



SOFTWARE: AN INTRODUCTION

2.1 Software: An Introduction

Software is the program that instructs a computer how to process data and generate required output.

2.2 Types of Software

Software can be divided into two categories:

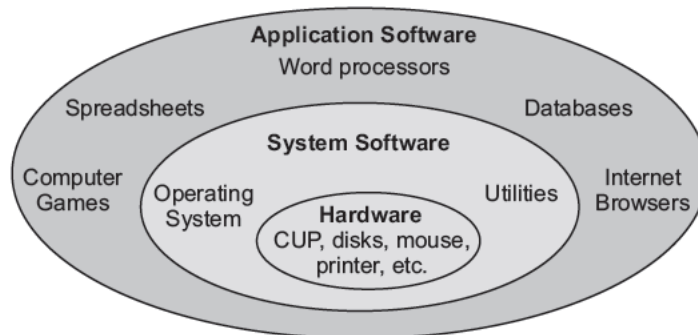
System Software: A computer user does not understand the machine language (a language that a computer understands), and similarly a computer cannot understand a high-level language in which a user communicates. So, if a user wants to work on a computer system and wants to run an application program, the computer needs to have system software. *System software* can be defined as a collection of programs that performs the following functions:

- Receiving and interpreting user commands (i.e., converting users' instructions into machine language and vice versa).
- Running application programs and storing them in the hard disk or any other secondary storage as per the directions given by the user.
- Retrieval of the stored programs from the hard disk or any other secondary storage device on the user's command.
- Creates an interface among the peripheral devices and the CPU, directs and produces the results on the user's command.

Thus, it can be said that system software is responsible for the coordination of all activities in a computer system.

Application Software: *Application software* is written with a specific purpose in mind. Application software allows users to work in English or give the commands in a format that is not dependent on computer hardware. To run application software there has to be system software. Application software includes high-level language programs like basic, C, C++, Pascal, etc., or packages like Microsoft Office, Lotus Smart Suite, etc. It is not necessary for the high-level programmer to know the machine-level programming. Application software can be further classified into two categories.

- The software development firms like Microsoft Corporation, Oracle etc., prepare general-purpose software. Software included in this category are Tally, Ex from TCS, Microsoft Office from Microsoft Corporation, and many more.
- Tailor-made application software is prepared by software development firms according to the needs and wants of their clients. For example, software development prepared by Infosys, Wipro, TCS, and many more software development firms for their clients.



2.3 Assembler/Interpreter/Compiler

Programming languages can be classified in the three categories. They are as follows:

- *machine language*
- *assembly language* (LLL)
- *high-level languages* (HLL)

Machine Language

These are the languages that are machines dependent. Machine language is directly understood by the computer system, as it consists of a series of binary digits (0s and 1s); therefore no conversion is required and the processing of the program is very fast. Because the instructions given in a machine language are in binary digits only, it is very difficult for the common user to write a program or work on the computer system; it is also a very time-consuming process. Not only do the programmers need to remember the codes representing the machine instruction set, but they also have to keep track of the storage location of the data and the instructions. It is very difficult to detect and rectify errors in a machine language program. For this reason, these languages are not popular.

Assembly Language (Low-Level Language)

Because of difficulties previously explained, a machine language is not widely used, and thus assembly language comes into existence. These languages were developed in the early 1950s and used mnemonic operation codes. For addresses, symbolic representations were used. The first step in programming involved replacing the numeric binary machine-language operations codes with the mnemonic names. Machine codes were still in use for processing data; therefore special software was required, which translates the programs written in assembly language into a machine language. The major limitation of the assembly language was that they were machine dependent, that is, programs written on one machine will not work on another machine.

Assembler: *Assembler* is a special software that converts the program in assembly language into machine language. The program written in assembly language is called *source code* and is converted into a machine-readable format by an assembler known as *object code*. Assembly languages have many advantages over machine languages, as they save time and reduce complexity. The number of errors committed is also reduced and the identification and removal of bugs becomes quite easy.

High-Level Languages (HLL)

Working with assembly language was easier in comparison to a machine language, but still the common user had several difficulties in writing programs; thus the high-level languages (HLL) were developed. With the advent of HLL, working on a computer system became an easy job for the common user. These languages are very close to the English language,

and they move away from machine dependency. Unlike the low-level languages, all instructions are written in English, which follows some rules called syntax of the language. But computers understand only machine language, therefore these high-level languages need to be translated into the machine code, and the translator used for this purpose is an interpreter or a *compiler*. In low-level languages, only one instruction was translated into a machine instruction at a time; however high-level language statements are translated into several machine code instructions by the translators.

Interpreter: An *interpreter* can be defined as software that is used for converting a program written in a high-level language into machine language. The interpreter takes the high-level language program code as input, line-by-line, and converts it into a machine language, line-by-line. No object code is produced in this process. If the interpreter finds any error in any line of the program, it stops there and notifies the user about the error. The user needs to correct the error and then rerun the interpreter. The translation of the program from the high-level language into machine code restarts from the beginning of the program, thus making the conversion of the program a time-consuming process.

To overcome this problem, another translator program called *Compiler* was designed.

