
UNIT 4: SOFTWARE

Contents

- 4.0 Aims and Objectives
- 4.1 Introduction
- 4.2 Types of System Software
 - 4.2.1 Function of Operating System
 - 4.2.2 Micro Computer Operating Systems
 - 4.2.3 Operating System Capabilities
 - 4.2.4 Language Translation and Utility Program
- 4.3 Application Software
 - 4.3.1 Generation of Programming Languages
 - 4.3.2 Popular Programming Languages
 - 4.3.3 Fourth Generation Languages
 - 4.3.4 New Software Tools and Approaches
- 4.4 Summary
- 4.5 Answer to Check Your Progress Exercise

4.0 AIMS AND OBJECTIVES

After studying this unit, you will be able to explain

- what software is
- the major types of software
- the major functions of operating systems
- the major types of application software

4.1 INTRODUCTION

Software is a detailed instruction that control the operations of a computer system. Without software, computer hardware could not perform the task we associate with computers. The functions of software are to:

- 1) manage the computer resources of the organization
- 2) provide tools for human beings to take advantage of these resources; and
- 3) act as an intermediary between organization and stored information.

A software program is a series of statements or instructions to the computer. The process of writing or coding programs is termed programming and individuals who specialize in this task are called programmers.

The stored program concept means that a program must be stored in the computer's primary storage along with the required data in order to execute, or have its instructions performed by the computer. Once the program has finished executing, the computer hardware can be used for another task when a new program is loaded into memory.

4.2 TYPES OF SYSTEM SOFTWARE

There are two types of software: systems software and application software.

