GCA(S1) 01-02

Computer Fundamentals

SEMESTER - 1

BACHELOR OF COMPUTER APPLICATION

Block 2

KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Anjana Kakati Mahanta, Gauhati University
Prof. (Retd.) Pranhari Talukdar, Gauhati University
Dr. Jyotiprokash Goswami, Assam Engineering College

Course Co-ordinator :

Dr. Sanjib Kr. Kalita, KKHSOU
Dr. Tapashi Kashyap Das, KKHSOU
Sruti Sruba Bharali, KKHSOU

SLM Preparation Team

UNITS	CONTRIBUTORS
7	Mrs. Gitimoni Talukdar, Royal Group of Institutions
8	Dr. Uzzal Sharma, Assam Don Bosco University
9	Dr. Tapashi Kashyap Das, KKHSOU
10	Sruti Sruba Bharali, KKHSOU
11	Mr. Saurav Sutradhar, Royal Group of Institutions

Editorial Team

Content : Dr. Guruprasad Khatorniar, Gauhati University
Language : Mr. Sawpon Dowerah, Rector Icon Academy
Structure, Format & Graphics: Dr. Tapashi Kashyap Das, Sruti Sruba Bharali, KKHSOU

June, 2018

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State University is made available under a Creative Commons Attribution-Non Commercial-ShareAlike4.0 License (International): http.//creativecommons.org/licenses/by-nc-sa/4.0.

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University.

Head Office : Patgaon, Rani Gate, Guwahati-781 017; Web: www.kkhsou.in
Sub-Head Office : Housefed Complex, Dispur, Guwahati-781 006

The university acknowledges with thanks the financial support provided by the **Distance Education Bureau, UGC** for the preparation of this study material.

BACHELOR OF COMPUTER APPLICATION

COMPUTER FUNDAMENTALS

DETAILED SYLLABUS

Block-2

UNIT 7: Instruction Types **Pages: 89-95**

Instruction Format; Different Types of Instruction Format; Bit Manipulation Instructions; Data Transfer Instructions

UNIT 8: Basic of Operating Systems **Pages: 96-116**

Functions of Operating System; Types of Operating System: DOS, LINUX, UNIX, XENIX, WINDOWS; Multiprocessing OS, Time Sharing Operating System

UNIT 9: Introduction to Programming **Pages: 117-140**

Algorithm: Conventions used in Writing Algorithm, Method for Developing an Algorithm; Pseudocode; Flowchart: Symbols of Flowchart, Advantages and Limitations of Flowchart; Computer Languages: Machine Language, Assembly Language, High Level Language

UNIT 10: File Structure **Pages: 141-152**

Data and Information; Records; Files: Operation on Files; Database Approach: Objectives of Database, Characteristics of Database

UNIT 11: Computer Security **Pages: 153-168**

Computer Security; Malware; Computer Viruses; Worms; Other Threats: Trojan Horses, Spyware, Adware, Scareware; Counter Measures: Antiviruses, Deception Techniques, Working of an Antivirus

BLOCK INTRODUCTION

This is the *Second Block* of the course “Computer Fundamentals”. After completing this block you will find yourself at a moderate level of expertise in knowledge of computer basics.

This block comprises the following six units:

Unit 7: introduces you to the different types of instruction format.

Unit 8: concentrates on the basics of different operating systems. Functions and types of operating systems like DOS and LINUX are discussed in this unit.

Unit 9: gives us an introduction to programming. Concepts of algorithm, flowchart, pseudocode are discussed in this unit.

Unit 10: introduces us to the basics of file structure. Files, records and different operations on files are described in this unit.

Unit 11: deals with computer security. Different types of virus and antivirus software are discussed in this unit.

The structure of Block 2 is as follows:

UNIT 7: Instruction Types

UNIT 8: Basic of Operating Systems

UNIT 9: Introduction to Programming

UNIT 10: File Structure

UNIT 11: Computer Security

While going through a unit, you will notice some along-side boxes, which have been included to help you know some of the difficult, unseen terms. Again, we have included some relevant concepts in ‘‘LET US KNOW’’ along with the text. And, at the end of each section, you will get “CHECK YOUR PROGRESS” questions. These have been designed to self-check your progress of study. It will be better if you solve the problems put in these boxes immediately after you go through the sections of the units and then match your answers with “ANSWERS TO CHECK YOUR PPROGRESS” given at the end of each unit.

UNIT 7: INSTRUCTION TYPES

UNIT STRUCTURE

7.1 Learning Objectives

7.2 Introduction

7.3 Instruction Format

7.4 Different Types of Instruction Format

7.5 Bit Manipulation Instructions

7.6 Data Transfer Instructions

7.7 Let Us Sum Up

7.8 Answers to Check Your Progress

7.9 Further Reading

7.10 Model Questions

7.1 LEARNING OBJECTIVES

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

- learn about instruction format
- describe the different types of instruction format
- explain bit manipulation
- learn about data transfer instruction.

7.2 INTRODUCTION

In this unit we shall be able to learn about instruction format and its different types including bit manipulation and data transfer instructions.

A microprocessor is a chip that performs all the essential functions of the central processor unit (CPU) in a computer. The function of a microprocessor system is implemented by a sequence of data transfer between register in the memory and the microprocessor as well as between register in the memory and Input / Output devices as shown in Figure 7.1. The data transformation occurs primarily in register within the microprocessor. Each programmable register in the memory has an address, allowing it to be singled out for use in a data transfer or transformation.

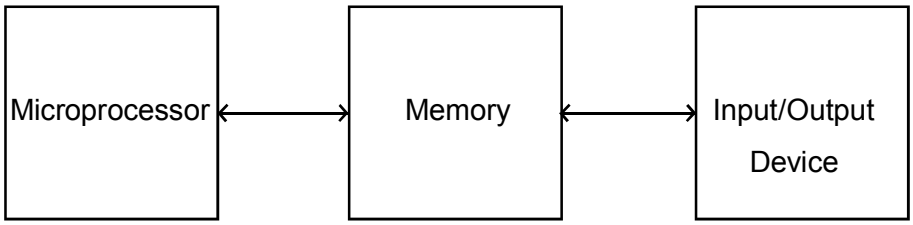


Fig. 7.1: Data transfer between Memory and Microprocessor, I/O Device

The types of individual transfer and transformation possible are specified by the microprocessor instruction sets. Each instruction in the set causes one or more data transfer or transformation. A sequence of such instructions consist a program.

7.3 INSTRUCTION FORMAT

Computer instructions are the basic components of a machine language program. A machine instruction has an Operation Code (also called opcode) and zero or more operands. The opcode denotes the type of operation that is going to be performed by the microprocessor and the operands are the data on which the respective operation will be performed. Based on the architecture of the microprocessor- the number of bits allowed per instruction, the number of operands allowed per instruction, the types of instructions and data it can process are different.

More specifically, the instruction sets are differentiated by the following features:

- Operand storage in the CPU (data can be stored in a stack structure or in registers)
- Number of explicit operands per instruction– zero, one, two and three.
- Operand location (e.g. In register-memory instructions, operands are fetched into the registers from the memory. In typical register-register instructions, operands are fetched from the registers. In memory-memory instructions, operands are read from the memory and stored in registers then again write in a different memory location)
- Operation (Based on the type of operation performed or based on whether the instructions are allowed to access memory and not)

- Type and size of operands (Whether the operand is an addresses or numbers or some other format of data)



CHECK YOUR PROGRESS

Q.1: How can you differentiate the instruction sets?

Let us now see what could be the different types of instruction format.

7.4 DIFFERENT TYPES OF INSTRUCTION FORMAT

The most common instruction formats include zero, one, two or three operands. Arithmetic and logical operations typically have two operands but can be executed with one operand if the accumulator is implicit (i.e. if the other operand is present in accumulator). We can extend this idea to three operands if we consider the final destination as a third operand. We can also use a stack that allows us to have zero operand instructions. The following are some common instruction formats:

- **OPCODE only** (zero address)

Opcode

  e.g.

  NOP (defined as: no operation)

- **OPCODE + 1 Address** (usually a memory address)

Opcode	Address 1

  e.g.

  LDA ADDR

  (defined as: LOAD ACCUMULATOR DIRECT. Which means, the content of the memory location ADDR is loaded into the accumulator)

- **OPCODE + 2 Addresses** (usually both registers or one is register and the other one is memory address)

Opcode	Address 1	Address 2

  e.g.

MOV r1,r2

(defined as: MOVE. Which means, move the contents of the register r2 into register r1)

- **OPCODE + 3** Addresses (usually registers or combinations of registers and memory)

Opcode	Address 1	Address 2	Address 3

e.g.

ADD r1, r2, r3

(defined as: Addition. Which means, the contents of the registers r2 and r3 are added and the result is stored in register r1)

The above 4 instruction formats are obtained based on the number of operands used. While on the basis of other features, instructions can further be classified as:

- Data transfer instructions
- Arithmetic instructions
- Branch instructions
- Boolean instructions
- Logical instructions



CHECK YOUR PROGRESS

Q.2: What are the different types of instruction format?

Q.3: What are the different types of instructions?

7.5 BIT MANIPULATION INSTRUCTIONS

A bit is the smallest unit of digital data. It has binary value either 0 or 1. The bit manipulation instructions are used to modify bit level data or binary data. They are mainly classified as:

1) Logic Instructions
2) Shift Instructions
   a) Logical Shifts
   b) Arithmetic Shifts

3) Rotate Instructions

1) Logic Instructions: The instructions which are used to support logical expressions by performing bitwise AND, OR, XOR and NOT operations are called logic instructions.

e.g. AND, OR, XOR, NOT

2) Shift Instructions: Shift instructions are generally used to shift the bits either to the left or right depending on some condition.

- Logical Shifts

  e.g. SHL , SHR, SHRD, SHLD

The logical shift instructions are mainly used when you want to get rid of bits on the beginning or end of a number. They are also used when you want to position the bits in a register.

- Arithmetic Shifts

  e.g. SAL , SAR

3) Rotate Instructions

e.g. ROL , ROR , RCL , RCR

7.6 DATA TRANSFER INSTRUCTION

Data transfer instruction consists of 15 basic operations and 86 variations. The basic operation involved is the transfer of DATA between two registers of a microprocessor system. One of the registers is always located in the microprocessor and the other may be located in an Input/ Output device or Memory in the microprocessor.

The registers located in the microprocessor are referred to as internal registers e.g. A, B, C, D, E, H, L, SP, PC. Here Register A is the accumulator, SP is the stack pointer, PC is the program counter and rest are general purpose registers. Accumulator is also a general purpose register. On the other hand ROM (Read Only Memory), RWM (Read Write Memory) and Input/Output are referred to as external registers. The Data transfer among these different registers is performed by Data transfer instructions. The register from which data is transferred is the source register and the register to which data is transferred is the destination register. A transfer involves copying the contents of the source register into the destination register.

Each data transfer instruction identifies the source register and the destination register. Identification of one or both of these registers may be implied by the instruction in memories or may be explicit. Internal registers are frequently implied whereas the external registers are usually identified by an explicit address that is part of the instruction.

Example of Data transfer instructions: MOV, MVI, LXI, LDA, STA, LHLD, SHLD, LDAX, STAX etc.



7.7 LET US SUM UP

- The function of a microprocessor system is implemented by a sequence of data transfer and transformation operations.
- Each programmable register in the memory has an address.
- A sequence of instructions consist a program.
- A machine instruction has an Operation Code (also called opcode) and zero or more operands.
- Features to classify instruction sets are: Operand storage in the CPU, Number of explicit operands per instruction, Operand location, Operation by the instruction, Type and size of operands.
- Based on the number operands there are 4 types of instruction formats containing zero, one, two or three operands.
- The BIT manipulation instructions are used to modify BIT level data or binary data.
- Data transfer instructions performs Data transfer among different registers.



7.8 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: The instruction sets are differentiated by operand storage in CPU, number of explicit operands per instruction, operand location, operation, type and size of operands.

Ans. to Q. No. 2: Different types of instruction formats are:

- Opcode only
- Opcode+1 Address
- Opcode+2 Addresses
- Opcode+3 Addresses

Ans. to Q. No. 3: Instructions can be further classified as:

- Data transfer instructions
- Arithmetic instructions
- Branch instructions
- Boolean instructions
- Logical instructions



7.9 FURTHER READING

1) Mano, M. M. (1993): *Computer System Architecture*; Prentice Hall, 3, 299.
2) Stallings, W. (2000): *Computer Organization and Architecture: Designing for Performance*; Pearson Education India.



7.10 MODEL QUESTIONS

Q.1: Discuss the different types of instruction formats along with examples.

Q.2: Explain different types of bit manipulation instructions with examples.

Q.3: What do you mean by data transfer instructions? Explain.

Q.4: What are the different logic instructions and shift instructions?

*** ***** ***

UNIT 8: BASICS OF OPERATING SYSTEMS

UNIT STRUCTURE

8.1 Learning Objectives

8.2 Introduction

8.3 Functions of Operating System

8.4 Types of Operating System

- 8.4.1 DOS
- 8.4.2 LINUX
- 8.4.3 UNIX
- 8.4.4 XENIX
- 8.4.5 WINDOWS

8.5 Multiprocessing OS

8.6 Time Sharing Operating System

8.7 Let Us Sum Up

8.8 Answers to Check Your Progress

8.9 Further Reading

8.10 Model Questions

8.1 LEARNING OBJECTIVES

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

- what is an operating system
- main functions of operating system
- brief idea of different operating systems commonly used
- the concept of multiprocessing operating system
- the concept of time sharing operating system.