Compiler: It can be defined as a translator program that converts the high-level language programs (source codes) into machine languages, and the object code is generated, stored, and can be executed at any time. *Object code* refers to a code that can be run to produce the output of the program written in a high-level language even without the availability of the actual program (source code). In the process of compilation, the compiler converts the complete program into an object code (machine readable) and notifies the user about all the errors in the program. Now the user needs to go to the source code (the actual program written in HLL) and rectify all the errors reported by the compiler, and then he needs to recompile the program. The recompilation process is a must, otherwise new corrected object code will not be generated and the user will have the earlier incorrect object code in the system, which will still produce errors. Compiler programs are very effective in comparison to an interpreter, as they take the whole source program and convert it into reusable object code which results time savings. Now, almost every high-level language comes with its own compiler, which makes them very efficient.

2.4 Generations of Computer Languages

Evolution or generations of programming languages are described as follows:

- **First-Generation Languages:** The first generation of languages started around 1940. The programs were written in machine languages that were very difficult to code. Later in the 1950s, programming in assembly languages started, and the assemblers were designed, which converted the assembly language code into the machine-level code.
- **Second-Generation Languages:** These languages came into existence between the late 1950s and early 1960s. These languages led the way for the introduction of many new concepts in programming. ALGOL60, COBOL, LISP, FORTRAN, etc., were some of the popular languages of the second generation. The concepts of data type and structured programming were also developed during this generation.
- **Third-Generation Languages:** Third-generation languages saw their evolution in the late 1960s and early 1970s. Some of the popular languages of this generation are SIMULA67, BASIC, SNOBOL 4, C, and PL/1. The concept of record structure and classes, arrays, pointers, storage classes, etc., were also introduced during this period.
- **Fourth-Generation Languages:** Most of these languages had their base in third-generation languages and evolved in late 1970s. Many software development tools were introduced which enhanced the productivity of fourth-generation languages. These languages interact with database management systems (DBMS) tools for storing, manipulating and retrieving data.

High-level languages are usually considered to be the procedural languages; on the other hand most of the fourth-generation languages are nonprocedural languages. *Procedural language* means a programming language that requires writing a series of systematic statements, functions, and fixing syntax to complete a program, for example, COBOL. Most of the statements in fourth-generation languages are simple and self-explanatory, which made them popular and widely used. However, fourth-generation languages also provide a user facility of just specifying the output without writing an actual program, wherein the program is automatically generated by the software. For example, the recording of a “MACRO” where the user simply records the steps and the program is generated by the software itself.

2.5 Uses of Computer Languages

Various applications of programming languages are as follows:

- **Scientific-Mathematical Applications:** These are programming applications that use the concept of science and a high level of mathematical computation. These programs are written to fulfil the mathematical, scientific, and statistical programming needs of the user for solving problems with the help of a computer. These programming applications are usually complex in nature, and to use these programs, users need to be well versed in mathematical principles, algorithms, and statistical principles so the problems can be solved properly by writing a program. A couple of examples of this are SPSS and R-Software for statistical analysis.
- **Text Processing Applications:** When a user needs to draft and format a letter, text processing applications are required. These applications involve manipulation of any natural language text as data. Word processors like MS WORD are included in this category. These text processing applications help to enhance the productivity of a business organization.
- **Data Processing Applications:** Today, data is considered an important asset for an organization. Organizations need to record, store, maintain, summarize, and retrieve data on a day-to-day basis. To fulfil this need, many data processing applications have been developed. The level of data involved in daily processing is very high and its management is very important. Therefore the data processing applications that have been developed can support huge levels of data transactions.
- **System Programming Applications:** System software is required to act as an interface between the user and the computer. This is because a computer system understands only the machine language that is not understood by the user. System software helps the user to work in any high-level language and converts it into the machine code that is understood by a computer. The operating system, compilers, interpreters, schedulers, etc. are included in this category of applications. “C” language is used to write system-programming applications.

- **Artificial Intelligence Applications:** Artificial intelligence means a situation where a computer can think like a human. The applications of this concept are varied and include logical games like chess, bridge etc. Most of these applications use LISP and PROLOG languages, which are logic based.

Advantages of High-Level Languages:

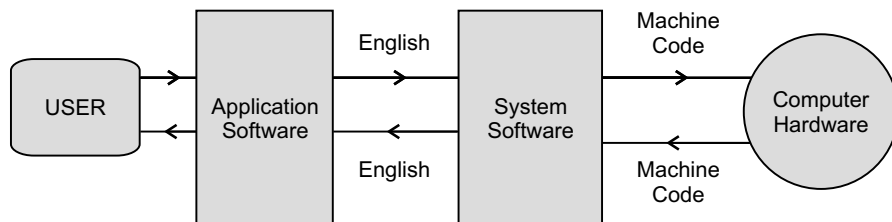
- easy to learn
- easy to understand
- easy to program
- easy to maintain
- easy to document
- easy to debug
- less time-consuming
- portable

Here, *debugging* means the removal of errors in the program, and *portability* means when a program is written in a high-level language, it can run on any computer irrespective of the hardware configuration.

2.6 Operating Systems

2.6.1 Definition

An *operating system* is system software. It is a set of programs that provides an interface between the user and the computer system (hardware). In other words, it coordinates the flow of information from the computer to the user and vice versa.