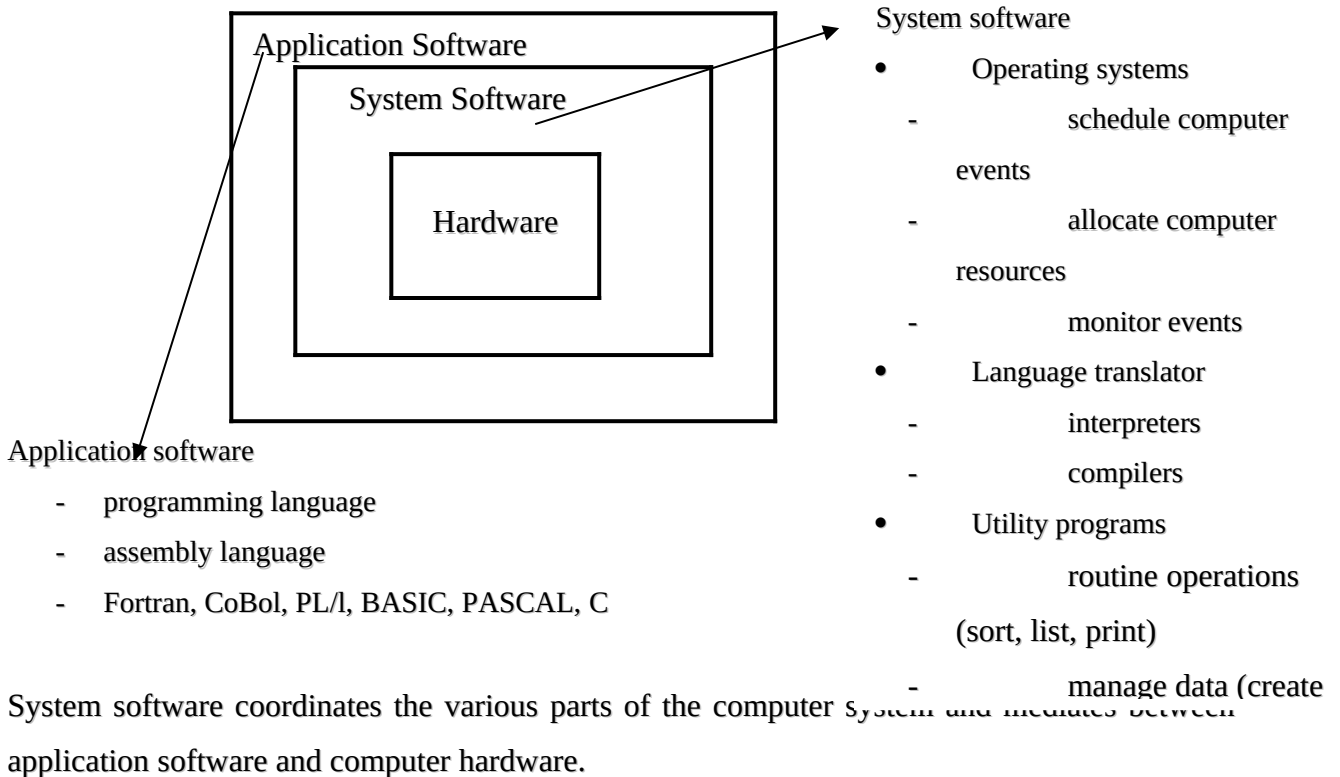
1. System Software

System software is a set of generalized programs that manage the resources of the computer, such as the central processor, communications links, and peripheral devices. Programmers who write system software are called system programmers.

Application software describes the programs that are written for or by users to apply the computer to a specific task. Software for processing an order or generating a mailing list is application software. Programmers who write Application software are called application programmers.

The types of software are interrelated and can be thought of as a set of nested boxes, each of which must interact closely with the other boxes surrounding it. The system software surrounds and controls access to the hardware. Application software must work through the system software in order to operate. End users work primarily with application software. Each type of software must be specially designed to a specific machine in order to ensure its compatibility. The following figure illustrates the relationship.

Figure 4.1 : The major types of software



The system software that manages and controls the activities of the computer is called operating system. Other system software consists of computer language translation programs that convert programming languages into machine language and utility program that perform common processing tasks.

4.2.1 Functions of the Operating System

Operating system is the system software that manages and controls the activities of the computer. One way to look at the operating system is as the system's chief manager. Operating system software decides which computer resources will be used, which programs will be run, and the order in which activities will take place.

An operating system performs three functions. It allocates and assigns system resources. It schedules the use of computer resources and computer jobs; and it monitors computer system activities.

- i) **Allocation and assignments:** The operating system allocates resources to the application jobs in the execution queue. It provides locations in primary memory for data and programs and controls the input and output devices such as printers, terminals, and telecommunication links.
- ii) **Scheduling:** thousand of pieces of work can be going on in a computer simultaneously. The operating system decides when to schedule the jobs that have been submitted and when to coordinate the scheduling in various areas of the computer so that different parts of different jobs can be worked on at the same time.
- iii) **Monitoring:** the operating system monitors the activities of the computer job and may also keep track of who is using the system, of what programs have been run, and of any unauthorized attempts to access the system.

4.2.2 Micro Computer Operating Systems

Like any other software, micro computer software is based on specific operation systems and computer hardware. A software package written for one microcomputer operating system generally cannot run on another. The microcomputer operating system they have distinct features – such as whether they support multitasking or graphics work that determine the types of applications they are suited for. The widely used microcomputers operating systems are: DOS, Windows 95, Windows NT, OS/2, Unix, and system 7.

- i) **DOS:** (Disk Operating System): is an operating system for 16 bit microcomputers based on the IBM personal computer standard. DOS was the most popular operating system for 16 bit microcomputers. It is still widely used today with more powerful microcomputers based on the IBM microcomputer standard because so much available application software has been written for systems using DOS. (PC-DOS is used exclusively with IBM microcomputers. MS-DOS, developed by Microsoft, is used with other 16 bit microcomputers that function like the IBM microcomputer. DOS itself does not support multitasking and limits the size of a program in memory to 640k.

DOS itself is command driven, but it can present a graphical user interface by using Microsoft windows, a highly popular graphical user interface shell that runs in conjunction with the DOS operating systems. Windows support multitasking and same form of networking but shares the memory limitations of DOS.

