Unit-3

Operating System & MS-DOS

3. Software:

The term "software" was first used in this sense by John W. Tukey in 1957. In computer science,computer software is nothing but all computer programs.To perform any task on computer, the programmer has to write a set of instruction. This sequence of instruction given to the computer is called a program. A set of program written for the computer is called software. Software is a program that enables a computer to perform a specific task, as opposed to the physical components of the system. This includes application software such as a word processor, which enables a user to perform a task, and system software such as an operating system, which enables other software to run properly, by interfacing with hardware and with other software. In computers, software is loaded into RAM and executed in the central processing unit. At the lowest level, software consists of a machine language specific to an individual processor. A machine language consists of groups of binary values signifying processor instructions i.e. object code. Software is an ordered sequence of instructions for changing the state of the computer hardware in a particular sequence. It is usually written in high-level programming languages that are easier and more efficient for humans to use than machine language. High-level languages are compiled or interpreted into machine language object code. Software may also be written in an assembly language, essentially, a mnemonic representation of a machine language using a natural language alphabet. Assembly language must be assembled into object code by the help of assembler. Software is categorized on the basis of application it performs. Software can be classified as:

3.1Custom made software

Custom-made programs are usually composed to meet the processing needs of a specific organization or an individual. The need for custom-made programs arises when the specific needs of users are not matched with the capabilities of any of the packages available in the market. The main advantage of such programs is that they can be tailored to the user's exact specifications.

Creation of specification software involves a series of steps to be taken in an ordered manner:

Step1: **Defining the problem**- the problem or the need of new software may result from a number of reasons, say, changing operating conditions. So, the problem should be defined and defined and specific goals outlined.

Step2: **System Analysis**- system analysis is the study of existing operations to learn what they accomplish, why they work as they do and what role they may have in future

processing activities. A system is a group of parts which are integrated to achieve some objective.
After the need for specific changes is identified, the next step is to analyze data about current processing operations. It involves

- Data collection
- System Flowchart
- Analysis of findings

Step3**: System design-** It is the process of creating alternative solutions for satisfying the set goals, evaluating different choices available and then drawing up the specifications for the chosen alternative solution. Once the best alternative approach is decided, the new specifications which include the outputs desired, the inputs data needed and the processing procedures requires for converting input data into output results are prepared.

Step4**: Programming Analysis-** It is the process in which the new are broken down into four operations.

- Input/Output
- Calculation/text manipulation
- Logic/comparison
- Storage/retrieval operations

This can be done by using analysis tools such as program flowchart.

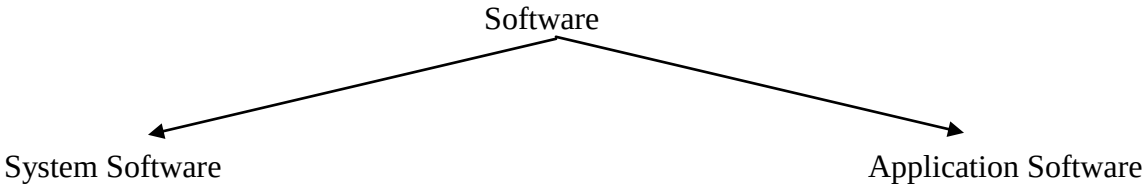
Step5**: Program Generation-** the operations identified during the programming analysis stage is converted into programs in a programming language.

Step6**: Implementation and Maintenance-** the new software is tested and implemented. Finally the successfully implemented systems and programs are usually subjects to continual change for maintenance purposes.

3.2 Pre – written software

Pre-written software packages typically address the processing needs of users. They can be separated into two divisions::

(a)System software packages
(b)Application packages



3.2.1 System Software

As you know that, a set of programs written for the computer is called software. The software required to execute user program is called system software. This software controls all processing activities and make sure that resources and the power of the computer are used in most efficient manner. The major purpose of system software is it controls the execution of program and helps in the development of software.

System software helps in running the computer hardware and computer system. It includes operating systems, device drivers, diagnostic tools, servers, windowing systems, utilities and more. The purpose of systems software is to insulate the applications programmer as much as possible from the details of the particular computer complex being used, especially memory and other hardware features, and such accessory devices as communications, printers, readers, displays, keyboards, etc. System software can be classfied as:

- ✓ System Control Software
- ✓ System Support Software
- ✓ System Development Software