2.6.2 Functions of an Operating System

The operating system performs a number of services and/or the functions for the users of the computer. These operating system (OS) services or functions that might be performed are as follows:

- **Memory Management:** Here memory means random access memory (RAM). When the programs are loaded into the memory and are executed, the OS reads them from the hard disk and loads them into the memory (RAM). But before it loads, the OS checks whether the memory is available or not, and if available, it allocates it to the program. Once the program execution is over, the OS removes the program from the memory (RAM) and this freed memory can be used for another program.
- **CPU Management:** A microprocessor executes a number of processes at one time. Deciding which process is to be executed first, and which next, is a complex problem. This problem is effectively handled by the OS and is known as *job scheduling*. The OS helps the processor to schedule its activities so as to fulfil the requirements of various parts of the computer system. The processor gets the job from the memory, processes the job, and passes the result onto the predetermined place and makes itself ready for the next job.
- **Disk Management:** Data and program storage on the hard disk or any other storage device is a complex phenomenon. It is very difficult for a common user to understand this. Thus, the OS spares the user from understanding this process as it coordinates the storing and retrieval of files.
- **Input/Output Management:** Numbers of peripheral devices such as keyboard, mouse, printer, projectors, etc., are attached to a computer system. To make these devices work properly, the system requires supporting software called *device drivers* in the memory. The device drivers are also known as the *support utility programs* of the OS and are controlled by it. All these devices communicate with the computer for obtaining the job and getting the job done, which is again monitored by the OS.
- **User Interface:** The system, but not the user, understands machine language, which consists of 0s and 1s. The OS bridges the gap between the machine understandable language and human understandable language by providing a command interpreter.

- **Communication:** A computer system runs a number of processes at a given point of time and there exist situations where one process needs to exchange information with another process. The OS facilitates these communication processes within a system and among the systems. Communication is done either via shared memory or by the technique of *message passing* in which packets of information are moved between processes by the OS.
- **Error Detection:** While working on a system, it is quite possible that a user may commit a number of errors. The OS constantly keeps track of all possible errors. Errors may be typing a wrong file name, wrong syntax of a command, network failure, a printer jam, or running out of paper. These are errors that are frequently committed by a user. For each type of error, the OS takes an appropriate action by showing an appropriate message on the computer screen prompting the user to correct their action.
- **Resource Allocation:** The OS does the proper allocation of computer resources as there might be multiple users or multiple jobs running at the same time. Resources like processing time, main memory (RAM), file storage, devices, etc. must be allocated to all of the users equally, and this is handled by the OS.
- **Accounting:** The OS keeps full track of how much and what kind of computer resources are used by a user. This record keeping can be used for improving computing services.
- **Protection:** The OS provides a multilevel protection mechanism to users. This protection is required by users from other users or an outsider. The OS provides a mechanism by which a user can protect his data, and this is done by controlling the access to system resources. For this, the OS uses the authentication process in which each user has to authenticate himself to the system. This is done by means of a multilevel password protection system provided by an OS.

2.6.3 Types and Classifications of Operating Systems

System software can be broadly classified into the following categories on the basis of their usage:

- **Batch Processing System Software:** A negligible interaction between the user and the program *batch processing system software*. In this type of system jobs are processed in the order in which they are entered, that is, on a “first in, first out” basis (FIFO). In a batch processing

system, memory is divided into two parts; one is permanently occupied by the software, whereas the other is used as per the need of the user. It simplifies the processing operations because the instructions are executed in batches, and thus saves the processor time.

- **Multi-User Operating System Software:** The *multi-user operating system* supports the multiple units of PCs called “terminals” that are attached to the main computer system as in mini and mainframe computer systems. It consists of only one central processing unit (a microprocessor) that performs all the operations. These systems are used when two or more users try to run programs at the same time. Examples of the multi-user operating system are UNIX, MSV, etc.
- **Multiprogramming or Multi-Tasking System Software:** This is the system software that is capable of running more than one program at the same time. *Multiprogramming* can be defined as a process of creating a situation in which more than one program may be held in the main memory at one time, thus making it possible to process several programs at a time. The main objective of developing this kind of system software is to minimize unused microprocessor time. A computer switches from one job to another at a rapid rate under the time-sharing mode. Different terminals are used to enter jobs into the computer. After processing the first user’s job, it proceeds to the second and then to the third and so on for a short period of time called the “time slices,” before returning to the first user’s job from where it earlier started. This cycle continues indefinitely. When one program is finished the other program replaces it. UNIX, OS/2, and Windows are commonly used multiprogramming or *multitasking operating systems*. The processor is kept busy while channels and buffers are occupied with a job of bringing data and writing out information.

For example, let us assume that three users are working on a system simultaneously. In this concept the program of each user will be divided into a number of pages (layers) of equal size. During execution, the processor will divide its processing time equally among all of the users. It will first process the layer 1 (L1) of the *program 1* (P1), then *layer 1* (L1) of the program 2 (P2), and finally the layer 1 (L1) of the program 3 (P3). The processor will give equal time to all the users, but it will appear to all of the users as if processor is giving its full time to them. When program 1 (P1) is finished, the processor will divide its time equally among the remaining programs (P2 and P3) and finally to the program 3 (L3).