8.2 INTRODUCTION

Computers have changed the way we think and live today. We have become dependent on the power and the capability of the computer system. We all are concerned with the application of computers. Whether it is booking a flight tickets or typing a document we all use computer now-a-days. Even

our smart phone can also be called a mini computer. The same computer can be seen from different perspectives. For the laymen, a computer is a very useful device, which can help them to fulfill their requirements. For engineers and other technical person, a computer is an electronic device which can be utilized to aid the different requirements. A computer system consists of some physical components known as hardware. The hardware components are connected to each other to perform any activity. In the past, any work to be carried out in the computer needed to be performed manually. Users needed to be very efficient and skilled to use the hardware. As time passes by, technology has changed a lot resulting in the development of new and improved hardware. However, hardware components are not functional if software is not installed. This software is known as Operating System. Operating system is an interface between the user and the hardware. In fact we can say that the hardware complexities are eliminated by the operating system software.

Operating system can be defined as a system software, which works as an interface between the user and computer hardware, which eliminates the hardware complexities.

Microprocessor system is implemented by a sequence of data transfer between register in the memory and the microprocessor as well as between

8.3 FUNCTIONS OF OPERATING SYSTEMS

Operating system performs a set of activities which can be broadly categorized into two different viewpoints. They are user viewpoint and system viewpoint. The user viewpoint describes the functions in a top-down manner where as the system viewpoint describes the functions in a bottom-up manner. Now let us discuss the functions of operating systems from both the views.

- **User View:** User view describes the functions without looking into the complexities of the core hardware and system. For example, a user can do lot of works in the computer. But how the data is got processed or how the resources are allocated to the process, the user is not concerned about it. The operating system itself is taking

care of it. Thus, the operating system makes the hardware abstraction and thereby provides a user friendly interface to the user. Thus the operating system acts as a mediator between the user and computer hardware. Some of the operating system functions that can be realized from the user point of view are as follows:

- **User Interface:** Operating system provides an interface so that the user can make use of the computer hardware. The interface can be categorized into two types. They are:
  a) **Command Line Interface (CLI):** This provides a command line interface to the user. The user need to type in the command to interact with the system.
  b) **Graphical User Interface (GUI):** This provides a graphical interface, where the users mainly need to interact with the computer using some pointing device such as mouse.
- **Program Execution:** When a program is executed, there are some events like allocating memory, loading the program to that memory need to take place. All these happen without the concern of the user. The operating system takes care of all these so that each program executes smoothly.
- **Controlling of I/O Devices:** When we interact with the computer, for example, giving input to the computer or receiving output from the computer it happens without any difficulty. Most of the time, the user is not required to control the input and output device manually. The operating system itself handles the input and output operations. For example, when we press some keys from the keyboard, the characters are displayed on the monitor. These are all controlled by the operating system.
- **Accessing File System:** Data in a computer can be stored either temporarily and upto some extent permanently. To sore the data upto some extent permanently, files are used. But file is a logical concept and it is associated with the mapping of some physical memory. The mapping of logical memory to physical memory is managed by the operating system. The operating system also

performs all the file management related activities without the concern of the user.

- **Error Detection and Recovery:** When we work with a computer, there is a chance that many errors may occur. If proper action is not taken in a timely manner, it ultimately leads to the pushing of the system to a dead lock situation. The operating system, detects any circumstances that may lead to a deadlock situation and takes preventive action against it. If in any case dead lock happens, the system automatically recovers from the deal lock. This corrective measure and prevention is performed by the operating system without the concern of the user.

- **System View:** Besides providing support to the user and helping user to interact with the computer, operating system also perform some other activity for the smooth functioning of the system. These functions can be considered under the category of the system view. Some of the important functions are:
  - **Resource Manager:** The CPU, memory and I/O devices are termed as resource present in a computer. Each of these resources to managed properly. In each operating system, there is a module that can be used to manage the resources.
  - **Control Program Address Space:** When multiple programs run in a computer, each of them need some amount of memory. At the same time, there should not be any conflict regarding the use of the memory by multiple programs. Operating system takes care of this so that same address space is not allocated to multiple programs running simultaneously.



CHECK YOUR PROGRESS

Q.1: Fill in the blanks:

a) The operating system acts as a .....

b) ..... ensures that two processes occupies different address space.

c) CLI stands for .....

d) GUI stands for .....

8.4 TYPES OF OPERATING SYSTEM

The following are some commonly used operating system.

8.4.1 DOS

DOS is the acronym for Disk Operating System. This operating system was originally developed by Microsoft, so it is also known as MS-DOS. It is one of the simplest types of operating systems available. It supports only a single user at a time and can perform only one task at a particular time. So, it is also known as a single user-single tasking operating system. MS-DOS provides a command line interface to interact with the computer. Although it is a very simple operating system and requires very less space to install, it is one of the most successful operating systems that have been developed to date.

The first version of MS-DOS, i.e. MS-DOS 1.0 was installed in IBM PC in August 1981. It was a 16 bit operating system.

A Quick and Dirty History: When IBM launched its revolutionary personal computer, the IBM PC, in August 1981, it came complete with a 16-bit operating system from Microsoft, MS-DOS 1.0. This was Microsoft's first operating system, and it also became the first widely used operating system for the IBM PC and its clones.

MS-DOS 1.0 is actually a renamed version of QDOS (Quick and Dirty Operating System). The QDOS was written by Tim Paterson, a Seattle Computer Products employee that supports Intel 16-bit 8086 CPU (central processing unit). Tim Paterson was later hired by Microsoft Corporation.

Microsoft eventually came up with two versions of DOS. They are PC-DOS (the IBM version) and MS-DOS (the Microsoft version). Although, initially there were no difference between PC-DOS and MS-DOS, but with time both PC-DOS and MS-DOS acquired specific features which distinguishes them from one another.

Gradually MS-DOS gained popularity in the personal computer market and it provided a support for Microsoft to grow further. Even after launching of Windows as well, MS-DOS provides a good contribution towards the income of the company. The subsequent versions of MS-DOS provide few features similar to the features possessed by UNIX. In the improved versions of MS-DOS, it has support for multiple HDD partitions, disk compression, improved memory management, improved text editors, etc.

8.4.2 LINUX

Linux is a UNIX-based operating system which has a graphical user interface. It was created by Linus Torvalds. Linus Torvalds a Helsinki university student has developed it as a hobby and announced formally on 25th August, 1991 and named it as Linux. It is a free operating system. It means it is free to download and use for customization. Anyone can change the code of the operating system and because of this it has gradually become very powerful and at the same time provides the best in security features. At the present time it has many features and capable to compete with most of the existing operating system including windows operating system.

Being an open source software the core of the operating system can be modified by anyone. There is no need of acquiring license to debug and improve the list of existing features. Linux is one of the operating system which is developed by using many programming languages. The operating system code is written mainly by using C and C++. Along with these two popular languages the contribution of Java, Perl and Lisp is also found in traces. As far as the kernel of java is concern, the presence of C code is about 95 per cent.

Another very significant benefit of Linux is the cost. The core operating system is free and most of the software that supports Linux comes with a GNU General public License. It means everything in LINUX is free. Another significant advantage of LINUX operating system is its support for strong networking.

Linux File Systems: LINUX supports many file systems each with its advantages and disadvantages. Linux doesn't require to have only one file system. Many file systems are supported at the same time in LINUX. The following is a small list of popular file systems used by LINUX.

- **The ext2 file system:** It is the oldest file system supported by LINUX yet simple which is still one of the popular operating system. Ext2 stands for extended 2 file system.
- **The ext3 file system:** It is an improvement over the ext2 file system, very significant addition to it is the concept of adding journaling.
- **The ext4 file system:** It is an improvement over the ext3 file system. In addition to the other features of that ext3, this file system has a support for large files.
- **The reiserfs file system:** It is completely a new file system written from scratch. The main feature of this file system is its speed. It can quickly access the location with hundreds of files inside.
- **The btrfs file system:** It is a file system supported by LINUX with some promising features. It supports huge storage backend volumes, multi-device spanning, snapshotting ets.
- **The xfs file system:** It is an enterprise-ready, high performance journaling file system. It offers very high parallel throughput and is therefore a common choice amongst enterprises.
- **The zfs file system (ZFSonLinux):** It is a multi-featured file system and most significantly it supports block-level checksumming, snapshotting, compression, extremely large volumes, copy-on-write, deduplication, remote replication and many more.

8.4.3 UNIX

In the mid 1960s, AT&T bell laboratories got involved in developing a new operating system called Multics. The main objective behind Multics is providing large-scale computing services to the others. The people that were involved in the designing of Multics

later involve d in the design of UNIX operating system. Like other operating systems UNIX also provides the layer between the hardware and the applications that runs on the computer.

UNIX is one of the very fast and powerful operating systems which can give a better competition to the windows operating system. The main advantage of UNIX is its security. As far as the architecture of the operating system is concerned, it is centered around the kernel. The kernel is considered as the heart of the UNIX operating system. The kernel is responsible for allocating memory, loading and executing programs, keeping track of the users file and directory. The kernel accomplishes these tasks by providing an interface between the other programs running under its control and physical hardware of the computer. When a request is made to use the system resource, the kernel determines whether or not the request is valid and if it is found valid, the kernel reads the required block of data and passes it back to the program. The UNIX system programs do not have access to physical hardware of the computer rather they uses the kernel services to fulfill the requirement. Apart from the kernel the UNIX operating system also contain the following:

- Shell

Whenever we login to a UNIX system, we are taken into a shell program. Each shell is provided with a prompt that usually displayed on the interface. To interact with the operating system, we need to enter command at this prompt. The shell is a command interpreter; which takes the command passes it to the operating system kernel to be acted upon. It then displays the results of this operation on your screen. There are several shells are available which are as follows:

- Bourne shell (sh)
- C shell (csh)
- Korn shell (ksh)
- TC Shell (tcsh)
- Bourne Again Shell (bash)

The different users has the flexibility to use different shells as pre their requirement. Initially the system administrator will supply a default shell, which can be overridden or changed. Apart from the above each shell also contain it own programming language known as shell script.

- Utilities

UNIX provides several hundred utility programs, often referred to as commands. The UNIX operating system accomplishes the following tasks with the help of utilities. Some of the common tasks are:

- editing
- file maintenance
- printing
- sorting
- programming support
- online info, etc.

The hierarchical File Structure of UNIX: All of the files in the UNIX file system are organized into a multi-leveled hierarchy called a directory tree. A family tree is an example of a hierarchical structure that represents how the UNIX file system is organized. The top of the file system consists of a single directory called ‘‘root’’ which is represented by a / (slash). All other files comes under the root.

File Types: The UNIX file system contains several different types of files:

- **Ordinary Files:** This type of files is used to store information, such as some text you have written or an image you have drawn. The file that we create comes under this category. The ordinary files are always located within or under a directory file.
- **Directories:** This types of file us used to organized group of files. It may contain ordinary files, special files or other directories.
- **Special Files:** In UNIX each hardware device is considered as a file. For example the keyboard and monitor is considered as standard input and output file respectively.

- **Pipes:** One of the important utility that UNIX provides is pipe. This concept is used to take the output of the previous command as input for the next command. The (|) symbol is used as the pipe symbol.

8.4.4 XENIX

Microsoft XENIX is a 16 bit operating system developed by Microsoft. It is one of the interactive, multiuser, multitasking operating system developed by Microsoft. The XENIX can be considered as advanced version of UNIX. It runs on Intel 8086, Zilog Z8000, Motorola M68000, and DEC PDP-11 series processor. The XENIX operating system is one of the early operating system based on UNIX. In fact XENIX is a version of Unix from Microsoft that was equipped with some interesting enhancements such as multiple virtual consoles accessible via Alt-F1, Alt-F2,... Alt-F10. These features were later acquired by LINUX in its successive versions.

The XENIX kernel performs all the functions that a typical operating system kernel performs. The basic functions of the kernel are management of mass storage, processes, memory, and devices, and it checks for any errors during operation.

The main features of the XENIX operating system are:

- **Standard System Calls:** As present in other operating system, it also has a set of system calls which can be used by programmers to perform different jobs. The programmers can use over 60 different system calls to request different services from the kernel.
- **Speed:** The speed at which the XENIX performs its job depends on the speed of the processor and because of the support of iAPX 286, the processing speed is convincing.
- **Small Size:** One of the promising features of the XENIX kernel is that it has about 10,000 lines of code. As a result of this the size of the kernel is very small and so it is easier to maintain and debug.

- **High-level Language:** The major portion of the kernel is written in C programming and not in assembly language. Because of this the operating system runs on variety of different computer.
- **Hierarchical File System:** Like UNIX, XENIX has a hierarchical file system. Because of this the organization of files become very easy and systematic.
- **Interpretation of Devices as File:** Like UNIX, each device I is treated as file. Because of this the exchange of data between device and operating system become easier. This introduces the concept of device independence within the operating system.
- **Code-Data Independence:** The Program code and data are kept in separate areas of memory independent of each other. This provides independence on the user's data and same code can be shared by multiple users at the same time.
- **Device Driver Support:** The XENIX has a set of pre-loaded drivers especially for terminals, Winchester disks, flexible disks, and tapes. It also has a set of guidelines for writing new drivers so that a new hardware can be added to the existing OS without much difficulty.

The Root Directory: Like other operating system, XENIX also contains a root directory (/), which has few subdirectories as discussed below:

/bin – This directory has the XENIX commands that users execute frequently.