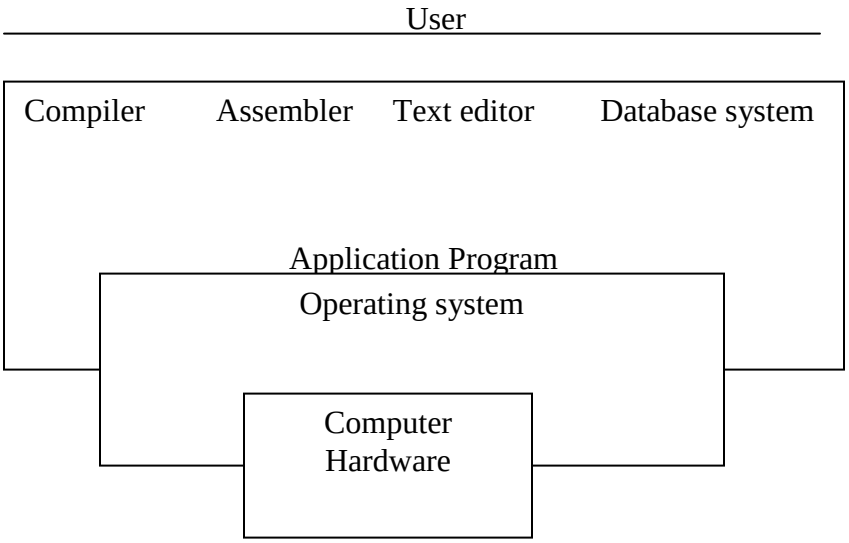
3.2.1.1 System Control Software:

System Control programs control the execution of programs, manage the storage and processing resources of the computer, and perform other management and monitoring functions. The most important of these program is the operating system & DBMS(Data Base Management System).

Operating system

An operating system is a program that acts as an intermediate between the user and computer hardware. It is a computer program that manages the hardware and software resources of a computer. At the foundation of all system software, the OS performs basic tasks such as controlling and allocating memory, prioritizing system requests, controlling input and output devices, facilitating networking, and managing files. It also may provide a graphical user interface for higher level functions. The purpose of an operating system is to provide an environment. In which user can execute the program.

Modern general-purpose computers, including personal computers and mainframes, have an operating system to run other programs, such as application software. Examples of operating systems for personal computers include Microsoft Windows, and Linux. The primary goal of an operating system is thus to make the computer system convenient to use. A secondary goal is to use the computer hardware in an efficient manner.



When the hardware provides the basic computing resources, then the application program defines the way in which these resources are used to solve certain problem. Operating system controls and coordinates the use of hardware among the application program. Efficiency of operating system can be measured on the basis of following 3 factors.

1. Turn around time : It is the time delay between the submission and completion of any job.
2. Response time: It is the time taken by the system to give first response.
3. Throughput: It is the no of job executed in an unit time.

A Function Of Operating system

- Process Management
- Memory Management
- File Management
- Device Management

Process Management: A program does nothing unless and until its instruction are executed by a CPU. A process can be thought of as a program in execution. Every action on a computer, be it background services or applications, is run inside a process. In older operating system, only one process per CPU can be run at a time. Older OS such as DOS did not attempt any artifacts to bypass this limit and only one process could be run under them. Modern operating systems are able to simulate execution of many processes at once via multitasking even with one CPU. Process management is an operating system's way of dealing with running multiple processes. Process management involves computing and distributing "timeshares". Most OSs allow a process to be assigned a process priority which impacts its timeshare. The OS is responsible for the following activities:

1. The creation and deletion of both the user and system processes.

2. The suspension and resumption of process
3. The provision of mechanism for process synchronization
4. The provision of mechanism for dealock handling.

When a program is submitted to the CPU, it may be one of the three states:

- Running
- Ready
- Wait

Running : A process is running when its instruction sequence is being executed by the processor.

Ready: A process is ready if all the conditions are satisfied for the process to be in the running state and it is waiting for the processor.

Wait: A process is blocked when it is waiting for an event to occur, before continuing execution.

Job scheduling is also a part of process management, there are two types of scheduling techniques:

- Non Preemptive Scheduling
- Preemptive Scheduling

Non Preemptive Scheduling:

In this type of scheduling, a scheduled job always completes first before another scheduling decision is made. In this the job finishes there working in the order in which they are scheduled. There are certain different types of Non Preemptive scheduling techniques, they are:

- FCFS (First Come First served)
- SJF(Shortest Job First)

First Come First Served: It is one of the simplest scheduling technique, i.e. the job is executed in the order of there arrival. Batch processing is an example of FCFS scheduling . in this case turn arround time for the very first job is the best and the for the very last job, it is the worst..

Shortest Job First: In this type of scheduling , a process or job is executed on the basis of having shortest execution time. If the two process have the same execution time, FCFS is used.

Preemptive Scheduling

In preemptive scheduling, as cheduling decision can be made even while the job is executing wherease in non preemptive scheduling , a scheduling decision can be made only when some job finishes its execution.