P1, L1
P2, L1
P3, L1
P2, L2
P3, L2
P3, L3

- **Multiprocessing Operating System Software:** In a *multiprogramming operating system*, more than one program is processed by an operating system, whereas in a *multiprocessing operating system*, one program is processed by more than one processor. A *multiprocessing operating system* software uses multiple processors that share a common memory. Instructions from different and independent programs are processed at the same time by different processors. On the contrary, the processors may simultaneously execute different instructions from the same program. Examples of commonly used multiprocessing operating systems are OS/2, UNIX, MSV, etc. Multiprocessing systems can be classified as:

- loosely coupled multiprocessing
- functionally specialized processors
- tightly coupled multiprocessing
- parallel processing

Multiprocessor systems usually consist of two or more processors. Each processor has its own CU, ALU, etc. An interconnection mechanism allows each processor to access shared main memory and input/output devices. The processors not only communicate with each other through memory

but also are able to directly exchange signals. Memory is organized in such a manner such that it provides a multiple simultaneous access to a separate block of memory. The operating system controls this entire system and provides interaction between processors and their programs.

- **Real-Time System Software:** Real-time systems are the systems in which response time is critical. These are the systems that are involved with the immediate processing of data, machines, and records. These systems are designed to accept the data in real time, which means as soon as an activity occurs, the system processes the data immediately and generates the output in time to have an effect on the ongoing activity. Real-time systems are online systems with tighter constraints on response time. Examples of the real-time operating systems are C Executive, communications control program (CCP), CTOS, CTRON, FADOS, etc.

2.6.4 Components of an Operating System

An operating system consists of two primary components:

- a supervisor
- an integrated set of support utilities

1. The Supervisor (or Control Program): The *supervisor* is defined as the component of an operating system that takes care of the overall working of the computer system. It is a set of programs that are integrated to one another. It performs the following three basic functions:

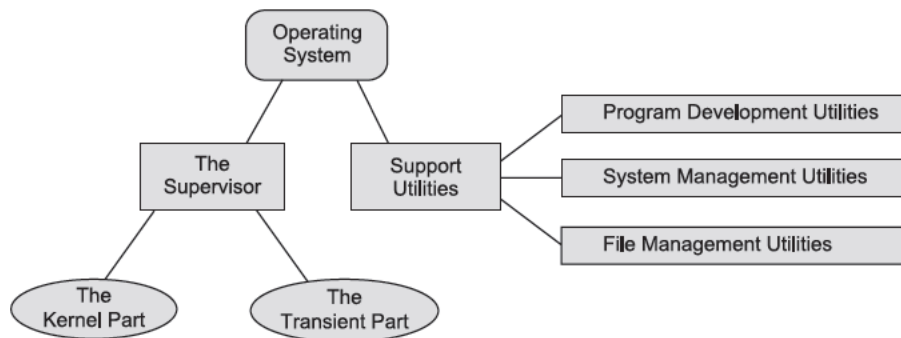
- It initializes the system at the time of start-up
- It allows running of the application programs
- It controls input and output devices attached to the system.

The supervisor also performs some additional functions such as keeping a track of computer time for different users, etc. The OS is generally found on a hard disk and sometimes in the form of a chip called *firmware*. The supervisor consists of two portions: a *kernel* and a *transient* portion. When the supervisor is loaded into the memory for the first time, both portions are loaded. The kernel part of supervisor always remains in the memory with an application program. This is to monitor the system operations. It's the part of an operating system that directly interacts with the hardware, and therefore it must be present in the memory as long as the computer is

being used. The transient portion need not always be present in the memory. When executed, a program may overwrite this part of the OS. Once the program execution is over, the transient portion is reloaded into the memory.

2. The Support Utilities: The *support utilities* are the system programs that perform useful functions. They are classified as follows:

- **Program Development Utilities:** These include assemblers, editors, interpreters, compilers, and linkers.
- **System Management Utilities:** These include the programs that keep track of more than one user, diagnostic routines, etc.
- **File Management Utilities:** These include programs for copying files, erasing files, printing files, renaming files, etc.



2.7 What is Graphical User Interface (GUI)?

Microsoft Corporation brought a revolution in the area of system software development when they marketed software called Windows. Prior to the launch of Windows it was not possible for the user to work on a computer with the help of graphics. What the disk operating system (DOS) gave users was termed command line interface (CLI), where users were required to type the right command with the help of the keyboard on the command prompt (C:>_). Remembering the exact syntax and spelling of the command was a tedious task; however with Windows the concept of a *graphical user interface* (GUI) was introduced. With a GUI, the user did not get the

black screen of DOS with the pointer movement limitations; rather, users started to get a graphical screen where they were not required to remember the typical DOS commands for the purpose of doing their work. Here they were supposed to simply click on a picture (icon) option with the help of a mouse (a pointing device) and the Windows software automatically executed the command selected by the user.

The introduction of Windows brought a revolution in the field of computers and more and more users started to work on computer systems because it became easier to operate a computer system with the help of the Windows software. Before Windows 95, the earlier versions launched for Windows were Windows 3.1 and Windows 3.11 (network), but these required DOS to run. Windows 95 is in itself an operating system with a facility of GUI. Various versions of Windows OS launched so far are Windows 98, Windows 2000, Windows ME, Windows XP, Windows 7.0, Windows 8.0, Windows 8.1, and latest in the series is Windows 10.

2.7.1 Elements of GUI-Based Operating Systems

The Windows OS has various advantages over the DOS operating system and these advantages are:

- It is very easy to learn.
- It allows the user to work on multiple applications simultaneously. For example, one can type a letter in the foreground and in the background, printing can take place.
- It supports networking with other computer systems.
- It contains many advanced built-in features that were not available in DOS.
- Windows is user-friendly with a facility of GUI, that is, commands are represented by pictures and there is no need to remember the right syntax of any command.
- Windows offers common menus, that is, the same command in many related softwares does the same job (for example commands to save and print).
- Windows provides a facility to transfer data between different applications, whereas this is not possible in DOS.

