

Document Information

Analyzed document	Computer Application.pdf (D166063658)
Submitted	5/6/2023 6:55:00 AM
Submitted by	Mumtaz B
Submitter email	mumtaz@code.dbuniversity.ac.in
Similarity	6%
Analysis address	mumtaz.dbuni@analysis.arkund.com

Sources included in the report

SA	Computer Fundamentals and PC Software (Block 1).pdf Document Computer Fundamentals and PC Software (Block 1).pdf (D148967917)	 1
W	URL: https://vibdoc.com/an-introduction-to-computer-in-banking.html Fetched: 12/11/2021 4:02:31 AM	 10

Entire Document

1 UNIT 1 -

COMPUTERS: AN INTRODUCTION Objectives: The main objective of this lesson is to introduce 'computer' to the students. After successful completion of the lesson the students will be able to understand the characteristics, capabilities as well as the limitations of computers. Structure 1.1 Introduction 1.2 What is a Computer? 1.3 Characteristics of a Computer System 1.4 Capabilities of Computers 1.5 Limitations of Computers 1.6 Careers in Computers 1.7 Self-Assessment Questions 1.8 Suggested Readings

2 1.1 INTRODUCTION Man's continuous search for fast and accurate calculating devices ultimately brings us to the modern age that can be rightly called as the Computer Age. The computer – a package of metal, plastic and wires is truly an amazing machine. This machine is creating sweeping and dramatic changes in all spheres of our present life and working environment. It has transformed and is transforming the ways in which we do our businesses; we learn; we conduct scientific and engineering probes; we play and entertain ourselves ... It is said that this machine will improve world's culture by allowing people not only to be more productive but to have more free time to create. 1.1.1 Why Computers? Two major factors that stimulate development in any field are the following: $\frac{3}{4}$ Demand of users and potential users $\frac{3}{4}$ Available technology Prior to the industrial revolution computers were only used for scientific research. Their potential was first recognized by the US Army during the world war. They were used to keep track of the movement of arms and ammunition. Following the industrial revolution, people felt that computers could also be used for commercial purpose. For example, in the banks to maintain efficient banking system, in industrial and business houses to maintain their inventory for accounts management, sales and budgeting, production and planning. At the same time dramatic development in the Electrical, Electronic and Communication Engineering together with software expertise has enabled many of these demands to be met. One such engineering technology is the Information Technology, popularly known as IT, which brought computers to our homes and offices. This led to a revolution in the

3 IT industry, keeping in mind the need of the modern era that believed in the speedy storage and retrieval of information, at any point of time. This resulted in an effective and efficient working environment. Thus, the modern age is rightly called the 'Computer Age', making computers indispensable. Current growth in the Information Technology is leading to the concept of Paperless Offices. Advantages of Computers

- 9 Perform repetitive tasks
- 9 More accurate while dealing with large numbers
- 9 Work at constant efficiency
- 9 Are used in almost every field
- 9 Process millions of instructions in a fraction of a second

Effective and efficient data storage and retrieval

1.2 WHAT IS A COMPUTER?

The word "computer" comes from the word 'compute', which means to calculate.

So a computer is

normally considered to be a calculating device.

In fact the original objective for inventing the

computer was to create a fast calculating machine. But more than 90% of the

work done by computers today is of non-mathematical or non-numerical nature. Hence, to define

a

computer merely as calculating device is to ignore over 90% of its

work. More accurately, a computer

may be defined as a device that operates upon information or data. Data can be anything like bio-data of various applicants when the computer is used for recruiting personnel, or the marks obtained by various students in various subjects when the computer is used to prepare results, or the details (name, age, sex etc.) of various passengers when the computer is employed for making airline or railway

4 reservations, or numbers of different types in case of application of computers for scientific research problems, etc.

Thus Computer can be defined as an electronic device for processing data that takes data input from its user, stores, processes data and generates the required output as per the processing instructions given to it by the user.

The fact that computers process data is so fundamental that

many people

have started calling it a data processor. The name data processor is more inclusive because modern computers not only compute in the usual sense but also perform other functions with

the

data that flow to and from them. For example, data processor may $\frac{3}{4}$ gather data from various sources; $\frac{3}{4}$ merge (process of mixing or putting together) them all; $\frac{3}{4}$ sort (process of arranging in some sequence-ascending or descending) them; and $\frac{3}{4}$ Print them in desired format.

Thus computers not only can add, subtract, multiply and divide numbers but can also do certain logical operations; can remember (i.e. store and recall information); can communicate with operators; can direct themselves in a predetermined manner; can process a large volume of data effortlessly; can interpret messages from remote locations. Computers undertake repetitive and boring tasks, relieving us for more critical, creative activities. Computers offer unmatched speed, performance, and accuracy in data processing. Computers work at constant efficiency and perform tasks repeatedly without errors, avoiding the fatigue, that affect human beings. Computers can be used in almost every field and for almost every purpose. Computers allow society to undertake new activities in various fields and to function more efficiently. Computers are impartial.

5 They offer a mean of data processing unaffected by social, religious or cultural bias and prejudice. Computers offer effective and efficient data storage and retrieval, highest degree of integrity and reliability. Figure 1-1 the Computer System Computers come in many varieties, including the personal computer, tiny computers built into appliances and automobiles, and mainframe machines used by many people simultaneously to run a business. Despite their differences in size and use, all these computers are part of a system. A complete computer system consists of four parts: hardware, software, people, and data (see Figure 1-1)

1.3 CHARACTERISTICS OF A COMPUTER SYSTEM

Computers are not just adding machines; they are capable of doing complex activities and operations. They can be programmed to do complex, tedious and monotonous tasks. All computers have certain common characteristics irrespective of their type and size. The following are the important characteristics which taken together, enable a computer to surpass its performance in some tasks in which the human beings cannot perform efficiently:

6 1. Speed: A computer is a very fast device capable of data processing at unbelievable speed.

It can perform in a few seconds the amount of work that a human being may not be able to

do in an entire year even if he works day and night and does nothing else.

Computers can process millions of instructions per second thus carrying out even the complex tasks in fractions of seconds without any mistake.

While talking about the speed of a computer, we do not talk in terms of seconds or even milliseconds (10^{-3}).

Our units of speed are the microseconds (10^{-6}), the nanoseconds (10^{-9}), and even the picoseconds (10^{-12}). A powerful computer is capable of performing about 3 to 4 million simple arithmetic operations per second.

2. Accuracy: In addition to speed, the computer has high accuracy in computing. The accuracy of a computer is consistently high and the degree of accuracy of a particular computer depends upon its design. But for a particular computer, each and every calculation is performed with the same accuracy.

Errors can occur in a computer, but these are mainly due to human rather than technological weakness. The errors in computer are due to errors in programming and operation by human and due to inaccurate data.

3. Versatility: A computer is a very versatile machine. Versatility is one of the most wonderful features of the computer in the sense that they are not only capable of handling complex arithmetical problems, but can do equally well other number of jobs. They can perform activities ranging from simple calculations to performing complex CAD modeling and simulations to navigating missiles and satellites. In

7 other words, computers can be programmed to perform any task that can be reduced to a series of logical steps. Computers can communicate with other computers and can receive and send data in various forms like text, sound, video, graphics, etc. We, now, live in a connected world and all this is because of computers and other related technologies.

4. Diligency: Unlike human beings, a computer is free from monotony, tiredness, lack of concentration etc. and hence

can work for hours together without creating any error and without grumbling.

Due to this property computers obviously score over human beings in doing routine type of jobs, which require greater accuracy.

They will perform the tasks that are given to them - irrespective of whether it is interesting, creative, monotonous or boring; irrespective of whether it is the first time or the millionth time - with exactly the same accuracy and speed.

5. Storage Capability: Computers have their main memory and auxiliary memory systems. A computer can store a large amount of data. With more and more auxiliary storage devices, which are capable of storing huge amounts of data, the storage capacity of a computer is virtually unlimited. The factor that makes computer storage unique is not that it can store vast amount of data, but the fact that it can retrieve the information that the user wants in a few seconds. Every piece of information can be retained as long as desired by the user and can be recalled as and when required. Even after several years, the information recalled is as accurate as on the day when it was fed to computer.

A computer forgets or loses certain information only when it is asked to do so. So it is entirely up to the user to make a computer retain or forget particular information.

8 6.

Reliability: Reliability of the computers is indeed very high. Modern electronic components have long failure free lives. A microprocessor chip is said to have a life of 40 years even under adverse conditions and much before it fails, it will become obsolete. Computers are also designed in modular form so as to make maintenance easy; when a component fails, it can be replaced or repaired at a minimal cost.

7. Automation: The level of automation achieved in a computer is phenomenal. It is not a simple calculator where you have to punch in the numbers and press the equal to sign to get the result. Once a task is initiated, computers can proceed on its own till its completion. Computers can be programmed to perform a series of complex tasks involving multiple programs. Computers will perform these things flawlessly. They will execute the programs in the correct sequence, they will switch on/off the machines at the appropriate time, they will monitor the operational parameters, and they will send warning signals or take corrective actions if the parameters exceed the control level, and so on. Computers are capable of these levels of automation, provided they are programmed correctly.

1.4 CAPABILITIES OF COMPUTERS Stepping down from the domains of technical people such as scientists and engineers; computer, today, is a very familiar household word. In 1950's computers were special purpose machines, which only huge institutions such as governments and universities could afford. In the 1960's modern computer began to revolutionize the business world and today it has become popular with all kinds of people from business to employees, from doctors to lawyers and from players to school going children. Today, computers are

9 directly or indirectly influencing every aspect of our lives. Wherever human intellect and technology meet, we will find computers. Computers of all sizes and shapes are used for every purpose imaginable - from selling railway tickets to running washing machines; from stock market analysis to playing games; from publishing a new letter to designing a building... They can perform activities ranging from simple calculations to performing complex CAD modeling and simulations to navigating missiles and satellites. Computers can communicate with other computers and can receive and send data in various forms like text, sound, video, graphics, etc. This ability of computer to communicate to one another has led to the development of computer networks, Internet, WWW and so on. Today, we can send e-mail to people all around the world. We, now, live in a connected world and all this is because of computers and other related technologies.

Uses of Computers During the last four decades, computers have revolutionized almost all disciplines of our life.

Computers have made possible many scientific, industrial and commercial advances that would have been impossible otherwise.

Computers

are being used in many areas of application viz. business, industry, scientific research, defence, space, communications, medicine, education and so on.

The utilisation of computers in different fields is summarised in Table 1-1 Table 1-1 Roles of Computers

in Various Fields Application Area Use of Computers Scientific Research Used to resolve complex scientific problems accurately in a short time Business Used in banks, airports, share markets, hotels, export houses, Government offices and others for rising business

10 applications like MIS, Payroll, Inventory, Financial Accounting etc. Defense Used to computerize warplanes, ships, radars and many advanced weapons

Space Used to design computerized space satellites, rockets and related technology

Data Communication Used to computerize geo-geographically separated offices through networking

Telecommunication Used in ISDN, E-mail, Internet, Intranet, VSAT, Videoconferencing, Paging, Cellular phones etc.

Medicine Used in hospitals and nursing homes/clinics for maintaining medical records, prescription writing, diagnostic applications and computerized scanning(CAT Scanning) Education Used in development of CBT (Computer Based Teaching)/ CAT (Computer Aided Teaching) programs for education Law & Order Used to records data of vehicles

Libraries Used to develop Library Management Systems Publishers Used for Desk Top Publishing (DTP) for designing & printing of books Engineering Used CAD (Computer Aided Designing)/CAM (

Computer Aided Manufacturing) by engineering companies Emerging Technologies Used in Artificial Intelligence (Expert Systems, Robotics etc.) and Virtual Reality 1.4.1 COMPUTERS IN BUSINESS Computers have completely altered the structure of business. They are reshaping the basics of business. Customer service, operations, product and marketing strategies, and distribution are heavily, or sometimes even entirely, dependent on computer-enabled applications. The computers that support these functions can be found on the desk, on the

11 shop floor, in the store, even in briefcases. Computers have become an everyday part of business life. Figure 1-2 illustrates the fundamental roles of computers in business. Computer systems perform three vital roles in any type of organization: $\frac{3}{4}$ Support of Business Operations $\frac{3}{4}$ Support of Managerial Decision-Making $\frac{3}{4}$ Support of Strategic Competitive Advantage Figure 1-2 Three Major Roles of Computer-based Information Systems Let's take a retail store as an example to illustrate these important roles. As a consumer, we have to deal regularly with the computer-based information systems that support business operations at many retail stores where we shop. For example, most retail stores now use computer-based information systems to help them record customer purchases, keep track of inventory, pay employees, buy new merchandise, and evaluate sales trends. Store operations would grind to a halt without the support of such information systems. See Figure 1-3.

Support of Strategic Competitive Advantage Support of Managerial Decision-Making Support of Business Operations

Support

of Strategic Competitive Advantage Support of Managerial Decision-Making Support of Business Operations

12

Computer-based information systems also help store managers make better decisions and attempt to gain a strategic competitive advantage. For example, decisions on what lines of merchandise need to be added or discontinued, or on what kind of investment they require, are typically made after an analysis provided by computer-based information systems. This not only supports the decision making of store managers but also helps them look for ways to gain an advantage over other retailers in the competition for customers. Figure 1-3 A Computer-based Sales Processing System that supports Sales Transaction Processing, Employee Inquiries and Responses, and the Access and Updating of Business Databases Gaining a strategic advantage over competitors requires innovative use of computers and information technology. For example, store managers might make a decision to install computerized touch-screen catalog ordering systems in all of their stores, tied in with computer-based telephone ordering systems and an Internet-based computer shopping

13 network. This might attract new customers and lure customers away from competing stores because of the ease of ordering provided by such innovative information systems. Thus, computer-based strategic information systems can help provide strategic products and services that give a business organization competitive advantage over its competitors. 1.5

LIMITATIONS OF COMPUTERS There is no doubt that computers surpasses human being in many aspects and can perform certain tasks better, faster and cheaper. But it cannot substitute man. The words of John F Kennedy are also 100% true "Man is still the most extraordinary Computers of all". The Computer, being an electronic device, has certain limitations, which can be summarized as follow: 1. No IQ: A computer is not a magical device. It can only perform tasks that a human being can. The difference is that it performs these tasks with unthinkable speed and accuracy.

It possesses no intelligence of its own. Its IQ is zero, at least till today.

Hence, only the user can

determine

what tasks a computer will perform. A computer cannot take its own decision in this regard.

Unlike the human brain, a computer cannot think on its own, but has to be given very explicit, step-by-step instructions to make it perform a task. 2.

No Feelings: Computers are devoid of emotions. They have no feelings and no instincts because they are machine.

Although men have succeeded in building a memory for

the

computer, but no computer possesses the equivalent of human heart and soul. Based on our feelings, taste, knowledge, and experience, we often make certain judgments in our day-to-day life. But computers cannot make such judgments on their own.

Their judgments are

based on the instructions given to

14 them in the form of programs that are written by us.

They are only as good as man makes and uses them. They do not learn from experiences. It is said for computers, "Garbage In Garbage Out (GIGO)". Many of the problems with computers occur because the computer can't tell the difference between doing something sensible versus something ridiculous. Erasing all its stored data is no different to a computer from adding two numbers. Computers operate logically, but they are incapable of acting prudently and rationally. Thus a computer is not intelligent: it is a fast, rule-following idiot. Fast because it works at electronic speeds; rule-following because it needs to be given very detailed and complete instructions before it can do even the simplest task; and an idiot because it will unhesitatingly follow instructions even when to us it would be obvious that they were nonsense. 1.6 **CAREERS IN COMPUTERS** Computers are growing in popularity very rapidly. Computers are running almost everything we can of think: from organizing records to directing traffic. The Information Technology (IT) industry is growing at an incredible rate. The impact of computers on our everyday lives is monumental, though taken for granted. Every time we make a bank deposit, purchase items on a credit card, pay an insurance premium or rent a video movie, innumerable computer operations are involved. Making all these operations happen, behind the scenes, is the work of a vast array of professionals: computer programmers, programmer analysts, systems analysts, hardware and software engineers, database

15 managers, etc. As our society becomes more computerized and technologically sophisticated, the need for highly skilled computer professionals increases accordingly.

1.6.1 Career Options The computer technology industry on the whole encompasses many fields of professional involvement and advancement. In broad terms, here are some of the career options that one can consider

- 1. Computer Science Jobs** Computer scientists are involved in designing computer systems and in researching ways to enhance the practical applications of such designs. Computer scientists address highly theoretical and complex problems associated with making new technology beneficial to all segments of society: academia, the military, civilian businesses, end-user consumers, etc. Included in this group of computer scientists are computer engineers, database administrators, computer support analysts, and other technically specialized professionals.
- 2. Systems Development Jobs** People working in this field analyze the informational needs within an organization and the ways by which various computer systems should properly relate to each other to enhance the overall operation of the organization. Systems analysts ensure that the functional areas of the organization - accounting, marketing, sales etc.- communicate properly with each other. To accomplish this task, systems analysts study and modify the capabilities of the computer hardware and software to meet the changing demands of an evolving organization.
- 3. EDP Auditor Jobs** This is a challenging career option for professionals who have keen interests and skills in computers as well as in accounting and finance. The fundamental goal of EDP (Electronic Data Processing) auditing is to ensure the accuracy, efficiency, and integrity of a company's computer system, which is at the heart of all its business operations. EDP auditors are concerned, in part, with the accuracy of computer input and output as this accuracy relates to the possibility of financial impropriety, security leaks, or fraud. Along with knowledge of computer systems, programming languages, and various applications, EDP auditors need a good understanding of business and financial management. In fact, many EDP auditors hold MBA degrees and/or CA certification.
- 4. Consulting Jobs** One who aspires to become computer consultant can follow a number of career paths. Some computer consultants are motivated by strong entrepreneurial instincts. With several years of industry experience, they choose the route of freelance consulting, often as stepping-stones for starting their own companies to carve their niches in the computer market. Talented young professionals may also consider a career with combined-practice companies, such as the prestigious consulting firms, or with major hardware/ software suppliers, or with international consulting firms that offer computer consulting as part of turnkey business services. Other rewarding computer careers include sales/marketing support, technical writing and instruction, quality assurance, network engineering, management information systems, and so forth.
- 5. Teaching Jobs** One can make career in computer teaching at both under graduate and post graduate levels.

1.6.2 Computers in Non-IT Professions Even if we are not interested in becoming a computer professional, it is a must that we have basic knowledge of computers and the commonly used applications. It is imperative that we become a computer savvy professional-a person who can use computers with ease to perform routine tasks like composing a letter or memo, send and receive e-mail, surf the Internet, make computer presentations, etc. Because in today's information age- where computers hold the center stage - computer proficiency is a must for our survival and success. The workers and professionals of the world will soon be divided into two distinct groups: those who will control computers and those who will be controlled by computers. It would be best for you to be in the former group.

Lewis D Eigen

1.7 SELF-ASSESSMENT QUESTIONS

1. What is a computer? Why is it known as data processor?
2. What are the motivating factors behind the development of computers?
3. Explain some of the important characteristics of computers.
4. Write a short note on " Capabilities of Computers". How the field of business is affected by the capabilities of computers?
5. Explain the limitations of computers.

1.8 Suggested Readings

1. Fundamentals of Computers by V Raja Raman. Prentice Hall of India Pvt. Ltd., New Delhi.
2. Computer Fundamentals by P K Sinha. BPB Publications., New Delhi.
3. Computer Fundamentals by B Ram. New Age International.
4. Introduction to Computers by Leon and Leon. Vikas Publishing House., New Delhi.

19

UNIT 2 -

FUNDAMENTAL OF COMPUTER SYSTEMS Objectives : The main objective of this lesson is to understand the computer as a system. After successful completion of the lesson the students will be able to understand the basic organization, components and functions of a computer system.

Structure

- 2.1 Introduction
- 2.2 The Computer System
- 2.3 Components of a Computer System
- 2.4 Basic Computer Organisation
- 2.5 How Does a Computer Works?
- 2.6 Self-Assessment Questions
- 2.7 Suggested Readings

20 2.1 INTRODUCTION A computer is basically a tool for data processing- converting the data into information that is useful to people. Any computer - regardless of its type - is controlled by programmed instructions, which give the machine a purpose and tell it what to do. Computers come in many varieties, including the personal computer, tiny computers built into appliances and automobiles, and mainframe machines used by many people simultaneously to run a business. Despite their differences in size and use, all these computers are part of a system and operate on the same fundamental principles. In this lesson we will understand the computer as a system. Understanding the computer as a system is vital to the effective use and management of computers. We should be able to visualize any computer this way, from a microcomputer to a large computer system whose components are interconnected by a telecommunications network and spread throughout a building or geographic area.

2.2 THE COMPUTER SYSTEM A system is a group of integrated parts that have the common purpose of achieving some objective(s). So, the following three characteristics are key to a system: $\frac{3}{4}$ a system has more than one element $\frac{3}{4}$ all the elements of a system are logically related $\frac{3}{4}$ all the elements of a system are controlled in such a way that the system goal is achieved Since a computer is made up of integrated components (input and output devices, storage, CPU) that work together to perform the basic system function of:

1. Inputting: The process of entering data and instructions into the computer system.
 2. Storing: Saving data and instructions so that they are available for initial or for additional processing as and when required.
 3. Processing: Performing arithmetic and logical operations on data in order to convert them in to useful information.
 4. Outputting: The process of producing useful information or results-such printed reports, or visual display-for the user.
 5. Controlling: Managing the manner and sequence in which all of the above functions are performed. All the components of a computer system (to be discussed shortly) are integrated and interacting. The Input or Output units cannot function until they receive signals from the CPU. Similarly, the Storage unit or the CPU alone is of no use. So the usefulness of each unit depends on other units and can be realized only when all units are put together (integrated) to form a system.

2.3 COMPONENTS OF A COMPUTER SYSTEM A complete computer system consists of four parts: hardware, software, people, and data (see Figure 2-1).
 2.3.1 Hardware The mechanical devices that make up the computer are called hardware. In other words, hardware is any part of the computer we can touch. Hardware consists of interconnected electronic devices that we can use to control the computer's operation, input and output. Computer hardware consists of:
 9 Input Devices: The input devices of a computer system include keyboards, touch screens, trackballs, joysticks, digital cameras, microphones, pens, electronic mice,
 22 optical scanners, and so on. They convert data into electronic machine-readable form for direct entry or through telecommunication links into a computer system. Table 2-1 highlights major trends in input devices of a computer system Table 2-1

Figure 2-1 The Computer System 9 Output Devices: The output devices of a computer system include video display units, printers, audio response units, and so on. They convert electronic information produced by the computer system into human-intelligible form for presentation to end-users.

First Generation Second Generation Third Generation Fourth Generation Fifth Generation Input Media/ Method Punched Cards, Paper Tape

Punched Cards Key to Tape/ Disk Key boards, Pointing Devices, Optical Scanning Voice Recognition, Touch Devices, Handwriting Recognition Trend: Towards Direct Input Devices that are more Natural and Easy to use

23

Table 2-2 highlights major trends in output devices of a computer system Table 2-2 9

Storage Devices: The storage function of a computer system takes place in the storage circuits of the computer's primary storage unit, or memory, and in secondary storage devices such as magnetic disk and tape units. These devices store data and program instructions needed for processing. Various secondary storage devices include tape drives, optical drives, removable hard drives, DVDs floppy disks, and different kinds of compact disks such as CD-ROM, CD-R, CD- RW.

Table 2-3 highlights major trends in storage devices of a computer system 9 Central Processing Unit (CPU): The central processing unit (CPU) is the main processing component of a computer system. (In microcomputers, it is the main microprocessor) The electronic circuits of the arithmetic-logic unit, one of the CPU's major components, perform the arithmetic and logic functions required in computer processing. First Generation Second Generation Third Generation Fourth Generation Fifth Generation

Output Media/ Method Punched Cards, Printed Reports and Documents Punched Cards, Printed Reports and Documents Printed Reports and Documents,

Video Displays Video Displays, Audio Responses, Printed Reports and Documents Video Displays, Hyperlinked Multimedia Documents, Voice Responses Trend: Towards Direct Output Methods that communicate Naturally, Quickly and Clearly

24

Table 2-3 2.3.2

Software Software is a set of electronic instructions consisting of complex codes (also known as programs) that make the computer perform tasks. In other words, software is the

non- tangible

component of the computer system that tells the computer what to do. They are generally in the form of electric or magnetic impulses. Computer software consists of:

- 9 System Software: These programs exist primarily for the computer's use and help the computer perform and manage its own tasks. System software has its orientation more towards the operation of the hardware components of the computer system. DOS, UNIX, Windows 2000 etc. are some examples of system software.
- 9 Application Software: These programs exist primarily for the user and enable the computer to perform tasks, such as creating documents or drawing pictures. Thus, application software has its orientation more towards performing user tasks.

First Generation Second Generation Third Generation Fourth Generation Fifth Generation Primary Storage Magnetic Drum

Magnetic Core Magnetic Core LSI Semiconductor Memory VLSI Semiconductor Memory Secondary Storage Magnetic Tape Magnetic Drum Magnetic Tape Magnetic Disk Magnetic Tape Magnetic Disk Magnetic Disk Optical Disk Magnetic Disk Optical Disk Trend: Towards Large Capacities Using Smaller Microelectronic Circuits Trend: Towards Massive Capacities Using Magnetic and Optical Media

25

Payroll System, Airline Reservation System etc. are some examples of application software. Table 2-4 highlights major trends in software of a computer system Table 2-4 2.3.3

People People are the personnel involved in using and maintaining the computer system. People are

- 9 Users: who actually uses the computer system
- 9 Trained Professionals: who design, build, program, repair and maintain the computer system.

2.3.4 Data Data consists of raw facts, which the computer stores and reads in the form of numbers. The computer manipulates data according to the instructions contained in the software and then forwards it for use by people or another computer. Data can consist of letters, numbers, sounds, or images. No matter what kind of data is entered into a computer,

First Generation Second Generation Third Generation Fourth Generation Fifth Generation Software Trends User-written Programs, Machine Language Packaged Programs, Symbolic Language Operating Systems, High-Level Language DBMS, Fourth- Generation Languages, Microcomputer Packages Natural and Object- Oriented Languages, Multipurpose Graphic- Interface, Network- Enabled, Expert-Assisted Packages

Trend: Towards Easy-to-Use Multipurpose Network- Enabled Application Packages for Productivity and Collaboration Trend: Towards Visual or Conversational Programming Languages and Tools

26

however, the computer converts it to numbers. Consequently, computerized data is digital, meaning that it has been reduced to digits, or numbers. Within the computer, data is organized into files. A computer file is simply a set of data or program instructions that has been given a name. A file that the user can open and use is often called a document. Although many people think of documents simply as text, a computer document can include many kinds of data. For example, a computer document can be a text file (such as a letter), a group of numbers (such as a budget), a video clip (which includes images and sounds), or any combination of these items. Programs are organized into files as well, but because programs are not considered data, they are not document files.

2.4 BASIC COMPUTER ORGANISATION The capacity, size, cost and internal architectural design of computers differ from one model to another. However, the basic organisation remains the same for all

computer systems. A block diagram is shown in Figure 2-2, which displays the five basic building blocks or functional units, of a digital computer system. These five units correspond to the five basic operations performed by all computer systems. The function of each of these units is described below.

2.4.1 INPUT UNIT Data and instructions must enter the computer system before any computation can be performed on the supplied data. The input unit that links the external environment with the computer system performs this task. Data and instructions enter input units in forms that depend upon the particular device used. For example, data is entered from a keyboard in a manner similar to typing, and this differs from the way in which data is

27 entered through a card reader which is another type of input device. However, regardless of the form in which they receive their inputs, all input devices must provide a computer with data that are transformed into the binary codes that the primary memory of a computer is designed to accept. This transformation is accomplished by units called input interfaces. Input interfaces are designed to match the unique physical or electrical characteristics of input devices to the requirements of the computer system. In short, an input unit performs the following functions:

- 9 It accepts (or reads) the list of

instructions and data from the outside world.

- 9

It converts these instructions and data in

the

computer acceptable form. 9 It supplies the converted instructions and data to the complete system for further processing.

2.4.2 OUTPUT UNIT

The

job of an output unit is just the reverse of that of an input unit. It supplies information and results of computation to the outside world. Thus, it links the computer with the external environment. As computers work with binary code, the results produced are also in the binary form. Hence, before supplying the results to the outside world, it must be converted to human acceptable (readable) form. This task is accomplished by units called output interfaces. Output interfaces are designed to match the unique physical or electrical characteristics of output devices (terminals, printers, etc.) to the requirements of the external environment. In short, an output unit performs the following functions: 9 It accepts the results produced by the computer, which are in coded form and hence cannot be easily understood by us.

28 9 It converts these coded results to human acceptable (readable) form. 9 It supplies the converted results to the outside world. Figure 2-2 Basic Organisation of Computer System

2.4.3 STORAGE UNIT

The data and instructions that are entered into the computer system through input units have to be stored inside the computer before the actual processing starts. Similarly, the results produced by the computer after processing must also be kept somewhere inside the computer system before being passed on to the output units. Moreover, the intermediate results produced by the computer must also be preserved for ongoing processing. The storage unit at the primary/main storage of a computer system is designed to cater to all these needs. It provides space for storing data and instructions; space for intermediate results; and also space for the final results.

29 In short the specific functions of the storage unit are to hold (store): 9 All the data to be processed and the instructions required for processing (received from input devices). 9 Intermediate results of processing. 9 Final results of processing before these results are released to an output device.

Two Kinds of Memory

The main memory, housed inside the computer unit, is built from two different kinds of memory chip: the first kind, called ROM (read only memory), has permanently built into information and instructions the computer needs to know in order to operate properly; the second kind of memory, called RAM (random access memory), holds the program and other information typed in at the keyboard. The RAM is a 'read and write' memory. This means we can store, or 'write', information into this memory and later recall it, or 'read' it out again. The ROM, on the other hand, can only be read; we cannot write information into it. This ensures that we do not destroy the vital information held in ROM by over-writing it.

Figure 2-3 Two Kinds of Memory

A computer memory can be thought of as an array of letterboxes, each holding a single character (a string of 0s and 1s). We can put information into the RAM and take information out of it when we need to. The ROM on the other hand is like a letterbox covered with glass. We can see what is inside but we cannot touch or change the content

30 An important difference between the two types of memory is that RAM is 'volatile', i.e. it loses all the information stored when the power is switched off. ROM, on the other hand, is 'non-volatile'; its information is not lost when the power is switched off. The secondary storage medium stores data, instructions and output for archival purpose so that whenever any data or instructions is required in the future it can be retrieved for reference or for further processing.

2.4.4 CENTRAL PROCESSING UNIT

The Arithmetic Logic Unit and

the Control Unit of a computer system are jointly known as the Central Processing Unit (CPU). The CPU is the brain of any computer

system. In a human body, the brain takes all major decisions and the other parts of the body function as directed by the brain. Similarly in a computer system, all major calculations and comparisons are made inside the CPU and the

CPU is also responsible for activating and controlling the operations of other units of a computer system.

2.4.5 ARITHMETIC LOGIC UNIT

The

Arithmetic Logic Unit (ALU) of a computer system is the place where the actual execution of the instructions takes place during the processing operation. To be more precise all calculations are performed and all comparisons (decisions) are made in the ALU. The data and instructions stored in the primary storage prior to processing, are transferred as and when needed to the ALU where processing takes place. No processing is done in the primary storage unit. Intermediate results generated in the ALU are temporarily transferred back to the primary storage until needed at a later time. Data may, thus, move from primary storage to ALU and back again to storage many times before

31 the processing is over. After the completion of processing the final results, which are stored in the storage unit, are released to an output device. The type and number of arithmetic and logic operations that a computer can perform is determined by the engineering design of the ALU. However almost all ALU's are designed to perform the four basic arithmetic operations (add, subtract, multiply, divide) and logic operations or comparisons such as less than, equal to, or greater than.

CONTROL UNIT How does the input device know that it is time for it to feed data into the storage unit? How does the ALU know what should be done with the data once they are received? And how is it that only the final results are sent to the output device and not the intermediate result? All this is possible because of the Control Unit of the computer system. By selecting, interpreting, and seeing to the execution of the program instructions, the Control Unit is able to maintain order and direct the operation of the entire system. Although, it does not perform any actual processing on the data, the Control Unit acts as a central nervous system for the other components of the computer. It manages and coordinates the entire computer system. It obtains instructions from the program stored in main memory, interprets the instructions, and issues signals that cause other units of the system to execute them.

2.5 HOW DOES A COMPUTER WORK? The working of a computer can best be illustrated by an example. Suppose we have a routine clerical task to carry out and are given a willing helper who can follow instructions but who is new to the job. How would we proceed? Firstly, we would sit our helper at a desk and provide him with an 'in'-tray for material coming in (the input) and an 'out'-tray for material that had been processed (the output).

32 If the work involved calculations, we would also provide a desk calculator to help with the arithmetic, and a pad of paper for working things out. Next, we would put in our helper's 'in'-tray all the figures and information, such as prices and orders placed, needed as the 'raw material' to work on. This is the data. Figure 2-4 The Way in Which A Computer Works is Analogous to a Routine Helper Who has been Given Precise Instructions

33 Our willing helper is now ready to start work - processing the data - but first needs to be told what to do. He knows nothing about the job and so the instructions we give him will have to be very detailed and precise. Such a set of instructions is called a program. He would have to remember our more general instructions, and so memory would play an important part in the whole process. He would probably note down the program of detailed instructions on his pad so that he can quickly refer to it should he forget what he is expected to do. The note-pad, then, is a type of memory, used not only for calculations but also for storing the program. We would also give him some reference books, such as a catalogue of prices or a telephone directory, which might be needed from time to time. These provide a second kind of memory: they store information, like the note-pad, but the information is permanent and cannot be changed. The books are 'read only' memories (they can only be read), while the note-pad is a 'read and write' memory, since information can be written on the pad and changed or deleted when necessary. Finally, he might also need as a 'backing' memory - a secondary storage- a filing cabinet for those records, such as customer accounts, to which he may have to refer occasionally. It is now up to our helper to co-ordinate and control the various operations required by the program to complete the task. A computer works in much the same way as this and, incorporates the same basic features

The original informationData is fed into the computer, Input together with the instructions.
Program These are storedMemory

34 and the computer carries out any calculations required, ALU and supervises and checks the whole process. Control The completed answer is read out of the computer.Output There is in every computer system a section to carry out each of these operations, and the operations are combined to solve problems in the same way as they were combined to complete the clerical task.

2.6 SELF-ASSESSMENT QUESTIONS

1. What is a system? Why do we refer to a computer as a system?
2. What are the five basic functions performed by any computer system?
3. Draw a block diagram to illustrate the basic organization of a computer system and explain the functions of various units.
4. What are primary and secondary storage devices?
5. What do you mean by memory? What are ROM and RAM?
6. Explain the working of a computer system.

2.7 Suggested Readings

1. Fundamentals of Computers by V Raja Raman. Prentice Hall of India Pvt. Ltd., New Delhi.
2. Computer Fundamentals by P K Sinha. BPB Publications., New Delhi.
3. Computer Fundamentals by B Ram. New Age International.
4. Introduction to Computers by Leon and Leon. Vikas Publishing House., New Delhi.

35

UNIT 3 -

HISTORY AND CLASSIFICATION OF COMPUTERS Objectives: The main objective of this lesson is to study the history and classification of computers. After successful completion of the lesson the students will be able to understand how man's search for better and faster ways to count, write and communicate bring us to the modern 'Computer Age'. Structure

- 3.1 Introduction
- 3.2 Early Computing Devices
- 3.3 Early Electronic Computers
- 3.4 Modern Computers
- 3.5 Classification of Computers
- 3.6 Self Assessment Questions
- 3.7 Suggested Readings

36 3.1 INTRODUCTION The development of computers is the story of man's continuous search for better and faster ways to count, write and communicate. Although modern electronic computers are only a recent phenomenon, the ideas and devices leading to the advent of computers date far back in the history. Early man, when started to live a settled life in the stone age, used pebbles and stone for counting items, and used marking on the walls for storing the counts. The discovery of zero by Indian mathematician laid the foundation stone of the number system. We can identify three distinct stages in the evolution of computers from these simple ideas and devices to the complex and sophisticated devices as they are now: $\frac{3}{4}$ Early Computing Devices $\frac{3}{4}$ Early Electronic Computers $\frac{3}{4}$ Modern Computers

3.2 EARLY COMPUTING DEVICES The earliest device that qualifies, as a digital computer is the Abacus also known as Soroban This device permits the user to represent number by the position of beads on a rack. Simple addition and subtraction can be carried out rapidly and efficiently by positioning the beads appropriately. Although, the Abacus was invented around 600 BC, it is interesting to note that it is still used in the Far East and its users can calculate at amazing speeds. Another manual calculating device was Napier's Bones, designed by John Napier, a Scottish scientist. It used a set of eleven rods called 'bones' with numbers carved on them. It was designed in early 17th century and its upgraded versions were in use even around 1890. Pascaline, designed by Blaise Pascal in 1642, was the first mechanical calculating machine. Pascaline used a system of gears, cogwheels and dials for carrying

37 out repeated additions and subtractions. Later in 1671, Baron Gottfried von Leibitz of Germany developed a similar mechanical calculator that could also perform multiplication and divisions. In 1820, a similar key – driven mechanical calculator, called Thomas Arithmometer met with commercial success. It was developed by Charles Xavier Thomas of France. 19th century witnessed major advances in computing technology. An important breakthrough in the development of computing was the concept of stored program to control calculations. In the early 19th century a Frenchman, Joseph Jacquard invented a loom that used punched cards to automatically control the manufacturing of patterned cloth. Jacquard's idea of storing a sequence of instructions on the cards is also conceptually similar to modern computer programs. The year 1822 could be considered a golden year in the history of Computer Science. It was in the year 1822, when an Englishman, Charles Babbage, a Professor of Mathematics at Cambridge University, developed and demonstrated a working model of a mechanical computer called Difference Engine. It could do complex algebraic equations and was built on the principal of gearing wheels of earlier era. It was also able to produce mathematical and statistical tables correct up to 20 digits. Encouraged by success of Difference Engine; Babbage, in 1833, came out with his new idea of Analytic Engine. It could store 1000 numbers of 50 decimals each. It was to be capable of performing the basic arithmetic function for any mathematical problem and it was to do so at an average speed of 60 additions per minute. Unfortunately, he was unable to produce a working model of this machine mainly because the precision engineering required to manufacture the machine was not available during that period.

38 However, his efforts established a number of principles, which have been shown to be fundamental to the design of any digital computer. There were many features of the Engine - punched card instructions, internal memory and an arithmetic unit to perform calculation – that were incorporated in the early computer, designed a 100 year later. His disciple, a brilliant mathematician, Lady Ada Lovelace, the daughter of the famous English poet Lord Byron, developed the binary number system for Babbage's machine. She is widely considered as the first programmer in the world, and programming language ADA is named after her. Babbage's Engines were more closely related to modern computers than any of their predecessors. Many people today credit Charles Babbage with the real father of computers. Keyboard machines originated in the United State around 1880 and are extensively used even today. Around this period only, Herman Hollerith came up with the concept of punched cards, which are extensively used as input media in modern digital computers. Business machines and calculators made their appearance in Europe and America towards the end of the

nineteenth century. In 1880's the Census Bureau of the United States appointed Harman Hollerith to develop a technique for speeding up the processing of census data. It was the major breakthrough of 19th century when he developed a machine which used punched cards to store the census data. This innovation reduced the time of processing from 8 years to less than 3 years. In 1890's many other countries like Canada, Australia, and Russia also used Hollerith Machine for processing their census data. Later many other large organizations like Insurance Company used Hollerith Machine to speed up their processing activity. In

39 1890's Hollerith left the Census Bureau and started the Tabulating Machine Company, which, in 1911, merged with several other firms to form International Business Machine (IBM) Corporation. During the early part of the 20th century there was a flurry of activities in the computing field. Due to the onset of World War II, there was a great need for devices, which can produce ballistic tables very quickly. In the period of late 1930's and early 1940's many projects went underway. In this period, under the direction of George Stibitz of Bell Telephone Laboratories, five large-scale computers were developed. These computers were built using electromechanical relays and were called as Bell Relay Computers. These were capable to perform calculations with a high speed and accuracy. The world's first electro-mechanical computer was developed by Dr Howard Aiken of Harvard University and produced by IBM in the year 1944. This computer used more than 3000 relays, was 50 feet long and 8 feet high and was named as Automatic Sequence Controlled Calculator also called Mark-1. It was a quite fast machine and could add two numbers in 0.3 seconds and multiply two numbers in 4.5 seconds. This computer may be regarded as the first realization of Babbage's Analytical Engine. IBM went on to develop Mark -II through IV.

3.3 EARLY ELECTRONIC COMPUTERS

3.3.1 Atanasoff-Berry Computer(1939-42)

This electronic machine was developed by Dr John Atanasoff to solve certain mathematical equations. It was called Atanasoff-Berry Computer, or ABC, after its inventor's name and his assistant, Clifford Berry. It used 45 vacuum tubes for internal

40 logic and capacitors for storage.

3.3.2 ENIAC(1943-46)

The Electronic Numerical Integrator And Calculator (ENIAC) was the first all electronic computer. It was constructed at the Moore School of Engineering of the University of Pennsylvania, USA by a design team led by Professors J. Presper Eckert and John Mauchly.

ENIAC was developed as a result of military need. It took up the wall space in a 20 x 40 square feet room and it used 18,000 vacuum tubes. The addition of two numbers was achieved in 200 microseconds, and multiplication in 2000 microseconds. Although, much faster in speed as compared to Mark I computer, ENIAC had two major shortcomings: it could store and manipulate only very limited amount of information, and its programs were wired on boards. These limitations made it difficult to detect errors and to change the programs. Hence its use was limited. However, whatever be the shortcomings of ENIAC, it represented an impressive feat of electronic engineering and was used for many years to solve ballistic problems.

3.3.3 EDVAC(1946-52)

The operation of ENIAC was seriously handicapped by the wiring board. This problem was later overcome by the new concept of stored program developed by Dr John V Neumann. The basic idea behind the stored program

concept is that a sequence of instructions as well as data can be stored in the memory of the computer for the purpose of automatically directing the flow of operations. The stored program feature considerably influenced the development of modern digital computers and because of this feature we often refer to modern digital computers as stored program digital computers.

41 The Electronic Discrete Variable Automatic Computer (EDVAC)

was designed on stored program concept. Von Neumann has also got a share of the credit for introducing the idea of storing both instructions and data in the binary form instead of the

decimal numbers or human readable words.

3.3.4 EDSAC(1947-49)

Almost simultaneously with EDVAC of USA, the Britishers developed Electronic Delay Storage Automatic Calculator (EDSAC). The machine executed its first program in May 1949. In this machine, addition operation was accomplished in 1500 microseconds, and multiplication operation in 4000 microseconds. The machine was developed by

a group, of scientists headed by Professor Maurice Wilkes of the Cambridge University.

3.3.5 MANCHESTER MARK-I (1948)

This computer was a small experimental machine based on the stored program concept. It was designed at Manchester University by a group of scientists headed by Professor M H A Newman. Its storage capacity was only 32 words, each of 31 binary digits. This was too limited to store data and instructions. Hence, the Manchester Mark I was hardly of any practical use.

3.3.6

UNIVAC -1(1951)

The Universal Automatic Computer (UNIVAC) was the first digital computer, which was not "one of a kind". Many UNIVAC machines were produced, the first of which was installed in the Census Bureau in 1951 and was used continuously for 10 years. The first business use of a computer, a UNIVAC-I, was by General Electric Corporation in 1954.

In 1952, the International Business Machines (IBM) Corporation introduces the 701 commercial computers. In rapid succession, improved models of the UNIVAC I and

42 other 700-series machines were introduced. In 1953, IBM produced the IBM-650 and sold over 1000 of these computers.

The commercially available digital computers

that could be used for business and scientific applications had arrived. During the period of late 1940's and early 1950's many other stored program computers like ILLIAC, JOHNIAC, MANIAC etc. were developed. With these developments in the computing field and advancement in technology, the stage was set for modern computers. 3.4 THE MODERN COMPUTERS Early electronic computers were exclusively used for the military, experimental and engineering purposes. In the early 1950's computers began to be sold commercially. The development of commercial computer industry was really the beginning of the computer revolution. The modern computers era can be divided into four generations distinguished by the basic electronic component within the computer. Each new logic unit has led to computers that are faster in speed, smaller in size, more reliable and less expensive than their predecessors. Modern computers came in a variety of shapes, sizes and costs. 3.4.1 The Computer Generations "

Generation" in computer talk is a step in technology. It provides a framework for the growth of the computer industry.

Originally,

the term "generation" was used to distinguish between varying hardware technologies.

But nowadays,

it has been extended to include both

the hardware and the software together that make up an entire computer system. The custom of referring to the computer era in terms of generations came into wide use

43 only after 1964. There are totally five computer generations known till today. Although there is a certain amount of overlapping between the generations, the approximate dates shown against each are normally accepted.

Figure 3-1 Electronic Devices used for Manufacturing Computers of Different Generations 3.4.1.1

First-Generation Computers (1942-1955) First-generation computing involved massive computers using hundreds or thousands of vacuum tubes for their processing and memory circuitry. These large computers generated enormous amounts of heat; their vacuum tubes had to be replaced frequently. Thus, they had large electrical power, air conditioning, and maintenance requirements. First-generation computers had main memories of only a few thousand characters and millisecond processing speeds. They used magnetic drums or tape for secondary storage.

44 Examples of some of the popular first generation computers include ENIAC, EDVAC, UNIVAC-I, IBM-701, IBM-650,

and IAS Machine. Figure 3-2 Three Stages of Development 3.4.1.2 Second-Generation Computers (1955-1964) Second-generation computing used transistors and other solid-state, semiconductor devices that were wired to circuit boards in the computers. Transistorized circuits were much smaller and much more reliable, generated little heat, were less expensive, and required less power than vacuum tubes. Tiny magnetic cores were used for the com- puter'

s memory, or internal storage. Many second-generation computers had main memory capacities of less than 100 kilobytes and microsecond processing speeds.

45 Removable magnetic disk packs were introduced, and magnetic tape emerged as the major input, output, and secondary storage medium for large computer installations. Examples of some of the popular second-generation computers include IBM-1620, IBM- 7094, CDC-1604, CDC-3600, UNIVAC-1108, PDP-I and NCR-304. 3.4.1.3 Third-Generation Computers (1964-1975) Third-generation computing saw the development of computers that use integrated circuits, in which thousands of transistors and other circuit elements are etched on tiny chips of silicon. Main memory capacities increased to several megabytes and processing speeds jumped to millions of instructions per second (MIPS) as telecommunications capabilities became common. This made it possible for operating system programs to come into widespread use that automated and supervised the activities of many types of peripheral devices and processing by mainframe computers of several programs at the same time, frequently involving networks of users at remote terminals. Integrated circuit technology also made possible the development and widespread use of small computers called minicomputers in the third computer generation. Examples of

some of the popular third-generation computers include IBM-360 Series, IBM-370 Series,

HCL-2900 Series, Honeywell-6000 Series, PDP-8 and VAX. 3.4.1.4 Fourth-Generation Computers (1975-2000) Fourth-generation computing relies on the use of LSI (large-scale integration) and VLSI (very-large-scale integration) technologies that cram hundreds of thousands or millions of transistors and other circuit elements on each chip. This enabled the development of microprocessors, in which all of the circuits of a CPU are contained on a single chip with processing speeds of millions of instructions per second. Main memory capacities

46 ranging from a few megabytes to several gigabytes can also be achieved by memory chips that replaced magnetic core memories. Microcomputers, which use microprocessor CPUs and a variety of peripheral devices and easy-to-use software packages to form small personal computer (PC), systems or client/server networks of linked PCs and servers, are a hallmark of the fourth generation of computing, which accelerated the downsizing of computing systems. Examples of some of the popular fourth-generation computers include DEC-10, STAR- 1000, PDP-II, CRAY-I (Supercomputer), CRAY-X-MP (Supercomputer), CRAY-2 and IBM PC/AT.

3.4.1.5 Fifth-Generation Computers (2000-...)

Computer scientists and engineers are now talking about developing fifth -generation computers, which can 'think'. The emphasis is now shifting from developing reliable, faster and smaller but 'dumb' machines to more 'intelligent' machines. Fifth-generation computers will be highly complex knowledge processing machines. Japan, USA and many other countries are working on systems, which use Artificial Intelligence. Automatic Programming, Computational Logic Pattern Recognition and Control of Robots are the processes, which are used in these computers. The speed of the computers will be billions of instructions per second, and will have unimaginable storage capacities. These computers will be interactive. Main characteristics: 9 will be able to do multiple tasks, simultaneously 9 will have a parallel structure as compared to the serial structure of fourth generation

47 9 will not be algorithmic 9 will be knowledge processing and not data processing and architecture will be KIPS (Knowledge Information Processing System) 9 applications will be based on Expert Systems 9 will interact with user in human language 9 very cheap, super speeds 9 decision-making capabilities

Figure 3-3 Major Trends in Computer System Capabilities Figure 3-3 highlights trends in the characteristics and capabilities of computers. Notice that computers continue to become smaller, faster, more reliable, less costly to purchase and maintain, and more interconnected within computer networks. In terms of the analogy with the automobile industry, if the automobile industry had grown like the 48 computer industry, the cost of a Rolls Royce would be around Rs 80 and would run a million miles on a litre of petrol but the size would be that of a match box. Even though computers in the last 50 years have become very fast, reliable and inexpensive, the basic logical structure proposed by Von Neumann has not changed. The basic block diagram of a CPU, memory and I/O is still valid today. With the improvements in integrated circuit technology, it is now possible to get specialized VLSI chips at a low cost. Thus an architecture, which makes use of the changes in technology and allows an easier and more natural problem solving, is being sought.

