabling and connectors, vendors might not implement them in such a way that they would function with other vendor's implementations, which in turn would make it impossible for any communications to occur whatsoever. Each of these wiring systems therefore follows very strict standards, ensuring that the systems will at least be able to communicate without having to worry about the underlying cabling.

2. The data-link layer is used to define how information is transmitted across the physical layer, and is responsible for making sure that the physical layer is functioning properly. Some networks - such as the public telephone system, AM/FM radio and television - use analog sine-waves to transmit information, while most computer networks use digital "square" pulses to achieve this objective. If there are any problems with transmitting the information on the physical cabling (perhaps due to a damaged wire or circuit), then this layer must deal with those errors, either attempting to retransmit the information or reporting the failure to the network layer.

3. The network layer is used to identify the addresses of systems on the network, and for the actual transmission of data between the systems. The network layer must be aware of the physical nature of the network, and package the information in such a way that the data-link layer can deliver it to the physical layer. For example, if a telephone line is the physical layer, then the network layer must package the information in such a way that the data-link layer can transmit it over an analog circuit. Likewise, if the physical layer is a digital Ethernet LAN, then the network layer must encapsulate the information into digital signals appropriate for Ethernet, and then pass it to the data link layer for transmission.

   On many networks, the network layer does not provide any integrity checking. It simply provides the packaging and delivery services, assuming that if the data-link layer did not report any error then the networks are operational. Broadcast television and radio work in this manner, assuming that if they can transmit a signal, then it can also be received. Many digital networking technologies also take this approach, leaving it up the higher level protocols to provide delivery tracking and reliability guarantees.

4. The transport layer provides the reliability services lacking from the network layer, although only for basic transmission services, and not for any application- or service-specific functions. The transport layer is responsible for verifying that the network layer is operating efficiently, and if not, then to either request a retransmission or to return an error to the layer above it. Since higher-level services have to go through the transport layer, all transport services are guaranteed when this layer is designed into the network software and used. Not all systems mandate that the transport layer provide reliability; indeed many networks provide unreliable transport layers for non-essential services such as broadcast messages.

5. The session layer is responsible for establishing "connections" between systems, applications or users. The session layer may receive this request from any higher layer, and

then will negotiate a connection using the lower layers. Once a connection is established, the session layer simply provides an interface to the network for the higher layers to communicate with. Once the higher layers are finished, the session layer is responsible for destroying the connection as well.

6. The presentation layer provides a consistent set of interfaces for applications and services to utilize when establishing connections through the session layer. Although these interfaces could also exist at the session layer, that would burden it unnecessarily. It is better to have the session layer only manage sessions and not worry about verifying data or providing other extended services. An example of a service provided by the presentation layer is data- compression, allowing applications to take advantage of the performance gains that compression provides without forcing the applications to develop these services themselves.

7. Finally, the application layer provides the network's interface to end-user applications and services such as printing or file-sharing. This layer also provides some management services to ensure that the interfaces are being addressed and used correctly.

4.0 CONCLUSION

You would have learned about the definition of network architecture and the OSI Model.

5.0 SUMMARY

You have learned about the definition of network architecture, as well as the functions of the 7 layers of the OSI model.

ACTIVITY B

1.0 Briefly state the function of each layer of the OSI Model

6.0 TUTOR MARKED ASSIGNMENT

1. What is network architecture?

7.0 REFERENCES/FUTHER READINGS

1. Introduction to Computer Networks by William Stallings 9th Edition, Publisher – Prentice Hall, ISBN: 0131392050.
2. Understanding and Designing Computer Networks by Graham King. Publisher – Edward Amold, ISBN: 0-340-61419-6.
3. Network Design by Mani Subramanian. Publisher-Addison-Wesley, ISBN: 0-201-35742-9.
4. Practical UNIX & Internet Security, by Simson Garfinkel and Gene Spafford, 2nd Edition, O’Reilly, 1996.

5. A Beginner's Guide to Network Security Cisco Systems Copyright © 2001 Cisco Systems

6. Introduction to Computer Networking by Jean-Yves Le Boudec .

UNIT FOUR

NETWORKING PROTOCOLS

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
3.1 Definition
3.2 Network Protocols

4.0 Conclusion

5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about what networking protocols are and their roles.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Define network protocols
- Describe various network protocols and their functions
- Identify the functions of protocols and protocol stacks.
- Map specific protocols to the appropriate OSI level.
- Define the protocols that make up the NetWare protocol suite.
- Relate the NetWare protocols to the OSI reference model.

3.0 MAIN CONTENTS

3.1 DEFINITION

Network protocols define the rules that govern the communications between two computers connected to the network.

A protocol specification consists of the syntax, which defines the kinds and formats of the messages exchanged, and the semantic, which specifies the action taken by each entity when specific events occur.

Example: HTTP protocol for communication between web browsers and servers.

The Roles of network protocols include:

1. Addressing and routing of messages
2. Error detection and recovery
3. Sequence and flow controls.

INTRODUCTION TO PROTOCOLS

This unit offers an introduction to protocols and their function in a networking environment. It explains the roles of protocols in network communications and describes how different protocols work at different OSI levels.

- The Function of Protocols

Protocols are rules and procedures for communicating. The term "protocol" is used in a variety of contexts. For example, diplomats from one country adhere to rules of protocol designed to help them interact smoothly with diplomats from other countries. Rules of protocol apply in the same way in the computer environment. When several computers are networked, the rules and technical procedures governing their communication and interaction are called protocols.

Keep three points in mind when you think about protocols in a network environment:

- There are many protocols. While each protocol facilitates basic communications, each has different purposes and accomplishes different tasks. Each protocol has its own advantages and restrictions.
- Some protocols work only at particular OSI layers. The layer at which a protocol works describes its function. For example, a protocol that works at the physical layer ensures that the data packet passes through the network interface card (NIC) and out onto the network cable.
- Protocols can also work together in a protocol stack or suite. Just as a network incorporates functions at every layer of the OSI reference model, different protocols also work together at different levels in a single protocol stack. The levels in the protocol stack "map," or correspond, to the layers of the OSI reference model. For instance, the TCP/IP protocol's application layer maps to the OSI reference model's presentation layer. Taken together, the protocols describe the entire stack's functions and capabilities.

- How Protocols Work

The entire technical operation by which data is transmitted over the network has to be broken down into discrete, systematic steps. At each step, certain actions take place that cannot take place at any other step. Each step includes its own rules and procedures, or protocol.

The protocol steps must be carried out in a consistent order that is the same on every computer in the network. In the sending computer, these steps must be executed from the top down. In the receiving computer, these steps must be carried out from the bottom up.

The Sending Computer

Protocols at the sending computer:

1. Break the data into smaller sections, called packets, which the protocol can handle.
2. Add addressing information to the packets so that the destination computer on the network can determine that the data belongs to it.
3. Prepare the data for transmission through the NIC and out onto the network cable.

The Receiving Computer

Protocols at the receiving computer carry out the same series of steps in reverse order. They:

1. Take the data packets off the cable.
2. Bring the data packets into the computer through the NIC.
3. Strip the data packets of all the transmitting information that was added by the sending computer.
4. Copy the data from the packets to a buffer for reassembly.
5. Pass the reassembled data to the application in a usable form.

Both sending and receiving computers need to perform each step in the same way so that the data will have the same structure when it is received as it did when it was sent.

For example, two different protocols might each break data into packets and add on various sequencing, timing, and error-checking information, but each will do it differently. Therefore, a computer using one of these protocols will not be able to communicate successfully with a computer that is using the other protocol.

- Routable Protocols

Until the mid-1980s, most local area networks (LANs) were isolated. A LAN served a single department or company and was rarely connected to any larger environments. As LAN technology matured, however, and the data communication needs of businesses expanded, LANs evolved, becoming components in larger data communication networks in which LANs talked to each other.

Data that is sent from one LAN to another along any of several available paths is said to be routed. The protocols that support multipath LAN-to-LAN communications are known as routable protocols. Because routable protocols can be used to tie several LANs together and create new wide-area environments, they are becoming increasingly important.

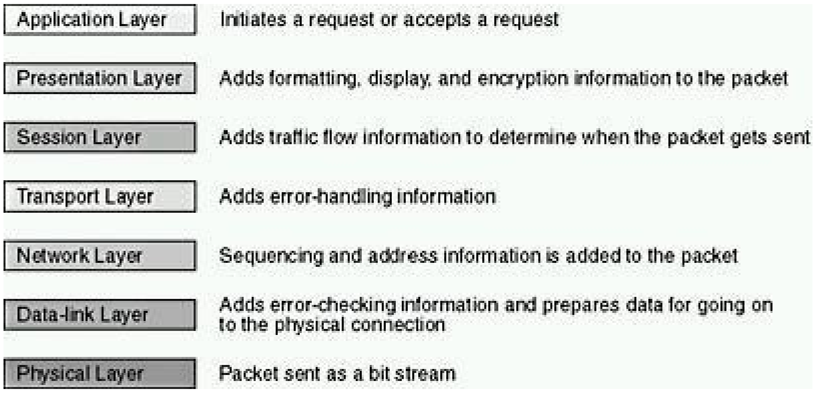
Protocols in a Layered Architecture

In a network, several protocols have to work together. By working together, they ensure that the data is properly prepared, transferred to the right destination, received, and acted upon.

The work of the various protocols must be coordinated so that no conflicts or incomplete operations take place. The results of this coordination effort are known as layering.

- Protocol Stacks

A protocol stack is a combination of protocols. Each layer of the stack specifies a different protocol for handling a function or subsystem of the communication process. Each layer has its own set of rules. In Module 3, Unit 3, “Network Architecture", we discussed the OSI reference model. Figure 2.3 shows the OSI reference model and the rules associated with each layer. The protocols define the rules for each layer in the OSI reference model.



Application Layer	Initiates a request or accepts a request
Presentation Layer	Adds formatting, display, and encryption information to the packet
Session Layer	Adds traffic flow information to determine when the packet gets sent
Transport Layer	Adds error-handling information
Network Layer	Sequencing and address information is added to the packet
Data-link Layer	Adds error-checking information and prepares data for going on to the physical connection
Physical Layer	Packet sent as a bit stream

Figure 2.3: The OSI reference model showing the layers of protocols

The lower layers in the OSI reference model specify how manufacturers can make their equipment connect to equipment from other manufacturers, for example, by using NICs from several manufacturers on the same LAN. As long as they operate with the same protocols, they are able to send and receive data from each other. The upper layers specify rules for conducting communications sessions (the time during which two computers maintain a connection) and the interpretation of applications. The higher they are in the stack, the more sophisticated the tasks and their associated protocols become.

- The Binding Process

The binding process—the process by which protocols become connected to each other and the NIC—allows a great deal of flexibility in setting up a network. Protocols and NICs can be mixed and matched on an as-needed basis. For example, two protocol stacks, such as Internetwork Packet Exchange and Sequenced Packet Exchange (IPX/SPX), NetWare Protocols, and Transmission

Control Protocol/Internet Protocol (TCP/IP). TCP/IP can be bound to one NIC. If there is more than one NIC in the computer, one protocol stack can be bound to either or both NICs.

- Standard Stacks

The computer industry has designated several kinds of stacks as standard protocol models. Hardware and software manufacturers can develop their products to meet any one or a combination of these protocols. The most important models include:

- The ISO/OSI protocol suite.
- The IBM Systems Network Architecture (SNA).
- Digital DECnet.
- Novell NetWare.
- Apple's AppleTalk.
- The Internet protocol suite, TCP/IP.

Protocols exist at each layer of these stacks, performing the tasks specified by that layer. However, the communication tasks that networks need to perform are grouped into one of three protocol types. Each type is comprised of one or more layers of the OSI. As shown in Figure 6.2, these three protocol types map roughly to layers of the OSI reference model (application, transport, and network).

NOTE: Many protocols were written long before the OSI reference model came into common use. Thus, it is not uncommon to find protocol stacks that do not map directly to the OSI model.

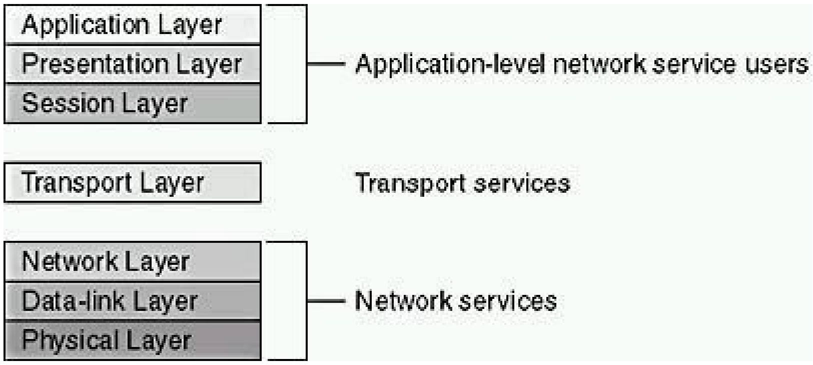


Figure 2.4 Communication tasks within the OSI reference model

Application Protocols

Application protocols work at the uppermost layer of the OSI reference model. They provide application-to-application interaction and data exchange. Popular application protocols are shown in Table 2.1.

Table 2.1: Popular Application Protocols

Protocol	Description
APPC (Advanced Program-to-Program Communication)	IBM's peer-to-peer SNA protocol, mostly used on AS/400 computers. APPC is defined as an applica- tion protocol, because it works in the presentation layer of the OSI reference model. However, it is also considered a transport protocol because APPC uses the LU 6.2 protocol that works in both the transport and session layers of the OSI reference model.
FTAM (File Transfer Access and Management)	An OSI file access protocol.
X.400	A CCITT protocol for international e-mail transmissions.
X.500	A CCITT protocol for file and directory services across several systems.
SMTP (Simple Mail Transfer Protocol)	An Internet protocol for transferring e-mail.
FTP (File Transfer Protocol)	An Internet file transfer protocol.
SNMP (Simple Network Management Protocol)	An Internet protocol for monitoring networks and network components.
Telnet	An Internet protocol for logging on to remote hosts and processing data locally.
Microsoft SMBs (Server Message Blocks) and	A client/server, request response protocol.

client shells or redirectors	
NCP (Novell NetWare Core Protocol) and Novell client shells or redirectors	A set of service protocols.
AppleTalk and AppleShare	Apple's networking protocol suite.
AFP (AppleTalk filing Protocol)	Apple's protocol for remote file access.
DAP (Data Access Protocol)	A DECnet file access protocol.

Transport Protocols

Transport protocols facilitate communication sessions between computers and ensure that data is able to move reliably between computers. Popular transport protocols are shown in Table 2.2.

Table 2.2: Popular Transport Protocols

Protocol	Description
TCP	The TCP/IP protocol for guaranteed delivery of sequenced data.
SPX	Part of Novell's IPX/SPX protocol suite for sequenced data.
NWLink	The Microsoft implementation of the IPX/SPX protocol.
NetBEUI (NetBIOS extended user interface)	Establishes communication sessions between computers (NetBIOS) and provides the underlying data transport services (NetBEUI).
ATP (AppleTalk Transaction Protocol) and NBP (Name Binding Protocol)	Apple's communication-session and data-transport protocols.

Network Protocols

Network protocols provide what are called "link services." These protocols handle addressing and routing information, error checking, and retransmission requests. Network protocols also define rules for communicating in a particular networking environment such as Ethernet or Token Ring. Popular network protocols are shown in table 2.3.

Table 2.3: Popular Network Protocols

Protocol	Description
IP	The TCP/IP protocol for packet-forwarding routing. IPX
	NetWare's protocol for packet forwarding and routing.
NWLink	The Microsoft implementation of the IPX/SPX protocol.
NetBEUI	A transport protocol that provides data-transport services for NetBIOS sessions and applications.
DDP (Datagram Delivery Protocol)	An AppleTalk data-transport protocol.

3.2 NETWORK PROTOCOLS

Internet Protocol (IP)

The IP protocol provides two main functionalities:

- Decomposition of the initial information flow into packets of standardized size, and reassembling at the destination.
- Routing of a packet through successive networks, from the source machine to the destination identified by its IP address.

Transmitted packets are not guaranteed to be delivered (datagram protocol).
The IP protocol does not request for connection (connectionless) before sending data and does not make any error detection.

Functions

- Decompose the initial data (to be sent) into datagrams.
- Each datagram will have a header including, the IP address and the port number of the destination.
- Datagrams are then sent to selected gateways, e.g. IP routers, connected at the same time to the local network and to an IP service provider network. Datagrams are transferred from gateways to gateways until they arrived at their final destination.

Transmission Control Protocol (TCP)

TCP provides by using IP packets as a basic service that does guarantee safe delivery:

- Error detection
- Safe data transmission
- Assurance that data are received in the correct order

Before sending data, TCP requires that the computers communicating establish a connection (connection-oriented protocol).

TCP provides support for sending and receiving arbitrary amounts of data as one big stream of byte data (IP is limited to 64Kb).

TCP does so by breaking up the data stream into separate IP packets.
Packets are numbered, and reassembled on arrival; using sequence and sequence acknowledge numbers.

TCP also improves the capability of IP by specifying port numbers.

There are 65,536 different TCP ports (sockets) through which every TCP/IP machine can talk.

TCP/IP Protocol Suite

First and foremost, TCP/IP is not a single protocol, but a term used to define the family of protocols in use on the Internet. The individual protocols in the TCP/IP suite are very specific in function, ranging from the mundane task of providing a simple transport service, to more esoteric function of transmitting the graphical pages found on the World Wide Web.

NetWare only has a few protocols that are "standard". Vendors that develop fax or e-mail servers that run over IPX develop their own proprietary protocols for these services. However, standard TCP/IP protocols exist for a wide variety of services. There are protocols that are used for sharing files and printers (e.g. NetWare), protocols for publishing HTML pages over the World Wide Web, protocols for sending and receiving electronic mail over the Internet, and many others. By having standard protocols for a variety of application-specific services, TCP/ IP is much more flexible than other protocol families like IPX, but it is also quite a bit larger, and many times it is also less efficient.

TCP/IP was not designed for the small networks that IPX was designed for, but instead was designed for world-wide networks of a tremendous scale and variety. For example, IPX networks use SAP tables to record and publish ever-changing lists of servers and resources on the local network. This works extremely well on a small network with no more than a few hundred devices. However, it would not work on the global Internet with its millions of nodes. The SAP status messages needed to keep the network working would flood the Internet to a point where it became unusable for anything else.

Another important aspect of TCP/IP is the issue of "openness". While NetWare and other LAN products (including Microsoft's Windows NT and Apple's offerings) all use proprietary protocols for sharing resources, the TCP/IP family of protocols are in the public domain and usable by anyone. Proposals for new specifications can be offered by anybody and the technology adoption process is executed in full sight. Thus, many companies already offer integrated TCP/IP protocols and services in their products.

These three elements (scalability, flexibility and openness) make TCP/IP an attractive choice for users in mixed environments. They can run the same protocols and services on almost all of their host and client systems. For this reason, many customers have made TCP/IP a check- off item for network purchases, and are deploying TCP/IP-based applications and services across their internal networks, and are also using it for their external connections. Figure 2.5 below shows the major protocols in the TCP/IP suite.

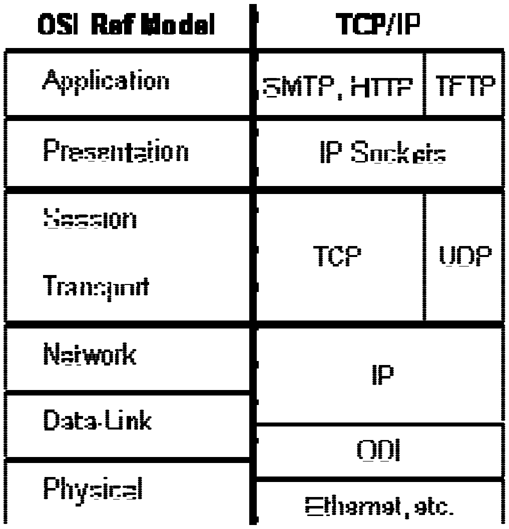


Figure 2.5: The TCP/IP protocol suites.

Internet Protocol (IP) and Address Resolution Protocol (ARP)

IP, or the Internet Protocol, is the basic building block for all TCP/IP traffic, and works at the network-layer of the OSI reference model. It is responsible for tracking the network addresses of devices, and for determining how packets are delivered to devices on other networks.

Unlike IPX, which uses physical hardware addresses as the node address, IP uses manually-defined 32-bit addresses for each node on the global Internet. Part of the address defines the network that the node is on, and the rest of the address defines the node itself, as illustrated by Figure 2.6 below.

If the destination system is on a different network, then IP is used to send the packet to a router that can forward it on to the destination network.

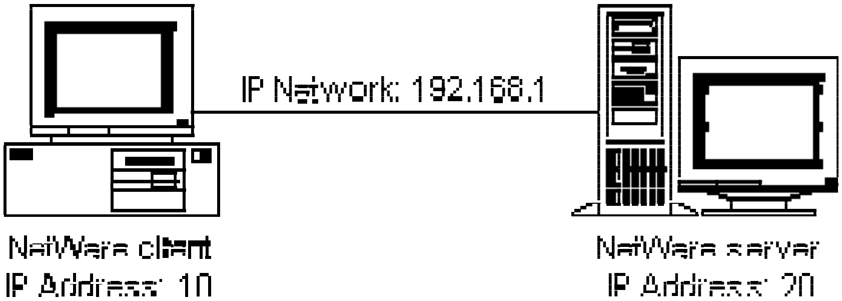


Fig 2.6: IP and ARP. IP uses a two-part address consisting of a network address and workstation address. These addresses are manually assigned by the local system administrators, and must be unique across the entire Internet.

Packet delivery is handled at the physical layer, so IP must convert the Internet addresses into network addresses used by the local medium, which are typically the physical Ethernet addresses of the network adapters in use. IP uses the Address Resolution Protocol (ARP) to build this map in memory, as illustrated in figure 2.7 below.

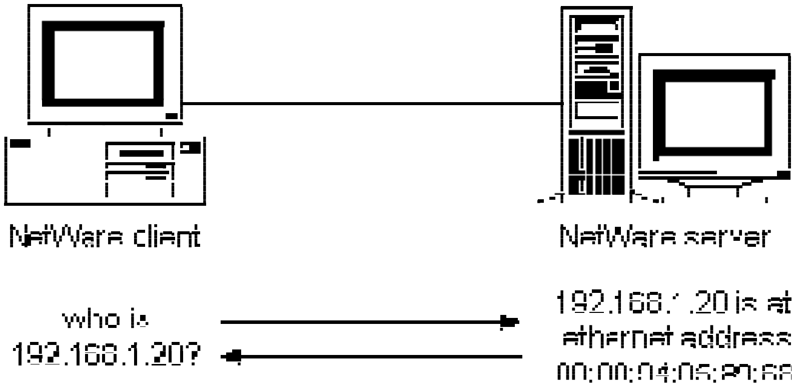


Fig. 2.7: IP uses ARP to build the map in memory. Because IP addresses do not correspond to hardware, the Address Resolution Protocol is used to determine who should receive which packets.

NOTE: IP does not guarantee delivery or provide error-correction services. These functions are provided by TCP.

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)

IP simply provides a transport service, shuttling packets between hosts and networks. The protocols that provide the transport-layer services for IP are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). Very few applications speak to IP directly.

TCP provides error-correction through the use of a connection-oriented transaction. A "start" packet is built and sent to the destination node (via IP), and when an "okay I'm ready" packet comes back, a monitored conversation between the hosts and/or applications begins. If a packet is lost or corrupted, TCP resends the data. The size, timeout interval, and other critical factors are determined by judging the strength of the media that the node is connected to, the number of retries necessary to successfully deliver data, and other aspects of the connection. Because of this, TCP can dynamically change the nature of a session so that it will work even if the state of the physical link changes. This process is illustrated in figure 2.8 below:

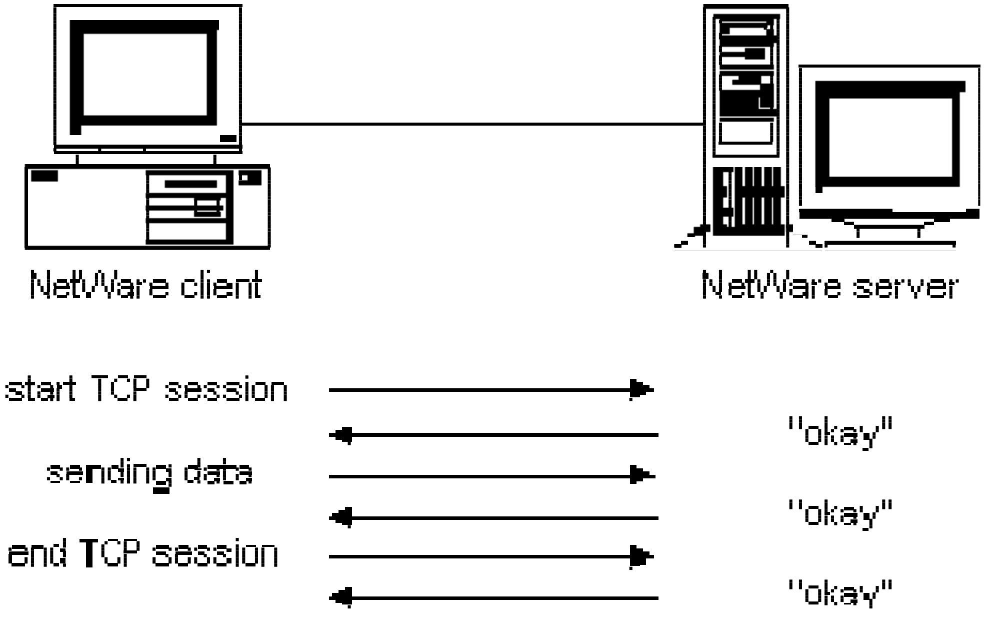


Fig. 2.8: TCP uses a "handshake" format to synchronize, monitor, and close sessions, providing very high reliability.