Directory Management in Windows

■ File

A *computer file* is an entity that contains text, record, or a specific piece of data. A computer file may be anything from an executable program to a user-created document. A computer file consists of a file name that has two parts: one is a primary file name that creates the file's identity, the other is an extension or a secondary file name that tells the operating system and associated programs what type of file it is. In DOS, the name of a file was limited to eight characters, but modern Windows systems allow for much longer file names. The primary file name is generally determined by the user and defines the nature of files. A secondary file name or the file's extension is typically a period followed by two to four characters. This part of the file name is used by the operating system for internal cataloging. The way an extension is used varies between operating systems; some require the extension and some completely ignore it. The extension helps identify the programs used with the file and acts as a shortcut when read. For example, “:.ppt” means it is a PowerPoint file. Some of the important characteristics of a file are:

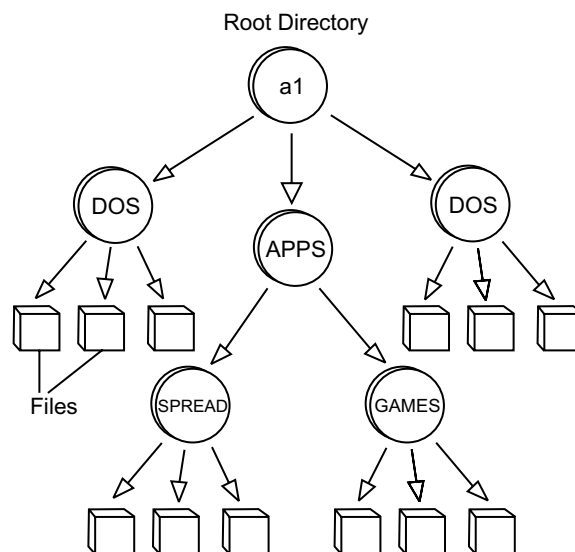
- All files stored in a system have a size, although it might be zero.
- A file is stored on a location in a computer system. This is maintained by creating an index by the system for all files stored.
- Certain attributes are associated with every file like read, write, archive hidden, etc. These attributes can be set by right-clicking on the file and clicking on the “properties” option.
- It is very easy to create, delete, or modify a computer file.

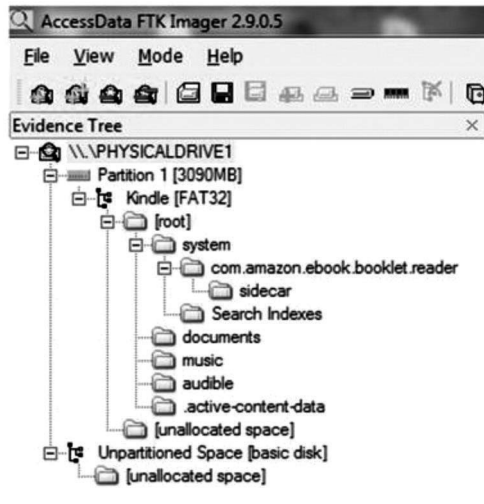
Some of the Computer File Types by Extension

Extension	Associated Program	Extension	Associated Program
AVI	Animated Video	MID	MIDI
BAT	DOS Batch File	MP3	Audio
BMP	Bitmap Graphic	PDF	Adobe Acrobat Document
COM	Program	PIC	Bitmap Graphic

DAT	Data file	ZIP	Compressed Archive of Files
DLL	Dynamic Link Library	PM4	PageMaker 4
DOC	Microsoft Word Document	PM5	PageMaker 5
EXE	Program	PPS	Microsoft PowerPoint
HTM	HTML Hyper Text Markup Language	PPT	Microsoft PowerPoint
ICO	Icon	RTF	Rich Text Format
JPG	Image	SYS	System Files
TMP	Temporary File	WAV	Audio File
TXT	ASCII Text	XLS	Microsoft Excel Spreadsheet

• Directory



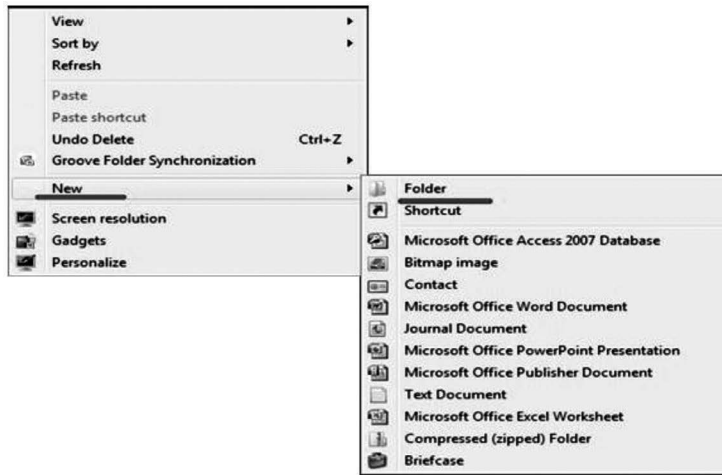


A directory commonly called *folder* can be defined as a place or a container that can be used to store files or sub folders. A user can consider a *directory* as a file cabinet that contains folders in which files can be kept. A directory is often defined in terms of an *inverted tree*. In Windows it is very easy to access a file in a directory. The topmost directory is called the *root directory*. A directory that is under another directory is called a *subdirectory*. A subdirectory may contain files or another subdirectory. A directory above a subdirectory is called the *parent directory*. In Windows the term *directory* is replaced by folder and subdirectory is by subfolder. Under DOS and Windows, the root directory is a back slash (\).