3.4.2 Moore's Law

In 1965 Gordon E Moore predicted, based on data available at that time that the density of transistors in integrated circuits will double at regular intervals of around 2 years. Based on the experience from 1965 to date, it has been found that his prediction has been surprisingly accurate. In fact the number of transistors per integrated circuit chip has approximately doubled every 18 months. The observation of Moore has been called "Moore's Law". In Figure 3-4 we have given two plots. One gives the number of transistors per chip in Dynamic Random Access Memory along the y-axis and years along x-axis. Observe that the y-axis is a logarithmic scale and the x-axis a linear scale. The second plot gives the number of transistors in microprocessor chips. Observe that in 1974 the largest DRAM chip had 16 Kbits whereas in 1998 it has 256 Mbits, an increase of 16000 times in 24 years. The increase in the number of components in microprocessors has been similar. It is indeed surprising that the growth has sustained over 30 years. The availability of large memory and fast processors has in turn increased the size and complexity of systems and applications software. It has been observed that

49 software developers have always consumed the increased hardware capability faster than the growth in hardware. This has kept up the demand for hardware. Figure 3-4 Moore's Law Another interesting point to be noted is the increase in disk capacity. In 1984 disk capacity in PCs was around 20 MB whereas it was 20 GB in 2000 - a 1000 fold increase in about 16 years again doubling every 18 months, which is similar to Moore's law. These improvements in capacity of computers have come about with hardly any increase in price. In fact the price of computers has been coming down. The implication of Moore's law is that in the foreseeable future we will be getting more powerful computers at reasonable cost. It will be up to our ingenuity to use this increased power of computers effectively. It is clear that a number of applications such as speech recognition - voice and video user interfaces which require large amount of memory and computing power will be extensively used.

50 3.5 CLASSIFICATION OF COMPUTERS The various generations of computers actually show the development in the computers from the early stages. But, even today, all the computers are not of the same type. Computers come in many different sizes and ranges of power, and different types of computer systems have varying capabilities. Computers can be divided into following categories by functional criteria (data representation): $\frac{3}{4}$ Digital Computers $\frac{3}{4}$ Analog Computers $\frac{3}{4}$ Hybrid Computers

3.5.1.1 Digital computers A digital computer, as the name suggests, works with digits. In other words, a digital computer is a counting device. All the expressions are coded into binary digits (0s and 1s) inside the computers and it maintains and manipulates them at a very fast speed. A digital computer can perform only one operation i.e. addition. The other operations of subtraction, multiplication and division are performed with the help of addition operation. The digital computer circuits are designed and fabricated by the manufacturers and are quite complicated ones. A digital computer manipulates data according to the instructions (program) given to it in a certain language. The instructions and data are fed to the computer in the form of discrete electrical signals. Digital computers are further divided into two categories: 1. General Purpose Computers: These computers are designed for use in different types of applications in different areas. They can be used to prepare pay bills, manage inventories, print sales or can be used to solve mathematical equations.

51 When one job is over another job can be loaded into the memory for processing. They are versatile; hence most businesses today use general-purpose computers. 2. Special Purpose Computers: The digital computers, which are made to meet requirements of a particular task or job, are called special purpose computers. For example, computers used for weather forecasting or for space applications. They are also known as dedicated computers. The typical special purpose computers are: (a) Word Processor: This computer is most versatile for office automation purposes and replaces the typewriters. It is widely used for the production of office documents, letters, contracts, pay bills etc. It is a computer that has to deal with the bulk of input, in production of bulk of printed output, not involving too many calculations or connecting programs. It also works as a duplicating machine with a very high speed and facilitates seeing through the old and other connected documents as and when desired. (b) Hidden Computers: Such computers are designed to control particular process or job and as such are installed inside the machine being automatised. These are digital computers with hybrid applications. LAN and WAN, using the electronic message switching systems employ such computer robots in intensive care units and other hospital instruments. Automatic washing machines, digital clocks, Hot shot camera use these computers.

52 (c) Optical Computers: These are under process of design and development. Here the concept of application of quadral logic in place of binary logic and use of optical fibre technology is being used. (d) Hand Held and Pocket Computers: These are very small in size with memory of 4 to 16 Kb. They are used for personal and scientific computing. (e) Knowledge Information Processing Systems (KIPS): They process information and not data. They incorporate artificial intelligence, (ability of reasoning, thinking and decision making). In such computers, hardware and software will work in parallel memory operations having capabilities of 100 million to a billion lips (logical interferences per second). A lip is equal to 100 to 1000 instructions per second. (f) Dedicated Word Processor: It is used for office automation. It is widely used for production of office documentation, letters, memos, contracts, pay bills etc. 3.5.1.2 Analog Computers Analog computers represent numbers by a physical quantity i.e. they assign numeric values by physically measuring some actual property, such as the length of an object, or the amount of voltage passing through a point in an electric circuit. Analog computers derive all their data from some form of measurement. Though effective for some applications, this method of representing numbers is a limitation of the analog computers. The accuracy of the data used in an analog computer is directly related to the precision of its measurement. Speedometers, Voltmeters, Pressure Gauges, Slide Rules, Flight Simulators for training pilots and Wall Clocks are some examples of analog computers

53 3.5.1.3 Hybrid Computers Hybrid computers combine the best features of analog and digital computers. They have the speed of analog computers and accuracy of digital computers. They are usually used for special problems in which input data derived from measurement is converted into digits and processed by computer. Hybrid computers for example, control National Defense and Passenger flight radars. Consider the Hybrid computer system in a hospital Intensive Care Unit (ICU). The analog device may measure a patient heart function, temperature and other signs. These measurements may, then, be converted into numbers and supplied to a digital device, which may send as immediate signal to a nurse's station if any abnormal readings are detected. To take another example of Hybrid computer, consider the system used in producing iron ore pellets for steel making. It controls manufacturing and prepares production data on inventory and costs. The computer accepts data both from sensors within the production area and from conventional Input/Output devices. Using the production data, the computer plans for future manufacturing and distributes existing inventories to activities that require the computer to act as digital computer. The computer can act like an analog computer converting measurements into numeric numbers. It can act as a digital computer processing stored data for management. We can also classify the computer systems into following categories by using the capacity performance criteria (size, cost, speed and memory): $\frac{3}{4}$ Supercomputers $\frac{3}{4}$ Mainframe computers

54 $\frac{3}{4}$ Minicomputers, or Midrange computers $\frac{3}{4}$ Workstations $\frac{3}{4}$ Microcomputers, or Personal computers All of these computers can be connected to form networks of computers, but each individual computer, whether or not it is on a network, falls into one of these five categories. As we will see, some of these categories-especially microcomputers-can be divided into subcategories, some of which are growing rapidly enough to become major categories in their own right.

3.5.2.1 Supercomputers Supercomputers are the most powerful computers made, and physically they are some of the largest. These systems are built to process huge amounts of data, and the fastest supercomputers can perform more than trillion calculations per second. Some supercomputers such as the Cray T-90 system; can house thousands of processors. This speed and power make supercomputers ideal for handling large and highly complex problems that require extreme calculating power such as numerical weather prediction, design of supersonic aircraft, design of drugs and modeling of complex molecules. Recently the use of supercomputers has expanded beyond scientific calculations. They are now used to analyse large commercial databases, produce animated movies and play games such as chess. Super computers can cost tens of million of dollars and consume enough electricity to power dozens of homes. They are often housed in protective rooms with special cooling systems, power protection, and other security features. Because of their size and cost, supercomputers are relatively rare, used only by large corporations, universities, and 55 government agencies that can afford them. Supercomputing resources are often shared to give researchers access to these precious machines. Figure 3-5 The IBM RS/6000SP: Super Computer that beat the World Chess Champion Garry Kasparov in May 1997

3.5.2.2 Mainframe Computers The largest type of computer in common use is the mainframe. Mainframe computers are used in large organizations like insurance companies and banks where many people need frequent access to the same data, which is usually organized into one or more huge databases. Mainframes are being used more and more as specialized servers on the World Wide Web, enabling companies to offer secure transactions with customers over the Internet. If we purchase an air line ticket over the Web, for example, there is a good chance that our transaction is being handled by a mainframe system. In this type of application, the 56 mainframe system may be referred to as an enterprise server or an electronic commerce (e-commerce) server. Figure 3-6 The IBM S/390: Mainframe Computer System In a traditional mainframe environment, each user works at a computer terminal. A terminal is a monitor and a keyboard (and sometimes a pointing device, such as a mouse) wired to the mainframe.

There are basically two types of terminals used with mainframe systems.

A dumb terminal does not have its own CPU or storage devices; these components are housed in the mainframe's system unit and are shared by all users. Each dumb terminal is simply an input/output (I/O) device that functions as a window into a computer located somewhere else. An intelligent terminal, on the other hand, has its own processor and can perform some processing operations.

Intelligent terminals, however, do not usually provide any storage. Many enterprises are now connecting personal computers and personal computer networks to their mainframe systems. This connection gives users access to mainframe data and services and also enables them to take advantage of local storage and process-

ing, as well as other features of the PC or network.

57 Figure 3-7 The IBM Netfinity 7000: A Powerful High-end Server designed for High-Volume Electronic Processing at Web Sites A mainframe system can house an enormous volume of data, containing literally billions of records.

Large mainframe systems can handle the input and output requirements of several thousand terminals. The IBM S/390 mainframe, for example, can support 50,000 users simultaneously while executing more than 1,600,000,000 instructions per second.

It used to be common for mainframe computers to occupy entire rooms or even an entire floor of a high-rise building. Typically, they were placed inside glass offices with special air conditioning to keep them cool and on raised floors to accommodate all the wiring needed to connect the system. This setup is not used much anymore. Today, a typical mainframe computer looks like an unimposing file cabinet-or a row of file cabinets- although

it may still require a somewhat controlled environment.

3.5.2.3 Mini Computers First released in the 1960s, minicomputers got their name because of their small size compared to other computers of the day. The capabilities of minicomputer are between

58 that of mainframes and personal computers. (For this reason, minicomputers are increasingly being called midrange computers). Like mainframes, minicomputers can handle much more input and output than personal computers can.

Figure 3-8 Midrange Computer Systems: (a) The IBM AS/400 (b) The HP 3000

Although some "minis" are designed for a single user, most are designed to handle multiple terminals

in a network environment that handle the data sharing needs of other computers on the network. The most powerful minicomputers can serve the input and output needs of hundreds of users at a time. Single-user minicomputers are commonly applied to sophisticated design tasks such as animation and video editing. 3.5.2.4 Workstations Somewhere between multi-user midrange computers and personal computers are workstations. Workstations are specialized, single-user computers with many of the features of a personal computer but with the processing power of a minicomputer. These powerful machines are popular among scientists, engineers, graphic artists, animators, and programmers—users that need a great deal of number-crunching power. Workstations

59 typically use advanced processors and feature more RAM and storage capacity than personal computers.

Workstations often have large, high-resolution monitors and accelerated graphics-handling capabilities, making them perfect for advanced design, modeling, animation, and video editing. Although workstations are often found in single-user

applications, they are more and more used as servers on personal computer networks and as Web servers. Figure 3-9

The Powerful Ultra Workstation from Sun Microsystems Until a few years ago, the term workstation implied certain differences in terms of chip design and operating system, making it distinct from a personal computer. (The term workstation is also used to describe a single computer on a network. In this context; a workstation is usually a personal computer.) Today, the differences between minicomputers, workstations, and personal computers are becoming blurred. Low-end minicomputers and high-end workstations are now similar in features and capabilities. The same is true for low-end workstations and high-end personal computers. Some manufacturers of workstations are Silicon Graphics (SIG), Digital Equipment Corporation (DEC), IBM, SUN Microsystems and Hewlett Packard (HP). The standard

60 Operating System in workstations is UNIX and its derivatives such as AIX (IBM), Solaris (SUN) and HP-UX (HP). 3.5.2.5 Microcomputers or Personal Computers The terms microcomputer and personal computer are interchangeable, but PC, which stands for personal computers— sometimes, has a more specific meaning. In 1981, IBM called its first microcomputer the IBM-PC. Within a few years, many companies were copying the IBM design, creating “clones” or “compatibles” that were meant to function like the original, for the reason, the term PC has come to mean the family of computers that includes IBMs and IBM compatibles. The vast majority of microcomputers sold today are part of this family. One source

of the PC’s popularity is the rate at which improvements are made in its technology. Microprocessors, memory chips, and storage devices

make continual gains in speed and capacity, while physical size and price remain stable – or in some cases are reduced.

For example, compared to the typical PC of ten years ago, a machine

of the same price today will have about ten times as much RAM, about 100 times more storage capacity, and a microprocessor at least 100 times

as fast. What’s more, many analysts believe that this pace of change will continue for another 10 or 20 years. The

microcomputer category has grown tremendously in the past decade. There are now several specific types of microcomputers, each with its own capabilities, features, and purposes. Within each subcategory of microcomputer, we can find dozens or even hundreds of unique systems. This range of options makes the microcomputer “the computer for the masses” and explains why so many systems have appeared in offices,

61 homes, briefcases, and even pockets over the past few years. Microcomputers include the following types: 1. Desktop Models including tower models 2. Notebook Computers, also called Laptop Computers 3. Network Computers 4.

Handheld Personal Computers (H/PCs) of all types 1. Desktop Models The first style of personal computer introduced was the desktop model. In common usage, the term desktop system means a full size computer that is small enough to be used at a desk but too big to carry around. Traditionally, a desktop computer’s main case (called the system unit) is horizontally oriented, meaning it can lie flat on a desk or table. (a) (b) Figure 3-10 Two common Designs For Personal Computers A variation of the desktop system is the tower model, where the system unit sits vertically and has more space for devices. Because of its design, the system unit is often placed on

62 the floor to preserve desk space, allowing more room to place external components, such as removable disk drives or scanners, on the desktop. Tower models have become increasingly popular in recent years – so much so that some PC maker has stopped offering horizontally oriented desktop system.

2. Notebook Computers, also called Laptop Computers Notebook computers are small, easily transportable, lightweight microcomputers that fit easily into a briefcase. Laptops and notebooks are designed for maximum convenience and transportability, allowing the users to have access to processing power and data without being bound to the office environment. The ability of providing Internet access anytime anywhere is an additional benefit of these computers.

3. Network Computers Network computers are major new microprocessor category designed primarily for use with the Internet and corporate Intranet by clerical workers, operational employees and knowledge workers with specialized or limited computing applications. Network computers are low cost, sealed, networked microcomputers with no or minimal disk storage. Users of network computers depend primarily on Internet and corporate Intranet servers for their operating system, and web browsers, Java-enabled applications software and data access and storage. SUN'S JAVA Station, IBM'S Network Station and NCD Explora Network Computers are some examples of network computers.

4. Handheld Personal Computers (H/PCs) Since the mid -1990s, many new types of small personal computing devices have been introduced, and all fall under the category of handheld personal computers (H/PCs). These tiny systems are also called palmtop computers.

63

Figure 3-11 Examples of Micro Computers (a) Using a Notebook (

b) Using a H/PC System (c) Microcomputer as a Professional Workstation (d) Microcomputer as a Technical Workstation

(e) Using a Notebook PC (f) A Multimedia PC (g) A Multimedia PC in a Video Conferencing Session (

h) A Mobile Computer by Nokia (i) Network

Computers

64 A handheld PC can be any sort of computer that fits in the user's hand, such as: 9 Personal Digital Assistant (PDA) 9

Cellular phone with internet, e-mail and fax capabilities 9 H/PC Pro Device (a) Personal Digital Assistants (PDAs) Personal

digital assistants (PDAs) are among the smallest of portable computers. Often they are no larger than a small

appointment book, but they are much less powerful than notebook or desktop computers. PDAs are normally used for

special applications, such as taking notes, displaying telephone numbers and addresses, and keeping track of dates or

agendas. Many PDAs can be connected to larger computers to exchange data. Depending on the model, PDAs may

include the following features. ---Build in microphone and speaker, enabling the user to record speech digitally ---

Personal information management (PIM) software ---Miniaturized versions of personal productivity applications ---

Internet, fax, or e-mail software (b) Cellular Phones Some new cellular phones are doubling as miniature PCs. Advanced

cellular devices combines analog and digital cell-phone service with e-mail capabilities. Such phones enable the user to

check and send e-mail and faxes over the phone. They offer features not normally found on a phone, such as personal

organizers or access to the Web. Some models even break in half to reveal a miniature keyboard. (c) H/PC Pro Devices

Probably the most curious new development in handheld technology is the H/PC Pro family of devices. These systems

are larger than PDAs or miniature notebooks, but they

65 are not quite as larger as typical notebook PCs, with features somewhere between the two. For example, H/PC Pro

systems boast nearly full-size keyboards and color displays. They can run more types of miniaturized applications than

their smaller counterparts, but those applications still do not provide the features of normal desktop software. H/PC Pro

units offer long battery life and instant on access (features still missing from many laptop systems), but they do not

include disks, Although they will gain speed and storage capacity quickly, H/PC Pro systems offer very limited RAM and

relatively slow processor speeds. 3.6 Self Assessment Questions 1. Write an essay on history of computers. 2. What is

meant by "generations" in computer terminology? How many computer generations are there till now? 3. List out the

various computer generations along with their basic characteristics. 4. Write a short note on fifth generation computing.

5. What are different categories of digital computers? 6. What are super computers and where are they used? 7. What is a

mainframe? What are its typical applications? 8. What is difference between minicomputer and microcomputer? 9. Write

a detailed note on personal computers. 3.7 Suggested Readings 1. Fundamentals of Computers by V Raja Raman.

Prentice Hall of India Pvt. Ltd., New Delhi. 2. Computer Fundamentals by P K Sinha. BPB Publications., New Delhi.

66 3. Computer Fundamentals by B Ram. New Age International. 4. Introduction to Computers by Leon and Leon. Vikas

Publishing House., New Delhi.

67 UNIT 4 -

COMPUTER

LANGUAGES

Objectives : After going through this chapter, you will be able to understand-the concept of

low level computer languages; concept of high level computer languages; programming concepts in high level computer languages and elementary idea of assembler, interpreter, compiler etc. Structure 4.1. Introduction 4.2. Machine language 4.3. Symbolic/assembly languages 4.4. High level languages 4.5. Assemblers, compilers and interpreters 4.6.

Self-test questions

68 4.1. Introduction A language usually consists of all the verbal or written symbols, grammatical rules and expressions that are used to exchange ideas and information. Communications can take place between individuals or between an individual and a computer. The two BASIC types of languages used in computers are low-level language and high-level languages. Low-level languages are again subdivided into machine and assembly languages. High-level languages may be general purpose or special purpose. They may be either compiler based or interpreter based. High-level languages look similar to our English language and hence are better understood by the programmers rather than the computers. Low-level languages are compatible with the hardware of the computer and consist of binary or mnemonic codes. Hence, low-level languages are understood by the computers in a better manner. It is difficult for the programmer to understand the low level language. 4.2. Machine language The lowest form of computer language is machine language. In the first generation computers, programs were written only in binary based machine level languages. It has to be remembered that the only language that is understood by the computer is the machine language. However, almost all the programs that are written today are in high level languages. Since computers can understand only machine language, these programs (written in higher level languages) are translated to machine level languages using a software program called

compiler/interpreter.

69

Machine language is the most BASIC form of programming and hence explicit instructions are to be given to the machine to perform each operation. Hence, it is necessary to tell the machine where to store numbers in its memory and how to add the numbers and so on. A statement of the form $P = Q + R$ is meaningless in a machine code. If you want to add Q with R , then you have to instruct the machine exactly how to transfer and process the numbers, where to store the result P , etc. Any set of instructions in a machine level language can be divided into the following four categories: Arithmetic, control, input-output and direct use. The functions of these statements are as follows:

Arithmetic : Add, subtract, multiply and divide
 Control : Load, store and jump
 Input-output : Read and write
 Direct use : Start, halt and end

The arithmetical operations or comparisons are done in the arithmetic logic unit with the use of accumulators. Hence, if you want to add two numbers together, we will need one instruction which will order the control unit to place a number in an accumulator and another instruction to identify the addition operation. Machine language instructions are represented by binary numbers, i.e., sequences consisting of 0s and 1s. For example, the binary sequence. 001010011010 could represent a 12 bit machine language instruction. The instruction is divided into two parts: operation code (or Op code) and an operand. Op code Operand 0010 10011010 70 The Op code specifies the operations such as add, multiply, etc. and the operand is the address of the data item that is to be operated upon. Thus, while using a machine language, we have to remember the code numbers for the operations and also keep track of the addresses of all the data items. Hence, machine language is highly complicated and complex and subject to error. Also, the programs written in machine language are machine dependent. That means, a program that has been developed for a particular machine cannot be run on another machine. It is clearly seen from the above program that coding a program in the binary form is very tedious. Hence, machine language is often coded in hexadecimal codes, which are still quite tedious. The programs written in machine language are also called as Object programs.

4.3. Symbolic/assembly languages

To overcome the difficulties of writing programs in machine code, symbolic languages (which are also called as assembly languages) were developed in 1950s and were used extensively in some of the second generation computers. These languages enabled instructions to be written using symbolic codes (called mnemonics) rather than in strings of 0s and 1s. One word (or mnemonic) represented a particular machine language instruction. Once a series of instructions or words were put together to form a program, they are translated (by referencing a symbolic equivalence table) by the computer into their machine language counterparts. Let us assume that the following symbolic operation codes are used (The meaning of the mnemonics are given in brackets). 0001 ADD (Addition) 0010 SUB (Subtraction) 0100 DIV (Division) 0101 LDA (Load accumulator with value) 0110 STO (Store contents of accumulator) 0111 IN (Read a value into accumulator) 1000 OUT (Output value in accumulator to output device) 1001 JUN (Jump unconditionally) 1010 JGT (Jump to the address location if the contents of the accumulator are greater than zero) 1011 JNE (Jump to the address location if the contents of the accumulator are not equal to zero). It is clearly seen that symbolic language program is much simpler than the program written in the machine code. Yet assembly language programming has many of the same advantages as machine language programming; i.e., the program has access to and control over the registers, I/O ports and other features of the computer. However, the BASIC disadvantages of having to work with binary, octal or hexadecimal numbers are eliminated. The program that has been written using a symbolic language has to be translated into a machine language and only then it can be executed by the computer. The software program that will translate the program written in symbolic language is called as the assembler program or 72 assembler. The source program acts as an input to the assembler, which is loaded in the computer memory. Then, the assembler performs the translation and generates the equivalent machine code, which is called as the object program or object code. Once the object program is generated, the computer can execute the program without the aid of the assembler program. In typical computer systems, the assembler is stored in a secondary storage device and when needed loaded into the Random Access Memory (RAM). The source program is then entered by the user through the visual display unit and is also stored in RAM. The assembler translates the source program into object program and stores it in another portion of RAM. The object code or program can also be stored on some external device in order to execute it again and again when needed. In addition to using mnemonics to represent instructions, decimal numbers or names can also be used to refer to addresses of data names. By having such a system, the programmer is relieved of the problem of remembering each absolute address. Such names can have three or four characters and may use alphabets or numbers. However, the first character should be an alphabet. Thus, using symbolic names, the symbolic program may be written as IN STO Q IN STO R LDA R ADD Q STO P LDA P OUT 73 The symbolic language is called as low-level language because it is designed for a particular machine. It cannot be developed without knowing the size of the main memory and the size of the location word.

4.4. High level languages

Unlike symbolic languages, high-level languages can be used with a number of different hardware makes with little or no modifications. Hence, the programming and reprogramming expense is greatly reduced when the program has to be translated to another machine. Other

advantages of high-level languages are as follows: 1. They are easier to learn than symbolic languages. 2. They require less time to write. 3. They are simpler in formats. 4. They are easier to maintain. 5. They are easier to understand. 6. They provide better documentation. 7. The programs written in such a language can be executed on any computer. 8. Four or five level instructions are reduced to a single high-level language statement. Some of the popular high-level languages are given in Table 4.1. Table 4.1. High Level Languages Languages Year Meaning Developed By Main application area BASIC 1965 Beginners All Kemeny & Teaching 74 Purpose Symbolic Instruction Code

Kurtz scientific and also commercial. FORTRAN 1956 Formula Translation IBM Scientific and Engineering COBOL 1958 Common Business Oriented Language US Dept. of Defence Commercial ALGOL 1958 Algorithmic Language International Committee Scientific PL/1 1963 Programming Language 1 IBM Scientific, Commercial and Systems PASCAL 1968 Named after the French Philosopher PASCAL N. Wirth Teaching C 1970 Programming language Dennis Ritchie General purpose

Obviously, a source program written in a high level language should also be translated into a machine language. The program that performs this operation is called the compiler (which is sometimes also called as a compiler program) or interpreter. High-level languages may be further subdivided into procedure oriented languages, problem-oriented languages and interactive programming languages. Procedure-oriented languages are useful for some special applications. For example, COBOL is a procedure-oriented language, which is used extensively in business applications. Other examples of procedure-oriented languages are FORTRAN and PL/1. Problem-oriented language just attempts to solve processing requirements with minimal programming effort allowing the user to focus on what results are desired rather than on the individual steps needed to get 75 those results. A typical example of problem-oriented language is RPG (Report Program Generator). Interactive programming languages allow the user to interact with the program in a conversational fashion. These languages are quite useful especially in computer aided design/computer aided manufacture (CAD/CAM). BASIC, PASCAL and APL are typical examples of interactive languages. High-level languages are sometimes classified as general purpose and special purpose languages. General-purpose languages (e.g. BASIC and PASCAL) are suitable for any type of application whereas special purpose languages (e.g. COBOL and LISP) are used only in some special application areas. Though there are about 150 computer languages, only a few are used extensively. In India, the languages which are used widely are : BASIC, FORTRAN (used mainly in scientific and engineering applications), COBOL (mainly used in business applications) and PASCAL. We shall consider some of the features of these four high level languages now.

4.4.1. BASIC language

BASIC is an acronym for Beginners All Purpose Symbolic Instruction Code. This language was developed as a teaching aid for computing at Dart-mouth College, USA, in 1965 by Professors Kemeny and Kurtz. BASIC is a simple and "friendly" language, easy to learn and particularly suitable for the non-specialist user. It also provides preparations for using more powerful languages like FORTRAN and PL/1. BASIC is widely used for commercial, scientific and educational purposes. BASIC has a fairly limited repertoire of statements and unlike any other high level language, allows only a certain range of identifiers to be used for the variables in the program. Hence, it is mainly used as an introductory language. However, after the introduction of micro and home computers, which use BASIC as the only language, a number of application programs in different areas have been written in BASIC. BASIC is oriented to direct access use and programs are usually submitted to the computer by means of a console or visual display. Some of the commonly used statements are READ, DATA, INPUT, FOR-NEXT, IF-THEN, DIM etc. In BASIC, a variable name is limited to a single letter followed by a digit or special characters. The READ statement in BASIC is always associated with a DATA statement. The IF-THEN statement is used for branching. The FOR-NEXT statement of BASIC is used to perform looping. Though BASIC compilers do exist, most of the installations use BASIC interpreter. Let us now consider an example program written in BASIC for the following problem which gives a fair idea about the structure of a BASIC program. The following program reads 100 positive numbers and outputs the highest and the sum of these numbers: 5 REM M is a variable used here to store the highest and T is used to store sum. 10 LET M = 0 20 LET T = 0 30 FOR I = 1 TO 100 40 PRINT "TYPE IN NUMBER" 50 INPUT N 60 IF N > M THEN 80 70 LET M = N 80 LET T = T + N 90 NEXT I 77 100 PRINT "THE MAXIMUM NO IS" ; M 110 PRINT "THE SUM OF THE NUMBERS =" ; T 120 END

4.4.2. FORTRAN

The name FORTRAN stands for formula translation and as the name suggests it was developed and used as a scientific application language. FORTRAN is the oldest of the high level languages. It was introduced by the IBM in 1957 for their IBM-704 computers. Efforts were taken to standardize this language from 1962 and the first standard was produced by the American National Standards Institute and is called as FORTRAN II. In 1966, another version was brought out. This is called as FORTRAN IV. This version remained in constant use until 1978, when a new standard was produced called FORTRAN 77 then later on FORTRAN 90. This latest standard showed marked difference from FORTRAN IV which is, however, still the most commonly used version. A FORTRAN source program is composed of statements that may cause data to be read, processing to be performed and results to be recorded. FORTRAN statements are composed of operation symbols (+, -, *, /) and expressions (of the type $A + B - C$). Statements may be divided into four groups. Input-output statements— which describe the operations that are necessary to read in data and to write the results of the program. Control statements— which may determine the sequence in which the statements are to be executed. Arithmetic statements— which specify the mathematical calculations to be performed. These statements very closely resemble a conventional arithmetic formulae; for example, $A = B + C$ is a valid statement. 78 Specification statements — which provide certain additional facts such as the size of the input data that is read by the program, or the placement of alphabetical and numerical information on printed page. Each statement in FORTRAN produces about five instructions in machine language. Although FORTRAN can handle a fairly large range of simple data types (real, integer, logical, complex, etc.), it cannot deal adequately with characters or characters strings as data types, and the only complex data structure available to the programmer is the array. Although it provides the ability to perform conditional statements, this ability is fairly-limited unless coupled with the GO TO (jump to another part of the

program) statement. FORTRAN is not suited for problems involving file maintenance, editing of data or production of documents. Despite these drawbacks, FORTRAN is still widely used. The program that was written in BASIC in section 4.4.1 can be rewritten in FORTRAN as follows: C M is used to store highest value and T is used to store the sum REAL M, N T = 0.0 M = 0.0 DO 30 I = 1,100 WRITE (*,40) READ (*,10) N IF (N.LT.M) GO TO 50 M = N 50 T = T + N 30 CONTINUE WRITE (*,20) M, T 10 FORMAT (F 10.4) 20 FORMAT (2 X, 'MAXIMUM NO. IS =',F10.4, 5X, 'SUM=', F12.4) 40 FORMAT (5 X, 'TYPE IN NUMBER') END 79 4.4.3. COBOL The name COBOL is derived from Common Business Oriented Language. In 1959, a group was set up to develop a common language that will be suitable for business applications and in December that year an interim language was proposed, which formed the basis for COBOL. It was standardized only in 1968 by the American National Standards Institute. A further standard was presented in 1974. COBOL is by far the most widely used programming language in the world today. The COBOL character set is composed of the 26 letters of the alphabet, the numbers 0 through 9, and 12 special characters. The COBOL language consists of names (can have up to 30 characters) to identify locations; constants and literals; operators that specify some action or relationship; key words essential to the meaning of a statement; expressions consisting of names, constants, operators or key words; statement containing a verb and an item to be acted on; and sentences composed of one or more statements properly punctuated. COBOL programs can be written in 'paragraph' form and the source program consists of statements in English. A program written in COBOL can be divided into four parts: 1. The identification division – which is used to attach a unique identification such as program name, program identification, date written, date compiled, author name etc. 2. The environment division – which is used to acquaint the processor with the computer on which the program is to be compiled and executed. 3. The data division - which is used to define the characteristics and format of the data to be processed. 80 4. The procedure division – which is used to describe the internal processing that is to take place. All input output operations, logical decisions, data movement, and computing operations must be performed in the procedure section. Data types are declared by picture clauses, which define the exact way in which data is structured. Statements are separated from each other with a comma or a blank space and particular parts of statement and paragraphs have to occupy certain positions on the coding line. The IF statement in COBOL closely parallels that used in FORTRAN. COBOL doesn't provide for absolute values, direct exponentiation of the base e, or trigonometric functions. The PERFORM statement in COBOL corresponds to FORTRAN DO statement and STOP RUN statement corresponds to the END statement of FORTRAN. COBOL is more difficult to learn than FORTRAN. However, the documentation provided by a COBOL source program is better than that provided by other languages. COBOL programs are organized in such a way that the data are physically separated from the computations (since they have separate data and procedure divisions) and this is generally accepted by present day computer scientists as being a natural representation of a programming problem. The language has the ability to handle conditional constructs and repetition constructs and is exceptionally good at dealing with records of different structures which may form files or data bases. The COBOL language is highly suitable where we have less computations and huge inputs and outputs in a given problem. However, the main drawback of the COBOL language is that it cannot be used for scientific applications. Further, COBOL processors require more CPU time to compile the programme. 81 The solution to the example problem (which was taken to illustrate and features of BASIC and FORTRAN languages) may be written using the COBOL languages as follows: IDENTIFICATION DIVISION. PROGRAM-ID. ADDNOS. AUTHOR. USER-NAME. ENVIRONMENT DIVISION. CONFIGURATION SECTION. SOURCE-COMPUTER. IBM-PC. OBJECT-COMPUTER. IBM-PC. DATA DIVISION. FILE SECTION. WORKING-STORAGE SECTION. 77 N PIC 9(10)V9(4). 77 T PIC 9(12).9(4) VALUE ZERO. 77 M PIC 9(10).9(4) VALUE ZERO. PROCEDURE DIVISION. START-PARA. PERFORM SUMMING 100 TIMES. DISPLAY "THE MAXIMUM NUMBER IS", M. DISPLAY "THE SUM OF THE NUMBERS IS", T. STOP RUN. SUMMING. DISPLAY "TYPE IN NUMBER". ACCEPT N. IF N < M MOVE N TO M. ADD N TO T. 82 4.4.4 PASCAL The language PASCAL is named after the French mathematician Blaise PASCAL who invented the first mechanical adding machine. Niklaus Wirth, a computer scientist from Switzerland developed PASCAL during 1968-70. The objective was to formulate an ALGOL-like language which could be used as a tool to teach people the methodology of structured programming. which is an efficient method of developing software? Since then PASCAL has been adopted as a primary teaching language for programming courses in many universities. Now-a-days, PASCAL is available with most of the microcomputer installations and used not only in educational applications but also in business and scientific applications. PASCAL is rich both in data structures and control structures. In addition to data types real, integer, character, boolean, the programmer can also device his own data type and define the values they can take (for example, he/she can choose the type, colour, which can have values such as red, green, black, etc.). The data structures include the array and the record structure, which allows the structuring of elements of differing types. The control structures include the full flexibility of the conditional statement, and three different forms of repeat statement. It provides for sub-program facilities which are similar to those of ALGOL. Pointer variables which are of considerable importance while using dynamic data structures (such as linked lists and trees) can also be implemented. PASCAL is well suited for both batch and interactive modes of processing. The major drawback of PASCAL is that it is not yet

standardized. Many versions are available which will make its transportability difficult. Compared to other languages like FORTRAN, PASCAL has poor input/output capabilities. Following is a version of the program written in PASCAL for the same example problem considered previously. 83 PROGRAM TOTAL (INPUT, OUTPUT); VAR I, N, M, T: INTEGER ; BEGIN T: = 0; M: = 0; FOR I : = 1 TO 100 DO BEGIN WRITELN ('TYPE IN NUMBER'); READLN (N); IF N < M THEN M : = N; T : = T + N; END; Writeln ('THE MAXIMUM NUMBER IS', M); Writeln ('THE SUM OF THE NUMBERS IS', T); END.

4.4.5. C Language C is a structured language, as is PASCAL, BASIC, COBOL and FORTRAN are non-structured languages. C language uses blocks. A block is a set of statements that are logically connected. A structured language gives you a variety of programming possibilities such as it supports the concept of functions. Because of clarity, C is not only easier to program in but also much easier to maintain. C was first used for system programming. System programs make the computer capable of performing useful work. C does have its benefits. It has 28 keywords to remember. C code is very portable. A 'C' program is a collection of one or more functions. To write a program, you first create functions and then put them together. 84 Following is a program written in C for the same example considered previously: # include >stdio-h< main () { int i; float t = 0, m = 0, n; for (

i = 1; i >= 100; i ++) { printf ("Type in number:"); scanf("%f",&n); if (

99%

MATCHING BLOCK 9/11

W

n < m) { m = n; } t + = n; } printf ("\n The Maximum Number is %f", m); printf ("\n The sum of Number is % f", t); } 4.5. Assemblers, compilers and interpreters As discussed already, any program that is not written in machine language has to be translated before it is executed by the computer. We shall see the details of three types of translator programs : assembler, compiler and interpreter now. 4.5.1 Assembler Assembler is a program which is used to translate programs written in assembly language into machine code. One machine instruction is 85 generated for each source program instruction. The resulting program can be executed only when the assembly process is completed. The functions of an assembler can be summarized as: 1. To translate mnemonic operation codes into machine code, and symbolic addresses into machine addresses. 2. To assign absolute address to any symbolic address or label names. 3. To check each instruction for its correctness and to generate diagnostic messages. 4. To place each instruction in central memory. 5. To provide a cross-reference table between all symbolic names used and their absolute addresses. 6. To provide the necessary linkages for closed subroutines. 7. To produce the object program on tape, or disk when required. 8. To produce printed listing of the source and object program with comments. 9. To inform the control unit (after all the errors have been corrected to execute the program starting from the first instruction of the object code. The main stages of assembling are: 1. Input source program (in assembly language) through an input device. 2. Use the assembler to produce an object program in machine language. If errors are found by the assembler, the source program must be modified and corrected. This process of modification and 86 correction should be repeated until all errors found have been corrected. 3. Load object program into main memory by using a loader. 4. Execute the object program (by using the operating system) in order to get the desired results. The above process is illustrated in Fig. 4.1.

100%

MATCHING BLOCK 3/11

W

Fig. 4.1. Schematic diagram of assembly process 4.5.2. Compiler A compiler translates a program written in a high level language to executable machine instructions. The compiler, like the assembler, is a program that resides on a disk or other mass-storage media. When the compiler is needed, it is called by the computer and loaded into RAM. It has to be remembered that the process for translating high level language source program into object code will be more lengthy and complex. Hence, compilers tend to be longer and more complex programs than assemblers and hence occupy more main memory space. Compilers can serve more purposes than mere translation. Many modern compilers have diagnostic capabilities that make them excellent quality assurance tools (quality assurance tools are some aids- in the form of

100%

MATCHING BLOCK 4/11

W

Source Program Assembler Program Object Program Printed Listing, Error Messages, Results, etc. 87

99%

MATCHING BLOCK 5/11

W

programs- which will assure the quality of software that is produced). They are useful at all stages after pre-coding. The job of a compiler include. 1. To translate the source program statements into machine code (one high level language statement is normally translated into four or five machine language statements), 2. To check for conformity to programming standards. 3. To perform some limited, but useful semantic analysis (e.g. indicating mixed mode arithmetic). 4. To trace variables. 5. To include linkage for closed subroutines. 6. To allocate areas of main storage. 7. To provide extensive maps that are useful in diagnosing information flow. 8. To generate object program on tape or disk- whenever required. 9. To produce printed listing of the source and object programs when required. Sometimes compilers are also used to indicate less efficient programming practices such as two branch IF statements. Although it may look that the compiler does the same job as an assembler, it does fat more, particularly in the translation of source statements and the linkage of subroutines. The process of compilation is shown in Fig. 4.2, which consists of the following steps: 1. The user enters the source program through the terminal i.e. key- board and stores it in the RAM. 2. The compiler, which is available in RAM is called and asked to perform the translation. The translation process is repeated again 88 and again till all the errors in the source program are identified and corrected. 3. The compiler transforms the source program into an equivalent object program (in machine language). 4. The object program thus generated is executed in order to get the desired results. It has to be noted that compilers are not needed to run the program and are needed only to translate it. Also, one compiler is capable of translating source programs written in only one high level language. Thus, a COBOL compiler cannot be used to translate a FORTRAN source program.

Fig. 4.2. Procedure for using a compiler

93%

MATCHING BLOCK 7/11

W

START LOAD COMPILER INTO MEMORY WRITE, FED AND THEN LOAD SOURCE PROGRAM RUN THE COMPILER IN ORDER TO GET OBJECT PROGRAM LOAD OBJECT PROGRAM EXECUTE OBJECT PROGRAM TO GET RESULTS STOP

ARE THERE ERRORS? YES NO
89 4.5.3.

100%

MATCHING BLOCK 8/11

W

Interpreter High level languages may also be translated using interpreters. Using a compiler, the whole program is translated completely and then the resulting machine language version is executed. Whereas using an interpreter, each instruction is translated and executed. Whereas using an interpreter, each instruction is translated and executed in turn- which means repetitive translation of instructions within loops. In other words, the interpreter takes one source program instruction, translates it into object code and executes it, then takes the next instruction, translates it, and so on. Use of an interpreter can save core space since the interpreter program itself is quite small in size. It also eliminates the need to store the program's translated object code in the computer. Interpreters are convenient and easy to use because they are totally interactive. The user can sit in front of the terminal and converse with the computer in order to solve his problems directly through the interpreter. Whereas if he is using a compiler, he has to generate the object code and if he wants to change something in the program (to add some new features or to correct errors), the entire process must be repeated. That is, the program is corrected, re-compiled, and executed. Whereas using an interpreter, changes and additions can be made interactively.

Fig. 4.3. Procedure for using an interpreter

100%

MATCHING BLOCK 10/11

W

START Load Interpreter into memory Enter and load source program into memory Translate and execute the program using interpreter

Stop Are there errors in the program No Yes
90

When he needs to modify or add something, he can do it immediately through the interpreter. When using an interpreter, a programmer gets a feeling that he is directly speaking to a computer and commanding its attention. However it has some disadvantages, program statements that are used multiple times must be translated each time they are executed. Another big drawback of an interpreter is its low speed. Interpreters execute programs at a speed 10-20 times lower than the equivalent machine code generated by an assembler or compiler. In some applications, this low speed is undesirable and also unacceptable. The process of executing programs using an interpreter is shown in Fig. 4.3. It has to be noted that on some micros and personal computers, the interpreter for the BASIC language is resident in ROM and hence there is no need to load them on to the memory.

4.6. Self-test questions

1. Distinguish between low-level language and high-level language.
2. Define the following terms: (a) Computer program (b) Object program (c) Source program
3. Compare and contrast between machine language and symbolic language programs.
4. Name the four popular high-level languages used in India.
5. Why is COBOL mainly used for business applications? What is its main drawback?
6. Explain the features of the following languages briefly: (a) BASIC (b) FORTRAN (c) COBOL

91

UNIT 5 -

BINARY AND OTHER NUMBER SYSTEMS Objective:

After going through this chapter, you will be able to understand: the concept of binary number system; binary arithmetic such as addition, subtraction, multiplication and division; Concept of complements and octal and hexadecimal number systems.

Structure

- 5.1. Binary Number System
- 5.2. Conversion from Decimal to Binary
- 5.3. Binary Arithmetic
- 5.4. Binary Fractions
- 5.5. Octal and Hexadecimal Number Systems
- 5.6. Self-test questions

92 5.1. Binary Number System Data is represented within the computer as a sequence of bits (which is called binary code) and is processed using the laws of binary number arithmetic. Computers can understand only the binary number system. The binary system has only two digits 0 and 1. Computers use binary system because the electrical devices can understand only 'on' (1) or 'off' (0) states. Also electrical and magnetic storage devices having two stable states that are simple to construct. Though these elementary devices could be combined to handle numbers with large bases, they are less efficient than two state devices and are too complicated and costly. Binary numbers can be constructed just like decimal numbers except that the base is 2 instead of 10. Thus, 646 (decimal) = $6 \times 10^2 + 4 \times 10^1 + 6 \times 10^0$ Similarly, 101 (binary) = $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$ (decimal) In the decimal system, the first place is for 1s, 2nd place is for 10s, and the 3rd place is for 100. On the other hand, in the binary number system, the factor is 2 instead of 10. The first place is still for 1s, but the second is for 2s, the 3rd for 4s, the 4th for 8s and so on. Examples

1. 1010101 (binary) = $1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 64 + 0 + 16 + 0 + 4 + 0 + 1 = 85$ (decimal)

2. 101011 (binary) = $1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 32 + 0 + 8 + 0 + 2 + 1 = 43$ (decimal)

3. 11111 (binary) = $1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 8 + 4 + 2 + 1 = 31$ (decimal)

4. 10001011 (binary) = $1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 128 + 0 + 0 + 0 + 8 + 0 + 2 + 1 = 139$ (decimal)

It is seen from the examples above that while converting binary number into decimal, the following rules have to be adopted:

1. Multiply the right most bit by 1.
2. Going right to left, multiply each succeeding bit by twice the factor used with the previous bit.
3. Add the products according to the usual rules of decimal addition.

5.2. Conversion from Decimal to Binary

Converting decimal number to binary is slightly difficult than converting binary to decimal. Normally a division procedure is used which has the following rules:

1. Divide the decimal number by 2, save remainder.
2. If the quotient is 0, proceed to step 3, if the quotient is not 0, replace the number with quotient and repeat step 1.
3. The binary representation of the decimal number is the remainders starting with the first remainder at the right.

94 Examples:

1. Find the binary equivalent of 32. Number to be divided Quotient Remainder

32 16 0 Most significant Bit

16 8 0 8 4 0 4 2 0 2 1 0 1 0 1 Least significant Bit

Thus, 32 (decimal) = (100000)₂ It can also be written as $(32)_{10} = (100000)_2$

Check: $100000_2 = 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 32 + 0 + 0 + 0 + 0 + 0 = 32$ (decimal)

2. The process of converting 47 into binary equivalent is shown in the following table: Number to be divided Quotient Remainder

47 23 1 Most significant Bit

23 11 1 11 5 1 5 2 1 2 1 0 1 0 1 Least significant Bit

Thus, $(47)_{10} = (101111)_2$

95 3. The equivalent of 77 in binary can be found out as follows : Number to be divided Quotient Remainder 77 38 1 Most significant Bit 38 19 0 19 9 1 9 4 1 4 2 0 2 1 0 1 0 1 Least significant Bit Thus, $(77)_{10} = (1001101)_2$

24. The number 27 in decimal can be converted into binary as follows : Number to be divided Quotient Remainder 27 13 1 Most significant Bit 13 6 1 6 3 0 3 1 1 0 1 Least significant Bit Thus $(27)_{10} = (11011)_2$

5.3. Binary Arithmetic 5.3.1. Binary Addition Binary addition is exactly like decimal addition except that the rules are much simpler. The binary addition rules are shown in Table 5.1.

96 Table 5.1. Binary Addition Table A B A+B 0 0 0 0 1 1 1 0 1 1 1 0 with carry 1 Note that, in the table above a carry is generated only when two ones are added. Examples 1. Add 101010 with 111111 101010 (42) + 111111 (63) 1101001 (105) Note that there are carries from the 2nd, 3rd, 4th, 5th and 6th bit positions (counting from the right and starting with the 0th position). 2. Add 1010 with 110 1 1 1 Carry 1 0 1 0 op 1 (10) + 1 1 0 op 2 (6) 1 0 0 0 0 (16) Here, there are carries from 2nd, 3rd and 4th bit position. Since binary subtraction can produce negative results, a method has to be evolved to represent negative numbers. Normally, the most significant

97 bit is used for the sign; the bit is zero if the number is positive and one if the number is negative. Of course, the numerical position of the most significant bit depends on the word length of the computer. Hence, in the following discussions, we shall assume the words are 8 bit long. Reserving one bit for the sign reduces the magnitude of the largest number that can be represented by a word, since one less bit is available. The representation of the numbers 22 and -22 in an 8 bit per word computer is shown below- 00010110 10010110 (Binary) +22 -22 (Decimal) Most of the computers have fixed number of words for storing integers. The maximum size of integer number that can be stored in a given number of bits can be found out easily by remembering that, one of these is a significant bit. This is illustrated in Table 5.2 which lists the maximum possible number both in binary and decimal for different word lengths. Thus, it is seen that a maximum permissible integer in a computer with an n bits/word and one word per integer is equal to $2^{n-1} - 1$. Table 5.2 Word length (bits) Maximum binary number Equivalent decimal No. Expressed in 2^n form 2 01 1 2 1 -1 3 011 3 2 2 -1 4 0111 7 2 3 -1 8 01111111 127 2 7 -1 16 0111111111111111 32767 2 16 -1

98 5.3.2. Binary Subtraction In the decimal system, we subtract decimal digit from smaller decimal digit by borrowing 1 from the next column. The same procedure can be adopted for binary subtraction also. The rules for binary subtraction are shown in Table 5.3. Table 5.3 A B A - B 0 0 0 1 0 1 1 1 0 0 1 1 with borrow 1 Note that, in the last row of the Table 5.3, the subtraction has been done with a borrow of 1 from the next column. These rules are illustrated in the following examples: Examples: 1. Subtract 1101 from 10101111 1 0 1 0 1 1 1 1 - 1 1 0 1 1 0 1 0 0 0 1 0 Note that, in the example above, there is no borrowing. 2. Subtract 1101 from 111001 10 10 10 0 1 - 1 1 0 1 1 0 1 1 0 0

99 In the example above, we borrowed one from the 4th column because of the difference 0-1 in the 3rd column. 3. Subtract 1111 from 10001 1 1 0 10 10 10 1 - 1 1 1 1 0 0 1 0 We observe that, in the example above, to subtract one from zero in the 2nd column, we borrowed from the 5th column. This is exactly similar to the following situation which occurs in decimal number system. 6 0 0 4 3 - 2 5 4 3 1 2 3 4 5 7 3 1 5.3.3 Complements The method described till now for binary subtraction is not well suited for the computer. Normally, in a computer, subtraction is transformed in the form of addition by use of one's complement scheme or two's complements scheme (It is also called as Radix-one complement or Radix complement scheme). In the well-known decimal system, the radix complement of a decimal number may be obtained by subtracting each digit of the number from nine. For example, the 9's complement of 512 is 487. The 10's complement of an integer is defined to be equal to its 9's complement plus one. Thus, the 10's complement of 512 is 488. It has to be noted that the sum of a number and its 10's complement is always equal to 10^n Thus, $512+488 = 1000=10^3$, where n is number of digits of a number.