/boot – This directory has the code for a program that is required to start the system.

/dev – This directory contains special device files.

/etc – This directory has commands that are usually reserved for the system administrator plus files that the system administrator uses for various administrative purpose.

/lib – This directory has libraries of subroutines.

/lost+found – This directory lists directories that are not linked into the file system because of some problem. The

entries are placed in this directory automatically by the fsck command that the System.

Administrator uses regularly to check the integrity of the file system.

/mnt – This directory is normally used for file systems that are mounted on the root file system.

/sys – This directory has the code for the XENIX kernel.

/tmp – This directory is used for temporary files that are created by programs. These files may be removed during normal operations and they are usually removed each time the system is started.

/usr – This directory is used for all login directories. It is the ancestor of all user files and directories.

/xenix – This file has executable code for the XENIX kernel for the hard disk system.

/xenix.f – This file has executable code for the XENIX kernel for the flexible disk system.

XENIX Shells: The XENIX operating system contains various shells which works as a command interpreter. The user can choose the most suitable shell appropriate as per his or her requirement. The most commonly used one is Bourne shell which interprets commands and passes information to the programs that we want to execute. Apart from the Bourne shell, the XENIX contains the following other shells:

- restricted shell
- visual shell and
- C shell

8.4.5 WINDOWS

History of Windows: Windows operating system is developed and sold by Microsoft. Microsoft started their operation commercially during August 1981 by releasing 1.0 version of the operating system Microsoft DOS (MS-DOS), which is a 16-bit command-line operating system.

As MS-DOS was missing the graphical user interface, the company realized the importance of user-friendly interface and they came up with the first version of Microsoft Windows (Microsoft Windows 1.0) in November 1985. Although it was able to attract the people because of the user-friendly interface, but it could not gained much popularity. In December 9, 1987, Microsoft launched Microsoft Windows 2.0, but it also did not do much better.

On May 22, 1990 Microsoft launched Windows 3.0, which was a success for them. The Windows 3.1 was launched in 1992, and finally the Microsoft Windows for Workgroups, later renamed it was renamed as Windows 3.11, which added network capabilities to the per-existing set of features.

Later, on August 24, 1995, Microsoft launched the operating system Microsoft Windows 95. Windows 95 gained popularity and reveals Microsoft's intension to transfer some of MS-DOS's capabilities into Windows. The operating system has got the disadvantage of heavy reliance 16-bit DOS and continue with the limitations of the FAT16 file system, that retards the capability of giving long file system.

Eventually many versions of Windows 95 were released namely 95A OSR1, Windows 95B OSR2, Windows 95B OSR2.1 and Windows 95C OSR2.5 to incorporate few additional features.

Microsoft released the next version of Windows on June 25, 1998, known as Windows 98. Windows 98 natively supported features other than those of MS-DOS but was still based upon it. Windows 98 also suffered from poor memory handling when multiple applications were running at the same time, which could cause system to not function properly. A second edition of Windows 98 came out on February 17, 2000; it was named Windows 98 SE (for “Second Edition”). The Windows 98 SE could handle some of the previously identified issues in a much better way.

On September 14, 2000, Microsoft released Windows Me (for Millennium Edition), also called Windows Millennium. The

foundation of Windows Millennium was based largely on Windows 98 and eventually on MS-DOS. This version of windows incorporated additional multimedia and software capabilities. Moreover, it also included a system-restore mechanism for returning to a previous state in the event of a crash.

Side by side Microsoft also released another operating system to cater the requirement of other sets of user. It was named as Windows NT (for Windows "New Technology").

On May 24, 1993, the first version of Windows NT was released. It was called Windows NT 3.1, and was followed by Windows NT 3.5 in September 1994 and Windows 3.51 in June 1995. Later on August 24, 1996, Windows NT 4.0 was launched for sale, and it became a true success for the company.

In July 1998, Microsoft released Windows NT 4.0 TSE (Terminal Server Emulation), the first Windows system that allowed terminals to be plugged into a server, i.e. use thin clients to open a session on the server.

The next version of windows NT was launced on February 17, 2000, and was named as Windows 2000. Windows 2000 is a 32-bit operating system with characteristics of Windows NT, along with improved task manager and full compatibility with USB and FireWire peripherals.

Finally, on April 24, 2003, a server operating system was released by Microsoft: Windows Server 2003.

On October 25, 2001, Windows XP was released and it gave a new look and feel to the existing operating system and in a short time it became very popular. It gave a new dimension towards navigating the Start menu, taskbar, and Control Panel.

In 1996, Windows Vista was released with the strongest security system yet. Windows Vista also featured enhancements to Windows Media Player as more and more people came to see their PCs as central locations for digital media. Here one could watch television, view and send photographs, and edit videos.

Because of the increasing demand by wireless technology, Windows 7 was released for the wireless world of the late 2000s. Gradually the desktop market replaced by laptops having the wireless features and many wireless device were developed to fulfill the requirement of the people. So, it became a need to change the operating system as well. In late 2000s, Windows 7 was released to fulfill this requirement.

Windows 7 made the interface more fun to use by including features like Snap, Peek, and Shake.

During the year 2012, Windows 8 was launched which had a state of the art windows interface which smoothly worked with touch, mouse and keyboard. This operating system provided a new dimension to a variety of hand held devices like tablet, smart phone etc. This operating system also provided an enhancement to the Windows desktop, a new look to the taskbar and provides new features towards the file management.

Windows 8 provide flexibility towards the interface by providing tiles that connects to different programs.

In the year 2013-14, some enhancement to the Windows was made and as a result of that Windows 8.1 was launched. The significant additions were introduction of many powerful apps as well connectivity to cloud. This version of windows also added so many new apps such as Bing Food & Drink, Bing Health & Fitness, and great utility apps like Reading List, Calculator, and Alarms.

During the early 2015, Windows 10 arrived and Microsoft made the initial versions of the operating system available to enthusiasts via the Windows Insider Program. They invited customers to contribute towards the development of the operating system. This Windows operating system can be considered the best operating system ever in which Insider Program plays an important role.

Windows File Systems: There are two major file systems that are used by Microsoft Windows operating system. They are FAT, which is inherited from MS DOS and later in was enhanced to FAT32, and

the second one is most widely used NTFS file systems. Recently Microsoft has also released ReFS a new generation file system for Windows 8 Servers.

FAT (File Allocation Table): FAT file system is one of the most widely used yet simple file systems used in windows environment. It consists of file system descriptor sector (boot sector or superblock), file system block allocation table (referenced as File Allocation Table) and plain storage space to store files and folders. Files on FAT are organized in directories. Each directory is an array of 32-byte records, each defines file or file extended attributes (e.g. long file name).

The number in FAT12, FAT16, FAT32 stands for the number if bits used to enumerate file system block. This means that FAT12 may use up to 4096 (212) different block references, FAT16 – 65536 (216) and FAT32 (232) – 4294967296. Actual maximum count of blocks is even less and depends on implementation of file system driver.

The old floppy disks uses FAT12. The FAT16 (or simply FAT) and FAT32 are widely used for flash memory cards and USB flash sticks. The mobile phones, digital cameras and other portable devices also support this file system.

FAT or FAT32 is a file system, used on Windows-compatible external storages such as external hard disks or disk partitions with size below 2GB (for FAT) or 32GB (for FAT32). Windows cannot create FAT32 file system over 32GB.

NTFS (New Technology File System): NTFS was introduced in Windows NT for the first time and at present it is the most widely used file system for Windows. This file system supports disk partitions above 32 GB and is a default file system for latest windows operating system. The file system has a support for access control, encryption, disk compression etc.

ReFS (Resilient File System): Another latest file system developed by Microsoft is ReFS.It is presently available for Windows 8 Servers. File system architecture of this file system is completely different

from the existing windows file system which is mainly organized in the form of B+-tree. ReFS has high tolerance to failures and included many new features such as Copy-on-Write (CoW) in which no data is written over the existing ones and rather into a new disk space. With any file modifications a new copy of metadata is created into any free storage space, and then the system creates a link from older metadata to the newer ones. As a result a system stores significant quantity of older backups in different places which provides for easy file recovery unless this storage space is overwritten.



CHECK YOUR PROGRESS

Q.2: Do as directed:

a) DOS stands for ..... (Fill in the blank)
b) DOS is a single user operating system. (True / False).
c) Linux was developed by ..... (Fill in the blank)
d) LINUX is a open source software. (True / False)
e) The core of UNIX operating system is the kernel. (True / False)
f) FAT stands for ..... (Fill in the blanks)
g) ReFS stands for Resilient File System (True / False).

8.5 MULTIPROCESSING OS

Multiprocessor system refers to the type of computer system that contains two or more central processing units (CPU) within a single computer system. These multiple CPUs are in a close communication with each other sharing the common computer bus, memory and other peripheral devices. All the CPU share the same memory among themselves. These systems are referred as tightly coupled systems.

These types of systems are used when time plays a very important role and it is needed to process data at high speed moreover the operation takes place on a large volume of data. These systems are generally used in environment like satellite control, weather forecasting, Air traffic control etc.

The operating system installed is known as multiprocessing operating system which is based on the symmetric multiprocessing model. In symmetric multiprocessing model, each processor runs an identical copy of operating system and these copies communicate with each other. In multiprocessing system each processor is assigned a specific task. There is a master processor that regulates the tasks and controls the entire system. In fact it follows a master-slave concept. This type of system can be considered batter sometime because they share the common bus, memory and other peripheral devices. The main advantage of multiprocessor system is to get more work done in a shorter period of time. At the same time it also provides reliability to the system because of the presence of the multiple CPUs. When any one of the CPU faces problem, its responsibility is taken care of by the other available CPU's. So, it does not stop the system from functioning, It only slows down the output.

The whole task of multiprocessing is managed by the operating system, which allocates different tasks to be performed by the various processors in the system.

In order that the multiprocessing operating system functions effectively, the computer system must have the followings:

1) **Motherboard Support:** As the system possess multiple CPUs the motherboard also need to be capable of handling multiple processors. This means additional sockets or slots for the extra chips and a chipset capable of handling the multiprocessing arrangement.
2) **Processor Support:** The processors capable of being a part of multiprocessing system must be used.

Applications designed for use in multiprocessing are said to be multi-threaded, which means that they are broken into smaller routines that can be run independently. This allows the operating system to let these threads run on more than one processor simultaneously, which is multiprocessing that results in improved performance.

In a multiprocessor system, the dynamically sharing of resources among the various processors may cause therefore, a potential bottleneck. There are three main sources of contention that can be found in a multiprocessor operating system.



CHECK YOUR PROGRESS

Q.3: Do as directed:

a) Multi processing system needs all the time multiple processor. (True / False)

b) Multi-processing is managed by ..... (Fill in the blanks)

8.6 TIME SHARING OPERATING SYSTEM

In some system it may become a necessity that the same system need to the shared by several users during the same period of time. A time sharing system allows many users to share the computer resources simultaneously. In this type of system, the resources of a computer are allocated to multiple users or multiple programs in time slots. For example when a server is shared by multiple users to run programs, each user feels that the server resources are exclusively granted to him /her. But by having a single CPU, it is not possible. So, the same CPU and other resources are shared among each user in terms of time slots.

The time sharing systems were developed to provide an interactive use of the computer system. A time shared system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer. It allows many users to share the computer resources simultaneously. As the system switches rapidly from one user to the other, a short time slot is given to each user for their executions.

The time sharing system the operating system schedules the programs so that each program gets fare share of the CPU. This is done by dividing the available time among all the programs. The OS allocates a slot of time to each user. When this time expires, it passes control to the next user on the system. The time allowed is extremely small and the users are given the impression that they each have their own CPU and they are the sole owners of the CPU. This short period of time that a user gets attention of the CPU; is known as a time slice or a quantum.

The time-shared systems are more complex than the multi-programming systems. In time-shared systems multiple processes are managed simultaneously which requires an adequate management of main memory so that the processes can be swapped in or swapped out within a short time.



CHECK YOUR PROGRESS

Q.3: Do as directed:

a) Time sharing system is shared by multiple users. (True / False).

b) Linux is not a Time Sharing System. (True / False).



8.7 LET US SUM UP

- Operating system can be defined as a system software, which works as an interface between the user and computer hardware, which eliminates the hardware complexities.
- Operating system performs a set of activity which can be broadly categorized into two different viewpoints. They are:
  - user view point and
  - system view point
- User view describes the functions without looking into the complexities of the core hardware and system.
- System view addresses the issues of smooth functioning of the system.
- There are different operating systems from different vendors. Some of the commonly used operating systems are:
  - DOS
  - LINUX
  - UNIX
  - XENIX
  - WINDOWS
- Multiprocessor system refers to the type of computer system that contains two or more central processing units (CPU) within a single computer system.

- The time sharing systems were developed to provide an interactive use of the computer system by multiple users.



8.8 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: a) Resource allocator; b) Operating System; c) Command Line Interface; d) Graphical User Interface.

Ans. to Q. No. 2: a) Disk Operating System; b) True; c) Linus Torvalds; d) True; e) True; f) File Allocation Table; g) False

Ans. to Q. No. 3: a) False; b) Operating system

Ans. to Q. No. 4: a) True; b) False



8.9 FURTHER READING

1) Andrew S. Tanenbaum (2002): *Modern Operating Systems* (Second Ed.); New Delhi: Prentice Hall of India.
2) Deitel, H. M.: *Operating Systems* (Second Ed.); Pearson Education.
3) Silberschatz, Galvin: *Operating System Concepts* (Fifth Ed.); John Wiley and Sons (Asia) Pte Ltd.
4) William Stallings: *Operating Systems* (Fourth Ed.); New Delhi: Prentice Hall of India.