Applications such as the Internet's Simple Message Transport Protocol (SMTP) and HyperText Transfer Protocol (HTTP) both require the reliable connection services that TCP provides.

Otherwise, mail messages sent over SMTP could get lost, or the graphic images and HTML documents sent over HTTP might get corrupted. TCP provides the reliability services so that the applications do not have to provide this within their internal application code.

UDP on the other hand simply sends data, and makes no pretense towards guaranteed delivery as illustrated in figure 2.9 below.

However, like TCP, UDP does make decisions about packet sizes based on the strength of the underlying media, and passes the fully contained parcel to IP for delivery.

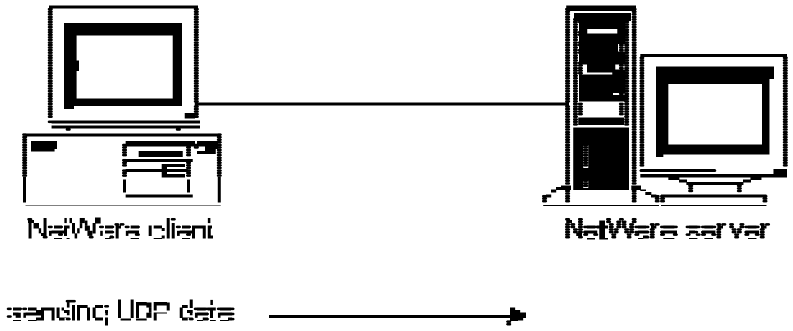


Fig. 2.9: NetWare client sending UDP data to the server

User Datagram Protocol (UDP)

Datagram protocol also built on top of IP. It has the same packet-size limit (64Kb) as IP, but allows for port number specification. It also provides 65,536 different ports. Hence, every machine has two sets of 65,536 ports: one for TCP and the other for UDP. It makes use of connectionless protocol, without any error detection facility. It provides only support for data transmission from one end to the other, without any further verification. The main interest of UDP is that since it does not make further verification, it is very fast and useful for sending small size data in a repetitive way such as time information.

- NetWare Protocols

Introduction to NetWare Protocols

Like TCP/IP, Novell provides a suite of protocols developed specifically for NetWare. The five main protocols used by NetWare are:

- Media Access Protocol.
- Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX).
- Routing Information Protocol (RIP).

- Service Advertising Protocol (SAP).
- NetWare Core Protocol (NCP).

Because these protocols were defined well before the finalization of the OSI reference model, they do not exactly match OSI. Figure 2.10 provides mapping of the NetWare protocols to the OSI reference model. In actuality, no direct correlation to the layer boundaries of the two architectures exists. These protocols follow an enveloping pattern. More specifically, the upper-lever protocols (NCP, SAP, and RIP) are enveloped by IPX/SPX. A Media Access Protocol header and trailer then envelop IPX/SPX.

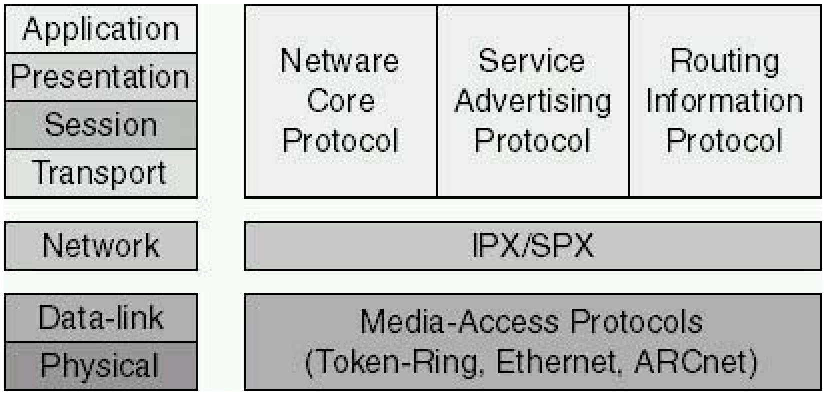


Figure 2.10 Comparing NetWare and OSI reference models

Media Access Protocols

Media Access Protocols define the addressing that distinguishes each node on a NetWare network. The addressing is implemented on the hardware or NIC. The most common implementations are:

- 802.5 Token Ring.
- 802.3 Ethernet.
- Ethernet 2.0.

This protocol is responsible for placing the header on the packet. Each header includes the source and destination code. After the packet has been transmitted and is on the media, each network card checks the address; if their address matches the destination address on the packet, or, if the packet is a broadcast message, the NIC copies the packet and sends it up the protocol stack.

In addition to addressing, this protocol provides bit-level error checking in the form of a cyclical redundancy check (CRC). With the CRC appended to the packet, it is virtually certain that all the packets will be free of corruption.

NOTE : CRC error checking uses a complex calculation to generate a number based on the data transmitted. The sending device performs the calculation before transmission and includes it in the packet that it sends to the receiving device. The receiving device repeats the same calculation after transmission. If both devices obtain the same result, it is assumed that the transmission was error-free. The procedure is known as a redundancy check because each transmission includes not only data but extra (redundant) error-checking values.

Internetwork Packet Exchange and Sequenced Packet Exchange (IPX/SPX)

Internetwork Packet Exchange (IPX) defines the addressing schemes used on a NetWare network, and Sequenced Packet Exchange (SPX) provides security and reliability to the IPX protocol. IPX is a datagram-based, connectionless, unreliable, network-layer protocol that is equivalent to the IP. It does not require an acknowledgement for each packet sent. Any acknowledgement control or connection control must be provided by the protocols above IPX. SPX provides connection-oriented, reliable servers at the transport layer.

Using the Xerox Network System (XNS) Internet Datagram Protocol, Novell adopted IPX protocol. IPX defines two kinds of addressing:

- Internetwork addressing The address of a segment on the network, identified by the network number assigned during installation.
- Intranode addressing The address of a process within a node that is identified by a socket number.

IPX protocols are used only on networks with NetWare servers and are often installed along with another protocol suite such as TCP/IP. Even NetWare is moving toward using TCP/IP as a standard.

Routing Information Protocol (RIP)

Facilitating the exchange of routing information on a NetWare network, RIP, like IPX, was developed from XNS. However, in RIP, an extra field of data was added to the packet to improve the decision criteria for selecting the fastest route to a destination. The broadcast of an RIP packet allows several things to occur:

- Workstations can locate the fastest route to a network number.
- Routers can request routing information from other routers to update their own internal tables.
- Routers can respond to route requests from workstations and other routers.
- Routers can make sure that all other routers are aware of the internetwork configuration.
- Routers can detect a change in the internetwork configuration.

Service Advertising Protocol (SAP)

The Service Advertising Protocol (SAP) allows service-providing nodes (including file servers, printer servers, gateway servers, and application servers) to advertise their services and addresses. Clients on the network are able to obtain the internetwork address of any servers they can access. With SAP, the adding and removing of services on the network becomes dynamic. By default, a SAP server broadcasts its presence every 60 seconds. A SAP packet contains:

- Operating Information Specifies the operation that the packet is performing.
- Service type Specifies the type of service offered by the server.
- Server name Specifies the name of the broadcasting server.
- Network address Specifies the network number of the broadcasting server.
- Node address Specifies the node number of the broadcasting server.
- Socket address Specifies the socket number of the broadcasting server.
- Total hops to server Specifies the number of hops to the broadcasting server.
- Operation field Specifies the type of request.
- Additional information One or more sets of fields can follow the operation field which contains more information about one or more servers.

NetWare Core Protocol (NCP)

The NetWare Core Protocol (NCP) defines the connection control and service request encoding that make it possible for clients and servers to interact. This is the protocol that provides transport and session services. NetWare security is also provided within this protocol.

Internet Application Protocols

On top of TCP/IP, several services have been developed in order to homogenize applications of same nature:

FTP (File Transfer Protocol) allows the transfer of collection of files between two machines connected to the Internet.

Telnet (Terminal Protocol) allows a user to connect to a remote host in terminal mode.

NNTP (Network News Transfer Protocol) allows the constitution of communication groups (newsgroups) organized around specific topics.

SMTP (Simple Mail Transfer Protocol) defines a basic service for electronic mails.

SNMP (Simple Network Management Protocol) allows the management of the network.

4.0 CONCLUSION

You would have learned what networking protocols are and their roles.

5.0 SUMMARY

What you have learned in this unit borders the meaning of network protocols and the various network protocols and their functions.

ACTIVITY B

1. List 10 network protocols and their functions

6.0 TUTOR MARKED ASSIGNMENT

1. Define network protocol

7.0 REFERENCES/FUTHER READINGS

1. Introduction to Computer Networks by William Stallings 9th Edition, Publisher – Prentice Hall, ISBN: 0131392050.
2. Understanding and Designing Computer Networks by Graham King. Publisher – Edward Amold, ISBN: 0-340-61419-6.
3. Network Design by Mani Subramanian. Publisher-Addison-Wesley, ISBN: 0-201-35742-9.
4. Practical UNIX & Internet Security, by Simson Garfinkel and Gene Spafford, 2nd Edition, O’Reilly, 1996.
5. Internet Communication Protocols by W. Richard Stevens. ISBN: 0-201633469.

MODULE THREE

LINUX OPERATING SYSTEM

Unit 1: Introduction to Linux Operating System

Unit 2: Linux commands and Utilities

Unit 3: Linux Utilities and Editor

Unit 4: UNIX System Administration

UNIT ONE

INTRODUCTION TO LINUX OPERATING SYSTEM

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
- 3.1 History
- 3.2 The User Interface
- 3.3 Properties of Linux

4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

We will start with an overview of how Linux became the operating system it is today. We will discuss past and future development and take a closer look at the advantages and disadvantages of this system. We will talk about distributions, about Open Source in general and try to explain a little about GNU.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Explain the history of Linux Operating System
- Describe the User Interface
- Explain the properties of Linux

3.0 MAIN CONTENTS

3.1 HISTORY

In order to understand the popularity of Linux, we need to travel back in time, about 30 years ago...
Imagine computers as big as houses. While the sizes of those computers posed substantial problems, there was one thing that made this even worse: every computer had a different operating system.

Software was always customized to serve a specific purpose, and software for one given system didn't run on another system. Being able to work with one system didn't automatically mean that you could work with another. It was difficult, both for the users and the system administrators.

Computers were extremely expensive then, and sacrifices had to be made even after the original purchase just to get the users to understand how they worked. The total cost per unit of computing power was enormous. Technologically the world was not quite that advanced, so they had to live with the size for another decade.

Fig 3.1: A modern Unix desktop environment

In 1969, a team of developers in the Bell Labs laboratories started working on a solution for the software problem, to address these compatibility issues. They developed a new operating system, which was

1. Simple and elegant.
2. Written in the C programming language instead of in assembly code.

3. Able to recycle code.

The Bell Labs developers named their project "UNIX." The code recycling features were very important. Until then, all commercially available computer systems were written in a code specifically developed for one system. UNIX on the other hand needed only a small piece of that special code, which is now commonly named the kernel. This kernel is the only piece of code that needs to be adapted for every specific system and forms the base of the UNIX system. The operating system and all other functions were built around this kernel and written in a higher programming language, C.

Ken Thompson then teamed up with Dennis Ritchie, the author of the first C compiler in 1973. They rewrote the UNIX kernel in C - this was a big step forwards in terms of the system's portability - and released the Fifth Edition of UNIX to universities in 1974. The Seventh Edition, released in 1978, marked a split in UNIX development into two main branches: SYSV (System 5) and BSD (Berkeley Software Distribution). BSD arose from the University of California at Berkeley where Ken Thompson spent a sabbatical year. Its development was continued by students at Berkeley and other research institutions. SYSV was developed by AT&T and other commercial companies. UNIX flavours based on SYSV have traditionally been more conservative, but better supported than BSD-based flavours.

The latest incarnations of SYSV (SVR4 or System 5 Release 4) and BSD Unix are actually very similar. Some minor differences are to be found in file system structure, system utility names and options and system call libraries as shown in Fig 3.2.

Feature	Typical SYSV	Typical BSD
kernel name	/unix	/vmunix
boot init	/etc/rc.d directories	/etc/rc.* files
mounted FS	/etc/mnttab	/etc/mtab
default shell	sh, ksh	csh, tcsh
FS block size	512 bytes->2K	4K->8K
print subsystem	lp, lpstat, cancel	lpr, lpq, lprm
echo command (no new line)	echo "\c"	echo -n
ps command	ps -fae	ps -aux
multiple wait syscalls	poll	select
memory access syscalls	memset, memcpy	bzero, bcopy

Fig. 3.2: Differences between SYSV and BSD

Linux is a free open source UNIX OS for PCs that was originally developed in 1991 by Linus Torvalds, a Finnish undergraduate student. Linux is neither pure SYSV or pure BSD. Instead, incorporates some features from each (e.g. SYSV-style startup files but BSD-style file system layout) and aims to conform with a set of IEEE standards called POSIX (Portable Operating System Interface).

This language was especially developed for creating the UNIX system. Using this new technique, it was much easier to develop an operating system that could run on many different types of hardware. UNIX did a great deal to help users become compatible with different systems.

Throughout the next couple of decades the development of UNIX continued. More things became possible to do and more hardware and software vendors added support for UNIX to their products. UNIX was initially found only in very large environments with mainframes and minicomputers. You had to work at a university, for the government or for large financial corporations in order to get your hands on a UNIX system.

Smaller computers were being developed, and by the end of the 80's, many people had home computers and there were several versions of UNIX available for the PC architecture, but none of them were truly free and more important: they were very slow, so most people ran MS DOS or Windows 3.1 on their home PCs.

3.2 THE USER INTERFACE

Companies such as RedHat, SuSE and Mandriva have sprung up, providing packaged Linux distributions suitable for mass consumption. They integrated a great deal of graphical user interfaces (GUIs), developed by the community, in order to ease management of programs and services.

As a Linux user today you have all the means of getting to know your system inside out, but it is no longer necessary to have that knowledge in order to make the system comply with your requests.
Nowadays you can log in graphically and start all required applications without even having to type a single character, while you still have the ability to access the core of the system if needed.

Because of its structure, Linux allows a user to grow into the system; it equally fits new and experienced users. New users are not forced to do difficult things, while experienced users are not forced to work in the same way they did when they first started learning Linux.
While development in the service area continues, great things are being done for desktop users, generally considered as the group least likely to know how a system works. Developers of desktop applications are making incredible efforts to make the most beautiful desktops you've ever seen, or to make your Linux machine look just like your former MS Windows or an Apple workstation. The latest developments also include 3D acceleration support and support for USB devices, single-click updates of system and packages, and so on. Linux has these, and tries to present all available services in a logical form that ordinary people can understand.

ARCHITECTURE OF THE LINUX OPERATING SYSTEM

Linux has all of the components of a typical OS

- Kernel
  The Linux kernel includes device driver support for a large number of PC hardware devices (graphics cards, network cards, hard disks etc.), advanced processor and memory

management features, and support for many different types of filesystems (including DOS floppies and the ISO9660 standard for CDROMs). In terms of the services that it provides to application programs and system utilities, the kernel implements most BSD and SYSV system calls, as well as the system calls described in the POSIX.1 specification.
The kernel (in raw binary form that is loaded directly into memory at system startup time) is typically found in the file /boot/vmlinuz, while the source files can usually be found in /usr/src/linux. The latest version of the Linux kernel sources can be downloaded from http://www.kernel.org/.

- Shells and GUIs

  Linux supports two forms of command input: through textual command line shells similar to those found on most UNIX systems (e.g. sh - the Bourne shell, bash - the Bourne again shell and csh - the C shell) and through graphical interfaces (GUIs) such as the KDE and GNOME window managers. If you are connecting remotely to a server your access will typically be through a command line shell.

- System Utilities

  Virtually every system utility that you would expect to find on standard implementations of UNIX (including every system utility described in the POSIX.2 specification) has been ported to Linux. This includes commands such as `ls, cp, grep, awk, sed, bc, wc, more`, and so on. These system utilities are designed to be powerful tools that do a single task extremely well (e.g. `grep` finds text inside files while `wc` counts the number of words, lines and bytes inside a file). Users can often solve problems by interconnecting these tools instead of writing a large monolithic application program.

  Like other UNIX flavours, Linux's system utilities also include server programs called daemons which provide remote network and administration services (e.g. `telnetd` and `sshd` provide remote login facilities, `lpd` provides printing services, `httpd` serves web pages, `crond` runs regular system administration tasks automatically). A daemon (probably derived from the Latin word which refers to a beneficient spirit who watches over someone, or perhaps short for "Disk And Execution MONitor") is usually spawned automatically at system startup and spends most of its time lying dormant waiting for some event to occur.

- Application programs

  Linux distributions typically come with several useful application programs as standard. Examples include the `emacs` editor, `xv` (an image viewer), `gcc` (a C compiler), `g++` (a C++ compiler), `xfig` (a drawing package), `latex` (a powerful typesetting language) and `soffice` (StarOffice, which is an MS-Office style clone that can read and write Word, Excel and PowerPoint files).

Redhat Linux also comes with `rpm`, the Redhat Package Manager which makes it easy to install and uninstall application programs.

3.3 PROPERTIES OF LINUX

Linux Pros

A lot of the advantages of Linux are a consequence of Linux' origins, deeply rooted in UNIX, except for the first advantage, of course:

- Linux is free

  Linux can be downloaded in its entirety from the Internet completely for free. No registration fees, no costs per user, free updates, and freely available source code in case you want to change the behavior of your system.

- Linux is portable to any hardware platform

  A vendor who wants to sell a new type of computer and who doesn't know what kind of OS his new machine will run (say the CPU in your car or washing machine), can take a Linux kernel and make it work on his hardware, because documentation related to this activity is freely available.

- Linux was made to keep on running

  As with UNIX, a Linux system expects to run without rebooting all the time. That is why a lot of tasks are being executed at night or scheduled automatically for other calm moments, resulting in higher availability during busier periods and a more balanced use of the hardware. This property allows for Linux to be applicable also in environments where people don't have the time or the possibility to control their systems night and day.

- Linux is secure and versatile

  The security model used in Linux is based on the UNIX idea of security, which is known to be robust and of proven quality. But Linux is not only fit for use as a fort against enemy attacks from the
  Internet: it will adapt equally to other situations, utilizing the same high standards for security. Your development machine or control station will be as secure as your firewall.

- Linux is scalable

  From a Palmtop with 2 MB of memory to a petabyte storage cluster with hundreds of nodes: add or remove the appropriate packages and Linux fits all. You don't need a supercomputer anymore, because you can use Linux to do big things using the building blocks provided with the system. If you want to do little things, such as making an operating system for an embedded processor or just recycling your old 486, Linux will do that as well.

- The Linux OS and most Linux applications have very short debug-times:
  Because Linux has been developed and tested by thousands of people, both errors and people to fix them are usually found rather quickly. It sometimes happens that there are only a couple of hours between discovery and fixing of a bug.

Linux Cons

- There are far too many different distributions

  At first glance, the amount of Linux distributions can be frightening, or ridiculous, depending on your point of view. But it also means that everyone will find what he or she needs. You don't need to be an expert to find a suitable release.

  When asked, generally every Linux user will say that the best distribution is the specific version he is using. So which one should you choose? All the releases contain more or less the same set of basic packages. On top of the basics, special third party software is added making, for example, TurboLinux more suitable for the small and medium enterprise, RedHat for servers and SuSE for workstations.

- Linux is not very user friendly and confusing for beginners

  It must be said that Linux, at least the core system, is less user friendly to use than MS Windows and certainly more difficult than MacOS, but... In light of its popularity, considerable effort has been made to make Linux even easier to use, especially for new users. More information is being released daily, such as this guide, to help fill the gap for documentation available to users at all levels.

- Is an Open Source product trustworthy?

  How can something that is free also be reliable? Linux users have the choice whether to use Linux or not, which gives them an enormous advantage compared to users of proprietary software, who don't have that kind of freedom. After long periods of testing, most Linux users come to the conclusion that Linux is not only as good, but in many cases better and faster that the traditional solutions. If Linux were not trustworthy, it would have been long gone, never knowing the popularity it has now, with millions of users. Now users can influence their systems and share their remarks with the community, so the system gets better and better every day. It is a project that is never finished, that is true, but in an ever changing environment, Linux is also a project that continues to strive for perfection.

4.0 CONCLUSION

You would have learned about the past and future development Linux as an operating system, its advantages and disadvantages.

5.0 SUMMARY

You have learned about the history of Linux operating system, the description of Linux interface as well as Pros and Cons of Linux OS.

6.0 TUTOR MARKED ASSIGNMENT

1. What are the pros and cons of Linux OS?

7.0 REFERENCES/FUTHER READINGS

1. Step-by-step tutorial on GNU/Linux based on Mandrace Linux by Augustin V. 2003
2. GNU/Linux Command line tools summary by Gareth Anderson, 2006.
3. Centos Essentials by Neil Smyth Techotopia, 2010.
4. Introduction to Linux by Machtelt Garrels.

UNIT TWO

LINUX COMMANDS AND UTILITIES

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
    3.1 Notational Conventions Used to Describe Linux Command
    3.2 Linux Commands

4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit, you will learn about Linux commands and notational conventions used to describe Linux commands.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Explain the Notational Conventions used to describe Linux commands
- Understand basic Linux commands

3.0 MAIN CONTENTS

3.1 NOTATIONAL CONVENTIONS USED IN LINUX COMMANDS

There is a set of accepted notational conventions used to describe, in a concise and consistent way, the correct syntax for any given Linux command. This specifies what options or parameters you must use, what options or parameters you can use or not use, and so on. Sometimes this set of conventions is used to give a complete and exhaustive listing of a command's syntax, showing every possible command and parameter. Sometimes it is used to make a particular example more general and the command's basic usage clearer.

If you remember the following six basic rules, you will be able, in principle, to understand the syntax of any Linux or UNIX command.

1. Any text standing by itself, and not within [], or {}, must be typed exactly as shown.

2. Any text within square brackets ([]) is optional. You can type it or not type it. For instance, the syntax ls [-l] means you must type ls (per the first rule), while adding -l is optional, but not necessary. Do not type the square brackets themselves! In our example, type ls or ls -l. Don't type ls [-l].

3. Angle brackets and the text within them must be replaced by appropriate text (usually a name or value). The text within the brackets usually indicates the nature of the replacement. For instance, the syntax more <filename> means that you should replace <filename> with the name of the file you wish to examine using more. If you want to look at the file test, you would type more test. Remember; do not use the angle brackets when you actually type the command!

4. Curly braces ({}) indicate that you must choose one of the values given within the braces. The values are separated by | (which in this case means or, not pipe!). For example, the syntax command -{a|b} means you must enter either command -a or command -b.

5. An ellipsis (...) means "and so on." It is normally used with parameters such as filenames, as described later.

6. The sixth basic rule states that the brackets can be combined as necessary. For instance, you don't have to type a filename with the more command. This would be indicated as more [<filename>]. The outer set of square brackets makes the entire parameter optional. If you do decide to use the parameter, replace the inner set of angle brackets with the appropriate value. Because the more command enables one or more filenames to be specified, the syntax becomes more [<filename>...]. The ellipsis means you can have as many <filenames> as you wish.

3.2: Directory and File Handling Commands

This section describes some of the more important directory and file handling commands.

- `pwd` (print [current] working directory)

  `pwd` displays the full absolute path to the your current location in the filesystem. So

  ```
  $ pwd↵
  /usr/bin
  ```

  implies that `/usr/bin` is the current working directory.