100 Decimal subtraction can be accomplished by the addition of the 9's complement plus 1, or the addition of 10's complement as shown below: 632 632 632 - 246 + 753 + 754 386 1385 1386 + 1 (add the carry of 1) Carry dropped 386 386 a) Ordinary decimal subtraction b) Decimal Subtraction by use of 9's complement c) Decimal Subtraction

by use of 10's complement The left most "one" is simply deleted in adding the 10's complement but is subjected to "end around carry": in adding the 9's complement. Thus, when subtracting by the complementary method, the following three steps are involved. Step 1:

Find the 9's complement of the number to be subtracted, you are subtracting; Step-2: Add this to the number from which it has to be subtracted; Step-3: If there is a carry of 1, add it to obtain the result. If there is no carry, add 0, re-complement and attach a negative sign to obtain the result. The above procedure is illustrated again with the following example:

101 Subtract 57 from 24. Using Normal Subtraction Complementary method 24 24 -57 Step 1 + 42 (9's complement of 57) -33

Step 2 066 Step 3 0 (add 0 as no carry) 66 Result - 33 re-complement (9's) and attach a negative sign. Similarly, the 1's complement of a binary number is obtained by subtracting each

binary digit from one, and the 2's complement is the 1's complement plus one. For example Binary number : 11010 111000 100101 1's complement : 00101 000111 011010 2's complement : 00110 001000 011011 Thus, a binary subtraction can be done exactly in the manner similar to as done using 9's or 10's complement in the case of a decimal number. The following illustration illustrates the use of 1's and 2's complement scheme in binary subtraction. 101011 101011 101011 - 010111 + 101000 + 101001 10100 1010011 1010100 + ----- 1 (add the carry of 1) Carry is deleted 10100 10100

102 a) Ordinary binary subtraction b) Binary Subtraction by using 1's complement

c) Binary subtraction by using 2's complement Let us take another example to illustrate the above 1's complement subtraction. Example Subtract 1001000 from 0001110 0001110 + 0110111 (1's complement of 1001000) 01000101 0 (add 0 as no carry) 1000101 Result - 0111010 (re-complement (1's) and attach a negative sign) Thus, we see that the process of subtraction is reduced to that of an addition. The computer performs the multiplication and division also as a series of additions and subtractions respectively. Thus, all the operations that are done by most computers are based on binary addition only. 5.3.4. Carry and Overflow Most computers have a one bit flag or condition code that saves the carry from the most significant bit of an addition or subtraction. This flag (or carry) can be used to implement carries or borrows between words in multiple precision arithmetic.

103 Example : Add the following two numbers 1110101110101111 + 0001001111110010 Step 1 : Add the least significant 8 bits of the numbers 10101111 +11110010 10100001 and a carry Step 2 : Add the most significant of the 8 bits of the two numbers plus a carry from step 1. 11101011 +00010011 + 1 11111111 Thus, the answer is 111111110100001 It is clear from the above that the addition of 16 bit numbers on an 8 bit computer requires more than simply two additions, since the second addition must include the carry. Many computers also have a twos complement overflow flag. This flag is set when the result overflows into the bit position used for the sign. 5.3.3. Binary Multiplication Binary multiplication can be accomplished using the same method that is used for the long multiplication of decimal numbers, with the

104 advantage that the binary multiplication table is very much simpler, as shown in Table 4.4. Table 4.4 A B A x B 0 0 0 1 0 0 1 0 0 1 1 1 This multiplication table is illustrated using the following example in which we shall find the product of 1011 x 110. 1011 multiplicand x 110 0000 1011 partial produce 1011 1000010 sum of partial products Thus, we see that the multiplication process is essentially a series of repeated additions of the multiplications appropriately shifted. Use of add shift principle is in fact made in the Arithmetic/Logic unit of the computer for binary multiplication. It has to be noted that the product is much longer than the individual number multiplied. In general, the product of two 'n' bit numbers will require up to '2n' bits to accommodate the result. Thus, the rules for binary multiplication are: 1. Copy multiplicand when multiplier is 1; do not when it is 0. 2. Shift as in decimal multiplication. 3. Add the resulting binary numbers according to the binary addition rules.

105 5.3.4. Binary Division Binary division can be accomplished using similar methods adopted for the long division of decimal numbers. The rules of binary division are shown in Table 5.5. Table 5.5 A B A/B 0 1 0 1 1 1 1 These rules are illustrated in the following example. 111 ÷ 10 (Dividend) / (diviser) = quotient 10) 111 (11.1 10 11 10 10 10 00 It has to be noted that division is analogous to binary multiplication. In this, although the diviser is repeatedly shifted it is subtracted rather than added. Thus, the rules are as follows : 1.

Start from the left of the dividend. 2. Perform a series of subtractions in which the diviser is subtracted from the dividend. 106 3. If subtraction is possible, put a 1 in the quotient and subtract the diviser from the corresponding digits of the dividend. 4.

If subtraction is not possible (diviser greater than remainder), record

a 0 in the quotient. 5. Bring down the next digit to add to the remainder digits. Proceed as before in

a manner similar to long division. 5.4. Binary Fractions Just as we use a decimal point to separate the whole and decimal fraction parts of a decimal number, we can use a binary point in binary numbers to separate the whole and fractional parts. The significance of the weighting of the bits behind the binary point is illustrated below and also summarized in Table 5.6. 0.111 (binary) = 2⁻¹ + 2⁻² + 2⁻³ = 0.5 + 0.25 + 0.125 = 0.875 (decimal) Table 5.6. Decimal and Corresponding Binary Fractions Proper fraction Power of 2 Decimal fraction Binary fraction 1 2 2⁻¹ 0.5 0.1 1/4 2⁻² 0.25 0.01 1/8 2⁻³ 0.125 0.001 1/16 2⁻⁴ 0.0625 0.0001 1/32 2⁻⁵ 0.03125 0.00001 1/64 2⁻⁶ 0.015625 0.000001 1/128 2⁻⁷ 0.0078125 0.0000001 1/256 2⁻⁸ 0.00390625 0.00000001 1/512 2⁻⁹ 0.001953125 0.000000001

107 To convert the decimal fraction to binary fraction, we employ the following rules: 1. Multiply the decimal fraction repeatedly by the base 2. The whole number part of the first multiplication gives the first 1 or 0 of the binary fraction. 2. The fractional part of the result is carried over and multiplied by 2. 3. The whole number part of the result gives the second 1 or 0, and so on. Examples : 1. Convert $(0.375)_{10}$ to binary fraction $0.375 \times 2 = 0.750 \notin 0$ $0.750 \times 2 = 1.500 \notin 1$ $0.500 \times 2 = 1.000 \notin 1$ Hence $(0.375)_{10} = (0.011)_2$ Check: $0 \times 1/2 + 1 \times 1/4 + 1/8 = 0 + 0.25 + 0.125 = 0.375$ 2. Convert 0.625_{10} to binary fraction $0.625 \times 2 = 1.250 \notin 1$ $0.250 \times 2 = 0.500 \notin 0$ $0.500 \times 2 = 1.000 \notin 1$ So, $0.625_{10} = 0.101_2$ Check: $0.101 = 1 \times 1/2 + 1/2 + 1/8 = 0.50 + 0 + 0.125 = 0.625_{10}$

108 5.5. Octal and Hexadecimal Number System Since binary numbers are long and cumbersome, more convenient representations combine groups of three or four bits into octal (base 8) or hexadecimal (base 16) digits respectively. Both representations are much shorter and more convenient and are closely related to the binary system.

5.5.1. Octal number system In octal number, there are only eight possible digits (0 to 7). The position of a digit to the left of the octal point carries a weightage in ascending powers of eight. Digits behind the octal points are also used (octal points are similar to decimal and binary points and are used to separate whole and fractional parts) and carry weightage of 8^{-1} . The octal and binary conversion is given in Table 5.7.

Table 5.7 Octal Binary

0	000	1	001	2	010	3	011	4	100	5	101	6	110	7	111
---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----

Example: $(1.68)_8 = 1 \times 8 + 6 \times (1/8) = 1.75 = 1 \times 8^0 + 6 \times 8^{-1} + 8 \times (1/64) = 1.875 = 1 \times 8^0 + 6 \times 8^{-1} + 8 \times 8^{-2} = 1.875$

109 To convert from binary to octal, we group the binary number from the least significant bit (LSB) in groups of 3 bits; each of these 3 groups is then expressed by its equivalent octal digit. For example, 111 110 011 is expressed as 111 110 011 $7 6 3 = (763)_8$ With whole and fractional parts we group in threes, starting from the octal point, moving left for the whole part, and right for the fractional part. Examples: 1. Convert 10101.111101 into its equivalent octal number

10101.111101 is divided into 3 bit segments as 010 101 . 111 101 2 5 . 7 5 i.e. $(25.75)_8$ Note that a leading zero is assumed in the first segment. Thus the octal equivalent of 10101.111101 is $(25.75)_8$ 2. Convert 1110110101.10 into octal. 001 110 110 101 . 100 = $(1665.4)_8$ 1 6 6 5 4 Note that zeros have been added in the beginning and at the end segment.

110 3. The reverse process- octal to binary- consists simply of writing the 3 bit binary number for each octal digit. Thus, $(5673)_8$ can be converted as $(5673)_8 = (101 110 111 011)_2$ 3. Convert 473.5_8 into binary $473.5_8 = 100 111 011.101_2$

5.5.2. Hexadecimal number system Hexadecimal numbers have a base of 16 and employ 16 distinct digits to represent the decimal numbers (0 to 15). These hexadecimal digits are 0- 9, A, B, C, D, E, F. The least significant digit in front of the hexadecimal point is weighted 16^0 , the next is 16^1 and so on. Similarly, successive hexadecimal digits behind the points are weighted 16^{-1} , 16^{-2} and so on. The letters A to F are used for the hexadecimal digits with values 10 through 15. Thus, their usage may be confusing because numbers in this way will resemble names or words. Converting a binary number to its hexadecimal representation requires coding of each 4 bit segments, starting with the four least significant bits. Table 5.8 shows the relationship between decimal, hexadecimal and binary numbers. Table 5.8.

Decimal, hexadecimal and binary numbers

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

The binary number 111101011 can be divided into the following three four bit segments: 0001, 1110, 1011. This can then be converted into hexadecimal digits using Table 5.8 as 1EB H. Hexadecimal representation is more difficult to understand than the octal representation but is shorter. Converting hexadecimal number to binary requires using Table 5.8 in reverse. Example: Convert EA9 (hexadecimal) into binary. Referring to Table 5.8, E = 1110 A = 1010 9 = 1001 Thus, EA9 (hexadecimal) = 111010101001 (binary) or EA9 H = 111010101001 2

112 Hexadecimal numbers can be added, subtracted, multiplied and divided just like decimal numbers except that the rules are slightly different, since the base is 16 instead of 10. 5.5.3. Comparison of octal and hexadecimal The hexadecimal representation has the following advantages over the octal representation. 1. It is significantly shorter, particularly for long computer word lengths, such as 48 or 64 bits. 2. The hexadecimal representation is more convenient when the word length is divisible by 4 but not by 3. A four, eight, sixteen or thirty two bit word divides evenly into hexadecimal digits but not into octal digits. The octal representation thus requires the placing of zeros ahead of the most significant bit to complete the leading digit and thus wasting memory. The octal representation has these advantages over the hexadecimal representation. 1. It is easier to derive from a binary number. 2. Octal numbers cannot be confused with names or words. 3. Converting octal numbers to decimal is simpler, since the digits need not be changed (all octal digits are also decimal digits, whereas hexadecimal has some letters as well). 4. Octal arithmetic is simpler and less confusing than hexadecimal arithmetic. It is seen that using either octal or hexadecimal notation as compared the binary system to describe the underlying binary digits is less error prone and more convenient. This is mainly due to the fact that these

113 representations are much shorter than the binary notation and it is very simple to convert from binary to octal or to hexadecimal. When a binary computer uses one of these systems for operator and programmer purposes, the computer messages regarding the instructions and data stored in memory are in that notation rather than straight binary. Many of the computer console displays on the operator panel may show a light for each binary digit, but the lights are marked off in groups of 3 or 4 bits so that the operator can read them in octal or hexadecimal notation. The contents of the bytes of the computers using 8 bit bytes (e.g. IBM 370) are printed out as two hexadecimal digits.

5.6. Self-test questions

1. Convert the following decimal numbers to equivalent binary numbers: (a) 32 (b) 57 (c) 2035 (d) 5.75 (e) 3.25

2. Convert the following binary numbers to equivalent decimal numbers: (a) 1001, (b) 11101, (c) 10101, (d) 0.1101, (e) 0.011011, (f) 0.0010010

3. Perform the following additions (you can check it independently by converting the binary numbers to decimal) (a) 1100011 + 100111, (b) 1100011 + 111011, (c) 11011 + 10.011, (d) 1010.01 + 1101.11

4. Perform the following subtractions in binary. (a) 1101 – 1001 (b) 1110 – 1001, (c) 1111.110 – 1010.001 (d) 1101.11 – 101.01

5. Perform the following binary multiplications: (a) 111010 × 1010 (b) 10111 × 1101 (c) 1111110 × 1010.001

6. Convert the following Octal numbers to binary numbers: (a) 64, (b) 45, (c) 31.234, (d) 453.45

114

UNIT 6 -

PC SOFTWARE PACKAGES Objectives: The motive of this chapter is to present an introduction of PC software packages and within that framework the key concepts of software and various types of software packages are discussed.

After going through this lesson, you will be able to: $\frac{3}{4}$ Describe the concept of

software. $\frac{3}{4}$ Explain the types of software packages. $\frac{3}{4}$ Differentiate between compiler and interpreter. $\frac{3}{4}$ Describe the various forms of system and application software packages. $\frac{3}{4}$ Describe some of the advantages and disadvantages of the different types of software packages.

Structure: 6.1 Introduction 6.2 Types of Software Packages 6.3 Summary 6.4 Self-Test questions 6.5 Suggested Readings

6.1 Introduction Software is a set of programs, documents, procedures and routines associated with the operation of a computer system

for specific application/task. It is the essential complement to computer hardware. Software usually refers to a set of advanced computer modules that allow the user to plan efficient surveys, organize and acquire satellite navigation data, verify and download data, process and analyze measurements, perform

115 network adjustments, and report and archive the final results. These are the written coded commands that tell a computer what tasks to perform. The term software has acquired several meanings in computer literature but essentially, it describes the 'Soft' portion of the computer installation, that is the computer programs, which when executed operate the hardware. For example, Word, PhotoShop, Picture Easy, and Photo Deluxe are software programs. Software is a sequence of instructions that can be executed by a computer (

CPU) that performs a certain task intentioned by the programmer. Within a software actions are defined that should be executed by a computer. These actions are put in a predefined order: a logical path. Software may also be referred to as the specific set of instructions, programs, procedures, rules, and any associated documentation pertaining to the operation of a system that tells the computer hardware precisely what to do. The entire set of programs, procedures, and related documentation associated with a computer system controls the operation of a computer and are distinguished from the actual hardware of the computer. Hence, a series of instructions for the computer that performs a particular task is called software. Now we can define software as: Software is a program, procedure or set of logical instructions which when fed into the memory of computer system, enables the computer hardware to perform the desired task of the user. What is a Software package? A software package is a special method for the distribution and installation of software on computer systems. It is the written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory. In simple words, software package is a program with its associated files, archived and compressed into a single file along with any necessary scripts or information to aid in managing the installation, upgrade, and removal of those files.

116 Software packages have evolved from stand alone software programs and notes based information systems to more integrated web enabled packages which have even the potential of working across the 'value network' of internal teams and management, customers, suppliers and sometimes competitors. 6.2 Types of Software Packages Software Packages are often divided into two major categories: (1) System software Package (2) Application software. Package (1) System software Package: A System Software Package is a collection of programs designed to operate, control and extend the processing capabilities of the computer itself. This software operates the computer itself and is made up of control programs. System software packages are generally prepared today by computer manufacturers and by independent software suppliers. Some important system software packages found in almost all computer installations are the operating system and all utility programs that manage computer resources at a low level. Others are compilers, interpreters and assemblers. System software package is required to support the production or execution of application software but which is not specific to any particular application. Examples of system software package would include the operating system, compilers, loaders, linkers, debuggers, editors, sorting programs and all the other utilities that enable the computer to function. A system software package may be further classified as follows: • Operating System • Language Translators • Utilities

117 Operating system

Operating system is the most important program that runs on a computer system. It is an organized collection of software that controls the overall operations of a computer and provides an environment for execution of programs.

It acts as an interface between user and the hardware of a computer system.

Every general-purpose computer must have this important system software package to run other programs. The operating system permits the computer to supervise and manage its resources. It performs the basic tasks, such as recognizing input from the keyboard, sending output to the

display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers.

Operating system

calls the application programs, translates the special service programs and manages data needed to produce the output as desired by users. User is not bothered about hardware functioning which is silently taken care by the operating system. For large systems, the operating system has even greater responsibilities and powers. It is like a traffic cop and makes sure

that different programs and users

running at the same time do not interfere with each other.

The operating system

is also responsible for security, ensuring that unauthorized users do not access the system.

Operating systems can be classified as follows: Single User O.S.: Allows only one user at a time. Most popular single user O.S. is MS- DOS. Batch O.S.: Allows processing in the form of groups of similar jobs consisting of programs, data, and appropriate system commands. Multi-programming O.S.: Allows the instruction and data form two or more separate processes to reside in primary memory simultaneously. Multi-user O.S.: Allows two or more users to run programs at the same time. Some operating systems permit hundreds or even thousands of concurrent users. Multiprocessing O.S.: Supports running a program on more than one CPU. Multitasking O.S.: Allows more than one program to run concurrently.

118 Multithreading O.S.: Allows different parts of a single program to run concurrently. Real time O.S.: Responds to input instantly. General-purpose operating systems, such as DOS and UNIX, are not real-time. Network O.S.: Supports the existence of multiple computers and users can log into remote machines and copy files from one machine to another machine. Distributed O.S.: acts as a virtual uniprocessor but not as a collection of distinct machines. Users are not aware of where there programs are being run or where there files are residing. Operating systems provide a software platform on top of which other programs, called application

programs, can run. The application programs must be written to run on top of a particular operating system. For PCs, the most popular operating systems are DOS, OS/2, and Windows, but others are available, such as Linux. The user normally interacts with the operating system through a set of commands. For example, the DOS operating system contains commands such as COPY and RENAME for copying files and changing the names of files, respectively. The commands are accepted and executed by a part of the operating system called the command processor or command line interpreter. Graphical user interfaces allow to enter commands by pointing and clicking at objects that appear on the screen. Language Translators Language translators are those program tools which translate the programs written in other computer languages to machine codes. These are the methods to "translate" source code into computer understandable language. Because programs written in a high level language are not directly understood by computers, they need a software tool to convert these programs into machine level language. The software packages available for this purpose are like COBOL compiler, C compiler etc. These are mainly of three types: • Interpreter • Compiler • Assembler

119 Interpreter: Interpreter is the oldest type of computer program that executes instructions that are written in the form of a program. The trick of an interpreter is that it loads the source code and translates the instruction into executable machine language line by line. And it does it over and over again any time that program is run. In order to run a program first (e.g. "load BASIC") the interpreter has to loaded into the memory of the computer, then the source code and then type "run to execute the program. An interpreter is unique for each high level language. It is relatively slow. Another disadvantage is that mostly these interpreters are expensive. Also to the designer of the program this can be a disadvantage. He or she will just give away its trade secrets and have no longer control over his own program, the distribution or avoid unwanted adaptations. Compiler: Compiler is a computer program that translates source code written in high level language into machine language. First the source code is read into the computer's memory and then it is translated in to a kind of object code.

A compiler creates a unique object program i.e. once a source program is compiled there is no need of that source program because output can be obtained by executing that object program.

For each high level language the machine requires a separate compiler which is personal for that language. Compilers are somewhat more compact and run, in some extend, faster. Basically the functions of compilers and interpreters are same, the only difference is that a compiler provides a separate object program, but the interpreter does not. In case of interpreter, every time the source code is required for the execution of program whereas compiler uses the object code developed by compilation of the source code. Another difference is that interpreter translates the source program line wise i.e. it will not execute the next line unless the first one has no

error but compiler translates the whole program at a single time and produce an object code for the execution purpose. Assembler: An assembler is a program that translates an assembly language program (i.e. symbolic instructions) to its machine language equivalent. A programmer did no longer have to work with one's and zero's when using an assembly language. He or she

120 can use symbols (e.g. STO = store) instead. These symbols are called mnemonics and each mnemonic stands for one single machine instruction. But an assembler still works on a very low level with the machine. So the assembler has to take these non machine language instructions (mnemonic name instructions) prepared by a programmer and convert them into a form that may be used by the computer. Examples of assemblers for Intel 8086/8088 Assembly are MASM (Macro Assembler) 5.0/6.0, TASM (Turbo Assembler), etc. Utilities Utility software (also known as utilities) performs a very specific task, usually related to managing system resources. Operating systems contain a number of utilities for managing disk drives, printers, and other devices. Utility software may be considered as system software which is used quite often in developing, writing, debugging and documenting programs. It differs from application software mostly in terms of size, complexity and function. For example, word processors, spreadsheet programs, and database applications are considered application software because they are large programs that perform a variety of functions not directly related to managing computer resources. In simple words, Utility software supports the operations, management, and users of a computer system by providing a variety of support services. This software is sometimes installed as memory-resident program and may include file conversion, data compression; data transfer utility, text editor, real time clock utility and so on. In real sense, utility software is developed to perform operations on files containing data and be used for transferring data from one storage device to another. Utility software can be divided into two parts: First, file management utilities which are parts of operating system and help the programmers in copying, printing, deleting and renaming files. And second, program development utilities to help users in writing and testing programs. Some of the utility programs are discussed as follows: Editor: These are the interactive programs which allow programmers to enter, modify and store a source code.

121 File Manager: These programs allow programmers to create copy, update or delete files on the storage device. Loader: This program loads the machines codes of a program into the memory of the computer system. Locator: This program allocates specific memory addresses for each machine code of a program which is to be loaded into the system memory. Linker: This program links smaller programs (modules) to form a single program. (2) Application software Package: Application software package comprises programs that help people accomplish specific tasks. These packages are designed for all types of applications such as business applications, home applications, engineering designs, teaching aids etc. If an end user who controls the particular task at hand desires to use application software he or she has to define his/her own specific requirements for which the application program may be developed so as to produce the specific required results. These packages are often written by users to meet their own specific needs in high level language and are under the complete control of the operating system. An application software package may be classified into the following two categories: • General Purpose Application Software Package • Specific Purpose Application Software Package General Purpose Application Software Package These packages help in increasing the productivity of computer and are commonly used in day-to-day operations performed by the end users. So this type of application software is also known as program productivity tool or productivity package. Some of the general purpose application software packages which enable a user to effectively utilize a computer are: • Word Processing Software • Desktop Publishing Software • Spreadsheet Software • Data Base Management Software

122 • Graphics Software • Data Communication Software Word Processing Software:

Of all computer applications, word processing is the most common.

A word processor enables the user to create a

document, store it electronically on a disk,

display it on a screen, modify it by entering commands and characters from the keyboard, and print it

on a printer.

In contrast to data processing where the focus is generally on numerically data, in word processing the main concern is with text. The main productivity improvement comes from the ease and rapidity with which document can be modified. Word processing is designed to enable the user to create, edit, and print documents conveniently using a computer instead of using a typewriter. So

the great advantage of word processing over using a typewriter is that

the user can make changes without retyping the entire document. The user can correct the mistakes, delete a paragraph without leaving a trace, and insert a word, sentence, or paragraph in the middle of a document. Word processors also

make it easy to move sections of text from one place to another within a document, or between documents. The user can send the file with all the changes made to a printer to get a hardcopy or can save the file in a storage device. Mail-

merge is also one of the popular features of word processors. Now-a-days Word Processors for languages other than

English are also available e.g. Akshar in Hindi. Desktop Publishing Software: The line dividing word processors from desktop publishing systems is constantly shifting. In general, though, desktop publishing applications support finer control over layout, and more support for full-color documents. Combining the functions of a traditional typesetter and a layout artist, a desktop publishing program merges the output of word processor and graphics programs to produce professional-looking pages that are ready for the printer. Although many word processors can do this to some extent, desktop publishing programs have many more sophisticated features. Businesses use them to create advertisements and sales catalogs. Publishers use them to lay out news papers, bulletins, magazines, books etc. Examples of some desktop publishing packages are Corel's Ventura, Adobe's Page Maker, Quark Inc.'s Quark XPress, Microsoft Publisher, Akruti Publisher etc.

123 Spreadsheet Software: A spreadsheet is simply a sheet of paper with rows and columns in which one can enter data in the form of numbers, formulae and text for analysis. In spreadsheet software, screen is used in place of paper.

Spreadsheet applications (sometimes referred to simply as spreadsheets) are computer programs that display a large grid of rows and columns that allow users to quickly create, manipulate and analyze data organized in these spreadsheets electronically. These are very helpful where large tabulated computations are used. In a spreadsheet application, a table of values is arranged in rows and columns and each value can have a predefined relationship to the other values. Each value sits in a cell. The user can define what type of data is in each cell and how different cells depend on one another.

The relationships between cells are called formulas, and the names of the cells are called labels.

The user can also modify selected values to see how all the other values change accordingly. This enables the user to study various what-if scenarios. Spreadsheet packages are widely used in accounting, sales, inventory control, and a number of business and non business problems. There are a number of spreadsheet applications in the market, Lotus 1-2-3 and Excel being among the most famous. The more powerful spreadsheet applications support graphics features that enable the user to produce charts and graphs from the data to show the relationships among numbers more vividly. Most spreadsheet applications are multidimensional, meaning that the user can link one spreadsheet to another. A three-dimensional spreadsheet, for example, is like a stack of spreadsheets all connected by formulas. A change made in one spreadsheet automatically affects other spreadsheets. Data Base Management Software: This software was developed to handle the problems of maintaining and integrating large volumes of data on computers. It extends the ability to organize collections of data stored in computer and provides features that help in finding subsets of the data selected through specified criteria. Data Base Management programs are nothing else but a collection of data combined with a sophisticated management program to retrieve and process all this data to create 124 information for the users. Data Base Management Software (DBMS) is a collection of programs that enables the user to store, modify, and extract information from a database. It is very simple and easy to use. DBMS provides a very high level language interface which is command oriented. DBMS helps in making queries by arranging or sorting data in some logical order such as alphabetically by name and generating reports based on the data files. There are many different types of DBMSs, ranging from small systems that run on personal computers to huge systems that run on mainframes.

Examples of database applications include computerized library systems, automated teller machines, flight reservation systems, computerized parts inventory systems and so on. From a technical standpoint, DBMSs can differ widely. The terms relational, network, and hierarchical all refer to the way a DBMS organizes information internally. The internal organization can affect how quickly and flexibly the user can extract information. Requests for information from a database are made in the form of a query, which is a stylized question. For example, SELECT ALL WHERE NAME = "SANJAY" AND AGE > 35 This query requests all records in which the NAME field is SANJAY and the AGE field is greater than 35. The set of rules for constructing queries is known as a query language. Different DBMSs support different query languages, although there is a semi-standardized query language called SQL (structured query language). Sophisticated languages for managing database systems are called fourth-generation languages, or 4GLs for short. The information from a database can be presented in a variety of formats. Most DBMSs include a report writer program that enables the user to output data in the form of a report. Many DBMSs also include a graphics component that enables the user to output information in the form of graphs and charts. Graphics Software: There is a general saying that one picture is worth a thousand words as graphical or pictorial form is easier to understand than a tabular form of information. The graphics Software offers the use of computers for manipulating (creating, editing,

125 viewing, storing, retrieving and printing) images, designs, drawings, pictures, graphs and anything else that can be drawn in the traditional manner. While spreadsheet and database packages are very useful in doing analysis, the output of such analysis is generally tables of numbers. Graphics Software Package is used to convert the data in a graphic form on the screen as well as on the paper using printers or plotters. Now-a-days business organizations use presentation graphics application software packages which help in creating professional looking visual aids such as computer images, paper printouts, or photo transparencies for an audience. These are specialized type of graphics software to present data in the form of charts and graphs. These packages facilitate the users to quickly convert tabular data to graphical form. Sophisticated graphs and complex drawings can be drawn in multicolor by pressing a few keys of computer keyboard. The user can orchestrate images and even sounds to produce automated "slide shows". A graphics software package can also be used in text publishing. A desktop publishing software includes graphics to produce pictures and other forms of graphics in the text where needed. One more advantage of graphic software is the use of multimedia, a type of application that incorporates images, text, sound, computer animation, and video and the programs that create multimedia are known as multimedia authoring software. Another important application of graphics software is Computer Aided Design (CAD) software. CAD software is often used by the architects and engineers to draw designs of buildings or products before the actual construction or manufacturing gets under way. Engineering is such an area where graphics software contributes a lot in the form of engineering drawings and different types of graphs. Data Communication Software: Data communication software is that software which put one computer in touch with others and provides facility of transferring data from one machine to other. In this way more than one computer using a modem may communicate with each other with the help of this communication software. Network Software is another type of data communication software which allows users to share data by

126 creating a group of connected computers i.e. network of computers. Most popular network communication software, which are almost considered as a necessity for business computing are Novell Netware and Microsoft's Windows NT. Other data communication software may include fax software for computers to work as a fax machine and the software distributed by the online service providers, such as CompuServe, America Online, and Microsoft Network.

Specific Purpose Application Software Packages These software packages are not generalized but are only meant for performing uncommon applications. Application-specific software packages may be further classified as follows: • Business Application Package • Scientific Application Package • On line Application Package • Customized Application Package

Business Application Package: This software is nothing but packages designed for a specific purpose such as account receivable, billing, inventory control etc. These packages are written by users generally in high level language to meet their own specific needs. Some of the Business Application Specific Packages are as follows: Payroll: This is designed for calculation of wages and salaries. The extended information like relationship of total wage costs to job in hand, total costs to date, individual jobs costing etc. make it easier for management to improve its efficiency.

Management Information System: MIS provides comprehensive and up to date information to the management. With the help of this software the decision making process becomes easy and more reliable. Forecasting in advance helps the decision maker to plan his activities accordingly.

127 **Library Management System:** It maintains the list of books, books borrowed, new arrivals, remainder for overdue books etc. It is also helpful in classifying the books for easy reference.

Scientific Application Package: Such a package is specific only for the scientific purposes. This type of software facilitates the designing and drafting of complex images and analysis of data. Examples of these packages are Computer Aided Designing and Computer Aided Manufacturing (CAD/CAM), Statistical Package for Social Science (SPSS), Statistical Analysis Software (SAS) and so on.

On line Application Package: In an online application data is transmitted directly from its point of origin to the computer, and results are sent directly to the location where they are required. The response time of the computer can be specified with certain limits. No intermediate data preparation or data transmittal steps are necessary. The online application software facilitates different users in different locations to use the same computer and this computer will respond to all enquiries answering promptly. Examples of Online application are Satellite communication, Air Line Reservation Systems etc.

Customized Application Package: When pre-written software packages do not meet the specific requirements of a user, either partially or fully, it becomes necessary to develop a customized software package to satisfy the specific requirements. These types of software are either developed in-house or by placing order to some professional organization. This trend has geared up with the development of non-procedural 4GLs and will continue to grow as the development in multimedia, graphics, artificial intelligence, networking solutions enhance.

6.3 Summary Software is a program or set of logically arranged electronic instructions that direct the computer to accomplish certain tasks. When the computer is using a particular program, it is referred to as running or executing the program. The entire set of programs, 128 procedures, and related documentation associated with a computer system controls the operation of a computer and are distinguished from the actual hardware of the computer. A software package is a special method for the distribution and installation of software on computer systems. It is a program with its associated files, archived and compressed into a single file along with any necessary scripts or information to aid in managing the installation, upgrade, and removal of those files. Software Packages fall in two major categories: (1) System software Package, and (2) Application software. Package System software packages are for the own specific requirements such as controlling or supervision of computer system and may include operating system, language translators and utilities. The operating system tells the computer how to interact with the user and how to use the hardware devices attached to the computer. Language translators convert computer languages other than machine level language to machine codes. They include compilers/interpreters and assemblers. Compilers/interpreters convert high level language programs into machine level language and assemblers convert programs written in assembly language into machine level language. Utilities are the programs that support the operations, management and users of a computer system by providing a variety of support services. Application software packages are developed for the specific requirements of a user or any organization. Application software packages are mainly of two types: General purpose software and Specific purpose software. General purpose software includes word processors, spreadsheets, data base management software, graphics software and data communication software. This type of software packages are also known as program productivity tools because without doing any programming and through simple commands the users can perform the desired tasks. The specific purpose or application specific software may include business applications like inventory control, billing, MIS etc., scientific application like SPSS, CAD/CAM etc., online applications like Airline Reservation Systems, and customized applications are also referred to as application specific software package.

129 6.4 Self-Test questions 1. What is a software Package? 2. List out the different types of software. 3. Describe the similarities and dissimilarities between a compiler and an interpreter. 4. What is utility software? 5. Discuss various types of system software. 6. What is customized software? 6.5 Suggested Readings 1. Gill, N.S., "Essentials of Computer Fundamentals", Khanna Book Publishing Co. (P) Ltd. 2. Norton, Peter, "Introduction to Computers", 2 nd Ed., Tata Mcgraw Hill. 3. Ram, B., "Computer Fundamentals: Architecture and Organization", 3 rd Ed., New Age Publishers. 4. Sinha, Pradeep K. and Sinha, Priti, "Computer Fundamentals", 3 rd Ed., BPB Publications. 5. V. Raja Raman, "Fundamental of Computers", PHI.

130

UNIT 7 -

OPERATING SYSTEMS Objectives: The motive of this lesson is to present an overview of Operating System and within that framework the key concepts, functions and different types of operating systems are discussed. Structure: 7.1 Introduction 7.2 Concept and Definitions of Operating System 7.3 Booting or Bootstrapping of Operating System 7.4 Functions of Operating System 7.5 Types of Operating System 7.6

Operating Systems (DOS and Windows) for Personal Computers 7.7 Future Trends in Operating System Design 7.8 Summary 7.9 Self-Test questions 7.10 Suggested Readings 7.1 Introduction

An Operating System is system software which may be viewed as an organized collection of software consisting of procedures for operating a computer and providing an environment for execution of programs.

It acts as an interface between user and the hardware of a computer system.

The operating system (OS) is the first thing loaded onto

the computer. The most important program that runs on a computer

and without the operating system,

a computer is useless. Every general-purpose computer must have an operating system to run other programs.

Operating systems

perform

basic tasks, such as recognizing input from the keyboard, sending output to the

display screen, keeping track of files and directories on the disk,

and controlling peripheral devices such as disk drives and printers.

Any desktop or laptop PC that you buy normally comes pre-loaded with

131 Windows OS. Macintosh computers come pre-loaded with OS X. Many corporate servers use the Linux or UNIX operating systems. Understanding the operations and functions of operating system used in computers for a user is quite important as per the following reasons: •

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

The purpose of an operating system is to organize and control hardware and software so that the device it lives in behaves in a flexible but predictable way. In this lesson, we will learn what a piece of software must do to be called an operating system, and to know how the operating system in our desktop computer works and how many types of operating systems are around. 7.2 Concept and Definitions of Operating System Operating System is that system software which when fed into the memory of computer enables the user to read or write on the computer. It acts as an interface between hardware and the user. All the hardware functioning are performed under the supervision of this operating system. It is always desired by the user that the operating system should be small in size and easily maintainable so that operations can be performed in an efficient manner with more reliability. For large systems, the operating system has even greater responsibilities and powers. It makes sure that different programs and users

running at the same time do not interfere with each other.

The operating system

is also responsible for security, ensuring that unauthorized users do not access the system.

Following are the main characteristics of an operating system: • Sharing of resources • Concurrency

132 • Long-term Storage • Non-determinacy We can define Operating System in the following manner: Operating System may be defined as system software which

controls all the resources of computer and provides a base upon which application programs can be

developed and handled. An Operating System may also be defined as a set of programs/procedures (manual and/or

automatic) that enables user or a user group to share a computer installation efficiently. An Operating System may also be viewed as an organized collection of system programs consisting of control routines

for operating a computer and for providing an environment for execution of programs. An Operating System (OS) is

a system program different from word processing programs, spreadsheets and all the other software programs. OS is the master control program. The Operating System is involved in performing the following activities:

- OS provides the instructions to display the on screen elements with which the user interact. Collectively, these elements are known as the user interface.
- OS loads programs such as word processing and spreadsheet programs into the computer's memory to be used by the users.
- OS coordinates how programs work with the central processing unit, keyboard, mouse, printer, and other hardware as well as with other software.
- OS manages the way information is stored on and retrieved from disks.

133 The Operating System mainly consists of the Control Programs and the Processing Programs.

- Control programs include Supervisory programs, Job Control programs and Input/Output Management programs. These programs help in supervising the system operations and perform tasks such as Input/output, scheduling, handling interrupts, and communication with computer users or programmer.
- Processing programs include language translator programs, link editors, library programs and a set of utility programs which are executed under the supervision of control programs. These programs are used by the programmers to simplify program operations for the computer system.

Operating systems have also two basic unrelated dimensions:

- $\frac{3}{4}$ Operating System as an Extended Machine
- $\frac{3}{4}$ Operating System as a Resource Manager

Operating System as an Extended Machine The Operating System isolates the hardware from the user. The user communicates only with the OS and it links the various languages and the systems and applications programs available. The OS performs all requests for services and totally eliminates

134 the need for the user to deal with the hardware or related problems. Thus the operating system allows the user to concentrate on his application and prevents him for having to spend a lot of time determining how things are to be specifically accomplished by the hardware.

Operating System as a Resource Manager An operating system's major function is to manage the resources of the computer system. It allocates the resources on the basis of the user need and system capability. As a Resource Manager, the operating system must perform the following functions for each resource viz. processor, memory, I/O devices and file systems:

- Keep track of status of each resource
- Decide which job should get the resource and for how much time based on certain policy
- Allocate the resource to the job decided
- After the resource is used by the job for the allocated time, deallocate the resource.

7.3 Booting or Bootstrapping of Operating System

The process of loading the operating system into memory is known as booting.

When a computer is turned on the operating system must be brought into the memory of computer from the hard-disk storage. The process is normally started by a small program called bootstrap loader. This program resides in a ROM as a firmware.

The bootstrap loader is a small program that has a single function: It loads the operating system into memory and allows it to begin operation. When the power is turned on of

a computer, the first program that runs is usually a set of instructions kept in the computer's read-only memory. This code examines the system hardware to make sure everything is functioning properly. This power-on self test (POST) checks the central processing unit, memory, and basic input-output systems (BIOS) for errors and stores the result in a special memory location. Once the power-on self test (POST) has successfully completed, the software loaded in read- only memory (sometimes called the BIOS or firmware) will begin to activate the

135 computer's disk drives. In most modern computers, when the computer activates the hard disk drive, it finds the first piece of the operating system: the bootstrap loader. Bootstrap Loader is a very simple program sufficient only to direct the CPU to look for specific file on the disk memory, and executes the instructions stored in the file. The file contains machine codes to direct the CPU to continue loading the rest of the operating system into the memory. When the operating system is fully loaded into the memory, the computer is ready to accept user's commands. In the most basic form, the bootstrap loader sets up the small driver programs that interface with and control the various hardware subsystems of the computer. It sets up the divisions of memory that hold the operating system, user information and applications. It establishes the data structures that will hold the myriad signals, flags and semaphores that are used to communicate within and between the subsystems and applications of the computer. Then it turns control of the computer over to the operating system.

7.4 Types of Operating System

Within the broad family of operating systems, there are generally five types, categorized based on the types of computers they control and the sort of applications they support. The

broad categories are: 1. Batch Operating Systems 2. Multiprogramming Operating Systems 3. Real-time operating system 4. Network Operating Systems 5. Distributed Operating Systems Batch Operating Systems This type of OS requires grouping of similar jobs consisting of programs, data, and appropriate system commands in the form of a batch to be dealt on the computer system. The suitability of this type of processing is in the programs with large computation time with no need of user interaction or involvement. Users are not required to wait while the job is being processed. The users can submit their programs

136 to operators and return later to collect them. Some of the examples of such programs which accept the batch processing environment are payroll, forecasting, statistical analysis and large scientific number-crunching programs. There are some difficulties with a batch operating system from the point of view of programmer or user. Firstly, batch operating systems provide a non interactive environment i.e. allow little or no interaction between user and executing programs. Secondly, due to turnaround delays and offline debugging, batch processing is not very convenient for program development. Therefore, only the programs that do not require interaction and programs with long execution times may be served well by a batch operating system. Multiprogramming Operating System These OS are fairly sophisticated as compared to batch operating system. It allows two or more users to run programs at the same time. Some operating systems permit hundreds or even thousands of concurrent users. In addition to supporting multiple concurrent processes (i.e. several processes in execution states simultaneously), multiprogramming operating system allows the instruction and data from two or more separate processes to reside in primary memory simultaneously. Multiprogramming has a significant potential for improving system throughput and resource utilization with very minor differences. Different forms of Multiprogramming operating system are: Multi-tasking operating system: Allows more than one program to run concurrently. Multi-process operating system: Supports running a program on more than one CPU. Multi-threading operating system: Allows different parts of a single program to run concurrently. Multi-user operating system: Permits hundreds or even thousands of concurrent users to run programs. Multi-tasking operating system: A running state of a program is called a process or a task. The ability to execute more than one task at the same time, a task being a program is known as multi-tasking processing. A multitasking operating system has the capability to execute more than one task concurrently on a computer system

137 having a single CPU. In these operating systems, CPU switches from one task to another so quickly that it gives the illusion to the user of executing all the programs (or tasks) at the same time.

This is the type of operating system most people use on their desktop and laptop computers today.

Microsoft's Windows and Apple's Mac OS platforms

are both examples of operating systems that will let a 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.

There are two basic types of multitasking: preemptive and cooperative. In preemptive multitasking, the operating system parcels out CPU time slices to each program. In cooperative multitasking, each program can control the CPU for as long as it needs it.

If a program is not using the CPU, however, it can allow another program to use it temporarily. OS/2, Windows 95, Windows NT, the Amiga operating system and UNIX use preemptive multitasking, whereas Microsoft Windows 3.x and the MultiFinder (for Macintosh computers) use cooperative multitasking. The terms multitasking and multiprocessing are often used interchangeably, although multiprocessing implies that more than one CPU is involved. Multi-processing operating system: These OS have the ability to support more than one process at the same time i.e. allowing more than one programs to run concurrently on the multiple CPUs in a single computer system. Multiprocessing systems are much more complicated than single-process systems because the operating system must allocate resources to competing processes in a reasonable manner. This type of operating system differs from multi-user operating system in the way that these refer to utilization of multiple CPUs. This is also called parallel processing. The examples of this type of operating system are OS/2 for high-end PCs, UNIX, MVS etc. Multi-threading operating system: This type of OS permits different portions of a single program to run concurrently and thus it is having the ability of an operating system to execute different parts of a program (also called threads) simultaneously. It

138 requires the programmer to design the program so carefully that all the threads can run at the same time without interfering with each other. Multi-user operating system:

A multi-user operating system allows many different users to take advantage 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.

These OS refer to computer systems that support two or more simultaneous users and allow simultaneous access to a computer system through two or more terminals. Although frequently associated with multiprogramming, multi-user operating system does not imply multi-programming or multi-tasking. A dedicated transaction processing system such as railway reservation system that supports hundreds of terminals under control of a single program is an example of multi-user operating system. All mainframes and minicomputers are multi-user systems, but most personal computers and workstations are not. Another term for multi-user is time sharing. The words multiprogramming, multiprocessing and multitasking are often confused. There are, of course, some distinctions between these similar, but distinct terms. The term multiprogramming refers to the situation in which a single CPU divides its time between more than one jobs. The term multitasking is described as any system that runs or appears to run more than one application program at one time. Whereas in multiprocessing, multiple CPUs perform more than one job at one time. Time-sharing is a special case of multiprogramming, where a single CPU serves a number of users at interactive terminals. Time-sharing operating system: These operating systems allow two or more users to run their respective programs at the same time by making these programs resident in the memory of computer. The concept of allowing more than one programs in the memory of computer is called multiprogramming. Therefore this OS is a form of

139 multiprogramming operating system which operates in an interactive mode with a quick response time. All time-sharing systems need to be multi-programmed but not necessarily all the multi-programmed systems to be time shared. In this type of OS, the user types a request to the computer through a keyboard. The computer processes it and a response is displayed on the user's terminal. A time sharing system allows the many users to simultaneously share the computer resources. Since each action or command in a time-shared system take a very fraction of time, only a little CPU time is needed for each user. As the CPU switches rapidly from one user to another user, each user is given impression that he has his own computer, while it is actually one computer shared among many users. Time-slice scheduling of CPU approach is used for time sharing processing in which programs are executed with rotating priority that increases during waiting and drops after the service is granted. In order to prevent a program from monopolizing the processor, a program executing longer than the system defined time-slice is interrupted by the operating system and placed at the end of the queue of waiting program. The examples of this type of operating system are VMS (on DEC Mainframes), UNIX, XENIX, MVS etc. Real-time operating system (RTOS) Real-time operating systems are used to control machinery, scientific instruments and industrial systems.

This OS

has very little user-interface capability, and no end-user utilities, since the system will be a "sealed box" when delivered for use.

A very important part of this type of OS

is managing the resources of the computer so that a particular operation executes in precisely the same amount of time every time it occurs.

In a complex machine, having a part move more quickly just because system resources are available may be just as catastrophic as having it not move at all because the system is busy.

These OS

are used in environments where a large number of events mostly external to computer systems, must be accepted and processed in a short time or within certain deadlines. The primary objective of real-time operating systems is to provide quick response time and thus meet the scheduling deadlines. User convenience and resource utilization are of secondary concern to real-time system. Real time can also refer to

140 events simulated by a computer at the same speed that they would occur in real life. In graphics animation, for example, a real-time program would display objects moving across the screen at the same speed that they would actually move. The major positive features of this type of operating system are fast response time, feedback mechanism and reliability. Real-time operating systems respond to input immediately. They are used for such tasks as navigation, in which the computer must react to a steady flow of new information without interruption. Most general-purpose operating systems are not real-time because they can take a few seconds, or even minutes, to react. Examples of such applications are flight control, Industrial Control, Telephone Switching Equipments, Real-time Simulations etc. Real time systems are also frequently used in military application. In case of real-time systems, memory management and file management are of least importance as compared to processor and I/O management therefore real-time operating systems are different from traditional operating systems like MS-DOS and UNIX or XENIX. Some of the examples of real-time operating systems are: Operating System for IBM System/3: Communication Control Program (CCP), Operating System: Duplex Multiple Environment Real-time (DMERT) etc. Network Operating System It is a collection of system software and associated protocols that allow a set of autonomous computers which are interconnected by a computer network to be used together in a convenient and cost-effective manner. In this OS, the users are aware of existence of multiple computers and can log into remote machines and copy files from one machine to another machine. Users are typically aware of where each of their files are kept and must move file from one system to another with explicit transfer commands instead of having file placement managed by the operating system. The system has little or no fault tolerance; only that personal computer will be out of business which crashes. A network operating system offers the following capabilities:

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

Distributed Operating System In this type of OS users view the system as a virtual uniprocessor but not as a collection of distinct machines. In a true distributed system, users are not aware of where their programs are being run or where their files are residing; they should be handled automatically and efficiently by the operating system. Thus the OS looks to its users as an ordinary centralized operating system but runs on multiple independent CPUs. The main focus here is transparency. In other words, component and resource distribution should be hidden from users and application programs unless they explicitly demand otherwise. These operating systems usually provide the means for system-wide sharing of resources, such as computational capacity, files and I/O devices. The main advantage of distributed operating systems is that there is a little fault-tolerance and are considered to be more reliable than uniprocessor based systems. They perform even if certain part of the hardware is malfunctioning. On the other side, the disadvantage is that unless one is careful, it is easy for the communication protocol overhead to become a major source of inefficiency. With the different types of operating systems in mind, it is time to look at the basic functions provided by an operating system.