8.10 MODEL QUESTIONS

Q.1: State and explain the functions of Operating System.

Q.2: What are the different views of operating system are there? Explain.

Q.3: What is DOS? Explain its main feature.

Q.4: Discuss the file system of UNIX.

Q.5: Explain the different file system of Linux.

Q.6: What is a shell in UNIX? Explain? State different types of Shell.

Q.7: What is a time sharing system? Explain.

Q.8: What is multiprocessing? Explain.

*** ***** ***

UNIT 9: INTRODUCTION TO PROGRAMMING

UNIT STRUCTURE

9.1 Learning Objectives

9.2 Introduction

9.3 Algorithm

- 9.3.1 Conventions used in Writing Algorithm
- 9.3.2 Method for Developing an Algorithm

9.4 Pseudocode

9.5 Flowchart

- 9.5.1 Symbols of Flowchart
- 9.5.2 Advantages and Limitations of Flowchart

9.6 Computer Languages

- 9.6.1 Machine Language
- 9.6.2 Assembly Language
- 9.6.3 High Level Language

9.7 Let Us Sum Up

9.8 Answers to Check Your Progress

9.9 Further Reading

9.10 Model Questions

9.1 LEARNING OBJECTIVES

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

- define algorithm
- learn the conventions used in writing algorithms
- develop algorithm for various computational problems
- define pseudocode
- write pseudocode for different problems
- pictorially represent algorithm in the form of flowchart
- learn different symbols of flowchart
- draw flowcharts for computations, decision making, loops etc.
- learn about machine, assembly and high level language
- learn about the classification of computer languages.

9.2 INTRODUCTION

In the previous unit we have learnt about different operating system. This unit deals with some introductory concept of programming. Learners will be acquainted with the key elements of programming to design and develop accurate and efficient programs. Algorithms, pseudocodes, and flowcharts are used in the process of program development to help the programmers as well as the users to clearly understand the solution to the problem at hand.

In this unit we will learn the method of writing algorithms and pseudocodes, and pictorially represent a schematic flow of logic in the form of flowcharts. At the end we will briefly discuss the three types of computer languages i.e., machine language, assembly language and high level language.

9.3 ALGORITHM

Algorithm is a sequence of instructions to solve a problem. An algorithm gives the logic of the program, that is, a step-by-step description of how to arrive at a solution. In general terms, an algorithm provides a blueprint to writing a program to solve a particular problem. Once we have an blueprint of a solution, we can implement it in any high-level language, such as C, C++, Java etc.

An algorithm has a finite number of steps and some steps may involve decision-making and repetition.

9.3.1 Conventions used in Writing Algorithms

The following are the conventions used in writing algorithms:

- **Name of the Algorithm:** Every algorithm is given a name which reflects the task performed by it.
- **Introductory Comments:** Within this, the task performed by the algorithm is described briefly. The variables used along with their data types are mentioned here.
- **Steps:** An algorithm comprises of a sequence of steps which should be numbered. The statements within a step are executed in a left to right manner.

- **Comments in Steps:** Each step is preceded by a brief comment describing it's function. Comments within a step are enclosed in parentheses.

The algorithm shown in Example 6:(Finding the largest of three numbers) is written considering all the above conventions. For better understanding, let us write some more algorithms of simple problems.

9.3.2 Method for Developing an Algorithm

- State the problem you are trying to solve in clear and concise terms.
- List the inputs (information needed to solve the problem) and the outputs (what the algorithm will produce as a result)
- Identify the steps needed to convert or manipulate the inputs to produce the outputs.
- Test the algorithm: choose data sets and verify that your algorithm works.

Example 1: Write an algorithm to add two numbers.

Solution: Step 1: Input the first number as A
Step 2: Input the second number as B
Step 3: Set Sum = A + B
Step 4: Display Sum
Step 5: End

The algorithm for adding two numbers can also be written as:

Step 1: Read a
Step 2: Read b
Step 3: Sum a + b
Step 4: Write Sum
Step 5: Stop

Both the two algorithms for adding two numbers are correct, but only the word/text are different. While writing algorithms, it is good to keep the following points in mind:

- Usually, words like Read, Input or Accept can be used to represent input operation to give values of variables to the computer.
- Display, Show, Write or Print can be used to represent output operation to show the result computed by the computer.
- Back arrow ‘’ represents the value obtained by evaluating the right side variable/expression to the left side variable. The symbol ‘=’can also be used instead of ‘’.
- In case of branching or conditional statements, **If-Then** or **If-Then-Else** is used. The conditional statement usually contains relational operators such as <, >, <=, >= etc.
- The iterative or repetitive statements can be written between **Repeat For or Repeat While.**

Example 2: Write an algorithm to find whether a number is even or odd.

Solution:
```
Step 1:  Input the first number as A
Step 2:  if A % 2 = 0
             then print “Even”
             else
                 print “Odd”
Step 3:  End
```

Example 3: Write an algorithm to find the larger of two numbers.

Solution:
```
Step 1:  Input first number as A
Step 2:  Input second number as B
Step 3:  if  A > B
                 then  print A
             else if A < B
                 then print B
             else
                 print “The numbers are equal”
Step 4:  End
```

Example 4: Write the algorithm to convert the temperature on F (Fahrenheit) to C(Centigrade) using the formula C = 5/9 (F – 32).

Solution: Step 1: Read F

Step 2: C=5/9*(F - 32)

Step 3: Print C

Step 4: Stop

The input variable is F (temperature in oFahrenheit) and the output variable is C (temperature in oCentigrade).

Example 5: Write an algorithm to find the sum of first N natural numbers.

Solution: Step 1: input N

Step 2: set i = 0, sum = 0

Step 3: repeat Step 3 and 4 while i <=N

Step 4: set sum = sum + i

set i = i + 1

Step 5: print sum

Step 6: end

Example 6: Write an algorithm to compute the largest of three numbers.

Solution: Algorithm: Largest

This algorithm computes the largest of three numbers.

The variable names are:

a, b, c: type integer

large: type integer, storing the value of the largest number

Step 1: [Input three integers]

read a, b, c

Step 2: [Compute the largest of three numbers]

large =a;

if (b > large) large = b

if (c > large) large = c

Step 3: [Display the largest number]

Print (large)

Step 4: [Finished]

exit

The steps in an algorithm are normally sequence, selection, and iteration type statement.

- **Sequence** means that each steps of the algorithm is executed in the specified order. The algorithm in **Example 1** performs the steps in a sequential order.
- **Selection** or Decision statements are used when the outcome of the process depends on some condition. The general form is if condition

  then statement1

  else statement2

  For example, if x = y

  then print "Equal"

  else

  print "Not Equal"

The algorithm shown in **Example 2** contains decision statements.

- **Iteration** or **Repetition** involves executing one or more steps for a number of times. This can be implemented using constructs **Repeat-For, Repeat-While**. Repition occur one or more steps untill some condition is true. The algorithm shown in **Example 5** contains repetitive statements.



CHECK YOUR PROGRESS

Q.1: State True / False:

i) An algorithm solves a problem in a finite number of steps.

ii) The conditional statement usually contains relational operators.

iii) Repition occur one or more steps untill some condition is false.

Q.2: Write an algorithm for interchanging/swapping two values.

9.4 PSEUDOCODES

Pseudocode is a kind of structured english for describing algorithms. It allows the designer to focus on the logic of the algorithm without being

distracted by details of language syntax. At the same time, the pseudocode needs to be complete.

There are no standards defined for writing a pseudocode, because a pseudocode is not an executable program. Indentation is used to increase clarity. It helps even non-programmers to understand the logic of the problem.

The aim of pseudocode is to get the idea quickly and also easy to read without details. It is like a young child putting sentences together without any grammar. For simplicity, let us represent one's daily life work in the form of pseudocode:

Brush teeth

Wash face

Comb hair

See in mirror

Some examples of pseudocodes in terms of computer programming are given below:

Example 7: Write the pseudocode for finding the area of rectangle.

Solution: READ height of rectangle
READ width of rectangle
COMPUTE area as height * width

Example 8: Write a pseudocode for finding the average of 5 numbers.

Solution: **input** 5 numbers
sum = add numbers together
avg = sum / 5
display avg

Example 9: Write a pseudocode to read the marks of 10 students. If marks is greater than 150, the student passes, else the student fails. Count the number of students passing and failing.

Solution:
1) **Set** totalpass to 0
2) **Set** totalfail to 0
3) **Set** no.of students to 0
4) **While** no. of students < 10
   a) Input the marks
   b) **If** marks >=150

Set totalpass = totalpass +1

Else

Set totalfail = totalfail + 1

EndIf

EndWhile

5) **Display** totalpass, totalfail

6) **End**

Variable used in Example 9 are: totalpass, totalfail, no.of students, marks.

Some **advantages** of pseudocodes are given below:

- The language independent nature of pseudocode helps the programmer to express the design in plain natural language.
- It can be designed based on the logic of the problem without concerning the syntax or rule.
- It is compact in nature.

The main **disadvantages** of pseudocode are:

- It does not have any standard format or syntax of writing.
- It cannot be compiled or executed.



CHECK YOUR PROGRESS

Q.3: Is there any standard rule when writing pseudocode?

Q.4: What is pseudocode?

9.5 FLOWCHART

We have already acquainted with the meaning of “pseudocode” and “algorithm” in the previous section. Before we start coding a program it is necessary to plan the step-by-step solution to the problem. Such a systematic plan can be symbolically represented with the help of a diagram. This diagram is called a flowchart. A flowchart is a symbolic representation of a solution to a given task. In this section we will learn to draw flowcharts using various symbols associated with it.

A flowchart is a pictorial representation of an algorithm. It shows the logic of the algorithm and the flow of control. The flowchart uses symbols to represent specific actions and arrows to indicate the flow of control.

Normally, an algorithm is expressed as a flowchart and then the flowchart is converted into a program with any programming language. Flowcharts are independent of the programming language being used. Hence one can fully concentrate on the logic of the problem solving at this stage.

It is always recommended for a beginner, to draw flowcharts prior to writing programs in the selected programming language.

9.5.1 Symbols of Flowchart

Flowchart uses many standard symbols. The boxes which are used in flowcharts are standardized to have specific meanings. These flowchart symbols have been standardized by the American National Standards Institute (ANSI). The symbols of a flowchart include:

- **Start and End:** (or,Start and Stop)Every flowchart has a unique starting point and an ending point. The Start and End symbols are also known as terminal symbols and are represented as **ovals**, or **rounded rectangles.** Flowchart begins at the start terminator and ends at the stop terminator. The starting point is indicated with the word START inside the terminator symbol. The ending point is indicated with the word STOP inside the terminator symbol.
- **Input/Output:** Input/Output symbols are used to denote any input/output function in the program. These are represented using a **parallelogram** and are used to get inputs from the users or display the results to them.

  Thus, if there is any input to the program via an input device, like a keyboard, tape etc. it will be indicated in the flowchart with the help of the Input/Output symbol. Similarly, all output instructions, for output to devices like printers, monitors etc. are indicated in the Input/Output symbol.

- **Process:** A process or computation represents arithmetic and data movement instructions in the flowchart. It is generally represented using a **rectangle**. All arithmetic processes of addition, subtraction, multiplication and division are indicated in the process symbol. If there are more than one process instructions to be executed sequentially, they can be placed in the same process box (rectangle), one below the other in the sequence in which they are to be executed.
- **Decision:** The decision symbol is represented by using a **diamond.** It is used in a flowchart to indicate the point where a decision is to be made and branching done upon the result of the decision to one or more alternative paths. The criteria for decision making is written in the decision box.

  It is basically used to depict a Yes/No question or a True/ False test The two arrows coming out of it, one from the bottom point and the other from the right point, corresponds to Yes or True, and No or False., respectively. The arrow should always be labeled.
- **Flow lines/Arrows:** Flow lines (or, Arrows) are solid lines with arrowheads which indicate the flow of operation. They show the exact sequence in which the instructions are to be executed. The normal flow of the flowchart is depicted from top to bottom and left to right.
- **Connectors:** In situations, where the flowcharts becomes big, it may so happen that the flow lines start crossing each other at many places causing confusion. This will also result in making the flowchart difficult to understand. Also, the flowchart may not fit in a single page for big programs. Thus whenever the flowchart becomes complex and spreads over a number of pages connectors are used.

  The connector represents entry from or exit to another part of the flowchart. A connector symbol is indicated by a circle and a letter or a digit is placed in the circle. This letter or digit indicates

a link. A pair of such identically labelled connectors is used to indicate a continued flow in situations where flowcharts are complex or spread over more than one page. Connectors do not represent any operation in the flowchart. Their use is only for the purpose of increased convenience and clarity.