■ Creating Directory/Folder

A directory (folder) is used to store files. A user can create a number of folders and subfolders inside a folder as per the need. Steps to create folder are as follows:

1. Right click on the blank area of desktop and right click the mouse.
2. A drop-down menu will appear. Click on the “New” option and under New on the “Folder” option as shown in the following figure. This process will create a folder on the desktop labeled “New Folder,” however the name can be changed by the user.



3. A new name can be typed for the new folder, for example in our case it is “Word Files” and then press the Enter key.



■ Creating Subdirectory/Subfolder

A subdirectory/subfolder is a directory/folder that is created in an already existing directory/folder. Users need to follow the same steps as taken earlier to create a folder with only one difference. While creating a folder, the user has to click on either the desktop or on the desired location instead

of creating a subfolder. The user first needs to move inside the folder by double clicking on it and then repeating the steps previously discussed to create a subfolder.

■ Renaming a Directory or Folder/File

The user needs to take the following steps to rename a directory/folder:

1. Right click on the folder; a drop-down menu will appear. Click on the rename option as shown in the following figure.
2. Change the name of the folder as per your choice.



3. The folder will appear with a changed name as shown in the following figure:

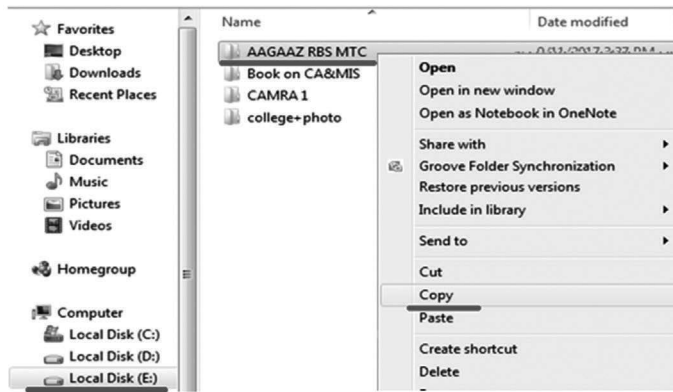


■ Copying the Directory or Folder/File

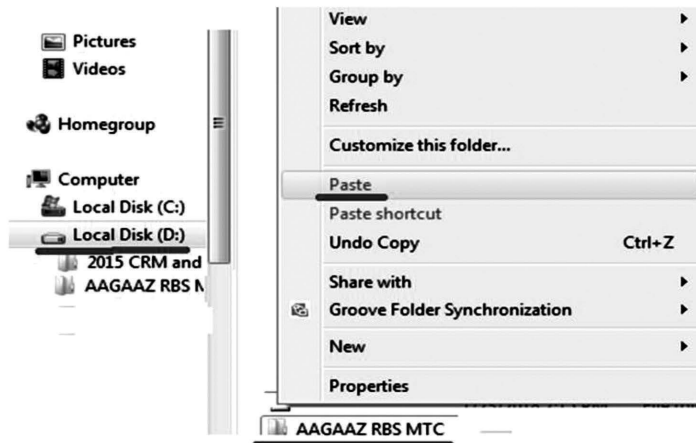
Copying a directory/folder means creating a duplicate of the original. The following are the steps for copying a directory/folder:

1. Open the location that contains the folder you want to copy. In our example, the folder exists in the disk partition E with a name "AAGAAZ RBSMTC."

2. Right-click on the folder and click on the “Copy” option from the drop-down menu.



3. Open the location where you want to store the copy. In our example, it is disk partition D. Now right click anywhere and then click on “Paste” option as shown in the following figure:



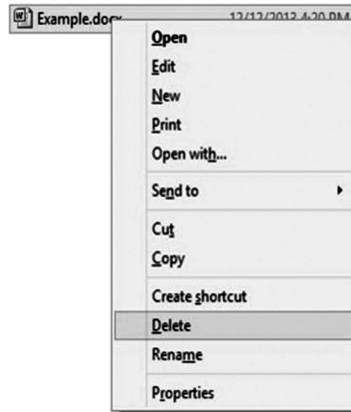
The copy of the original file or folder appears at the new location.

■ Deleting the Directory or Folder/File

- **Delete key:** The easiest method of deleting files in Microsoft Windows is by left clicking on the file, folder, or shortcut the user

wants to delete. The folder, file, or shortcut will now be highlighted. Pressing the delete key on the keyboard will delete the particular file, folder, or shortcut.

- **Delete file by right clicking:** Right click on the file, folder, or shortcut you want to delete, a drop-down menu will appear as shown in the below figure. Click on the delete option to complete the action.