7.5 Functions of Operating System

An operating system concerns itself with every single and minute detail of computer working. It manages everything that runs on computer straight from running application software, entering data, displaying information on monitor, printing a report to store data on external storage device (disk). It does many jobs on a user's behalf without showing him each and every step of processing. For example, if someone gives a command to open a document file in a word processor, the document

142 is immediately displayed on the screen. But, how this file is opened from the disk (storage device) and how information is displayed on the monitor (output device). All these types of functions are performed by the operating system. In order to enable computer to effectively and efficiently utilize its resources to monitor the system performance and to provide services to the users, an operating system is needed. So, an operating system performs basically three types of functions.

(a) Essential Functions: The essential functions of an operating system are concerned with effective utilization of computer resources. Storage management and processes management are two important essential functions of an operating system. Storage management is concerned with allocation and reclamation of storage when a program is initiated and terminated. Processor management is concerned with scheduling of programs in a time sharing system.

(b) Monitoring Functions: These functions are concerned with collection of resource utilization and system performance information.

CPU and IO devices form the resources whose utilizations are collected and monitored by device management functions of operating system.

(c)

Service Functions: These functions of operating system enhance facilities provided to the users. They include mainly automatic procedures for recovery due to hardware errors, file editing facilities and file organization and access methods. These functions of operating system, in the most general sense, can be classified into six categories:

- Processor management
- Memory and Storage management
- File management
- Device Management
- Security
- Command Interpretation

While there are some who argue that an operating system should do more than these six tasks, and some operating-system vendors do build many more utility programs

143 and auxiliary functions into their operating systems, these six tasks define the core of nearly all operating systems. Let's look at the tools the operating system uses to perform each of these functions. Processor Management: The process management module of an operating system takes care of the creation and deletion of processes, scheduling of various resources to the different processes requesting them, and providing mechanisms for synchronization and communication among processes.

The heart of managing the processor comes down to two related issues: • Ensuring that each process and application receives enough of the processor's time to function properly. • Using as many processor cycles for real work as is possible. It is processes, rather than applications, that the operating system controls and schedules for execution by the CPU. In a single-tasking system, the schedule is straightforward. The operating system allows the application to begin running, suspending the execution only long enough to deal with interrupts and user input. Memory and Storage Management: The

memory management module of an operating system takes care of the allocation and deallocation of memory space to the various programs in need of this resource. When an operating system manages the computer's memory, there are two broad tasks to be accomplished: • Each process must have enough memory in which to execute, and it can neither run into the memory space of another process nor be run into by another process. • The different types of memory in the system must be used properly so that each process can run most effectively. Disk storage is only one of the memory types that must be managed by the operating system, and is the slowest. Ranked in order of speed, the types of memory in a computer system are: High-speed cache - This is fast, relatively small amounts of memory that are available to the CPU through the fastest connections.

144 Main memory - This is the RAM that you see measured in megabytes when you buy a computer. Secondary memory - This is most often some sort of rotating magnetic storage that keeps applications and data available to be used, and serves as virtual RAM under the control of the operating system. The operating system must balance the needs of the various processes with the availability of the different types of memory, moving data in blocks between available memories as the schedule of processes dictates. File Management: The file management module of an operating system takes care of file related activities such as organization, storing, retrieval, naming, sharing, and protection of files. The important concepts used for file management are as follows: File Access Methods: To use the information stored in a file, it must be accessed and read into computer memory. The two commonly supported file access methods at operating system level are sequential and random access. File Operations: An operating system provides a set of operations to deal with files and their contents. File operations like creating, renaming, copying, deleting, opening, closing, reading, writing, seeking the files or setting the attributes of the files are some of the operations provided by operating system. File Naming: When a file is created, it is given a name, which can be later used to access the file. The rules for naming files vary from one operating system to another. Device Management: Device Management refers to coordination and control of various I/O devices and is an important function of the operating system. This involves receiving the request for I/O interrupts and communicating back to the requesting process. The path between the operating system and computer hardware goes through a special program called a driver. Much of a driver's function is to be the translator between the electrical signals of the hardware subsystems and the high-level programming languages of the operating system and application programs. Drivers take data that the operating system has defined as a file and translate them into streams of bits placed in specific locations on storage devices, or a series of laser pulses in a printer.

145 One reason that drivers are separate from the operating system is so that new functions can be added to the driver and thus to the hardware subsystems without requiring the operating system itself to be modified, recompiled and redistributed. Through the development of new hardware device drivers, development often performed or paid for by the manufacturer of the subsystems rather than the publisher of the operating system, input/output capabilities of the overall system can be greatly enhanced. Security Management: The security module of an operating system protects the resources and information of a computer system against destruction and unauthorized access.

A total approach to computer security involves both external and internal security. External Security: It deals with securing the computer system against external factors such as fires, floods, leaking out of stored information by a person who has access to the information and so on. Internal Security: It mainly deals with User authentication (user identification must be checked), Access Control (prohibit the user from accessing those information for which not authorized to access) and cryptography (a method to ensure access control). Command Interpretation: The command interpretation module of an operating system takes care of interpreting user commands given to computer for getting some job done by it. The commands supported by the command interpretation module are known as system calls. When a user gives instructions by using these system calls, the command interpreter takes care of interpreting these commands and directing the system resources to handle the requests. With this mode of interaction with the system, the user is usually not too concerned with the hardware details of the system. In this manner, it greatly contributes to the 'ease of use' objective of an operating system.

146 7.6 Operating Systems (DOS and Windows) for Personal Computers Although a number of companies have offered operating systems over the years, one company, Microsoft Corporation, has dominated the market for microcomputer operating systems. In practice, their products have evolved from one to another in a reasonably sequential fashion. The following is a brief summary. MS-DOS: MS-DOS stands for Microsoft Disk Operating System and is a single user operating system developed by Microsoft. Microsoft is a software company in Redmount, Washington, U.S.A. It was introduced in 1981 jointly by Microsoft and IBM and was the most popular operating system for personal computers in 1980s. Some of the disk and file maintenance commands of MS-DOS are as follows: DIR To list the directory TREE To list all disk subdirectories and their subdirectories TYPE To see the contents of a file on the screen REN To rename a disk file DEL To erase files from a disk COPY To copy files from one location of disk to another DISKCOPY To copy the entire diskette FORMAT To realign the tracks and sectors of the diskette CHKDSK To check the disk space and get information of the diskette VER To see the version of the operating system being used DOS was adequate for the IBM-compatible PCs of the early 1980s, but it has certain limitations that became noticeable as PCs became more powerful. The limitations of this OS are: 1. Under DOS, only a single program can be loaded into memory at a time. 2. DOS was not designed to handle the large amounts of RAM that today's PCs typically use. 3. DOS was designed for 16-bit CPUs and can not take advantage of the 32-bit architecture of the 486, Pentium, and later chips. 4. DOS filenames are limited to eight characters, plus a three character "extension" following a period sign.

147 5. DOS command-line interface is more difficult to learn than a well designed GUI. Microsoft Windows: Microsoft Windows operating system was developed by Microsoft to overcome the limitations of its own MS-DOS operating system. The first successful version of this operating system was Windows 3.0, which was released in 1990. The subsequently released versions were Windows 95, Windows 98, Windows 2000, and Windows XP. The main features of Microsoft Windows are as follows: 1. Its command-line interface is GUI. Hence, it is easier for a new user to learn and use the system. 2. Microsoft Windows was designed to be not just an operating system, but also a complete operating environment. For example, Microsoft word processor works similarly the way a Microsoft spreadsheet works. This means that the experience gained by learning one Windows program is useful while using any other Microsoft windows program. 3. It is a single user, multi-tasking operating system. It means that a user may run more than one program at a time. Microsoft Windows NT: It is a 32-bit multi-user, timesharing operating system developed by Microsoft. Its use is not restricted to only Intel's microprocessors but can also run on microprocessors of other companies like DEC's Alpha processor, MIPS processors etc. It was designed to have UNIX like features so that it can be used for powerful workstations, networks and database servers. 7.7 Future Trends in Operating System Design Operating system technology is one of the fastest-moving areas in the microcomputer world. Almost every year, the manufacturers of the PC operating systems, primarily Microsoft, IBM, and Apple, come up with new, more capable versions of their operating systems. Current systems already offer the ability to connect with a network and share files and programs with others while protecting against loss of data. New versions will be increasingly good at networking, so that working in a workgroup on a

148 network will become as easy as working on a stand alone PC. However, a number of clear trends are emerging which will point the way the future operating systems will be designed. These are as follows: • Microcode will support all the functions of operating system integrated into it, currently performed by software, therefore, the execution time will be faster. • Use of distributed operating system will increase because the new trends are towards distributed control among the localized processors in place of centralized system. • The concurrency is becoming an important feature of programming language. Hardware and operating systems are being designed to execute concurrent programs more efficiently. • Lots of developments are taking place in computer networking. The data transmission rate is also increasing. Therefore, the use of networking operating system will also increase. More recently, operating systems have started to pop up in smaller computers as well. If you like to tinker with electronic devices, you are probably pleased that operating systems can now be found on many of the devices we use every day, from cell phones to wireless access points. The computers used in these little devices have gotten so powerful that they can now actually run an operating system and applications. The computer in a typical modern cell phone is now more powerful than a desktop computer from 20 years ago, so this progression makes sense and is a natural development. In any device that has an operating system, there is usually a way to make changes to how the device works. This is far from a happy accident; one of the reasons operating systems are made out of portable code rather than permanent physical circuits is so that they can be changed or modified without having to scrap the whole device. For desktop systems, access to a LAN or the Internet has become such an expected feature that in many ways it is hard to discuss an operating system without making reference to its connections to other computers and servers. Operating system developers have made the Internet the standard method for delivering crucial

149 operating system updates and bug fixes. Although it is possible to receive these updates via CD, it is becoming increasingly less common. In fact, some entire operating systems themselves are only available through distribution over the Internet. Further, NetBooting allows the operating system for one computer to be served over a network connection, by a remote computer connected anywhere in the network. One NetBoot server can serve operating systems to several dozen client computers simultaneously, and to the user sitting in front of each client computer the experience is just like they are using their familiar desktop operating system like Windows or Mac OS.

7.8 Summary
An Operating System is system software which may be viewed as an organized collection of software consisting of procedures for operating a computer and providing an environment for execution of programs.

It acts as an interface between user and the hardware of a computer system.

All the hardware functioning are performed under the supervision of this operating system. It is always desired by the user that the operating system should be small in size and easily maintainable so that operations can be performed in an efficient manner with more reliability. The Operating System mainly consists of two types of programs: the Control Programs and the Processing Programs.

The process of loading the operating system into memory is known as booting.

When a computer is turned on the operating system must be brought into the memory of computer from the hard-disk storage. The process is normally started by a small program called bootstrap loader. Operating system provides a number of services. At the lowest level, there are system calls, which allow a running program to make a request from operating system directly. At a higher level, there is a command interpreter, which supports a mechanism for a user to issue a request without writing a program. Operating systems perform following two basically unrelated functions: ¾ Operating System as an Extended Machine

150 ¾ Operating System as a Resource Manager
Operating System may be classified in the following categories: 1. Batch Operating Systems 2. Multiprogramming Operating Systems 3. Real-time operating system 4. Network Operating Systems 5. Distributed Operating Systems

The functions of operating system fall into these categories: • Processor management • Memory and Storage management • File management • Device Management • Security • Command Interpretation
Operating system technology is one of the fastest-moving areas in the microcomputer world. Almost every year, the manufacturers of the PC operating systems, primarily Microsoft, IBM, and Apple, come up with new, more capable versions of their operating systems. Although a number of companies have offered operating systems over the years, one company, Microsoft Corporation, has dominated the market for microcomputer operating systems. In practice, their products have evolved from one to another in a reasonably sequential fashion.

7.9 Self-Test questions
1. What is an operating system? 2. How operating system works as a resource manager? 3. What are the main functions of operating system? 4. Differentiate the following operating systems: i. Multi-programming and Multi-tasking ii. Time-sharing and Real Time

151 iii. Network and Distributed 5. Give a brief idea of Microsoft operating systems for personal computers. 7.10 Suggested Readings 1. Gill, N.S., "Essentials of Computer Fundamentals", Khanna Book Publishing Co. (P) Ltd. 2. Milan Milenkovic, "Operating Systems Concepts and Design", 2 nd Ed., Tata McGraw Hill 3. Norton, Peter, "Introduction to Computers", 2 nd Ed., Tata McGraw Hill. 4. Ram, B., "Computer Fundamentals: Architecture and Organization", 3 rd Ed., New Age Publishers. 5. Sinha, Pradeep K. and Sinha, Priti, "Computer Fundamentals", 3 rd Ed., BPB Publications. 6. V. Raja Raman, "Fundamental of Computers", PHI.

152

UNIT 8 -

INTRODUCTION

TO WORD PROCESSOR Objectives: The motive of this lesson is to present an introduction of Word Processor Software and within that framework the key concepts, applications and functioning of word processor software are discussed.

Structure: 8.1 Introduction 8.2 Concept of Word Processing Software 8.3 Getting Started with Microsoft Word 8.4 Structure of Microsoft Word Window 8.5 Working with Word Document - File Related Operations 8.6 Some Important Editing Functions 8.7 Mail Merge 8.8 Summary 8.9 Self-Test questions 8.10 Suggested Readings 8.1 Introduction Word processing software is one of the miracles of the modern age. It evolved from the needs of writers as word processing software gives users an extensive set of tools for working with text. It is used to create all kinds of documents, from simple notes and memos, to brochures, resumes, and long reports. A word processor is a computer program used to create and print text documents that might otherwise be prepared on a typewriter. The key advantage of a word processor is its ability to make changes easily, such as correcting spelling, adding, deleting, and relocating text. Once created, the document can be printed quickly and accurately and saved for later modifications.

153 The most recent word processors are practically powerful editing devices that only a few years ago would have been the envy of professional printers. But it is not only a matter of graphical arrangement and page layout, the effects of writing through this medium are far reaching on a number of more profound levels. A number of softwares are available for word processing like Microsoft Word, Lotus WordPro or WordPerfect etc. Although these are expensive and sophisticated products, many of the facilities of word processing are also to be found in free software, like Microsoft WordPad, which comes as part of the Microsoft Windows standard installation. If one cannot afford to purchase a word processing application, then there are many powerful programs free, either as downloads from the Web or on cover disks with magazines. A word processing program is a good introductory program to learn for two important reasons: • Word processors are good examples of competing programs that all run and work in essentially the same way. Learning one program allows us to use almost all of the competing programs. • Skills such as saving, copying, pasting, formatting, printing, and using old files as templates for new work are easy to understand within word processors. Many of the skills utilized in word processing programs are used in other popular applications such as spreadsheets, web browsers and e-mail programs. 8.2 Concept of Word Processing Software The word processor is the modern version of paper, pen, typewriter, eraser, dictionary and thesaurus... all in one. They were introduced on the market in the 1960s and became a typical office tool by the 1970s. Modern word processors display documents in graphics mode, thus enhancing the WYSIWYG function (What You See Is What You Get); this means that the way the text and images are displayed on the screen is the way they will appear when printed. Their formatting features include font changes, page layout, and paragraph indentation. They also check spelling, find synonyms, incorporate graphics, perform calculations, and so on.

154 The basic concept of word processing software is that one can type the text into the computer, instead of onto paper. Then errors can be corrected, text can be rearranged and all sorts of modifications can be made. It can then be seen on the screen and if we are satisfied with it, then only take a print-out on paper. It is a four steps process-type, store, edit, and print. A word processor is software that enables us to write, edit, format, and print text. As we type on a keyboard, the words appear on the screen to see. Mistakes can be corrected easily before printing the text on paper. Words, sentences, and even entire paragraphs can be moved by special commands. Nothing is printed until we are satisfied with the results. Word processor and the hardware simply provide a convenient way to display, store, edit, and recall the work done. Standard Features of Word Processing Software • Word-wrap: automatic arrangement of text in lines of specified length without the necessity of touching the return key. • Discretionary Hyphenation: option of inserting a hyphen to break a word that ends a line: the hyphen does not print if later editing moves the word to the middle of a line. • Justification: automatic alignment of text to both the left and right margins. • Adjustment: realignment of text to new margin and tab settings. • Alignment: positioning text or numbers to specified margin and tab settings. • Decimal Alignment: positioning columns of numbers with the decimal points vertically aligned. • Indents: the setting of temporary margins within a document differing from the primary margins used. • Centering text on a line. • Insertion: the entry of new text within previously typed material without erasing the existing material. • Overstriking: the substitution of new text for old by typing over the old text. • Deletion: erasure of text from the screen, or of whole documents from the disk.

155 • Search and Replace: moving directly to specified words or parts of words within a document and replacing them with different words or word portions. • Copying or Cutting: the duplication or moving of blocks of text within a document. • Boilerplate: the separate storage and retrieval of blocks of text from which standard documents can be built. • Pagination: automatic division of a document into pages of specified numbers of lines. • Page Numbering: automatic sequential numbering of pages. • Line spacing: Line spacing or leading is used to change the amount of space between lines of text. Normal text is single line spaced. This can be altered to 1.5 times or double or a number of points. Leading is a term which originates from typesetters who used strips of lead to place in-between the lines of text to increase the amount of space between each line. • Headers and Footers: option of creating standard blocks of text that will automatically appear at the top or bottom of each page in a document. • Footnoting: automatic sequential numbering of footnotes and positioning of the footnotes at the bottom of their appropriate pages during pagination. • Table of Contents and Index Generators. Programs that create these based on the text of a document. • Form Letter Merging: automatic combining of a form letter with a mailing list to generate multiple copies of the letter with the different addresses and other variable information filled in. • Automatic Spelling Checker and Corrector. Program that compares words in the text against an on-line dictionary, flagging items not found in the dictionary and offering alternative spellings and a means of correcting the errors. In this chapter, we will learn how to use Microsoft Word assaulted with the above features.

8.3 Getting Started with Microsoft Word

156 Before getting started with Microsoft Word, we need to locate and open it from the computer desktop. On computer, we should be able to see a MS Word icon on the desktop. From the computer desktop: 1. Double-click on the MS Word icon Go to the Start menu if the MS Word icon is not on the desktop: 1. Click ► Start ► Programs ► Microsoft Word MS Word will open a blank page called Document 1.

8.4 Structure of Microsoft Word Window

When you open the word a blank document opens. You can start typing on it directly. But for performing various other options you have to learn about the different components in the window and the options under these bars.

157 1.

Title Bar: Title bar shows the name of the document and is situated in the top of the window application. 2.

Menu Bar: Menu bar contains the various commands under the various topics to perform some special tasks. Menu bar is located under the title bar. 3. Standard Toolbar: Toolbar is

nothing more than the shortcut of the Menu options to execute or perform the menu options in the easiest and fastest way. Standard toolbar contains

the commands

for the Word's most frequently used commands. 4. Formatting Toolbar: Formatting toolbar contains the various

commands related to the formatting of the document. 5. Ruler: From Ruler margins and indents can be adjusted in the

easier way and it also provides measurement for the page formatting. Microsoft Word Window 6. Cursor: Cursor is Word pointer, which tells that from or on the cursor position you can perform your action. 7. Drawing Toolbar: Drawing toolbar

has the various

commands related to the graphics and the drawing options. 8.

Status Bar: It displays the positioning of the cursor, displays the status of some important keys of keyboard, shows the messages for the toolbar

button when a

mouse

158 points to it, displays messages for menu option when a menu option is selected or pointed out by a user. 9.

View Buttons: View buttons are shortcuts of various views in the View Menu. 10.

Scroll Bars:

There are two types of scroll bars i. Horizontal Scrollbar ii. Vertical Scrollbar

Working of both the scroll bar is to scroll the document in either direction. 11.

Document Navigator: Document Navigator allows navigating the document in different types of objects and is activated when clicked on the ball type button on the vertical scroll bar. 12. Office Assistant: Office assistant provides you the

online help, real-time tips while working.

Common Key Definitions in a Word Processor Action Key or Key Combination Moves Left or Right one character ← Or → Moves

Left or Right one word

Ctrl + ← or Ctrl + →

Moves Up or Down one line ↑ Or ↓ Moves Up or Down one paragraph Ctrl + ↑ or Ctrl + ↓

Moves to the

start or end of a line Home or End Moves Up or Down one screen Page up or Page Down

Moves to the

top or bottom of the current screen Ctrl + Page up or Ctrl + Page Down

Moves to the

start or end of the document Ctrl + Home or Ctrl + End 8.5

Working with Word Document - File Related Operations When WORD is opened you will be presented with a blank page. All what you have to do is start writing. Of course you don't have to worry about overflowing the text line because the program will take care of breaking it up appropriately. After writing your first document you have to save it; if you don't and turn off the computer what you have written will be cancelled and nothing will remain. However the program asks you what you want to do of the document the moment you shut off WORD without having saved it first. If you intend to keep the document you will be prompted

159 to tell the computer where you want to save it. By default it will be saved as a file in a folder called My Documents. Give a name to the document and choose the file's destination: it will be saved with a .DOC extension; this is to tell Windows that this document is a WORD file. From now on the document you have written will be available for you to do whatever you like: you can open it again and modify it, or print it, or send it via e-mail or fax. We can do all file or document related operations from FILE menu. Open a New File: To open a new file, Click on the File menu. Then Click on the new option or you can directly press the Ctrl+N from the keyboard. Select the Blank document from the General Tab from the following dialog box and then press OK button.

Save a Document: For saving a document, Click on the File ► Save option. OR Press Ctrl+S OR Press the Save tool from standard toolbar. If you are saving the file for the first time the Save as window will appear. Choose the appropriate folder from Save in combo box. Write the proper file name in the File name text box.

And then

press the Save button Save and Save As options do the same work of saving a document but the difference between both options is that the Save As command every time gives you the Save As dialog box and you can save the file with different names also. But the Save option will gives you the Save As dialog box when you are saving the document for very first time. Other times it will save the document with the previous name without giving any dialog box.

160

Opening of an Existing Document: To open an existing document, Select the Open option from the File menu OR Press Ctrl+O OR Click on the Open tool from the standard toolbar. Then the open dialog box will appear. Select the appropriate folder from Look in combo box. Select the required file from the file window Or Write the required file's name in the File name window. Click on open button on the right hand side.

Page Setup: Click ► File ► Page Setup Note. If you can not see Page Setup, scroll down to the two little arrows pointing down and click once to see the entire menu. A Page Setup window will appear.

From the Page setup option one can setup the page layout according to the job. You can

adjust the different margins or apply different options from the margin tab where: i) In top margin enter the distance you want between the top of the page and the top of the first line on the page. ii) In Bottom margin enter the distance you want between the bottom of the page and the bottom of the last line on the page. iii) In Left option enters the distance you want between the left edge of the page and the left edge of unindented lines. iv) In Right option enter the distance you want between the right edge of the page and the right end of a line with no right indent. v) In Gutter option enter the amount of extra space you want to add to the margin for binding. Word adds the extra space to the left margin of all pages if you

161 clear the Mirror margins check box, or to the inside margin of all pages if you select the Mirror margins check box. vi) In Header option under From edge frame enter the distance you want from the top edge of the paper to the top edge of the header. If the Header setting is larger than the Top setting, Word prints the body text below the header. vii) In Footer option under From edge frame, enter the distance you want from the bottom edge of the paper to the bottom edge of the footer. If the Footer setting is larger than the Bottom setting, Word stops printing the body text above the footer. viii) Check Mirror margin check box to Adjusts left and right margins so that when you print on both sides of the page the inside margins of facing pages are the same width and the outside margins are the same width. ix) Check the 2 pages per sheet checkbox to Print the second page of a document on the first page. This check box is used when the printed page is folded in half with the two pages on the inside. The outer margins (gutter) of the page will be the same width, and the inner margins will be the same width. x) In the Apply to list box, Click the portion of the document you want to apply the current settings to in the Page Setup dialog box. And the options of this list box are Whole document, This point forward etc can be

changed according to the situation. Sometimes we create documents that do not fit on a standard piece of paper in its normal standing up position. If you are creating something by hand then you can simply turn the page so it is lying on its side. With MS Word you can change the orientation of your page anytime you want while you are creating your document. To change the orientation of your document to landscape (sideways):

From the Paper size tab you can set the length or width of the page. When you click on the Paper size tab the following window will appear.

162 1.

Click on the Paper Size tab 2. Click in the circle besides Landscape 3. Click ► OK Your document should now be in a landscape position. If you wish to place it back in a portrait position, go through the same steps but instead of clicking in the circle besides Landscape, click in the circle besides Portrait.

In the above window you can adjust the following: i) From Paper size list box you can select the predefined Paper sizes.

ii)

From

Width and Height text boxes the custom Paper size can be defined by adjusting the Height and width of the paper.

iii) Select the orientation of the paper from Landscape or Portrait orientation frame.

Print Option: For taking the printout you have to select the print option of the file menu.

To print your MS Word document: i) Click ► File ► Print from the menu bar and a Print window will come up on the screen ii) Click ► OK for it to start printing OR iii) Change some of the automatic settings first, such as which pages or how many copies will be printed iv) Click ► OK to start printing

163 You can set various options before taking the

printout. i) From the Name combo box you can select the printer if more than one printer is installed.

ii) You can select the range of pages i.e. all pages or current page or number of pages you require from Page range frame. iii) From the print what option you can choose which part of a document you want to print i.e. the whole document or comments or anything else. iv) From print option the types of pages can be selected to print i.e. all pages or even pages or odd pages. v) You can choose number of copies from Number of copies option under Copies frame. vi)

From Pages per sheet option under Zoom frame you can Select the number of pages in the document that you want to print on each sheet of paper.

vii)

From scale to paper size option you can select the paper size on which you want to print the document. For example, you can specify that a B4-size document prints on A4-size paper by decreasing the size of the font and the graphics. This feature is similar to the reduce/enlarge feature on a photocopy machine. The collate check box can print the copies of the document in proper binding order.

If you want to check what your document will look like before you print: i) Click ► File ► Print Preview from the menu bar at the top or Click on the Print Preview icon on the standard toolbar

164 ii) Click on Close at the top of the Print Preview page to go back to the regular view Closing MS Word: Be sure to leave yourself enough time to save or print your work before the end of your session. When you are finished, Click ► File ► Exit from the menu bar OR Click on the X in the top right corner of the computer screen 8.6 Some Important Editing Functions When you start up a word-processing program, a window will appear for entering text, with a ruler above to indicate the paper width. You will probably also see lots of little buttons with pictures on them, as seen below. The buttons are shortcuts to certain popular commands. If you hover your mouse over these buttons (without clicking) a little tag will appear telling you what the button does. The cursor (called a 'flashing i-beam') will blink on an off, indicating the text insertion point. Anything typed will appear at the insertion point. By clicking the mouse within the typed text, you can reposition the cursor. Additional text can then be typed, or text can be deleted:

165 Cut, Copy, Paste and Copy Format: To copy words, portions of text, images, or just about whatever you have in your document, you can click on the COPY icon or hit Ctrl+C; this will memorize what you have copied in the clipboard and you can paste it any place you want by clicking on the PASTE icon or hitting the keys Ctrl+V. However, if you wish to delete something in order to copy it somewhere else, use the CUT command (the scissors icon) or hit Ctrl+X. Finally, clicking on COPY FORMAT (the brush icon) you can format text as you wish: to do so, position your pointer on the word whose format you want to copy; click on COPY FORMAT icon; move pointer to word or text you want to modify and highlight it with your mouse, and finally release the mouse button. The text will show the new format (font, character size and colour). Reverse Commands:

To reverse changes while editing, undo command is used. To perform

undo you can Click on the Undo option under Edit menu Or Click on the Undo button on Standard toolbar Or Press Ctrl+Z. If an undo has to perform on last reversed action that

is known as Redo . To perform redo you can Click on the Redo option under Edit Menu Or Click on the Undo button on Standard Toolbar Or Press Ctrl+Y

Find, Replace And Go To Options: Find: Some times while working in document you need to find a particular text. To find a particular text click on the Edit menu, click Find or Ctrl+F. Then you will see the window shown below.

In the Find what box, enter the text that you want to search for. Select the direction of searching from Search list box. Select any other options that you want.

166 Match case: To find the characters that are a word by themselves and are not a part of another word. Find whole words only: To find the characters that are a word by themselves and are not a part of another word.

Use wildcards: To specify the wildcard characters (? Or *) in the Find what text box Sounds like: To find words that sound similar but are spelt differently. For example hair, heir, hear and hare sound similar to here. Find all word forms: To find all grammatical forms of the word. For example, on entering the word eat it also searches ate, eaten and eating words.

If you want find similar word again in the rest of the document click on the Find next button. Replace: If you have to replace a word in the document with another word you can use find and replace command. On the Edit menu, click Replace

or Ctrl+H. Then you will find a window.

The Find what box, enter the text that you want to search for. In the Replace with box, enter the replacement text. Select any other options that you want.

Click Find Next, Replace, or Replace All. To cancel a search in progress, press ESC.

167 Go to:

To go to on a particular location or particular item use Go to option under the Edit menu click on the Go to option under the Edit menu or Ctrl+G. Then the following window will appear on the screen. Select according to what you want to navigate in the document from Go to what combo box. Enter the parameter as page number or name of the bookmark. Click on the previous or next depending upon the direction you want to go.

168 Formatting a Document:

Once the text of a document has been typed, it is required to be formatted.

MS-Word provides Format menu to give a meaningful format of

your document, so that a user can read and understand the document in an easy manner. Following are the options available in the Format menu. Text Formatting:

Format of font i.e. the font type, font size, font style, font Color, character spacing etc. can be changed by going in the option Font of Format menu. Changing of Fonts: The word font means the way of writing characters. To change

the font-using Font dialog box gives you the opportunity to see the result within the dialog box before applying it on

the text. Other benefit of using the font dialog box is that you can apply several effects within the same dialog box like font color, bold or font size etc.

Select the desired text to change the font. Select the Font option from Format menu.

A window will appear on the screen.

Scroll through the Font list and click the font that you want. Select the font color, underline style and Effects of your choice. Select the font style according to the text. Click on Ok button

Paragraphs Formatting: Alignment of the Paragraph:

Alignment refers to the way the right and left edges of a paragraph line up along the right and left margins of your document.

Word gives you four alignment choices - left, centered, right and justified.

169

Working with Columns: You can use columns to create documents such as newsletters or book pages. The column option controls the width and number of columns. Narrower column widths make your text easier to read. Often, creating columns enables you to fit more text on a single page. Columns work particularly well where you have several shorter topics making up a larger document, as with a newspaper. To create a column selects the text which you want to convert in column Click on the column button in the Standard toolbar. A drop-down list appears with four columns across it. Click on the first column in the drop-down list and drag across to highlight desired number of columns.

Center Aligned Left Aligned Justified Right Aligned

170

Zoom In and Out of Text: It is worth mentioning here that the zoom function that is on the standard tool bar of WORD can be used to zoom in and out the text, this way even if you decide for a very small font size you don't have to kill your eyes to read what you are writing. Simply select a higher value on the pull down menu or directly type in a value in the box.

Spelling Checker: Another important feature is the Spelling Checker and vocabulary, whereas the Grammar Checker is highly unreliable yet and therefore not so useful. Normally most modern WORD versions have this feature of English in-built vocabulary that may be an invaluable helping tool for synonyms and antonyms. To activate this function, position the pointer on the word you want to check and hit F7 key, or alternatively right-click on your mouse and work through the pop-up menu.

Microsoft Word automatically underlines any words that it does not recognize in red, assuming that they are not spelled right. Right-click the red underlined word and click on either one of the corrections suggested or Ignore, to make no changes or add, to make no changes but to add the word to Microsoft Word's list of words that it thinks are spelled correctly. This is especially useful for names of people or places that you type often, so that Word will not think it is wrong the next time it sees it. Green underlines mean that Microsoft Word thinks there is something wrong with the way the sentence is written. Right-click the underlined words and either click on the corrections suggested or click Ignore.

Inserting Images From Clip Gallery: To insert an image from Clip Gallery you have to move the cursor where you want to insert the image. Select Clip art option from Insert ► Picture menu. Click the picture category from which you want to insert the picture. If you know what type of image you're looking for, type a descriptive word or two in the Search for Clips text box, and press Enter. Word displays the images that are most closely match your keywords. When you find out the image that you want to use, click it. Clicking on the

171 image will display a small toolbar. From that toolbar, click the Insert button as given in the figure given below. This will insert a clipart in your document.

8.7 Mail Merge Mail merge is a pretty simple concept. Mail merge allows standard letters to be 'personalized' by merging it with data from a database. The result is a batch of letters which contain virtually the same information with the exception of names, addresses and any other information stored in a database. You take a document (such as a form letter) and a set of information (such as the names and addresses of all your clients) and merge them. When you merge the form letter and the list of names, you can generate a mass mailing. Each copy of the document you print will be personalized with a different name and address. Typing the letter once, combining it with your client list, and printing it can therefore accomplish a mailing to hundreds of clients. Before you start learning mail merge process, you need to understand the two documents that make up a mail merge: 1. Main Document: In a mail-merge operation, the document containing the text and graphics stays the same for each version of the merged document. For example, the return address and the body of a form letter. 2. Data Source: A file that contains the text and graphics that varies with each version of a mail-merge document. For example, a list of names and addresses for a form letter you want to send to a list of client's or in any report in which you want to use information from a database.

Creating the Main Document The main document can be a form letter, an invoice, an invitation or any type of document. The idea here is that the document will go to many recipients, and some unique information must be included to personalize it. If you have an existing letter that you want to use for your form letter, open it, and modify it according to your need or if you want to start a letter from scratch, start a new blank document. Save the document. Choose tools, Mail Merge to display the Mail Merge Helper Dialog box. Click the Create button. Select Form Letter from the drop-down list. A dialog box will appears with two choices: to create a form letter within the active window or in new window. Click the Active window Button. Click on edit button if you haven't written the letter body.

>>
 TITLE<< >>FORENAME<< >>SURNAME<< >>ADDRESS 1<< >>ADDRESS 2<< >>POST CODE<< Dear >>TITLE<< >>SURNAME<<,
 You have been selected among thousands of people to take part in our prize draw to win Rs. 10,000. A cheque to
 >>
 TITLE<< >>SURNAME<< could be on its way to >>ADDRESS 1<< >>
 if you fill out the reply slip below and post within 7 days. In return we will enter you in the prize draw and send you our current catalogue. Yours sincerely, Raj Shekhar Promotions Manager A standard letter is written and mail merge markers are placed in it to show where the data from the database will be inserted into the letter to personalize it.

173 Building Data Source In this second phase, you tell Word which document you want to use as your data source. You can either create a new one or open an existing one. A data source is a file that contains the data that varies in each copy of a merged document. For example, a data source can include the name and address of each recipient of a form letter. Create a New Word Data Source Make sure you've already set up the main document. If you don't see the Mail Merge Helper dialog box, click in the main document, and then click Mail Merge on the Tools menu. Under Data source, click Get Data, and then click Create Data Source. In the Field names in header row box, specify the data fields (A category of information in a data source. A data field corresponds to one column of information in the data source. The name of each data field is listed in the first row (header row) of the data source. "PostalCode" and "LastName" are examples of data field names.) that you want to include in the data source. To delete a data field, click the field name in the Field names in header row box, and then click Remove Field Name. To add a data field, type a new field name in the Field name box, and then click Add Field Name. To change the order of the data fields, click a field name in the Field names in header row box, and then click one of the arrow buttons. When you finish specifying the data fields, click OK. Locate the folder that you want to save the data source in, type a file name, and then click Save.

174 Click Edit Data Source. In the Data Form dialog box, fill in the information for each data record (A complete set of related information in a data source. A data record corresponds to one row of information in the data source. All information about one client in a client mailing list is an example of a data record. To fill in a record, type information for a data field, and then press ENTER to move to the next field. If you don't want to include information for a particular field, press ENTER to skip the field. Don't type spaces in the box. To start a new data record, click Add New. To return to the main document, click Mail Merge Main Document on the Database toolbar. Now the main document displays out with the new toolbar of Mail Merge Inserting the Merge Fields It's time to enter merge fields in the form letter you typed. The merge fields in the letter will correspond to the fields in your database. Place your cursor halfway between the date and salutation. Choose the Insert Merge field's button on the Mail Merge toolbar.

Mail Merge Main

175 Word inserts the merge field surrounded by chevron brackets. Insert the remaining merge fields in the address block, pressing Enter and adding spaces and commas where necessary. If you have a salutation field, add it after Dear, and follow it with a colon or comma. Save the main document. Merging Document and DATA Merging to a Printer or Document Choose the Mail Merge Helper button. Now the look of Mail Merge Helper dialog box changed and some new buttons get activated. Choose Merge button. The merge dialog box appears. Open the Merge to drop-down list. Select the option from Printer, New Document, Electronic mail for the destination of the mail merge. Select the New Document option and click on the Merge button. Then you will get a new document named as Form Letters 1 with a copy of letter with different names and address on different pages.

176 8.8 Summary The word processor is the modern version of paper, pen, typewriter, eraser, dictionary and thesaurus... all in one. It is used to create all kinds of documents, from simple notes and memos, to brochures, resumes, and long reports. The key advantage of a word processor is its ability to make changes easily, such as correcting spelling, adding, deleting, and relocating text. Once created, the document can be printed quickly and accurately and saved for later modifications. In a word-processing program, a window appears for entering text, with a ruler above to indicate the paper width and lots of little buttons with pictures on them. The buttons are shortcuts to certain popular commands. If mouse is hovered over these buttons (without clicking) a little tag will appear telling you what the button does. Word processing software has various facilities like Mail merge. Mail merge allows standard letters to be 'personalized' by merging it with data from a database. The result is a batch of letters which contain virtually the same information with the exception of names, addresses and any other information stored in a database.

177 At last, a word processing program is not a typewriter, although it may seem so. In order to avoid problems with printing or formatting your document, you should keep in mind the following: ¾ Do not use the Enter key to change the line, except at the end of a paragraph. ¾ Do not use the Tab key or Spacebar to indent your text, since the lines will be hard to set right if you need to rewrite something. ¾ Do not hyphenate your document manually. ¾ Do not enter page numbers manually. ¾ Do not create footnotes manually. ¾ Do not create a table of contents manually. ¾ Do not create tables manually. 8.9 Self-Test questions 1. Describe the standard features of a Word Processor. 2. Explain the structure of Microsoft Word Window. 3. What do you understand by document formatting in MS-Word? 4. List some of the editing functions of Word Processor. 5. Describe briefly the use of graphics in a Word Processor Software. 6. What is a spelling checker facility? 2. List the basic steps for setting up a mail-merge operation. 8.10 Suggested Readings Marquis, Annette and Courter, Gini, "Microsoft Office 2000", 1 st Ed., BPB Publications. Norton, Peter, "Introduction to Computers", 2 nd Ed., Tata McGraw Hill.

178

UNIT 9 -

SPREADSHEET SOFTWARE: AN OVERVIEW Objectives: The motive of this lesson is to present an overview of Spreadsheet Software and within that framework the key concepts, applications and functioning of spreadsheet software are discussed. Structure: 9.1 Introduction 9.2 Concept and Definitions of Spreadsheet Software 9.3 Structure of Worksheet 9.4 Creation of Spreadsheet 9.5 Creating Range, Formulas and Functions 9.6 Graphics on Spreadsheet 9.7 Some future trends for spreadsheets 9.8 Summary 9.9 Self-Test questions 9.10 Suggested Readings 9.1 Introduction Though most business computations can be handled by general accounting software, more complex calculations often are needed in order to help businesses make wise investment decisions. A computer aids the business person in performing these complex mathematical tasks. Perhaps the most common use of the computer is in financial analysis that involves the electronic spreadsheet use for designing solutions for complex calculations. A spreadsheet is a grid that organizes data into columns and rows. Spreadsheets make it easy to display information, and one can insert formulas to work with the data. We use spreadsheet programs to learn about different kinds of things, and to make decisions. Spreadsheets are based on different varieties of subjects. Although electronic spreadsheets are useful in many areas of business, nowhere they are more helpful than in financial analysis, or forecasting, which determines profit margins, sales, and long term strategies. The reason for the great impact of the electronic

179 spreadsheet in this area is its ability to answer "What if?" type queries on available data, as with the help of formula and function facility one can intelligently setup complex relationships between individual items in a table. A spreadsheet program uses the memory capability of computer to solve mathematically oriented problems. With a spreadsheet program, columns of numbers can be setup to keep track of money or objects. These items are stored in a place known as cells. The user may setup relationships and see the affect of varying independent quantities. With the high speed computing power of the digital computer, complex problems can be modeled on the computer and various options can be tried. For example, there is a particular icon that has a formula to sum up numbers that are given. This icon is called auto sum. Information can also be sorted and filtered. This chapter will present an overview of the spreadsheet software and highlight the use of spreadsheets through its applications. Concept and Definitions of Spreadsheet Software Spreadsheet

is a very powerful and easy to use software package, which is being commonly used these days. It is basically an end user application package. It works very well with numbers and their complex calculations. Spreadsheet helps to prepare data in an organized, orderly and meaningful fashion. Spreadsheet finds its major contribution in creating reports, using formulas and performing calculations. It is best suited for scientific and statistical analysis. Spreadsheet can also be used to prepare Profit and Loss accounts, Balance sheet and other Tax statements. It works well for both simple and complicated numbers. It can be used for preparing analytical reports including statistical analysis, forecasting and regression analysis. Good looking and attractive charts can be created which depict data in clearer and meaningful fashion. Spreadsheet can also used to create relationships between different types of data. Spreadsheet can do all the work of a full-fledged word-processor but it lacks the advanced features of desktop publishing. It also supports the high level features of object linking and embedding which means data from word processor

can be safely and easily put and linked with data in spreadsheet and vice versa is also true. Spreadsheet History

180 TM 1961 - Professor Richard Mattessich was the first to develop a computerized spreadsheet. TM 1978 - Robert Frankston and Dan Bricklin invented VisiCalc, the first spreadsheet. It came out with the Apple II computer. VisiCalc did very well in its first year because it could run. On personal computers, could perform simple math formulas, and gave immediate results. TM 1980 - DIF formats were invented. Data was more portable and could be shared with other programs. TM 1980-83 - New versions of DIF, SuperCalc, and some other programs came out to the public. TM 1983 - Lotus 123 was introduced. It allowed people to chart information and identify cells. For example cell A1. TM 1985 - Lotus 123 number 2. TM 1987 - New spreadsheet programs such as Excel and Corel Quattro Pro were introduced. This allowed people to add graphics. They are different because they include graphic capabilities. TM 2001 - Spreadsheet programs in use today are Excel, Appleworks, Filemaker, and Corel Quattro Pro. We can define spreadsheet in the following manner: A spreadsheet is a software tool for entering, calculating, manipulating, and analyzing sets of numbers. An electronic spreadsheet is a relational grid of columns and rows used to store and manipulate numeric information by means of a computer. The grid appears on the display screen, and data is stored in the memory of computer. We have seen from historical evolution many of the spreadsheet software, from learning point of view we shall consider only Microsoft Excel.

Structure of Worksheet: Start with Microsoft Excel: To start Excel, follow the following steps: 1. Click on the Start button at windows taskbar. 2. Select Program option. 181 3. Select Microsoft Excel option. 4.

Selecting this option will open up Microsoft Excel and the main screen will appear. Let us discuss the structure of Worksheet and Functions of Various Components of Microsoft Excel from the main screen. Functions of various components of Microsoft Excel are given below: 1.

Title Bar: Title bar shows the name of the document and situated in the top of the window application. 2.

Menu Bar: Menu bar contains the various commands under the various topics to perform some special tasks. Menu bar is located under the title bar 3. Standard Toolbar: Toolbar is

nothing more than the shortcut of the Menu options to execute or perform the menu options in the easiest and faster way. Standard toolbar contains the commands

for the Excel's most frequently used commands. 4. Formatting Toolbar: Formatting toolbar contains the various commands related to the formatting of the document. 5.

Status Bar: It displays the positioning of the cursor, displays the status of some important keys of keyboard, shows the messages for the toolbar

button when a

mouse points to it, display messages for menu option when a menu option is selected or pointed out by a user. 182 6.

Formula Bar: Formula bar shows you the formula instead of the result whenever you click on a cell in which you have a formula. 7. Rows:

Rows are combination of all the horizontal cells joined together in single horizontal line.

An Excel workbook has maximum 65,536 rows

in a worksheet 8. Columns: Columns are combination of all the vertical cells joined together in single vertical line.

An Excel workbook has maximum 256 columns in a worksheet 9.

Cell: A Cell is an intersection point of row and column. It is used to store data. There are $65536 \times 256 = 16777216$ cells in a single worksheet. 10.

Name Box: This box shows the name or the address of the currently active cell. 11.

Office Assistant: Office assistant provides you

the online help, real-time tips while working. 12. Scroll Bars:

There are two types of scroll bars i. Horizontal Scrollbar ii. Vertical Scrollbar

Working of both the scroll bar is to scroll the Worksheet in the either directions 13.

Worksheets: As name suggests it is the working area in which you can work.

Each worksheet of a single workbook has its own identity

they are

separate from other worksheets. The maximum limit of the worksheet in a workbook is 255.

Creation of Spreadsheet: Open a New Workbook To open a new workbook, follow the steps: 1. Click on the File Menu. 2. Select Option New. 3.

Click the OK Button. Now you will get a fresh new workbook to work on. Save a Workbook

For saving a document, following are the steps: Click Save option

on the File OR Press Ctrl+S OR Press the Save Button from standard toolbar. If you are saving the file for the first time, the Save As... window

183 will appear. Choose the appropriate folder from Save in combo box. Write the proper file name in the File name text box.

And then press the Save button Opening of an Existing Workbook To open an existing workbook, the following steps are required: 1.

Select the Open option from the File menu OR Press Ctrl+O OR Click on the Open tool from the standard toolbar. Then the open dialog box will appear. 2. Select the appropriate folder from Look in combo box. 3. Select the required file from the file window OR write the required file's name in the File name window 4. Click on open button on the right hand side

OR Press Enter Closing of Workbook To close an already opened workbook just choose the Close option from file menu but keep it in your mind that only the current window or workbook will close because Microsoft Excel works in MDI

(Multi document interface) environment unlike notepad which works in SDI(Single document interface).

Page Setup From the Page setup option, one can setup the page layout according to his work. For using the Page setup option, you have to perform the following steps: 1. Click on the Page setup option from the file menu. Then

a window will appear. 2. Select the page Orientation from Portrait or Landscape. 3. Setting the Adjust to setting will

Reduces or enlarges the printed worksheet. 4. Select the Adjust to check box, and then enter a percentage number in the % normal size box. You can reduce the worksheet to 10 percent of normal size or enlarge it to 400 percent of normal

size.

184 Give the value to 'Fit to'. Setting to 'Fit To' reduces the worksheet or selection when you print so that it fits on the specified number of pages. Select the 'Fit to' check box, enter a number in the page(s) wide by box, and enter a number in the tall box. To fill the paper width and use as many pages as necessary, type 1 in the pages(s) wide by box and leave the tall box blank. Select the Paper size

Print Option For taking the printout you have to select the print option of the file menu. After selecting the print option from file menu the window given below will appear.

185 You can set various options before taking printout. 1. From the Name combo box, you can select the printer if more than one printer is installed. 2.

You can select the range of pages i.e. all pages or current page or number of pages you require from Print range frame.

3. From the print what option you can choose that which part of a worksheet you want to print i.e. some selection or Entire workbook or active sheet only. 4. You can choose number of copies from Number of copies option under Copies frame.

Creating Range, Formulas and Functions: Creating Range: Range is a specified area from one location (cell address) to another location (cell address) on what some sort of operation has to be performed. Here as shown in the window the specified area is from B3 to I18 which is selected with the help of mouse. Perform the following steps to select a range of cells: 1. Point your mouse to cell address B3. 2. By pressing the left button of the mouse drag upto cell address I18. 3.

Release the button. The range is created for performing any operation on the worksheet.

186 Creating Formulas: After entering the data into the worksheet, calculations are performed with that data. This is done using formulas, the true power of a spreadsheet program. For creating the formula, syntax is required which describe the sequence of character used in a formula. The syntax of a formula begins with an equal sign (=) in Excel and is followed by a combination of values, operators and cell references. Operators: Operators specify the type of operation that has to be performed on the parts of a formula. Basically we use three types of operators. 1. Arithmetic Operators: To perform mathematical operation on numeric values and thereby produces numeric results. 2. Comparison Operators: To compare the values or labels in two cells. The result of this formula is either TRUE or FALSE. 3. Text Operators: To join one or more text values into a single combined text value. The ampersand (&) (which is the only text operator) is used to join one or more labels into a single combined label. For example, =D2&" "&F3 This formula combines the labels in cells D2 and F3. Various Arithmetic and Comparison operators are shown in the table. Arithmetic Operators Comparison Operators Operators Meaning Operators Meaning + Addition = Equals - Subtraction or Negation >< Does not equal / Division < Greater than * Multiplication <= Greater than or equal to % Percent, e.g. 35% > Less than ^ Exponentiation (e.g. 4^2=16) >= Less than or equal to

187 When arithmetic operators are used in formulas, spreadsheets calculate the results using the rules of precedence followed in mathematics. The order is as follows: 1. Exponentiation (^) 2. Negation (-) 3. Multiplication and Division (*,/) 4. Addition and Subtraction (+, -) Otherwise, operators are evaluated from left to right. In order to change the order of precedence to suit your needs, you add parentheses around any part of the formula that you want to be calculated first. Creating Functions:

Spreadsheets come with many built formulas, called functions that perform specialized calculations automatically. We can include these functions in our own formulas. Some functions are quite simple, such as the COUNT function (to count the number of values in a range of cells). Many functions, however are very complex. To insert a function

you have to follow the following steps. 1. Click on the cell in which you want to insert the function. 2.