- `ls` (list directory)

  `ls` lists the contents of a directory. If no target directory is given, then the contents of the current working directory are displayed. So, if the current working directory is `/`,

```
$ ls↵
bin   dev  home  mnt   share  usr  var
boot  etc  lib   proc  sbin   tmp  vol
```

Actually, `ls` doesn't show you all the entries in a directory - files and directories that begin with a dot (.) are hidden (this includes the directories '.' and '..' which are always present). The reason for this is that files that begin with a . usually contain important configuration information and should not be changed under normal circumstances. If you want to see all files, `ls` supports the `-a` option:

```
$ ls -a↵
```

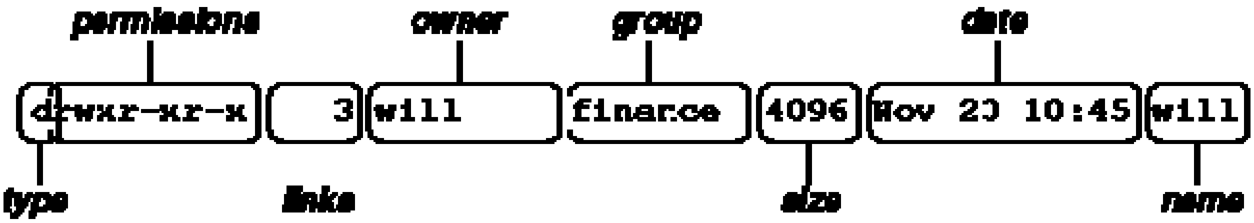
Even this listing is not that helpful - there are no hints to properties such as the size, type and ownership of files, just their names. To see more detailed information, use the `-l` option (long listing), which can be combined with the `-a` option as follows:

```
$ ls -a -l↵
```

(or, equivalently,)

```
$ ls -al↵
```

Each line of the output looks like this:



```
drwxr-xr-x    3 will     finance  4096 Nov 23 10:45 will
```

where:

- type is a single character which is either 'd' (directory), '-' (ordinary file), 'l' (symbolic link), 'b' (block-oriented device) or 'c' (character-oriented device).
- permissions is a set of characters describing access rights. There are 9 permission characters, describing 3 access types given to 3 user categories. The three access types are read ('r'), write ('w') and execute ('x'), and the three users categories are the user who owns the file, users in the group that the file belongs to and other users (the general public). An 'r', 'w' or 'x' character means the corresponding permission is present; a '-' means it is absent.
- links refers to the number of filesystem links pointing to the file/directory (see the discussion on hard/soft links in the next section).
- owner is usually the user who created the file or directory.
- group denotes a collection of users who are allowed to access the file according to the group access rights specified in the permissions field.

  - size is the length of a file, or the number of bytes used by the operating system to store the list of files in a directory.
  - date is the date when the file or directory was last modified (written to). The `-u` option display the time when the file was last accessed (read).
  - name is the name of the file or directory.

`ls` supports more options. To find out what they are, type:

```
$ man ls↵
```

`man` is the online UNIX user manual, and you can use it to get help with commands and find out about what options are supported. It has quite a terse style which is often not that helpful, so some users prefer to the use the (non-standard) `info` utility if it is installed:

```
$ info ls↵
```

- `cd` (change [current working] directory)

```
$ cd path
```

changes your current working directory to path (which can be an absolute or a relative path). One of the most common relative paths to use is '..' (i.e. the parent directory of the current directory).

Used without any target directory

```
$ cd↵
```

resets your current working directory to your home directory (useful if you get lost). If you change into a directory and you subsequently want to return to your original directory, use

```
$ cd -↵
```

- `mkdir` (make directory)

```
$ mkdir directory
```

creates a subdirectory called directoryin the current working directory. You can only create subdirectories in a directory if you have write permission on that directory.

- `rmdir` (remove directory)

  ```
  $ rmdir directory
  ```

  removes the subdirectory directory from the current working directory. You can only remove subdirectories if they are completely empty (i.e. of all entries besides the '.' and '..' directories).

- `cp` (copy)

  `cp` is used to make copies of files or entire directories. To copy files, use:

  ```
  $ cp source-file(s) destination
  ```

  where source-file(s) and destination specify the source and destination of the copy respectively. The behaviour of `cp` depends on whether the destination is a file or a directory. If the destination is a file, only one source file is allowed and `cp` makes a new file called destination that has the same contents as the source file. If the destination is a directory, many source files can be specified, each of which will be copied into the destination directory. Section 2.6 will discuss efficient specification of source files using wildcard characters.

  To copy entire directories (including their contents), use a recursive copy:

  ```
  $ cp -rd source-directories destination-directory
  ```

- `mv` (move/rename)

  `mv` is used to rename files/directories and/or move them from one directory into another. Exactly one source and one destination must be specified:

  ```
  $ mv source destination
  ```

  If destination is an existing directory, the new name for source (whether it be a file or a directory) will be destination/source. If source and destination are both files, source is renamed destination. N.B.: if destination is an existing file it will be destroyed and overwritten by source (you can use the `-i` option if you would like to be asked for confirmation before a file is overwritten in this way).

- `rm` (remove/delete)

  ```
  $ rm target-file(s)
  ```

removes the specified files. Unlike other operating systems, it is almost impossible to recover a deleted file unless you have a backup (there is no recycle bin!) so use this command with care. If you would like to be asked before files are deleted, use the `-i` option:

```
 $ rm -i myfile↵
rm: remove 'myfile'?
```

`rm` can also be used to delete directories (along with all of their contents, including any subdirectories they contain). To do this, use the `-r` option. To avoid `rm` from asking any questions or giving errors (e.g. if the file doesn't exist) you used the `-f` (force) option. Extreme care needs to be taken when using this option - consider what would happen if a system administrator was trying to delete user `will`'s home directory and accidentally typed:

```
 $ rm -rf / home/will↵
```

(instead of `rm -rf /home/will`).

- `cat` (catenate/type)

  `$ cat` target-file(s)

  displays the contents of target-file(s) on the screen, one after the other. You can also use it to create files from keyboard input as follows (> is the output redirection operator, which will be discussed in the next chapter):

  ```
  $ cat > hello.txt↵
  hello world!↵
  [ctrl-d]
  $ ls hello.txt↵
  hello.txt
  $ cat hello.txt↵
  hello world!
  $
  ```

- `more` and `less` (catenate with pause)

  `$ more` target-file(s)

  displays the contents of target-file(s) on the screen, pausing at the end of each screenful and asking the user to press a key (useful for long files). It also incorporates a searching facility (press '`/`' and then type a phrase that you want to look for).

You can also use `more` to break up the output of commands that produce more than one screenful of output as follows (`|` is the pipe operator, which will be discussed in the next chapter):

```
$ ls -l | more↵
```

`less` is just like `more`, except that has a few extra features (such as allowing users to scroll backwards and forwards through the displayed file). `less` not a standard utility, however and may not be present on all UNIX systems.

3.3: Making Hard and Soft (Symbolic) Links

Direct (hard) and indirect (soft or symbolic) links from one file or directory to another can be created using the `ln` command.

```
$ ln filename linkname
```

creates another directory entry for filename called linkname (i.e. linkname is a hard link). Both directory entries appear identical (and both now have a link count of 2). If either filename or linkname is modified, the change will be reflected in the other file (since they are in fact just two different directory entries pointing to the same file).

```
$ ln -s filename linkname
```

creates a shortcut called linkname (i.e. linkname is a soft link). The shortcut appears as an entry with a special type ('`l`'):

```
$ ln -s hello.txt bye.txt↵
$ ls -l bye.txt↵
lrwxrwxrwx   1 will finance 13 bye.txt -> hello.txt
$
```

The link count of the source file remains unaffected. Notice that the permission bits on a symbolic link are not used (always appearing as `rwxrwxrwx`). Instead the permissions on the link are determined by the permissions on the target (`hello.txt` in this case).

Note that you can create a symbolic link to a file that doesn't exist, but not a hard link. Another difference between the two is that you can create symbolic links across different physical disk devices or partitions, but hard links are restricted to the same disk partition. Finally, most current UNIX implementations do not allow hard links to point to directories.

3.4: Specifying multiple filenames

Multiple filenames can be specified using special pattern-matching characters. The rules are:

- '`?`' matches any single character in that position in the filename.
- '`*`' matches zero or more characters in the filename. A '`*`' on its own will match all files. '`*.*`' matches all files with containing a '`.`'.
- Characters enclosed in square brackets ('[' and ']') will match any filename that has one of those characters in that position.
- A list of comma separated strings enclosed in curly braces ("{" and "}") will be expanded as a Cartesian product with the surrounding characters.

For example:

1. `???` matches all three-character filenames.
2. `?ell?` matches any five-character filenames with '`ell`' in the middle.
3. `he*` matches any filename beginning with '`he`'.
4. `[m-z]*[a-l]` matches any filename that begins with a letter from '`m`' to '`z`' and ends in a letter from '`a`' to '`l`'.
5. `{/usr,}{/bin,/lib}/file` expands to `/usr/bin/file /usr/lib/file /bin/file` and `/lib/file`.

Note that the UNIX shell performs these expansions (including any filename matching) on a command's arguments before the command is executed.

3.5: File and Directory Permissions

As we have said earlier on, every file or directory on a UNIX system has three types of permissions, describing what operations can be performed on it by various categories of users. The permissions are read (`r`), write (`w`) and execute (`x`), and the three categories of users are user/owner (`u`), group (`g`) and others (`o`). Because files and directories are different entities, the interpretation of the permissions assigned to each differs slightly, as shown in Fig 3.3.

Permission	File	Directory
Read	User can look at the contents of the file	User can list the files in the directory
Write	User can modify the contents of the file	User can create new files and remove existing files in the directory
Execute	User can use the filename as a UNIX command	User can change into the directory, but cannot list the files unless (s)he has read permission. User can read files if (s)he has read permission on them.

Fig 3.3: Interpretation of permissions for files and directories

File and directory permissions can only be modified by their owners, or by the superuser (`root`), by using the `chmod` system utility.

- `chmod` (change [file or directory] mode)

  `$ chmod` options files

  `chmod` accepts options in two forms. Firstly, permissions may be specified as a sequence of 3 octal digits (octal is like decimal except that the digit range is 0 to 7 instead of 0 to 9). Each octal digit represents the access permissions for the user/owner, group and others respectively. The mappings of permissions onto their corresponding octal digits is as follows:

---	0
--x	1
-w-	2
-wx	3
r--	4
r-x	5
rw-	6
Rwx	7

For example the command:

```
$ chmod 600 private.txt
```

sets the permissions on `private.txt` to `rw-------` (i.e. only the owner can read and write to the file).

Permissions may be specified symbolically, using the symbols `u` (user), `g` (group), `o` (other), `a` (all), `r` (read), `w` (write), `x` (execute), `+` (add permission), `-` (take away permission) and = (assign permission). For example, the command:

```
$ chmod ug=rw,o-rw,a-x *.txt
```

sets the permissions on all files ending in `*.txt` to `rw-rw----` (i.e. the owner and users in the file's group can read and write to the file, while the general public do not have any sort of access).

`chmod` also supports a `-R` option which can be used to recursively modify file permissions, e.g.

```
$ chmod -R go+r play
```

will grant group and other read rights to the directory `play` and all of the files and directories within `play`.

- `chgrp` (change group)

  ```
  $ chgrp group files
  ```

  can be used to change the group that a file or directory belongs to. It also supports a `-R` option.

3.6: Inspecting File Content

Besides `cat` there are several other useful utilities for investigating the contents of files:

- `file` filename(s)

  `file` analyzes a file's contents for you and reports a high-level description of what type of file it appears to be:

  ```
   $ file myprog.c letter.txt webpage.html↵
  myprog.c:      C program text
  letter.txt:    English text
  webpage.html:  HTML document text
  ```

- `head, tail` filename

  `head` and `tail` display the first and last few lines in a file respectively. You can specify the number of lines as an option, e.g.

  ```
  $ tail -20 messages.txt↵
  $ head -5 messages.txt↵
  ```

  `tail` includes a useful `-f` option that can be used to continuously monitor the last few lines of a (possibly changing) file. This can be used to monitor log files, for example:

  ```
  $ tail -f /var/log/messages↵
  ```

  continuously outputs the latest additions to the system log file.

- `objdump` options binaryfile

objdump can be used to disassemble binary files - that is it can show the machine language instructions which make up compiled application programs and system utilities.

- od options filename (octal dump)

  od can be used to displays the contents of a binary or text file in a variety of formats, e.g.

```
$ cat hello.txt↵
hello world
$ od -c hello.txt↵
0000000  h   e   l   l   o       w   o   r   l   d \n
0000014
$ od -x hello.txt↵
0000000 6865 6c6c 6f20 776f 726c 640a
0000014
```

There are also several other useful content inspectors that are non-standard (in terms of availability on UNIX systems) but are nevertheless in widespread use. They are summarised in Fig. 3.4.

File type	Typical extension	Content viewer
Portable Document Format	.pdf	Acroread
Postscript Document	.ps	Ghostview
DVI Document	.dvi	Xdvi
JPEG Image	.jpg	Xv
GIF Image	.gif	Xv
MPEG movie	.mpg	mpeg_play
WAV sound file	.wav	Realplayer
HTML document	.html	Netscape

Fig 3.4: Other file types and appropriate content viewers.

3.7: Finding Files

There are at least three ways to find files when you don't know their exact location:

- find

  If you have a rough idea of the directory tree the file might be in (or even if you don't and you're prepared to wait a while) you can use find:

```
$ find directory -name targetfile -print ↵
```

find will look for a file called targetfile in any part of the directory tree rooted at directory. targetfile can include wildcard characters. For example:

```
$ find /home -name "*.txt" -print 2>/dev/null↵
```

will search all user directories for any file ending in ".txt" and output any matching files (with a full absolute or relative path). Here the quotes (") are necessary to avoid filename expansion, while the 2>/dev/null suppresses error messages (arising from errors such as not being able to read the contents of directories for which the user does not have the right permissions).

find can in fact do a lot more than just find files by name. It can find files by type (e.g. -type f for files, -type d for directories), by permissions (e.g. -perm o=r for all files and directories that can be read by others), by size (-size) etc. You can also execute commands on the files you find. For example,

```
$ find . -name "*.txt" -exec wc -l '{}' ';'
```

counts the number of lines in every text file in and below the current directory. The '{}' is replaced by the name of each file found and the ';' ends the -exec clause.

For more information about find and its abilities, use man find and/or info find.

- which (sometimes also called whence) command

  If you can execute an application program or system utility by typing its name at the shell prompt, you can use which to find out where it is stored on disk. For example:

  ```
  $ which ls↵
  /bin/ls
  ```

- locate string

  find can take a long time to execute if you are searching a large filespace (e.g. searching from / downwards). The locate command provides a much faster way of locating all files whose names match a particular search string. For example:

  ```
  $ locate ".txt"↵
  ```

will find all filenames in the filesystem that contain "`.txt`" anywhere in their full paths.

One disadvantage of `locate` is it stores all filenames on the system in an index that is usually updated only once a day. This means `locate` will not find files that have been created very recently. It may also report filenames as being present even though the file has just been deleted. Unlike `find`, `locate` cannot track down files on the basis of their permissions, size and so on.

3.8: Finding Text in Files

- `grep` (General Regular Expression Print)

```
$ grep options pattern files ↵
```

`grep` searches the named files (or standard input if no files are named) for lines that match a given pattern. The default behaviour of `grep` is to print out the matching lines. For example:

```
$ grep hello *.txt ↵
```

searches all text files in the current directory for lines containing "hello". Some of the more useful options that `grep` provides are: `-c` (print a count of the number of lines that match), `-i` (ignore case), `-v` (print out the lines that don't match the pattern) and `-n` (printout the line number before printing the matching line). So

```
$ grep -vi hello *.txt ↵
```

searches all text files in the current directory for lines that do not contain any form of the word hello (e.g. Hello, HELLO, or hELlO).

If you want to search all files in an entire directory tree for a particular pattern, you can combine `grep` with `find` using backward single quotes to pass the output from `find` into `grep`. So

```
$ grep hello `find . -name "*.txt" -print` ↵
```

will search all text files in the directory tree rooted at the current directory for lines containing the word "hello".

The patterns that `grep` uses are actually a special type of pattern known as regular expressions. Just like arithmetic expressions, regular expressions are made up of basic subexpressions combined by operators.

The most fundamental expression is a regular expression that matches a single character. Most characters, including all letters and digits, are regular expressions that match themselves. Any other character with special meaning may be quoted by preceding it with a backslash (\). A list of characters enclosed by '[' and ']' matches any single character in that list; if the first character of the list is the caret `^', then it matches any character not in the list. A range of characters can be specified using a dash (-) between the first and last items in the list. So [0-9] matches any digit and [^a-z] matches any character that is not a digit.

The caret `^' and the dollar sign `$' are special characters that match the beginning and end of a line respectively. The dot '.' matches any character. So

```
    $ grep ^..[l-z]$ hello.txt↵
```

matches any line in hello.txt that contains a three character sequence that ends with a lowercase letter from l to z.

egrep (extended grep) is a variant of grep that supports more sophisticated regular expressions. Here two regular expressions may be joined by the operator `|'; the resulting regular expression matches any string matching either subexpression. Brackets '(' and ')' may be used for grouping regular expressions. In addition, a regular expression may be followed by one of several repetition operators:

`?' means the preceding item is optional (matched at most once).
`*' means the preceding item will be matched zero or more times.
`+' means the preceding item will be matched one or more times.
`{N}' means the preceding item is matched exactly N times.
`{N,}' means the preceding item is matched N or more times.
`{N,M}' means the preceding item is matched at least N times, but not more than M times.

For example, if egrep was given the regular expression

```
    '(^[0-9]{1,5}[a-zA-Z ]+$)|none'
```

it would match any line that either:

- begins with a number up to five digits long, followed by a sequence of one or more letters or spaces, or
- contains the word none

You can read more about regular expressions on the grep and egrep manual pages.

Note that UNIX systems also usually support another `grep` variant called `fgrep` (fixed grep) which simply looks for a fixed string inside a file (but this facility is largely redundant).

3.9: Sorting files

There are two facilities that are useful for sorting files in UNIX:

- `sort` filenames

  `sort` sorts lines contained in a group of files alphabetically (or if the `-n` option is specified) numerically. The sorted output is displayed on the screen, and may be stored in another file by redirecting the output. So

  ```
  $ sort input1.txt input2.txt > output.txt↵
  ```

  outputs the sorted concentenation of files `input1.txt` and `input2.txt` to the file `output.txt`.

- `uniq` filename

  `uniq` removes duplicate adjacent lines from a file. This facility is most useful when combined with `sort`:

  ```
  $ sort input.txt | uniq > output.txt↵
  ```

3.10: File Compression and Backup

UNIX systems usually support a number of utilities for backing up and compressing files. The most useful are:

- `tar` (tape archiver)

  `tar` backs up entire directories and files onto a tape device or (more commonly) into a single disk file known as an archive. An archive is a file that contains other files plus information about them, such as their filename, owner, timestamps, and access permissions. `tar` does not perform any compression by default.

  To create a disk file `tar` archive, use

  ```
  $ tar -cvf archivenamefilenames
  ```

  where archivename will usually have a `.tar` extension. Here the `c` option means create, `v` means verbose (output filenames as they are archived), and `f` means file. To list the contents of a `tar` archive, use

```
$ tar -tvf archivename
```

To restore files from a `tar` archive, use

```
$ tar -xvf archivename
```

- `cpio`

  `cpio` is another facility for creating and reading archives. Unlike `tar`, `cpio` doesn't automatically archive the contents of directories, so it's common to combine `cpio` with `find` when creating an archive:

  ```
  $ find . -print -depth | cpio -ov -Htar > archivename
  ```

  This will take all the files in the current directory and the directories below and place them in an archive called archivename. The `-depth` option controls the order in which the filenames are produced and is recommended to prevent problems with directory permissions when doing a restore. The `-o` option creates the archive, the `-v` option prints the names of the files archived as they are added and the `-H` option specifies an archive format type (in this case it creates a `tar` archive). Another common archive type is `crc`, a portable format with a checksum for error control.

  To list the contents of a `cpio` archive, use

  ```
  $ cpio -tv < archivename
  ```

  To restore files, use:

  ```
  $ cpio -idv < archivename
  ```

  Here the `-d` option will create directories as necessary. To force `cpio` to extract files on top of files of the same name that already exist (and have the same or later modification time), use the `-u` option.

- `compress`, `gzip`

  `compress` and `gzip` are utilities for compressing and decompressing individual files (which may be or may not be archive files). To compress files, use:

  ```
  $ compress filename
  ```

  or

  ```
  $ gzip filename
  ```

In each case, filename will be deleted and replaced by a compressed file called filename.Z or filename.gz. To reverse the compression process, use:

```
    $ compress -d filename
```

or

```
    $ gzip -d filename
```

3.11: Handling Removable Media (e.g. floppy disks)

UNIX supports tools for accessing removable media such as CDROMs and floppy disks.

- `mount, umount`

  The `mount` command serves to attach the filesystem found on some device to the filesystem tree. Conversely, the `umount` command will detach it again (it is very important to remember to do this when removing the floppy or CDROM). The file `/etc/fstab` contains a list of devices and the points at which they will be attached to the main filesystem:

  ```
  $ cat /etc/fstab↵
  /dev/fd0   /mnt/floppy  auto    rw,user,noauto  0 0
  /dev/hdc   /mnt/cdrom   iso9660 ro,user,noauto  0 0
  ```

  In this case, the mount point for the floppy drive is `/mnt/floppy` and the mount point for the CDROM is `/mnt/cdrom`. To access a floppy we can use:

  ```
  $ mount /mnt/floppy↵
  $ cd /mnt/floppy↵
  $ ls (etc...)
  ```

  To force all changed data to be written back to the floppy and to detach the floppy disk from the filesystem, we use:

  ```
  $ umount /mnt/floppy
  ```

- `mtools`

  If they are installed, the (non-standard) `mtools` utilities provide a convenient way of accessing DOS-formatted floppies without having to mount and unmount filesystems. You can use DOS-type commands like "`mdir a:`", "`mcopy a:*.* .`", "`mformat a:`", etc. (see the `mtools` manual pages for more details).

4.0 CONCLUSION

You would have learned about the Linux commands and notational conventions used to describe Linux commands

5.0 SUMMARY

You have learned about Linux commands and the notational conventions used to describe Linux commands

ACTIVITY B

1.0 What do these brackets signify?

- i. { }
- ii. [ ]
- iii. < >

6.0 TUTOR MARKED ASSIGNMENT

1. List 5 Linux commands and their functions

7.0 REFERENCES/FUTHER READINGS

1. Step-by-step tutorial on GNU/Linux based on Mandrace Linux by Augustin V. 2003
2. GNU/Linux Command line tools summary by Gareth Anderson, 2006.
3. Centos Essentials by Neil Smyth Techotopia, 2010.
4. Introduction to Linux by Machtelt Garrels.

UNIT THREE

LINUX UTILITIES AND EDITOR

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
  3.1 Mastering an Editor
  3.2 Gnu Emacs
  3.3 Vi Improved
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit, we will discuss the importance of mastering an editor. We will focus mainly on the improved VI editor.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Explain GNU Emacs
- Explain Vi improved

3.0 MAIN CONTENTS

3.1 MASTERING AN EDITOR

It is very important to be able to use at least one text mode editor. Knowing how to use an editor on your system is the first step to independence, we need it to edit files that influence our environment.

As an advanced user, you may want to start writing scripts, or books, develop websites or new programs.

Mastering an editor will immensely improve your productivity as well as your capabilities.

Our focus here is on text editors, which can also be used on systems without a graphical environment and in terminal windows. The additional advantage of a text editor is in using it on remote machines.

Since you don't need to transfer the entire graphical environment over the network, working with text editors tremendously improves network speed.

3.2 GNU EMACS

Emacs is the extensible, customizable, self-documenting, real-time display editor, known on many UNIX and other systems. The text being edited is visible on the screen and is updated automatically as you type your commands. It is a real-time editor because the display is updated very frequently, usually after each character or pair of characters you type. This minimizes the amount of information you must keep in your head as you edit. Emacs is called advanced because it provides facilities that go beyond simple insertion and deletion: controlling sub-processes; automatic indentation of programs; viewing two or more files at once; editing formatted text; and dealing in terms of characters, words, lines, sentences, paragraphs, and pages, as well as expressions and comments in several different programming languages.

Self-documenting means that at any time you can type a special character, Ctrl+H, to find out what your options are. You can also use it to find out what any command does, or to find all the commands that pertain to a topic.