- ii) **Windows 95:** Microsoft's Windows 95 is a 32 bit operating system designed to remedy many of the deficiencies of DOS. A 32 bit operating system can run faster than DOS because it can address data in 32 bit chunks.

Windows 95 provides a streamlined graphical user interface that arranges icons to provide instant access to common tasks. It can support software written for DOS and Windows but it can also run programs that take up more than 640k of memory. Windows 95 features multitasking, multithreading (the ability to manage multiple independent tasks simultaneously), and powerful networking capabilities, including the capability to integrate fax, e-mail, and scheduling programs. The operating system requires a fast 386 or 486 microprocessor, 8 megabytes of RAM, and 24-48 megabytes of hard disk storage. Advanced Microsoft windows operating systems are widely in use today. To name a few: Windows 98, Window 2000, and Window XP.

- iii) **Windows NT:** is another operating system developed by Microsoft with features that make it appropriate for critical applications in large networked organizations. Windows NT uses the same graphical user interface as Windows but it has powerful multitasking and memory management capabilities. Windows NT can support existing software written for DOS and Windows, and it can provide mainframe like computing power for new applications with massive memory and file requirements. It can address data in 32 bit chunks if required and can even support multiprocessing with multiple CPUs.

- iv) **OS/2 (Operating System/2):** is a powerful operating system used with 32 bit IBM personal system/2 micro computer workstations that supports multitasking, networking, and more memory intensive applications than DOS.

OS/2 is being used for more complex memory intensive applications or those that require networking, multitasking or large programs. OS/2 supports multitasking, accommodates larger applications, allows applications to be run simultaneously, supports networked multimedia and pen computing applications, and is much more protected operating system.

This operating system requires powerful computer hardware – a minimum 80386 or 80486 microprocessor, 4 megabytes of RAM, and a 60 megabytes hard disk. OS/2 provides powerful system – like capabilities, such as multitasking and supporting multiple users in networks.

- v) **UNIX:** was developed at Bell laboratories in 1969 to help scientific researchers share data and programs while keeping other information private. It is an interactive, multi user, multi tasking operating system. Many people can use UNIX simultaneously to perform the same kind of task, or one user can run many tasks on UNIX highly supportive of communication and networking.

Unix was initially designed for minicomputers but now has versions for microcomputers, workstations, and mainframes. UNIX can run on many different kinds of computers and can be easily customized. It can also store and manage a large number of files. At present, UNIX is primarily used for workstations, minicomputers, and inexpensive multi-user environments in small businesses, but its use in large business is growing because of its machine – independence.

- vi) **System 7:** is the latest version of Macintosh system software, features multitasking as well as powerful graphics capabilities, and mouse driven graphical user interface. An extension of this operating system called Quick Time allows Macintosh users to integrate video clips, stereo sounds, and animated sequences with conventional text and graphics software.

4.2.3 Operating System Capabilities

How is it possible for 1000 or more users sitting at remote terminals to use a computer information system simultaneously? Can a computer execute only one instruction from one program at a time? How can computers run thousands of programs? The answer is that the computer has a sense of specialized operating system capabilities. These includes: multiprogramming; virtual storage, time sharing; and multiprocessing.

- i) **Multiprogramming:** the most important operating system capability for sharing computer resource is multiprogramming. Multiprogramming permits multiple programs to share a computer system's resources at any one time through concurrent use of a CPU. By concurrent use, we mean that only one program is actually using the CPU at any given moment but that the input/output needs of other programs can be serviced at the same time. Two or more programs are active at the same time, but

they do not use the same computer resources simultaneously. With multiprogramming, a group of programs takes turns using the processor.