Select the option Function from insert menu. 3. Above dialog box will appear. 4. Select the desired function and click on OK button

188 Using Various Functions in MS-Excel: The leading spreadsheets come with hundreds of these functions. Here we are discussing some of the most commonly used ones with the help of examples: 1. Date and Time Functions i. DATE: Represents the date number that represents a particular date. Syntax: DATE(year, month, day) Argument Type: Number, Number, Number Return Type: Number Example: =date(2005, 04, 17) MS-EXCEL returns 4/17/05 ii. DATEVALUE: Returns the serial number of the date represented by date_text. Syntax: DATEVALUE(date_text) Argument Type: Text like date Return Type: Number Example: =DATEVALUE("2005/04/17") MS-EXCEL returns 38459 iii. DAY: Returns the day of a date, which is represented by a serial number. The day is given as an integer ranging from 1 to 31. Syntax: DAY(serial_number) Argument Type: Number(Date number) Return Type: Number Example: =DAY(38459) MS-EXCEL returns 17 iv. MONTH: Returns the month of a date, which is represented by a serial number. The month is given as an integer ranging from 1 to 12. Syntax: MONTH(serial_number) Argument Type: Number(Date number) Return Type: Integer Example: =MONTH(38459) MS-EXCEL returns 4 v. NOW: Returns the serial number of the current date and time i.e. this function returns sum of current date number and current time number. Syntax: NOW() 189 Argument Type: None Return Type: Number Example: =NOW() MS-EXCEL returns 4/17/2005 21:58 vi. TIME: Returns the decimal number for a particular time. Syntax: TIME(hour, minute, second) Argument Type: Number, Number, Number Return Type: Number (the time number) Example: =TIME(21, 58, 12) MS-EXCEL returns 9:58 PM vii. TODAY: Returns the serial number of current date i.e. the date number of current date. Syntax: TODAY() Argument Type: None Return Type: Number Example: =TODAY() MS-EXCEL returns 4/17/2005 2. Financial Functions i. FV: Returns the future value of an investment based on periodic, constant payments and a constant interest rate. Syntax: FV(rate, nper, pmt, pv, type) Argument Type: Number, Number, Number, Number, Return Type: Number Example: =FV(0.005, 18, -500, -2000, 0) MS-EXCEL returns \$11,580.75 ii. PMT: Calculates the payment for a loan based on constant payments and a constant interest rate. Syntax: PMT(rate, nper, pv, fv, type) Argument Type: Number, Number, Number, Number, Number Return Type: Number Example: =PMT(8%/12,10,10000,0,1) MS-EXCEL returns (\$1,030.16) iii. PV: Returns the present value of an investment. The present value is the total amount that a series of future payments is worth now. 190 Syntax: PV(rate, nper, pmt, fv, type) Argument Type: Number all Return Type: Number Example: =PV(0.08/12,240,500, ,0) MS-EXCEL returns (\$59,777.15) 3. Logical Functions Examples based on the following table: R. No. Name Marks Grade 2001 Anil 96 A+ 2002 Sanjay 43 F 2003 Jagdeep 66 A 2004 Rajneesh 55 B 2005 Renu 80 A+ i. AND: The logical functions are used to see whether a condition is true or false or to check for multiple conditions. Syntax: AND(Logical1, Logical2, ...) Argument Type: Logical all Return Type: Logical Example: =AND(c2:c6) MS-EXCEL returns TRUE ii. OR: Returns TRUE if any argument is TRUE; returns FALSE if all arguments are FALSE. Syntax: OR(Logical1, Logical2, ...) Argument Type: Logical all Return Type: Logical Example: =OR(c2:c6) MS-EXCEL returns TRUE Type: =OR(b2:b6) MS-EXCEL returns #VALUE iii. NOT: Reverses the value of its expression that can be evaluated to TRUE or FALSE. Syntax: NOT(logical) Argument Type: Logical 191 Return Type: Logical Example: =NOT(FALSE) MS-EXCEL returns TRUE Type: =NOT(1+4=5) MS-EXCEL returns FALSE 4. Math Functions i. ABS: Returns the absolute value of a number. Syntax: ABS(Number) Argument Type: Number Return Type: Number Example: =ABS(5) MS-EXCEL returns 5 Type: =ABS(-5) MS-EXCEL returns 5 ii. EXP: Returns e raised to the power of a number e.g. e^x . The constant $e=2.71828182845904$, the base of the natural logarithm. Syntax: EXP(number) Argument Type: Number Return Type: Number Example: =EXP(1) MS-EXCEL returns 2.718281828 Example: =EXP(2) MS-EXCEL returns 7.389056099 iii. INT: Rounds a number down to a nearest integer. Syntax: INT(number) Argument Type: Number Return Type: Number Example: =INT(8.9) MS-EXCEL returns 8 Example: =INT(-8.9) MS-EXCEL returns -9 iv. LOG: Returns the logarithm of a number to the base we specify. Syntax: LOG(Number, base) Argument Type: Number, Number 192 Return Type: Number Example: =LOG(10) MS-EXCEL returns 1 Example: =LOG(8,2) MS-EXCEL returns 3 v. MOD: Returns the remainder after number is divided by divisor. The result has the same sign as the divisor. Syntax: MOD(Number, divisor) Argument Type: Number, Number Return Type: Number Example: =MOD(3,2) MS-EXCEL returns 1 Example: =MOD(3, -2) MS-EXCEL returns -1 vi. ROUND: Returns a number to a specified number of digits Syntax: ROUND(number, num_digits) Argument Type: Number Return Type: Number Example: =ROUND(2.15,1) MS-EXCEL returns 2.2 Example: =ROUND(2.149,1) MS-EXCEL returns 2.1 vii. SQRT: Returns a positive square root. Syntax: SQRT(number) Argument Type: Number Return Type: Number Example: =SQRT(16) MS-EXCEL returns 4 Example: =SQRT(-16) MS-EXCEL returns #NUM viii. SUM: Adds all the numbers in a range of cells. Syntax: SUM(number1, number2, ...) Argument Type: Number Return Type: Number

193 Example: =SUM(3,2) MS-EXCEL returns 5 Example: =SUM(a2, b2, 2) MS-EXCEL returns 2 ix. TRUNC: Truncates a number to an integer by removing the fractional part of the number. Syntax: TRUNC(number, num_digits) Argument Type: All number Return Type: Number Example: =TRUNC(8.9) MS-EXCEL returns 8 Example: =TRUNC(-8.9) MS-EXCEL returns -8 5. Statistical Functions Examples based on the following table: R. No. Name Marks Grade 2001 Anil 96 A+ 2002 Sanjay 43 F 2003 Jagdeep 66 A 2004 Rajneesh 55 B 2005 Renu 80 A+ i. AVERAGE: Returns the average of the arguments. Syntax: AVERAGE(number1, number2, number3,...) Argument Type: All numbers Return Type: Number Example: =AVERAGE(c2:c6) MS-EXCEL returns 68 ii. COUNT: Counts the number of cells that contain numbers and numbers within the list of arguments. Syntax: COUNT(value1, value2,...) Argument Type: Any type Return Type: Number Example: =COUNT(c2:c6)

194 MS-EXCEL returns 5 iii. MAX: Returns the largest value in a set of values. Syntax: MAX(number1, number2,...) Argument Type: All numbers Return Type: Number Example: =MAX(c2:c6) MS-EXCEL returns 96 iv. MIN: Returns the smallest value in a set of values. Syntax: MIN(number1, number2,...) Argument Type: All numbers Return Type: Number Example: =MIN(c2:c6) MS-EXCEL returns 43 6. Text Functions i. CHAR: Returns the character specified by a number. The number is assumed to be the ASCII value and its equivalent is returned. Syntax: CHAR(number) Argument Type: Number Return Type: Character Example: =CHAR(65) MS-EXCEL returns A ii. CONCATENATE: Joins several text strings into one string. Syntax: CONCATENATE(text1, text2,...) Argument Type: All Text Return Type: Text Example: CONCATENATE("Ram", "Avtar") MS-EXCEL returns RamAvtar iii. LEFT: Returns the first character or the characters in a text string, based on the number of characters we specify. Syntax: LEFT(text, num_chars) Argument Type: Text, Number Return Type: Text Example: =LEFT("SALES PRICE", 4) MS-EXCEL returns SALE

195 iv. LEN: Returns the number of characters in a text string i.e. Length of a string. Syntax: LEN(text) Argument Type: Text Return Type: Number Example: =LEN("SALES PRICE") MS-EXCEL returns 11 v. LOWER: Converts all uppercase letters in a text string to lowercase. Syntax: LOWER(text) Argument Type: Text Return Type: Text Example: =LOWER("SALES PRICE") MS-EXCEL returns sales price vi. RIGHT: Returns the last character or the characters in a text string, based on the number of characters we specify. Syntax: RIGHT(text, num_chars) Argument Type: Text, Number Return Type: Text Example: =RIGHT("SALES PRICE", 5) MS-EXCEL returns PRICE vii. TRIM: Removes all spaces from text except for single spaces between the words.. Syntax: TRIM(text) Argument Type: Text Return Type: Text Example: =TRIM("SALES PRICE ") MS-EXCEL returns SALES PRICE viii. UPPER: Converts all lowercase letters in a text string to uppercase. Syntax: UPPER(text) Argument Type: Text Return Type: Text Example: =UPPER("sales price") MS-EXCEL returns SALES PRICE

196 Data Management Data menu of the Excel provides various commands, which you can apply to your data. Sorting of Data To sort

the data you have to perform the following steps. 1. Select the columns which you want to sort. 2. Choose Sort option from Data menu. You will find out the following dialog box on the screen. 3. Choose the Column on which you want to do the sorting. You can select more than one column. 4. Select the Ascending or descending option. 5. Select the Header row if there is any header row in the table. 6. Click on the OK button.

197 Filtering List Sometimes the user wants to locate some information from an extensive list. To find the information quickly, Microsoft Excel provides Filter list option. This option enables the user to display the information regarding the selected options. Microsoft Excel provides two commands for filtering lists: AutoFilter, which includes filter by selection, for simple criteria Advanced Filter, for more complex criteria Using Auto Filter You can apply filters to only one list on a worksheet at a time. 1. Click a cell in the list you want to filter. 2. On the Data menu, point to Filter, and then click AutoFilter. 3. To display only the rows that contain a specific value, click the arrow in the column that contains the data you want to display. 4. Click the value. 5. To apply an additional condition based on a value in another column, repeat steps 3 and 4 in the other column. Filter a List by using Advanced Criteria Your worksheet should have at least three blank rows above the list that can be used as a criteria range. The list must have column labels. 1. Select the column labels from the list for the columns that contain the values you want to filter, and click Copy.

198 2. Select the first blank row of the criteria range, and click Paste . 3. In the rows below the criteria labels, type the criteria you want to match. Make sure there is at least one blank row between the criteria values and the list. 4. Click a cell in the list. 5. On the Data menu, point to Filter, and then click Advanced Filter. 6. To filter the list by hiding rows that don't match your criteria, click Filter the list, in-place. To filter the list by copying rows that match your criteria to another area of the worksheet, click Copy to another location, click in the Copy to box, and then click the upper-left corner of the area where you want to paste the rows. 7. In the Criteria range box, enter the reference for the criteria range, including the criteria labels. To move the Advanced Filter dialog box out of the way temporarily while you select the criteria range, click Collapse Dialog. About importing data By importing data, you don't have to retype the data you want to analyze in Microsoft Excel. You can also update your Excel reports and summaries automatically from the original source database whenever the database is updated with new information. Criteria

199 Importing data: You can import data to Excel from most data sources by pointing to Import External Data on the Data menu, clicking Import Data, and then choosing the data you want to import in the Select Data Source dialog box. The Data Connection Wizard, available when you click New Source in the Select Data Source dialog box, makes it possible to import data from external data connections not available from the Select Data Source dialog box. Graphics on Spreadsheet: Inserting Images, Auto shapes and Word Art

Inserting Images from Clip Gallery To insert an image from Clip Gallery you have to

follow these steps: 1. Move the cursor on the cell where you want to

insert the image. 2. Select Clip art option from Insert->Picture

menu. 3. Click the picture category from which you want to insert the picture. 4. If you know what type of image you are looking for, type a descriptive word or two in the Search for Clips text box, and press Enter. Excel displays the images that most closely match your keywords. 5. When you find out the image that you want to use, click it. 6. Clicking on image will display a small toolbar, from that toolbar click the insert button. This will insert a clipart in your worksheet. WordArt WordArt lets you add flair to your text itself. It provides a different look to your text as well worksheet. It is perfect for creating flashy headings and titles. To insert a word art in your worksheet do the following steps. 1. Click the place on the worksheet where you want the WordArt image to be placed. 2. Choose Insert, Picture, WordArt or click on icon. This will open the WordArt Gallery dialog box. 3. Choose the desired format and press Ok button. 4. Then the figure given below will appear. 5. Write the desired text instead of Write your Text Here. 6. Select the desired Font and Font size.

200 7. Press Ok button. Insert Objects Excel provides you the facility to link or embed the objects of other software's in your worksheet and work on the linked or embedded objects as you are working on that software itself. To embed an object you have to follow these steps. 1. Click in the document where you want to place the embedded object. 2. On the Insert menu, click Object, and then click the Create New tab. 3. In the Object type box, click the type of object you want to create. 4. To display the embedded object as an icon — for example, if others are going to view the worksheet online — select the Display as icon check box. Note: Only programs that are installed on your computer and that support linked and embedded objects appear in the Object type box. When you click Microsoft Word Document in the Object type box, an entire document is inserted into your worksheet. The document displays only one page at a time. To display a

different page,

double- click

the Microsoft Word object, and then click a page down button. Insert Chart Pictures are more understandable than words. Charts are visually appealing and make it easy for users to see comparisons, patterns, and trends in data. For instance, rather than having to analyze several columns of worksheet numbers, you can see at a glance whether sales are falling or rising over quarterly periods, or how the actual sales compare to the projected sales. You can create a chart on its own sheet or as an embedded object on a worksheet. You can also publish a chart on a Web page. To create a chart, you must first enter the data for the chart on the worksheet. Then select that data and use the Chart Wizard to step through the process of choosing the chart type and the various chart options. Using chart wizard One of the easiest ways to generate a chart from data you enter in a worksheet is by using Chart wizard. Chart wizard is a simple process of making selections to produce a finished chart.

201 1. Select the range of data on which you want to make a chart. 2. Select the option chart from insert menu following dialog box will appear. 3. Select the type of chart you want. 4. Click on the Press and Hold to view Sample button to preview your chart. 5. If the given charts are not of your choice then click on the Custom Type tab.

202 6. Select the graph type and Click on the Next button. 7. If you have forgotten to select the data range or you want to change the data range then you can select the data range from here by clicking the button next to Data range combo box. 8. Click on the Series tab. Following dialog box will appear. 9. You can add or remove the series (Columns) from the series list box by clicking add or remove button. 10. Click on the Next button the option dialog box will appear on the screen. 11. Fill up the entries as Heading of the chart, (x) axis category heading, (Z) axis values heading. 12. Click on the Legend tab

203 13. Legend dialog box will be displayed on the screen 14. Check mark the Show legend check box to show the legends 15. Select the Placement position. 16. Click on the Data Labels Tab. 17. Choose the type of labels you want to see in your graph. 18. Click on the Next button.

204 19. Select the destination of the Chart on new sheet or in the current sheet. 20. Select New Sheet and click on the Finish button. Some future trends for spreadsheets To a large extent, the future trends for the software industry are now available as part of "Office Suites" that package several programs like spreadsheet, word processor, drawing or presentation graphics and a database management program together and also offer a common interface. Group computing is another trend. As more companies create networks for their computers to share information, there is need to allow more than one person at a time to work on a file. Finally, it is always desired to make it easier for use of the software. For the future, one can expect spreadsheets to offer more special features for the advanced user while simplifying frequently used tasks for beginning and advanced users alike. Summary A spreadsheet is a grid that organizes data into columns and rows. Spreadsheets make it easy to display information, and people can insert formulas to work with the data. People use spreadsheet programs to learn about different kinds of things, and to make decisions. It is basically an end user application package. It works very well with numbers and their complex calculations. Spreadsheet helps to prepare data in an organized, orderly and meaningful fashion. Spreadsheet finds its major contribution in

205 creating reports, using formulas and performing calculations. It is best suited for scientific and statistical analysis. Spreadsheet can also be used to prepare Profit and Loss accounts, Balance sheet and other Tax statements.

After entering the data into the worksheet, calculations are performed with that data. This is done using formulas and functions, the true power of a spreadsheet program. Spreadsheets come with many built formulas, called functions that perform specialized calculations automatically. We can include these functions in our own formulas. Some functions are quite simple however, many functions are very complex. Spreadsheets can also be applicable in presenting the graphics and managing the database. For the future, one can expect spreadsheets to offer more special features for the advanced user while simplifying frequently used tasks for beginners and advanced users alike. Self-test Questions 1. What are the uses of electronic spreadsheet? 2. Describe the structure of MS-EXCEL spreadsheet. 3. Write a short note on inserting a chart. 4. Differentiate between a formula and a function. 5. Write the syntaxes of logical functions and explain giving examples. 6. What is the use of giving name to cells? 7. How data is imported in MS-EXCEL? Suggested Readings Marquis, Annette and Courter, Gini, "Microsoft Office 2000", 1 st Ed., BPB Publications. Norton, Peter, "Introduction to Computers", 2 nd Ed., Tata McGraw Hill.

206

UNIT 10 -

DATA FILES: TYPES AND ORGANIZATION Objectives: The motive of this lesson is to present an introduction of Data Files and within that framework the key file concepts, types and organizations of Data Files are discussed. Structure: 10.1 Introduction to Files 10.2 Data File Types 10.3 Master and Transaction Files 10.4 File Organization 10.5 Serial Files 10.6 Sequential and Indexed Sequential Files 10.7 Direct or Relative Files 10.8 Summary 10.9 Self-Test questions 10.10 Suggested Readings 10.1 Introduction to Files Traditionally the term file has been used to refer to the folder that holds related material in ordered storage for quick reference. A file may be used to describe the object as well as its contents. A file is a block of information stored under a single name. For example, all the text of a term paper could be stored under the name "final_paper." A file may also be referred to as an arrangement of its contents according to user's expected requirements for future reference. For example, if we have a file of date of joining of employees in an organization, it would be better to arrange them by date of joining. We can suitably name these files such as Employees, Bills etc. Almost all information stored in a computer must be in a file. A file is named and is referred to by its name. It has certain other properties such as its type, the time of its creation, the name of its creator, its length, and so on.

207 The information in a file is defined by its creator. Many different types of information may be stored in a file: source programs, object programs, numeric data, text, payroll records, and so on. A file has a certain defined structure according to its use. A text file is a sequence of characters organized into lines (and possibly pages); a source file is a sequence of subroutines and functions, each of which is further organized as declarations followed by executable statement; an object file is a sequence of words organized into loader record blocks. All files are identified by their extension, the ".xxx" that follows any file name. For example, Microsoft Word documents end with the file extension ".doc". Microsoft Word creates word files; FrontPage creates HTML files. Listed below are some common file extension and the programs that create them. .html or .htm- Hypertext Markup Language, the code for most web pages .pdf- Portable Document Format. Allows for universal printing without format changes .exe- An executable program file .jpg- JPEG (Joint Photographic Experts Group) A graphic file .gif- Graphics Interchange Format; another graphic file .mp3- MPEG Layer III compressed audio. This file can be played with QuickTime Player, Real Player, and some other sound players. .wav- Sound file .mov- QuickTime movie. This file can be viewed with QuickTime Player. .avi- AVI movie format .zip- Compressed or "zipped" file; can be opened with WinZip Now we can define a file as: "An associated group of records is a file" Or A file may be defined as named collection of records where record is defined as value assigned to field and a field is the smallest addressable element.

208 In simple words we can define a file as a recommended space in computer to store large amount of information in a systematic manner so as to make the user easy to study the particulars of a matter. There are many different types of files: data files, text files, program files, directory files, and so on. Different types of files store different types of information. For example, program files store programs, whereas text files store text. In the same manner a file containing data is a data file. Thus a data file is a collection of organized data (a group of records). A data file is a collection of logical related records. A record is a structure of logically related fields or elements of information. The data hierarchy of a file in ascending order of complexity is as shown below: Simple bit - a 1 or 0 byte - 8 bits, one character's worth of storage field - a collection of characters, usually in a record record - a common logical unit of data organization file - a collection of records data base - a collection of one or more files Most Complex database For Example: Data File Employee Number Employee Name Date of Joining Department Salary Employee Address E001 E002 E003

E004 Ravinder Nawneet Sanjay Joginder 10/12/1995 08/03/1998 06/07/1993 21/09/1999 Marketing R & D Marketing Accounts 13000 12500 13200 11400 Sirsa Rohtak Hisar Jhajjar A file is an abstract data type. To define a data file properly, we need to consider the operations which can be performed on files. To understand these files, let us look at various types of data files and their operations in more detail. 10.2 Data File Types:

209 Data Files are the backbone of data processing and data management. Depending upon the nature of data, some common types of data files include: 1. Master Files 2. Transaction Files or Activity Files 3. Output Files 4. Report Files 5. Backup Files 6. Scratch Files 7. Reference Files or Table Files 8. Suspense Files or Error Files 9. Open Files 10. History Files 11. Archive Files 10.3 Master and Transaction Files 1. Master files: For any business organization master file is one of the most important file where permanent records are stored. A master file contains relatively permanent data or historical status data. A special kind of master file is a dictionary file which contains descriptions of data rather than the data themselves. A master file contains all the permanent or semi-permanent data needed by an application. Some of the fields may be regularly updated from transactions. All the current data relevant to an application is stored in a master file. Master files do not contain event or activity data but their balances are updated by this data. Adding, deleting, and changing data values or records in the master files are the normal means of updating. These are of two types: 1. Static/Reference Master Files: These files are permanent or semi-permanent in nature and are subject to occasional revision. Examples of these types of files include Vendors, Customers, and Employees etc. of a business organization. A business organization may contain information about existing customers or previous customers or details of all items. In master file some of the records are rarely updated. For Example, General Information about any organization such as name of the company, address, starting date, board of directors are unlikely to be changed.

210 2. Dynamic/Table Master Files: These files are transitory in nature. Examples of these types of files include Price List, Customer Orders etc. of a business organization. Some records of master file need to be updated periodically. For Example, Balances, Stock Balances, Reservation, Records of files, Financial year, Opening balance, Credit/Debit of Banks etc. A master file presents a static view of some aspect of an organization's business at a point of time. For Example, A manufacturing organization may have payroll file, a customer file, a personnel file, an inventory file, a material requirement as master file. 2. Transaction or activity files (temporary): A transaction file is a temporary file of all the transactions (items, bought sold, added, deleted etc) which have taken place in a given period. A transaction file is used to store input data until it can be processed. It is designed to capture transaction or event data. In this type of files, data relating to events are recorded before a further stage of processing. Because transaction files are used to update master files, they often link applications together. For Example, in a payroll processing application for generating pay slips of employees on a weekly basis, the transaction file will contain weekly data of each employee. The transaction file which is temporary in nature contains all relevant data about all the transactions of one type and the purpose of transaction file is to accumulate data about relevant events as they occur and to update master files to reflect the results of most recent transactions.

This file is also called movement or activity file. For example, for an organization it may contain daily sales-records, daily production entries etc. This file after insertion of records is used to update master file. The life of this type of file is very short. The file is in-processing most of time. Changes which are applied to a master file are collected on a transaction file. A transaction file may contain data to add a new record or to remove or modify an existing record. The data normally stored in journals in a traditional accounting system are examples of transaction files. They contain the activity data used

211 to update the running balance listed in the subsidiary and general ledger master files. Batches of source documents are also examples of transaction files. In a batch processing system, groups of transactions are batched or stored together in a transaction file. In an online real time processing system, transactions are processed as they occur (individually) and may then be written to a transaction activity log file as backup. 3. Output Files: The output produced by one program is used as input to another program in many of the applications using multiple programs for data processing. This output is stored in file known as output file, which is later on used by the latter program. 4. Report Files: The reports generated by data processing applications in computer accessible form are stored, until it is convenient to print them. These files are referred to as report files. Keeping report files instead of printed reports or documents is always beneficial as the files are easier to store and carry. 5. Backup files: These files are duplicate copies of a file for security reasons and created as a safety precaution against loss of data, which may be caused due to corruption or inadvertent deletion of the original file. It is very important to create backup files regularly. 6. Scratch files: These are very temporary files and are used to collect data for file update or file sort temporary backup. 7. Reference files or table files: It is a type of master file containing referential data. Reference file contains data that are necessary to support data processing i.e. price lists, dictionary files, present value schedules, amortization tables, table of inventory codes or categories, lookup tables etc. 8. Suspense files or error files: These files include transactions which are not able to be fully processed due to the presence of erroneous data. These transactions are suspended

212 from further processing and may be classified as incomplete or erroneous, needing correction and reentry for processing. 9. Open files: These files are used to record incomplete transactions. They have characteristics of both master and transaction files and is also known as hybrid files. An example would be the open sales order file, open purchase order file, or open invoice file. Open files are waiting for an event to occur to resume processing. These files are not error files. Closing the record usually amounts to setting a flag alerting computer software to the changed status. Later, the closed records will be purged from the file, and perhaps written to a history file. Open files usually remain open indefinitely. Only individual records are purged as the transactions are completed. 10. History files: This type of files provide a permanent record of transactions processed during certain time period. The examples would be bank records, court documentation and records, previous years' financial statements. They could also contain data to support those statements, such as sales history file, purchase order history file, etc. They are accessed or read infrequently and contain inactive past or historical data. 11. Archive Files: An archive file is a file of data in long-term storage usually for legal reasons. Archive files contain all information about the past dealings of the business and would normally be stored on a different site in case of fire etc. 10.4 File Organization In general speaking, a file consists of a collection of records and file management is the way in which the records are organized inside the file. File organization heavily affects system performances as far as record finding and access is concerned. File organization deals with the structure of data in secondary storage devices. Organization refers to the logical arrangement of the records in the file that helps in determining the physical ordering of the records in storage and the access method i.e. set of operations necessary to find a particular record. Access method is the way a record is accessed or read from a file. The main objectives behind organizing a file are ease of file

213 creation, maintenance and to provide an effective way of storing and retrieval of information. File organization represents the arrangement of records within a file whereas data organization represents the arrangement of data within a record. The criteria usually considered important are: • Easy record adding/update/removal, without disrupting fast access to single record or collection of related records • Storage efficiency • Redundancy as a warranty against data corruption Needless to say, these requirements are in contrast with each other for all but it is the designer job to find a good compromise among them, yielding adequate solution to the problem. There are four commonly used file organization techniques that help in computer data processing for business applications which are as follows: 1. Serial Files 2. Sequential Files 3. Indexed Sequential Files 4. Direct or Relative Files Choosing a file organization is a design decision, hence it must be done having in mind the achievement of good performance with respect to the most likely usage of the file. The selection of a particular file organization depends upon the factors like, the type of application, the method of processing for updating files, size of files etc. The best organization to use in a given application is the one that meets the application's data access and processing requirements in the most effective and economical manner. Hence, an application designer must evaluate the strengths and weaknesses of each file organization, before making a choice of the type of organization to be used for the application. Data in files is organized in the following common ways according to user's needs:

214 10.5 Serial Files Serial Records on a serial file are not in any particular order sequence so the serial file is equivalent to an unordered sequence. The data are collected in the file in the order in which they arrive.

Each record is written after the last record in the current file.

This type of organization would not be used for a master file as there would be no way to find a particular record except by reading through the whole file, starting at the beginning until the right record was located.

The order of records in the serial file is according to the time when the data was generated.

Serial files are used as temporary files to store transaction data. Memory dumps, archival files, records of events are the examples of this type of file organization. Location of a record in a file can be done by sequentially searching the records till the desired value of the key attribute is reached. New records are always at the end of the file. Changes and deletions of records in a serial file stored on random access media can be done by locating the record and changing its contents and flagging the record to indicate that the record has been invalidated. The file may be reorganized periodically to remove the holes created by deleting of records. However, updating of the serial file on sequential access media can be done only by creating a new file.

10.6 Sequential and Indexed Sequential Files

Sequential As with serial organization, records are stored one after another, but in a sequential file the records are sorted into key sequence i.e. according to some identifying characteristic called a key and the records are accessible only in the order stored, (in key order). This means that records are stored in ascending or descending order or chronological order of a key field in a sequential file organization. For example, the records of the employee file may be organized sequentially by employee code sequence. Files that are stored on storage media (magnetic tape, cartridge tape etc.) are always either serial or sequential, as it is impossible to write records onto a tape in anyway except one after the other. From the computer's point of view, there is essentially no difference between a serial and sequential file. In both cases, in order to find a particular record, each record must be read, starting from the beginning of the file, until the required record is located. This kind of file organization works well for tasks which need to access nearly every record in a file, e.g. payroll. Hence, sequential files are not suitable for applications that process only one or a few records at a time however, when the whole file has to be processed, sequential file is fast and efficient. The updating and processing of records of a sequential file stored on a sequential access media is carried out in a batch mode. The updating involves the reading of records from both transaction and the master file and matching the two for the key attribute. The additions, deletions, and changes are then carried out in the records of the master file and the updated records are written on the new updated master file.

Advantages of Sequential Files: 1. Simple to understand and use. 2. Easy to organize and maintain. 3. Most economic and efficient to use in applications in which the activity ratio is high. 4. Relatively inexpensive I/O media and devices for storage and data processing.

Disadvantages of Sequential Files: 1. Very uneconomic and inefficient to use for applications in which the activity ratio is low. 2. Entire sequential file requires to be read as to retrieve and update few records. 3. Prior sorting of records in both master and transaction files is must in sequential processing. 4. Same data is stored in many files in case of sequential processing which leads to increased data redundancy.

Indexed sequential

Indexed Sequential files are organized like a large dictionary, i.e., records are stored in order by key and an index is kept which also permits a type of direct access. In sequential files search for a given record on average requires access to half the records in the file which is an inefficient and time consuming operation in case of large files. An index file can be used to effectively overcome this problem, and to speed up the key search as well. A sequential file (records sorted on primary keys) that is indexed and stored on a direct access device such as magnetic disk from where data can be accessed either in random or sequential fashion is called an indexed sequential file. In fact, this type of file organization includes two files for every data file. The data file containing the records stored in the file and an index file containing the key and disk address of each record stored in the data file. The records may be stored randomly but the index table is in the sorted sequence on the key value. With this type of arrangement file can be processed randomly as well as sequentially. The index provides for random access to records, while the sequential nature of the file provides easy access to the subsequent records as well as sequential processing. This technique is referred to as Indexed Sequential Access Method (ISAM).

Advantages of Indexed Sequential Files: 1. Most economic and efficient use for sequential processing where the activity ratio is high. 2. Efficient use for direct access processing where the activity ratio is low.

Disadvantages of Indexed Sequential Files: 1. Requires relatively expensive hardware and software resources as to be stored on a direct access device such as magnetic disk. 2. Requires extra space as compared to other types of files because of the need for index file.

217 10.7 Direct or Relative Files Direct Access or Relative files An alternative access method is direct access, which is based upon the concept that data can be retrieved directly in any order. This is a more sophisticated file organization but essential for certain types of applications, e.g. airline reservation system, teller facility in banking applications etc. Because only a single transaction has to be processed in a single time, the activity ratio is very low in these types of applications so the use of serial/sequential file organization is much inefficient and uneconomical. Therefore a direct/random or relative file organization is recommended for this type of transaction where desired record pertaining to the current transaction at hand can be directly located by its key field value, without having to search through a sequence of other records. For direct access, the file is viewed as a numbered sequence of blocks or records. A direct access file allows arbitrary blocks to be read or written. Thus we may read block 14, then read block 53, and then write block 7. There are no restrictions on the order of reading or writing for direct access files and need to be stored on a direct access storage device (DASD) such as magnetic/optical disk to enable direct access of records. If a record has to be accessed from a number of records stored in the file in which the records are organized physically in a manner to directly access, a record is stored by its key field value. The direct file organization uses an address generating function to convert the record key value into a storage address on the disk on which the file is stored. Each record in the file is stored at the location to which the address generating function maps the record's key field value. The procedure used for mapping the key field value to the locations is called hashing algorithm. To search a record, given its key value, the computer applies this hashing algorithm on the given key to generate its corresponding address. Direct access files are of great use for immediate access to large amounts of information. They are often used in accessing large data bases. When a query concerning a particular 218 subject arrives, the user computes which block contains the answer and then read that block directly to provide the desired information. Not all operating systems support both sequential and direct access for files. Some systems allow only that a file is defined as sequential or direct when it is created; such a file can only be accessed in a manner consistent with its declaration. Advantages of Direct Access or Relative files: 1. Records can be quickly located and retrieved directly, given the key value, in this type of file organization. 2. No ascending or descending order of records in a file is required for transaction processing. 3. Accumulation of transactions into batches is not required before processing them. They may be processed as and when generated. 4. Supports also online applications that need to provide up-to-the minute information in response to enquiries from users. 5. If required, it is also possible to process direct file records sequentially in a record key sequence. Disadvantages of Direct Access or Relative files: 1. Requires relatively expensive hardware and software resources as to be stored on a direct access device such as magnetic disk. 2. It is more difficult to add or delete records in direct files as compared to sequential files. 3. Extra security measures are required for online direct files because these files are to be accessed simultaneously from multiple online locations. 4. The direct file organization is considered to be more complex and costly design. 10.8 Summary A file consists of a number of records. Each record is made up of a number of fields and each field consists of a number of characters. The files when created will have key field which helps in accessing a particular record within a file. The records can be recognized 219 accordingly by that particular key. The records are read into memory one at a time in sequence. The files are stored normally on backing storage media, as they are too big to fit into the main storage all at once. Some files are processed at regular intervals to provide information. Others are processed at irregular intervals. There are various types of files and a variety of file accessing methods. In a traditional file environment, data records are organized using either a sequential file organization or a direct or random file organization. Records in a file can be accessed sequentially or they can be accessed directly if the sequential file is on disk and uses an indexed sequential access method. Records on a file with direct file organization can be accessed directly without an index. By allowing different functional areas and groups in the organization to maintain their own files independently, the traditional file environment creates problems such as data redundancy and inconsistency, programs data independence, inflexibility, poor security and lack of data sharing and availability. Some of the basic methods of file organization are: (i) Serial File (ii) Sequential File (iii) Indexed Sequential File (v) Direct File Serial File contains a set of records in no particular order but the records are stored as they arrive. They do not follow any particular sequence of attribute values. This method of storing records is adopted when it is not possible to arrange the records in any logical order, when the files of the record are not well defined, and the exact use of the file cannot be anticipated. Files are created in this mode by punching the documents in the order of arrival. Then, it is organized into another mode. File on sequential access media are generally organized in the sequential mode. Sequential files may also be maintained on the random access media. The records are arranged in the ascending or descending order of the values of a key attribute in the record. The sequence of records in the file can be changed from one key attribute to

220 another key attribute by sorting the file. The key for sequencing the records may also consist of a group of attributes. Search in a long sequential file can be very time consuming with unnecessary delays in the location of the record and unavoidable. Computer time is being spent in the search routine. Search can be made faster if an index to the file is provided. Such a file organization is called indexed sequential file. Direct files are not maintained in any particular sequence. Instead, the value of the key attribute is converted into the sector address by a predetermined relationship. This converts the value of the key attribute to sector address for the storage and retrieval of the record. This method is generally used where the range of the key attribute values is large as compared to the number of records. Direct and Random access has the same concept. An element of data (byte or record) or instructions can be directly stored and retrieved by selecting and using any of the locations on the storage media. It also means each storage position has a unique address and can be accessed in approximately the same length of time without searching other storage positions.

10.9 Self-Test questions 1. What is a data file? What are its types? 2. Differentiate between master and transaction files. 3. Explain the concept of file organization. 4. State the advantages of sequential file organization over direct file organization. 5. Explain the steps involved in indexed sequential file processing. 6. List the advantages and disadvantages of direct or relative file organization.

10.10 Suggested Readings Desai, C. Bipin, An Introduction to Database Systems, Sp. Ed., Galgotia Publications Pvt. Ltd.

221 Sinha, Pradeep K. and Sinha, Priti, Computer Fundamentals, Third Ed., BPB Publications.

222

UNIT 11 -

INTRODUCTION TO DBMS Objective : Objective of this lesson is to make learn about basic concepts of data base management system. After going through this lesson you will be understand: the concept of database, its characteristics; the functions of DBMS and its advantages; comparison of file processing and data base processing system; Data model, its components & their types; the concept of schema and subschema used in a database. LESSON STRUCTURE: 11.1 Introduction 11.2 Characteristics of Data in Data Base 11.3 Data Base Management System 11.4 File Processing v/s Database Processing 11.5 Data Model 11.6 Types of Data Models 11.7 Data Independence 11.8 Schema and Subschema 11.9 Self – Test questions 11.1 INTRODUCTION A database is an organized collection of facts. In other words we can say that it is collection of information arranged and presented to serve an assigned purpose.

223 An example of a database is a dictionary, where words are arranged alphabetically. Another example is telephone directory, where subscriber names are listed in an alphabetically. Similarly, when you think of a box of cards with names and addresses written as a mailing list, then the box and its contents form a mailing database. Figure 1.1 contains cards with mailing addresses. All these cards are placed in alphabetic order of names and the collection of these cards would be called a database. Fig. 1.1 In order to keep database updated, we may need to perform operations like adding information, removing information, editing existing information, etc. Take the case of a telephone diary, in which you note down the names, addresses and phone numbers of your friends. This is also a type of database. If you make a new friend, you note the information about him/her in the diary, which is equivalent to adding data to a database. If address of your friend changes then you note the new address in your diary, which means editing a database. If you want to send a letter to your friend then you look into the diary to locate the address. This is called searching a database. In case your telephone diary contains a very large number of names, addresses, phone numbers. Moreover, the names are not arranged in any order. Thus, it would be very difficult to locate the names, addresses, and

224 phone number of you friends. Thus, it is essential to arrange the name in some order, say alphabetically, to make the search easy. However, as the number of friends gets larger, managing the database manually becomes difficult. A database management software package is a helpful tool in such a situation. 11.2 Characteristics of data in Database 2.1 Field The smallest piece of meaningful information in a file called a data item or field. Figure 1.2 displays four data items(fields) in a record. NAME, LOCALITY, CITY, STATE, are all known as data items or fields. Figure 1.2 Concept of Fields, Records and File

225 2.2 Record The collection of related fields is called a record. For example, Figure 1.2 contains six records and each record has four fields namely; NAME, LOCALITY, CITY, STATE. 2.3 File File is the collection of all related records. In Figure 1.2 the file contains the list of addresses of six friends. 2.4 Database Database is a collection of related file. Figure 1.3 shows Employees database having related files containing records of employees.

Figure 1.3 Employees database File-1 contains employee records with field Emp_Code and Address. File-2 contains employee records with field Emp_Code and Salary. File-3 contains employee records with field Emp_Code and Name. File-4 contains employee records with field Emp_Code and Designation. Combination of all the four related files, as shown in Figure 1.3, is known as a database of Employees. It removes all data items in files 1, 2,3 and 4, containing common fields.

File-1 File-2 File-3 File-4 Data Base Managemen t System (DBMS) Data Base Managemen t System (DBMS) Data Base
226 2.5

Key Field The key field in a record is a unique data item, which is used to identify the record for the purpose of accessing and manipulating database. In Figure 1.3 all four files File 1, File 2, File 3 and File 4 have one common field, namely Emp_Code. Such a field, called the key field, is used for identifying and relating records.

11.3 Data Base Management System(DBMS) A system is needed that will integrate the data files into database that can provide different views to different users. DBMS is a complex program that acts as a data librarian. Data Base Management System(DBMS) is a software package that allows data to be effectively stored, retrieved and manipulated. The data contained in DBMS package can be accessed by multiple application programs and users. For example, SQL Server, MS-Access, Oracle, DB2 are some commercially used DBMS packages.

3.1 ADVANTAGES OF DBMS DBMS has got the following advantages: (a) It represents complex relationships among different data items. (b) Keeps a tight control on data redundancy. (c) Enforces user defined rules to ensure the integrity of data in a table form. (d) Maintains data dictionary for the storage of information pertaining to data fields and data manipulation. (e) Ensures that data can be shared across all applications. (f) Enforces data access authorization. (g) Has an automatic intelligent backup and recovery procedure of data. (h) Has different interfaces through which users can manipulate data.

11. 4 File Processing system v/s Data Base Processing System

227

Figure 1.4 File processing system These are file-processing systems; they are predecessors of database systems. With file processing each file is processed independently. Payroll System - processed only the faculty data file. Class Scheduling System - processed only class data file Grade Posting System - processed only the student data file. But, suppose someone wants to know the salary paid to instructor who teaches a class scheduled by the class scheduling system. To obtain this information a new program must be written to extract data from both the faculty data file and the class data file. Unfortunately, there is no guarantee that these two files are compatible.

Figure 1.5 Integration of Data Files to Data Base Faculty Data File Class Data File Student Data File Payroll System Class scheduling System Grade Posting System Reports Reports Reports Data Base Managemen t System (DBMS) Data Base Managemen t System (DBMS)

Data Base Faculty Data File Class Data File Student Data File Faculty Data File Class Data File Student Data File

228 Figure 1.6 Data Base Processing This is database-processing system. Since the files (Faculty data file, Class data file, and Student Data file) have been created by the DBMS, all the data is compatible. Three files in the file processing system are integrated into a database that is processed indirectly by the application programs.

4.1 Advantages of Data Base Processing System ¾ Data Base processing enables more information to be produced from a given amount of data. ¾ Elimination or reduction of data duplication ¾ Program/data independence For data base application, application programs obtain data from an intermediary, the DBMS. Obviously, the application programs need not contain data structure. Only DBMS will need this structure. ¾ Better data management When data is centralized in a database, one department can specialize in the maintenance of data. ¾ Affordable sophisticated programming These may also be considered as disadvantages of file processing system.

4.2 Disadvantages of Data Base Processing ¾ Expensive • DBMS: The DBMS may cost as much as \$10,000 to buy. •

More Hardware: Extra memory space required for the storage of DBMS, extra hardware required for client/server set up etc.

Faculty Data File Class Data File Student Data File Data Base Management system (DBMS) Reports Class Scheduling System Grade Posting System Reports Reports Payroll System

229 •

Higher Operating Cost: Operating costs for some systems will be higher. ¾ Complex Large amount of data in different formats can be interrelated in the database. This means database tends to be complex. Both database system and the application program must be able to process their structure. ¾ Recovery more difficult Backup and recovery are more difficult in the database environment. This is because of increased complexity and because databases are often processed by several users concurrently. ¾ Data Vulnerability to Failure Integrated data base system may threaten privacy. Integration of data into central place may create some problems. Because of the data from individual data files is integrated in a central place in a data base, the sense of owner and as a result, the responsibility for data are easily lost. As a result inaccurate data may not be detected. This can cause serious problems unless extensive data integrity data validity measures are implemented. A failure in one component of an integrated system may stop the entire system.

11.5 Components of Data Models

5.1 Entity An entity is a class of persons, places objects, events or concepts about which the information is recorded.

- Persons: Employees, Customer, Students, Suppliers etc.
- Places: Branch Office, Building, Room etc.
- Objects: Book, Machine, Vehicle etc.
- Events: Sale, Reservation Registration Order etc.
- Concepts: Qualifications, Account, Course, Stocks etc.

In a banking environment examples of entities are CUSTOMERS, BANK ACCOUNTS, TRANSACTIONS. In a warehouse the entities are SUPPLIERS, PARTS, SHIPMENTS.

230 5.2 Attributes (Data Elements) Every entity has some basic 'attributes' that characterize it. An attribute may be elementary or logically grouped in to super attributes called compound attribute. In a banking environment entity 'CUSTOMER' may be described by such attributes NAME, ADDRESS, Identification No. SEX, AGE.(fig. 1.7) and compound attribute 'NAME' is described in fig 1.8. CUSTOMER NAME ADDRESS Identification No SEX AGE Figure 1.7 Elementary Attributes of Entity CUSTOMER CUSTOMER NAME ADDRESS Identification SEX AGE First Middle Last Figure 1.8 Compound Attributes NAME of Entity CUSTOMER ACCOUNT ACCOUNT NUMBER ACCOUNT TYPE ACCOUNT

BALANCE Figure 1.9 Elementary Attributes of Entity ACCOUNT An attribute is often called a data element, data field, a field, a data item or elementary item. 5.3 Data value A data value is the actual data or information contained in each data element. Data element Account no can take the value 5567

231 Account Type can take the value Saving Account Balance Can take the value 22345-50 Entity Attributes Values Customer Identification No. Customer Name Customer Address 2456789 ABC 55, Snow Street New York. Account Account Type Account Number Account Balance Saving 62430 2253.50 Supplier Supplier Name Supplier Number Supplier Address XYZ 245678 80,

Nehru Place, New Delhi The information about an enterprise can be represented with help of a number of entities and an individual entity can be represented by a number of data elements. The values taken the data elements are referred to as data. One set of value taken by the data elements of an entity is called an entity occurrence. The entities are interconnected by certain relationships. "An inherent model of the entities with the data elements representing them, together with the relationship interconnected the entities, is called a conceptual model" 5.4 Relationships between entities Three types of relationships exist among entities. These are: a) One-to-One b) One-to-many c) Many-to-many a) One-to-One Relationship (1:1)

232 A one to one (1:1) relationship is an association if one entity is associated to one and only one entity. For example, in a hospital environment one patient has one bed. This shows one to one relationship between bed and patient. Another example of such relationship between in Student and Class Figure 1.9(a) and 1.9(b) b) One-to-Many Relationship (1:M) A one-to-many (1:M) relationship exists when one entity is related to more than one entity. For example, in a hospital environment in a single room their may be more than one bed. This shows one too many relationship between bed and room. Another example of such relationship between in Teacher and Course Figure 1.10(a) and 1.10(b) Fig. 1.10 One-to-many(1:M) RELATIONSHIP c) Many-to-Many Relationship (M:M)

BED STUDENT PATIENT CLASS 1.9a

ROOM COURSE BED TEACHER 1.10a 1.9b 1.10b Fig. 1.9 One-to-one(1:1) RELATIONSHIP

233 A many-to-many (M:M) relationship describes entities that may have many relationships among each other. For example, in a hospital environment, a patient may be treated by many surgeons and a surgeon may treat many patients. Another example of such relationship between in Teacher and Student. Figure 1.11(a) and 1.11(b) Fig. 3.5 Many-to-many(M:M) RELATIONSHIP 11.6 Data Model A database consists of data items and the associations among them. There are large numbers of different data item type and so we need a map showing how they are associated. This map is called a data model. A data base model is a vocabulary for describing the structure and processing of a database. There are two reasons for studying database model. - They are important data base design tool. Data base Models can be used for both logical and physical data box design much as flow charts or pseudopodia are used for program design. - Data base models can be used to categorize DBMS products Data base design phases - Conceptual Model or Enterprise model or External Model

Patient TEACHER Surgeon Student 1.11 a 1.11 b

234 -

Logical Model - Physical Model or Internal Model or Physical Structure.