Customizable means that you can change the definitions of Emacs commands in little ways. For example, if you use a programming language in which comments start with "<**" and end with "**>", you can tell the Emacs comment manipulation commands to use those strings. Another sort of customization is rearrangement of the command set. For example, if you prefer the four basic cursor motion commands (up, down, left and right) on keys in a diamond pattern on the keyboard, you can rebind the keys that way.

Extensible means that you can go beyond simple customization and write entirely new commands, programs in the Lisp language that are run by Emacs's own Lisp interpreter. Emacs is an online extensible system, which means that it is divided into many functions that call each other, any of which can be redefined in the middle of an editing session. Almost any part of Emacs can be replaced without making a separate copy of all of Emacs.

Most of the editing commands of Emacs are written in Lisp already; the few exceptions could have been written in Lisp but are written in C for efficiency. When run under the X Window System (started as xemacs) Emacs provides its own menus and convenient bindings to mouse buttons. But Emacs can provide many of the benefits of a window system on a text-only terminal. For instance, you can look at or edit several files at once, move text between files, and edit files while running shell commands.

3.3 VI IMPROVED

Vim stands for "Vi IMproved". It used to be "Vi IMitation", but there are so many improvements that a name change was appropriate. Vim is a text editor which includes almost all the commands from the UNIX program vi and a lot of new ones.

Commands in the vi editor are entered using only the keyboard, which has the advantage that you can keep your fingers on the keyboard and your eyes on the screen, rather than moving your arm repeatedly to the mouse. For those who want it, mouse support and a GUI version with scrollbars and menus can be activated. We will refer to vi or vim throughout this unit for editing files, while you are of course free to use the editor of your choice.

However, it is recommended to at least get the vi basics in the fingers, because it is the standard text editor on almost all UNIX systems, while Emacs can be an optional package. There may be small differences between different computers and terminals, but the main point is that if you can work with vi, you can survive on any UNIX system.

Apart from the vim command, the Vim packages may also provide gvim, the Gnome version of vim. Beginners might find this easier to use, because the menus offer help when you forgot or don't know how to perform a particular editing task using the standard vim commands.

4.0 CONCLUSION

You would have learned about the importance of mastering an editor. We will focus mainly on the improved VI editor.

5.0 SUMMARY

You have learned about the importance of mastering at least an editor, the GNU Emacs editor and Vim editor.

ACTIVITY B

1.0 Discuss the properties of GNU Emacs editor

6.0 TUTOR MARKED ASSIGNMENT

1. Why do you have to master at least one editor?

7.0 REFERENCES/FUTHER READINGS

1. Step-by-step tutorial on GNU/Linux based on Mandrace Linux by Augustin V. 2003

2. GNU/Linux Command line tools summary by Gareth Anderson, 2006.

3. Centos Essentials by Neil Smyth Techotopia, 2010.

4. Introduction to Linux by Machtelt Garrels.

5. GNU Emacs manual by Richard Stallman, 2007.

UNIT FOUR

UNIX SYSTEM ADMINISTRATION

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
3.1 The UNIX File system
3.2 UNIX Processes
3.3 UNIX System Startup, Shutdown and Reboot
3.4 Building UNIX Kernel
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about UNIX File system, UNIX processes and reasons for building a new kernel

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Describe UNIX File system
- Explain UNIX Processes
- How to Startup, Shutdown and Reboot UNIX System
- Explain Reasons for building a kernel

3.0 MAIN CONTENTS

3.1 THE UNIX FILE SYSTEM

Under UNIX we can think of the file system as everything being a file. Thus directories are really nothing more than files containing the names of other files and so on. In addition, the file system is used to represent physical devices such as tty lines or even disk and tape units.

The UNIX operating system is built around the concept of a filesystem which is used to store all of the information that constitutes the long-term state of the system. This state includes the operating system kernel itself, the executable files for the commands supported by the operating system, configuration information, temporary workfiles, user data, and various special files that are used to give controlled access to system hardware and operating system functions.

Each file on the system has what is called an inode that contains information on the file. To see the fields of the inode look at manual page of the stat system call. This shows the following fields:

```
 struct stat {
    dev_t   st_dev;    /* device inode resides on */
    ino_t   st_ino;    /* this inode's number */
    u_short st_mode;   /* protection */
    short   st_nlink;  /* number or hard links to the file */
    short   st_uid;    /* user-id of owner */
    short   st_gid;    /* group-id of owner */
    dev_t   st_rdev;   /* the device type, for inode that is device */
    off_t   st_size;   /* total size of file */
    time_t  st_atime;  /* file last access time */
    int     st_spare1;
    time_t  st_mtime;   /* file last modify time */
    int     st_spare2;
    time_t  st_ctime;   /* file last status change time */
    int     st_spare3;
    long st_blksize; /* optimal blocksize for file system i/o ops */
    long st_blocks;  /* actual number of blocks allocated */
    long st_spare4;
    u_long st_gennum; /* file generation number */
      };
```

The key fields in the structure are st_mode (the permission bits), st_uid the UID, st_gid the GID, and st_*time (assorted time fields).

The ls -l command is used to look at all of those fields.

Every item stored in a UNIX filesystem belongs to one of four types:

1. Ordinary files

   Ordinary files can contain text, data, or program information. Files cannot contain other files or directories. Unlike other operating systems, UNIX filenames are not broken into a name part and an extension part (although extensions are still frequently used as a means to classify files). Instead they can contain any keyboard character except for '/' and be up to 256 characters long (note however that characters such as *,?,# and & have special meaning in most shells and should not therefore be used in filenames). Putting spaces in filenames also makes them difficult to manipulate - rather use the underscore '_'.

2. Directories
   Directories are containers or folders that hold files, and other directories.

3. Devices
   To provide applications with easy access to hardware devices, UNIX allows them to be used in much the same way as ordinary files. There are two types of devices in UNIX - block-oriented devices which transfer data in blocks (e.g. hard disks) and character-oriented devices that transfer data on a byte-by-byte basis (e.g. modems and dumb terminals).
4. Links
   A link is a pointer to another file. There are two types of links - a hard link to a file is indistinguishable from the file itself. A soft link (or symbolic link) provides an indirect pointer or shortcut to a file. A soft link is implemented as a directory file entry containing a pathname.

3.2 Typical UNIX Directory Structure

The UNIX filesystem is laid out as a hierarchical tree structure which is anchored at a special top-level directory known as the root (designated by a slash '/'). Because of the tree structure, a directory can have many child directories, but only one parent directory. Fig. 3.5 illustrates this layout.

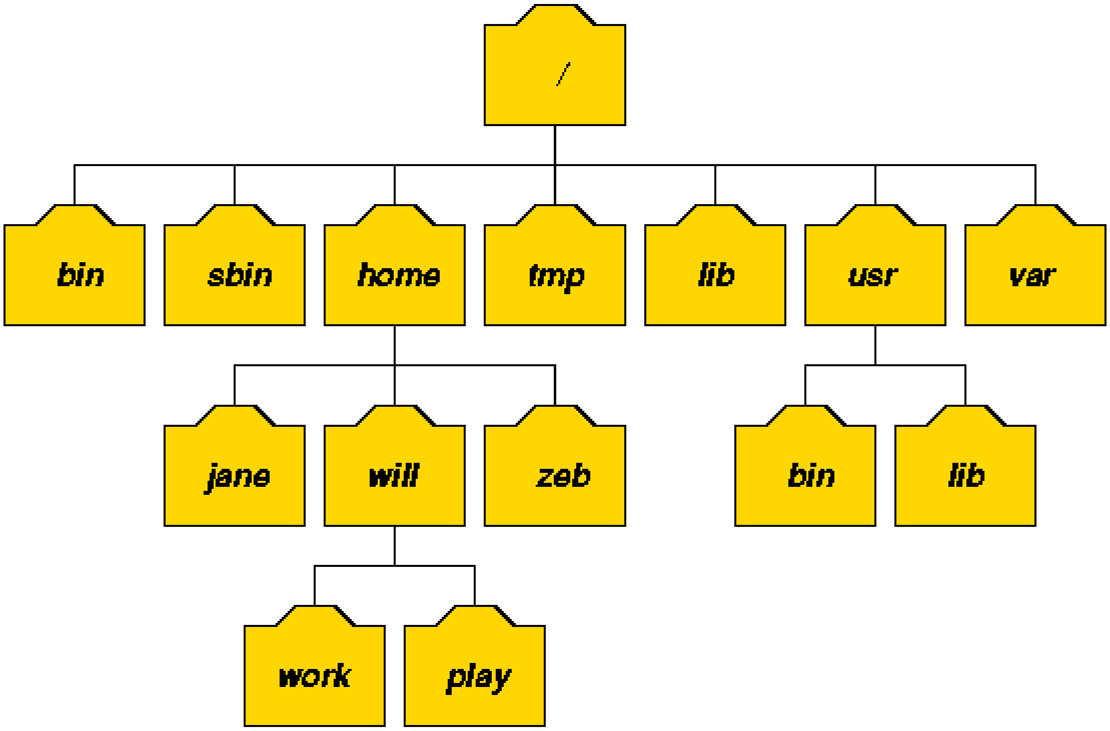


Fig. 3.5: Part of a typical UNIX filesystem tree

To specify a location in the directory hierarchy, we must specify a path through the tree. The path to a location can be defined by an absolute path from the root /, or as a relative path from the current working directory. To specify a path, each directory along the route from the source to the destination must be included in the path, with each directory in the sequence being separated by a slash. To help with the specification of relative paths, UNIX provides the shorthand "`.`" for the current directory and "`..`" for the parent directory. For example, the absolute path to the directory "`play`" is `/home/will/play`, while the relative path to this directory from "`zeb`" is `../will/play`.

Fig. 3.6 shows some typical directories you will find on UNIX systems and briefly describes their contents. Note that these although these subdirectories appear as part of a seamless logical filesystem, they do not need be present on the same hard disk device; some may even be located on a remote machine and accessed across a network.

Directory	Typical Contents
/	The "root" directory
/bin	Essential low-level system utilities
/usr/bin	Higher-level system utilities and application programs
/sbin	Superuser system utilities (for performing system administration tasks)

/lib	Program libraries (collections of system calls that can be included in programs by a compiler) for low-level system utilities
/usr/lib	Program libraries for higher-level user programs
/tmp	Temporary file storage space (can be used by any user)
/home or /homes	User home directories containing personal file space for each user. Each directory is named after the login of the user.
/etc	UNIX system configuration and information files
/dev	Hardware devices
/proc	A pseudo-filesystem which is used as an interface to the kernel. Includes a sub-directory for each active program (or process).

Fig. 3.6: Typical UNIX directories

When you log into UNIX, your current working directory is your user home directory. You can refer to your home directory at any time as "`~`" and the home directory of other users as "`~<login>`". So `~will/play` is another way for user `jane` to specify an absolute path to the directory `/homes/will/play`. User `will` may refer to the directory as `~/play.`

File times.

UNIX records three file times in the inode; these are referred to as ctime, mtime, and atime. The ctime field refers to the time the inode was last changed, mtime refers to the last modification time of the file, and atime refers to the time the file was last accessed.

The ctime file of the inode is updated whenever the file is written to, protections are changed, or the ownership changed. Usually, ctime is a better indication of file modification than the mtime field. The mtime and atime fields can easily be changed through a system call in C (or a perl script). The ctime field is a little harder to change, although not impossible.

File times are important because they are used in many ways by system administrators. For example, when performing backups, an incremental dump will check the mtime of the inode to see if a file has been modified and should be written to tape. Also, system administrators often check the mtime of certain key system files when looking for signs of tampering (while sometimes useful, a hacker will sufficient skill will reset the mtime back).

Finally, when managing disk space, some sites have a policy where files not accessed in a certain time are marked for archival; it is not uncommon to have certain users deliberately set the atime or mtime to defeat this policy.

File Permissions

File permissions are used to control access to files on the system. Clearly in a multi-user system some method must be devised that allows users to select files for sharing with other users while at the same time selecting other files to keep private. Under UNIX, the inode maintains a set of 12 mode bits. Three of the mode bits correspond to special permissions, while the other nine are general user permissions.

The nine general file permissions are divided into three groups of three. The three groups correspond to owner, group, and other. Within each group there are three distinct permissions, read, write, and execute. The nine general file permissions are listed via the ls -l.

Read (r)

Read access means you can open the file with the open system call and can read the contents of the file with the read system call.

Write (w)

Write access means you can overwrite the file or modify its contents. It gives you access to the system calls write and truncate.

Execute(x)

Execute access means you can specify the path of this file and run it as a program. When a file name is specified to the shell the shell examines the file for execute access and calls the exec system call. The first two bytes of the file are checked for the system magic number, signifying the file is an executable. If the magic number is not contained in the first two bytes the file is assumed to be a shell script.

The file permissions described above apply to plain files, devices, sockets, and FIFOs. These permissions do not apply to directories and symbolic links. Symbolic links have no permission control on the link; all access is resolved by examining the permissions on the target of the link.

Some anomalies can develop, for example, it is possible to set permissions so that a program can be run but the file cannot be read. Also, it is possible to set permissions so that anyone on the system, except members of your group can read the file.

UNIX Processes

A process is a program in execution. Every time you invoke a system utility or an application program from a shell, one or more "child" processes are created by the shell in response to your command. All UNIX processes are identified by a unique process identifier or PID. An important process that is always present is the `init` process. This is the first process to be created when a

UNIX system starts up and usually has a PID of 1. All other processes are said to be "descendants" of `init`.

A process under UNIX consists of an address space and a set of data structures in the kernel to keep track of that process. The address space is a section of memory that contains the code to execute as well as the process stack.

The kernel must keep track of the following data for each process on the system:

- The address space map
- The current status of the process
- The execution priority of the process
- The resource usage of the process
- The current signal mask
- The owner of the process

A process has certain attributes that directly affect execution, these include:

- PID - The PID stands for the process identification. This is a unique number that defines the process within the kernel.
- PPID - This is the processes Parent PID, the creator of the process.
- UID - The User ID number of the user that owns this process.
- EUID - The effective User ID of the process.
- GID - The Group ID of the user that owns this process.
- EGID - The effective Group User ID that owns this process.
- Priority - The priority that this process runs at.

To view a process you use the ps command.

```
umbc9[8]# ps -l

 F  S   UID   PID       PPID  C PRI NI P   SZ:RSS     WCHAN    TTY       TIME COMD

30 S    0    11660   145  1  26 20   *      66:20       88249f10  ttyq6    0:00    rlogind

30 S 14066 11662   11661  26 36   *      129:43      88249f10  ttyq6    0:00    zwgc
```

The man page for ps describes all the fields displayed with the ps command as well as all the command options. Some important fields you must know are the following:

The F field

> This is the flag field. It uses hexadecimal values which are added to show the value of the flag bits for the process. For a normal user process this will be 30, meaning it is loaded into memory.

The S field

The S field is the state of the process, the two most common values are S for Sleeping and R for Running. An important value to look for is X, which means the process is waiting for memory to become available. When you see this frequently on your system you are out of memory.

UID field

The UID field shows the User ID (UID) of the process owner. For many processes this is 0 because they are run setuid.

PID field

The PID shows the Process ID of each process. This value should be unique. Generally, PID is allocated lowest to highest, but wrap at some point. This value is necessary for you to send a signal to a process such as the KILL signal.

PPID field

This refers to the Parent Process ID. This identifies the parent process that started the process. Using this it allows you to trace the sequence of process creation that took place.

PRI field

This stands for priority field. This refers to the process NICE value. It ranges from 0 to 39. The default is 20, as a process uses the CPU the system will raise the nice value. This value is used by the scheduler to compute the next process to get the CPU.

The P flag

This is the processor flag. On the SGI this refers to the processor the process is running on.

SZ field

This refers to the SIZE field. This is the total number of pages in the process. Each page is 4096 bytes. The sort command is your friend when looking at the system. Use the sort command as the pipe output to sort by size or PID. For example to sort by SZ field use the command ps -el | sort +9 (remember sort starts numbering fields with zero).

RSS field

This refers to Resident Set Size and refers to the pages in memory. Note the RSS size should ALLOWS to be less than the SZ.

TTY field

This is the terminal assigned to your process. On SGI based systems tty's with the letter "q" in them are psuedo, or network, tty's.

Time field

The cumulative execution time of the process in minutes and seconds.

COMD field

The command that was executed.

As a system administrator you often want to look at all processes, this is done under SV5 with the command ps -el or under BSD with the command ps -al. There are a number of variations that control what information is printed out.

Pipes

The pipe ('`|`') operator is used to create concurrently executing processes that pass data directly to one another. It is useful for combining system utilities to perform more complex functions. For example:

```
$ cat hello.txt | sort | uniq↵
```

creates three processes (corresponding to `cat`, `sort` and `uniq`) which execute concurrently. As they execute, the output of the who process is passed on to the `sort` process which is in turn passed on to the `uniq` process. `uniq` displays its output on the screen (a sorted list of users with duplicate lines removed). Similarly:

```
$ cat hello.txt | grep "dog" | grep -v "cat"↵
```

finds all lines in `hello.txt` that contain the string "`dog`" but do not contain the string "`cat`".

Sending a Signal

UNIX supports the idea of sending software signals to a process. These signals are ways for other processes to interact with a running process outside the context of the hardware. The kill command is used to send a signal to a process. In addition, it is possible to write a signal handler in either C or the Shell that responds to a signal being sent. For example, many system administration utilities, such as the name server, respond to SIGHUP signal by re-reading their configuration file. This can then be used to update the process while running without having to terminate and restart the process.

For many signals there is really nothing that can be done other than printing an appropriate error message and terminating the process. The signals that system administrators will use the most are the HUP, KILL, and STOP signals. The HUP signal as mentioned previously is used by some utilities as a way to notify the process to do something. The KILL signal is used to abort a process. The STOP command is used to pause a process.

A common problem system administrators will see is one where a user made a mistake and is continually forking new processes. While all users have some limit on the number of processes they can fork, as they reach that limit they will wait, if you kill a process the system will resume creating new processes on behalf of the user. The best way to handle this is to send the STOP signal to all processes. In this way, all processes are now suspended, and then you can send a KILL signal to the processes. Since the processes were first suspended they can't create new processes as you kill the ones off.

The Process Termination Command - kill

The kill command is used to terminate processes that can't be stopped by other means.

Before going through the following procedure, if it's a program you're stuck in, make sure you can't stop or exit it by typing Ctrl-C or some other key combination.

1. Switch to another virtual console and log in as root.

2. Run ps -u and identify the offending process. You will use its PID in the next step.

3. Use the kill program by typing kill <PID>, where PID is the Process ID you wish to kill. Make sure that you have correctly identified the offending process! As root, you can kill any user process, including the wrong one if you misread or mistype the PID.

4. Verify that the process has been killed by using ps -u again. You can type ps -u <PID>, which shows you the status of only the specified PID. If there's a null result and you're just given the Linux prompt again, the PID is dead, so go to step 8. However, it's best to look at the complete ps -u list if it's not too long. Sometimes the offending process reappears with a new PID! If that is the case, go to step6.

5. If the process is still alive and has the same PID, use kill's 9 option. Type kill -9 <PID>. Check it as in step 4. If this does not kill the process, go to step 7. If the process is now dead, go to step 8.

6. If the offending process has reappeared with a new PID that means that it's being created automatically by some other process. The only thing to do now is to kill the parent process, which is the true offender! You might also have to kill the parent process when kill -9 does not work.

7. Use ps -l to identify the troublesome process's PPID. This is the PID of the parent process. You should check the parent's identity more closely by typing ps -u <Parent PID> before going ahead and killing it as described in step 3, using the PID of the parent in the kill command. You should follow through with step 4 and, if necessary, step 5, making sure the parent process has been killed.

8. The process is killed. Remember to log off. You should not leave root logged in on virtual consoles, because you will forget that the root logins are there!

Sometimes processes are unkillable, in this case, it better to shut down the Linux system and reboot the system.

Linux keeps ordinary users (as opposed to root) from killing other users' processes (maliciously or otherwise). For instance, if you are an ordinary user and you try to kill the init process, which always has PID=1, you will see

```
darkstar:~$ kill 1

kill: (1) - Not owner
```

Setting processes priority.
UNIX attempts to manage priority by giving those who have used the least access first.

In addition, those users who are sleeping on an event (e.g. such as a keyboard press) get higher priority than those jobs that are purely CPU driven. On any large system with a number of competing user groups the task of managing resources falls on the system administrator. This task is both technical and political. As a system administrator one MUST understand the company goals in order to manage this task. Often, the most prolific users of a machine are in fact the most important.

Once you understand the political implications on who should get priority you are ready to manage the technical details. As root, you can change the priority of any process on the system. Before doing this it is critical to understand how priority works and what makes sense. First, while CPU is the most watched resource on a system it is not the only one. Memory usage, disk usage, IO activity, number of processes, all tied together in determining throughput of the machine. For example, given two groups, A and B both groups require large amounts of memory, more than is available when both are running simultaneously. Raising the priority of group A over Group B may not help things if Group B does not fully relinquish the memory it is using. While the paging system will do this over time, the process of swapping a process out to disk can be intensive and greatly reduce performance, especially if this becomes a recurring problem if process B gets swapped back in. Possibly a better alternative is to completely stop process B with a signal and then continue it later when A has finished.

3.4. SYSTEM STARTUP, SHUTDOWN AND REBOOT

System Startup

Note: System startup is machine dependent; it is better to consult system manual for exact details.

Below is an outline of the steps that go into bringing UNIX up on a machine:

1. Bootstrapping UNIX into memory and Initializing the Kernel's data structure.
2. Hardware probing and configuration for SCSI
3. Machine independent initialization.
4. Operator intervention (Single User Mode) .
5. Execution of initialization Scripts.
6. Multi-user operation.

Problems that can keep the system from booting fall into the following categories:

- Hardware problems.
- Boot problems such as defective media or a broken network.
- Damaged file systems.
- Improperly configured kernel.
- Errors in the startup scripts.

UNIX Shutdown and Reboot

It is critical for system administrators to have a firm understanding of how the machine is being used and actively involve the users in scheduling downtime. For example, on most systems, a shutdown will cause all user processes to be killed. If users on a system are running jobs that take days or weeks to complete then shutting the system down and causing all processes to be killed could severely impact the productivity of users. Whenever possible, users should be given as much lead time as possible when scheduling a shutdown. Once brought up to multi-user mode it is not uncommon for the system to run for many days, possibly even months, without being shutdown or rebooted. There are valid reasons for shutting down the system, these include:

- Many systems now have a graphics display and use an assortment of X11 based applications. Also, it is not uncommon for a server machine to support remote X11 applications. Under many vendors version of X11 there are known memory leaks. These memory leaks result in the X11 server or application allocating memory and never releasing it. Over time you may find that free memory becomes tight. Rebooting will eliminate that.
- Installation of system software or changes in hardware often requires a system reboot to take effect.

- Devices can get in a state where they don't function properly. The only fix is to shutdown the system and power off the component. Likewise, system software may get in a confused state and require a reboot to be corrected.
- Often, system administrators bring the system down to single-user mode to perform full backups of file systems. Performing a full backup on a quiescent is one way of guaranteeing a complete backup.

Methods of shutting down and rebooting

There are three possible states you can end up in when you start to shutdown the system, these are:

- Single-user mode;
- The system is completely shut down and ready to be powered off;
- The system is shutdown put then brought immediately back up without any intervention.

Single-user mode

Previously when we discussed single-user mode we went over some of the tasks you may want to accomplish here. To leave multi-user mode under a BSD system you can enter the command shutdown time [message], where time can be in absolute or relative terms. For relative time, use a value such as +5 to refer to five minutes from now. Absolute time is referenced as HH:MM and uses 24 hour notation. Finally, the keyword now may be specified to start a shutdown immediately. The message parameter is optional, but highly recommended. This should be enclosed in quotes and give a brief explanation of the reason for the shutdown and when the machine may be back up again.

Under System V, shutdown is accomplished by issuing the command shutdown -y -i1 -g###. Where the -y option informs shutdown to auto-answer all questions with yes; -i1 instructs the system to go to init level 1 (single-user mode); -g### is the grace period to give users before shutting down. The ### symbols should be replace with the number of seconds to wait. Note that there is no message field to use under System V. It is strongly recommended that the system manager use a command such as wall to send a message to all users informing them of the reason for the shutdown and the time when the machine will be available again.

A complete shutdown

A complete shutdown is often done when hardware maintenance is planned for the machine or any other time the machine is to be powered off. On BSD based systems the shutdown command may be specified with the command option of -h to specify that the system should be completely shutdown and the processor halted. As mentioned above, the shutdown command accepts options for the grace time to give before shutdown and a message to send to users. In addition, most systems have a command name halt. In fact, the shutdown -h command usually just invokes the halt command. When you halt the system, all processes are killed, and the sync command is called to write the memory-resident disk buffers to disk. After which, the CPU is halted.