Consider the following figure

Figure4.2: Single Vs Multiprogramming Environment

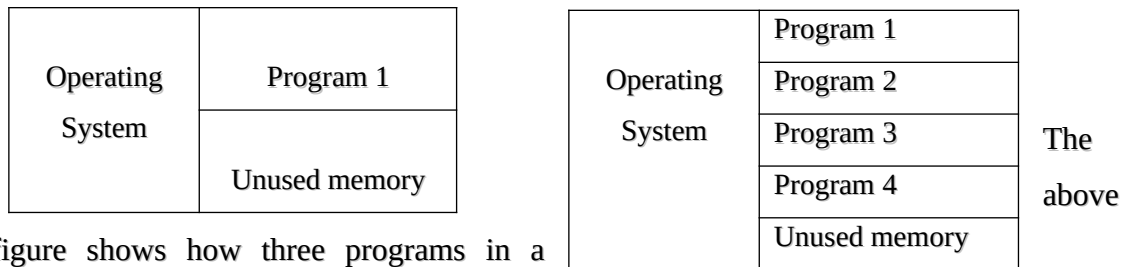


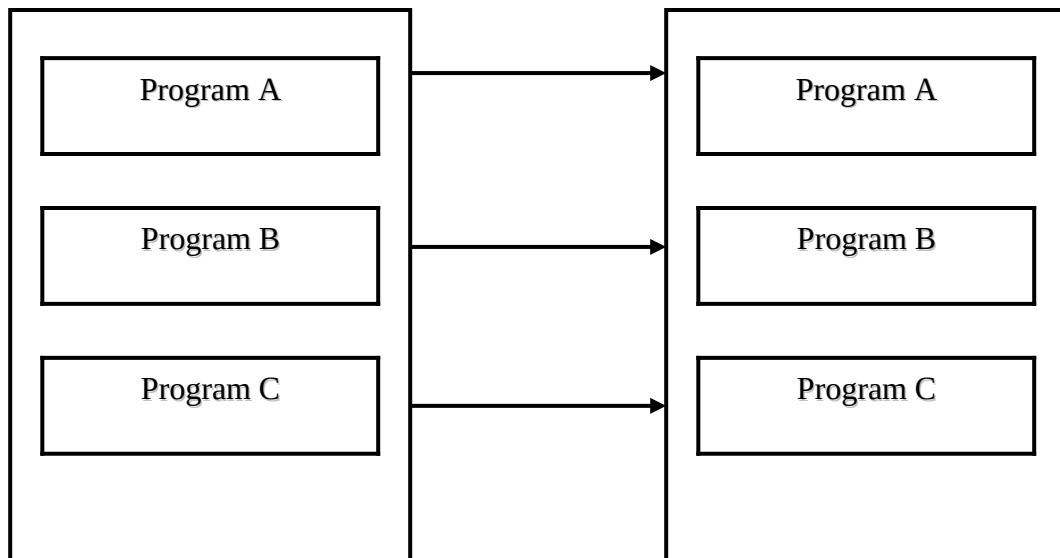
figure shows how three programs in a multiprogramming environment can be stored in primary storage. The first program executes until an input/output event is read in the program. The operating system then directs a channel (a small processor limited to input and output functions) to read the input and move the output to an output device. The CPU moves to the second program until an input/output statement occurs. At this point, the CPU switches to the execution of the third program, and so forth, until eventually all three programs have been executed.

- ii) **Multitasking:** refers to multiprogramming on single user operating systems such as those in microcomputers. One person can run two or more programs concurrently on a single computer. For example, a sales representative could write a letter to prospective clients with a word processing program while simultaneously using a database program to search for all sales contacts in a particular city or geographic area. Multitasking allows the sales representatives to display both programs on the computer screen and work with them at the same time.
- iii) **Virtual storage:** was developed after some problems of multiprogramming became apparent. Virtual storage handles programs more efficiently because the computer divides the programs into small fixed or variable length portions, storing only a small portion of the program in primary memory at one time.

First, although two or three large programs can be read into memory, a certain part of main memory generally remains underutilized because the programs add up to less than the total amount of primary storage space available.