Figure 1.12 Various levels of Database As we move from external level to stored database data independence goes on lost. 6.1 Conceptual Model One of the major responsibilities of the data administration function is the development of a conceptual model, also called an enterprise model, of the organization. The model embodies the entities and their relationships and is the tool used to represent the conceptual organization of data. The DBMS is not a Conceptual Model Logical Model Physical Model EXTENAL VIEW (User-1) EXTENAL VIEW (User-n) CONCEPTUAL SCHEMA INTERNAL SCHEMA EXTENAL LEVEL External/conceptual mapping CONCEPTUAL LEVEL Conceptual/ internal mapping INTERNAL LEVEL Stored Database

EXTRNAL VIEW (User-2)

235 factor

in designing conceptual model. The choice of DBMS should be made after the conceptual model is designed. 6.2 Logical Model The conceptual or enterprise model has to be mapped to the logical model being used as the underlying structure for a DBMS. Design of a logical model is dependent on the DBMS to be used. 6.3 Physical Model Logical Model is mapped to physical model. Railway map does not bear much relation to the physical layout of the tracks and stations. It does not show bends in the tracks as they are in reality and does not show which are long and which short links are. It merely shows the associations between stations. It does not resemble the reality closely. We might think it as logical representation of logical model. Before attempting to describe the physical representation relationships between data, we must discuss the way in which the users of the data base (terminal users or applications programmers) see the relationships. If describes the physical storage structure of the database. Note : We use the term data model in the generic sense, i.e. it may be a conceptual or a logical or an inter (physical) model. 11.7 Types of Data Models There are five types of data models depending upon the relationships between entities 7.1 Relational data model 7.2 Hierarchical data model 7.3 Network data model 7.4 Object-oriented data model 7.5 Distributed data model 7.1 Relational Data Model

236 The relational data model was first proposed by Dr. E. F. Codd in a seminar paper in 1970. In relational data model, the entities and their relationships are represented with two dimensional tables. e.g. Col. 1 Col. 2 Col. 3 Col. 4 Employee No Name Age Sex Row 1 010110

Ram 21 M Row 2 010100 Sham 22 M Row 3 010111 Anita 21 F Row 4 201101 Hari 22 M Row 5 301101 Sunita 19 F Row 6 411001 Mohan 20 M Row 7 422002 Sohan 19 M Fig. 1. Employee table The data is represented in a two dimensional table which is called a relational model of the data. In relational data model such type of table is know as relation. But to avoid confusion between relation and relationship between entities, sometimes is called table. Every column in a relation is an attribute Each row of the relation is known as table. If the relation has n columns, then each raw is referred as n table. A relation that has n columns or attributes is send to be degree n. Each attribute has a domain, which is the set of values that attribute can have. {M,F} e.g. domain of the Sex attribute is the two value M and F. Domain of the Age attribute is all positive integer less than 100. A relation of degree n has n domains, all of which need to be unique. The generalized format EMPLOYEE (Employee_number, Name, Age, Sex) is called relation structure. 7.1.1 Properties of Relations :- With in any one relation:

237 • No. duplicate rows • Entries in any column are all of the same find • Order of the columns is immaterial. • All value atomic; i.e. they can not split further (without loss of information) Normalization :- The process of crystallizing the entities and their relationship in table formats using relational concepts is called the normalization process. 7.1.2 Advantages of Relational Data Model • Simplicity One of the major advantages of the relational approach is its simplicity, i.e. the ease of understanding for the end-user. The end-user do not have to be concerned about the physical storage structure, they can be oriented words the information content of their data and need not worry about physical representation details. The end user is concerned with the end-user's views supported by the date base management system. The end-users is not concerned with row the date is stored, how it is accessed, and in what format it is stored. The ability to use the data box without showing the representation details is called data independence. • Non-procedural request Because there is no positional dependency between the relations requests do not have to reflect any preferred. Structure and therefore can be non-procedural. • Data independence This should be one of the major objectives of any DBMS. In this model end-user is not concerned with how the date is shared, how it is accessed, and in what format it is stored. The model provides a relatively higher degree of data independence. To be able to make use of this property of the relational data model, however, the design of the relation must be complete and accurate. • Theoretical Foundation

238 The relational data model is based on well-developed mathematical theory of relations. The rigorous method of designing a database gives this model a sold foundation. This found of foundation does not exist for the other data models. 7.1.3 Disadvantages of Relational Data Model Although some data base management system based on the relational data model are commercially available today, the performance of a relational DBMS has not been comparable with the performance of a DBMS based on hierarchical data model, network data model, object oriented data model or distributed data model. Term Alternative Appropriate equivalent Relation

Attribute Tuple Table Column Row File Field Record Figure 1.12 Hierarchical Structure Level 1 Root Node Level 2 Level 3 239 A column or set of columns is called a 'candidate key' (often shortened to key) when its value uniquely identity the rows of the table. 7.2 Hierarchical Data Model 7.2.1 Hierarchical Tree Structure A hierarchical tree structure is made up of nodes and branches. The highest node of hierarchical tree structure is called root. Nodes at lower levels (Second, third,) are known as dependent nodes. A node is a collection of data attributes describing the entity at that node. A hierarchical tree structure has to satisfy that following conditions :- • A hierarchical tree structure always starts with a root node. All nodes except the root are related to one and only one node on a higher level than themselves i.e. every node occurring at level 2 has to be connected with one and only one node accuracy at level 1, every node at level 3 has to be connected with one and only one node at level 2 and so on. A B E G C D I F J H K L

Attribute Tuple Table Column Row File Field Record Figure 1.12 Hierarchical Structure Level 1 Root Node Level 2 Level 3 239 A column or set of columns is called a 'candidate key' (often shortened to key) when its value uniquely identity the rows of the table. 7.2 Hierarchical Data Model 7.2.1 Hierarchical Tree Structure A hierarchical tree structure is made up of nodes and branches. The highest node of hierarchical tree structure is called root. Nodes at lower levels (Second, third,) are known as dependent nodes. A node is a collection of data attributes describing the entity at that node. A hierarchical tree structure has to satisfy that following conditions :- • A hierarchical tree structure always starts with a root node. All nodes except the root are related to one and only one node on a higher level than themselves i.e. every node occurring at level 2 has to be connected with one and only one node accuracy at level 1, every node at level 3 has to be connected with one and only one node at level 2 and so on. A B E G C D I F J H K L

Attribute Tuple Table Column Row File Field Record Figure 1.12 Hierarchical Structure Level 1 Root Node Level 2 Level 3

239 A column or set of columns is called a 'candidate key' (often shortened to key) when its value uniquely identity the rows of the table. 7.2 Hierarchical Data Model 7.2.1 Hierarchical Tree Structure A hierarchical tree structure is made up of nodes and branches. The highest node of hierarchical tree structure is called root. Nodes at lower levels (Second, third,) are known as dependent nodes. A node is a collection of data attributes describing the entity at that node. A hierarchical tree structure has to satisfy that following conditions :- • A hierarchical tree structure always starts with a root node. All nodes except the root are related to one and only one node on a higher level than themselves i.e. every node occurring at level 2 has to be connected with one and only one node accuracy at level 1, every node at level 3 has to be connected with one and only one node at level 2 and so on. A B E G C D I F J H K L

240 • Every node consists of one or more attributes describing the entity at that node. • Except that, every node has one node related to it at a higher level, and this is called its parent. No element can have more than one parent. Each element can have one or more elements related to it at a lower level. These are called children. • Every node except, of course, the root has to be accessed through its parent node. Therefore the access path to every node with a hierarchical data model is unique. e.g. node I can be reached only through the path A to G to H to I. Node D can be reached only through the path A to B to D. Hence a hierarchical data model consists of linear path. [Child - Parent relationship is hierarchical tree structure] A hierarchical data model is one that organizes data in hierarchical tree structure. Every occurrence of root node began a logical data base record i.e. a hierarchical database made up of a number tree. In hierarchical data model, the node at level 1 is called the parent of the nodes at level 2. 7.2.2 Storage operations with a Hierarchical Data Model • Insertion : A child node occurrence cannot exist without parent node. If the hierarchical data model A, no new student can be added to the data base unless he or she is taught by a teacher. • Deletion: When a parent node occurrence is deleted, the child occurrence is deleted too. In the hierarchical data model A, when a teacher node occurrence is deleted, all the node occurrence of student s taught by the teacher is also deleted. This results in the loss of information about those students.

241 Note :- Many-to-many relationship cannot be efficiently represented with one hierarchical data model, which always represents a one to many relationship. This problem can be overcome by introducing two hierarchical data models.

Example A + B Hierarchical Data Model :Teacher database Level 1 (Root /Parent) Deptt # Teacher Name Teacher Address Subject Code Level 2 Child Student Roll No.

Student Name Class Section Age Teacher Student Teacher Student Teacher Student

242

The entity TEACHER is the root node; every teacher has a root node occurrence. A hierarchical data base record occurrence for the hierarchical data model Teacher. The data base record is for the teacher S.Kumar T111 S.Kumar 12F,Nehru Place, N. Delhi. C 001 The data base record is for the teacher P.Kumar T222 P.Kumar 12 , Defense Colony, Hisar. C 002 Hierarchical data model for entity STUDENT Level 1 (Root /Parent) Roll No. Student Name Class 1111 AAAA M.C.A. V A 28 3333 EEEE M.Tech. B 26 1111

AAAA M.C.A. V A 28 2222 BBBB M.Tech. B 26 4444 FFFF MSc A 24 Many One Teacher Student

243 Level 2 Child Dept No. Teacher Name Subject Address Entity STUDENT is root node, every student has root node occurrence. Data Base record for Student AAAA 1111 AAAA MCA V T111 S. Kumar C001 ----- T222 P. Kumar C002 -- ----- The hierarchical data models A and B are different; these data models are of the same university. The difference

is the result of the two ways of linking of the entity relationships. 7.3 Advantages of a Hierarchical Data Model • The Major advantage of the hierarchical data model is existence of proven data base management systems that use hierarchical data model as the basic structure. • There is a reduction of data dependency. When the two hierarchical data models from fig c are used. • The relative simplicity and ease of use of the hierarchical data model and the familiarity of data processing users with a hierarchy are major advantages. 7.4 Disadvantages of Hierarchical Data Model. • The many-to-many relationship can be implemented only in a clumsy way. This may result in redundancy in stored data. We

244 know that redundancy at logical level is not necessarily bad on the contrary. It promotes simplicity redundancy at the physical level, however is undesirable. • As a result of the strict hierarchical ordering, the operations of insertion and deletion become unduly complex. • Deletion of parent results in the deletion of children. As a result, users have to be careful in performing a DELETE operation. • Root is the dominant node type. Any child node is accessible only through its parent node. 7.3 Network Data Model 7.3.1 Network Data Structure Any item in a network structure can be linked to any other item. In this structure a child can has more than one parent, in other words a node at lower level can be linked to more than one nodes at a level higher to it. I II Fig. Network data structure are network structures, one or more nodes have multiple parents. Family tree is a tree structure only because it does not include woman and if they were both shown in the diagram, it would be a network(plex) structure. Any network structure can be reduced to a tree structure by introducing redundancy. In some cases the redundancy involved in doing so is small and might be tolerated. In other cases it is excessive. A data model that interconnects the entities of an enterprise in network structure is known as network data model. 1 2 3 4 1 2 3 4 5

245 7.3.2 Advantages of Network Data model • The major advantage of network data model is there are successful data base management systems that use the network data model as the base structure. • The many-to-many relationship, which occurs quite frequently in real life, can be implemented easily. 7.3.3 Disadvantages of network model The main disadvantage of the network data model is its complexity. 7.3.4 Storage operations with network data modal • Insertion: A new owner/ member can be added without member/owner. The information about a patient can be added without assigned surgeon. Similarly a surgeon can be added without having new patient to operate it. • Deletion: If a parent is deleted then a child connected to it may or may not be deleted. The information about a patient can be deleted without deleting a

surgeon connected to it. 7.4 Object Oriented Data Model(OODM) 7.4.1 Object

245 7.3.2 Advantages of Network Data model • The major advantage of network data model is there are successful data base management systems that use the network data model as the base structure. • The many-to-many relationship, which occurs quite frequently in real life, can be implemented easily. 7.3.3 Disadvantages of network model The main disadvantage of the network data model is its complexity. 7.3.4 Storage operations with network data modal • Insertion: A new owner/ member can be added without member/owner. The information about a patient can be added without assigned surgeon. Similarly a surgeon can be added without having new patient to operate it. • Deletion: If a parent is deleted then a child connected to it may or may not be deleted. The information about a patient can be deleted without deleting a

surgeon connected to it. 7.4 Object Oriented Data Model(OODM) 7.4.1 Object

Patient Surgeon

246

An object is defined as an entity with a set of predefined operations and data items that manipulate and access with in it or out side it. Fig. Architecture of Object 7.4.2 Class It is a generalized description that describes a collection of similar objects.

OBJECT OPERATIONS DATA

Class : Furniture Buy Sell Move Cost Dimensions Colour Object : Chair Buy Sell Move Cost Dimensions Colour Object : Table Buy Sell Move Cost Dimensions Colour

247 In fact objects are variable of type class. A class is thus a collection of objects of similar type. For example, mango, apple, and orange are members of class fruit. Each object essentially consists of some data that is private to the object and set of functions (Operations or Methods) that operate on those data. 7.4.3 Abstraction It is the selective examination of the certain aspects of a problem while ignoring the other aspects of the problem. In other sense we can say that abstraction is to consider only those aspects of the problem that are useful for a certain purpose and to hide all other aspects that are not relevant for the given purpose, does the abstraction mechanism helps us to represent a problem in a simple way. Many different abstractions of a same problem are possible depending upon the requirements. It is a powerful tool of object oriented technique for reducing complexity of software. Abstraction can be a way of increasing productivity of a software. 7.4.4. Inheritance

m 1 m 2 m 3 m 4 m 5 m 6 m 7 m 8 Data

Library Members Faculty Students Staff Undergrad Postgrad Research

248 This feature helps us to define a new class helps us a new class or modifying extending or modifying and existing class. The original Class is called base class or super class and the new class obtained through inheritance is called derived class or sub class. Fig Library member is a based class for the derived classes Faculty, Students and Staff. An important advantage of inheritance mechanism is code reuse. 7.4.5 Message Objects communicate with one another by sending and receiving information much like same way the people messages to one another. Methods are the only means available to other objects for accessing a manipulated the data for private the data. The methods of an object are invoked by sending messages to it. A set of valid messages to an object constitutes its protocol. 7.4.6 Polymorphism It means poly (many) morphism(forms) i.e. ability to take more tan one form. With the help of a polymorphism a programmer can send a generalized message to a set of objects, which may be of different types. The main advantage of polymorphism is code reuse and maintenance 7.4.7 Object-Oriented Data Models(OODM) Object-Oriented Data Models (OODM) are dependent on the object- oriented programming paradigm. It enables us to create classes, organize objects, structure an inheritance hierarchy and call methods of other classes. Besides these, it also provides the facilities associated with standard database systems. However, object oriented database systems have not yet replaced the RDBMS in commercial business applications. Following are the two different approaches for designing an object- oriented

data model: (a) Designed to store, retrieve and manage objects created by programs written in some object oriented language (OOL) such as C++ or Java. (b) Designed to provide object-oriented facilities to users of non object-oriented programming languages (OOPLs) such as C or Pascal. The user will create classes, objects, inheritance hierarchy, and so on and the database system will store and manage these objects and classes. This second approach, thus, turns non-OOPLs into OOPLs. A translation layer is required to map the objects created by user into objects of the database system. 7.4.1 Storing and Accessing Objects The data part of all objects of a class is stored individually in a database. The program part of the class are stored as part of the database schema along with the type definitions of the classes. Objects can also be given names to allow their access. However, such an approach will not work when the numbers of objects in a database are large in numbers, say in the range of millions. Another method is to collect objects into a group. A group is given a name. The programs will then iterate over a group to find out the required object. A special case of a collection is the class-extent. A class-extent is the collection of all the objects belonging to a class. Whenever a new object needs to inserted be inserted into a class, it is inserted into its class-extent automatically. Similarly, whenever a new object needs to be deleted, it is deleted from the respective class-extent. 7.4.2 A Sample Object Oriented Data Model A sample object oriented database is shown in Figure. 1.14 The solid lines (--) in the figure represents class-subclass link. The dash(...)represents attribute-domain link. The class Vehicle is the root of a class composition hierarchy that involves Vehicle Specs and Company. The class Vehicle is also the root of a class hierarchy that involves the subclasses Two Wheeler and Four Wheeler. The class Company is in turn the root of a class hierarchy that involves the subclasses Domestic Company and Foreign Company. It is also the root of a class composition hierarchy that involves the class Employee.

250 7.4.3 Advantages of OODBMS over RDBMS Following are the advantages of Object Oriented Database System over Relational Database System: (a) RDBMS does not allow nested structure. Nested structure applications are found in CAD/CAM, CAE, aerospace science, etc. It is easier to navigate through these complex structures in the form of objects rather than in the form of tables and records. (b) Relational databases have only a limited number of data types like integer, char, text, etc. Although we can create complex data types in relational database by combining fields, these are restricted by their base data types. But OODBMS allow arbitrary data types and sub-classing with inheritance.

251 (c) Object oriented databases are useful in those applications where the relationships among data elements are more important than data items themselves. However, relational databases are used in those applications where the values of data elements are more important. If a record makes sense only when used in context of other records, then an object-oriented database would be more appropriate.

7.5 Distributed Data Model (DDM) Some organizations use distributed database system rather than a centralised database system. In the centralized database system, data resides in one single location. But in a distributed database system, database is stored in several computers. The computers in a distributed system communicate and exchange data among one another using leased lines, telephone lines or other means of communication. These computers do not share memory or clock. Each of the computers in a distributed system participates in the execution of transactions. These computers can also be known as sites or nodes.

7.5.1 Advantages of Distributed databases (a) Data Sharing: Since data is distributed on multiple computers, user operating on one computer can work on data available on any another computer. (b) Distributed Control: Unlike Centralized database system, where a single database administrator controls the database, in a distributed system responsibility of control is divided among local administrators for each computer. Thus, local administrators have a certain degree of control over the database stored locally. (c) Reliability and Availability: Even if one site fails in the distributed system, the remaining sites can continue working. If data is replicated among multiple computers, failure of anyone computer does not cause shutdown of the entire database system. (d) Faster Query Processing: Queries involving data at several sites can be split into sub queries. These sub queries can then be executed in parallel by several sites. Such a parallel computation allows faster

252 processing. 7.5.2 Disadvantages of Distributed databases (a) Increased cost: It is more difficult to develop software packages to implement distributed database system. Also cost is involved in physically

linking different sites. (b) More Error Prone: Since the sites that comprise distributed database system operate in parallel, it is very difficult to ensure the correctness of data. The algorithms are more error prone and have greater potential for bugs. (c) Increased Overhead: Maintaining physical links between the sites and exchanging messages among computers is an additional overhead associated with distributed systems, that are absent in the case of centralized systems.

11.8 Data Independence "The ability to use the data base without knowing the representation details about data regarding (Where, how and what) is called data independence" A database is view through three levels of abstraction.

The ability to modify a schema definition in one level without affecting a schema definition in the next higher level without effecting is called data independence Data independence is crucial from the economic viewpoint. An enterprise may spend thousands of dollars for implementation very simple change such as increasing field length or adding new field. An ideal data base design should provide, to some degree, for changes of this nature without affecting the application programs. It should be noted that extent of data independence provided depends on the data base design, as well as base management system (DBMS). Data independence provides a solution to the problems mentioned above. Unfortunately, DBMS packages available today do not appear to achieve full data independence. Since a data base design depends on the DBMS package, it seems difficult to achieve total data independence, even with the best data base design.

253 8.1 Why data independence? • To allow the DBA (Data Base Administrator) to make changes in the content, location, representation and organization of a database without causing reprogram of application programs which use the data base. • To allow the supplier of data processing equipment and software to introduce new technologies without causing re-progressing of the customer's applications. • To facilitate data sharing by allowing the same data to appear to be organized differently for different application programs. • To provide the centralization of control needed by the data base administrator to insure the security and integrity of the database.

8.2 Two stages of Data Independence • Logical data independence • Physical data independence • Logical Data Independence Logical data independence means that the overall logical structure of the data may be changed without changing the application programs. • Physical data independence Physical data independence means that the physical layout and organization of data may be changed without changing either the overall logical structure of the data or application programs.

It is more difficult to achieve logical data independence rather than physical data independence.

254

FIGURE Two stages of data independence 8.3 How to achieve Data Independence. • For an application viewpoint, data independence is not a programming technique but a programming discipline. For example to avoid recompilation for any change, it is advisable not to define constants in a program. A better technique is to pass the values as parameters to the program. • All information about present needs of the enterprise and adequate information about intended needs should be reflected in the conceptual model in the design phase of course, all possible uses of the data base cannot be anticipated, but for most enterprise to basis data i.e. entities and relationships between them, is relatively stable. • Another aspect of data independence is that the internal representation of the data may differ from the form in which the application program

External Model External Model External Model External Model Application 1 Application 2 Application 3 Application 4
 Conceptual Requirement Application 1 Conceptual Requirement Application 2 Conceptual Requirement Application 3
 Conceptual Requirement Application 4 Conceptual Model Logical Model Internal Model One Stage of Data
 Independence (Logical Independence) One Stage of Data Independence (Physical Independence)

255

wishes to use it. Unfortunately, few DBMSs can provide that kind of data independence. 9 The degree of data independence provided depends on the thoroughness of the data base design phase. A thorough analysis of the enterprise's entities and their relationship minimizes the impact on other programs of changes in the data requirements of one program. That is what data independence is really all about? 11.9 Schema and Subschema 9.1 Schema The overall logical database description is referred to as a schema. The term logical means the data as it would be presented to a human. In fact, the data may be shared on the file in a completely different form. Schema is sometimes also referred as an overall model of data, a conceptual model or conceptual schema. A schema is a chart of the types of data that are used. It gives the names of entities and attributes and specifies the relations between them. It is a framework into which the values of the data items can be fitted. Employee# Employee Name Date of Joining Sex Designation Salary Like the display at an airport giving arrival and departure information. The schema will remain the same while the values fitted into it change from instant to instant. When the schema framework is fitted in with data item values, it is referred to as an instance of the schema. 9.2 Subschema The subschema defines a subset of schema to be seen by a application programme or uses. It is the application programmer's view of the data he or she uses. It is also known as external view. Many different subschema can be derived from one schema.

256 The application programmer in terminal uses does not need to know about the entire database schema. The programmer or terminal user is concerned with his particular applications and the records, which they require. That is to say it gives to a users a window through which he can view only that port of the database, which is of interest to him/her. Schema Subschema

Figure An example shows the Concept of Schema and Subschema 11.10. Self TEST QUESTIONS: 1. Define the following terms: DBMS, DBA, Data Independence. 2. Explain the difference between a file processing system and data base processing. 3. Discuss the various phases of data base design. 4. Explain the difference between external, internal and conceptual schema and subschema. 5. Explain the various data models. 6. Explain the various relationships among entity. 7. Define primary key and foreign key. 8. What do you mean by normalization? Explain. 9. Define classes, objects, data hiding, data abstraction, and encapsulation. 10. Explain the concept of centralized database distributed database. 11. Discuss the advantages and disadvantages of using distributed database. 12. What is logical data independence why is it important? 13. Explain the difference between logical and physical independence.

MAN MAN WOMAN WOMAN E N G A G E D

M A R R I E D D I V O R C E D M A R R I E D

257

UNIT12 -

NETWORK FUNDAMENTALS Objectives: The motive of this chapter is to present an introduction to computer networks and within that framework the key network concepts, types, purpose, benefits and limitations of computer networks are discussed.

Structure: 12.1 Introduction to Computer Networks 12.2 Defining Computer Networks 12.3 Why Computer Networks? 12.4 Uses of Computer Networks 12.5 Benefits and Limitations of Computer Networks 12.6 Types of Computer Networks 12.7 Some Other Computer Networks 12.8 Summary 12.9 Self-Test questions 12.10 Suggested Readings

12.1 Introduction to Computer Networks The rapid growth of computer networks has spurred the spread of data communications and these networks have become an integral part of business life. The productivity and profitability of organizations has been enhanced significantly by this revolutionary tool. Computer Networks allow a business to streamline the overall operation and to interact with suppliers and customers very efficiently. In many cases computer networks are seen as the solution to every problem within business, and to some extent the magic wand to tackle all problems with. This technology is accelerating as more people discover the power of computers and communications networks both for business and for homes. Individuals use computer networks almost daily to conduct personal and professional business. The day-to-day transactions at department stores, banks, reservation counters, and other businesses are all dependent upon computer networks. Whilst it is not necessary to understand fully the technical details, it is necessary to understand in full the meaning of computer networks and what precise purpose the introduction of a computer network into an organization serves.

12.2 Defining Computer Networks Communications system connecting two or more computers and their peripheral devices to exchange information and share resources is known as a computer network. It is a group of devices such as computers and printers connected together or to a hub or switch in order to share information or tasks. Networks also include software and cables so that they can communicate with each other and share information, software, peripheral devices, and processing power. A computer network can be seen as an interconnected collection of autonomous computers. Two computers are said to be interconnected if they are able to exchange information, and when no other computer can start, stop or control, the computer is said to be autonomous i.e. there is no dictatorship. A computer network may also be referred to as an arrangement consisting of a group of computers linked in such a way so they can send information back and forth between themselves. It is a system for data communication among two or more computers that interconnects computer systems at different sites. These networks may be fixed i.e. connected through cables permanently or temporary i.e. connected as via modems. Thus, a computer network may be defined as a collection of two or more computers and related devices, connected through some communication channel in such a way that allows them to share data, hardware, and software.

12.3 Why Computer Networks? The first and foremost need of computer networks is to enable people to share information or tasks and that can be achieved by sharing all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource and the user. Secondly, to provide high reliability by having alternative sources of supply such as having replicated files on two or three machines, so if one of them is unavailable due to a hardware failure, the other copies could be used. Thirdly, the organizations have a tendency to limit their resources but have a desire to share these resources so as it may result in cost reduction. Another need may be scalability i.e. the ability to increase system performance gradually as the workload grows just by adding new processors. Lastly, a powerful communication medium among widely spread people can be a computer network.

12.4 Uses of Computer Networks The following uses of computer networks serve the purpose of computer networks in organizations.

- Share resources, such as hard drives, printers, scanners, etc., to get more use out of office equipment.
- Share data on a central server, where all data is collected.
- Share files in electronic form to reduce printing costs and paper waste.
- Share software with multi-user licenses.
- Perform data backups quicker and easier with files on shared drives.
- Exchange information via an internal network.
- Allow access to the Internet.
- Use the network and the Internet to allow access to data from anywhere in the world
- Integrate the whole operations of business organizations into a networked operation, including sales activity, stock holding, quotations, ordering raw materials, control the production process, process invoices, process all the accounts, analyze business performance, quality control.

260 Computer networks also have many uses to deliver services to private individuals at home. The people can use computer networks for achieving the following purposes: • Access to remote information (e.g. Home shopping, Billing etc.) • Person-to-person communication (e.g. E-mail, Video conferencing etc.) • Interactive entertainment (e.g. Video on demand, Live shows on TV & Radios)

12.5 Benefits and Limitations of Computer Networks The most obvious benefit of computer network is that one can store virtually any kind of information at, and retrieve it from, a central location on the network as well as access it from any connected computer. One can store, retrieve, and modify textual information such as letters and contracts, audio information such as voice messages, and visual images such as facsimiles, photographs, medical x-rays, and even video segments. A network also enables to combine the power and capabilities of diverse equipment and to provide a collaborative medium to combine the skills of different people regardless of physical location. Computer networking enables people to share information and ideas easily, so they can work more efficiently and productively. Networks also improve commercial activities such as purchasing, selling, and customer service. Networks are making traditional business processes more efficient, more manageable, and less expensive. The key benefits of computer networks for business organizations can be summarized in the following manner: • Increase employee productivity • Reduce office equipment costs • Backup and protect your data • Communicate better with customers • Provide access to your data 24-hours a day • Cost-Effective Resource Sharing • Streamlined Business Processes • Freedom to Choose the Right Tool • Powerful, Flexible Collaboration between Companies

261 • Improved Customer Relations • Secure Management of Sensitive Information • Worldwide, Instantaneous Access to Information

Limitations of Computer Networks Although the benefits of having a network are significant, technology also presents some limitations to the networks. • A network's sharing capabilities can be difficult to configure initially •

Network cabling can be difficult to manage • Shared Internet access may not perform satisfactorily • Network connections drop unexpectedly • Network is too insecure

12.6 Types of Computer Networks Computer networks come in many different shapes and sizes. Over the years, the networking industry has projected terms like "LAN", "WAN" and "MAN" attempting to define sensible categories for the major types of network designs. The precise meaning of this terminology remains lost on the average person. However, Computer Networks can be classified based on their characteristics and according to the distance between individual computers that are attached to the network. Following are

the characteristics used to categorize different types of networks: Topology The geometric arrangement of positioning computer

systems is known as a network topology. Common topologies include a bus, star, and ring. Architecture Network Architecture basically means how computer networks should be made and can be broadly classified as using either a peer-to-peer or client/server architecture. Computers on a network are sometimes called nodes. Computers and devices that allocate resources for a network are called servers.

262

Peer-to-Peer: Sometimes called P2P, these networks are the simplest and least expensive networks to set up but provide less performance under heavy loads. In this type of architecture, each computer (workstation or node) has equal capability and responsibility i.e. none of the node is superior over other nodes. P2P networks are simple in the sense that the computers are connected directly to each other and share the same level of access on the network, hence the name. Computer 'X' will connect directly to Computer 'Y' and will share all files with the appropriate security or sharing rights. If many computers are connected a hub may be used to connect all these computers and/or devices. One can opt for Peer-to-Peer when: • have five or fewer PC's. • want to share files and printers. • want easy installation. • want users to manage their own computers. • do not frequently share large files such as CAD or graphic files. • there is a tight budget.

Client/Server: They consist of the server which stores the files or runs applications and the client machines, which are the computers used by workers. Here, server is having a superiority over other nodes and will serve the client nodes. Hence, server is more powerful and clients have to completely rely on server. Using a client/server setup can be helpful in many ways. It can free up disk space by providing a central location for all the files to be stored. It also ensures the most recent copy of that file is available to all. A server can also act as a mail server which collects and sends all the e-mail or a print server which takes all the print jobs and sends them to the printer, thus freeing computing power on the client machine to continue working. The client/server network is the most widely-used type of networking because of its efficiency. Whereas the computers in a peer-to-peer network can perform multiple operations, including running applications, displaying documents onscreen, and

263 printing, computers in a client/server network are more specialized. One can opt for client/server when: • have six or more computers. • need strong central security and reliable back up. • need the additional speed and performance.

Protocol The rules and encoding specifications for sending data are known as protocols. The protocols also determine whether the network uses a peer-to-peer or client/server architecture.

In networking, the communication language used by computer devices is called the protocol.

The protocol defines a common set of rules and signals that computers on the network use to communicate. One of the most popular protocols for LANs is called Ethernet. Another popular LAN protocol for PCs is the IBM token-ring network. The classification of computer networks includes the following:

Local Area Network (LAN) Wide Area Network (WAN) Metropolitan Area Network (MAN) Local Area Network (

LAN) LAN stands for Local Area Network which supplies connectivity to a group of computers in close proximity to each other such as in an office building, a school or a home. In simple words,

it is a group of computers and associated devices that share a common communications line or wireless link and typically

share the resources of a single processor or server within a small geographic area For example:

a building, office, department, home, manufacturing plant, or other work sites. Besides operating in a limited space, LANs include a distinctive feature.

LANs are typically owned, controlled, and managed by a single person or organization.

264

A LAN consisting of two or more computers and peripheral devices such as printer, scanner, hard disk drive etc. has software to control all the devices connected to the LAN through special cables like fibre optic or coaxial and a plug-in board to handle the data transmissions. The rate of data transmission usually ranges from 1 Mbit/s to 20 Mbit/s. The length of cable connecting computers to a LAN varies about 1000 feet to several miles depending upon the LAN. The main benefits of this type of network are resource sharing and cost reduction as several computers and users can share the same peripheral devices and programs. The error rate on a LAN is considerably better than other types of networks due to its small size. On the other hand, security and privacy are two major concerns for data transmission. Only authorized users may gain access to the data and transmission of data should be ensured in correct form at its destination on a LAN. LANs are very common in offices but can also connect several offices together. There is no limit as to how many computers are networked for the network to be classified as LAN. Even a simple network of one computer connected to one printer can be considered a LAN.

A local area network may serve as few as two or three users for example, in a home network or as many as thousands of users for example, in an FDDI network.

Most LANs connect workstations and personal computers. Each node (individual computer) in a LAN has its own CPU with which it executes programs, but it is also able to access data and devices anywhere on the LAN.

Usually, the server has applications and data storage that are shared in common by multiple computer users.

This means that many users can share expensive devices, such as laser printers, as well as data. Users can also use the LAN to communicate with each other, by sending e-mail or engaging in chat sessions. There are many different types of LANs Ethernets being the most common for PCs. Most Apple Macintosh networks are based on Apple's AppleTalk network system, which is built into Macintosh computers. LANs have become commonplace in many

265 organizations for providing telecommunications network capabilities that link end users in offices, departments, and other work groups. LANs are distinguished from other kinds of networks by these characteristics: Size: The number of computers connected in a network through some communication channel. Transmission Technology: Devices can be

connected by twisted-pair wire, coaxial cables, or fiber optic cables. Some networks do without connecting transmission media altogether, communicating instead via radio waves. LAN's are capable of transmitting data at very fast rates, much faster than data can be transmitted over a telephone line; but the distances are limited, and there is also a limit on the number of computers that can be attached to a single LAN Topology: The geometric arrangement of devices on the

network. For example, devices can be arranged in a ring or in a straight line. Protocols: The rules and encoding specifications for sending data. The protocols also determine whether the network uses a peer-to-peer or client/server architecture. LAN Hardware The basic hardware devices for establishing a LAN are: 1. Data Transmission Channel: Four

types of communication channels are generally used in a LAN. These may be Coaxial Cable, Twisted Pair Cable, Fibre Optic Cable, Infrared or Radio Waves. 2. Network Interface Circuitry (NIC): This unit in the individual personal computers may be part of the system motherboard or it may be on a separate plug-in card and is connected to the network. It contains the rules and logic to access the LAN.

The function of this circuitry depends on the type of topology used in LAN. 3. Server and Workstations: The server

provides all of the services and information and handles printing requests, for example. The server generally controls all the resources and puts them at the client's disposal. Networks may also consist of more than one server. Each of these servers has a different and

266 more specialized job. A print server handles the print jobs of its workstations and servers. A file server stores or sends files. An application server runs applications and sends the results. A modem server is used to connect other network simply using a telephone wire to share expensive resources by all connected workstations in a network ring. The workstations are the computer users interact with. If the workstation needs something it doesn't have, it can send its request to the server. LAN Software Local Area Networks use special operating systems and other system software to manage the network, its resources and its users. Network Operating Systems should be very secure and robust to provide reliable network services. Examples of some of the popular LAN Operating System Software are: • Windows 2000 Server • Novell NetWare • Ethernet • ArcNet • Corvus Major Local Area Network technologies are: (1) Ethernet (2) Token Ring (3) FDDI Ethernet is by far the most commonly used LAN technology. A number of corporations use the Token Ring technology. FDDI is sometimes used as a backbone LAN interconnecting Ethernet or Token Ring LANs. Another LAN technology, ARCNET, once the most commonly installed LAN technology, is still used in the industrial automation industry. Wide Area Network (WAN) As the term implies, a wide area network spans a large geographic dispersed area, involving a vast array of computers, generally covering a couple miles, sometimes

267 connecting computers thousands of miles apart. A WAN can also be a collection of LANs, bringing together many smaller networks into one large network, so that users and computers in one location can communicate with users and computers in other locations. Many WANs are built for one particular organization and are private. A WAN can constitute a very large corporate or government network, spanning a state, a country, or even the world. Computers connected to a wide-area network are often connected through public networks, such as the telephone system. They can also be connected through high-speed fiber-optics, microwave dishes, leased lines or satellites links. The largest WAN in existence is the Internet. WANs are used for many different purposes. Some are designed as a communications backbone for a large distributed organization. Other WANs focus on particular transaction packages. Many WANs are used to transfer and consolidate corporate data, such as daily transaction summaries from branches. LANs and WANs in general are similar in the sense that both are collections of computers. However, there are huge differences between the LAN and a WAN. Due to the small coverage of areas, LANs are capable of transmitting data at a very high speed with a lower error rate and at low cost as compared to WANs. In case of LAN there is complete control of the owner whereas WAN requires involvement of external authority like telecom department. WAN Hardware In a large geographically dispersed area, to establish links between computers, it requires four basic network devices: Repeaters, Bridges, Routers, and Gateways. Repeater: A repeater connects two segments of network cable. It retimes and regenerates the signals to proper amplitudes and sends them to the other segments. Repeaters require a small amount of time to regenerate the signal. This can cause a propagation delay which can affect network communication when there are several repeaters in a row. Many network architectures limit the number of repeaters that can be used in a row. Repeaters work only at the physical layer of the OSI network model.

268 Bridge: Bridges do not normally allow connection of networks with different architectures and are used to connect two LANs following identical protocols. A bridge reads the outermost section of data on the data packet, to tell where the message is going. It reduces the traffic on other network segments, since it does not send all packets. Bridges can be programmed to reject packets from particular networks. Bridging occurs at the data link layer of the OSI model, which means the bridge cannot read IP addresses, but only the outermost hardware address of the packet. This may be of following types: • Transparent Bridging: They build a table of addresses (bridging table) as they receive packets. If the address is not in the bridging table, the packet is forwarded to all segments other than the one it came from. This type of bridge is used on ethernet networks. • Source route bridging: The source computer provides path information inside the packet. This is used on Token Ring networks. Router: A router is used to route data packets between two networks that may not be similar. It reads the information in each packet to tell where it is going. If it is destined for an immediate network it has access to, it will strip the outer packet, readdress the packet to the proper ethernet address, and transmit it on that network. If it is destined for another network and must be sent to another router, it will re-package the outer packet to be received by the next router and send it to the next router. Routing occurs at the network layer of the OSI model. They can connect networks with different architectures such as Token Ring and Ethernet. Although they can transform information at the data link level, routers cannot transform information from one data format such as TCP/IP to another such as IPX/SPX. Routers do not send broadcast packets or corrupted packets. If the routing table does not indicate the proper address of a packet, the packet is discarded. Gateway: A gateway can translate information between different network data formats or network architectures. It can translate TCP/IP to AppleTalk so computers supporting TCP/IP can communicate with Apple brand computers. Most gateways operate at the application layer, but can operate at the network or session layer of the

269 OSI model. Gateways will start at the lower level and strip information until it gets to the required level and repackage the information and work its way back toward the hardware layer of the OSI model. To confuse issues, when talking about a router that is used to interface to another network, the word gateway is often used. This does not mean the routing machine is a gateway as defined here, although it could be. WAN Connection Technology X.25 Standard Interface: This is a set of protocols developed by the CCITT/ITU (International Telegraph and Telephone Consultative Committee/International Telecommunication Union) which specifies how to connect computer devices over a internetwork. These protocols use a great deal of error checking for use over unreliable telephone lines. Their speed is about 64Kbps. Normally, X.25 is used on packet switching PDNs (Public Data Networks). A line must be leased from the LAN to a PDN to connect to an X.25 network. A PAD (packet assembler/disassembler) or an X.25 interface is used on a computer to connect to the X.25 network. WAN Connections Wide Area Networks (WAN) refers to the technologies used to connect organizations at remote locations. The size of a network is limited due to size and distance constraints. However networks may be connected over a high speed communication links and are usually:

- Dial up connection
- Dedicated connection: It is a permanent full time connection. When a dedicated connection is used, the cable is leased rather than a part of the cable bandwidth and the user has exclusive use.
- Switched network: Several users share the same line or the bandwidth of the line. There are two types of switched networks: Circuit switching: This is a temporary connection between two points such as dial-up or ISDN.

270 Packet switching: This is a connection between multiple points. It breaks data down into small packets to be sent across the network. A virtual circuit can improve performance by establishing a set path for data transmission. This will shave some overhead of a packet switching network. A variant of packet switching is called cell- switching where the data is broken into small cells with a fixed length. Classification of Wide Area Networks WANs can be further divided into two types: Public Networks: These are the networks installed and run by telecom authorities and can be accessed through subscription. Private Networks: These are the networks installed and run to link the locations privately. There is a complete control of the owner and is free to use the network. Private networks can carry large volume of data at a higher data transmission speed. Metropolitan Area Network (MAN) Metropolitan Area Network, in essence a computer network that is meant to span a whole metropolitan area. This network interconnects users with computer resources in a geographic area or region

larger than that covered by a large local area network but smaller than the area covered by a wide area network, such as a city, with dedicated or high-performance hardware. When banks adopted the use of automatic teller machines (ATMs), they were connected within the confines of a city and the nearby countryside. Thus, network of ATMs may be seen as metropolitan area network. It is also used to mean the interconnection of several local area networks by bridging them with backbone lines. Metropolitan Area Networks are usually characterized by very high-speed connections using fiber optical cable or other digital media (wireless infrastructure). Distributed Queue Dual Bus (DQDB), is the Metropolitan Area Network standard for data communication. Using DQDB, networks can be up to 30 miles long and operate at speeds of 34 to 155 Mbit/s. Some technologies used for this purpose are ATM, FDDI and SMDS. These older technologies are in the process of being displaced by 271 Ethernet-based MANs (e.g. Metro Ethernet) in most areas. MAN links between LANs have been built without cables using either microwave, radio, or infrared free space optical communication links. This concept was very popular in the early 1990s when various cities envisaged establishing such networks. In practice these networks are very rare, and today only the concept survives. This is mainly due to the increasingly private providers that develop networks as opposed to cities and councils taking the development of such networks into their own hands.

12.7 Some Other Computer Networks Value Added Network (VAN) A value-added network (VAN) is a specialized application service provider (ASP) that acts as an intermediary between trading partners sharing data or business processes. The provider of such services must process, store and manipulate the data that can be carried on the network (adding value to the data). So these are the public networks that "add value" by transmitting data and by providing access to commercial databases and software. The use of VAN is usually sold by subscription, with users paying for the amount of data they move. VAN can send data between computers in different cities or even different countries. They are often used in electronic data interchange (EDI) systems because they reduce the complexity of connecting to the disparate EDI systems of various trading partners. In this application they collect forms in an electronic mailbox, translate and forward them to recipients, and guarantee they will reach their destinations intact. Other common VAN services include electronic mail, access to stock market data and other public databases, and access to electronic banking and other transaction processing services. VAN is used for a number of reasons. They usually service a given industry and provide value-added services such as data transformation. They can be considered as a way of transmitting computerized data, offering a service similar to what the 272 telephone networks do for telephone calls. At one end a VAN hosts only business-

to-business (B2B) application integration services, hosting general-purpose integration services for any process or industry. At the other end a VAN also hosts

industry- specific

pre-defined integration capabilities and applications (e.g., supply chain order visibility). Traditionally, most VANs primarily only supported general-purpose B2B integration capabilities focused on EDI but these service providers are quickly evolving to become more industry-specific over time, particularly in industries such as retail and hi-tech manufacturing.

Internet The Internet is a global network of networks that use a specific set of protocols. The Internet exploded into the public consciousness in the mid-1990s with the World Wide Web, a hyper-linking of documents located in computers around the world. The documents, described in an easy-to-write language called HTML, can include text, graphics, video clips, audio and links to other documents. Users navigate in this worldwide collection of documents by simple mouse clicks on links or locate them by using a search engine. The World Wide Web with its popular interfaces Netscape Navigator and Microsoft Explorer, breaks down distances, integrates multimedia, enables users to find information, and, equally importantly makes every user a potential publisher. The Web merges computers and communication and transforms every personal computer into a personal communication device.

Intranet An intranet is a private network. It is an implementation of Internet technologies such as TCP/IP and HTTP servers for use within an enterprise and not for connection with the Internet. It may consist of many interlinked local area networks (LAN) and also use leased lines in the wide area network (WAN). Typically, an intranet includes connections through one or more gateway computers to the outside Internet. The main purpose of an intranet is to share company information and computing resources among employees. An intranet can also be used to facilitate working in groups and for teleconferences.

Extranet 273 An extranet is an intranet with a wider range than one organization. It

is a private network that uses the Internet protocols and the public telecommunication system to securely share part of a business'

information or operations with suppliers, vendors, partners, customers, or other businesses. An extranet can be viewed as part of a company's intranet that is extended to users outside the company.

12.8 Summary

Business organizations currently benefit from the use of personal computers. However, many businesses have not taken the next step to increase productivity and reduce office equipment costs by installing a computer network. A network consists of two or more computers, printers, faxes and other electronic devices linked together with cables enabling all of the "pieces" to communicate together to share data and resources. The simple idea behind computer networking is to allow users to access more information and give them access to devices not directly attached to their "local" system, such as printers or storage devices. Three of the main types of computer networks are:

Local Area Network (LAN) Wide Area Network (WAN) Metropolitan Area Network (

MAN) A LAN is comprised of computers located within close proximity, such as in an office or building. A WAN is a network of computers that are situated further apart, but still connected. Such a network might be of computers within a single state agency or of a multi-national corporation worldwide. Man

is a data network designed for a town or city.

In terms of geographic breadth, MAN is larger than LAN, but smaller than WAN. A PC must have a Network Interface Card, frequently called "nics", installed to connect it to a network. The computers can be configured in several ways. The most popular of these "architectures" are peer-to-peer and client/server. The first is a loose connection of local computers with no central authority, and the later relies on a "server" to manage the network and its resources.

274 Computer networks use special operating systems and other system software to manage the network, its resources and its users. Network Operating Systems should be very secure and robust to provide reliable network services. After LAN, WAN and MAN, many other types of network designs also most commonly encounter the network technologies.

Some of them are Value Added Network (established to add value to the data resources), Internet (Network of networks to share world wide data resources), Intranet (Network within an organization to share its data resources by the own company's employees) and Extranet (Network, shared even outside the organization, is an intranet with a wider range than one organization to share part of a business's information with customers, or other businesses). The key benefits of computer networks for business organizations include:

- Increased employee productivity
- Reduced equipment costs
- Backup and protection of data
- Resource Sharing
- Streamlined Business Processes

Some of the limitations to computer networks:

- Sharing capabilities can be difficult to configure initially
- Shared Internet access may not perform satisfactorily
- Network connections drop unexpectedly
- Network is too insecure

12.9 Self-Test questions

1. What is a computer network? 2. What are the major benefits of computer networks? 3. What are the types of network architectures? 4. What are the types of computer networks?

275 5. Differentiate the following: • LAN and WAN • Intranet and Extranet • Routers and Gateways • Peer-to-Peer Architecture and Client/Server Architecture 6. What are Value Added Networks?

12.10 Suggested Readings

1. Black, Uyles, "Computer Networks: Protocols, Standards, and Interfaces", 2 nd Ed., Prentice-Hall India. 2. Norton, Peter, "Introduction to Computers", 2 nd Ed., Tata McGraw Hill. 3. Ram, B., "Computer Fundamentals: Architecture and Organization", 3 rd Ed., New Age Publishers. 4. Sinha, Pradeep K. and Sinha, Priti, Computer Fundamentals, Third Ed., BPB Publications. 5. Tanenbaum, Andrew S., "Computer Networks", 3 rd Ed., Prentice-Hall India.

DATA HIERARCHY AND FILE STRUCTURES Objectives: The motive of this lesson is to present an introduction to File Structure and within that framework the key data hierarchy and file concept, types of files, and different file structures are discussed. Structure: 13.1 Introduction to Data Hierarchy 13.2 Concept of File and its Types 13.3 File Structure 13.4 Basic Terminology to Understand File Structure 13.5 Different File Structures 13.6 Summary 13.7 Self-Test questions 13.8 Suggested Readings 13.1 Introduction to Data Hierarchy

In data processing, storage of data is often conceptualized as data hierarchy. Broadly, a data hierarchy may be considered to be either natural, which arises from the alphabet or syntax of the language in which the information is expressed, or machine, which reflects the facilities of the computer, both hardware and software. A natural data hierarchy might consist of bits, characters, words, phrases, sentences, paragraphs, and chapters. One might use components bound to an application, such as field, record, and file, and these would ordinarily be further specified by having data descriptors such as name field, address field, etc. On the other hand, a machine or software system might use bit, byte, word, block, partition, channel, and port. Programming languages often provide types or objects which can create data hierarchies of arbitrary complexity, thus allowing software system designers to model language structures described by the linguist to greater or lesser degree.

277 The distinction between the natural form of data and the facilities provided by the machine may be obscure, because users force their needs into the molds provided, and programmers change machine designs. As an example, the natural data type "character" and the machine type "byte" are often used interchangeably, because the latter has evolved to meet the need of representing the former. In simple words,

Data Hierarchy refers to the systematic organization of data, often in a hierarchical form. Data organization involves bit, character, fields, records, files and

database and can be shown at the following six levels: Bit (A single binary digit 0 or 1) Character (Combination of many relative bits) Field (Combination of many relative characters) Record (Combination of many relative fields) File

(Combination of many relative records) Database (Combination of many relative files) Data Hierarchy Bit: It is the smallest element of data or a single binary digit, either a 0 or 1. Character: A number of related bits are combined to form a byte or character. Consider a letter P, the number 2, and the special character \$ and can be treated as character. Field: A field (data field) is a meaningful collection of related characters which holds a single fact. It is the smallest logical data entity which is treated as single unit in data processing. Consider a date field, e.g. 21 Sep 1969. This can be treated as a single date field (e.g. birthdates), or 3 fields, namely, day of month, month and year. Record: A record is a collection of related fields. An Employee record may contain a name field(s), address fields, birthdates field and so on.

278 File: An associated group of related records is a file or a file is a collection of related records. Every record in a file has the same set of fields. If there are 500 employees, then each employee would have a record (e.g. called Employee Personal Details record) and the collection of 500 such records would constitute a file (in this case, called Employee Personal Details file). Database: Files are integrated into a database. This is done using a Database Management System. If there are other facets of employee data that we wish to capture, then other files such as Employee Training History file and Employee Work History file could be created as well. In a database, the data from multiple files are integrated in a manner that data redundancy is minimized and made available to all applications. The above structure can be seen in the Hierarchical Model, which is one way to organize data in a database. The above is a view of data seen by a computer user. In terms of data storage, data fields are made of bytes and these in turn are made up of bits.