The flowchart symbols discussed above are given in together in figure 9.1:

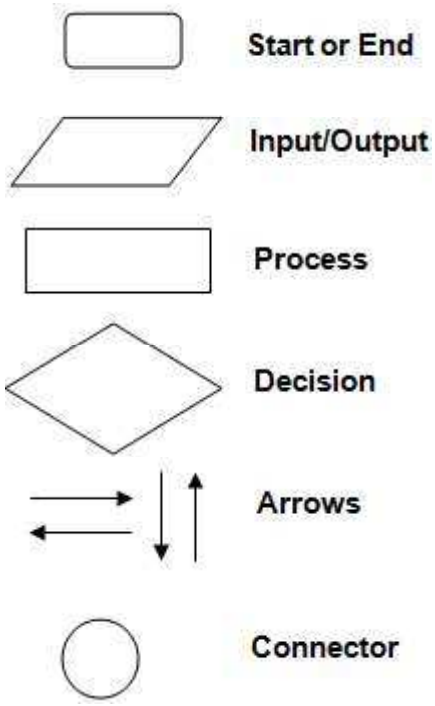


Fig. 9.1: Flowchart Symbols

9.5.2 Advantages and Limitations of Flowchart

Advantages of Flowchart: There are a number of advantages of flowcharts in problem solving.

- A flowchart being a pictorial representation of a program makes it easier for the programmer to explain the logic of the program to others rather than a program.
- It shows the execution of logical steps without the syntax and language complexities of a program.

- In real life programming situations a number of programmers are associated with the development of a system and each programmer is assigned a specific task of the entire system. Hence, each programmer can develop his own flowchart and later on all the flowcharts can be combined for depicting the overall system. Any problems related to linking of different modules can be detected at this stage itself and suitable modifications carried out. Flowcharts can thus be used as working models in design of new software systems.
- Flowchart has become a necessity for better documentation of complex programs.
- Flowchart also enables us to trace and detect any logical or other errors before the programs are written. Hence, a flowchart is very helpful in the process of debugging a program.
- Flowcharts are very helpful during the testing of the program as well as incorporating further modifications.

Limitations of Flowchart:

- Drawing flowchart for large complex problem is a laborious and time-consuming activity. Many a times, the flowchart of a complex problem becomes complex and clumsy.
- At time, a little bit of alteration in the solution may require complete re-drawing of the flowchart.



CHECK YOUR PROGRESS

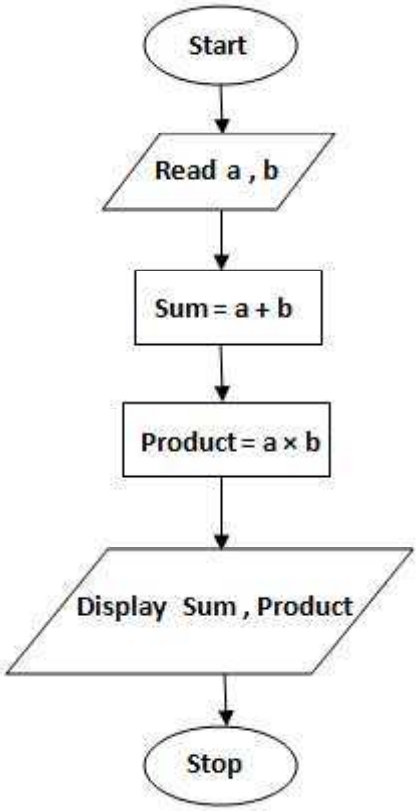
Q.5: What is a flowchart?

Q.6: Give any two advantages of flowcharts.

Here are few examples of flowcharts:

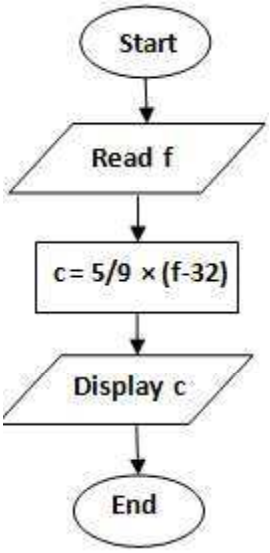
Example 1: Draw the flowchart to find the sum and product of given two numbers.

Solution:



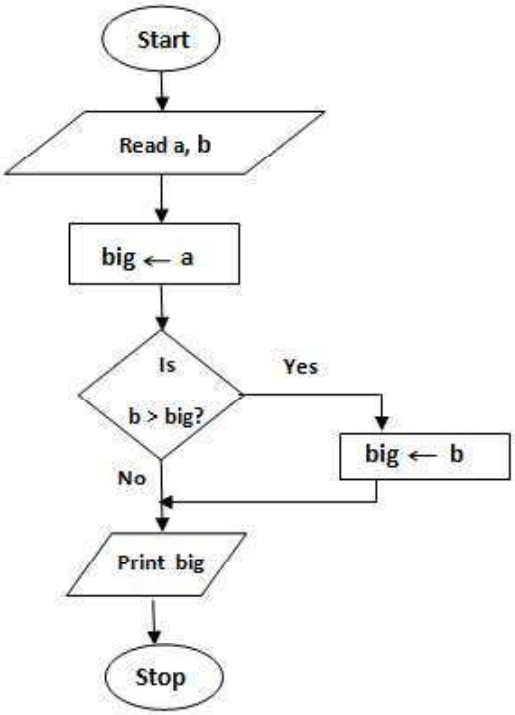
Example 2: Draw the flowchart to convert the temperature in Fahrenheit (0f) to Centigrade (0c).

Solution:



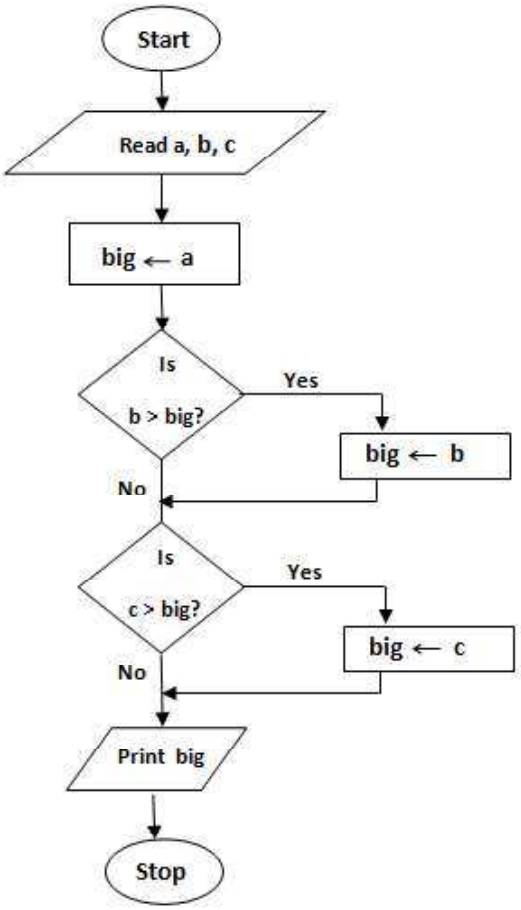
Example 3: Draw the flowchart to find the biggest of the two given numbers.

Solution:



Example 4: Draw a flowchart to find the biggest of three given numbers.

Solution:





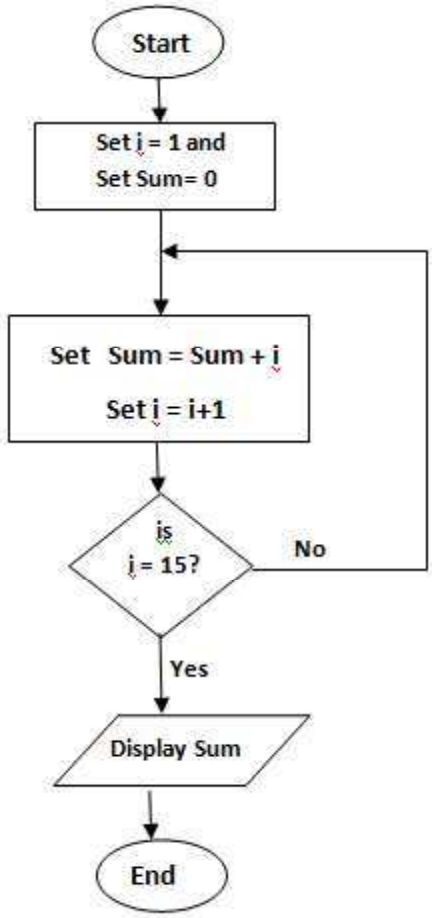
ACTIVITY 9.1

1) Draw a flowchart to calculate the salary of a daily wager.
[Salary=(no_of_hours*pay_per_hour)+Travel_allowance]

2) Draw a flowchart to subtract two numbers.

Example 5: Draw a flowchart to calculate the sum of first 15 natural numbers.

Solution:





CHECK YOUR PROGRESS

Q.7: In a flowchart, which symbol is represented using a rectangle?

a) Decision b) Connector

c) Input/Output d) Process

Q.8: Which one of the following is a graphical or symbolic representation of a process?

a) Algorithm b) Pseudocode

c) Flowchart d) Program

Q.9: The ..... symbol is always the first and the last symbol in a flowchart. (Fill in the blanks)

9.6 COMPUTER LANGUAGES

Language is a means of communication. To communicate our ideas, thoughts and emotions to other we use natural language such as Assamese, English, Hindi etc. In the same way we need computer language to instruct a computer to perform some task according to pur requirements. The main difference between a natural language and a computer language is that natural languages have a large vocabulary but most computer languages use a very limited or restricted vocabulary. Every natural language has some definite rules for writing the words and symbols in that language defined by the grammar of the language. Similary, there exists syntax rules for writing symbols and words of a computer language. All computer languages can be classified broadly into following three categories:

- Machine Language
- Assembly Language
- High-level Language

9.6.1 Machine Langauge

Every computer understands only one language without using translation program. This language is called ***machine language*** of the computer. The machine language of a computer is written normally as strings of binary 1s and 0s. The circuitry of a computer is wired in such a way that it recognizes the machine language instructions immediately, and converts them into electrical signals needed to execute them.

There are two parts in a machine language instruction. The first part is operation code (or opcode) that tells the computer what

function to perform, and the second part is operand that tells where to find or store the data to be manipulated. Hence, each instruction tells the computer what operation to perform and the length and location of the data fields involved in the operation. Every computer has a set of operation codes called its instruction set. Each operation code in the instruction set is meant to perform a specific basic operation or task. These operations include mainly Arithmetic, logical, braching, data movement operations between memory locations, registers and input/output devices.

Advantage of Machine Language:

- The only advantage of machine language is that machine instructions are executed very fast by the computer. This is due to the reason that machine instructions are in binary format (combinations of 0's and 1's) which are easily understood by the CPU without the need for any translation.

Disadvantages Machine Language:

- **Machine Dependent:** Machine language differs from computer to computer as the internal circuit design of every type of computer is different. Thus programs written in machine language of one computer will not work with computers of different circuit design.
- **Difficult to Program:** It is difficult to write programs in machine language because the programmer has to know details of hardware to write program. Again it is difficult for the programmer to memorize dozens of operation code numbers for commands in the machine instruction set. Sequence of binary codes are not easily readable. Again memory locations are manipulated directly, requiring the programmer to keep track of every memory location on the computer.
- **Error Prone:** The programmer has to remember a lot of codes(binary as well as hex code) to write a program in machine language which results in program errors.
- **Difficult to Modify:** It is very difficult to debug or modify machine language programs.

9.6.2 Assembly Langauge

In order to make programming easier and to communicate with the machine, more “human-like” languages were developed by computer scientists. These languages need to be translated into machine language so that computer can easily recognize and use. These language are known as **assembly language.**

A processor understands only machine language instructions, which are strings of 1's and 0's. A symbolic code written in assembly language can be translated to machine language with the use of an ***assembler,*** and the resultant machine language code can be directly executed on a computer. Assembly language is one step closer to human languages as the commands are like English words or combination of letters. For example, START, SET, LOD, SUB, ADD etc. This language is designed for a specific family of processors that represents various instructions in symbolic code and a more understandable form as compared to machine language.

Advantages of Assembly Language:

- The use of symbolic names for commands rather than arbitrary numeric codes gives assembly language an important advantage over machine language. For example, the assembly language code for addition is ADD which is binary 1110 in machine language. Similarly, for subtraction, we use SUB instead of binary 1111.
- Learning Assembly language makes one aware of how programs interface with operationg system, processor and BIOS.
- Knowing Assembly language helps programmers aware of how data is represented in memory and other external devises.
- Assemble language programs requires less memory and execution time.

Disadvantages of Assembly Language:

- In assembly language,there are no symbolic names for memory locations. We need to keep track of the exact memory location that a piece of data is stored.

- Assembly language code is machine dependent. We still need to rewrite every piece of code for every machine.
- Difficult to read as the command names are not always clear.
- Hard to maintain and debug. It is difficult in finding mistakes in machine code.
- Proper documentation is necessary to understand assembly language program.

9.6.3 High Level Language

A high level language enables a programmer to write programs that are more or less independent of a particular type of computer. Such languages are considered high level because they are closer to human languages. The first high level languages were designed in the 1950s. Now there are dozens of different high level programming languages, including COBOL, FORTRAN, ALGOL, BASIC, Pascal, Prolog, C, C++, Java, etc.

Before discussing the advantages of high level language, let us recall the common limitations of machine and assembly languages. They have the following common limitations:

1) They are machine dependent. A machine and assembly language program cannot be executed on any computer other than the one for which it is written.
2) The programmers of both the languages must have an adequate knowledge of the internal structure of the computer used.
3) It is difficult, error prone and time consuming to write programs in machine/assembly language.

Due to these limitations, machine and assembly languages are often referred to as ***low level programming languages.*** High level programming languages were designed to overcome these limitations of low level programming languages.