Second, given the limited size of primary memory, only a small number of programs can reside in primary storage at any given time. Only a few statements of a program actually execute at any given moment. Virtual storage takes advantage of this feature of processing. Virtual storage breaks a program into a number of fixed length portions called pages or into variable length portions called segments. Each of these portions is relatively small (a page is approximately 2 to 4 kilobytes). This permits a very large number of programs to reside in primary memory, in as much as only one page of each program is actually located there. Consider the following figure:

Figure 4.3: Virtual storage



As you see from the above figure, all other programs pages are stored on a peripheral disk unit until they are ready for execution. Virtual storage provides a number of advantages. First, the central processor is utilized more fully. Many other programs can be in primary storage because only one page of each program actually resides there. Second, programmers no longer have to worry about the size of the primary storage area.

- iv) **Time sharing:** is an operating system capability that allows many users to share computer processing resources simultaneously. It differs from multiprogramming in that the CPU spends a fixed amount of time on one program before moving onto another. In a time sharing environment, thousands of users are each allocated a tiny slice of computer time (2 milliseconds).
- v) **Multiprocessing:** is an operating system capability that links together two or more CPUs to work in parallel in a single computer system. The operating system can assign multiple CPUs to execute different instructions from the same program or from different programs simultaneously, dividing the work between the CPUs. While multiprogramming uses concurrent processing with one CPU, multiprocessing uses simultaneous processing with multiple CPUs.

4.2.4 Language Translation And Utility Software

When computers execute programs written in languages such as COBOL, FORTRAN, or C, the computer must convert these "human readable" instructions into a form it can understand. Computers interpret binary ones and zeros, and the language translator found in system software makes the conversion. This type of system software is called a compiler or interpreter. The program in the high-level language before translation into machine language is called source code. A compiler translates source code into machine code called object code.

Some programming languages like BASIC do not use a compiler but an interpreter, which translates each source code statement one at a time into machine code and executes it. Interpreter languages like BASIC provide immediate feedback to the programmer if a mistake is made, but they are very slow to execute because they are translated one statement at a time.

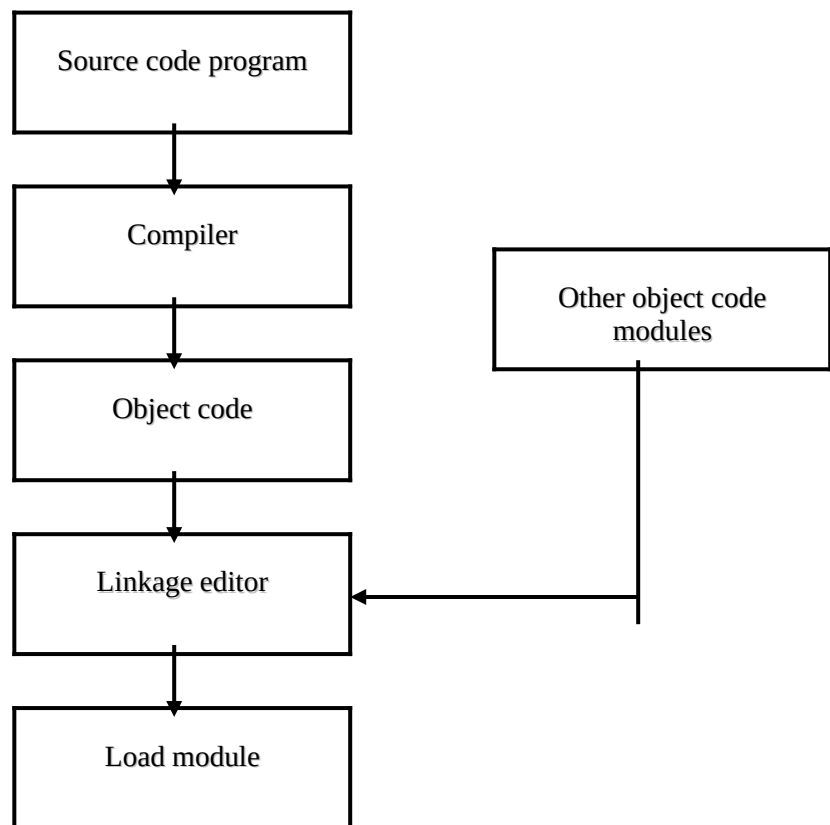
An assembler is similar to compiler but is used to translate only assembly language into machine code. System software includes utility programs for routine, repetitive tasks, such as copying.