13.2 Concept of File and its Types The most valuable and visible feature of any computer system to its user is a file as it provides a long term, nonvolatile storage of data. Files stored in the concentric circular tracks of a floppy or hard disk can be referred to as the system of data objects which provide the methods for information storage and retrieval. So file is the basic unit of information storage on a secondary storage device and almost every form of data and information reside on these secondary storage devices in the form of a file whether audio data or video, whether text or binary. However, files may be classified on different bases as follows: 1. On the basis of content: Text files: Files containing data/information in textual form. It is merely a collection of characters. e.g. Document files etc. Binary files: Files containing machine code. The contents are non- recognizable and can be interpreted only in a specified way using the same

279 application that created it. e.g. executable program files, audio files, video files etc. 2. On the basis of direction of data flow: Input files: Files that supply data/information to some application or program. In this role it behaves as data/information producer. Output files: Files that receive data/information from some application or program. In this role it behaves as data/information consumer. 3. On the basis of frequency of updation: Master files: Files that contain data/information which change very infrequently. Transaction files: Files that contain data/information which change very frequently. 4. On the basis of role of the contents of the file: Data file: Files that contain data/information participating in processing. Meta file: Files that contain data/information about other files. E.g. index files etc. 13.3 File Structure The terminology of file structures has evolved higgledy-piggledy without much concern for consistency, ambiguity, or whether it was possible to make the kind of distinctions that were important. It was only much later that the need for a

well- defined,

unambiguous language to describe file structures became apparent. In particular, there arose a need to communicate ideas about file structures without getting bogged down by hardware considerations so that user may be able to:

- use data files in general application programs.
- write programs that will write data within a program into a file.
- explain the structure of a file created in programs.
- determine file size as a function of measurement precision and number of data points.

The structure of a data file defines how records, or rows of data, are related to fields, or columns of data. The most common data structures are:

280 • Rectangular or Logical Record • Card Image or Multiple Record • Hierarchical • Relational Rectangular or Logical Record File: In a rectangular file, each physical record or line of data is the same length. This is also referred to as an LRECL (logical record length) file because the length of the logical record (all data for a given record) is equal to the length of the physical record (one line of data). In other words, the data for each case or unit of analysis are stored in a single physical record. Card Image or Multiple Record File: In a card image file, all of the data for a particular case are contained on more than one physical record or line of data. This format is a remnant of the time when data were input on punch cards that were physically limited to 80 characters per card. Therefore, if a line of data was longer than 80 characters, it had to be entered on multiple cards. Each card, or "deck," was numbered and stored in numerical sequence. With this format multiple records have to be read to build a complete case. Hierarchical File: A hierarchical file contains multiple units of analysis, such as a household and an individual, that are related to one another through a structural hierarchy. Each unit of analysis has its own record structure or record type. The records for each unit of analysis are not necessarily the same length. When reading multiple levels into a statistical package at one time, cases at the lowest level of the hierarchy will define the unit of analysis. Relational File: A relational file contains multiple units of analysis that are related to one another based on a predefined structure or variable, but not necessarily a hierarchy. For example, a data collection could have different record types for households, families, persons, wages, and general income, but these record types are not related to each other by a hierarchy of membership.

13.4 Basic terminology to Understand File Structure

281 Given a set of 'attributes' A and a set of 'values' V , then a record R is a subset of the Cartesian product $A \times V$ in which each attribute has one and only one value. Thus R is a set of ordered pairs of the form (an attribute, its value). For example, the record for a document which has been processed by an automatic content analysis algorithm would be $R = \{(K_1, x_1), (K_2, x_2), \dots, (K_m, x_m)\}$ The K_i '

s are keywords functioning as attributes and the value x_i can be thought of as a numerical weight. Frequently documents are simply characterized by the absence or presence of keywords, in which case $R = \{K_{t1}, K_{t2}, \dots, K_{ti}\}$ where K_{ti} is present if $x_{ti} = 1$

and is absent otherwise. Records are collected into logical units called Files. They enable one to refer to a set of records by name, the file name. The records within a file are often organized according to relationships between the records. This logical organization has become a file structure (or data structure). It is difficult in describing file structures to keep the logical features separate from the physical ones. The latter are characteristics forced upon by the recording media (e.g. tape, disk). Some features can be defined abstractly (with little gain) but are more easily understood when illustrated concretely. One such feature is a field. In any implementation of a record, the attribute values are usually positional, that is the identity of an attribute is given by the position of its attribute value within the record. Therefore the data within a record is registered sequentially and has a definite beginning and end. The record is said to be divided into fields and the n th field carries the n th attribute value. Pictorially an example of a record with associated fields in Figure 13.1 is given below.

282 Figure 13.1: An example of a record with associated field The fields are not necessarily constant in length. To find the value of the attribute K_4 , first find the address of the record R (which is actually the address of the start of the record) and read the data in the 4th field. In the same picture it has also been shown some fields labeled P_i . They are addresses of other records, and are commonly called pointers. Now it has extended the definition of a record to a set of attribute-value pairs and pointers. Each pointer is usually associated with a particular attribute-value pair. For example, (see Figure 13.2) pointers could be used to link all records for which the value x_1 (of attribute K_1) is a , similarly for x_2 equal to b , etc. Figure 13.2: A demonstration of the use of pointers to link record To indicate that a record is the last record pointed to in a list of records use the null pointer [[logical end]]. The pointer associated with attribute K in record R will be called a K -pointer. An attribute (keyword) that is used in this way to organize a file is called a key.

283 To unify the discussion of file structures some further concepts are needed. Define a list L of records with respect to a keyword K , or more briefly a K -list as a set of records containing K such that: 1) the K -pointers are distinct; 2) each non-null K -pointer in L gives the address of a record within L ; 3) there is a unique record in L not pointed to by any record containing K ; it is called the beginning of the list; and 4) there is a unique record in L containing the null K -pointer; it is the end of the list. From previous example: K_1 -list : R_1, R_2, R_5 K_2 -list : R_2, R_4 K_4 -list : R_1, R_2, R_3 Finally, the definition of a directory of a file is needed. Let F be a file whose records contain just m different keywords K_1, K_2, \dots, K_m . Let n_i be the number of records containing the keyword K_i , and h_i be the number of K_i -lists in F . Furthermore, denote

a_{ij}

the beginning address of the j th K_i -list. Then the directory is the set of sequences ($n_i, h_i, a_{i1}, a_{i2}, \dots, a_{ih_i}$) $i = 1, 2, \dots$

m

13.5 Different File Structures Sequential files A sequential file is the most primitive of all file structures. It has no directory and no linking pointers. The records are generally organized in lexicographic order on the value of some key. In other words, a particular attribute is chosen whose value will determine the order of the records. Sometimes when the attribute value is constant for a large number of records a second key is chosen to give an order when the first key fails to discriminate. The implementation of this file structure requires the use of a sorting routine. Its main advantages are: (1) It is easy to implement; (2) It provides fast access to the next record using lexicographic order.

284 Its disadvantages: (1) It is difficult to update—inserting a new record may require moving a large proportion of the file; (2) Random access is extremely slow. Sometimes a file is considered to be sequentially organized despite the fact that it is not ordered according to any key. Perhaps the date of acquisition is considered to be the key value, the newest entries are added to the end of the file and therefore pose no difficulty to updating. Inverted list An inverted list is a file structure in which every list contains only one record. Remember that a list is defined with respect to a keyword K , so every K -list contains only one record. This implies that the directory will be such that $n_i = h_i$ for all i , that is, the number of records containing K_i will equal the number of K_i -lists. So the directory will have an address for each record containing K_i . For document retrieval this means that given a keyword one can immediately locate the addresses of all the documents containing that keyword. For the previous example let us assume that a non-black entry in the field corresponding to an attribute indicates the presence of a keyword and a black entry its absence. Then the directory will point to the file in the way shown in Figure 13.3. The definition of an inverted file does not require that the addresses in the directory are in any order. However, to facilitate operations such as conjunction ('and') and disjunction ('or') on any two inverted lists, the addresses are normally kept in record number order. This means that 'and' and 'or' operations can be performed with one pass through both lists. The penalty we pay is of course that the inverted file becomes slower to update.

285 Figure 13.3: An inverted file Index-sequential files An index-sequential file is an inverted file in which for every keyword K_i , $n_i = h_i = 1$ and $a_{i1} > a_{i2} > \dots > a_{im}$. This situation can only arise if each record has just one unique keyword, or one unique attribute-value. In practice therefore, this set of records may be ordered sequentially by a key. Each key value appears in the directory with the associated address of its record. An obvious interpretation of a key of this kind would be the record number. In this example none of the attributes would do the job except the record number. Diagrammatically the index-sequential file would therefore appear as shown in Figure 13.4. It has deliberately been written R_i instead of K_i to emphasize the nature of the key.

286 Figure 13.4: index sequential file In the literature an index-sequential file is usually thought of as a sequential file with a hierarchy of indices. This does not contradict the previous definition; it merely describes the way in which the directory is implemented. It is not surprising therefore that the indexes ('index' = 'directory' here) are often oriented to the characteristics of the storage medium. For example (see Figure 13.5) there might be three levels of indexing: track, cylinder and master. Each entry in the track index will contain enough information to locate the start of the track, and the key of the last record in the track, which is also normally the highest value on that track. There is a track index for each cylinder. Each entry in the cylinder index gives the last record on each cylinder and the address of the track index for that cylinder. If the cylinder index itself is

287 stored on tracks, then the master index will give the highest key referenced for each track of the cylinder index and the starting address of that track. No mention has been made of the possibility of overflow during an updating process. Normally provision is made in the directory to administer an overflow area. This of course increases the number of bookkeeping entries in each entry of the index. Figure 13.5: An example of an implementation of an index-sequential file.

288 Multi-lists A multi-list is really only a slightly modified inverted file. There is one list per keyword, i.e. $h_i = 1$. The records containing a particular keyword K_i are chained together to form the K_i -list and the start of the K_i -list are given in the directory, as illustrated in Figure 13.6. Since there is no K_3 -list, the field reserved for its pointer could well have been omitted. Figure 13.6: A multi-list So could any blank pointer field, so long as no ambiguity arises as to which pointer belongs to which keyword. One way of ensuring this, particularly if the data values (attribute-values) are fixed format, is to have the pointer not pointing to the beginning of the record but pointing to the location of the next pointer in the chain.

289 The multi-list is designed to overcome the difficulties of updating an inverted file. The addresses in the directory of an inverted file are normally kept in record-number order. But, when the time comes to add a new record to the file, this sequence must be maintained, and inserting the new address can be expensive. No such problem arises with the multi-list, one updates the appropriate K-lists by simply chaining in the new record. The penalty one pay for this is of course the increase in search time. This is in fact typical of many of the file structures. Inherent in their design is a trade-off between search time and update time. Cellular multi-lists A further modification of the multi-list is inspired by the fact that many storage media are divided into pages, which can be retrieved one at a time. A K-list may cross several page boundaries, which means that several pages may have to be accessed to retrieve one record. A modified multi-list structure which avoids this is called a cellular multi-list. The K-lists are limited so that they will not cross the page (cell) boundaries. At this point the full power of the notation introduced before comes into play. The directory for a cellular multi-list will be the set of sequences (

$K_i, n_i, h_i, a_{i1}, \dots, a_{ih_i}$) $i = 1, 2, \dots,$

m Where the h_i have been picked to ensure that a K_i -list does not cross a page boundary. In an implementation, just as in the implementation of an index-sequential file, further information will be stored with each address to enable the right page to be located for each key value. Ring structures A ring is simply a linear list that closes upon itself. In terms of the definition of a K-list, the beginning and end of the list are the same record. This data-structure is particularly useful to show classification of data.

290 Figure 13.7: A dendrogram Let us suppose that a set of documents $\{D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8\}$ has been classified into four groups, that is $\{(D_1, D_2), (D_3, D_4), (D_5, D_6), (D_7, D_8)\}$ Furthermore these have themselves been classified into two groups, $\{(D_1, D_2), (D_3, D_4)\}, \{(D_5, D_6), (D_7, D_8)\}$

The dendrogram for this structure would be that given in Figure 13.7. To represent this in storage by means of ring structures is now a simple matter (see Figure 13.8). The

D_i indicates a description (representation) of a document. Notice how the rings at a lower level are contained in those at a higher level. The field marked C_i normally contains some identifying information with respect to the ring it subsumes. For example, C_1 in some way identifies the class of documents $\{D_1, D_2\}$.

291 Figure 13.8: An implementation of a dendrogram vis ring structure Were we to group documents according to the keywords they shared, then for each keyword we would have a group of documents, namely, those which had that keyword in common. C_i would then be the field containing the keyword uniting that particular group. The rings would of course overlap (Figure 19.9), as in this example: $D_1 = \{K_1, K_2\}$ $D_2 = \{K_2, K_3\}$ $D_3 = \{K_1, K_4\}$ Figure 13.9: Two overlapping rings The usefulness of this kind of structure will become more apparent when to discuss searching of classifications. If each ring has associated with it a record, which

292 contains identifying information for its members, then, a search strategy searching a structure such as this will first look at C_i (or

K_i

in the second example) to determine whether to proceed or abandon the search. 13.6 Summary

Data Hierarchy refers to the systematic organization of data, often in a hierarchical form. Data organization involves bit, character, fields, records, files and

database. In terms of data storage, data fields are made of bytes and these in turn are made up of bits. File is the basic unit of information storage on a secondary storage device and almost every form of data and information reside on these secondary storage devices in the form of a file whether audio data or video, whether text or binary. The structure of a data file defines how records, or rows of data, are related to fields, or columns of data. The most common data structures are:

- Rectangular or Logical Record
- Card Image or Multiple Record
- Hierarchical
- Relational Records

are collected into logical units called Files. They enable one to refer to a set of records by name, the file name. The records within a file are often organized according to relationships between the records. This logical organization has become a file structure. A sequential file is the most primitive of all file structures. It has no directory and no linking pointers. The records are generally organized in lexicographic order on the value of some key. An inverted list is a file structure in which every list contains only one record. An index-sequential file is an inverted file in which for every keyword K_i , $n_i = h_i = 1$ and a 11 > ; a 21 . . . > ; a m1 . This situation can only arise if each record has just one unique keyword, or one unique attribute-value. A multi-list is really only a slightly modified inverted file. There is one list per keyword, i.e. $h_i = 1$. The records containing a particular keyword K_i are chained

293 together to form the K_i -list and the start of the K_i -list are given in the directory. A further modification of the multi-list is inspired by the fact that many storage media are divided into pages, which can be retrieved one at a time. A ring is simply a linear list that closes upon itself. In terms of the definition of a K- list, the beginning and end of the list are the same record. This data-structure is particularly useful to show classification of data.

13.7 Self-Test questions

- 1) What do you understand by data storage hierarchy?
- 2) What is a file?
- 3) What are the types of files?
- 4) What do you understand by File Structure?
- 5) Explain different types of file structures.
- 6) What is the difference between Inverted File Structure and Multilist file structure?
- 7) What is a ring structure? Define it with example.

13.8 Suggested Readings

- 1) Data Structures through C Language, First Ed., Excel Books.
- 2) Salaria, Anita, Data Structures & Algorithms Using C, Second Ed., Khanna Book Publishing Co. (P) Ltd.
- 3) Singh, Bhagat, and Naps, Thomas L., Introduction to Data Structures, First Ed., Galgotia Booksources P. Ltd.
- 4) Sinha, Pradeep K. and Sinha, Priti, Computer Fundamentals, Third Ed., BPB Publications.

294

UNIT 14 -

NETWORK TOPOLOGY AND APPLICATIONS Objectives: The motive of this lesson is to present an introduction to network topologies and within that framework the key network topology concepts, types, benefits and limitations of network topologies are discussed. Applications of computer networks are also tried to bring to the fore. Structure:

14.1 Introduction to Network Topology 14.2 Defining Network Topology 14.3 Physical Topologies 14.4 Logical Topologies 14.5 Network Protocols 14.6 Applications of Computer Networks 14.7 Summary 14.8 Self-Test questions 14.9 Suggested Readings

14.1 Introduction to Network Topology The term topology derived from Greek word topos, which means a place, is a description of any kind of locality in terms of its layout. The word topology is commonly used to discuss the properties of various types of networks. Topology is the branch of mathematics that examines the characteristics of geometric shapes. One can think of a topology as a network's "shape" and the shape a network takes has much to do with the way it functions.

This shape does not necessarily correspond to the actual physical layout of the devices on the network. For example, the computers on a home LAN may be arranged in a circle, but it would be highly unlikely to find an actual ring topology there. Most of the networking softwares support all topologies.

295 A network topology describes the configuration of a network i.e. how the network components are connected together. Topology also refers to the way in which the network of computers is connected. Each topology is suited to specific tasks and has its own advantages and disadvantages.

14.2 Defining Network Topology The geometrical arrangement of positioning the computers in the form of a network for communication is referred to as network topology. In communication networks, a topology is a usually schematic description of the arrangement of a network, including its nodes and connecting lines. There are two ways of defining network geometry:

Physical topology: If one can see it and touch it, it is physical. Logical (or signal) topology: If one cannot see it or touch it, it is logical. The physical topology of a network is the layout or actual appearance of the cabling scheme used on a network whereas logical topology of a network describes how the data flows through the physical topology. Similar looking networks can have quite different logical topologies. The physical topologies of networks are the actual geometric layout of workstations. The choice of topology is dependent upon:

- type and number of equipment being used
- planned applications and rate of data transfers
- required response times
- cost

14.3 Physical Topologies Types of Network

Topology Network topologies are categorized into the following basic types:

- Bus Topology

296 • Ring Topology • Star Topology • Tree Topology • Mesh Topology

More complex networks can be built as hybrids of two or more of the above basic topologies. Bus Topology

The

bus topology (not to be confused with the system bus of a computer) connects a number of workstations using a common backbone to connect all devices. A single cable, the backbone functions as a shared communication medium, attaches or taps into with an interface connector with the devices. The medium has a physical beginning and end. All buses are implemented using electrical cable, usually coaxial, and the ends of the cable must be terminated with a terminating resistor that matches the impedance of the cable. The terminating resistor prevents data reflections from coming across as data corruption. The bus is considered a multipoint system because all devices tap into the same backbone cable. An important characteristic to remember in bus topologies is that all data signals are broadcast throughout the bus structure. A device wanting to communicate with another device on the network sends a broadcast message onto the wire that all other devices see, but only the intended recipient actually accepts and processes the message. In simple words, the bus cable carries the transmitted message along the cable. As the message arrives at each workstation, the workstation computer checks the destination address contained in the message to see if it matches it's own. If the address does not match, the workstation does nothing more. If the address of the workstation matches that contained in the message, the workstation processes the message. The message is transmitted along the cable and is visible to all computers connected to that cable. Computers on a bus network send their data packets to all the computers, even if the data packet is for a specific computer only. Data Packets addressed to one particular computer 297 are received by that computer; all other computers delete the packet upon reception. In the following diagram, if second node sends a signal to the last node, the signal propagates for the length of the cable and is seen by all the nodes as well. It is necessary to have an addressing mechanism so that each node understands which messages it is to receive and which to ignore. In this topology network messages are sent to each terminal on the network so if one of the terminals breaks then the others are not affected. The cabling cost of bus systems is the least of all the different topologies. The common implementation of this topology is Ethernet. A message transmitted by one workstation is heard by all the other workstations. Examples of Bus topology The following networks are examples of bus topology: • ARCnet, (Token bus) • Ethernet, (10Base2) Advantages of Bus topology • Bus topology has the following advantage: • Cabling costs are minimized because of the common trunk. • Easy to implement. • A relatively cheap and reliable topology. Disadvantages of Bus topology Disadvantages of bus topology are as follows: • Difficult to trouble shoot because no central distribution points exist. • Limits on cable length and Workstation numbers. 298 • Cable breaks can disable the entire segment because they remove the required termination from each of the two cable fragments. • As the number of workstations increase, the speed of the network slows down. Ring Topology The ring topology is a physical, closed loop configuration consisting of point-to-point links. It is a topology of computer networks where each user is connected to two other users, so as to create a ring. Ring network systems use 4 pair cables (separate send/receive). A failure in any cable breaks the loop and can take down the entire network. In a ring network, every device has exactly two workstations for communication purposes. Adjacent pairs of workstations are directly connected. All messages travel through a ring in the same direction effectively either "clockwise" or "counterclockwise", each node passing on the data to the next node till it reaches its destination. However, a ring may be unidirectional or bi-directional. In case of unidirectional ring network, data moves only in one direction and if any node breaks down, specialized software is required for the proper functioning of the network. Whereas data moves in both the direction in bi-directional ring networks, but only one direction at a single time and if any node breaks down, a message can usually be sent in the opposite direction still allowing the node to communicate with the other workstations of the active network. Information travels around the ring from one workstation to the next. Each packet of data sent on the ring is prefixed by the address of the station to which it is being sent to. When a data packet arrives, the workstation checks to see if the packet address is the same as its own. If it is, it grabs the data in the packet. If the packet does not belong to it, it sends the packet to the next workstation in the ring. Ring networks tend to be inefficient when compared to client/server networks because data must travel through more points before reaching its destination. For example, if a given ring network has ten computers on it, to get from computer one to computer five,

299 data must travel from computer one, through computers two, three and four, and to its destination at computer five. It could also go from computer one through ten, nine, eight, seven, and six, until reaching five, but this method is slower because it travels through more computers. In the diagram, it can be seen how each node on the ring acts as a repeater. It receives a transmission from the previous node and amplifies it before passing it on. Faulty workstations can be isolated from the ring. When the workstation is powered on, it connects itself into the ring. When power is off, it disconnects itself from the ring and allows the information to bypass the workstation. The ring topology may prove optimum when system requirements are modest and workstations are at scattered points. Rings are found in some office buildings or school campuses. Examples of Ring topology The following are examples of ring topologies: • IBM Token Ring (although wired as a star) • Fiber Distributed Data Interface (FDDI) Advantages of Ring topology • Growth of the system has minimal impact on performance • All stations have equal access • Each node on the ring acts as a repeater, allowing ring networks to span greater distances than other physical topologies. • Because data travels in one direction high speeds of transmission of data are possible

300 • When using a coaxial cable to create a ring network the service becomes much faster. Disadvantages of Ring topology • Often the most expensive topology • Failure of one computer may impact others Star Topology Star network is one of the most common computer network topologies. In its simplest form, star network consists of one central point (a special junction box) called a "hub" or hub computer which acts as a router to transmit messages and through which all components are connected. In a computer network, the central hub is the host computer, and at the end of each connection is a terminal. The star network topology works well when workstations are at scattered points. It is easy to add or remove workstations. This type of topology is frequently used to connect terminals to a large time-sharing host computer. One of the examples may be banks typically using this type of topology to control and coordinate several of its other branch offices. A star network uses a significant amount of cable (each terminal is wired back to the central hub, even if two terminals are side by side several hundred meters away from the host). All routing decisions are made by the central hub, and all other workstations can be simple. In a star network, a cable failure will isolate the workstation that it links to the central computer, but only that workstation will be isolated. All the other workstations will continue to function normally, except that they will not be able to communicate with the isolated workstation. If any workstation goes down, none of the other workstations will be affected. But if the central computer goes down, the entire network will suffer degraded performance or complete failure.

301 Examples of Star topology The following types of networks are examples of star topologies: • ARCnet • 10Base-T, 100Base-TX • StarLAN Advantages of Star Topology • Most modern cabling systems are designed in a star physical topology. The advantages of the star topology are many, including the following: • Easy to implement and extend, even in large networks • Well suited for temporary networks (quick setup) • The failure of a non central node will not have major effects on the functionality of the network. However, the big advantage of the star network is that it is fast. This is because each computer terminal is attached directly to the central computer. Disadvantages of Star topology Star topology has the following disadvantages: • Hub failures can disable large segments of the network. • Limited cable length and number of stations. • Very expensive due to use of a lot of cable. • Maintenance costs may be higher in the long run. • Failure of the node can disable the entire network. • One virus in the network will affect them all.

302 Tree Topology In Tree network topology all the nodes are connected in a hierarchical fashion where each branch with multiple nodes, demands a flow of information. All the branches may be assigned different types of jobs to perform. If any of the branch stops functioning, no other branch may be affected in this type of topology. Advantages of Tree Topology The advantages of tree topology include the following: • Easy to create functional grouping at branch level i.e. separate branches may perform different jobs assigned at their end. Disadvantages of Tree topology Tree topology has the following disadvantage: • Rate of data transmission through this network is very slow. Mesh Network Topology The mesh topology connects all computers to each other. It is a way to route data, voice and instructions between nodes. It allows for continuous connections and reconfiguration around blocked paths by "hopping" from node to node until a connection can be established. Mesh networks are self-healing which means that the network can still operate even when a node breaks down or a connection goes bad. As a result, a very reliable network is formed. A mesh network is a networking technique which allows inexpensive peer network nodes to supply back haul services to other nodes in the same network. A mesh network effectively extends a network by sharing access to higher cost network infrastructure. The cable requirements are high, but there are redundant paths built in. Any failure of one computer allows all others to continue, as they have alternative paths to other computers. The mesh network topology employs either of two schemes, called full mesh topology or partial mesh topology. In the full mesh topology, each node (workstation or other device) is connected directly to each of the others. In the partial mesh topology, some nodes are

303 connected to all the others, but some of the nodes are connected only to those other nodes with which they exchange the most data. The illustration shows a full mesh network with five nodes. Each node is shown as a sphere, and connections are shown as straight lines. The connections can be wired or wireless. Advantages of Mesh topology Mesh topology has the following advantage: • A mesh network is reliable and offers redundancy. • If one node can no longer operate, all the rest can still communicate with each other, directly or through one or more intermediate nodes. • Mesh networks work well when the nodes are located at scattered points that do not lie near a common line. Disadvantages of Mesh topology Disadvantages of mesh topology are as follows: • The chief drawback of the mesh topology is expense, because of the large number of cables and connections required. Hybrid Topology Different network configurations have their own merits and demerits. In real sense, one pure network topology is rarely used in organizations. Instead, an organization may use a

304 combination of two or more different topologies like a star-bus hybrid, star-ring network with connections between various computers on the network. This scheme of combining multiple topologies (star, bus and/or ring) into one large topology is known as hybrid topology. The hybrid network is common in large wide-area networks. Because each topology has its own strengths and weaknesses, several different types can be combined for maximum effectiveness.

Examples of Hybrid topology An example of a hybrid topology is as follows: A company can place its accounting database users on a ring for better throughput, and its secretarial staff on a bus for ease of cabling. Advantages of Hybrid topology Hybrid topology has the following advantages: • One company can combine the benefits of several different types of topologies. • Workgroup efficiency and traffic can be customized. Disadvantages of Hybrid topology The disadvantages of hybrid topology include the following: • Devices on one topology cannot be placed into another topology without some hardware changes.

14.4 Logical Topologies Logical topologies have the same names as physical topologies, but the logical topologies describe the network from the viewpoint of the data traveling on the network. The following four common types of networks can have different logical topologies: • Ethernet • Token Ring • ARCnet 305 • FDDI Ethernet An older, common wiring system for Ethernet (10Base2) and (10Base5) uses coaxial cable in a linear bus topology. In the most common type of Ethernet, each node connects to the coax through a T-connector (BNC) that taps into the signals on the coaxial cable. The nodes both transmit and receive through the same connector. Therefore, 10Base2 Ethernet is a logical as well as a physical bus. A newer variation of Ethernet, 10Base-T and 100Base-TX, are cabled using wiring hubs (concentrators). Each station is connected to the hub via an individual UTP twisted pair cable. Within the hub, however, the individual signals are combined into a bus. Therefore 10Base-T and 100Base-TX are physical stars, but logical buses. Token Ring By examining the wiring of a Token Ring, it can be seen that it meets all the specifications of a star. Token Ring uses central wiring hubs, and each node is wired to the hub with an individual run of cable. Starting at the hub, the signal travels through a pair of wires to the receive circuit on the node's network interface. The receive circuit passes the signal to the transmit circuit, which repeats the signal on a separate pair of wires and sends the signal back to the hub. If follow the signal around the entire network, one can see that it completes a circuitous path, proving that Token Ring has a ring logical topology. Token Ring is wired in a physical star to obtain the advantages of a central wiring hub. All stations can be connected and disconnected at a central point, and the wiring hub can be equipped with hub management and diagnostic systems. Therefore Token Ring sometimes is referred to as a star-wired-ring. ARCnet

306 ARCnet is wired in a star using coaxial cable. That ARCnet is a logical bus also can be determined by examining the network cards. They both receive and transmit through the same coax connector. ARCnet is wired as a physical star, and has some of the advantages of a star network. However, ARCnet functions as a logical bus. Because of its signaling characteristics, ARCnet is also called a token-passing-bus. FDDI Fiber Distributed Data Interface (FDDI) can be configured as both a physical and a logical ring. FDDI is commonly used to connect widely dispersed areas with a high-speed fiber network. When doing so, a physical ring is the most cost-effective cabling plan. FDDI uses the same signaling mechanism as Token Ring; each node repeats the incoming signal and transmits it to the next node in the ring. 14.5 Network Protocols Protocols define the rules for sending data from one point to another and are used in different network topologies. Some of them are the following: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) This protocol is commonly used in bus (Ethernet) implementations. Multiple access refers to the fact that in bus systems, each station has access to the common cable. Carrier sense refers to the fact that each station listens to see if no other station is transmitting before sending data. Collision detection refers to the principle of listening to see if other stations are transmitting whilst we are transmitting. Token Ring This protocol is widely used in ring networks for controlling workstation access to the ring. A short message called a token is circulated around the ring, being passed from one

307 node to another node which is originated from a controller or master station which inserts it onto the ring. Polling This protocol uses a central node to monitor other nodes in the network. It is used predominantly in star networks. The master station sends a message to each slave station in turn. If the slave station has information to send, it passes it to the master station. The master station then sends the information to the desired slave station, or keeps it if the information is for itself. Slave stations are very simple and have limited functionality. It is appropriate for industrial and process control situations

14.6 Applications of Computer Networks

The following are the major applications of computer networks:

Central Data Back-Up and Security: Computer Networks allow an easy central data back-up to protect important files and security for important data. In this way, one can secure access to authorized users only and insure that important files cannot be accidentally deleted, moved or changed. By implementing fault tolerance on network, one can minimize network down time in the event of a hardware failure.

File sharing: File and print sharing is the biggest advantage of having a network. Instead of "walking" files back-and-forth between computers, employees can share files without leaving their desks. This will increase communication and make everyone in office more productive.

Applications Sharing: Sharing applications involves storing software applications on a file server. There will be better software version control because all users will be sharing the same application. In addition, software upgrades are installed only once on the server, not multiple times on each machine. Storing applications on a file server can also reduce software costs. Instead of purchasing one copy of every software application for each

308 computer in office, only one copy of the application may be purchased and stored on the file server.

Printer and peripheral sharing: Networks allow sharing expensive peripherals such as printers, plotters and scanners. Instead of purchasing a printer for each computer, one printer can be shared among several users. Other devices can be shared as well including CD-ROM's and fax modems.

Fax sharing: A network allows everyone in office to share one or more fax modems. The ability to send documents via a fax machine instead of by mail or courier reduces operating costs by saving money on stamps, paper etc. Fax modems also save time.

CD-ROM servers sharing: If a company has substantial amount of information stored on CD-ROM's, a CD-ROM server will allow to easily sharing this information with all users on the network.

Telecommuting and Remote Access: By setting up Remote Access to a network, allow employees to quickly connect to the network at any time from any location. If a person travels with a laptop or has a computer at home, he can check e-mail, download files or access a database from the road or at home.

Wide Area Networks: A WAN can be setup between office and its other branch or main offices. This enables remote offices to synchronize data, transfer and back-up important files, exchange E-mail, etc. Many different WAN technologies are available dependent on the companies bandwidth needs.

E-Mail and Internet Access: The Internet has become the largest information database in the world. E-mail is also a facility that provides a quick and easy way for persons to communicate with each other. This information and knowledge gained through internet access can keep the companies informed and competitive in today's marketplace.

309 **Creating an Intranet or Internet Website:** The difference between an Intranet and an Internet Website is that an intranet is located on a privately owned network and is usually only accessed by local network users whereas an Internet Website is a public site that can be accessed by anyone with access to the Internet. Intranets provide company with an "on-line bulletin board". Employees can easily post information that can be accessed with a Browser. An Internet Website allows providing products and services to the customers. A simple Website can be "on-line brochure" allowing reaching a large number of customers at a very low advertising rate. The company becomes "virtually available" to customers 24 hours a day.

Summary

Topology, the geometric arrangement of positioning the computers, remains an important part of network design theory. One can probably build a small network without understanding the difference between a bus design, a star design or a ring design, but understanding the concepts behind these gives a deeper understanding of important aspects like fast data transmission, cost effectiveness, reliability, optimum solution for transferring data etc.

In communication networks, a topology is a usually schematic description of the arrangement of a network, including its nodes and connecting lines. There are two ways of defining network geometry.

Physical topology: If one can see it and touch it, it is physical.

Logical (or signal) topology: If one cannot see it or touch it, it is logical. The physical topology may be classified in to a number of types. The bus topology connects a number of workstations using a common backbone to connect all devices. The ring topology is a physical, closed loop configuration consisting of point-to-point links. It is a topology of computer networks where each user is onnected to two other users, so as to create a ring. All messages travel through a ring in the same direction effectively either "clockwise" or "counterclockwise", each node passing on the data to the next node till it

310 reaches its destination. Star network consists of one central point called a "hub" which acts as a router to transmit messages and through which all components are connected. In Tree network topology all the nodes are connected in a hierarchical fashion where each branch with multiple nodes, demands a flow of information. A mesh network is a networking technique which allows inexpensive peer network nodes to supply back haul services to other nodes in the same network. It allows for continuous connections and reconfiguration around blocked paths by "hopping" from node to node until a connection can be established. The scheme of combining multiple topologies into one large topology is known as hybrid topology. The hybrid network is common in large wide-area networks. Logical topologies have the same names as physical topologies, but the logical topologies describe the network from the viewpoint of the data traveling on the network. The following four common types of networks can have different logical topologies: • Ethernet • Token Ring • ARCnet • FDDI Protocols define the rules for sending data from one point to another and are used in different network topologies. Some of them are the following: • Carrier Sense Multiple Access with Collision Detection (CSMA/CD) • Token Ring • Polling Some of the applications of computer networks are: • Central Data Back-Up and Security • File sharing • Applications Sharing • Printer and peripheral sharing • Fax sharing • CD-ROM servers sharing • Telecommuting and Remote Access

311 • Wide Area Networks • E-Mail and Internet Access • Creating an Intranet or Internet Website 14.9 Self-Test questions 1. What do you understand by a network topology? 2. What are the major types of network topology? 3. What are the key benefits of star networks? 4. What are the merits and demerits of mesh topology? 5. Differentiate the following: • Star and Bus • Mesh and Hybrid • Token Ring and Ethernet • Physical topology and Logical Topology 6. What are the major applications of Computer Networks? 14.10 Suggested Readings 1. Black, Uyles, "Computer Networks: Protocols, Standards, and Interfaces", 2 nd Ed., Prentice-Hall India. 2. Norton, Peter, "Introduction to Computers", 2 nd Ed., Tata McGraw Hill. 3. Ram, B., "Computer Fundamentals: Architecture and Organization", 3 rd Ed., New Age Publishers. 4. Sinha, Pradeep K. and Sinha, Priti, Computer Fundamentals, Third Ed., BPB Publications. 5. Tanenbaum, Andrew S., "Computer Networks", 3 rd Ed., Prentice-Hall India.

312

UNIT 15 -

INTERNET EXPLORER: AN OVERVIEW Objectives: The motive of this lesson is to present an introduction to Internet Explorer and within that framework the key browser concepts, Internet Explorer environment, information downloading, printing and other options of Internet Explorer are discussed. Structure: 15.1 Introduction to Web Browser 15.2 Defining Internet Explorer 15.3 Features of Internet Explorer 15.4 Internet Explorer Environment 15.5 Working with Internet Explorer 15.6 Downloading, Emailing, and Printing Web Documents for Later Use 15.7 Customizing Internet Explorer Options 15.8 Summary 15.9 Self-Test questions 15.10 Suggested Readings 15.1 Introduction to Web Browser In recent years, users have come to expect a lot more from Internet, one of the most popular means of communication. A state-of-the-art web browser provides a full multimedia experience, complex with pictures, sound, video and even 3-D imaging. It even provides secure connections through encryption and much more. Browsers are the "window to the Web" for Internet user around the world. A browser is an application program that provides a way to look at and interact with all the information on the World Wide Web. The word "browser" seems to have originated prior to the Web as a generic term for user interfaces that help to browse (navigate through and read) text files online.

313 Because it is an internet service, the World Wide Web is based on clients and servers, a world wide client is called a Web browser or simply a browser, and a world Wide Web server is called a Web server or sometimes just a server. Web Browser is a powerful tool used to browse the cyber space and is mainly used to access pages of the World Wide Web. The web browser contents a web-server and sends a request for information; and receives the information and then displays it on the user's computer. Web browsers and Web servers use a set of communication rules called Hypertext Transfer Protocol (HTTP). By clicking on the hypertext links on a page it is possible to jump from one internet site to another regardless of its location. Modern Web browsers are capable of browsing WWW, Gopher sites, FTP sites and also provide facilities for e-mail, etc. Initially NCSA's web browser Mosaic hit the market, which actually made the browsing popular. Now Web browsers from Netscape and from Microsoft are the user's choice. User can get hold of any such browser and start browsing the Net. A commercial version of the original browser, Mosaic, is in use. Many of the user interface features in Mosaic, however, went into the first widely-used browser, Netscape Navigator. Microsoft followed with its Microsoft Internet Explorer. Today, these two browsers are the only two browsers that the vast majority of Internet users are aware of. Although the online services, such as America Online, originally had their own browsers, virtually all now offer the Netscape or Microsoft browser. Lynx is a text-only browser for UNIX shell and VMS users. Another recently offered and well-regarded

browser is Opera. In simple words, a Web browser is a program that navigates through the World Wide Web and displays pages. The Web browser requests a page from a server based on its internet address. It retrieves the document from the server and displays the contents. A web-browser can be graphical or text-oriented and can make the Internet more easy to use. A text-oriented browser shows the user only the textual matter. A graphical browser allows the user to see more of what WWW has to offer such as graphics, photographs, and multimedia. Some of the examples are:

314 • Microsoft's Internet Explorer • Netscape Communication's Netscape Navigator • Opera • Amaya • HotJava • Lynx (text oriented) • NCSA Mosaic • Netcom's Web Cruiser • Intercon's netshark Web browsers are very versatile and allow performing the following activities: • Chat with other users on-line • Play games with others on-line • Access on-line multimedia including radio and video broadcasts • Visit web-sites • Contributing articles • On-line shopping • Send and receive E-mail • Read and send articles in newsgroup • Using Reminder Service of E-mail • Searching person's details • Sending gifts and flowers • Download files at machine • Search for information on the Internet • Subscription to electronic newsletters, journals, etc. • Participating in contests • Posting of resumes • Creating own web-site, and so on.

15.2 Defining Internet Explorer

315 Internet Explorer, often abbreviated as IE, sometimes also referred to as Microsoft Internet Explorer (MSIE), is a proprietary web browser made by Microsoft and currently available as part of Microsoft Windows. Microsoft's Web browser is the most widely used World Wide Web browser to search and view information files. Internet Explorer is an integrated component of all current versions of Microsoft Windows operating system and can also be downloaded from Microsoft's Web site. The Internet Explorer browser competes with an earlier popular browser, Netscape Navigator, now owned by AOL. Microsoft Internet Explorer is a graphical Web browser, which enables a user to fully experience the hypertext, photographs, sound, video etc. that are available on the world wide Web. Internet Explorer utilizes "point-and-click" technology to select hyperlinks and uses drop-down menu and toolbar buttons to navigate and access resources on the Internet. The major differences between Internet Explorer and Navigator are: Internet Explorer supports ActiveX and VBScript, while Navigator does not. Internet Explorer runs only under Windows and on Macintoshes whereas Navigator runs on these platforms as well as UNIX. Otherwise, the two browsers are very similar. Both support Java and JavaScript. Internet Explorer also supports ActiveX.

15.3 Features of Internet Explorer

Internet Explorer has been designed to view the broadest range of web pages and to provide certain features within the operating system, including Windows Update. Some of the progressive features of Internet Explorer are: • Handles requests for HTML files, interprets links, and deals with embedded images, audio, and video elements. • Keeps the history of websites visited. • Save a collection of documents or pages for later retrieval. • Provides convenient browsing through a row of buttons at the top of browser window.

316 • Supports web standards currently in use such as HTML, HTTP, Javascript, and Unicode.

15.4 Internet Explorer Environment

Internet Explorer runs inside a window that provides with an array of menus, buttons, labels, and information displays. When all of its components are made visible (not all are shown when first start, the Internet Explorer window looks like as shown below from top to bottom. Depending on what users are trying to do and how familiar are with the working of the browser, the elements of the Internet Explorer window may be either useful tools or distracting clutter. Fortunately, most of what users see can be customized to display exactly what users find worthwhile. Users can display or hide the elements of the Internet Explorer window and can change the amount of space devoted to each element. When Internet Explorer is first opened up on computer, the main screen of the program will appear. From top to bottom, a user may find these basic elements: • Title Bar • Main Menu bar • Internet Explorer Toolbars • Address or Location window • Link Bar • Main Browser window • Status bar

317 Anatomy of the Internet Explorer Window

1. Title Bar: The "Title Bar" at the very top of the window tells what the title of the page is viewing. The "Title Bar" will also tell what Internet Explorer application is currently active.
2. Main Menu Bar: Directly under the "Title Bar" is the "Main Menu Bar". This bar has many different sub-menus which control all options, functions, and commands for the entire Internet Explorer program. Some of the browsing controls can also be found in these sub-menus.
3. Internet Explorer Toolbar: Beneath this menu is the "Internet Explorer Toolbar". This toolbar contains all of the most frequently used commands and all of the browsing functions.

318 The Explorer bar is a way to browse through a list of links, such as History or channels, while displaying the pages those links open in the right side of the browser window. For example, if user clicks the Search button on the toolbar, the Explorer bar opens and user can use it to search for the needed Web site. User can display his/her Favorites list, History list, channels, or Search by clicking the toolbar. In addition, user can gain access to these items by clicking the View menu, and then pointing to Explorer bar. 4. Address Bar: Under the toolbar is the "Address Bar". This will tell the exact HTTP/URL location of the page user is currently viewing. User can also type a Web address directly into this bar and then press enter to go to that site. An Address bar is a space to type and display the address for a Web page. User does not even need to type the entire Web site address to go to that page. Just start typing, and the AutoComplete feature suggests a match based on Web sites visited previously. 5. Link Bar: Below the "Address Bar" is the "Link Bar". These buttons will take user to pages at Microsoft's Main home site where they have applications and information specifically designed for their easy use. The Link bar is a row of icons linked to Web pages. It is the most convenient, easiest-to-access place to put links to Web sites that user visits regularly. 6. Main Browser Window: Underneath the "Link Bar" is the "Main Browser Window". This window will display all of the information that is located at the Web site user is currently located at. Any text, images, movies, animation, links, or any other application files will be shown in this window. The scroll bars located on the right side and on the bottom of this window allow user to continue

319 viewing the page user is located at even when the page is too large to fit in the screen. This is the window where Web page will be displayed. The only way to hide this window is to minimize the entire Internet Explorer window, since the main point of running a browser is to view pages. 7. Status Bar: The very bottom of the page is the "Status Bar", placed at the bottom of the Internet Explorer window. It displays a variety of useful information. This bar tells what the progress of the browser is while it downloads files to the page, where links go to when user move over them, whether or not a document is secure, and any other information that the program feels is necessary for user to know. For example, when the cursor passes over a link in the viewing window, the URL of the link appears in the status bar. The progress of downloading a Web page is also displayed on this bar. Main Explorer Toolbar The main toolbar is composed of eleven different buttons. Each of these buttons has a different function and purpose in Internet Explorer. The individual buttons may be discussed as follows: 1. Back Button: Pressing this button immediately takes user back one document. If user has browsed into a multi-page document, pressing it repeatedly will continue to back user up one page at a time. Once user reaches starting location, it will be unavailable.

320 2. Forward Button: This button will take user forward to the next document if user has previously browsed multiple documents and had then backed-up to the page user is currently viewing. 3. Stop Button: The stop button stops ANY current operations by Internet Explorer. It will stop any type of file from loading. It can also be used to stop animations from continuing once a page is loaded. 4. Refresh Button: This button will reload the current document that user is viewing. If user is loading a document and the transfer was interrupted, user can reload the full document again by clicking here. 5. Home Button: This button will return user to the page selected as the default start-up page for Internet Explorer. It will not take user back to the beginning of web browsing, it will just return user to home location from where user is. 6. Search Button: This button will take user to the page selected as the default Web search page for Internet Explorer. 7. Favorites Button: This button will open up the Favorites menu. User can choose a favorite that he/she wishes to go to from the list, add a favorite to the list, or organize favorites from this menu. 8. Print Button: The print button will bring up a Print dialog box. In the box user can decide to print the contents of the viewing page, how many pages to print, and also how many copies to print. 9. Font Button: Pressing this button causes Internet Explorer to cycle through the available font sizes. This button is useful if the text is too small to read, or too large to fit comfortably in the window.

321 10. Mail Button: This button will open into a drop down menu from which, user can select to read or send E-Mail. User can also open up newsgroups from this menu. 11. Edit Button: This button will ONLY be on a user toolbar if he/she has a Windows system Web editor (such as Microsoft Frontpage or Microsoft Word) installed on his/her computer. If user presses this button, it will launch that editor and open the document user is currently viewing in it.

15.5 Working with Internet Explorer Opening Internet Explorer Internet Explorer can be opened up by performing any of the following steps:

- Double-click the Internet Explorer icon located on the desktop.
- Click Start, and select Internet Explorer.
- Click Start, select All Programs, and then select Internet Explorer.

Basic Browsing using Internet Explorer Internet Explorer has some built-in features which help user to make it easier to browse the web. When beginning browsing the web firstly, a user must know the Web address of a site to visit. Go up to the "Address Bar" near the top of the page, and click on it. Now user can type in the Web address of the site, and then may press enter. Internet Explorer will go to this site directly from whatever document user was currently viewing. If working on a Web page, user may click on:

- words or images which change the shape of the mouse pointer from an arrow to a hand and display a URL on the bottom of the screen when the mouse pointer is placed over it
- the blue words on the display screen
- the purple words on the display screen (the purple color indicates that the resource has been recently accessed on the terminal)

322 Nowadays, Web page creators are coloring their links in all sorts of ways. The best way to figure out which text represents a link is to position mouse over the words and see if the pointer shape changes from an arrow to a hand. The hand represents a link. If user wants to use pre-installed links, Internet Explorer offers a collection of Web sites in its Favorites collection. Click on Favorites on either the text bar or the tool bar at the top of the screen to access these resources. Some Useful Options on Menu Bar The menu bar at the top of the screen includes some useful options. File/New/Window: User can open up a second copy of Internet Explorer by using this feature. This allows user to visit more than one Web page at a time. File/Edit with...: User can edit the current Web page using the editor of his/her choice. Select the editor by going back to the Menu Bar and choosing Tools/Internet Options/Programs. User choices will be determined by software installed on computer. Edit/Find (on This Page): Internet Explorer allows user to do a text search of the document on the screen.

323 Tools/Show Related Links: Internet Explorer will display pages that are related in content to the current page. The Tools menu offers user many ways to customize Internet Explorer. Some Useful Tips for Doing Web Searches When user is using a search engine to find something on the Web, internet explorer helps to find ways to narrow search for faster and more efficient accessing of information. Most of the search engines use the same set of operators and commands in their search vocabulary. The following are the most commonly used operators and a brief description of each. These would be used when typing in a keyword or phrase into a search engine. • Quotes (" "): Putting quotes around a set of words will only find results that match the words in that exact sequence. • Wildcard Use (*): Attaching an * to the right-hand side of a word will return partial matches to that word. • Using Plus (+): Attaching a + in front of a word requires that the word be found in every one of the search results. • Using Minus (-): Attaching a - in front of a word requires that the word not be found in any of the search results. 15.6 Downloading, Emailing, and Printing Web Documents for Later Use User can download to disk, email, or print the Web page on the Internet Explorer screen. Downloading Click on File/Save As. A pop-up window will appear.

324 Save in: Select the desired drive. Save as type: User should make sure to save the page to the file type that will be useful to User. If User saves the page as a Web page, User will need a Web browser or HTML editor to view it. A text file (txt) can be viewed in a word processing program such a Word or WordPerfect. Click on Save E-Mailing Click on File/Send option. User may send the current page as an email message, or may insert the link to the current page within an email message. Once User makes the selection, email software will open. User can change the default software by going to the Menu Bar and choosing Tools/Internet Options/Programs. User choices will be determined by software installed on the computer. Printing To print the entire document, click on the Print icon on the Tool Bar Click on OK 325 To print selected pages Click on File/Print Preview. Click through the pages using the navigation arrows and make a note of which pages User want to print Click on Print. Click on the circle next to "Pages" Type in the pages separate by commas, e.g., 2, 4-7, 8, 9 OR, to print the page displayed in the Print Preview window, choose Current Page . Click on OK.