Round Robin Scheduling Algorithm:

In this the CPU time is divided in to small parts which is known as time slices. Each process allocated a small time slice while it is running. No process can run for more than one time slice when there are others process waiting in the queue. If the process needs more CPU time to complete after exhausting one time slice, iot goes to the end of the queue to wait for the next alloccation.

Memory Management: The main memory is central to the operation of a modern computer. Main memory is a large array of words or bytes. Each words or bytes has its own address. For a program to be executed, it must be mapped to the absolute address and loaded into the memory. As the program executes it access program instruction from the memory by generating these absolute address. Besides this, the memory manager in an OS coordinates the memories by tracking which one is available, which is to be allocated or deallocated and how to swap between the main memory and secondary memories. This activity which is usually referred to as virtual memory management greatly increases the amount of memory available for a process.

Another important part of memory management activity is managing virtual addresses, with help from the CPU. If multiple processes are in memory at once, they must be prevented from interfering with each other's memory. This is achieved by having separate address spaces. Each process sees the whole virtual address space as uniquely assigned to it. The CPU stores some tables to match virtual

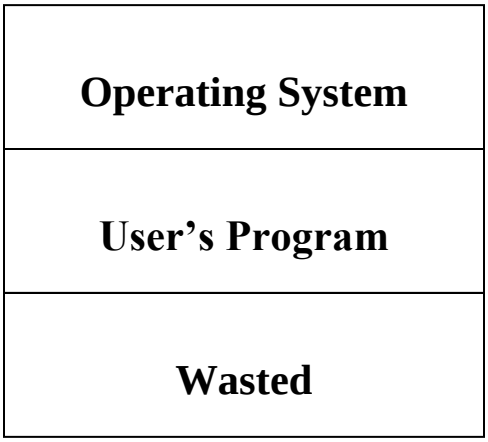
addresses to physical addresses. This process, as it is known, is called paging. The OS is responsible for the following activity:

1. Keep track of which part of the memory are currently being used and by whom
2. Decide whichprocess is to be loaded in to the memory when the memory space becomes available.
3. Allocate and deallocate the memory space as needed.

Techniques For Memory Management

Single Contiguous Allocation

In this allocation, the operating system usually resides in either the upper or lower part of the core memory. A job is assigned all of the core although it typically desires and uses only a small fraction. The job has complete control on the CPU until completion. In this resources are not managed efficiently. There is not enoughhardware flexibility to allow effective allocation of memory. This type of allocation is used for small computing system.



Partitioned Allocation

This technique is used to solved the problem of wasted memory. This approach is based on partitioning the memory to accommodate more programs in to the main memory. There are two ways for this type of allocation:

- Fixed Size Partiton Allocation
- Variable Size Partitioning Allocation

A. Fixed Size Partiton Allocation

In this the main memory is divided in to fixed partitions. This partitions can be of equal size or unequal size.the number and sizes of partitions are fixed, thus also fixing the degree of multiprogramming. In this case user must be aware of the size of partiton:

Case1: In case of equal partiton , if a program of size 30 k then it cannot resides in the memory, therefore cannot be executed. Similarly if a program size is 8K or 12K the the memory will be wasted.

Case2: In case of unequal sized partitionsif aprogram size is of 10 kbthen it can be executed in any of the partition except * KB but again memory is wasted.

To overcome this type of problem variable sized partitioning is made:

B. Variable Size Partitioning:

In this a memory of exactly the same size is alklocated when a program is brought into the memory. Thus no wastage of memory is there. As job terminates, the system keep track of the released storage. When a new job is initiated , it creates a partition to suit its storage requirement.

File Management: A file isa collection of related information. Commonly file represent programs and data. A file consist of a sequence of bits, byte, lines or record whose meaning are defined by the creators. File management is one of the important component of the operating system. Computer can store information on several different type of information media such as magnetic disk or magnetic tape etc. The operating system is responsible for the following activities:

1. The creation and deletion of file.
2. The creation and deletion of directories.
3. The mapping of file on to the secodary storage.
4. The backup files on stable storage media.

Device Management: A computer system comprises of peripheral devices apart from the basic CPU. These peripheral devices are usually the input/output devices like printer, drives, tapes, card readers etc. These type of devices also have to be managed by the operating system. The operating system is responsible for the following activities:

1. It should activate them, control them, issue command to the device, handle errors etc.
2. It should also provide an interface between the devices and the system, so that it becomes easy to use this devices.
3. The operating system provide coordination between all the peripheral devices, instead of they arev quite different.