The source code, the program in a high level language is translated by the computer into object code so that the instructions can be "understood" by the machine. These are grouped into modules. Prior to execution, the object code modules are joined together by the linkage editor to create the load module. It is the load module that is actually executed by the computer.

Clearing primary storage, computing a square root, or sorting. If you have worked on computer and have performed such functions as setting up new files, deleting old files, or formatting diskettes, you have worked with utility programs.

Utility programs are pre written programs that are stored so that they can be shared by all users of a computer system and can be rapidly used in much different information system application when requested.

Figure4.3: The language translation process



Graphical User Interfaces

Whenever users interact with a computer, even a microcomputer, the interaction is controlled by an operating system. The user interface is the part of an information system that users interact with. Users communicate with an operating system through the user interface of that operating system. Early microcomputer operating systems were command driven, but the graphical user interface, often called a GUI, has become the dominant model for the user interface or microcomputer operating systems.

Many graphical user interfaces use a system of pull down menus to help users select commands and pop-up boxes to help users select among various command options. Windowing features allow users to create, stack, size, and move around various boxes of information. Graphical user interfaces can promote superior screen and print output communicated through graphics.

4.3 APPLICATION SOFTWARE

Application software is primarily concerned with accomplishing the tasks of end users. Many different programming languages can be used to develop application software. Each has different strengths and drawbacks.

4.3.1 Generations of Programming Language

To communicate with the first generation of computers, programmers had to write programs in machine language – the Os and Is of binary code. End users who wanted applications had to work with specialized programmers who could understand, think and work directly in the machine language of a particular computer. Programming in Os and Is, reducing all statements such as add, subtract, and divide into a series of Os and Is, made early programming a slow, labor intensive process.

Machine language was the first generation programming language. The second generation of programming languages occurred in the early 1950s with the development of assembly language. Instead using Os and Is, programmers could now substitute language like acronyms and words such as add, sub (subtract), and load in programming statements. A language translator called a compiler converted the English like statements into machine language.

When the third hardware generation was underway, programming languages entered the third generation as well. From the mid 1950 – 1970, the first high level languages emerged. These languages permitted mathematicians for the first time to work with computers through the use of languages such as FORTRAN.

Mathematicians were now able to define variables with statements such as $Z = A + B$. The software translated these definitions and mathematical statements into a series of Os and Is.

COBOL permitted the use of English statements such as print and sort to be used by programmers, who did not have to think in terms of Os and Is.

These higher-level languages require much faster, more efficient compilers to translate high-level languages into machine codes.

Fourth generation computer languages emerged in the late 1970s, and their development is still in progress. These languages dramatically reduce programming time and make software tasks so easy that non-technical computer users can develop applications without the help of professional programmers. Fourth-generation tools also include pre-written application software packages that can be used directly by end users. Using the software package Lotus 1-2-3, for instance, users can create their own financial spreadsheets and manipulate data without programmer intervention.

4.3.2 Popular Programming Languages

Most managers need not be expert programmers, but they should understand how to evaluate software applications and to select programming languages that are appropriate for their organization's objectives. The more popular high level languages are:

- i) **Assembly language:** is a programming language developed in the 1950s that resembles machine language but substitutes mnemonics for numeric codes.

For example: AR 5, 3

This sample assembly language command adds the content of register 3 to register 5 and stores the result in register 5.

- ii) **FORTRAN (Formula Translator):** is a programming language developed in 1956 for scientific and mathematical applications.

For example: READ (5,100) ID, Quant, PRICE

Total = QUANT * PRICE

This sample FORTRAN program code is part of a program to compute sales figures of a particular program.

- iii) **COBOL (Common Business Oriented Language):** COBOL is a predominant programming language for business applications because it can process large data files with alphanumeric characters.