Under System V. based systems the same shutdown command is used as was described above except the init-level is set to zero, as in shutdown -y -i0 -g### . Again, as in BSD based systems, all processes are killed and the sync command is called to write the memory-resident disk buffers to disk before halting the CPU.

The system being rebooted

Systems are rebooted when changes have been made to the operating system and the Unix kernel must be restarted to take advantage of those changes. This is the case when Unix kernel parameters are changed. Often, for many changes software application changes a reboot is not required but may simplify the process of installing a new or updated piece of software.

Under BSD based systems, the shutdown command is again used to accomplish a reboot. The -r option is specified to the shutdown command and causes the system to be shutdown and then automatically rebooted. Similar to the halt, command there is a separate command named reboot which is what the shutdown -r command actually invokes.

Under System V. based systems the same shutdown command is used as was described above except the init-level is set to six, as in shutdown -y -i6 -g###.

As was mentioned previously, it is good policy to issue a wall command before starting the shutdown so you can inform users of the upcoming reboot.

3.5 Building a Kernel

Warning: Understand what you are doing before attempting this on your system. If is very easy to build a kernel that will not boot! Make sure you keep a copy of your original kernel and you understand how to boot an alternate kernel.

What is a Kernel

We discussed this in previous sections such as system startup. However, it is important to understand the role of the kernel. Through the use of system procedure calls the kernel provides controlled access to the underlying hardware on behalf of the user. The kernel is responsible for:

1) Scheduling the CPU

2) Accessing devices on behalf of the user

3) Controlling resource allocation.

4) Creation and deletion of processes.

System routines such as read or write are executed within a user process; however these calls are ultimately dispatched to the kernel to perform the actual read or write of the data. The kernel then returns the data back to the user's process address space. As a user, you may not realize that the kernel is acting on your behalf.

Importance of building a kernel

Conceivably UNIX can (and in some cases does) ship with a generic kernel that can be run across an entire line of systems. However, most vendors offer a tremendously wide range of hardware and software options in which to run UNIX. A generic kernel must include all possible combinations of devices and cannot optimally be sized. A generic kernel usually requires much more memory than a customized kernel.

In addition, in order to add new devices we must generally build a kernel that understands the devices we have on the system. A program named a device driver functions as the intermediary between the kernel and the actual device. Under most versions of UNIX device drivers must be pre-defined within the kernel. Some systems, such as Solaris, now provide dynamic loading and unloading of device drivers which can eliminate this need.

Over the years as systems have evolved the need to build kernels has been reduced. Now many system administrators will use a generic kernel, even though that kernel may require more RAM than one customized. Losing one megabyte of memory of modern day workstation is not that critical. Previously, UNIX might be running in a system with 4 MB or 8MB of RAM, freeing up a single megabyte could be critical.

The NeXT computer system is based on the CMU Mach Operating system and has no kernel configuration files that must be built. The Mach kernel is a very small kernel that provides features to additional kernel modules such as device drivers. Mach was designed to allow dynamic loading of these services on top of the small kernel. As such, the NeXT dynamically configures itself. This model, call micro-kernel, will be the basis for most future variants of UNIX as well as other operating systems (e.g. Windows NT).

Legitimate reasons for building a new kernel

- You are adding a new device and driver to the operating system.
- You are removing a device and driver from the system.
- When you upgrade your system's hardware or change the maximum number of users supported.
- As you add new software modules requiring kernel support.
- Tuning your system to match application requirements.

For example, when you add an application such as a database package you often have to radically increase semaphore resources or shared memory data sizes. Thus, you must reconfigure the kernel to handle that application.

4.0 CONCLUSION

You would have learned about UNIX File system, UNIX processes and how to build a UNIX kernel.

5.0 SUMMARY

You have learned about the UNIX file system and UNIX process. Also, you have learned how to startup, shutdown and reboot a UNIX system and as well as the reasons for building a kernel.

6.0 TUTOR MARKED ASSIGNMENT

1. Why do you have to build a kernel?

7.0 REFERENCES/FUTHER READINGS

1. Step-by-step tutorial on GNU/Linux based on Mandrace Linux by Augustin V. 2003

2. GNU/Linux Command line tools summary by Gareth Anderson, 2006.

3. Centos Essentials by Neil Smyth Techotopia, 2010.

4. Introduction to Linux by Machtelt Garrels.

5. , Practical UNIX & Internet Security by Simson Garfinkel and Gene Spafford, 2nd Ed (Sebastopol, CA: O'Reilly, 1996)

MODULE FOUR WINDOWS

OPERATING SYSTEM

Unit 1: Introduction to windows operating system

Unit 2: Windows 2000 Networking

Unit 3: Windows XP Networking

UNIT ONE

INTRODUCTION TO WINDOWS OPERATING SYSTEM

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
 3.1 History of Operating system
 3.2 Types of windows operating system
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit, you will learn about history and types of windows operating system.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Explain the history of operating system
- Describe the types of operating system

3.0 MAIN CONTENTS

3.1 HISTORY OF WINDOWS OPERATING SYSTEM

The history of Windows dates back to September 1981, when Chase Bishop, a computer scientist, designed the first model of an electronic device and project "Interface Manager" was started. It was announced in November 1983 (after the Apple Lisa, but before the Macintosh) under the name "Windows", but Windows 1.0 was not released until November 1985. The shell of Windows 1.0 was a program known as the MS-DOS Executive. Other supplied programs were Calculator, Calendar, Cardfile, Clipboard viewer, Clock, Control Panel, Notepad, Paint, Reversi, Terminal, and Write. Windows 1.0 did not allow overlapping windows. Instead all windows were tiled. Only dialog boxes could appear over other windows.

Windows 2.0 was released in October 1987 and featured several improvements to the user interface and memory management. Windows 2.0 allowed application windows to overlap each other and also introduced more sophisticated keyboard-shortcuts. It could also make use of expanded memory.

Windows 2.1 was released in two different versions: Windows/386 employed the 386 virtual 8086 mode to multitask several DOS programs, and the paged memory model to emulate expanded memory using available extended memory. Windows/286 (which, despite its name, would run on the 8086) still ran in real mode, but could make use of the high memory area.

The early versions of Windows were often thought of as simply graphical user interfaces, mostly because they ran on top of MS-DOS and used it for file system services. However, even the earliest 16-bit Windows versions already assumed many typical operating system functions; notably, having their own executable file format and providing their own device drivers (timer, graphics, printer, mouse, keyboard and sound) for applications. Unlike MS-DOS, Windows allowed users to execute multiple graphical applications at the same time, through cooperative multitasking. Windows implemented an elaborate, segment-based, software virtual memory scheme, which allowed it to run applications larger than available memory. Code segments and resources were swapped in and thrown away when memory became scarce, and data segments moved in memory when a given application had relinquished processor control.

3.2 TYPES OF WINDOW OPERATING SYSTEM

Windows 3.0 and 3.1

Windows 3.0 (1990) and Windows 3.1 (1992) improved the design, mostly because of virtual memory and loadable virtual device drivers (VxDs) that allowed them to share arbitrary devices between multitasked DOS windows. Also, Windows applications could now run in protected mode (when Windows was running in Standard or 386 Enhanced Mode), which gave them access to several megabytes of memory and removed the obligation to participate in the software virtual memory scheme. They still ran inside the same address space, where the segmented memory provided a degree of protection, and multi-tasked cooperatively. For Windows 3.0, Microsoft also rewrote critical operations from C into assembly.

Windows 95, 98, and Me

Main articles: Windows 95, Windows 98, and Windows Me

Windows 95 was released in August 1995, featuring a new user interface, support for long file names of up to 255 characters, and the ability to automatically detect and configure installed hardware (plug and play). It could natively run 32-bit applications, and featured several technological improvements that increased its stability over Windows 3.1. There were several OEM Service Releases (OSR) of Windows 95, each of which was roughly equivalent to a service pack.

Microsoft's next release was Windows 98 in June 1998. Microsoft released a second version of Windows 98 in May 1999, named Windows 98 Second Edition (often shortened to Windows 98 SE).

In February 2000, Windows 2000 (in the NT family) was released, followed by Windows Me in September 2000 (Me standing for Millennium Edition). Windows Me updated the core from Windows 98, but adopted some aspects of Windows 2000 and removed the "boot in DOS mode" option. It also added a new feature called System Restore, allowing the user to set the computer's settings back to an earlier date.

Windows NT family

The NT family of Windows systems was fashioned and marketed for higher reliability business use. The first release was NT 3.1 (1993), numbered "3.1" to match the consumer Windows version, which was followed by NT 3.5 (1994), NT 3.51 (1995), NT 4.0 (1996), and Windows 2000, which is the last NT-based Windows release that does not include Microsoft Product Activation. Windows NT 4.0 was the first in this line to implement the "Windows 95" user interface (and the first to include Windows 95’s built-in 32-bit runtimes).

Microsoft then moved to combine their consumer and business operating systems with Windows XP that was released in August 2001. It came both in home and professional versions (and later niche market versions for tablet PCs and media centers); they also diverged release schedules for server operating systems. Windows Server 2003, released a year and a half after Windows XP, brought Windows Server up to date with Windows XP. After a lengthy development process, Windows Vista was released toward the end of 2006, and its server counterpart, Windows Server 2008 was released in early 2008. On July 22, 2009, Windows 7 and Windows Server 2008 R2 were released as RTM (release to manufacturing). Windows 7 was released on October 22, 2009.

64-bit operating systems

Windows NT included support for several different platforms before the x86-based personal computer became dominant in the professional world. Versions of NT from 3.1 to 4.0 variously supported PowerPC, DEC Alpha and MIPS R4000, some of which were 64-bit processors, although the operating system treated them as 32-bit processors.

With the introduction of the Intel Itanium architecture (also known as IA-64), Microsoft released new versions of Windows to support it. Itanium versions of Windows XP and Windows Server 2003 were released at the same time as their mainstream x86 (32-bit) counterparts. On April 25, 2005, Microsoft released Windows XP Professional x64 Edition and Windows Server 2003 x64 Editions to support the x86-64 (or x64 in Microsoft terminology) architecture. Microsoft dropped support for the Itanium version of Windows XP in 2005. Windows Vista was the first end-user version of Windows that Microsoft released simultaneously in x86 and x64 editions. Windows Vista does not support the Itanium architecture. The modern 64-bit Windows family comprises AMD64/Intel64 versions of Windows 7 and Windows Server 2008, in both Itanium and x64 editions. Windows Server 2008 R2 drops the 32-bit version, although Windows 7 does not.

Windows CE

Main articles: Windows CE and Windows Phone 7

The latest upcoming version of Windows CE, Windows Embedded Compact 7, displaying a possible UI for what the media player can look like.

Microsoft Windows CE 5.0

Windows CE (officially known as Windows Embedded Compact), is an edition of Windows that runs on minimalistic computers, like satellite navigation systems and some mobile phones. Windows Embedded Compact is based on its own dedicated kernel, dubbed Windows CE kernel. Microsoft licenses Windows CE to OEMs and device makers. The OEMs and device makers can modify and create their own user interfaces and experiences, while Windows CE provides the technical foundation to do so.

Windows CE was used in the Dreamcast along with Sega's own proprietary OS for the console. Windows CE is the core from which Windows Mobile is derived. Microsoft's latest mobile OS, Windows Phone 7, is based on components from both Windows CE 6.0 R3 and the upcoming Windows CE 7.0.

Windows Embedded Compact is not to be confused with Windows XP Embedded or Windows NT 4.0 Embedded, modular editions of Windows based on Windows NT kernel.

4.0 CONCLUSION

You would have learned about history and types of windows operating system.

5.0 SUMMARY

You have learned about the history and evolution of windows operating system.

6.0 TUTOR MARKED ASSIGNMENT

1. State the types of Microsoft windows

7.0 REFERENCES/FUTHER READINGS

1. An Operating system vade mecum by Ralphel A. Finkel, , 2nd edition.
2. A short introduction to operating system by Mark Burgess, 2002.
3. Operating system Handbook by Bob Ducharme, McGraw Hill, 1994.

UNIT TWO

WINDOWS 2000 NETWORKING

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
    3.1 Windows 2000 networking
    3.2 Network Load Balancing Models
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References/Further Readings

1.0 INTRODUCTION

In this unit you will learn about the Cluster service and network load balancing system models.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Explain cluster service of windows 2000
- Explain Network load balancing models

3.0 MAIN CONTECT

3.1 WINDOWS 2000 NETWORKING

Windows 2000 Professional offers users and administrators a greatly improved networking configuration interface when compared to Windows NT and even Windows 98. The menus and option locations are more intuitive, the wizards are easier to walk through, and you no longer have to reboot after changing your network settings. The networking component of Windows 2000 Professional should be a welcome change for those familiar with previous Windows operating systems.

With the move to more Internet-centric computing models, the need for highly available and scalable Internet services is greater than ever before. These services must run 24 hours a day, seven days a week, and be able to quickly and easily increase capacity to meet the often rapid growth in demand for Internet services.

To address these issues, Windows 2000® offers two clustering technologies designed for this purpose: Cluster service, which is intended primarily to provide failover support for critical line-of-business applications such as databases, messaging systems, and file/print services; and Network Load Balancing, which serves to balance incoming IP traffic among multi-host clusters.

Using Windows 2000 Server in a Server- Based Model

It enables extensive file, folder, and printer sharing

Access to resources can be centralized, decentralized, or a combination of both

It provides robust management of software applications

It provides a strong platform for e-mail, Web services, and e-commerce

It enables coordinated backups of network data resources

Sharing of computer resources can be arranged to reflect the work patterns of groups within an organization

Server administration can save time and money when installing software and software upgrades

Windows 2000 Server Services

Windows 2000 Server: A full featured server operating system Supports up to four processors Handles up to 4 GB of RAM Offers a wide range of services and user connectivity options Example Windows 2000 Server Services Handles virtually unlimited user connections (depending on the hardware)

- Active Directory management
- Network management
- Web-based management services
- Network-wide security management
- Network storage management
- Remote network access
- Terminal services
- Distributed file services
- High-speed network connectivity
- Application services and network printer management

3.2 NETWORK LOAD BALANCING MODELS

Network Load Balancing is a clustering technology included in the Microsoft® Windows® 2000 Advanced Server and Datacenter Server operating systems, enhances the scalability and availability of mission-critical, TCP/IP-based services, such as Web, Terminal Services, virtual private networking, and streaming media servers. This component runs within cluster hosts as part of the Windows 2000 operating system and requires no dedicated hardware support. To scale performance, Network Load Balancing distributes IP traffic across multiple cluster hosts. It also ensures high availability by detecting host failures and automatically redistributing traffic to the surviving hosts.

Network Load Balancing provides scalability and high availability to enterprise-wide TCP/IP services, such as Web, Terminal Services, proxy, Virtual Private Networking (VPN), and streaming media services. It is available in two versions of Windows 2000 ie Windows® 2000 Advanced Server & Windows® 2000 Datacenter Server, or bundled free with Application Center 2000.

There are four models for configuring Network Load Balancing – Single Network Adapter (Unicast), Multiple Network Adapters (Unicast), Singe Network Adapter (Multicast), and Multiple Network Adapters (Multicast). Each model has advantages and disadvantages, and suits a particular scenario. This document will provide a comprehensive explanation of each model, and a detailed, step-by-step guide for how to configure them.

Network Load Balancing can be configured using one of four different models. This section describes the models and sets forth the advantages and disadvantages of each, along with possible scenarios. The following section provides step by step examples of how to configure each model.

Important

1. It is worth noting that the most commonly deployed model is Single Network Adapter (Unicast), followed by Multiple Network Adapters (Unicast).

2. The terms virtual IP address (VIP), cluster IP address and primary IP address are often used interchangeably in Microsoft documentation. It is worth noting that the first VIP in an NLB cluster is called the primary IP address (or cluster IP address).

SINGLE NETWORK ADAPTER (UNICAST)

Description

A single network adapter has two or more IP addresses bound to the cluster MAC address: one for cluster traffic (e.g. client access or cluster heartbeats), and another for dedicated traffic (e.g. server management).

Advantages

- Only one network adapter is required; there is no need to install a second adapter.
- Provides the most straightforward configuration, because unicast mode is the default.
- Works with all routers.

Disadvantages

- Ordinary network communication among cluster hosts is not possible.
- Overall network performance may suffer, since both cluster traffic and dedicated traffic use the same network adapter.
- Cluster traffic and dedicated traffic travel through the same network adapter, which may be a security risk e.g. if the cluster traffic is going to the Internet, there is a chance that the dedicated traffic may be "sniffed" from the Internet.

MULTIPLE NETWORK ADAPTERS (UNICAST)

Description

Two or more network adapters with one or more IP addresses bound to one MAC address per network adapter: one network adapter for cluster traffic (e.g. client access or cluster heartbeats), and another network adapter for dedicated traffic (e.g. server management or access to back end resources).

Advantages

- Improved overall performance, since cluster and dedicated traffic travel through different network adapters.
- Permits ordinary network communication among cluster hosts.
- Works with all routers.
- Improved security, since cluster and dedicated traffic travel through different network adapters.

Disadvantages

- Requires a second network adapter per host.

SINGLE NETWORK ADAPTERS (MULTICAST)

Description

A single network adapter has two or more IP addresses bound to two MAC addresses: one for cluster traffic (e.g. client access or cluster heartbeats), and another for dedicated traffic (e.g. server management).

Advantages

- As only one network adapter is required, there is no need to install a second adapter.
- Permits ordinary network communication among cluster hosts.

Disadvantages

- Because there is only one adapter, overall network performance may suffer, since both cluster traffic and dedicated traffic use the same network adapter.

- Some routers may not support the use of a multicast MAC address mapped to a unicast IP address. See the Routers section under Advanced Issues for a solution.
- Cluster traffic and dedicated traffic travel through the same network adapter, which may be a security risk e.g. if the cluster traffic is going to the Internet, there is a chance that the dedicated traffic may be "sniffed" from the Internet.

MULTIPLE NETWORK ADAPTERS (MULTICAST)

Description

Two or more network adapters with one or more IP addresses bound to one or more MAC addresses per network adapter: one network adapter for cluster traffic (e.g. client access or cluster heartbeats), and another network adapter for dedicated traffic (e.g. server management or access to back end resources).

Advantages

- Improved overall performance, since cluster and dedicated traffic travel through different network adapters.
- Permits ordinary network communication among cluster hosts.
- Cluster traffic and dedicated traffic travel through different network adapters, providing better security.

Disadvantages

- Requires a second network adapter.
- Some routers may not support the use of a multicast MAC address mapped to a unicast IP address. See the Routers section under Advanced Issues for a solution.

3.3 PROPERTIES OF WINDOWS 2000

Windows 2000 Server Host and Client System Compatibility:

Windows 2000 Server can communicate with many kinds of other host operating systems IBM, mainframe, Novell, NetWare, UNIX, Banyan, DEC.

Typical operating systems that access Windows 2000 Server as clients are: MS-DOS Windows 3.x, Windows 95/98, Windows NT, Windows 2000, Macintosh, UNIX.

Reliability

Windows 2000 Server is reliable because the kernel operates in privileged mode MS-DOS and Windows 16-bit programs run in the virtual DOS machine so they do not impact 32-bit programs and the operating system, which are running at the same time

Operating System Kernel

An essential set of programs and computer code that allows a computer operating system to control processor, disk, memory, and other functions central to the basic operation of a computer.

Windows 2000 Privileged Mode

A protected memory space allocated for the Windows 2000 kernel that cannot be directly accessed by software applications.

Virtual DOS Machine: In Windows 2000, a process that emulates an MS-DOS window in which to run MS-DOS or 16-bit Windows programs in a designated area of memory

Multitasking and Multithreading

Windows 2000 reliability includes multitasking and multithreading. Multitasking is the capability of a computer to run two or more programs at the same time while Multithreading is the running of several program processes or parts (threads) at the same time. Windows 2000 uses preemptive multitasking.

Fault Tolerance

Fault Tolerance is the techniques that employ hardware and software to provide assurance against equipment failures, computer service interruptions, and data loss.

Windows 2000 Fault Tolerance Features include Recovery from hard disk failures, Recovery from lost data in a file, Recovery from system configuration errors, Protection from power outages, Advanced warning about system and hardware problems, Internet Integration and Electronic Commerce

Windows 2000 Server comes with many Internet-related services. Web server, Intranet and VPN services, Media services, HTML and XML compatibility, FTP Services

New Windows 2000 Server Features Active Directory. A Windows 2000 database of computers, users, shared printers, shared folders, and other network resources, and resource groupings that is used to manage a network and enable users to quickly find a particular resource.

Web-based Enterprise Management (WBEM) standardizes the tools and interfaces used by administrators for a complete picture of the relationship between networks and the devices connected to networks

Hierarchical Storage Management (HSM) A storage management system that enables administrators to establish storage policies, archiving techniques, and disk capacity planning through automated procedures and the coordinated use of different media including tapes, CD-ROMs, hard drives, and zip drives.

Power management Enables portions of a system, such as a monitor, to "sleep" when they are not in use

International language capability Supports more languages and even multiple versions of the same language, such as English used in Britain or English used in the United States

FAT16 Advantages Supported by may small computer systems Low operating overhead Partitions up to 4 GB (in Windows NT or 2000) File sizes up to 2 GB Disadvantages Can become corrupted over time Limited file and folder security and no auditing Does not support long filenames

FAT32 Advantages More robust then FAT16 Enables smaller allocation units than FAT16 (in Windows 2000) Supports volumes up to 32 GB in Windows 2000 Supports long file names Disadvantages Limited file and folder security and no auditing Cannot decrease cluster size Concept: NTFS 4 NTFS 4 is used in Windows NT 4.0 and has the following features Support for long file names Files can be compressed Large file capacity File activity tracking Volume striping and volume extensions

NTFS 5 is used in Windows 2000 and has the following new features Ability to encrypt files No system reboot after creating extended or spanned volumes Ability to reduce drive designations (mount drives) Indexing for fast access Ability to retain shortcuts and other file information when files are transferred between volumes Ability to set disk quotas

CDFS and UDF Windows 2000 supports CDFS and UDF Compact disk file system (CDFS) is a 32-bit file system used on standard capacity CD-ROMs. Universal Disk Format (UDF) is a removable disk formatting standard used for large capacity CD-ROMs and DVD-ROMs. Choosing a File System As a general rule, plan to use NTFS unless you need to use FAT16 or FAT32 for backward compatibility on a system, such as for a dual boot system.

Network servers are used in familiar and expected places. One example of a familiar place is as a Web server. The use of server-based networks is outpacing peer-to-peer networks. Windows 2000 Server offers traditional and new server capabilities, File and printer sharing, C2-compatible security, Web and network communications, Network management capabilities, Active Directory

NTFS is a central feature of Windows 2000 because it offers: Strong security Fault tolerance File compression Indexing Disk quotas and File encryption

Windows 2000 retains backward compatibility with FAT16 and FAT32

4.0 CONCLUSION

You would have learned about the Cluster service and network load balancing system models.

5.0 SUMMARY

You have learned about windows 2000 networking and advance windows 2000 networking. Also, you learned about cluster service and network load balancing models.

ACTIVITY B

1. Discuss the advantages that the Server and Advanced Server editions of Windows 2000 introduced.

6.0 TUTOR MARKED ASSIGNMENT

1. State the network load balancing models

7.0 REFERENCES/FURTHER READINGS

1. An Operating system vade mecum by Ralphel A. Finkel, , 2nd edition.

2. A short introduction to operating system by Mark Burgess, 2002.

3. Operating system Handbook by Bob Ducharme, McGraw Hill, 1994.

UNIT THREE

WINDOWS XP NETWORKING

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content

3.1 Windows XP networking features and enhancements.

4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

With Windows XP, one of Microsoft's primary focuses was to improve the user and administrator experience when networking personal computers. Many of the networking features added or enhanced in Windows XP serves that end.

As more and more home computer users are adding second and third PCs, or bringing laptops home from work, the need to connect these computers together and share resources has increased. Some of the features which include the Networking Setup Wizard, Network Bridging support, and Network Diagnostics, make home networking easier and more convenient.

Connecting these newly networked home computers to the Internet safely is often the next step following creation of the home network. Some of the networking features added to Windows XP makes the PC the best gateway to the Internet for the home network. These features include Internet Connection Sharing, Point-to-Point Protocol over Ethernet support (PPPOE), and Internet Connection Firewall.

Additional networking enhancements and features have been added to improve the telecommuting or remote user experience, improve user to user communication such as instant messaging, and support more networking media choices for today's networks. This paper also discusses those features.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Explain windows XP networking features and enhancements.