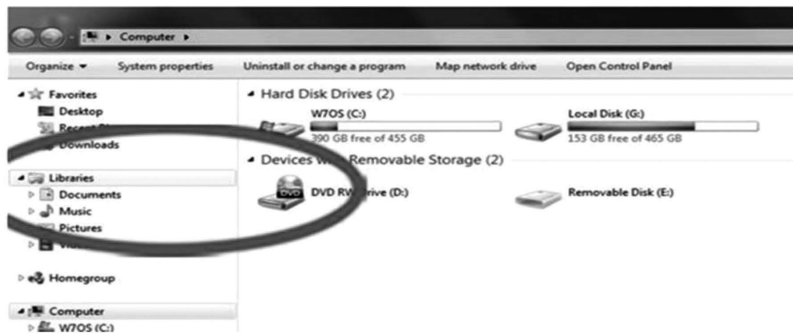
■ Creating a File

A file may consist of text, data, or records and is to be stored by assigning it a file name. A file name can be created by following the steps shown here:

1. Click the “File” menu in the program you are using (for example Microsoft Word, Excel, or PowerPoint), and then click “New.” This opens a new document.
2. When done working in the file, click on the File menu again, and then click Save As to name the file and save it to the desired location on the computer.

Another way to create a file on your computer is:

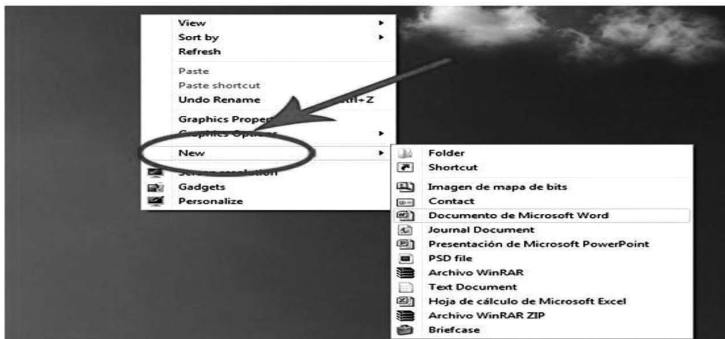
1. Navigate to the folder or desktop, you would like to create your file, e.g., “My Documents.”



2. Right click on empty section of the desktop:



3. Select "New" from the menu:



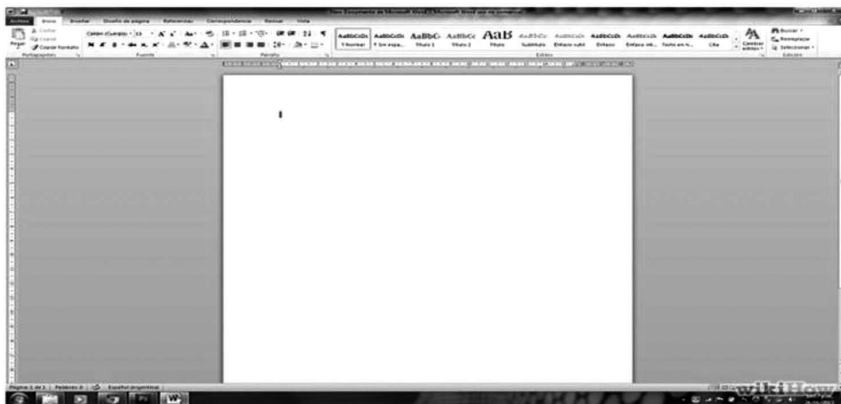
4. Select the type of file you would like to create:



5. Entre the name of the newly created file:



6. Open the new file to start work:



■ Use of Menus, Tools and Commands of Windows

Windows Explorer

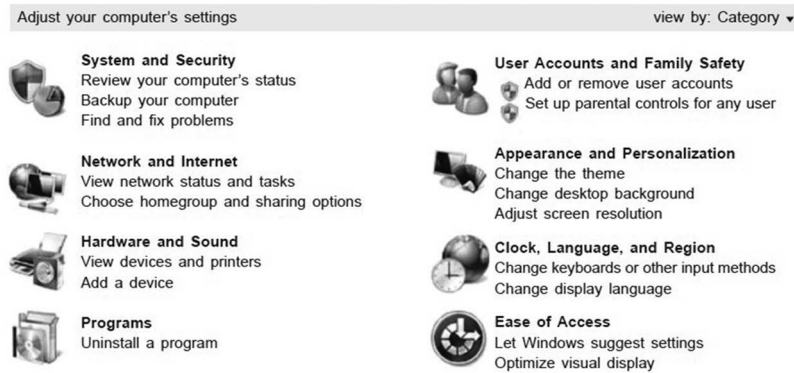
Windows Explorer is a utility provided by the Windows operating system that performs various functions for the management of the disk and file system. Some of the activities performed by Windows Explorer are as follows:

To cancel the last action within a program or in My Computer or Windows Explorer	To change the appearance of items in a folder	To rename a file or folder
To change which program starts when you open a file	To control access to a folder or printer	To copy a file or folder
To create a file type	To create a folder	To delete a file or folder
To determine how much space is available on a disk	To make a copy of a disk	To modify a file type
To move a file or folder	To name a disk	To open a document from within a program
To preview a file	To put a shortcut on the desktop	To quickly send files and folders to another place
To remove a program from the Start or Programs menu	To select multiple files and folders	To show all files and file name extensions

Apart from this, there are many more options that can be performed with the help of Windows Explorer.

Control Panel: *The control panel* is that utility provided by the Windows operating system, which helps in setting the working environment

for the user in which he or she wants to work. Various options provided by the control panel are shown in the following figure:



Options provided by the control panel allow the user to add new hardware, add or remove installed programs, set date and time, change the VDU display, set up fonts, set up game controllers, set up the keyboard, set up modems, set up a mouse, manage network/Internet settings, create a user's account, set up passwords, install printers, and many more options.