For example, MULTIPLY QUANT-SOLD BY UNIT-PRICE GIVING SALES-TOTAL

This sample COBOL program code is part of a routine to compute total sales figures for a particular item.

- iv) **BASIC (Beginners All-Purpose Symbolic Instruction Code):** BASIC is a general purpose programming language used with microcomputers and for teaching programming.
- v) **PL/I (Programming Language):** PL/I was developed by IBM in 1964. It is the most powerful general-purpose programming language because it can handle mathematical and business applications with ease, is highly efficient in input/output activities and can handle large volumes of data.
- vi) **Pascal:** is a programming language used on microcomputers and to teach sound programming practices in computer science courses. The language is weak at file handling and input/output and is not easy for beginners to use.
- vii) **C:** is a powerful programming language with tight control and efficiency of execution. It is portable across different microprocessors etc.

4.3.3 Fourth Generation Languages

Fourth generation languages consist of a variety of software tools that enable end users to develop software applications with minimal or no technical assistance or that enhance the productivity of professional programmers.

Fourth generation languages tend to be non-procedural or less procedural than conventional programming languages.

The followings the major categories of fourth generation languages: query languages, report generators, graphics languages, application generators, application software packages, and microcomputer tools.

Details are as follows:

- a) *Query Languages*: are high-level languages for retrieving data stored in a databases or files. They are usually interactive, online, and capable of supporting requests for information that are not predefined. Query languages can search a database or file using simple or complex selection criteria to display information relating to multiple records.
- b) *Report generation*: are facilities for creating customized reports. They extract data from files or databases and create reports in many formats. Report generators generally provide more control over the way data are formatted, organized, and displayed than query languages. The more powerful report generators can manipulate data with complex calculations and logic before they are output.
- c) *Graphics languages*: graphic language retrieves data from files or database and displays them in graphic format. Users can ask for data and specify how they are to be charted. Some graphics software can perform arithmetic or logical operations on data as well.
- d) *Application generators*: contain pre-programmed modules that can generate entire applications, greatly speeding development. A user can specify what needs to be done, and the application generator will create the appropriate code for input, validation, update, processing, and reporting. Most full function application generators consists of a comprehensive, integrated set of development tools: a database management system, data dictionary, query language, screen painter, graphics generator, reports generator, decision support/modeling tools, security facilities, and a high level programming language.
- e) *Application software packages*: a software package is a pre-written, pre-coded, commercially available set of programs that eliminates the need for individuals or organizations to write their own software programs for certain functions. There are software packages for system software, but the vast majority of package software is application software.

Application software packages consist of pre-written application software that is marketed commercially. These packages are available for major business applications on main frames, minicomputers, and microcomputers. They contain customization features so that they can be tailored somewhat to an organization unique requirement.

- f) *Microcomputer tools*: some of the most popular and productivity promoting fourth generation tools are the general purpose application packages that have been developed

for microcomputers, especially word processing spreadsheet, data management, graphics, and integrated software packages.

- *Word processing software:* stores text data electronically as a computer file rather than on paper. The word processing software allows the user to make changes in the document electronically in memory.

This eliminates the need or retypes an entire page in order to incorporate corrections. The software has formatting options to make changes in line spacing, margins, character size, and column width. Microsoft word and Word perfect are popular word processing packages.

- *Spreadsheets:* electronic spread sheet software provides computerized version of traditional financial modeling tools such as the accountant's columnar pad, pencil, and calculator. An electronic spread sheet is evident when one changes a value or values, because all other related values on the spreadsheet will be automatically recomputed.

Many spreadsheet packages include graphics functions that can present data in the form of line graphics, bar graphs, or pie charts. The most popular spreadsheet packages are Lotus 1-2-3, Quattro, and Microsoft Excel.

- *Data management software:* while spreadsheet programs are powerful tools for manipulating quantitative data, data management system is more suitable for creating and manipulating lists and combining information from different files. Microcomputer database management packages have programming features and easy to learn menus that enable non-specialists to build small information systems.