3.0 MAIN CONTENT

`

3.1 Windows XP Networking Features and Enhancements

! Internet Connection Firewall (ICF)

When a computer is connected to the Internet or other pathway to the outside world, there is the threat of unauthorized attempts to access the computer and data. Whether the computer connecting to the external network is a standalone computer, or is acting as a gateway for a network behind the computer (see Internet Connection Sharing below), a firewall can guard your home network against the threat of unsafe network traffic while allowing appropriate network traffic to pass.

Windows XP includes the Internet Connection Firewall to be used to protect your computers and home networks connected in such a manner. This software-based firewall is enabled automatically when the Network Setup Wizard (below) is run; setting your firewall up with default settings that will work for most networks. The firewall can also be enabled or disabled manually through the Network Connections folder.

The Internet Connection Firewall monitors connections and traffic that originated from inside the firewall to use in determining what traffic should be allowed from the external network. Traffic originating from the external network will not be allowed through the firewall by default. When hosting services or programs (such as a web server) behind the firewall, these settings can be changed to suit your needs.

! Internet Connection Sharing (ICS) Enhancements

Windows 2000 included ICS to enable sharing of a single Internet connection among all of the computers on a home or small office network. The computer connected to the Internet would have ICS enabled and provide addressing and network address translation services for all of the computers on the network.

Besides providing a DHCP allocator for automatic addressing and a DNS proxy for name resolution, the Windows XP ICS service has also been enhanced to leverage Universal Plug and Play (UPnP) in Windows XP.

ICS participates in the UPnP network as a device hosted on Windows XP, announcing its presence on the network periodically. Windows XP ICS clients use UPnP to detect and locate ICS hosts on the network. Once discovered, UPnP is used to access and control the ICS host.

The system running ICS broadcasts information about the status of the service to the network, including connection status, uptime, and statistics. ICS will also broadcast whenever there is a change in the service's state, such as connection or disconnection.

The client can then use UPnP to perform various actions against ICS. These actions include the ability to connect or disconnect ICS, to list network address translation port mappings, or to create or modify port mappings. This enables clients internal to the network to accept incoming connections.

! Network Bridging Support

When building a network in a home or small office, you may find that a particular network medium works well in one area of the network, but not in another. For example, several computers may be located near telephone jacks enabling them to be connected using HomePNA networking devices. Other computers may be nowhere near a phone jack, requiring selection of another network medium such as wireless. Many medium types are supported by Windows XP, including Ethernet, Phoneline, Wireless and IEEE 1394.

Traditionally, connecting these networks together would require configuring multiple IP address sub-networks and routers to connect the different mediums together. The Network Bridge enables a Windows XP system to act as a bridge for these multiples network mediums. When multiple network connections are added to a Windows XP system and the Network Setup Wizard used to configure the system, the Network Bridge will automatically bridge the networks for you.

This results in a network configuration consisting of a single, easily configured network segment connecting all network mediums. The Windows XP Network Bridge will forward packets on the appropriate segments based on the device address and maintain information about what system is on which physical medium.

- Network Location Awareness and Winsock API Enhancements

Windows XP includes components that detect information about the network the system is attached to. This allows for seamless configuration of the network stack for that location. This information is also made available through a Windows Sockets API, allowing applications to retrieve information about the current network or be notified when the network information changes.

Components in Windows XP also use the network location to provide the appropriate services. For example, the Network Setup wizard will use the location information for multiple adapters in the system to figure out which device is your connection to the Internet. The group policy for ICF is also location aware. ICF will check to see if group policy is set, and then use location information to determine how to apply the policy.

Additional Microsoft extensions to Windows Sockets have been added to Windows XP. This includes ConnectEx() – Used to send a block of data after establishing a connection and TransmitPackets() – Used to transmit in memory and/or file data over a connected socket.

For more information on NLA and the Windows Sockets API, refer to the Windows XP online help and the Windows Platform SDK.

- Wireless LAN Enhancements

Several features and enhancements have been added to Windows XP to improve the experience in deploying Wireless LAN networks. These enhancements are summarized here.

Enhanced Ethernet and Wireless Security (IEEE 802.1X Support) – Previously wireless LAN networking lacked an easy to deploy security solution with a key management system Microsoft and several Wireless LAN and PC vendors worked with the IEEE to define IEEE 802.1X, a standard for port-based network access control. This is applicable to Ethernet or Wireless LANs. Microsoft has implemented IEEE 802.1X support in Windows XP and worked with wireless LAN vendors to support the standard in their access points.

Wireless Zero Configuration – In conjunction with the wireless network adapter, Windows XP can choose from available networks to configure connections to preferred networks without user intervention. Settings for specific networks can be saved and automatically used the next time that network is associated with. In the absence of an infrastructure network, Windows XP can configure the wireless adapter to use ad-hoc networking.

Wireless Roaming Support - Windows 2000 included enhancements for detecting the availability of a network and acting appropriately. These enhancements have been extended and supplemented in Windows XP to support the transitional nature of a wireless network. Features added in Windows XP include re-authentication when necessary and choosing from multiple configuration options based on the network connected to.

! IPv6 Development Stack

Windows XP includes a complete IP version 6 protocol stack. The stack is intended as a development stack to enable and assist developers in making their applications IPv6 capable. This allows for a head start in preparing for the inevitable migration to IPv6 networks. A later version of Windows will include a production level IPv6 protocol stack.

! Internet Protocol over IEEE 1394 (IP/1394)

The ability to network computers and devices on IEEE 1394 using TCP/IP has been added to Windows XP. With this capability, a new network medium is available that is commonly used to connect audio and video equipment. This feature includes enhancements in Windows XP to perform translational bridging of TCP/IP frames between IEEE 1394 and the other IEEE 802 interfaces. To do this, Windows XP uses the Network Bridge already discussed.

4.0 CONCLUSION

You would have learned about the Internet Connection Sharing, Point-to-Point Protocol over Ethernet support (PPPOE), and Internet Connection Firewall.

5.0 SUMMARY

You have learned about the features and enhancement of windows XP

6.0 TUTOR MARKED ASSIGNMENT

1. Briefly explain three windows XP enhancement

7.0 REFERENCES/FURTHER READINGS

1. An Operating system vade mecum by Ralphel A. Finkel, , 2nd edition.

2. A short introduction to operating system by Mark Burgess, 2002.

3. Operating system Handbook by Bob Ducharme, McGraw Hill, 1994.

4. Quick start to Windows XP by Jean Paul van Belle, 2003.

MODULE FIVE

COMPUTER SECURITY AND MANAGEMENT

Unit 1: Computer Security

Unit 2: Computer Security and Management

Unit 3: Fault Tolerant System

Unit 4: Maintaining a Healthy Network Environment

Unit 5: Avoiding Data Loss

UNIT ONE COMPUTER SECURITY

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
3.1 What is computer security?
3.2 Taxonomy of computer security
3.3 Security domains
3.4 Security Models
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about the meaning of computer security, taxonomy of computer security and security domains.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Define computer security
- Discuss the taxonomy of computer security

- Explain security domains

3.0 MAIN CONTENT

3.1 WHAT IS COMPUTER SECRITY

Defining "computer security" is not trivial. The difficulty lies in developing a definition that is broad enough to be valid regardless of the system being described, yet specific enough to describe what security really is. In a generic sense, security is "freedom from risk or danger." In the context of computer science, security is the prevention of, or protection against,

- access to information by unauthorized recipients, and
- intentional but unauthorized destruction or alteration of that information[1]

This can be re-stated: "Security is the ability of a system to protect information and system resources with respect to confidentiality and integrity." Note that the scope of this second definition includes system resources, which include CPUs, disks, and programs, in addition to information.

Computer Security is the protection of computing systems and the data that they store or access

Therefore, "system security" is defined as:

The ongoing and redundant implementation of protections for the confidentiality and integrity of information and system resources so that an unauthorized user has to spend an unacceptable amount of time or money or absorb too much risk in order to defeat it, with the ultimate goal that the system can be trusted with sensitive information.

3.2 TAXONOMY OF COMPUTER SECURITY

Computer security is frequently associated with three core areas, which can be conveniently summarized by the acronym "CIA":

- Confidentiality -- Ensuring that information is not accessed by unauthorized persons.
- Integrity -- Ensuring that information is not altered by unauthorized persons in a way that is not detectable by authorized users.
- Authentication -- Ensuring that users are the persons they claim to be.

A strong security protocol addresses all three of these areas. Take, for example, Netscape's SSL (Secure Sockets Layer) protocol. It has enabled an explosion in ecommerce which is really about trust (or more precisely, about the lack of trust). SSL overcomes the lack of trust between transacting parties by ensuring confidentiality through encryption, integrity through checksums, and authentication via server certificates

Computer security is not restricted to these three broad concepts. Additional ideas that are often considered part of the taxonomy of computer security include:

- Access control -- Ensuring that users access only those resources and services that they are entitled to access and that qualified users are not denied access to services that they legitimately expect to receive
- Non-repudiation -- Ensuring that the originators of messages cannot deny that they in fact sent the messages
- Availability -- Ensuring that a system is operational and functional at a given moment, usually provided through redundancy; loss of availability is often referred to as "denial-of-service"
- Privacy -- Ensuring that individuals maintain the right to control what information is collected about them, how it is used, who has used it, who maintains it, and what purpose it is used for

These additional elements don't neatly integrate into a singular definition. From one perspective, the concepts of privacy, confidentiality, and security are quite distinct and possess different attributes. Privacy is a property of individuals; confidentiality is a property of data; and security is a property assigned to computer hardware and software systems. From a practical perspective, the concepts are interwoven. A system that does not maintain data confidentiality or individual privacy could be theoretically or even mathematically "secure," but it probably wouldn't be wise to deploy anywhere in the real world.

A Functional View

Computer security can also be analyzed by function. It can be broken into five distinct functional areas:

- Risk avoidance -- A security fundamental that starts with questions like:
  Does my organization or business engage in activities that are too risky?
  Do we really need an unrestricted Internet connection?
  Do we really need to computerize that secure business process?
  Should we really standardize on a desktop operating system with no access control intrinsics?

- Deterrence -- Reduces the threat to information assets through fear. Can consist of communication strategies designed to impress potential attackers of the likelihood of getting caught.
- Prevention -- The traditional core of computer security. Consists of implementing safeguards like the tools covered in this book. Absolute prevention is theoretical, since there's a vanishing point where additional preventative measures are no longer cost-effective.
- Detection -- Works best in conjunction with preventative measures. When prevention fails, detection should kick in, preferably while there's still time to prevent damage. Includes log-keeping and auditing activities

- Recovery -- When all else fails, be prepared to pull out backup media and restore from scratch, or cut to backup servers and net connections, or fall back on a disaster recovery facility. Arguably, this function should be attended to before the others

Analyzing security by function can be a valuable part of the security planning process; a strong security policy will address all five areas, starting with recovery. This book, however, is primarily concerned with prevention and detection.

3.3 SECURITY DOMAINS

Computer security is also frequently defined in terms of several interdependent domains that roughly map to specific departments and job titles:

- Physical security -- Controlling the comings and goings of people and materials; protection against the elements and natural disasters
- Operational/procedural security -- Covering everything from managerial policy decisions to reporting hierarchies
- Personnel security -- Hiring employees, background screening, training, security briefings, monitoring, and handling departures
- System security -- User access and authentication controls, assignment of privilege, maintaining file and filesystem integrity, backups, monitoring processes, log-keeping, and auditing
- Network security -- Protecting network and telecommunications equipment, protecting network servers and transmissions, combating eavesdropping, controlling access from untrusted networks, firewalls, and detecting intrusions

This text is solely concerned with the latter two. System and network security are difficult, if not impossible, to separate in a UNIX system. Nearly every UNIX distribution in the past fifteen years has included a TCP/IP protocol implementation as well as numerous network services such as FTP, Telnet, DNS, and, more recently, HTTP.

3.4 SECURITY MODELS

After implementing security for the network's physical components, the administrator needs to ensure that the network resources will be safe from both unauthorized access and accidental or deliberate damage. Policies for assigning permissions and rights to network resources are at the heart of securing the network.

Two security models have evolved for keeping data and hardware resources safe:

- Password-protected shares
- Access permissions

These models are also called "share-level security" (for password-protected shares) and "user-level security" (for access permissions).

Password-Protected Shares

Implementing password-protected shares requires assigning a password to each shared resource. Access to the shared resource is granted when a user enters the correct password.

In many systems, resources can be shared with different types of permissions. To illustrate, we use Windows 95 and 98 as examples. For these operating systems, "Establishing Network Shares and Accounts," directories can be shared as Read Only, Full, or Depends On Password.

- Read Only If a share is set up as Read Only, users who know the password have Read access to the files in that directory. They can view the documents, copy them to their machines, and print them, but they cannot change the original documents.
- Full With Full access, users who know the password have complete access to the files in that directory. In other words, they can view, modify, add, and delete the shared directory's files.
- Depends On Password Depends On Password involves setting up a share that uses two levels of passwords: Read access and Full access. Users who know the Read access password have Read access, and users who know the Full access password have Full access.

The password-protected share system is a simple security method that allows anyone who knows the password to obtain access to that particular resource.

Access Permissions

Access-permission security involves assigning certain rights on a user-by-user basis. A user types a password when logging on to the network. The server validates this user name and password combination and uses it to grant or deny access to shared resources by checking access to the resource against a user- access database on the server.

Access-permission security provides a higher level of control over access rights. It is much easier for one person to give another person a printer password, as in share-level security. It is less likely for that person to give away a personal password.

Because user-level security is more extensive and can determine various levels of security, it is usually the preferred model in larger organizations.

Resource Security

After the user has been authenticated and allowed on the network, the security system gives the user access to the appropriate resources.

Users have passwords, but resources have permissions. In a sense, a security fence guards each resource. The fence has several gates through which users can pass to access the resource. Certain gates allow users to do more to the resource than other gates. Certain gates, in other words, allow the user more privileges with the resource.

The administrator determines which users should be allowed through which gates. One gate grants the user full access to or full control of a resource. Another gate grants the user read-only access.

As shown in Figure 4.1, each shared resource or file is stored with a list of users or groups and their associated permissions (gates).

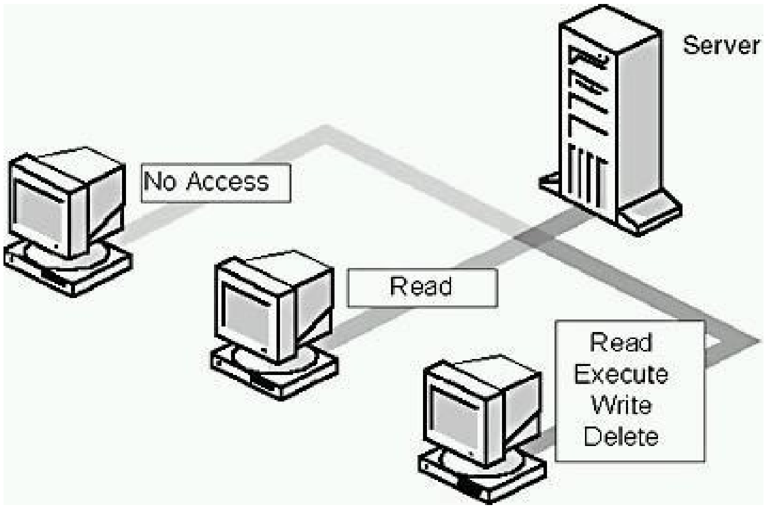


Figure 4.1: Permissions control the type of access to a resource

NOTE: Different network operating systems (NOSs) give different names to these permissions. The following table shows some of the typical permissions that can be set on Windows NT Server directories.

Table 4.1 below contains common access permissions assigned to shared directories or files.

Table 4.1: Windows NT Server Permissions

Permission	Functionality
Read	Reads and copies files in the shared directory.
Execute	Runs (executes) the files in the directory.
Write	Creates new files in the directory.
Delete	Deletes files in the directory.
No Access	Prevents the user from gaining access to directories, files, or resources.

A Practical Definition

In the spirit of practicality, definition promulgated by Simson Garfinkel and Gene Spafford in Practical UNIX & Internet Security" A computer is secure if you can depend on it and its software to behave as you expect." In essence, a computer is secure if you can trust it. Data entered today will still be there tomorrow in unaltered form. If you made services x, y, and z available yesterday, they are still available today.

These practical definitions circumvent an obvious element: a secure system should be hard for unauthorized persons to break into -- i.e., the value of the work necessary for an unauthorized person to break in should exceed the value of the protected data. Increasing attacker workload and the risks of detection are critical elements of computer security.

4.0 CONCLUSION

You would have learned about the meaning of computer security, taxonomy of computer security and security domains.

You have learned about the meaning of computer security, the areas majorly addressed by computer security as well as security domains.

ACTIVITY B

1. Discuss two types of security models that can be use for keeping data and hardware resources safe.

6.0 TUTOR MARKED ASSIGNMENT

State the three major areas addressed by computer security

8.0 REFERENCES/FUTHER READINGS

1. Handbook of Information Security management by Micki Krause and Harold F. Tipton. Publisher: CRC press LLC, ISBN: 0849399475.

2. The protection of Information Security Management by Sean Boran

3. IT Security Cookbook by sean Boran

4. A Structured Approach to Computer Security by Olovsson. Technical Report no 122, 1992.

UNIT TWO

COMPUTER SECURITY AND MANAGEMENT

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
3.1 Computer security concept
3.2 Importance of security
3.3 Threats to data
3.4 Security tools
3.5 Security Policies
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

In this unit you will learn about computer security concept, importance of security, security tools and security policies.

2.0 OBJECTIVES

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

- Explain the consequences for security violations
- Explain the importance of security
- Explain threats to data

3.0 MAIN CONTENT

3.1 IMPORTANCE OF SECURITY

The Internet has undoubtedly become the largest public data network, enabling and facilitating both personal and business communications worldwide. The volume of traffic moving over the Internet, as well as corporate networks, is expanding exponentially every day.

More and more communication is taking place via e-mail; mobile workers, telecommuters, and branch offices are using the Internet to remotely connect to their corporate networks; and commercial transactions completed over the Internet, via the World Wide Web, now account for large portions of corporate revenue.

While the Internet has transformed and greatly improved the way we do business, this vast network and its associated technologies have opened the door to an increasing number of security threats from which corporations must protect themselves. Although network attacks are presumably more serious when they are inflicted upon businesses that store sensitive data, such as personal medical or financial records, the consequences of attacks on any entity range from mildly inconvenient to completely debilitating—important data can be lost, privacy can be violated, and several hours, or even days, of network downtime can ensue.

Despite the costly risks of potential security breaches, the Internet can be one of the safest means by which to conduct business. For example, giving credit card information to a telemarketer over the phone or a waiter in a restaurant can be more risky than submitting the information via a Web site, because electronic commerce transactions are usually protected by security technology. Waiters and telemarketers are not always monitored or trustworthy. Yet the fear of security problems can be just as harmful to businesses as actual security breaches. General fear and suspicion of computers still exists and with that comes a distrust of the Internet. This distrust can limit the business opportunities for companies, especially those that are completely Web based. Thus, companies must enact security policies and instate safeguards that not only are effective, but are also perceived as effective.
Organizations must be able to adequately communicate how they plan to protect their customers.

In addition to protecting their customers, corporations must protect their employees and partners from security breaches. The Internet, intranets, and extranets enable fast and effective communication between employees and partners. However, such communication and efficiency can of course be impeded by the effects of a network attack. An attack may directly cause several hours of downtime for employees, and networks must be taken down in order for damage to be repaired or data to be restored. Clearly, loss of precious time and data can greatly impact employee efficiency and morale.
Legislation is another force that drives the need for network security. Governments recognize both the importance of the Internet and the fact that substantial portions of the world's economic output are dependent on it. However, they also recognize that opening up the world's economic infrastructure to abuse by criminals could cause major economic damage. National governments are therefore developing laws intended to regulate the vast flow of electronic information.

3.2 CONSEQUENCES OF SECURITY VIOLATION

- Risk to security and integrity of personal or confidential information e.g. identity theft, data corruption or destruction, unavailability of critical information in an emergency, etc.
- Loss of valuable business information

- Loss of employee and public trust, embarrassment, bad publicity, media coverage, news reports
- Costly reporting requirements in the case of a compromise of certain types of personal, financial and health information
- Internal disciplinary action(s) up to and including termination of employment, as well as possible penalties, prosecution and the potential for sanctions / lawsuits

3.3 THREATS TO DATA

As with any type of crime, the threats to the privacy and integrity of data come from a very small minority of vandals. However, while one car thief can steal only one car at a time, a single hacker working from a basic computer can generate damage to a large number of computer networks that wreaks havoc around the world.

Perhaps even more worrisome is the fact that the threats can come from people we know. In fact, most network security experts claim that the majority of network attacks are initiated by employees who work inside the corporations where breaches have occurred.

Employees, through mischief, malice, or mistake, often manage to damage their own companies' networks and destroy data. Furthermore, with the recent pervasiveness of remote connectivity technologies, businesses are expanding to include larger numbers of telecommuters, branch offices, and business partners. These remote employees and partners pose the same threats as internal employees, as well as the risk of security breaches if their remote networking assets are not properly secured and monitored.

Whether you want to secure a car, a home, a nation, or a computer network, a general knowledge of who the potential enemies are and how they work is essential.

The Enemies

- Hackers

This generic and often over-romanticized term applies to computer enthusiasts who take pleasure in gaining access to other people's computers or networks. Many hackers are content with simply breaking in and leaving their "footprints," which are joke applications or messages on computer desktops. Other hackers, often referred to as "crackers," are more malicious, crashing entire computer systems, stealing or damaging confidential data, defacing Web pages, and ultimately disrupting business. Some amateur hackers merely locate hacking tools online and deploy them without much understanding of how they work or their effects.

- Unaware Staff

As employees focus on their specific job duties, they often overlook standard network security rules.

For example, they might choose passwords that are very simple to remember so that they can log on to their networks easily.

However, such passwords might be easy to guess or crack by hackers using simple common sense or a widely available password cracking software utility.

Employees can unconsciously cause other security breaches including the accidental contraction and spreading of computer viruses. One of the most common ways to pick up a virus is from a floppy disk or by downloading files from the Internet. Employees who transport data via floppy disks can unwillingly infect their corporate networks with viruses they picked up from computers in copy centers or libraries. They might not even know if viruses are resident on their PCs.

Corporations also face the risk of infection when employees download files, such as PowerPoint presentations, from the Internet. Surprisingly, companies must also be wary of human error. Employees, whether they are computer novices or computer savvy, can make such mistakes as erroneously installing virus protection software or accidentally overlooking warnings regarding security threats.

- Disgruntled Staff

Far more unsettling than the prospect of employee error causing harm to a network is the potential for an angry or vengeful staff member to inflict damage. Angry employees, often those who have been reprimanded, fired, or laid off, might vindictively infect their corporate networks with viruses or intentionally delete crucial files. This group is especially dangerous because it is usually far more aware of the network, the value of the information within it, where high-priority information is located, and the safeguards protecting it.

- Snoops

Whether content or disgruntled, some employees might also be curious or mischievous. Employees known as “snoops” partake in corporate espionage, gaining unauthorized access to confidential data in order to provide competitors with otherwise inaccessible information.

Others are simply satisfying their personal curiosities by accessing private information, such as financial data, a romantic e-mail correspondence between co-workers, or the salary of a colleague. Some of these activities might be relatively harmless, but others, such as previewing private financial, patient, or human resources data, are far more serious, can be damaging to reputations, and can cause financial liability for a company.

What the Enemies Do

- Viruses

Viruses are the most widely known security threats, because they often garner extensive press coverage.

Viruses are computer programs that are written by devious programmers and are designed to replicate themselves and infect computers when triggered by a specific event. For example, viruses called macro viruses attach themselves to files that contain macro instructions (routines that can be repeated automatically, such as mail merges) and are then activated every time the macro runs.

The effects of some viruses are relatively benign and cause annoying interruptions such as displaying a comical message when striking a certain letter on the keyboard.
Other viruses are more destructive and cause such problems as deleting files from a hard drive or slowing down a system.

A network can be infected by a virus only if the virus enters the network through an outside source—most often through an infected floppy disk or a file downloaded from the Internet. When one computer on the network becomes infected, the other computers on the network are highly susceptible to contracting the virus.

- Trojan Horse Programs

Trojan horse programs, or trojans, are delivery vehicles for destructive code. Trojans appear to be harmless or useful software programs, such as computer games, but they are actually enemies in disguise. Trojans can delete data, mail copies of themselves to e-mail address lists, and open up computers to additional attacks. Trojans can be contracted only by copying the trojan horse program to a system, via a disk, downloading from the Internet, or opening an e-mail attachment. Neither trojans nor viruses can be spread through an e-mail message itself—they are spread only through e-mail attachments.