Advantages of High Level languages: The main advantage of high level languages over low level languages is that they are easier to read, write, and maintain. Programs written in a high level language

must be translated into machine language by a translator like compiler or interpreter. High level language have the following advantages over the machine as well as assembly language programming:

- High-level languages are machine independent. These are standardized in such as way that they may run on any machine/ operating system/platform and produce the same results when run anywhere.
- Symbolic names are used rather than direct memory locations.
- Logical and branching operations are included in high level language. Due to this, certain code might or might not be performed depending on certain conditions.
- The facility of adding comments helps programmer in documentation inside the program.
- With proper use of loop, the same block of code can be repeatedly executed depending on certain conditions in high level language.



CHECK YOUR PROGRESS

Q.10: Fill in the blanks:

a) ..... language run very fast because no translation program is required for the CPU.

b) Every computer has a set of operation codes called its .....

c) ..... language is the only language that is directly understood by the computer.

d) High-level languages are machine .....



9.7 LET US SUM UP

- An *algorithm* is a precise specification of a sequence of instructions to be carried out in order to solve a given problem.

- In the context of computer programming, an *algorithm,* is defined as a well-ordered collection of unambiguous and effectively computable operations, that when executed, produces a result and halts in a finite amount of time.
- *Pseudocode* consists of English-like phrases describing an algorithm. An ideal pseudocode must be complete, describing entire logic of the algorithm, so that it can be translated straightway into a program using any programming language. It facilitates programmers to focus on the logic of the algorithm.
- A flowchart is a pictorial representation of an algorithm. It shows the logic of the algorithm and the flow of control.
- The flowchart uses symbols to represent specific actions and arrows to indicate the flow of control.
- For ***Start*** and ***End/Stop, oval*** or ***rounded rectangle*** is used.
- For ***input/output*** opereation, ***parallelogram*** is used.
- For ***process*** or ***computation, rectangle*** is used.
- For ***decision***, the ***diamond*** symbol is used.
- For indicating ***flow of control, arrows*** with different heads are used.
- For indicating ***connections,*** labelled ***circle*** are used.
- Machine Language is the only language that is directly understood by the computer. It does not needs any translator program.
- Machine language codes are written as strings of 1's (one) and 0's (zero).
- Machine language programs are efficient for the computer to understand as it does not require any translator programs but very inefficient for programmers.
- Program of machine language run very fast because no translation program is required for the CPU.
- Assembly language is a low level programming language.
- Assembly languae is machine dependent. Each assembly language is specific to a particular computer architecture.
- In assembly language, symbolic names for commands are used rather than arbitrary numeric codes.

- A symbolic code written in assembly language can be translated to machine language with the use of an assembler.
- High level language is any programming language that enables development of a program in much simpler programming context.
- High level language is machine independent. I.e., independent of the computer's hardware architecture.
- High level language focuses more on the programming logic rather than the underlying hardware components such as memory addressing, register utilization etc.
- BASIC, C, C++, Java are popular examples of high level languages.



9.7 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: i) True; ii) True; iii) False

Ans. to Q. No. 2: Algorithm for inter changing/swapping two values:

Step 1: Input first number as A
Step 2: Input second number as B
Step 3: Set temp = A
Step 4: Set A = B
Step 5: Set B = temp
Step 6: Display A, B
Step 7: End

Ans. to Q. No. 3: There are no such set rules, but the code should provide clear descriptions of the algorithms being outlined.

Ans. to Q. No. 4: Pseudocode is not an actual programming language. It uses short phrases to describe an algorithm before you actually create it in a specific programming language. Once you know what the program is about and how it will function, then you can use pseudocode to create statements to achieve the required results for your program.

Ans. to Q. No. 5: A flowchart is a diagrammatic or symbolic representation of an algorithm. It uses various symbols to represent the operations to be performed.

Ans. to Q. No. 6: The advantages of flowcharts are:

i) Flowcharts are helpful for better documentation of complex programs.

ii) Flowchart also enables us in tracing and detecting any logical or other errors before the programs are written.

Ans. to Q. No. 7: d) Process

Ans. to Q. No. 8: c) Flowchart

Ans. to Q. No. 9: Start and End

Ans. to Q. No. 10: a) Machine; b) instruction set; c) Machine; d) independent



9.8 FURTHER READING

1) Goel, A. (2010): *Computer Fundamentals*; Pearson Education India.

2) Rajaraman, V. & Adabala, N. (2014): *Fundamentals of Computers*; PHI Learning Pvt. Ltd.

3) Sinha, P. K. & Sinha, P. (2010): *Computer Fundamentals* (Vol. 4); BPB publications.



9.9 MODEL QUESTION

Q.1: Define algorithm. How is it useful in the context of software development?

Q.2: What do you understand by the term pseudocode?

Q.3: Differentiate between algorithm and pseudocode.

Q.4: Write an algorithm to find the smallest of the given three numbers.

Q.5: Write the algorithm to find the average of given three values.

Q.6: Write an algorithm to find the sum of natural numbers upto N.

Q.7: Write an algorithm to find the area of a triangle.

Q.8: Write an algorithm to find the sum of a set of number.

Q.9: Write an algorithm to find the factorial of a given number.

Q.10: Write a pseudocode to compute the average of n numbers.

Q.11: What is a flowchart? What are the significance of flowchart?

Q.12: With the help of an example explain the use of a flowchart.

Q.13: How is a flowchart is different from an algorithm? Do we need to have both of them for developing a program?

Q.14: Draw the flowchart to find the area and circumference of a circle of radius r.

Q.15: Draw the flowchart to find the smallest of three given numbers.

Q.16: Discuss the advantages and limitations of flowchart.

Q.17: Describe the symbols used in flowchart.

Q.18: List the relative advantages and disadvantages of Machine and Assembly language programming.

Q.19: Briefly discuss the advantages of high level language.

Q.20: What are the different types of computer language? Briefly discuss.

*** ***** ***

UNIT 10: FILE STRUCTURE

UNIT STRUCTURE

10.1 Learning Objectives

10.2 Introduction

10.3 Data and Information

10.4 Records

10.5 Files

10.5.1 Operation on Files

10.6 Database Approach

10.6.1 Objectives of Database

10.6.2 Characteristics of Database

10.7 Let Us Sum Up

10.8 Answers to Check Your Progress

10.9 Further Reading

10.10 Model Questions

10.1 LEARNING OBJECTIVES

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

- define the term data and information
- describe the concept of files
- describe records and its types
- describe the operations on files
- describe database concepts.

10.2 INTRODUCTION

In the earlier units we have learnt about the basic components of the computer. We discussed computer classification, computer memory and buses. In the previous units we also learnt basic operating systems and introduction to programming.

This unit will help us to understand the concept of data and information. Also in this unit, we will be made familiar to records and their

types. In addition to these we will also learn about files and the operation on files. A brief introduction to database concepts is also included in this unit.

10.3 DATA AND INFORMATION

- **Data:** Data can be random facts or unorganized statements. Data can be something simple like numbers and figures and is useless until it is organized. Imagine a scenario of an India versus England World Cup cricket match. The batting scores or bowling scores corresponding to each individual player in the match are all data. In this example the individual player's batting and bowling scores are not of much use.

  Data refers to a set of facts or statements that are stored together but not according to any criteria or structure. When data is processed, organized, structured or presented in a given context so as to make it useful, it is called information. In the above example, the outcome of the match can be gathered by using the collected scores of players together and this is information. The batting and bowling averages can also be calculated and then later analyzed to check the performance of the players.

  In a general scenario, data can be considered as raw materials and inputs and information as the produced goods and output. Examples of data can be lists of some names with numbers, or names of some places, or some numbers etc.

- **Information:** Information is the data that is useful in some way. Organized or structured data which is useful in some way is information. For example, let us assume that we have the details of temperature for the whole of India for the last 100 years. This is an example of data. But if the data is used for extracting information like whether temperatures are rising or falling over the years or if it is related to global warming in any way then the data becomes information.

  Information can be obtained from data by either extracting some parts of the data, or by analyzing the data or adding relevant facts to

the data or by using some formula or technique on the data to get some meaningful information. Information retrieval from data is important. Examples of information can be connecting the list of names and numbers as student name and student roll number.

- **Difference between Data and Information:** Data and information are often interchangeably used. But there is a difference between the two terms. Data is the raw material from which information is produced. For example, data can be a series of alphanumeric characters like AA123, BC124, XY345, LK876, which have no meaning on its own. But when we provide the context along with the numbers that these are item codes used in a shopping mall then the information can be gathered that the item with the above item codes have been selected for purchase by a particular customer.

10.4 RECORDS

A record is a collection of fields of information. An example of a field of information can be a student name. A collection of such information like student's name, student's roll number, course name, percentage obtained is an example of a record. A record can also be said to be a set of fields of information.

- **Types of Records:** The fields of information in a record can be of different data types. It can be of numeric type, character type or alphanumeric type. Different or similar data types of fields of information can make a record. Let us take the above students record example, the students name and course name will be of character type, the students roll numbers can be of alphanumeric or numeric data type, and percentage obtained will be of numeric data type.

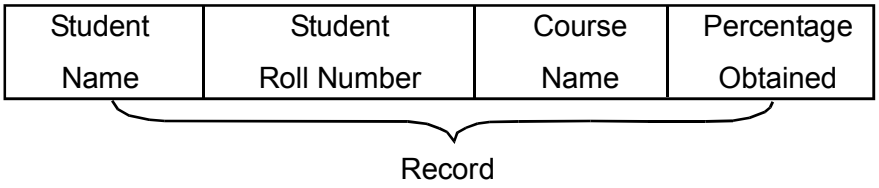


Fig. 10.1: Example of a Record

10.5 FILES

Files are collection of related records of information. The information is stored as records and a collection of these related records are stored as files. We can store several of the records together in a file. If we consider the earlier example, then a collection of different student records can be stored in a file. Let us look at the example below. Here five student's records are stored in the file.

Student Name	Student Roll Number	Course Name	Percentage Obtained
Rahul Sarma	BCA 16_01	BCA	78%
Gagan Deka	MCA 15_04	MCA	81%
Rahul Sarma	BCA 14_03	BCA	64%
Riya Gogoi	MSCIT 14_02	MSCIT	56%

Fig. 10.2: Example of Student Information File

- **Key:** A key is a unique identifier of a record. The key distinguishes each record uniquely. Suppose we need to find the marks of a particular student. Then how do we identify that particular student in a file? One way would be to check the name of the student. But, if the student file contains student records with similar names then we would require some other identifier to uniquely identify the student in the file. For the student file this unique identifier or the **key** could be the student roll number, as it is unique for each student.

  Let us now look at the above Figure 10.2. Here, two students have the same name "Rahul Sarma" so if we want to distinguish between the two students we need some unique identifier. Roll number field is one such unique identifier to uniquely distinguish between the two students.

10.5.1 Operations on Files

The different operations that can be performed on files are basically categorized as either update operations or retrieval operations. Let us look at the different operations in detail.

- **Creation:** A file is created in the creation operation by giving the file a specific name. After it is created the file is allotted some space in the computer's memory.
- **Open:** This operation is used to open an already created file in either read mode or in write mode. In the read mode, changes are not allowed to be made to the data in the opened file. Whereas in the write mode, changes to the data are allowed.
- **Read:** A file may be opened in the read mode and the pointer here by default always points to the beginning of the file.
- **Write:** To write in a file the file is first opened and then the pointer is placed at the position where we intend to write. By default, the pointer points to the beginning of the file.
- **Repositioning within a File:** There is a file pointer called file seek, which is used to relocate the file pointer to the location where we intend to read or write.
- **Delete:** To delete a file first the file is searched using its name and then the selected file is deleted. After deletion, the memory space occupied by the file is now free for use for other files and processes.
- **Truncate:** The user may want to delete the contents of the file but not the file itself. Deletion of the file and then recreating it again wastes a considerable amount of time. Instead, we can use the truncate operation. By using this function, we can delete just the contents of the file and not the file itself. This operation will also adjust the file length to zero from its previous length.

Disadvantage of File Systems: The file system consists of a few files along with many application programs. In the beginning, files system was used to perform commercial transaction mostly and the preferred language to do so was COBOL. Many application programs were written to access data files and perform a specific task. Each of the application programs and their data files were independent of each other.

All these programs contain some amount of similar information like name or registration number. This information is common to all the files and is replicated three times for three functions. Suppose the student's names are updated and two more students are added in the file. But these changes may not be simultaneously updated in the other files. As a result, wrong information may be provided to the users. This is one of the most serious disadvantages of file systems among many.



CHECK YOUR PROGRESS

Q.1: What do you mean by data? How is data different from information?

Q.2: What are the different operations that can be done on files?

Q.3: Define record.

10.6 DATABASE APPROACH

Database is a collection of related data. Database models part of the real world called enterprise. Databases are organized by fields, records and files. Field is a single unit of information, a record is set of fields and a file is a collection of records. A database can be a collection of related files about anything. It can be a student information database of a university, an employee database of an organization or a sales information database of a commercial industry.

Database management systems (DBMS) are some computer software applications that help the users to store and analyze data. A DBMS is a collection of data, software, hardware, users and applications that help to create and manage a database. Some well-known DBMSs are MySQL, Microsoft SQL Server, and Oracle among many.

Example: Let us look at the figure below about a student information database system in a university. We have four files in the database named ‘‘STUDENT_DATABASE’’. The first is ‘‘STUDENT_INFO’’ file which keeps all the information about the students. The second is ‘‘COURSE_INFO’’ file

which keeps track of the courses offered by the university. The third is "GRADE_INFO" file which keeps the grade details of the students. And the fourth is "LIBRARY_INFO" file which keeps details of the books taken by the students.