Print Manager: Print manager is the utility of Windows that helps the user in manage printer settings. To do this one has to follow these steps:

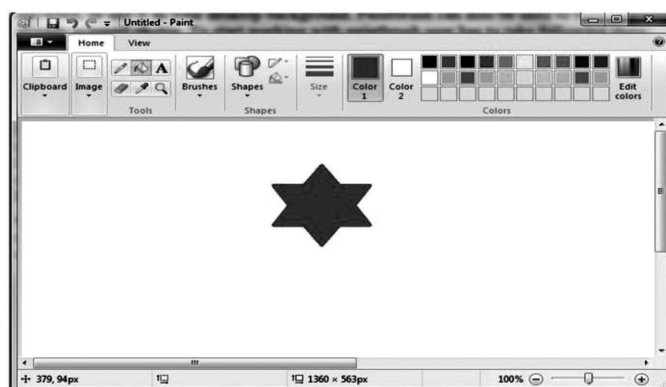
- Click on the “Start” option from the Windows opening screen.
- Click on the “Devices and Printers” option from the popup menu.
- Select the “Add a printer” option.



Paint: The “Paint” option provided by Windows is used for the purpose of creating, editing, and viewing pictures. The user can paste a Paint picture into another document such as Microsoft Word, or use it as a desktop background. Paint can also be used to view and edit scanned photos. To start working with Paint, the user has to take the following steps:

- Click on Start, point first to Programs, then to Accessories, and then click on the *Paint* option.

The Paint program provides various tools for making a painting on the drawing screen provided. Various tools provided for this purpose consist of color eraser, fill with color, pick color, magnifier, pencil, brush, air brush, text, line, curve, rectangle, polygon, ellipse, and rounded rectangle. All of these tools are used for the purpose of creating drawings. Apart from these tools, the option also provides various color blocks below the drawing area. From these color boxes the desired color can be selected by the user for the purpose of using it in their drawing.

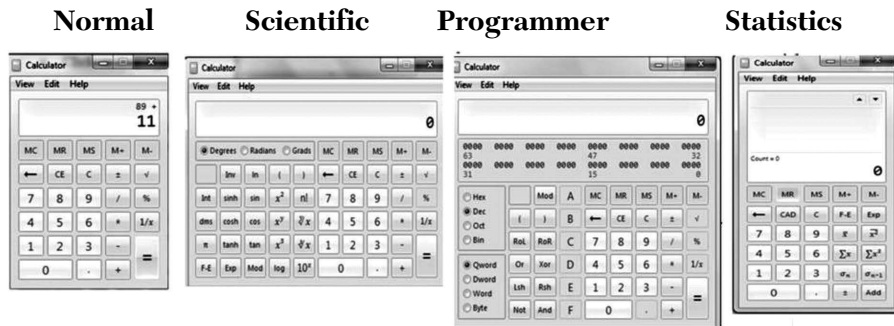


Calculator: The calculator provided by Windows is used for the purpose of normal mathematical calculation. To work with the calculator, the user has to take following steps:

- Click on Start, go to the Calculator option and click on it.
- Alternatively, if it is not visible there, then click on Start, go to the Accessories option, and under it click on the Calculator option.

Once all these steps are taken, a calculator appears on the screen and with the help of mouse or the keyboard the user can perform all the mathematical

operations on the calculator. Calculator provides options such as normal, scientific, programmer, or statistics. The user can select the option and the calculator type will change accordingly.



Desktop in Windows: The Desktop is the screen that appears once Windows is completely loaded. It is the screen from which users can start selecting options for their work. Active desktop makes it possible for the user to customize the desktop, launch programs, and switch between files. The desktop can be changed with the help of following steps:

- Right click the mouse button on a desktop; a drop-down menu will appear on the screen. From that menu select the “personalize” option to customize the desktop.

Taking following steps can make various changes to a desktop:

- The user can install desktop themes by clicking on “Control Panel,” and then double-clicking on “Appearance and Personalization” option. Under this option there are numerous choices to change the settings of the desktop. Some of them are as follows:
 - change the theme
 - change desktop background
 - adjust screen resolution
 - add/uninstall gadgets
 - customize the start menu

Test Your Knowledge

1. Define software. Describe the many types of software.
2. Discuss the various functions performed by an operating system.
3. Explain the various steps involved in the booting process of a computer system.
4. What do you understand about programming languages? Explain the use of assembler, compiler, and interpreter with these programming languages.
5. Explain the development of the programming languages used by a computer system. For what purpose are assemblers, compilers, and interpreters used?
6. Define generations of programming languages. Discuss.
7. What is a machine language? How it is different from a high-level language?
8. Write down the various applications of programming languages with their advantages.
9. Explain multiprogramming and multiprocessing systems.
10. Explain the different types of system software available.
11. What do you understand about the classification of an operating system? Explain.
12. Explain the various components of an operating system.
13. Describe the meaning of a “file.” How it is different from a “folder”?
14. What is a file name? Explain with an example.
15. What is a file extension? What is the significance of it?
16. What is a graphical user interface (GUI)? How it is different from DOS?
17. What is a directory/folder? Write down steps to create a directory/folder.
18. Explain how Windows Explorer in Windows operates.
19. Explain how the control panel in Windows operates.

20. Explain how print manager in Windows operates.
21. Explain how “Paint” in Windows operates.
22. For what purposes does Windows provide the calculator option?
23. Describe desktop in Windows.