326 15.7 Customizing Internet Explorer Options Internet Explorer offers a number of customization options. Some of the useful features available under Tools/Internet Options on the Menu Bar can be divided into six tabs. Each one is explained below. 1. General Home Page: Specify the URL of the page user want to appear whenever Internet Explorer is opened, or whenever user clicks on the Home icon. Temporary Internet Files: This option allows user to view the files in the browser's cache. The cache holds viewed Web pages for subsequent quick viewing. Retrieving a file from the cache is much faster than repeated trips to the remote Web server where the file originated.

327 History: This option customizes access to pages user has visited with the History function. Here User can set the number of days to keep pages in history. Colors: Choose colors for links, visited links, and link hovers (the color appearing when mouse is over a link). User can also set a default text and background color. Fonts: Select the language script, the font displayed in Web pages, and the font displayed as plain text. Languages: Select the language that will display Web pages accessed with Internet Explorer. Accessibility: Choose to ignore colors, font sizes and font styles on Web pages. User can also set a style sheet as the display template for all Web pages viewed with Internet Explorer. 2. Security Here User can set levels of security for individual Web pages. 3. Content Content Advisor: User can enable ratings of objectionable content to control the pages that may be viewed with this browser. Certificates: This feature allows user to manage the identification certificates User may have. Personal Information: This consists of two options. AutoComplete: It will store entered Web address, information entered into forms, and usernames and passwords needed to access sites users have visited. When User is using browser, previous entries will come up as choices so that user don't have to retype the

328 information. This can make work go much faster. User can customize these options, and delete settings. My Profile: It offers a template for entering personal information. If a Web site requests this information, user can give permission for it to be used. 4. Connections Here User can store the information about Internet Service Provider, configure LAN settings, or send browser requests through a proxy server. 5. Programs Here user can set the programs user want the browser to use for HTML editing, email, Usenet news, collaboration ("Internet Call"), calendar and contact list. 6. Advanced This screen offers a number of options in the categories of accessibility, browsing, HTTP settings, Microsoft VM (Virtual Machine), multimedia access, printing, searching and security. User can set these options if is comfortable with them. 15.8 Summary A Web browser is client software that uses the Hypertext Transfer Protocol (HTTP) to make requests of Web servers throughout the Internet on behalf of the browser user. These Client software (user machine is a client to ISP's server) have various graphics capabilities to access the information from the Internet. Modern Web browsers are capable of browsing WWW, Gopher sites, FTP sites and also provide facilities for e-mail, etc. Web browsers from Netscape and from Microsoft are the user's choice. User can get hold of any such browser and start browsing the Net. Both are state-of-the-art browsers and the competition between them is fierce. Internet Explorer, sometimes also referred to as Microsoft Internet Explorer (MSIE), is a proprietary web browser made by Microsoft and currently available as part of Microsoft

329 Windows. Microsoft Internet Explorer is a graphical Web browser, which enables a user to fully experience the hypertext, photographs, sound, video etc. that are available on the world wide Web. Internet Explorer utilizes "point-and-click" technology to select hyperlinks and uses drop-down menu and toolbar buttons to navigate and access resources on the Internet. The major differences between Internet Explorer and Navigator are: Internet Explorer supports ActiveX and VBScript, while Navigator does not. Internet Explorer runs only under Windows and on Macintoshes whereas Navigator runs on these platforms as well as UNIX. Otherwise, the two browsers are very similar. Both support Java and JavaScript. Internet Explorer also supports ActiveX. Internet Explorer has some built-in features which help user to make it easier to browse the web. When beginning browsing the web firstly, a user must know the Web address of a site to visit. Go up to the "Address Bar" near the top of the page, and click on it. Now user can type in the Web address of the site, and then may press enter. Internet Explorer will go to this site directly from whatever document user was currently viewing. The Internet Explorer helps in information downloading, printing, emailing etc. There are also a number of options available with internet explorer for accessing the information in a faster and efficient manner from the web.

330 15.9 Self-Test questions 1. What is a web browser? 2. How can we surf the net through browser? 3. What are the main features of internet explorer? 4. Describe the internet explorer environment. 5. How does the Internet explorer help when user type a wrong web page address? 6. Explain the basic browsing options of Internet Explorer. 7. How can we download information from the web through internet explorer? 15.10 Suggested Readings 1. Introduction to Information Technology, IITL Education Solutions Limited, Pearson Education. 2. Norton, Peter, "Introduction to Computers", 2 nd Ed., Tata McGraw Hill. 3. Sinha, Pradeep K. and Sinha, Priti, Computer Fundamentals, Third Ed., BPB Publications.

Hit and source - focused comparison, Side by Side

Submitted text As student entered the text in the submitted document.

Matching text As the text appears in the source.

1/11

SUBMITTED TEXT

16 WORDS

76% MATCHING TEXT

16 WORDS

Objectives : After going through this chapter, you will be able to understand-the concept of

SA Computer Fundamentals and PC Software (Block 1).pdf (D148967917)

Machine language is the most BASIC form of programming and hence explicit instructions are to be given to the machine to perform each operation. Hence, it is necessary to tell the machine where to store numbers in its memory and how to add the numbers and so on. A statement of the form $P = Q + R$ is meaningless in a machine code. If you want to add Q with R, then you have to instruct the machine exactly how to transfer and process the numbers, where to store the result P, etc. Any set of instructions in a machine level language can be divided into the following four categories: Arithmetic, control, input-output and direct use. The functions of these statements are as follows: Arithmetic : Add, subtract, multiply and divide Control : Load, store and jump Input-output : Read and write Direct use : Start, halt and end The arithmetical operations or comparisons are done in the arithmetic logic unit with the use of accumulators. Hence, if you want to add two numbers together, we will need one instruction which will order the control unit to place a number in an accumulator and another instruction to identify the addition operation. Machine language instructions are represented by binary numbers, i.e., sequences consisting of 0s and 1s. For example, the binary sequence. 001010011010 could represent a 12 bit machine language instruction. The instruction is divided into two parts: operation code (or Op code) and an operand. Op code Operand 0010 10011010 70 The Op code specifies the operations such as add, multiply, etc. and the operand is the address of the data item that is to be operated upon. Thus, while using a machine language, we have to remember the code numbers for the operations and also keep track of the addresses of all the data items. Hence, machine language is highly complicated and complex and subject to error. Also, the programs written in machine language are machine dependent. That means, a program that has been developed for a particular machine cannot be run on another machine. It is clearly seen from the above program that coding a program in the binary form is very tedious. Hence, machine language is often coded in hexadecimal codes, which are still quite tedious. The programs written in machine language are also called as Object programs. 4.3. Symbolic/assembly languages To overcome the difficulties of writing programs in machine code, symbolic languages (which are also called as assembly languages) were developed in 1950s and were used extensively in some of the second generation computers. These languages enabled instructions to be written using symbolic codes (called mnemonics) rather than in strings of 0s and 1s. One word (or mnemonic) represented a particular machine language instruction. Once a series of instructions or words were put together to form a program, they are translated (by referencing a symbolic equivalence table) by the computer into their

Machine language is the most BASIC form of programming and hence explicit instructions are to be given to the machine to perform each operation. Hence, it is necessary to tell the machine where to store numbers in its memory and how to add the numbers and so on. A statement of the form $P = Q + R$ is meaningless in a machine code. If you want to add Q with R, then you have to instruct the machine exactly how to transfer and process the numbers, where to store the result P, etc. Any set of instructions in a machine level language can be divided into the following four categories: Arithmetic, control, input-output and direct use. The functions of these statements are as follows: Arithmetic : Add, subtract, multiply and divide Control : Load, store and jump Input-output : Read and write Direct use : Start, halt and end The arithmetical operations or comparisons are done in the arithmetic logic unit with the use of accumulators. Hence, if you want to add two numbers together, we will need one instruction which will order the control unit to place a number in an accumulator and another instruction to identify the addition operation. Machine language instructions are represented by binary numbers, i.e., sequences consisting of 0s and 1s. For example, the binary sequence. 001010011010 could represent a 12 bit machine language instruction. The instruction is divided into two parts: operation code (or Op code) and an operand. Op code Operand 0010 10011010 69 The Op code specifies the operations such as add, multiply, etc. and the operand is the address of the data item that is to be operated upon. Thus, while using a machine language, we have to remember the code numbers for the operations and also keep track of the addresses of all the data items. Hence, machine language is highly complicated and complex and subject to error. Also, the programs written in machine language are machine dependent. That means, a program that has been developed for a particular machine cannot be run on another machine. It is clearly seen from the above program that coding a program in the binary form is very tedious. Hence, machine language is often coded in hexadecimal codes, which are still quite tedious. The programs written in machine language are also called as Object programs. 4.3. Symbolic/assembly languages To overcome the difficulties of writing programs in machine code, symbolic languages (which are also called as assembly languages) were developed in 1950s and were used extensively in some of the second generation computers. These languages enabled instructions to be written using symbolic codes (called mnemonics) rather than in strings of 0s and 1s. One word (or mnemonic) represented a particular machine language instruction. Once a series of instructions or words were put together to form a program, they are translated (by referencing a symbolic equivalence table) by the computer into their

machine language counterparts. Let us assume that the following symbolic operation codes are used (The meaning of the mnemonics are given in brackets). 0001 ADD (Addition) 0010 SUB (Subtraction) 0100 DIV (Division) 0101 LDA (Load accumulator with value) 0110 STO (Store contents of accumulator) 0111 IN (Read a value into accumulator) 1000 OUT (Output value in accumulator to output device) 1001 JUN (Jump unconditionally) 1010 JGT (Jump to the address location if the contents of the accumulator are greater than zero) 1011 JNE (Jump to the address location if the contents of the accumulator are not equal to zero). It is clearly seen that symbolic language program is much simpler than the program written in the machine code. Yet assembly language programming has many of the same advantages as machine language programming; i.e., the program has access to and control over the registers, I/O ports and other features of the computer. However, the BASIC disadvantages of having to work with binary, octal or hexadecimal numbers are eliminated. The program that has been written using a symbolic language has to be translated into a machine language and only then it can be executed by the computer. The software program that will translate the program written in symbolic language is called as the assembler program or 72 assembler. The source program acts as an input to the assembler, which is loaded in the computer memory. Then, the assembler performs the translation and generates the equivalent machine code, which is called as the object program or object code. Once the object program is generated, the computer can execute the program without the aid of the assembler program. In typical computer systems, the assembler is stored in a secondary storage device and when needed loaded into the Random Access Memory (RAM). The source program is then entered by the user through the visual display unit and is also stored in RAM. The assembler translates the source program into object program and stores it in another portion of RAM. The object code or program can also be stored on some external device in order to execute it again and again when needed. In addition to using mnemonics to represent instructions, decimal numbers or names can also be used to refer to addresses of data names. By having such a system, the programmer is relieved of the problem of remembering each absolute address. Such names can have three or four characters and may use alphabets or numbers. However, the first character should be an alphabet. Thus, using symbolic names, the symbolic program may be written as IN STO Q IN STO R LDA R ADD Q STO P LDA P OUT 73 The symbolic language is called as low-level language because it is designed for a particular machine. It cannot be developed without knowing the size of the main memory and the size of the location word. 4.4. High level languages Unlike symbolic languages, high-level languages can be used

machine language counterparts. Let us assume that the following symbolic operation codes are used (The meaning of the mnemonics are given in brackets). 0001 ADD (Addition) 0010 SUB (Subtraction) 0100 DIV (Division) 0101 LDA (Load accumulator with value) 0110 STO (Store contents of accumulator) 0111 IN (Read a value into accumulator) 1000 OUT (Output value in accumulator to output device) 1001 JUN (Jump unconditionally) 1010 JGT (Jump to the address location if the contents of the accumulator are greater than zero) 1011 JNE (Jump to the address location if the contents of the accumulator are not equal to zero). It is clearly seen that symbolic language program is much simpler than the program written in the machine code. Yet assembly language programming has many of the same advantages as machine language programming; i.e., the program has access to and control over the registers, I/O ports and other features of the computer. However, the BASIC disadvantages of having to work with binary, octal or hexadecimal numbers are eliminated. The program that has been written using a symbolic language has to be translated into a machine language and only then it can be executed by the computer. The software program that will translate the program written in symbolic language is called as the assembler program or 71 assembler. The source program acts as an input to the assembler, which is loaded in the computer memory. Then, the assembler performs the translation and generates the equivalent machine code, which is called as the object program or object code. Once the object program is generated, the computer can execute the program without the aid of the assembler program. In typical computer systems, the assembler is stored in a secondary storage device and when needed loaded into the Random Access Memory (RAM). The source program is then entered by the user through the visual display unit and is also stored in RAM. The assembler translates the source program into object program and stores it in another portion of RAM. The object code or program can also be stored on some external device in order to execute it again and again when needed. In addition to using mnemonics to represent instructions, decimal numbers or names can also be used to refer to addresses of data names. By having such a system, the programmer is relieved of the problem of remembering each absolute address. Such names can have three or four characters and may use alphabets or numbers. However, the first character should be an alphabet. Thus, using symbolic names, the symbolic program may be written as IN STO Q IN STO R LDA R ADD Q STO P LDA P OUT 72 The symbolic language is called as low-level language because it is designed for a particular machine. It cannot be developed without knowing the size of the main memory and the size of the location word. 4.4. High level languages Unlike symbolic languages, high-level languages can be used

with a number of different hardware makes with little or no modifications. Hence, the programming and reprogramming expense is greatly reduced when the program has to be translated to another machine. Other advantages of high-level languages are as follows: 1. They are easier to learn than symbolic languages. 2. They require less time to write. 3. They are simpler in formats. 4. They are easier to maintain. 5. They are easier to understand. 6. They provide better documentation. 7. The programs written in such a language can be executed on any computer. 8. Four or five level instructions are reduced to a single high-level language statement. Some of the popular high-level languages are given in Table 4.1.

Table 4.1. High Level Languages Languages Year Meaning Developed By Main application area BASIC 1965 Beginners All Kemeny & Teaching 74 Purpose Symbolic Instruction Code

with a number of different hardware makes with little or no modifications. Hence, the programming and reprogramming expense is greatly reduced when the program has to be translated to another machine. Other advantages of high-level languages are as follows: 1. They are easier to learn than symbolic languages. 2. They require less time to write. 3. They are simpler in formats. 4. They are easier to maintain. 5. They are easier to understand. 6. They provide better documentation. 7. The programs written in such a language can be executed on any computer. 8. Four or five level instructions are reduced to a single high-level language statement. Some of the popular high-level languages are given in Table 4.1.

Table 4.1. High Level Languages Languages Year Meaning Developed By Main application area BASIC 1965 Beginners All Kemeny & 73 Teaching Purpose Symbolic Kurtz Instruction Code

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

3/11

SUBMITTED TEXT

131 WORDS

100% MATCHING TEXT

131 WORDS

Fig. 4.1. Schematic diagram of assembly process 4.5.2. Compiler A compiler translates a program written in a high level language to executable machine instructions. The compiler, like the assembler, is a program that resides on a disk or other mass-storage media. When the compiler is needed, it is called by the computer and loaded into RAM. It has to be remembered that the process for translating high level language source program into object code will be more lengthy and complex. Hence, compilers tend to be longer and more complex programs than assemblers and hence occupy more main memory space. Compilers can serve more purposes than mere translation. Many modern compilers have diagnostic capabilities that make them excellent quality assurance tools (quality assurance tools are some aids- in the form of

Fig. 4.1. Schematic diagram of assembly process 4.5.2. Compiler A compiler translates a program written in a high level language to executable machine instructions. The compiler, like the assembler, is a program that resides on a disk or other mass-storage media. When the compiler is needed, it is called by the computer and loaded into RAM. It has to be remembered that the process for translating high level language source program into object code will be more lengthy and complex. Hence, compilers tend to be longer and more complex programs than assemblers and hence occupy more main memory space. Compilers can serve more purposes than mere translation. Many modern compilers have diagnostic capabilities that make them excellent quality assurance tools (quality assurance tools are some aids- in the form of 86

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

4/11

SUBMITTED TEXT

21 WORDS

100% MATCHING TEXT

21 WORDS

Source Program Assembler Program Object Program Printed Listing, Error Messages, Results, etc. 87

Source Program Assembler Program Object Program Printed Listing, Error Messages, Results, etc.

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

programs- which will assure the quality of software that is produced). They are useful at all stages after pre-coding. The job of a compiler include. 1. To translate the source program statements into machine code (one high level language statement is normally translated into four or five machine language statements), 2. To check for conformity to programming standards. 3. To perform some limited, but useful semantic analysis (e.g. indicating mixed mode arithmetic). 4. To trace variables. 5. To include linkage for closed subroutines. 6. To allocate areas of main storage. 7. To provide extensive maps that are useful in diagnosing information flow. 8. To generate object program on tape or disk-whenver required. 9. To produce printed listing of the source and object programs when required. Sometimes compilers are also used to indicate less efficient programming practices such as two branch IF statements. Although it may look that the compiler does the same job as an assembler, it does fat more, particularly in the translation of source statements and the linkage of subroutines. The process of compilation is shown in Fig. 4.2, which consists of the following steps: 1. The user enters the source program through the terminal i.e. key- board and stores it in the RAM. 2. The compiler, which is available in RAM is called and asked to perform the translation. The translation process is repeated again 88 and again till all the errors in the source program are identified and corrected. 3. The compiler transforms the source program into an equivalent object program (in machine language). 4. The object program thus generated is executed in order to get the desired results. It has to be noted that compilers are not needed to run the program and are needed only to translate it. Also, one compiler is capable of translating source programs written in only one high level language. Thus, a COBOL compiler cannot be used to translate a FORTRAN source program.

programs- which will assure the quality of software that is produced). They are useful at all stages after pre-coding. The job of a compiler include. 1. To translate the source program statements into machine code (one high level language statement is normally translated into four or five machine language statements), 2. To check for conformity to programming standards. 3. To perform some limited, but useful semantic analysis (e.g. indicating mixed mode arithmetic). 4. To trace variables. 5. To include linkage for closed subroutines. 6. To allocate areas of main storage. 7. To provide extensive maps that are useful in diagnosing information flow. 8. To generate object program on tape or disk-whenver required. 9. To produce printed listing of the source and object programs when required. Sometimes compilers are also used to indicate less efficient programming practices such as two branch IF statements. Although it may look that the compiler does the same job as an assembler, it does fat more, particularly in the translation of source statements and the linkage of subroutines. The process of compilation is shown in Fig. 4.2, which consists of the following steps: 1. The user enters the source program through the terminal i.e. keyboard and stores it in the RAM. 2. The compiler, which is available in RAM is called and asked to perform the translation. The translation process is repeated again 87 and again till all the errors in the source program are identified and corrected. 3. The compiler transforms the source program into an equivalent object program (in machine language). 4. The object program thus generated is executed in order to get the desired results. It has to be noted that compilers are not needed to run the program and are needed only to translate it. Also, one compiler is capable of translating source programs written in only one high level language. Thus, a COBOL compiler cannot be used to translate a FORTRAN source program.

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

Obviously, a source program written in a high level language should also be translated into a machine language. The program that performs this operation is called the compiler (which is sometimes also called as a compiler program) or interpreter. High-level languages may be further subdivided into procedure oriented languages, problem-oriented languages and interactive programming languages. Procedure-oriented languages are useful for some special applications. For example, COBOL is a procedure-oriented language, which is used extensively in business applications. Other examples of procedure-oriented languages are FORTRAN and PL/1. Problem-oriented language just attempts to solve processing requirements with minimal programming effort allowing the user to focus on what results are desired rather than on the individual steps needed to get 75 those results. A typical example of problem-oriented language is RPG (Report Program Generator). Interactive programming languages allow the user to interact with the program in a conversational fashion. These languages are quite useful especially in computer aided design/computer aided manufacture (CAD/CAM). BASIC, PASCAL and APL are typical examples of interactive languages. High-level languages are sometimes classified as general purpose and special purpose languages. General-purpose languages (e.g. BASIC and PASCAL) are suitable for any type of application whereas special purpose languages (e.g. COBOL and LISP) are used only in some special application areas. Though there are about 150 computer languages, only a few are used extensively. In India, the languages which are used widely are : BASIC, FORTRAN (used mainly in scientific and engineering applications), COBOL (mainly used in business applications) and PASCAL. We shall consider some of the features of these four high level languages now. 4.4.1. BASIC language BASIC is an acronym for Beginners All Purpose Symbolic Instruction Code. This language was developed as a teaching aid for computing at Dartmouth College, USA, in 1965 by Professors Kemeny and Kurtz. BASIC is a simple and "friendly" language, easy to learn and particularly suitable for the non-specialist user. It also provides preparations for using more powerful languages like FORTRAN and PL/1. BASIC is widely used for commercial, scientific and educational purposes. BASIC has a fairly limited repertoire of statements and unlike any other high level language, allows only a certain range of identifiers to be used for the variables in the program. Hence, it is mainly used as an 76 introductory language. However, after the introduction of micro and home computers, which use BASIC as the only language, a number of application programs in different areas have been written in BASIC. BASIC is oriented to direct access use and programs are usually submitted to the computer by means of a console or visual display. Some of the

Obviously, a source program written in a high level language should also be translated into a machine language. The program that performs this operation is called the compiler (which is sometimes also called as a compiler program) or interpreter. High-level languages may be further subdivided into procedure oriented languages, problem-oriented languages and interactive programming languages. Procedure-oriented languages are useful for some special applications. For example, COBOL is a procedure-oriented language, which is used extensively in business applications. Other examples of procedure-oriented languages are FORTRAN and PL/1. Problem-oriented language just attempts to solve processing requirements with minimal programming effort allowing the user to focus on what results are desired rather than on the individual steps needed to get 74 those results. A typical example of problem-oriented language is RPG (Report Program Generator). Interactive programming languages allow the user to interact with the program in a conversational fashion. These languages are quite useful especially in computer aided design/computer aided manufacture (CAD/CAM). BASIC, PASCAL and APL are typical examples of interactive languages. High-level languages are sometimes classified as general purpose and special purpose languages. General-purpose languages (e.g. BASIC and PASCAL) are suitable for any type of application whereas special purpose languages (e.g. COBOL and LISP) are used only in some special application areas. Though there are about 150 computer languages, only a few are used extensively. In India, the languages which are used widely are : BASIC, FORTRAN (used mainly in scientific and engineering applications), COBOL (mainly used in business applications) and PASCAL. We shall consider some of the features of these four high level languages now. 4.4.1. BASIC language BASIC is an acronym for Beginners All Purpose Symbolic Instruction Code. This language was developed as a teaching aid for computing at Dartmouth College, USA, in 1965 by Professors Kemeny and Kurtz. BASIC is a simple and "friendly" language, easy to learn and particularly suitable for the non-specialist user. It also provides preparations for using more powerful languages like FORTRAN and PL/1. BASIC is widely used for commercial, scientific and educational purposes. BASIC has a fairly limited repertoire of statements and unlike any other high level language, allows only a certain range of identifiers to be used for the variables in the program. Hence, it is mainly used as an 75 introductory language. However, after the introduction of micro and home computers, which use BASIC as the only language, a number of application programs in different areas have been written in BASIC. BASIC is oriented to direct access use and programs are usually submitted to the computer by means of a console or visual display. Some of the

commonly used statements are READ, DATA, INPUT, FOR-NEXT, IF-THEN, DIM etc. In BASIC, a variable name is limited to a single letter followed by a digit or special characters. The READ statement in BASIC is always associated with a DATA statement. The IF-THEN statement is used for branching. The FOR-NEXT statement of BASIC is used to perform looping. Though BASIC compilers do exist, most of the installations use BASIC interpreter. Let us now consider an example program written in BASIC for the following problem which gives a fair idea about the structure of a BASIC program. The following program reads 100 positive numbers and outputs the highest and the sum of these numbers: 5 REM M is a variable used here to store the highest and T is used to store sum. 10 LET M = 0 20 LET T = 0 30 FOR I = 1 TO 100 40 PRINT "TYPE IN NUMBER" 50 INPUT N 60 IF N > M THEN 80 70 LET M = N 80 LET T = T + N 90 NEXT I 77 100 PRINT "THE MAXIMUM NO IS" ; M 110 PRINT "THE SUM OF THE NUMBERS =" ; T 120 END

4.4.2. FORTRAN The name FORTRAN stands for formula translation and as the name suggests it was developed and used as a scientific application language. FORTRAN is the oldest of the high level languages. It was introduced by the IBM in 1957 for their IBM-704 computers. Efforts were taken to standardize this language from 1962 and the first standard was produced by the American National Standards Institute and is called as FORTRAN II. In 1966, another version was brought out. This is called as FORTRAN IV. This version remained in constant use until 1978, when a new standard was produced called FORTRAN 77 then later on FORTRAN 90. This latest standard showed marked difference from FORTRAN IV which is, however, still the most commonly used version. A FORTRAN source program is composed of statements that may cause data to be read, processing to be performed and results to be recorded. FORTRAN statements are composed of operation symbols (+, -, *, /) and expressions (of the type $A + B - C$). Statements may be divided into four groups. Input-output statements— which describe the operations that are necessary to read in data and to write the results of the program. Control statements— which may determine the sequence in which the statements are to be executed. Arithmetic statements— which specify the mathematical calculations to be performed. These statements very closely resemble a conventional arithmetic formulae; for example, $A = B + C$ is a valid statement. 78 Specification statements— which provide certain additional facts such as the size of the input data that is read by the program, or the placement of alphabetical and numerical information on printed page. Each statement in FORTRAN produces about five instructions in machine language. Although FORTRAN can handle a fairly large range of simple data types (real, integer, logical, complex, etc.), it cannot deal adequately with characters or characters strings as data types, and the only complex

commonly used statements are READ, DATA, INPUT, FOR-NEXT, IF-THEN, DIM etc. In BASIC, a variable name is limited to a single letter followed by a digit or special characters. The READ statement in BASIC is always associated with a DATA statement. The IF-THEN statement is used for branching. The FOR-NEXT statement of BASIC is used to perform looping. Though BASIC compilers do exist, most of the installations use BASIC interpreter. Let us now consider an example program written in BASIC for the following problem which gives a fair idea about the structure of a BASIC program. The following program reads 100 positive numbers and outputs the highest and the sum of these numbers: 5 REM M is a variable used here to store the highest and T is used to store sum. 10 LET M = 0 20 LET T = 0 30 FOR I = 1 TO 100 40 PRINT "TYPE IN NUMBER" 50 INPUT N 60 IF N > M THEN 80 70 LET M = N 80 LET T = T + N 90 NEXT I 76 100 PRINT "THE MAXIMUM NO IS" ; M 110 PRINT "THE SUM OF THE NUMBERS =" ; T 120 END

4.4.2. FORTRAN The name FORTRAN stands for formula translation and as the name suggests it was developed and used as a scientific application language. FORTRAN is the oldest of the high level languages. It was introduced by the IBM in 1957 for their IBM-704 computers. Efforts were taken to standardize this language from 1962 and the first standard was produced by the American National Standards Institute and is called as FORTRAN II. In 1966, another version was brought out. This is called as FORTRAN IV. This version remained in constant use until 1978, when a new standard was produced called FORTRAN 77 then later on FORTRAN 90. This latest standard showed marked difference from FORTRAN IV which is, however, still the most commonly used version. A FORTRAN source program is composed of statements that may cause data to be read, processing to be performed and results to be recorded. FORTRAN statements are composed of operation symbols (+, -, *, /) and expressions (of the type $A + B - C$). Statements may be divided into four groups. Input-output statements— which describe the operations that are necessary to read in data and to write the results of the program. Control statements— which may determine the sequence in which the statements are to be executed. Arithmetic statements— which specify the mathematical calculations to be performed. These statements very closely resemble a conventional arithmetic formulae; for example, $A = B + C$ is a valid statement. 77 Specification statements— which provide certain additional facts such as the size of the input data that is read by the program, or the placement of alphabetical and numerical information on printed page. Each statement in FORTRAN produces about five instructions in machine language. Although FORTRAN can handle a fairly large range of simple data types (real, integer, logical, complex, etc.), it cannot deal adequately with characters or characters strings as data types, and the only complex

data structure available to the programmer is the array. Although it provides the ability to perform conditional statements, this ability is fairly-limited unless coupled with the GO TO (jump to another part of the program) statement. FORTRAN is not suited for problems involving file maintenance, editing of data or production of documents. Despite these drawbacks, FORTRAN is still widely used. The program that was written in BASIC in section 4.4.1 can be rewritten in FORTRAN as follows: C M is used to store highest value and T is used to store the sum REAL M, N T = 0.0 M = 0.0 DO 30 I = 1,100 WRITE (*,40) READ (*,10) N IF (N.LT.M) GO TO 50 M = N 50 T = T + N 30 CONTINUE WRITE (*,20) M, T 10 FORMAT (F 10.4) 20 FORMAT (2 X, 'MAXIMUM NO. IS =',F10.4, 5X, 'SUM=', F12.4) 40 FORMAT (5 X, 'TYPE IN NUMBER') END 79 4.4.3. COBOL The name COBOL is derived from Common Business Oriented Language. In 1959, a group was set up to develop a common language that will be suitable for business applications and in December that year an interim language was proposed, which formed the basis for COBOL. It was standardized only in 1968 by the American National Standards Institute. A further standard was presented in 1974. COBOL is by far the most widely used programming language in the world today. The COBOL character set is composed of the 26 letters of the alphabet, the numbers 0 through 9, and 12 special characters. The COBOL language consists of names (can have up to 30 characters) to identify locations; constants and literals; operators that specify some action or relationship; key words essential to the meaning of a statement; expressions consisting of names, constants, operators or key words; statement containing a verb and an item to be acted on; and sentences composed of one or more statements properly punctuated. COBOL programs can be written in 'paragraph' form and the source program consists of statements in English. A program written in COBOL can be divided into four parts: 1. The identification division – which is used to attach a unique identification such as program name, program identification, date written, date compiled, author name etc. 2. The environment division – which is used to acquaint the processor with the computer on which the program is to be compiled and executed. 3. The data division - which is used to define the characteristics and format of the data to be processed. 80 4. The procedure division – which is used to describe the internal processing that is to take place. All input output operations, logical decisions, data movement, and computing operations must be performed in the procedure section. Data types are declared by picture clauses, which define the exact way in which data is structured. Statements are separated from each other with a comma or a blank space and particular parts of statement and paragraphs have to occupy certain positions on the coding line. The IF statement in COBOL closely parallels that used in FORTRAN. COBOL doesn't

data structure available to the programmer is the array. Although it provides the ability to perform conditional statements, this ability is fairly-limited unless coupled with the GO TO (jump to another part of the program) statement. FORTRAN is not suited for problems involving file maintenance, editing of data or production of documents. Despite these drawbacks, FORTRAN is still widely used. The program that was written in BASIC in section 4.4.1 can be rewritten in FORTRAN as follows: C M is used to store highest value and T is used to store the sum REAL M, N T = 0.0 M = 0.0 DO 30 I = 1,100 WRITE (*,40) READ (*,10) N IF (N.LT.M) GO TO 50 M=N 50 T = T + N 30 CONTINUE WRITE (*,20) M, T 10 FORMAT (F 10.4) 20 FORMAT (2 X, 'MAXIMUM NO. IS =',F10.4, 5X, 'SUM=', F12.4) 40 FORMAT (5 X, 'TYPE IN NUMBER') END 78 4.4.3. COBOL The name COBOL is derived from Common Business Oriented Language. In 1959, a group was set up to develop a common language that will be suitable for business applications and in December that year an interim language was proposed, which formed the basis for COBOL. It was standardized only in 1968 by the American National Standards Institute. A further standard was presented in 1974. COBOL is by far the most widely used programming language in the world today. The COBOL character set is composed of the 26 letters of the alphabet, the numbers 0 through 9, and 12 special characters. The COBOL language consists of names (can have up to 30 characters) to identify locations; constants and literals; operators that specify some action or relationship; key words essential to the meaning of a statement; expressions consisting of names, constants, operators or key words; statement containing a verb and an item to be acted on; and sentences composed of one or more statements properly punctuated. COBOL programs can be written in 'paragraph' form and the source program consists of statements in English. A program written in COBOL can be divided into four parts: 1. The identification division – which is used to attach a unique identification such as program name, program identification, date written, date compiled, author name etc. 2. The environment division – which is used to acquaint the processor with the computer on which the program is to be compiled and executed. 3. The data division - which is used to define the characteristics and format of the data to be processed. 79 4. The procedure division – which is used to describe the internal processing that is to take place. All input output operations, logical decisions, data movement, and computing operations must be performed in the procedure section. Data types are declared by picture clauses, which define the exact way in which data is structured. Statements are separated from each other with a comma or a blank space and particular parts of statement and paragraphs have to occupy certain positions on the coding line. The IF statement in COBOL closely parallels that used in FORTRAN. COBOL doesn't

provide for absolute values, direct exponentiation of the base e, or trigonometric functions. The PERFORM statement in COBOL corresponds to FORTRAN DO statement and STOP RUN statement corresponds to the END statement of FORTRAN. COBOL is more difficult to learn than FORTRAN. However, the documentation provided by a COBOL source program is better than that provided by other languages. COBOL programs are organized in such a way that the data are physically separated from the computations (since they have separate data and procedure divisions) and this is generally accepted by present day computer scientists as being a natural representation of a programming problem. The language has the ability to handle conditional constructs and repetition constructs and is exceptionally good at dealing with records of different structures which may form files or data bases. The COBOL language is highly suitable where we have less computations and huge inputs and outputs in a given problem. However, the main drawback of the COBOL language is that it cannot be used for scientific applications. Further, COBOL processors require more CPU time to compile the programme. 81 The solution to the example problem (which was taken to illustrate and features of BASIC and FORTRAN languages) may be written using the COBOL languages as follows:

```
IDENTIFICATION DIVISION. PROGRAM-ID. ADDNOS.
AUTHOR. USER-NAME. ENVIRONMENT DIVISION.
CONFIGURATION SECTION. SOURCE-COMPUTER. IBM-PC.
OBJECT-COMPUTER. IBM-PC. DATA DIVISION. FILE SECTION.
WORKING-STORAGE SECTION. 77 N PIC 9(10)V9(4). 77 T PIC 9(12).9(4) VALUE ZERO. 77 M PIC 9(10).9(4) VALUE ZERO.
PROCEDURE DIVISION. START-PARA. PERFORM SUMMING 100 TIMES. DISPLAY "THE MAXIMUM NUMBER IS", M. DISPLAY "THE SUM OF THE NUMBERS IS", T. STOP RUN. SUMMING. DISPLAY "TYPE IN NUMBER". ACCEPT N. IF N &lt; M MOVE N TO M. ADD N TO T. 82
```

4.4.4 PASCAL

The language PASCAL is named after the French mathematician Blaise PASCAL who invented the first mechanical adding machine. Niklaus Wirth, a computer scientist from Switzerland developed PASCAL during 1968-70. The objective was to formulate an ALGOL-like language which could be used as a tool to teach people the methodology of structured programming. which is an efficient method of developing software? Since then PASCAL has been adopted as a primary teaching language for programming courses in many universities. Now-a-days, PASCAL is available with most of the microcomputer installations and used not only in educational applications but also in business and scientific applications. PASCAL is rich both in data structures and control structures. In addition to data types real, integer, character, boolean, the programmer can also device his own data type and define the values they can take (for example, he/she can choose the type, colour, which can have values such as red, green, black,

provide for absolute values, direct exponentiation of the base e, or trigonometric functions. The PERFORM statement in COBOL corresponds to FORTRAN DO statement and STOP RUN statement corresponds to the END statement of FORTRAN. COBOL is more difficult to learn than FORTRAN. However, the documentation provided by a COBOL source program is better than that provided by other languages. COBOL programs are organized in such a way that the data are physically separated from the computations (since they have separate data and procedure divisions) and this is generally accepted by present day computer scientists as being a natural representation of a programming problem. The language has the ability to handle conditional constructs and repetition constructs and is exceptionally good at dealing with records of different structures which may form files or data bases. The COBOL language is highly suitable where we have less computations and huge inputs and outputs in a given problem. However, the main drawback of the COBOL language is that it cannot be used for scientific applications. Further, COBOL processors require more CPU time to compile the programme. 80 The solution to the example problem (which was taken to illustrate and features of BASIC and FORTRAN languages) may be written using the COBOL languages as follows:

```
IDENTIFICATION DIVISION. PROGRAM-ID. ADDNOS.
AUTHOR. USER-NAME. ENVIRONMENT DIVISION.
CONFIGURATION SECTION. SOURCE-COMPUTER. IBM-PC.
OBJECT-COMPUTER. IBM-PC. DATA DIVISION. FILE SECTION.
WORKING-STORAGE SECTION. 77 N PIC 9(10)V9(4). 77 T PIC 9(12).9(4) VALUE ZERO. 77 M PIC 9(10).9(4) VALUE ZERO.
PROCEDURE DIVISION. START-PARA. PERFORM SUMMING 100 TIMES. DISPLAY "THE MAXIMUM NUMBER IS", M. DISPLAY "THE SUM OF THE NUMBERS IS", T. STOP RUN. SUMMING. DISPLAY "TYPE IN NUMBER". ACCEPT N. IF N &lt; M MOVE N TO M. ADD N TO T. 81
```

4.4.4 PASCAL

The language PASCAL is named after the French mathematician Blaise PASCAL who invented the first mechanical adding machine. Niklaus Wirth, a computer scientist from Switzerland developed PASCAL during 1968-70. The objective was to formulate an ALGOL-like language which could be used as a tool to teach people the methodology of structured programming. which is an efficient method of developing software? Since then PASCAL has been adopted as a primary teaching language for programming courses in many universities. Now-a-days, PASCAL is available with most of the microcomputer installations and used not only in educational applications but also in business and scientific applications. PASCAL is rich both in data structures and control structures. In addition to data types real, integer, character, boolean, the programmer can also device his own data type and define the values they can take (for example, he/she can choose the type, colour, which can have values such as red, green, black,

etc.). The data structures include the array and the record structure, which allows the structuring of elements of differing types. The control structures include the full flexibility of the conditional statement, and three different forms of repeat statement. It provides for sub-program facilities which are similar to those of ALGOL. Pointer variables which are of considerable importance while using dynamic data structures (such as linked lists and trees) can also be implemented. PASCAL is well suited for both batch and interactive modes of processing. The major drawback of PASCAL is that it is not yet standardized. Many versions are available which will make its transportability difficult. Compared to other languages like FORTRAN, PASCAL has poor input/output capabilities. Following is a version of the program written in PASCAL for the same example problem considered previously. 83

```
PROGRAM TOTAL (INPUT, OUTPUT); VAR I, N, M, T:
INTEGER ; BEGIN T: = 0; M: = 0; FOR I : = 1 TO 100 DO
BEGIN WRITELEN ('TYPE IN NUMBER'); READLN (N); IF N
&lt; M THEN M : = N; T : = T + N; END; WRITELN ('THE
MAXIMUM NUMBER IS', M); WRITELN ('THE SUM OF THE
NUMBERS IS', T); END. 4.4.5. C Language C is a structured
language, as is PASCAL, BASIC, COBOL and FORTRAN are
non-structured languages. C language uses blocks. A
block is a set of statements that are logically connected.
A structured language gives you a variety of programming
possibilities such as it supports the concept of functions.
Because of clarity, C is not only easier to program in but
also much easier to maintain. C was first used for system
programming. System programs make the computer
capable of performing useful work. C does have its
benefits. It has 28 keywords to remember. C code is very
portable. A 'C' program is a collection of one or more
functions. To write a program, you first create functions
and then put them together. 84 Following is a program
written in C for the same example considered previously:
# include &lt;stdio-h&lt; main () { int i; float t = 0, m = 0,
n; for (
```

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

etc.). The data structures include the array and the record structure, which allows the structuring of elements of differing types. The control structures include the full flexibility of the conditional statement, and three different forms of repeat statement. It provides for sub-program facilities which are similar to those of ALGOL. Pointer variables which are of considerable importance while using dynamic data structures (such as linked lists and trees) can also be implemented. PASCAL is well suited for both batch and interactive modes of processing. The major drawback of PASCAL is that it is not yet standardized. Many versions are available which will make its transportability difficult. Compared to other languages like FORTRAN, PASCAL has poor input/output capabilities. Following is a version of the program written in PASCAL for the same example problem considered previously. 82

```
PROGRAM TOTAL (INPUT, OUTPUT); VAR I, N, M, T:
INTEGER ; BEGIN T: = 0; M: = 0; FOR I : = 1 TO 100 DO
BEGIN WRITELEN ('TYPE IN NUMBER'); READLN (N); IF N
&lt; M THEN M : = N; T : = T + N; END; WRITELN ('THE
MAXIMUM NUMBER IS', M); WRITELN ('THE SUM OF THE
NUMBERS IS', T); END. 4.4.5. C Language C is a structured
language, as is PASCAL, BASIC, COBOL and FORTRAN are
non-structured languages. C language uses blocks. A
block is a set of statements that are logically connected.
A structured language gives you a variety of programming
possibilities such as it supports the concept of functions.
Because of clarity, C is not only easier to program in but
also much easier to maintain. C was first used for system
programming. System programs make the computer
capable of performing useful work. C does have its
benefits. It has 28 keywords to remember. C code is very
portable. A 'C' program is a collection of one or more
functions. To write a program, you first create functions
and then put them together. 83 Following is a program
written in C for the same example considered previously:
# include main () { int i; float t = 0, m = 0, n; for (
```

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

7/11

SUBMITTED TEXT

36 WORDS

93% MATCHING TEXT

36 WORDS

START LOAD COMPILER INTO MEMORY WRITE, FED AND THEN LOAD SOURCE PROGRAM RUN THE COMPILER IN ORDER TO GET OBJECT PROGRAM LOAD OBJECT PROGRAM EXECUTE OBJECT PROGRAM TO GET RESULTS STOP

START LOAD COMPILER INTO MEMORY WRITE, FED AND THEN LOAD SOURCE PROGRAM RUN THE COMPILER IN ORDER TO GET OBJECT PROGRAM ARE THERE ERRORS? NO LOAD OBJECT PROGRAM EXECUTE OBJECT PROGRAM TO GET RESULTS STOP

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

Interpreter High level languages may also be translated using interpreters. Using a compiler, the whole program is translated completely and then the resulting machine language version is executed. Whereas using an interpreter, each instruction is translated and executed. Whereas using an interpreter, each instruction is translated and executed in turn- which means repetitive translation of instructions within loops. In other words, the interpreter takes one source program instruction, translates it into object code and executes it, then takes the next instruction, translates it, and so on. Use of an interpreter can save core space since the interpreter program itself is quite small in size. It also eliminates the need to store the program's translated object code in the computer. Interpreters are convenient and easy to use because they are totally interactive. The user can sit in front of the terminal and converse with the computer in order to solve his problems directly through the interpreter. Whereas if he is using a compiler, he has to generate the object code and if he wants to change something in the program (to add some new features or to correct errors), the entire process must be repeated. That is, the program is corrected, re-compiled, and executed. Whereas using an interpreter, changes and additions can be made interactively.

Interpreter High level languages may also be translated using interpreters. Using a compiler, the whole program is translated completely and then the resulting machine language version is executed. Whereas using an interpreter, each instruction is translated and executed. Whereas using an interpreter, each instruction is translated and executed in turn- which means repetitive translation of instructions within loops. In other words, the interpreter takes one source program instruction, translates it into object code and executes it, then takes the next instruction, translates it, and so on. Use of an interpreter can save core space since the interpreter program itself is quite small in size. It also eliminates the need to store the program's translated object code in the computer. Interpreters are convenient and easy to use because they are totally interactive. The user can sit in front of the terminal and converse with the computer in order to solve his problems directly through the interpreter. Whereas if he is using a compiler, he has to generate the object code and if he wants to change something in the program (to add some new features or to correct errors), the entire process must be repeated. That is, the program is corrected, re-compiled, and executed. Whereas using an interpreter, changes and additions can be made interactively.

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

9/11

SUBMITTED TEXT

449 WORDS

99% MATCHING TEXT

449 WORDS

```
n &lt; m) { m = n; } t + = n; } printf ("\n The Maximum Number is %f", m); printf ("\n The sum of Number is % f", t); } 4.5. Assemblers, compiles and interpreters As discussed already, any program that is not written in machine language has to be translated before it is executed by the computer. We shall see the details of three types of translator programs : assembler, compiler and interpreter now. 4.5.1 Assembler Assembler is a program which is used to translate programs written in assembly language into machine code. One machine instruction is 85 generated for each source program instruction. The resulting program can be executed only when the assembly process is completed. The functions of an assembler can be summarized as: 1. To translate mnemonic operation codes into machine code, and symbolic addresses into machine addresses. 2. To assign absolute address to any symbolic address or label names. 3. To check each instruction for its correctness and to generate diagnostic messages. 4. To place each instruction in central memory. 5. To provide a cross-reference table between all symbolic names used and their absolute addresses. 6. To provide the necessary linkages for closed subroutines. 7. To produce the object program on tape, or disk when required. 8. To produce printed listing of the source and object program with comments. 9. To inform the control unit (after all the errors have been corrected to execute the program starting from the first instruction of the object code. The main stages of assembling are: 1. Input source program (in assembly language) through an input device. 2. Use the assembler to produce an object program in machine language. If errors are found by the assembler, the source program must be modified and corrected. This process of modification and 86 correction should be repeated until all errors found have been corrected. 3. Load object program into main memory by using a loader. 4. Execute the object program (by using the operating system) in order to get the desired results. The above process is illustrated in Fig. 4.1.
```

```
n; for (i = 1; i m) { m = n; } t + = n; } printf ("\n The Maximum Number is %f", m); printf ("\n The sum of Number is % f", t); } 4.5. Assemblers, compiles and interpreters As discussed already, any program that is not written in machine language has to be translated before it is executed by the computer. We shall see the details of three types of translator programs : assembler, compiler and interpreter now. 4.5.1 Assembler Assembler is a program which is used to translate programs written in assembly language into machine code. One machine instruction is 84 generated for each source program instruction. The resulting program can be executed only when the assembly process is completed. The functions of an assembler can be summarized as: 1. To translate mnemonic operation codes into machine code, and symbolic addresses into machine addresses. 2. To assign absolute address to any symbolic address or label names. 3. To check each instruction for its correctness and to generate diagnostic messages. 4. To place each instruction in central memory. 5. To provide a cross-reference table between all symbolic names used and their absolute addresses. 6. To provide the necessary linkages for closed subroutines. 7. To produce the object program on tape, or disk when required. 8. To produce printed listing of the source and object program with comments. 9. To inform the control unit (after all the errors have been corrected to execute the program starting from the first instruction of the object code. The main stages of assembling are: 1. Input source program (in assembly language) through an input device. 2. Use the assembler to produce an object program in machine language. If errors are found by the assembler, the source program must be modified and corrected. This process of modification and 85 correction should be repeated until all errors found have been corrected. 3. Load object program into main memory by using a loader. 4. Execute the object program (by using the operating system) in order to get the desired results. The above process is illustrated in Fig. 4.1.
```

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

10/11

SUBMITTED TEXT

21 WORDS

100% MATCHING TEXT

21 WORDS

START Load Interpreter into memory Enter and load source program into memory Translate and execute the program using interpreter

START Load Interpreter into memory Enter and load source program into memory Translate and execute the program using interpreter

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>

11/11**SUBMITTED TEXT**

233 WORDS

100% MATCHING TEXT

233 WORDS

When he needs to modify or add something, he can do it immediately through the interpreter. When using an interpreter, a programmer gets a feeling that he is directly speaking to a computer and commanding its attention. However it has some disadvantages, program statements that are used multiple times must be translated each time they are executed. Another big drawback of an interpreter is its low speed. Interpreters execute programs at a speed 10-20 times lower than the equivalent machine code generated by an assembler or compiler. In some applications, this low speed is undesirable and also unacceptable. The process of executing programs using an interpreter is shown in Fig. 4.3. It has to be noted that on some micros and personal computers, the interpreter for the BASIC language is resident in ROM and hence there is no need to load them on to the memory. 4.6. Self-test questions 1. Distinguish between low-level language and high-level language. 2. Define the following terms: (a) Computer program (b) Object program (c) Source program 3. Compare and contrast between machine language and symbolic language programs. 4. Name the four popular high-level languages used in India. 5. Why is COBOL mainly used for business applications? What is its main drawback? 6. Explain the features of the following languages briefly: (a) BASIC (b) FORTRAN (c) COBOL 91

When he needs to modify or add something, he can do it immediately through the interpreter. When using an interpreter, a programmer gets a feeling that he is directly speaking to a computer and commanding its attention. However it has some disadvantages, program statements that are used multiple times must be translated each time they are executed. Another big drawback of an interpreter is its low speed. Interpreters execute programs at a speed 10-20 times lower than the equivalent machine code generated by an assembler or compiler. In some applications, this low speed is undesirable and also unacceptable. The process of executing programs using an interpreter is shown in Fig. 4.3. It has to be noted that on some micros and personal computers, the interpreter for the BASIC language is resident in ROM and hence there is no need to load them on to the memory. 4.6. Self-test questions 1. Distinguish between low-level language and high-level language. 2. Define the following terms: (a) Computer program (b) Object program (c) Source program 3. Compare and contrast between machine language and symbolic language programs. 4. Name the four popular high-level languages used in India. 5. Why is COBOL mainly used for business applications? What is its main drawback? 6. Explain the features of the following languages briefly: (a) BASIC (b) FORTRAN (c) COBOL 90

W <https://vibdoc.com/an-introduction-to-computer-in-banking.html>