Data management software typically has facilities for creating files and databases and for storing, modifying, and manipulating data for reports and queries. Popular database management software for the personal computer includes Microsoft Access, Paradox, etc.

- *Integrated software packages:* combine the functions of the most important microcomputer software packages, such as word, processing, spreadsheets, graphics, and data management.

4.3.4 New Software Tools and Approaches

A growing backlog of software projects and the need for businesses to fashion systems that are flexible and quick to build have spawned a new approach to software development with "object-oriented" programming tools.

- i) **Object-oriented programming:** combines data and the specific procedures that operate on those data into one "object". The object combines data and program code. Instead of passing data to procedures, program send a message for an object to perform a procedure that is already embedded into it (procedures are termed "methods" object oriented languages). The same message may be sent to many different objects, but each will implement that message differently.

For example, an object-oriented financial application might have customer objects sending debit and credit messages to Account objects. The account objects in turn might maintain cash-on-hand, accounts payable, and accounts receivable objects.

An object's data are hidden from other parts of the program and can only be manipulated from inside the object. The method for manipulating the object's data can be changed internally without affecting other parts of the program. Programmers can focus on what they want an object to do, and the object decides how to do it.

Object oriented programming is based on the concept of class and inheritance. Class is a feature of object oriented programming so that all objects belonging to a certain class have all of the features of that class.

Inheritance on the other hand is a feature of object oriented programming in which a specific class of objects receives the features of a more general class.

Program code is not written separately for every object but for classes, or general categories of similar objects. Objects belonging to a certain class have the features of that class. Classes of objects in turn can inherit all the structures and behaviors of a more general class and then add variables and behaviors unique to each object. New classes of objects are created by choosing an existing class instead of starting from scratch each time.

- ii) **Visual programming:** Object oriented programming has spawned a new programming technology known as visual programming. With visual programming, programmers do not write code. Rather they use a mouse to select and move around programming objects, copying an object from a library into a specific location in a program, or drawing a line to connect two or more objects. The window on technology more fully describes drop and drag, one visual programming method.

Check Your Progress Exercise

1. What is an operating system? Discuss.

2. What are the major capabilities of an operating system?

3. Compare the major microcomputer operating system.

4. Explain the major types of software.

5. What is graphical user interface? Explain.

4.4 SUMMARY

Software is a detailed instruction that control the operation of a computer system. A software program is a series of statements or instructions to the computer. Basically, there are two types of software: system software and application software.

System software manages the computer resources and mediates between application software and computer hardware.

Application software is used by application programmers and some end users to develop systems and specific business applications. Applications software works through system software, which controls access to computer hardware.

System software coordinates the various parts of the computer system and mediates between the application software and computer hardware. The system software that manages and controls the activities of the computer is called the operating system. Other system software includes computer language translation program that converts programming languages into machine language and utility programs that perform common processing task.

The operating system acts as the chief manager of the information system, allocating, assigning, and scheduling system resources and monitoring the use of the computer.

Multiprogramming, Multitasking, Virtual storage, Time sharing, and Multiprocessing, enable systems resources to be used more efficiently so that the computer can attack many problems at the same time.

Microcomputer operating systems are starting to develop sophisticated capabilities such as multitasking and support for multiple users on networks. Leading microcomputer operating systems include Windows 95, 98, 2000, XP, Windows NT, OS/2, Unix, System 7, and DOS.

There have been four generations of software development: (1) machine languages, (2) assembly language, (3) high level language, and (4) four generation language.

Fourth generation languages include query language, report generators, graphics languages, application generators, application software packages, and microcomputer tools.

The new concept of object oriented programming combines data and procedure into one “object”, which can act as an independent software building block. Each object can be used in many different systems without changing program code.

4.5 ANSWER TO CHECK YOUR PROGRESS EXERCISE

1. Refer Section 4.2.1
2. Refer Section 4.2.3
3. Refer Section 4.2.2
4. Refer Section 4.2 & 4.3
5. Refer Section 4.2.4