- Vandals

Web sites have come alive through the development of such software applications as ActiveX and Java Applets.
These devices enable animation and other special effects to run, making Web sites more attractive and interactive.
However, the ease with which these applications can be downloaded and run has provided a new vehicle for inflicting damage. A vandal is a software application or applet that causes destruction of varying degrees. A vandal can destroy just a single file or a major portion of a computer system.

- Attacks

Innumerable types of network attacks have been documented, and they are commonly classified in three general categories: reconnaissance attacks, access attacks, and denial of service (DoS) attacks.

• Reconnaissance attacks are essentially information gathering activities by which hackers collect data that is used to later compromise networks.

Usually, software tools, such as sniffers and scanners, are used to map out network resources and exploit potential weaknesses in the targeted networks, hosts, and applications. For example, software exists that is specifically designed to crack passwords. Such software was created for network administrators to assist employees who have forgotten their passwords or to determine the passwords of employees who have left the company without telling anyone what their passwords were. Placed in the wrong hands, however, this software can become a very dangerous weapon.

• Access attacks are conducted to exploit vulnerabilities in such network areas as authentication services and File
Transfer Protocol (FTP) functionality in order to gain entry to e-mail accounts, databases, and other confidential information.

• DoS attacks prevent access to part or all of a computer system. They are usually achieved by sending large amounts of jumbled or otherwise unmanageable data to a machine that is connected to a corporate network or the Internet, blocking legitimate traffic from getting through. Even more malicious is a Distributed Denial of Service attack (DDoS) in which the attacker compromises multiple machines or hosts.

- Data Interception

Data transmitted via any type of network can be subject to interception by unauthorized parties. The perpetrators might eavesdrop on communications or even alter the data packets being transmitted. Perpetrators can use various methods to intercept the data. IP spoofing, for example, entails posing as an authorized party in the data transmission by using the Internet Protocol (IP) address of one of the data recipients.

- Social Engineering

Social engineering is the increasingly prevalent act of obtaining confidential network security information through non-technical means. For example, a social engineer might pose as a technical support representative and make calls to employees to gather password information. Other examples of social engineering include bribing a coworker to gain access to a server or searching a colleague's office to find a password that has been written in a hidden spot.

- Spam

Spam is the commonly used term for unsolicited electronic mail or the action of broadcasting unsolicited advertising messages via e-mail. Spam is usually harmless, but it can be a nuisance, taking up the recipient's time and storage space.

3.4 SECURITY TOOLS

After the potential sources of threats and the types of damage that can occur have been identified, putting the proper security policies and safeguards in place becomes much easier. Organizations have an extensive choice of technologies, ranging from anti-virus software packages to dedicated network security hardware, such as firewalls and intrusion detection systems, to provide protection for all areas of the network.

SECURITY SOFTWARE

Most organizations use several types of network-based and host-based security software to detect malicious activity, protect systems and data, and support incident response efforts. Accordingly, security software is a major source of computer security log data. Common types of network-based and host-based security software include the following:

Anti-virus Packages. Virus protection software is packaged with most computers and can counter most virus threats if the software is regularly updated and correctly maintained.
The anti-virus industry relies on a vast network of users to provide early warnings of new viruses, so that antidotes can be developed and distributed quickly. With thousands of new viruses being generated every month, it is essential that the virus database is kept up to date. The virus database is the record held by the anti-virus package that helps it to identify known viruses when they attempt to strike. Reputable anti-virus software vendors will publish the latest antidotes on their Web sites, and the software can prompt users to periodically collect new data

Antimalware Software. The most common form of antimalware software is antivirus software, which typically records all instances of detected malware, file and system disinfection attempts, and file quarantines.[3] Additionally, antivirus software might also record when malware scans were performed and when antivirus signature or software updates occurred. Antispyware software and other types of antimalware software (e.g., rootkit detectors) are also common sources of security information.

Intrusion Detection and Intrusion Prevention Systems. Intrusion detection and intrusion prevention systems record detailed information on suspicious behavior and detected attacks, as well as any actions intrusion prevention systems performed to stop malicious activity in progress. Some intrusion detection systems, such as file integrity checking software, run periodically instead of continuously, so they generate log entries in batches instead of on an ongoing basis.

Remote Access Software. Remote access is often granted and secured through virtual private networking (VPN). VPN systems typically log successful and failed login attempts, as well as the dates and times each user connected and disconnected, and the amount of data sent and received in each user session. VPN systems that support granular access control, such as many Secure Sockets Layer (SSL) VPNs, may log detailed information about the use of resources.

Web Proxies. Web proxies are intermediate hosts through which Web sites are accessed. Web proxies make Web page requests on behalf of users, and they cache copies of retrieved Web pages to make additional accesses to those pages more efficient. Web proxies can also be used to restrict Web access and to add a layer of protection between

Web clients and Web servers. Web proxies often keep a record of all URLs accessed through them.

Vulnerability Management Software. Vulnerability management software, which includes patch management software and vulnerability assessment software, typically logs the patch installation history and vulnerability status of each host, which includes known vulnerabilities and missing software updates.

Vulnerability management software may also record additional information about hosts' configurations. Vulnerability management software typically runs occasionally, not continuously, and is likely to generate large batches of log entries.

Authentication Servers. Authentication servers, including directory servers and single sign-on servers, typically log each authentication attempt, including its origin, username, success or failure, and date and time.

Routers. Routers may be configured to permit or block certain types of network traffic based on a policy. Routers that block traffic are usually configured to log only the most basic characteristics of blocked activity.

Firewalls. Like routers, firewalls permit or block activity based on a policy; however, firewalls use much more sophisticated methods to examine network traffic. Firewalls can also track the state of network traffic and perform content inspection. Firewalls tend to have more complex policies and generate more detailed logs of activity than routers.

Network Quarantine Servers. Some organizations check each remote host's security posture before allowing it to join the network. This is often done through a network quarantine server and agents placed on each host. Hosts that do not respond to the server's checks or that fail the checks are quarantined on a separate virtual local area network (VLAN) segment. Network quarantine servers log information about the status of checks, including which hosts were quarantined and for what reasons.

3.5 SECURITY POLICIES

Network security policy should stipulate that all computers on the network are kept up to date and, ideally, are all protected by the same anti-virus package—if only to keep maintenance and update costs to a minimum. It is also essential to update the software itself on a regular basis. Virus authors often make getting past the anti-virus packages their first priority.

When setting up a network, whether it is a local area network (LAN), virtual LAN (VLAN), or wide area network (WAN), it is important to initially set the fundamental security policies. Security policies are rules that are electronically programmed and stored within security equipment to control such areas as access privileges. Of course, security policies are also written or verbal regulations by which an organization operates.

In addition, companies must decide who is responsible for enforcing and managing these policies and determine how employees are informed of the rules and watch guards.

Security Policy, Device, and Multi-device Management functions as a central security control room where security personnel monitor building or campus security, initiate patrols, and activate alarms.

What are the policies?

The policies that are implemented should control who has access to which areas of the network and how unauthorized users are going to be prevented from entering restricted areas. For example, generally only members of the human resources department should have access to employee salary histories. Passwords usually prevent employees from entering restricted areas, but only if the passwords remain private. Written policies as basic as to warn employees against posting their passwords in work areas can often preempt security breaches. Customers or suppliers with access to certain parts of the network must be adequately regulated by the policies as well.

Who will enforce and manage the policies?

The individual or group of people who police and maintain the network and its security must have access to every area of the network. Therefore, the security policy management function should be assigned to people who are extremely trustworthy and have the technical competence required. As noted earlier, the majority of network security breaches come from within, so this person or group must not be a potential threat. Once assigned, network managers may take advantage of sophisticated software tools that can help define, distribute, enforce, and audit security policies through browser-based interfaces.

How will you communicate the policies?

Policies are essentially useless if all of the involved parties do not know and understand them. It is vital to have effective mechanisms in place for communicating the existing policies, policy changes, new policies, and security alerts regarding impending viruses or attacks.

Identity

Once your policies are set, identity methods and technologies must be employed to help positively authenticate and verify users and their access privileges.

Access Control Servers function like door access cards and the gatekeeper that oversees site security, providing centralized authorization, authentication and accounting (AAA) for traffic and users.

Passwords

Making sure that certain areas of the network are “password protected”—only accessible by those with particular passwords—is the simplest and most common way to ensure that only those who have permission can enter a particular part of the network. In the physical security analogy above, passwords are analogous to badge access cards.

However, the most powerful network security infrastructures are virtually ineffective if people do not protect their passwords. Many users choose easily remembered numbers or words as passwords, such as birthdays, phone numbers, or pets’ names, and others never change their passwords and are not very careful about keeping them secret. The golden rules, or policies, for passwords are:

- Change passwords regularly
- Make passwords as meaningless as possible
- Never divulge passwords to anyone until leaving the company

In the future, some passwords may be replaced by biometrics, which is technology that identifies users based on physical characteristics, such as fingerprints, eye prints, or voice prints.

Digital Certificates

Digital certificates or public key certificates are the electronic equivalents of driver's licenses or passports, and are issued by designated Certificate Authorities (CAs).

Digital certificates are most often used for identification when establishing secure tunnels through the Internet, such as in virtual private networking (VPN).

Access Control

Before a user gains access to the network with his password, the network must evaluate if the password is valid. Access control servers validate the user's identity and determine which areas or information the user can access based on stored user profiles. In the physical security analogy, access control servers are equivalent to the gatekeeper who oversees the use of the access card.

Access Control Lists and Firewalls are analogous to door locks on building perimeters that allow only authorized users (those with keys or badges) access in or out.

Firewalls

A firewall is a hardware or software solution implemented within the network infrastructure to enforce an organization's security policies by restricting access to specific network resources. In the physical security analogy, a firewall is the equivalent to a door lock on a perimeter door or on a door to a room inside of the building—it permits only authorized users, such as those with a key or access card, to enter. Firewall technology is even available in versions suitable for home use. The firewall creates a protective layer between the network and the outside world. In effect, the firewall replicates the network at the point of entry so that it can receive and transmit authorized data without significant delay.

However, it has built-in filters that can disallow unauthorized or potentially dangerous material from entering the real system. It also logs an attempted intrusion and reports it to the network administrators.

Encryption

Encryption technology ensures that messages cannot be intercepted or read by anyone other than the authorized recipient. Encryption is usually deployed to protect data that is transported over a public network and uses advanced mathematical algorithms to "scramble" messages and their attachments. Several types of encryption algorithms exist, but some are more secure than others. Encryption provides the security necessary to sustain the increasingly popular VPN technology. VPNs are private connections, or tunnels, over public networks such as the Internet. They are deployed to connect telecommuters, mobile workers, branch offices, and business partners to corporate networks or each other.

All VPN hardware and software devices support advanced encryption technology to provide the utmost protection for the data that they transport.

Virtual Private Networks (VPNs) are analogous to armored cars that carry precious cargo to an assigned drop-off point to ensure secure and confidential passage.

Intrusion Detection

Organizations continue to deploy firewalls as their central gatekeepers to prevent unauthorized users from entering their networks. However, network security is in many ways similar to physical security in that no one technology serves all needs—rather, a layered defense provides the best results. Organizations are increasingly looking to additional security technologies to counter risk and vulnerability that firewalls alone cannot address. A network-based intrusion detection system (IDS) provides around-the-clock network surveillance. An IDS analyzes packet data streams within a network, searching for unauthorized activity, such as attacks by hackers, and enabling users to respond to security breaches before systems are compromised. When unauthorized activity is detected, the IDS can send alarms to a management console with details of the activity and can often order other systems, such as routers, to cut off the unauthorized sessions. In the physical analogy, an IDS is equivalent to a video camera and motion sensor; detecting unauthorized or suspicious activity and working with automated response systems, such as watch guards, to stop the activity.

4.0 CONCLUSION

You would have learned about computer security concept, importance of security, security tools and security policies.

5.0 SUMMARY

You have learned about the concept of computer security, the importance of security, the security tools and security policies.

ACTIVITY B

1.0 State three Security tools used in computer security

6.0 TUTOR MARKED ASSIGNMENT

1. State 3 threats to data

7.0 REFERENCES/FUTHER READINGS

1. Handbook of Information Security management by Micki Krause and Harold F. Tipton. Publisher: CRC press LLC, ISBN: 0849399475.

2. The protection of Information Security Management by Sean Boran

3. IT Security Cookbook by sean Boran

4. A Structured Approach to Computer Security by Olovsson. Technical Report no 122, 1992.

5. Computer Security Management by Donn B. Parker, 1981.

UNIT THREE

FAULT TOLERANT SYSTEM

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
3.1 Definition
3.2 RAID
3.3 Implementing Fault tolerance
3.4 Disaster Recovery
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUTION

In this unit you will learn about Fault tolerant systems, Redundancy array of Independent Disk and Disaster Recovery.

2.0 OBJECTIVES

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

- Explain the various types of RAID
- Implement fault tolerance
- Explain disaster recovery procedures

3.0 MAIN CONTENT

3.1 Definition

Fault-Tolerant Systems

Fault-tolerant systems protect data by duplicating data or placing data in different physical sources, such as different partitions or different disks. Data redundancy allows access to data even if part of the data system fails. Redundancy is a prominent feature common to most fault-tolerant systems.

Fault-tolerant systems should never be used as replacements for regular backup of servers and local hard disks. A carefully planned backup strategy is the best insurance policy for recovering lost or damaged data.

Fault-tolerant systems offer these alternatives for data redundancy:

- Disk striping
- Disk mirroring
- Sector sparing
- Mirrored drive arrays
- Clustering

3.2 Redundant Array of Independent Disks (RAID)

Fault-tolerance options are standardized and categorized into levels. These levels are known as redundant array of independent disks (RAID), formerly known as redundant array of inexpensive disks. The levels offer various combinations of performance, reliability, and cost.

Level 0—Disk Striping

Disk striping divides data into 64K blocks and spreads it equally in a fixed rate and order among all disks in an array. However, disk striping does not provide any fault tolerance because there is no data redundancy. If any partition in the disk array fails, all data is lost.

A stripe set combines multiple areas of unformatted free space into one large logical drive, distributing data storage across all drives simultaneously. In Windows NT, a stripe set requires at least two physical drives and can use up to 32 physical drives. Stripe sets can combine areas on different types of drives, such as small computer system interface (SCSI), enhanced small device interface (ESDI), and integrated device electronics (IDE) drives.

Figure 4.2 shows three hard disks being used to create a stripe set. In this case, the data consists of 192 K of data. The first 64 K of data is written to a stripe on disk 1, the second 64 K is written to a stripe on disk 2, and the third 64 K is written to the stripe on disk 3.

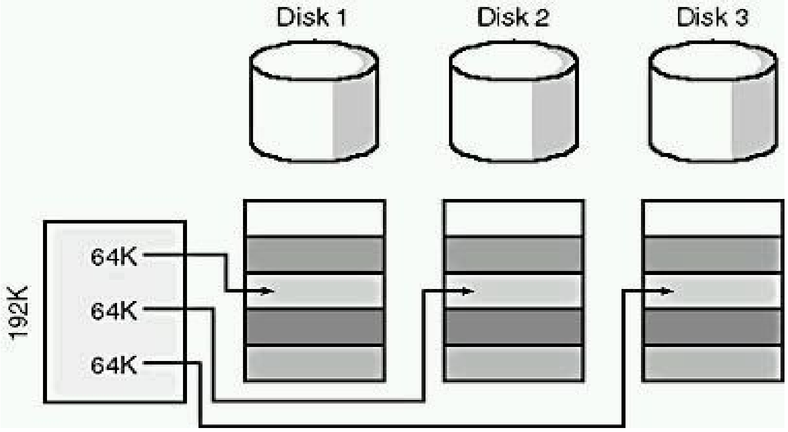


Figure 4.2 Disk striping combines areas on multiple drives

Disk striping has several advantages: it makes one large partition out of several small partitions, which offers better use of disk space; and multiple disk controllers will result in better performance.

Level 1—Disk Mirroring

Disk mirroring actually duplicates a partition and moves the duplication onto another physical disk. There are always two copies of the data, with each copy on a separate disk. Any partition can be mirrored. This strategy is the simplest way to protect a single disk against failure. Disk mirroring can be considered a form of continual backup because it maintains a fully redundant copy of a partition on another disk.

Duplexing

Disk duplexing, as shown in Figure 4.3, consists of a mirrored pair of disks with an additional disk controller on the second drive. This reduces channel traffic and potentially improves performance. Duplexing is intended to protect against disk controller failures as well as media failures.

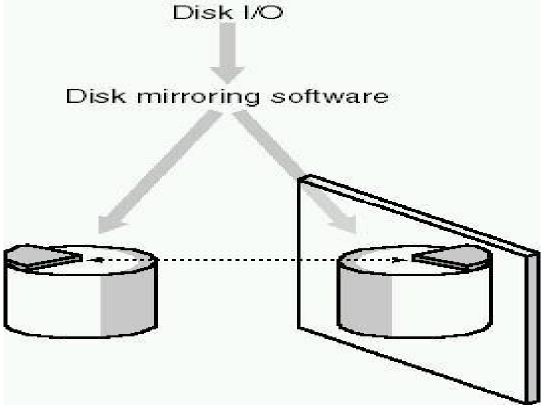


Figure 4.3: Disk mirroring duplicates a partition on another physical disk

Level 2—Disk Striping with ECC

When a block of data is written, the block is broken up and distributed (interleaved) across all data drives. Error-correction code (ECC) requires a larger amount of disk space than parity-checking methods, discussed under Level 3. Although this method offers marginal improvement in disk utilization, it compares poorly with level 5, discussed later.

Level 3—ECC Stored As Parity

Disk striping with ECC stored as parity is similar to level 2. The term parity refers to an error-checking procedure in which the number of 1s must always be the same—either odd or even—for each group of bits transmitted without error. In this strategy, the ECC method is replaced with a parity-checking scheme that requires only one disk to store parity data.

Level 4—Disk Striping with Large Blocks

This strategy moves away from data interleaving by writing complete blocks of data to each disk in the array. The process is still known as disk striping, but is done with large blocks. A separate check disk is used to store parity information. Each time a write operation occurs, the associated parity information must be read from the check disk and modified. Because of this overhead, the block-interleaving method works better for large block operations than for transaction-based processing.

Level 5—Striping with Parity

Striping with parity is currently the most popular approach to fault-tolerant design. It supports from a minimum of three to a maximum of 32 drives and writes the parity information across all the disks in the array (the entire stripe set). The data and parity information are arranged so that the two are always on different disks.

A parity stripe block exists for each stripe (row) across the disk. The parity stripe block is used to reconstruct data for a failed physical disk. If a single drive fails, enough information is spread across the remaining disks to allow the data to be completely reconstructed.

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, and RAID 5 distributes parity evenly across all disks.

Level 10—Mirrored Drive Arrays

RAID level 10 mirrors data across two identical RAID 0 drive arrays.

Sector Sparing

The Windows NT Server operating system offers an additional fault-tolerant feature called "sector sparing," also known as "hot fixing." The three steps of sector sparing are shown in Figure 4.4.

This feature automatically adds sector-recovery capabilities to the file system while the computer is running.

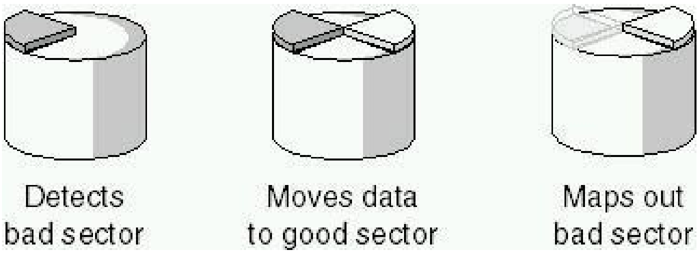


Figure 4.4: Sector sparing or hot-fixing steps

If bad sectors are found during disk I/O (input/output), the fault-tolerant driver will attempt to move the data to a good sector and map out the bad sector. If the mapping is successful, the file system is not alerted. It is possible for SCSI devices to perform sector sparing, but ESDI and IDE devices cannot. Some network operating systems, such as Windows NT Server, have a utility that notifies the administrator of all sector failures and of the potential for data loss if the redundant copy also fails.

Microsoft Clustering

Microsoft Clustering is Microsoft's implementation of server clustering. The term "clustering" refers to a group of independent systems that work together as a single system. Fault tolerance is built into the clustering technology. Should a system within the cluster fail, the cluster software will disperse the work from the failed system to the remaining systems in the cluster. Clustering is not intended to replace current implementations of fault-tolerant systems, although it does provide an excellent enhancement.

3.3 Implementing Fault Tolerance

Most advanced network operating systems offer a utility for implementing fault tolerance. In Windows NT Server, for example, the Disk Administrator program is used to configure Windows NT Server fault tolerance. The graphical interface of Disk Administrator makes it easy to configure and manage disk partitioning and fault tolerant options. If you move the disk to a different controller or change its ID, Windows NT will still recognize it as the original disk. Disk Administrator is used to create various disk configurations, including:

- Stripe sets with parity, which accumulates multiple disk areas into one large partition, distributing data storage across all drives simultaneously, adding fault tolerant parity information.
- Mirror sets, which make a duplicate of one partition and place it onto a separate physical disk.

- Volume sets, which accumulate multiple disk areas into one large partition, filling the areas in sequence.
- Stripe sets, which accumulate multiple disk areas into one large partition, distributing data storage across all drives simultaneously.

Optical Drives and Disks

The term "optical drive" is a generic term that is applied to several devices. In optical technology, data is stored on a rigid disk by altering the disk's surface with a laser beam.

The use of optical drives and discs is becoming increasingly popular. As the technology evolves from the original read-only and read-write CD-ROMs to the new DVD technologies, these devices are being used more and more to store large amounts of retrievable data. Optical-drive manufacturers provide a large array of storage configurations that are either network-ready or can be used with a network server. They are an excellent choice for permanent backup. Several variations of this technology exist.

CD-ROM Technology

Compact discs (CD-ROMs) are the most common form of optical data storage. CD-ROMs, for the most part, only allow information to be read. The advantages of using CDs for storage are many. The ISO 9660 specification defines an international format standard for CD-ROM. Their storage capacity is high—up to 650 MB of data on a 4.72-inch disc. They are portable and replaceable, and because data on a CD-ROM cannot be changed (it is read-only), files cannot be accidentally erased. Standard recording formats and inexpensive readers make CDs ideal for data storage. CD-ROMs are also available in a multisession format called "CD-recordable" (CD-R). This media can now be used for incremental updates and inexpensive duplication. CD-ROMs are also offered in a rewritable format called CD-rewritable.

Digital Video Disc (DVD) Technology

The digital video disc (DVD) family of formats is replacing the CD-ROM family of formats. Digital video disc technology, also known as "digital versatile disc," is newer and, hence, relatively immature. DVD has five formats: DVD-ROM, DVD-Video, DVD-Audio, DVD-R (the "R" stands for "recordable"), and DVD-RAM. DVD-R is the format for write-once (incremental updates). It specifies 3.95 GB for single-sided discs and 7.9 GB for double-sided discs. DVD-RAM is the format for rewritable discs. It specifies 2.6 GB for single-sided discs and 5.2 GB for double-sided discs, with a disc cartridge as an option. DVD-ROMs (read-only discs) are similar to CD-ROMs and have a storage capacity of 4.7 GB (single-sided, single-layer), 9.4 GB (double-sided, single-layer), 8.5 GB (double-layer, single-sided), 17 GB (dual-layer, double-sided). These are backward-compatible with CD-audio and CD-ROM. DVD-ROM drives can play DVD-R and all the DVD formats. UDF is the file system for DVD-R.

WORM (Write Once, Read Many) Technology

Write once, read many (WORM) technology has helped initiate the document-imaging revolution. WORM uses laser technology to permanently alter sectors of the disc, thereby permanently writing files onto the media. Since this alteration is permanent, the device can write only once to each disc. WORM is typically employed in imaging systems in which the images are static and permanent.

Rewritable Optical Technology

Two new technologies are being employed that utilize rewritable optical technology. These technologies include magneto-optical (MO) and phase change rewritable (PCR) discs. MO drives are more widely accepted because the media and drive manufacturers use the same standards and their products are cross-compatible. PCR devices, however, come from one manufacturer (Matsushita/Panasonic), and the media comes from two manufacturers (Panasonic and Plasmon).

Multifunction Drives

There are two versions of multifunction optical drives. One uses firmware in the drive that first determines whether a disc has been formatted for write-once or rewritable recording and then acts on that disc accordingly. In the other MO version, two entirely different media are used. The rewritable discs are conventional MO discs, but write-once media are traditional WORM media.

3.4 Disaster Recovery

Trying to recover from a disaster, regardless of how it was caused, can be a terrifying experience. How successful the recovery is depends on the extent to which the network administrator has implemented disaster prevention and preparedness.