STUDENT_DATABASE

STUDENT_INFOR	
STUDENT_ID	**STUDENT_NAME**
001	ABC
002	DEF
003	GHI

COURSE_INFOR	
COURSE_ID	**COURSE_NAME**
11	BBP
13	BA (EDUCATION)

GRADE_INFOR			
STUDENT-ID	**COURSE-ID**	**SEMESTER**	**GRADE**
001	11	1	A
001	11	2	B
002	13	1	A
003	13	1	C

GRADE_INFOR				
BOOK_ID	**STUDENT_ID**	**BOOK_NAME**	**BOOK_AUTHO**	**DUT_DATE**
00001	001	AB	XY	20-04-2015
00002	003	ER	EN	12-05-2015
00003	001	RT	JH	05-03-2015
00004	002	GH	RT	01-07-2015
00005		AS	EG	
00006	001	UI	YT	15-04-2015

Fig. 10.3: Example of a Database

10.6.1 Objectives of Database

Some of the objectives of database are:

- To provide efficient storage, management and retrieval of data
- Reduce of data redundancy
- Providing data independence
- Sharing of data
- Providing data security from unauthorized access and use
- Handling different user views of the same stored data
- Being adaptable and scalable according to user requirements

10.6.2 Characteristics of Database

The main characteristics of database are as follows:

- **Concurrent Use:** Several users are able to access a database concurrently. The database is able to retrieve information for different queries from different users using the same data. The data is centrally controlled and there is no redundancy of data. The figure below describes concurrent use of data.
- **Structured and Described Data:** The essential information about database is that it not only stores and maintains the data but also the entire description and definition of the data. These descriptions are basically details about the structure, the type and the format of all data and the relationship between the data.
- **Data Integrity:** Data integrity refers to the reliability and quality of data in a database system. It also includes the protection of data from unauthorized access and usage.
- **Transactions:** A transaction is a set of actions that are performed within a database to take from one consistent state to another. Any data in between these two states are inconsistent. A transaction can be thought of as an atomic statement, which cannot be divided any further. In a transaction, either all the actions are performed or none are performed.

- **Data Persistence:** Data persistence means that in a database all data is maintained as long as it is not deleted explicitly.
- **Data Views:** A database has many users and each of these users depending on the access rights and queries forwarded have their own individual views. These views can be of the whole data, a subset of the data or some kind of derived data.

Advantages of Database:

- Helps in controlling redundancy
- Concurrent access of data is possible
- Data integrity can be imposed
- Sharing of data is possible
- Protection from unauthorized access and usage of data
- Data recovery and backup is possible
- Time and cost needed is less
- Tools available for easy access of database
- Inconsistency can be avoided

Disadvantages of Database:

- More complex than file systems
- Performance time needed for simple applications may be more
- Cost may be high depending on functionality provided



EXERCISE - 10.1

1) Explain two scenarios you observe around your daily life where the use of database would be recommended. Mention why database is recommended.
2) Also explain another two scenarios where usage of file would be recommended but not database according to you. Give reasons.



CHECK YOUR PROGRESS

Q.4: What do you mean by database and DBMS?

Q.5: Mention a few characteristics of database.

Q.6: Give two applications of database.

Q.7: Give two examples of DBMS software.



10.7 LET US SUM UP

- Data can be random facts or unorganized statements. Data is the raw material from which information is produced.
- Data refers to a set of facts or statements that are stored together but not according to any criteria or structure. When data is processed, organized, structured or presented in a given context so as to make it useful, it is called information.
- A record is a collection of fields of information. The fields of information in a record can be of different data types
- A key is a unique identifier of a record. The key distinguishes each record uniquely.
- Files are collection of related records of information. The information is stored as records and a collection of these related records are stored as files.
- Database is a collection of related data. Database models part of the real world called enterprise. Databases are organized by fields, records and files. Field is a single unit of information, a record is set of fields and a file is a collection of records.
- Database management systems (DBMS) are some computer software applications that help the users to store and analyze data. A DBMS is a collection of data, software, hardware, users and applications that help to create and manage a database.

10.8 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: Data is the raw material from which information is produced. For example, data can be a series of alphanumeric characters like AA123, BC124, XY345, LK876, which have no meaning on its own. Organized or processed data is information. So when we provide the context along with the numbers that these are item codes used in a shopping mall then the information can be gathered that the item with the above item codes have been selected for purchase by a particular customer.

Ans. to Q. No. 2: The different operations that can be performed on files are: creating a file, opening a file, reading a file, writing a file, repositioning the file pointer within a file deleting a file and truncating a file.

Ans. to Q. No. 3: A record is a collection of fields of information.

Ans. to Q. No. 4: Database is a collection of related data. Database models part of the real world called enterprise. Databases are organized by fields, records and files. Field is a single unit of information, a record is set of fields and a file is a collection of records. Database management systems (DBMS) are some computer software applications that help the users to store and analyze data. A DBMS is a collection of data, software, hardware, users and applications that help to create and manage a database.

Ans. to Q. No. 5: A few characteristics of database are: Concurrent Use: Several users are able to access a database concurrently, Structured and Described Data: The essential information about database is that it only stores and maintains the data but also the entire description and definition of the data, Data integrity: Data integrity refers to the reliability and quality of data in a database system, Transactions: A transaction is a set of actions that are performed within a database to take from one consistent state to another, Data Persistence: Data persistence means that in a database all data is maintained as long as it is not deleted explicitly, and Data Views: A

database has many users and each of these users depending on the access rights and queries forwarded have their own individual views.

Ans. to Q. No. 6: Two applications of database are: 1.Banking: Database used for customer information, accounts, and loans, and banking transactions and 2.Airlines: Database used for reservations and schedule information.

Ans. to Q. No. 7: Two examples of DBMS software are MySQL and Oracle.



10.9 FURTHER READING

1) Data, C. J. (1975): *An Introduction to Database Systems*; Addison-Wesley.
2) Elsmari, R. & Navathe, S. (2000): *Fundamentals of Database Systems*; Pearson Education Publication.



10.10 MODEL QUESTIONS

Q.1: How is information different from data? Explain in detail.

Q.2: What is file system? Give the operations that can be performed on files.

Q.3: Explain heap file organization with an example.

Q.4: Give the advantages and disadvantages of sequential file organization.

Q.5: How is hash file organization better than heap and sequential file organization?

Q.6: What is database? What do you understand by DBMS?

Q.7: Mention some advantages and disadvantages of database.

Q.8: Give reasons why you think database is necessary.

Q.9: What are the objectives of database? Explain in brief.

*** ***** ***

UNIT 11: COMPUTER SECURITY

UNIT STRUCTURE

11.1 Learning Objectives

11.2 Introduction

11.3 Computer Security

11.4 Malware

11.5 Computer Viruses

11.6 Worms

11.7 Other Threats

- 11.7.1 Trojan Horses
- 11.7.2 Spyware
- 11.7.3 Adware
- 11.7.4 Scareware

11.8 Counter Measures

- 11.8.1 Antiviruses
- 11.8.2 Deception Techniques
- 11.8.3 Working of an Antivirus

11.9 Let Us Sum Up

11.10 Answers to Check Your Progress

11.11 Further Reading

11.12 Model Questions

11.1 LEARNING OBJECTIVES

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

- learn about computer security and its goals
- describe different threats to computer security
- learn the classification of the threats
- counter measures of threats.

11.2 INTRODUCTION

The advancement of computer networking and information technology has been explosive in recent times. This technology has been propagated

around the world rapidly and has affected every aspect of day-to-day life. Most computer facilities continue to protect their physical machine better than their data, but in reality the value of the data is greater than the value of the hardware. Throughout this chapter, the discussion of computer security emphasizes the problem of protecting information from unauthorized disclosure, or information secrecy.

11.3 COMPUTER SECURITY

The state of being free from danger or threat is called security. So computer security means protection of the computer systems from theft or damage to its hardware or to its software or to the information it stores, as well as from disruption or misdirection of the services it provides. More precisely we can explain with the CIA triad of information security, which maintains these three security goals

- **Confidentiality:** It means protecting the information from disclosure to unauthorized parties.
- **Integrity:** It means keeping the data pure and trustworthy by protecting system data from intentional or accidental changes.
- **Availability:** It refers to ensuring that authorized parties are able to access the information when needed.

Let us describe a day-to-day example to clear this CIA triad. Suppose I own an bank account and I am not willing to share my ATM card number and its pin bank statements also. So me and my bank is protecting the information so its confidentiality. One day I was making a online transaction to a 3rd party of rupees 1000.00 but in the mean time somehow someone tampered the information and made the amount rupees 10000.00. I don't want to have such tampering with my confidential data. That is Integrity. I want my bank account to be accessed and manipulated only by me not by others, my bank should protect my account from unauthorized attempt of access. This is an example of Availability.

11.4 MALWARE

If we can ensure these goals for our system then we can say our system is protected and any threat to these is a threat to computer security. The threats to computer security can be in the form of unauthorized access or deception or disruption or usurpation to the services or data. Generally these violations to security are caused by some malicious software. A software is potentially malicious if it can be used to harm either the computer on which it is hosted or another computer. Software may also be considered malicious if it is designed to install itself on a computer without the permission of the owner of that computer, particularly if it does so in a way that may compromise the security of the computer. Malicious may be loosely interpreted. A piece of software may be considered malicious even though it may have been launched with the intention of providing an arguable benefit. These types of software are referred as **malware** (= **mal** icious + soft **ware**) in general.

Malware can be classified as–

- Transient– They need a host to exist (Viruses, Trapdoors)
- Resident– They can exist on their own (Worms)

Most malicious software takes advantage of the vulnerability-exploit cycle in software development if it is not managed properly. Vulnerability is a system susceptibility or flaw, and vulnerability-exploit cycle is the continuous process of identifying, classifying, remediating, and mitigating vulnerabilities as they are discovered.

In this unit our main focus will be mainly on malware and its exploiting techniques, other threats to computer security is beyond the scope of this unit.

11.5 COMPUTER VIRUSES

A computer virus is an executable program that replicates itself or attached to a file where it infects the computer when the file is executed. Sometimes it replaces an existing program also. There are many techniques used by virus to spread or replicate or attached to a file, but the self replicating

nature is the main criterion that distinguish a virus from other software. And not all viruses show harmful activities on system.

But why is such malicious software is termed with the biological term "Virus"? This is so because most computer viruses are parasitic like biological virus i.e.: they work by attaching themselves to a carrier object (e.g.: a file). And when this carrier object is transmitted to other systems the virus is also transmitted and it is activated when the host program is executed. After activation the computer virus like its biological one search for another host and gets attached to it.

Working of a Computer Virus: A virus does not appear as an object in itself. A virus always resides hidden in some useful object and it may perform the main parts of a virus' code which are the **Replication routine** and the **Payload Routine.** A virus is not a virus without the replication routine, but pay load routine for a virus is optional.

- **Virus replication:** The replication routine contains a code that locates suitable objects to attach the virus to and to copy the virus to these elements. There are different techniques for this purpose.

  Virus' code can be executed when the object is executed. Viruses that infect program files may attach the virus code at the start or at the beginning of the program and patch the entry point so that when the program is run the virus code is executed first. The virus usually transfers control to the original program when it has finished its tasks. This ensures that the original program works properly and the virus avoids detection.

- **Payload Execution:** Payload routine of a virus is optional, some of the viruses may not have the payload routine which give them the efficiency to replicate more quickly. Payload part of the virus code generally contains the malicious activity such as slowing down the PC, deleting system files, making short cuts of every folder, swiping out the memory etc. The payload routine usually contains some activation criteria that must be met before the payload activates. This gives the virus some time to spread without detection. The activation criteria may be any data that can be found in computer

like a particular date after infection or a lapse of time period or execution of certain application etc.

From the above discussion one thing is clear: a virus always need a host to spread and it cannot spread on its own. It needs some activation criterion to be fulfilled. Then only it can be executed or it can take over the computer resources. But there are some other types of malicious software commonly mistaken as virus but unlike the viruses. They can survive without

11.6 WORMS

A **worm** is more independent than a virus. A pure worm works by itself as an independent object. The main feature that separates worms from viruses is that worms are self replicating. A worm can initiate network communication by itself and send copies to other computers. So there is no need for activation criteria or host process. As worms can spread rapidly across a network infecting every PC so they are the most notorious of malwares. There are some malicious software that resemble both viruses and worms.

One common example of this kind is mass mailing worms. They are attached to documents or other object like a virus, but uses email clients to mass mail themselves in a worm like way. This type of worm needs some kind of activation as for example, in mass mail worms can only activated when a receiver opens an attachment. It makes the worm slow but as it initiate the network by its own they are much faster transmitted than viruses. Worms and viruses are the most common wide-spread malicious software .There are some other programs also which work may not be so vicious like the virus or worm, but still they are malicious.



CHECK YOUR PROGRESS

Q.1: What are the three security goals?

Q.2: What is computer virus?

Q.3: Differentiate between worm and virus.

11.7 OTHER THREATS

In this section we will go briefly into the other malicious software other than virus and worm which may not exploit or damage the system like virus or worms do but may be a potential threat in point of view of security.

11.7.1 Trojan Horses

The term Trojan horse comes from the Trojan war described in Greek mythological epic. As the tale describes in the Trojan War the Greeks were unable to enter the city of Troy despite 10 years of long siege and finally they constructed a large wooden horse and hid some soldiers in it. The citizens of Troy thought it was just a mere wooden horse and they let it in their city and in the night the soldiers inside the horse opened the city gates, which eventually became the downfall of their city. The wooden horse is coined as Trojan horse in the epic. The Trojan horses in computer security also serve the same purpose.