Disaster Prevention

The best way to recover from a disaster is to prevent it from happening in the first place. When implementing disaster prevention:

- Focus on factors over which you have control.
- Determine the best method of prevention.
- Implement and enforce the preventive measures you select.
- Check continually for new and better methods of prevention.
- Perform regular and routine maintenance on all network hardware and software components.
- Remember that training is the key to preventing network disasters of the human kind.

Disaster Preparation

Not all disasters can be prevented. Every jurisdiction has a disaster-preparedness plan, and many hours are spent every year in practicing for such an event. Because each community is different, recovery plans will have to take different factors into account. If, for example, you live in a flood zone, you should have a plan to protect your network from high water.

When considering disaster protection, you will need a plan for hardware, software, and data. Hardware and software applications and operating systems can be replaced. But to do this, it's necessary first to know exactly what assets you have. Take inventory of all hardware and software, including date of purchase, model, and serial number.

Physical components of a network can be easily replaced and are usually covered by some form of insurance, but data is highly vulnerable to disaster. In the case of a fire, you can replace all the computers and hardware, but not the files, drawings, and specifications for the multimillion dollar project that your organization has been preparing for the last year.

The only protection from a data-loss disaster is to implement one or more of the methods described earlier to back up data. Store your backups in a secure place, such as a bank safe deposit box, away from the network site.

To fully recover from any disaster you will need to:

- Make a disaster-recovery plan.
- Implement the plan.
- Test the plan.

4.0 CONCLUSION

You would have learned about fault tolerant system and it’s strategies as well as network disaster management.

5.0 SUMMARY

The following points summarize the main elements of this lesson.

- Planning for a disaster is an essential part of implementing a successful network.
- A network disaster plan should encompass the loss of hardware and data.
- Tape backup is the most common method of preventing data loss.
- Loss of electrical power can causes files to become corrupted, and any data being held in RAM to be lost.
- An uninterruptible power supply provides temporary power so that critical data can be properly stored before the network or computer goes down.
- Fault tolerance is the automatic duplication of data to prevent loss.
- Fault tolerant strategies are called redundant arrays of independent disks (RAID) and include disk striping and disk mirroring.
- Sector sparing is an advanced method of fault tolerance.

ACTIVITY B

1. What is fault tolerance?

6.0 TUTOR MARKED ASSIGNMENT

1. Explain the types of RAID

7.0 REFERENCES/FUTHER READINGS

1. Handbook of Information Security management by Micki Krause and Harold F. Tipton. Publisher: CRC press LLC, ISBN: 0849399475.
2. The protection of Information Security Management by Sean Boran
3. IT Security Cookbook by sean Boran
4. A Structured Approach to Computer Security by Olovsson. Technical Report no 122, 1992.

UNIT FOUR

MAINTAINING A HEALTHY NETWORK ENVIRONMENT

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
 3.1 Computer and the environment
 3.2 Creating the right environment
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUCTION

The physical environment in which a network resides is an important factor to consider in keeping a computer network physically secure. This lesson explores this frequently overlooked aspect of network management: ensuring a safe environment for computers, peripherals, and the associated network, and looks at what you can do to maintain the health of your network environment.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Describe the impact of environmental conditions on a network.
- Describe the environmental conditions required for proper network operation.
- Describe several methods for protecting network equipment in a harsh environment.

3.0 MAIN CONTENT

3.1 COMPUTERS AND THE ENVIRONMENT

Most kinds of electronic equipment, such as computers, are rugged and reliable, operating for years with little maintenance. Computers have even been to the moon and back. However, negative environmental impacts on electronic equipment, while not always dramatic, do exist. A slow and steady deterioration process can lead to intermittent but ever-more-frequent problems until a catastrophic system failure occurs. By recognizing these problems before they occur and taking appropriate steps, you can prevent or minimize such failures.

Like humans, computers and electronic equipment are affected by environmental conditions. Although more tolerant and less likely to complain, computers and network equipment require specific environments in order to function properly. Most computers are installed in environmentally controlled areas; but even with such controls in place, computers are not immune from the effects of their surroundings. When assessing how environmental conditions will affect a computer network, your first step is to consider the climatic conditions of the region. As shown in Figure 4.5, a network installation in an Arctic or Antarctic location will be subjected to very different conditions than a network located in a tropical jungle.

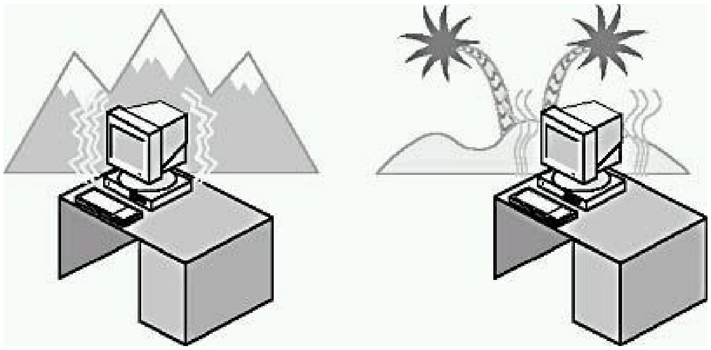


Fig 4.5: Environmental extremes affect computers

A network installed in an arctic climate will undergo extreme changes in temperature, whereas a network installed in a tropical environment will experience high humidity. Different climatic circumstances require that different steps be taken to ensure that the environment does not negatively affect the network.

Environmental conditions for computers are assumed to be the same as prevailing office conditions. For a single personal computer or workstation, this assumption is usually accurate.

However, an individual workstation comprises only part of the network. Remember that network wiring, runs through walls and in ceilings, basements, and sometimes outside. Many environmental factors can affect these components and ultimately lead to a network deterioration or breakdown.

When planning or maintaining a network, it is important to think in terms of the global (entire) network, visible or out of sight, and not just the local components that we see every day, as illustrated in Figure 4.6.

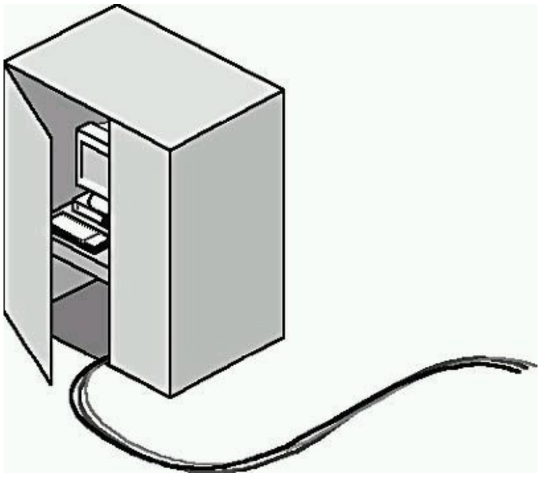


Figure 4.6: Include hidden network components in environmental assessment

Environmentally triggered disasters are usually the result of a long period of slow deterioration, rather than a sudden catastrophe. As an example, consider an iron nail. Left outside and exposed to the elements, it will gradually rust, becoming useless for its original purpose and, eventually, disintegrate. Similarly, networks implemented in poor environments might work well for years; however, eventually intermittent problems will start to occur and the number and frequency of the problems increase until eventually the network goes down.

3.2 CREATING THE RIGHT ENVIRONMENT

In most large organizations, management or the personnel department is responsible for providing a safe and comfortable environment for employees. Governmental organizations regulate the human work environment. There are no such regulations or guidance for networks. It is the responsibility of the network administrator to create policies governing safe practices around network equipment and to implement and manage an appropriate working environment for the network.

A healthy environment for network equipment is much like a healthy human environment; electronic equipment is designed to operate within the same range of temperature and humidity that feels comfortable to human beings.

Temperature

The basic environmental parameter that we control is temperature. Homes, offices, and work places usually have some means of controlling the temperature. Because electronic equipment generates heat during normal operation, it usually has a cooling fan designed to maintain the temperature within the specified limits. If, however, the room temperature in which the equipment is located is too high, the cooling fan and ventilation slots will be unable to maintain the correct operating temperature and components will begin to overheat and fail. Alternatively, if the temperature outdoors is too cold, the components may not function at all. Figure 4.7 shows the back and side of a computer with its cooling fan and ventilation slots.

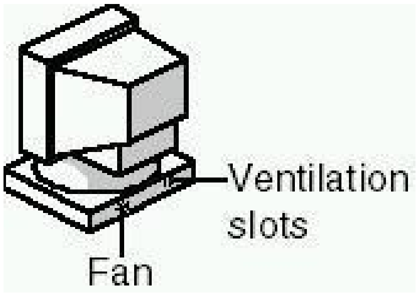


Figure 4.7: Keeping the computer cool

An environment in which the temperature is constantly cycling between hot and cold presents the worst scenario for electronic equipment. These extreme changes cause metal components to expand and contract, which eventually can lead to equipment failure.

Humidity

Factors related to humidity (moisture in the air) can have two negative effects on electronic equipment. High humidity promotes corrosion. Usually occurring first on electrical contacts, corroded contacts on cable connections and expansion cards will cause intermittent failures. Corrosion can also increase the resistance of electrical components, leading to a temperature increase that can be followed by component failure or fire.

In heated buildings, low humidity is common. Static electrical discharge is more common in low-humidity environments and can damage electronic components.

Because we have less control over humidity, network administrators need to be aware of the consequences of very high or low humidity and implement appropriate safeguards where such conditions exist. Most equipment will function adequately between 50 and 70 percent relative humidity.

When implementing a large network that includes a dedicated server room, you should consider controlling temperature and humidity in that room.

Dust and Smoke

Computers and electronic equipment do not function well with dust or smoke. Dust is electro statically attracted to electronic equipment. An accumulation of dust causes two negative effects: dust acts as an insulator that affects the cooling of components, causing them to overheat, and dust can hold electrical charges, making them conductive. Excessive dust on electronic equipment can cause electrical shorts and catastrophic equipment failure.

Smoke causes a kind of contamination that is similar to the effects of dust. It coats the surfaces of electronic components, acting as both insulator and conductor. Smoke residue also enhances the accumulation of dust.

Human Factors

In designing a network, we can control many environmental factors, such as temperature, humidity, and ventilation. Although it is theoretically possible to create a perfect physical environment for computers, the arrival of human beings on the scene will bring changes that are bound to have an impact on the network. Picture a new, environmentally correct, equipment-friendly office with an up-to-date computer, printer, and desk. Into this pristine space, employees bring plants, pictures, radios, coffee cups, books, papers, and space heaters for cold days. Soon the office is filled up with employees, furniture, storage units, and office supplies. More changes occur; the tops of computers and monitors serve as end tables, and empty boxes are stored under desks next to computers. Because few employees have any awareness of the ventilation requirements for computer equipment, they impede the natural flow of air in and around the equipment. Once this happens, maintaining the proper temperature is impossible and failures begin.

The spilling of liquid refreshment takes a toll on keyboards and computers. When it gets cold outside, space heaters are used in under-heated offices and are usually placed under the desk, often in close proximity to computers. This can present two problems: the computer becomes overheated, and the space heaters can overload power outlets, tripping circuit breakers or even causing fires.

Hidden Factors

As stated earlier, much of a network is out of sight and, therefore, often out of mind. Because we don't see these hidden elements on a daily basis, we assume that all is well until something goes wrong.

Wiring is one network component that can cause problems, especially wires lying on the floor. Wires that run through an attic can easily be damaged by accident during repairs to other objects in the attic.

Bugs and rodents of all kinds are another hidden factor; these unwanted guests are likely to dine on the network materials or use them for construction purposes of their own.

Industrial Factors

Computers are not limited to the office setting; they are vital to the manufacturing sector as well. At first, computers were used to manage the flow of work through manufacturing operations. In modern plants, computers also run the equipment. By integrating network technology into this environment, the entire manufacturing process can be monitored and controlled from a central location. The equipment can even telephone maintenance personnel at home when there is a problem.

These improvements in manufacturing have led to an increase in productivity, while presenting unique issues for the network administrator. The operation of network equipment in a production environment presents many challenges. Issues that need to be addressed when networks are implemented in a manufacturing setting include the presence of:

- Noise.
- Electromagnetic interference (EMI).
- Vibration.
- Corrosive and explosive environments.
- Untrained and unskilled workers.

Manufacturing environments often have little or no control over temperature and humidity, and the atmosphere can be contaminated with corrosive chemicals. A corrosive atmosphere with high humidity can destroy computer and network equipment within months and even, in some cases, days. Manufacturing environments that utilize heavy equipment with large electrical motors can wreak havoc on the stability of computer-operated systems and networks. To minimize problems that stem from operating a computer network in an industrial environment:

- Install the networking equipment in separate enclosures with outside ventilation.
- Use fiber-optic cabling. This will reduce electrical interference and corrosion problems with the cable.
- Make sure that all equipment is properly grounded.
- Provide proper training to all employees that need to use the equipment. This will help ensure the integrity of the system.

4.0 CONCLUSION

You would have learned about Computer and the environment and creating the right environment.

5.0 SUMMARY

You have learned about the impact of environmental conditions on a network, the environmental conditions required for proper network operation and several methods for protecting network equipment in a harsh environment.

ACTIVITY B

1. Describe the ways in which heat, humidity, dust, and smoke can each have an adverse effect on computer health. For each, describe preventive measures that can be taken to protect computers in such environments.
2. Identify at least three of the human factors that can unintentionally alter a computer's operating environment. Describe how each of these factors can affect the computer and suggest some preventive measures for each.

6.0 TUTOR MARKED ASSIGNMENT

Identify the principal hidden and industrial factors that can affect a network's health. Include out-of-view network equipment in both an office, and a manufacturing environment.

Discuss what precautions can be taken, or what changes might need to be made, for each of these hidden and industrial factors.

7.0 REFERENCES/FUTHER READINGS

1. Handbook of Information Security management by Micki Krause and Harold F. Tipton. Publisher: CRC press LLC, ISBN: 0849399475.

2. The protection of Information Security Management by Sean Boran

3. IT Security Cookbook by sean Boran

4. A Structured Approach to Computer Security by Olovsson. Technical Report no 122, 1992.

UNIT FIVE

AVOIDING DATA LOSS

TABLE OF CONTENTS

1.0 Introduction
2.0 Objectives
3.0 Main content
    3.1 Data Protection
    3.2 Implementing a backup system
    3.3 Uninterrupted Power Supply

4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 References

1.0 INTRODUTION

In this module, we have covered maintaining network hardware and data security and keeping computer components safe from harm. However, making networks secure also includes protecting the data from corruption or loss. This unit presents an overview of the possible causes of data loss and how to protect the network against them. You will also learn about systems and processes for preventing data loss.

2.0 OBJECTIVES

At the end of this unit, you should be able to:

- Identify the reasons for implementing a backup system.
- Select a backup approach that is appropriate for a given site, including the method and schedule.
- List the considerations for implementing an uninterruptible power supply.
- Describe each of the following types of fault-tolerant systems: disk striping, disk mirroring, sector sparing, clustering.

3.0 MAIN CONTENT

3.1 DATA PROTECTION

A site disaster is defined as anything that causes you to lose your data. Many large organizations have extensive disaster-recovery plans to maintain operations and rebuild after a natural disaster such as an earthquake or a hurricane. Many, but not all, include a plan to recover the network. However, a network can incur a disastrous failure from many more sources than natural disasters. Disaster recovery for a network goes beyond the replacing of the physical hardware; the data must be protected as well. The causes of a network disaster, ranging from human acts to natural causes, include:

- Component failure.
- Computer viruses.
- Data deletion and corruption.
- Fire caused by arson or electrical mishaps.
- Natural disasters, such as lightning, floods, tornadoes, and earthquakes.
- Power-supply failure and power surges.
- Theft and vandalism.

In the event of a site disaster, the downtime spent recovering data from backup storage (if you have backups) could result in a serious loss of productivity. And without backups, the consequences are even more severe, possibly resulting in significant financial losses. There are several ways to prevent or recover from data loss, including:

- Tape backup systems.
- An uninterruptible power supply (UPS).
- Fault-tolerant systems.
- Optical drives and disks.

Any or all of these approaches can be used, depending on how valuable the data is to the organization and on the organization's budget constraints.

Tape Backup

The simplest, most inexpensive way to avoid disastrous loss of data is to implement a schedule of periodic backups with storage offsite. Using a tape backup is still one of the few simple and economical ways to ensure that data remains safe and usable.

Experienced network engineers advise that a backup system should be the first line of defense against data loss. A secure backup strategy minimizes the risk of losing data by maintaining a current backup—copies of existing files—so that files can be recovered if harm comes to the original data. To back up data requires:

- Appropriate equipment.
- A regular schedule for periodic backups.
- Ensuring that backup files are current.
- Personnel assigned to make sure this schedule is carried out.

The equipment usually consists of one or more tape drives and tapes or other mass storage media. Any expense incurred in this area is likely to be minimal compared to the value of what will be saved in the event of data loss.

3.2 IMPLEMENTING A BACKUP SYSTEM

The rule is simple; if you cannot get along without it, back it up. Whether you back up entire disks, selected directories, or files depends on how fast you will need to be operational after losing important data. Complete backups make restoring disk configurations much easier, but can require multiple tapes if there are large amounts of data. Backing up individual files and directories might require fewer tapes, but could require the administrator to manually restore disk configurations.

Critical data should be backed up according to daily, weekly, or monthly schedules, depending on how critical the data is and how frequently it is updated. It is best to schedule backup operations during periods of low system use. Users should be notified when the backup will be performed so that they will not be using the servers during server backup.

Selecting a Tape Drive

Because the majority of backing up is done with tape drives, the first step is to select a tape drive, weighing the importance of a variety of factors, such as:

- How much data needs to be backed up.
- The network's requirements for backup reliability, capacity, and speed.
- The cost of the tape drive and related media.
- The tape drive's compatibility with the operating system.

Ideally, a tape drive should have more than enough capacity to back up a network's largest server. It should also provide error detection and correction during backup and restore operations.

Backup Methods

As listed in Table 4.2, an efficient backup policy uses a combination of methods:

Table 4.2: Backup Methods

Method	Description
Full backup	Backs up and marks selected files, whether or not they have changed since

	the last backup.
Copy	Backs up all selected files without marking them as being backed up.
Incremental backup	Backs up and marks selected files only if they have changed since the last time they were backed up.
Daily copy	Backs up only those files that have been modified that day, without marking them as being backed up.
Differential backup	Backs up selected files only if they have changed since the last time they were backed up, without marking them as being backed up.

Tapes can be backed up based on a multiple-week cycle, depending on how many tapes are available. No rigid rules govern the length of the cycle. On the first day of the cycle, the administrator performs a full backup and follows with an incremental backup on succeeding days. When the entire cycle has finished, the process begins again. Another method is to schedule streaming backups throughout the day.

Testing and Storage

Experienced administrators test the backup system before committing to it. They perform a backup, delete the information, restore the data, and attempt to use the data.

The administrator should test the backup procedures regularly to verify that what is expected to be backed up is actually being backed up. Additionally, the restore procedure should be tested to ensure that important files can be restored quickly.

Ideally, an administrator should make two copies of each tape: one to be kept onsite, and the other stored offsite in a safe place. Remember that although storing tapes in a fireproof safe can keep them from actually burning, the heat from a fire will ruin the data stored on them. After repeated usage, tapes lose the ability to store data. Replace tapes regularly to ensure a good backup.

Maintaining a Backup Log

Maintaining a log of all backups is critical for later file recovery. A copy of the log should be kept with the backup tapes, as well as at the computer site. The log should record the following information:

- Date of backup
- Tape-set number
- Type of backup performed
- Which computer was backed up

- Which files were backed up
- Who performed the backup
- Location of the backup tapes

Installing the Backup System

Tape drives can be connected to a server or a computer, and backups can be initiated from the computer to which the tape drive is attached. If you run backups from a server, backup and restore operations can occur very quickly because the data does not have to travel across the network.

Backing up across the network is the most efficient way to back up multiple systems; however, it creates a great deal of network traffic and slows the network down considerably. Network traffic can also cause performance degradation. This is one reason why it is important to perform backups during periods of low server use.

If multiple servers reside in one location, placing a backup computer on an isolated segment can reduce backup traffic. As shown in Figure 4.8, the backup computer is then connected to a separate NIC on each server.

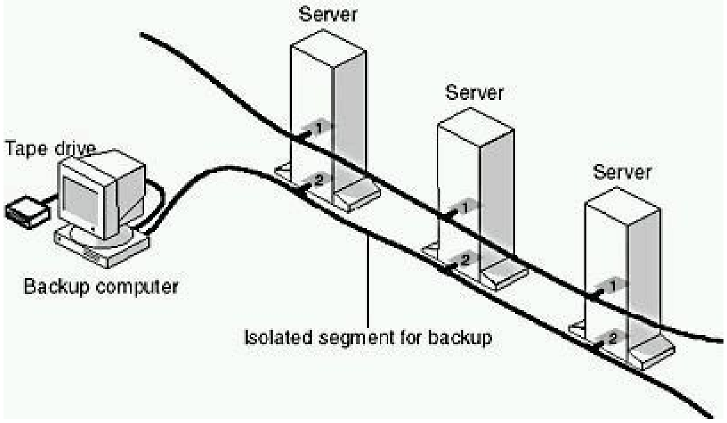


Figure 4.8: Network traffic is reduced by backing up to a separate segment

3.3 UNINTERRUPTIBLE POWER SUPPLY (UPS)

An uninterruptible power supply (UPS) is an automated external power supply designed to keep a server or other device running in the event of a power failure. The UPS system takes advantage of uninterruptible power supplies that can interface with an operating system such as Microsoft Windows NT. The standard UPS provides a network with two crucial components:

- A power source to run the server for a short time
- A safe shutdown management service

The power source is usually a battery, but the UPS can also be a gasoline engine running an AC power supply.

If the power fails, users are notified of the failure and warned by the UPS to finish their tasks. The UPS then waits a predetermined amount of time and performs an orderly system shutdown.

A good UPS system will:

- Prevent any more users from accessing the server.
- Send an alert message to the network administrator through the server.

The UPS is usually located between the server and a power source as depicted in figure 4.9.

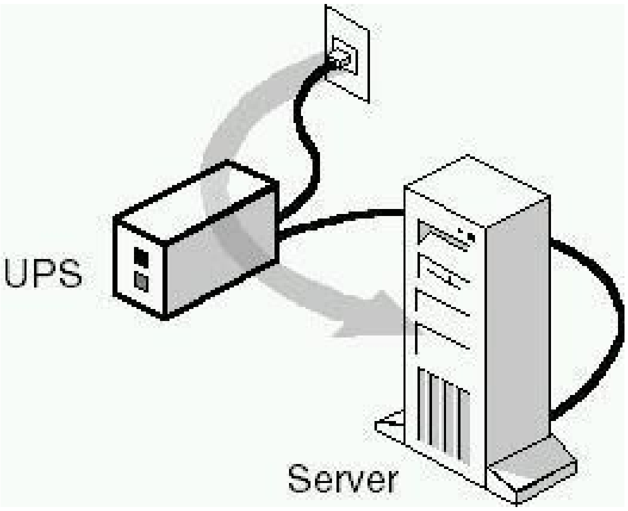


Figure 4.9: Uninterruptible power supply as a backup power source

If power is restored while the UPS is active, the UPS will notify the users that the power has returned.

Types of UPS Systems

The best UPS systems perform online. When the power source fails, the UPS batteries automatically take over. The process is invisible to users.

There are also stand-by UPS systems that start when power fails. These are less expensive than online systems, but are not as reliable.

Implementing UPS

Answering the following questions will help the network administrator determine which UPS system best fits the needs of the network:

- Will the UPS meet the basic power requirements of this network? How many components can it support?
- Does the UPS communicate with the server to notify it when a power failure has occurred and the server is running on batteries?
- Does the UPS feature surge protection to guard against power spikes and surges?
- What is the life span of the UPS battery? How long can it be inactive before it starts to degrade?
- Will the UPS warn the administrator and users that it is running out of power?

4.0 CONCLUSION

You would have learned about Data Protection, implementing a backup system and Uninterrupted Power Supply.

5.0 SUMMARY

You have learned about Identify the reasons for implementing a backup system, how to select a backup approach that is appropriate for a given site, including the method and schedule.

ACTIVITY B

1. What is a site disaster?
2. Enumerates different ways of Backup methods in preventing data loss

6.0 TUTOR MARKED ASSIGNMENT

3. Discuss at least four ways of preventing a data loss

4. Mention at least five causes of network disaster.

7.0 REFERENCES/FUTHER READINGS

1. Handbook of Information Security management by Micki Krause and Harold F. Tipton. Publisher: CRC press LLC, ISBN: 0849399475.
2. The protection of Information Security Management by Sean Boran
3. IT Security Cookbook by sean Boran
4. A Structured Approach to Computer Security by Olovsson. Technical Report no 122, 1992.