Trojan horses are applications which appear to be useful, legitimate applications, but run malicious, hidden code. The application may appear to perform tasks, such as scanning for spyware, but actually steal personal data from your system. Executing Trojans may result in changing desktop wallpaper, creating popups, or even deleting files. One of their most popular uses is creating backdoors to allow malicious users to gain access to files and information. Unlike viruses or worms, Trojans cannot infect other systems.

11.7.2 Spyware

Spyware are the software that collects user sensitive information like browsing and searching habit, password, social interactions without your knowledge and uses such information to gain profit. Information collection by spyware is done using technique like key logging, browser activity monitoring. Spyware is meant to

make money by using personal information so it does not exploit the PC like other malware do. In fact, many spyware may be running in our PC without our knowledge.

11.7.3 Adware

Adware are the software that supports advertising. In general it is the entire program that downloads or displays unwanted banner advertisements in the software being used. Adware can be also used to collect user personal data and based on these data the adware shows up with classified advertisement. This type of software doesn't harm the system but collecting information and running advertisement without user knowledge is a threat.

11.7.4 Scareware

Scareware is a new type of attack, where a user is tricked into downloading a software that appears to be a useful one. After installing, it shows that the machine is infected and it can clean all the infection if you purchase the full license version of the software. These Scareware applications are malware that hold the PC hostage until people don't pay the ransom. In most of cases, these Scareware can't be uninstalled also.

11.8 COUNTER MEASURES

‘‘Prevention is better than cure’’. To detect and eliminate security threat is always tougher than to prevent it. We can say a standalone system is more secure than any other. But it is not ideal scenario that a PC will never expose to security threats. A threat to computer security may be in many forms and hence we have to deploy different technique also to encounter them. But in this chapter we will mainly focus on counter measures techniques for malware.

11.8.1 Anti Viruses

Malware uses different strategies for deception and replication. Hence, for detection and elimination of different malware involved, different techniques and tools are used which is impossible to be done manually. A special kind of software is developed for this purpose commonly known as Antivirus. So, antivirus software is a program or set of programs designed for searching, preventing, detecting and eliminating malware. The detection techniques of anti viruses are generally based on malware deception strategies. So let's go through a brief overview of different deceptions strategies used by the malware.

11.8.2 Deception Techniques

Different malicious software uses different strategies to spread or replicate them and exploit the system. Deception helps malware to stay in the system and replicate for a longer time by disguising as a part of the host program. Malware employ different kinds of deception to hide them such as

- **Overwriting:** In this technique the virus overwrites files with their own copy. It's a primitive technique but it is simple and easy to implement. This type of viruses can be disinfected from system simply by deleting the infected files.
- **Companion Infection:** In this approach the virus disguise itself as a companion of .exe file by naming it with same base name as the targeted program, but use a .COM extension instead of .EXE, because Windows gives priority to a file with the .COM extension over a file with the .EXE extension.
- **Appending:** In this method, a jump (JMP) or Goto instruction is encoded at the start of the of the host program that will point at the end of the original host. And the starting of virus code is lied there.
- **Pre-appending:** In this strategy the virus code has been inserted at the front of host programs. This is a simple efficient method.

- **Cavity filling:** Here the virus tries to install itself in the empty space of the host while not damaging the actual program itself. An advantage of this technique is that the virus does not increase the length of the program but it also minimizes a number of possible hosts, cavity viruses are rare.
- **Compressing:** Generally after infection size of the affected files increase, but by using compression techniques in this method size of the affected files has reduced so that change in file become negligible.
- **Encryption:** In this scenario the virus consists of a constant decryptor or set of decryptor, followed by the encrypted virus code, after decryption of the encrypted code with different decryptor yield different virus. This type of malware is called Oligomorphic virus.
- **Boot Sector Execution:** Due to the executable nature of master boot record (MBR) and partition boot sector (PBS) if any executable code is present in these section it will execute when machine booted and hence bypass all the security measure
- **Macros:** Macros are generally targeted for Microsoft word or similar products and damaged only such documents. They are embedded in such application and while activation triggered some other application or insert unreadable character in the documents.
- **Malicious Mobile Code (MMC):** Mobile codes are lightweight programs that are downloaded from a server and executed with minimal or no user intervention on your local machine. JavaScript scripts, VB Script, and ActiveX controls are some of examples of mobile code that we encounter while browsing. These code are used to monitor browsing activities, getting unauthorized access to local file system, showing pop ups, adding unwanted browser add-ons so. Spyware, adware, Trojan are such malware that uses MMC.

11.8.3 Working of an Antivirus

Antivirus software scans all the file system and search for virus definition in there and try to disinfect it by removing the malware code or deleting the infected file. Virus definition of a malware is signature by which the malware can be identified. The signature is determined based on the strategy adopt by the malware to replicate themselves. These definitions are provided by the antivirus company. Antivirus program try to match the files with these definition. The basic searching methods used by the antivirus can be broadly classified into five categories. These are:

1) **Signature Based Detection:** Once a virus is detected, it is analyzed precisely and a unique sequence of bytes extracted from the virus code. This string is the signature of that virus. The antivirus engine scans the binary code of files to find these strings? if it encounters with a known pattern, it alerts detection of the matching virus. Some of commonly known signature based techniques are:
   - **String Scanning Method:** Searches for sequence of bytes (strings) that are unique to a specific virus but can't be found in other programs. In this process a know virus is analyzed precisely and a unique sequence of bytes extracted from the virus code. For example a virus Die.448 has a unique bit string like B440 B9E8 0133 D2CD 2172 1126 8955 15B4 40B9 0500 BA5A 01CD in its code by which we can detect it.
   - **Wildcards Method:** Including a wild card character like ? or * allows skipping bytes or byte ranges of code in the signature code and try to match next which will enable the antivirus engine to detect slightly modified virus. For example: presence of a wildcard character * in the malware code means that scanner will try to match the next byte in the file.
   - **Mismatches Method:** This method allows a negligible number of bytes in a signature string to be of random values,

regardless of their position. For example virus X has signature string of 2A 80 4D 01 F6 and following strings contains the same signature values in different places and also mismatched by 2.

11 2A 32 01 17 80 4D 24

0A D9 1E 17 80 F6 24 AA

But using Mismatched method we can still say that these two strings contains the virus X.

- **Generic Detection Method:** This technique uses one common generic string as a signature string to detect several or all known variants of a family of viruses. When a virus has lots of its variant then the variants are analyzed to extract a common string that can identify the whole genre. This type of technique often uses the wildcard and mismatched technique to determine a common unique string.
- **Bookmarks Method:** Bookmark method can ensure more reliable detection of viruses while it is used along with string scanning by negating the false positive. The byte distance between the start of the virus code and start the signature string is vary in different viruses. So it can be used as bookmark particular viruses when string bases scanning are used.
- **Smart Scanning:** Smart scanning optimizes the signature based detection. Some new variants of viruses try to hide their signature by inserting junk code or instruction in their code which have no effect in execution. Smart scanning skip junk instructions, such as NOPs (No Operation) in the host file to obtain the actual virus body.
- **Skeleton Detection:** The scanner parses the statements of a file line byline and drops all nonessential statements like statements which are not part of the virus or any blank gaps before actual procedure start and try to find essential virus code for execution of the virus which is called skeleton of

the virus. An antivirus named as Kaspersky used this particular method very efficiently to detect macro viruses.

2) **Heuristic Based Detection:** This type of detection helps in detecting new unknown or altered version of malware, but it produces false positive often. So is used with along with signature based scanner. As the name suggest this detection mainly focus on heuristics (experimental trial and error methods) like analysis of file structure and the code organization of the virus or CPU emulation of virus code .Some common heuristic based detection methods are

   - **Heuristics Analysis:** It is an expert based analysis that determines the susceptibility of a malware code by running it on virtual runtime environment and analyzing system towards particular risk using various decision rules or weighing methods. Multi Criteria analysis (MCA) is one of the means of weighing.
   - **Virus Specific Detection:** There are cases when the standard algorithm of the virus scanner cannot detect a particular malware. In such cases, a new detection code must be introduced to implement a virus specific detection algorithm to detect only specific malware. This method includes.
     - **Filtering:** A virus only effects a particular type of known objects. So viruses can be classified according to its infection type. For example, executable viruses infect only programs such as .EXE and .COM,, macro viruses only attack to files or that can perform macro statements. So whenever a particular set of files are scanned we have to check only the specific virus signature that is relevant to those files. Such filtering minimize the searching time
     - **Static Decryptor Detection:** Some viruses encrypt their body to bypass string matching so they have a decryptor in the virus code. Detection of this Decryptor code also

result in efferent detection but it also produces false negative and false positive if not used with specific types of viruses which used that particular decryption .So this method is used after filtering of encrypted virus like Oligomorphic virus.

- **XRay scanning:** This method tries to find the weakness in encryption algorithm of viruses. It is also virus specific can be only use with encrypted viruses and give more suitable result than Static Decryptor Detection as its try to find the encryption strategy.

3) **Behavioral Based Detection:** When a malware attach to its host or execute its code some particular behavioral pattern or dynamic signature they exhibit which is not common to other programs and therefore by observing such signature we may detect them. But as these signatures are dynamic we need a continuous process for monitoring such abnormal behavior. A malware may produce dynamic signature like–

- Opening an executable file, with both read and write permission.
- Reading the portion of the file header containing the executable's start address
- Writing the same portion of the file header.
- Seeking to the end of the file.
- Appending to the file.

This detection method monitors a running program's behavior in real time, watching for suspicious activity like the dynamic signature mentioned above. If any such activities are detected, it report to the antivirus engine that as a response prevents such operations from being executed or terminate the program. This detection method can only work when malware perform some malicious activity during runtime.

4) **Sandbox Detection or Emulation Detection:** This type of detection is also based on behavioral traits of the malware. Here

the malware code under analysis is executed in an emulated environment or sandbox and the antivirus try to detect it during runtime based on behavioral traits. But unlike the Behavioral Based detection malware code, it is run in a virtual environment and not in real system and hence, minimizes the risk of being affected.

5) **Data Mining Techniques:** It is the recent technology used for detecting malware which uses a data mining techniques like analyzing a very huge sets of infected data and try to find out particular behavioral pattern or try to find out a common signature strings from a large number of different variants of malware and using this signature try to detect a known malware or to classify a code for malware.

After detection, antivirus program try to remove the malware code from host program if it is not possible to remove the viral code and the host program is not so significant than the antivirus delete the infected file itself. On the other hand, if the infected file is important from system point of view then it pushes it to quarantine.



CHECK YOUR PROGRESS

Q.4: What are Trojan horses?

Q.5: Differentiate between spyware and adware.

Q.6: Define antivirus.



11.9 LET US SUM UP

The basic goal of computer security is Confidentiality, Availability and Integrity. Violation of these goals leads to a security treats. Most common security threats are of in form of malware .Malware is a term for all malicious programs, such as viruses, worms, Trojan horses, spyware, adware and scareware which are designed to infect and perform malicious activity on computers. Antivirus software is designed to protect system against malware.

It is available as a standalone product and is also included in most security software packages. They use different static and dynamic method to identify malware and neutralize them.



11.10 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: The three security goals are:

- **Confidentiality:** It means protecting the information from disclosure to unauthorized parties.
- **Integrity:** It means keeping the data pure and trustworthy by protecting system data from intentional or accidental changes.
- **Availability:** It refers to ensuring that authorized parties are able to access the information when needed.

Ans. to Q. No. 2: A computer virus is an executable program that replicates itself or attached to a file where it infects the computer when the file is executed.

Ans. to Q. No. 3: A worm is more independent than a virus. A pure worm works by itself as an independent object. The main feature that separates worms from viruses is that worms are self replicating. A worm can initiate network communication by itself and send copies to other computers. So there is no need for activation criteria or host process. As worms can spread rapidly across a network infecting every PC so they are the most notorious of malwares.

Ans. to Q. No. 4: Trojan horses are applications which appear to be useful, legitimate applications, but run malicious, hidden code. The application may appear to perform tasks, such as scanning for spyware, but actually steal personal data from your system. Executing Trojans may result in changing desktop wallpaper, creating popups, or even deleting files.

Ans. to Q. No. 5: Spyware are the software that collects user sensitive information like browsing and searching habit, password, social interactions without your knowledge and uses such information to gain profit. Adware are the software that supports advertising. In

general it is the entire program that downloads or displays unwanted banner advertisements in the software being used.

Ans. to Q. No. 6: Antivirus software is a program or set of programs designed for searching, preventing, detecting and eliminating malware.



11.11 FURTHER READING

1) Canavan, J. E. (2001): *Fundamentals of Network Security*; Artech House.
2) Stallings, W. (2006): *Cryptography and Network Security: Principles and Practices*; Pearson Education India.



11.12 MODEL QUESTIONS

Q.1: What is computer security?

Q.2: What are the goals of computer security?

Q.3: Give an example of computer virus. Explain its working principle.

Q.4: What is the difference between computer virus and worms?

Q.5: Why is scareware considered as malware?

Q.6: Name any antivirus and explain how it works.

Q.7: Briefly describe different detection mechanism used by antivirus software.

Q.8: Why deception is needed for malware? Explain various deception strategy used by malwares.

Q.9: Is deception strategy of malware and detection method of antivirus related to each other? Explain.

Q.10: List out some of the common malware we encounter and list out whether they are viruses, worms, Trojan or any other malicious software.

*** ***** ***