Security

Many operating systems include some level of security. Security is based on the two ideas that:

- The operating system provides access to a number of resources, directly or indirectly, such as files on a local disk, privileged system calls, personal information about users, and the services offered by the programs running on the system;

- The operating system is capable of distinguishing between some requestors of these resources who are authorized (allowed) to access the resource, and others who are not authorized (forbidden). While some systems may simply distinguish between "privileged" and "non-privileged", systems commonly have a form of requestor identity, such as a user name. Requestors in turn divide into two categories:
- Internal security: an already running program. On some systems, a program once it is running has no limitations, but commonly the program has an identity which it keeps and is used to check all of its requests for resources.
- External security: a new request from outside the computer, such as a login at a connected console or some kind of network connection. To establish identity there

may be a process of authentication. Often a username must be quoted, and each username may have a password.

B Need Of Operating System

1. The operating system provides access to a number of resources, directly or indirectly, such as files on a local disk.
2. The operating system is capable of distinguishing between some requestors of these resources who are authorized to access the resource, and others who are not authorized.
3. Many operating systems allow the user to install or create any user interface they desire.
4. A device driver is a specific type of computer software developed to allow interaction with hardware devices.

C Types Of Operating System

Batch Operating System

Batch operating system is one of the earliest operating system. In this the user does not has any interaction with the user. In this the user take his/her request to the operator. The operator than prepare a punch card and the card reader is used to read the information from the card. In this type of operating system, to speed up the processing the program of similar needs were batched together and were run to the computer as a group. This operating system takes hours or even sometimes days to give the output. In these OS only one job is processed at a time,therfore, sometime is also known as single user operating system.

Multiprogramming Operating System

The intermediate execution of two or more program is known as multiprogramming.It is a multi user operating system. Some important terms whichis used in these operating system:

Job Pool: It is queue of task stored on the disk.

Job Scheduler: It decides which program goes to the main memory.

CPU Scheduler: Executes the program selected by the job scheduler.

The basic idea behind the multiprogramming operating system is that the CPU should not remain idle at any instant of time. This work is done by lining of the programs in main memory means the the program is in ready state. Then scheduler picks up the program for execution means the program is now in active state.Suppose the active program does not need CPU at a given time then, it comes under the wait

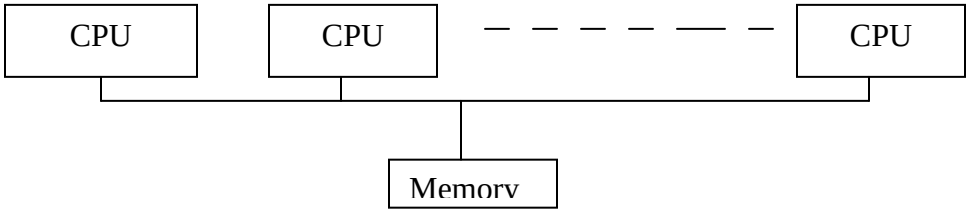
state.Meanwhile the CPU is given another program for execution. When the previous process completes its I/O operations it enters to the ready queue. The advantages of multiprogramming operating system is:

Short Response Time: The multiprogramming operating system has a very short response time because the CPU devoting its time to each of the process in turns. The majoer aspect in any response time is the CPU idle time, which is removed or reduced so that it results in the shporter response time.

Increased Productivity: Productivity is a measure of the total amount of processing which can be done by a CPU in a fixed amount of time. Multiprogramming operating system is capable to execute more than one program at a time. When there is a need of input/output the CPU instead of waiting will swiotch over the next program, by that time the input/outp[ut operation takes place, thereby there is no wastage of CPU time, thus this type of operating system can execute more number of programs in a given instance of time, which leads to higher productivity of the system.

Multiprocessing Operating System

The multiprocessing operating system is capable of handling more than one processor. Such system have more than one processor in close communication, sharing the computer bus, clock, sometimes memory and peripheral devices. In these types of operating system, by increasing the no of processor, we hope to get more work done in less time. These type of operatng system can save money because they can share peripherals, mass storage and power supplies. If several program operate on same set of data then it is cheaper to store those data on one disk and have all the processor share them. If function can be distriubted properly among all the processor then the failure of one processor should not halt the system. Instead of this, the work of failed processor has been taken up by the remaining processor.



The most common multiprocessing operating system in use is Symmetric multiprocessing operating system refers to the OS that has the abiity to assign tasks dynamically to the next available processor, wherease asymmetrical multiprocessing requires that the original program designer choose the processor to use for a given taskat a time of writing a program.

Time- Sharing Operating System

It make use of time sharing technique. It is an interactive computer systemthat provide communication between the user and the system. The user gives the instruction to9 the

operating system directly by making use of a keyboard , or a mouse. It allows many user to share the computer simultaneosy. In this only a little CPU time is shared for each user. It make use of CPU scheduling alogrithms for executing the programs.

Time sharing operating system are more complex than multiprogrammed operating system. In both several jobs must be kept in memory, so the system must has memory management. It also has file management as well as disk management. In Time sharing operating system the time is distributed in between each user.

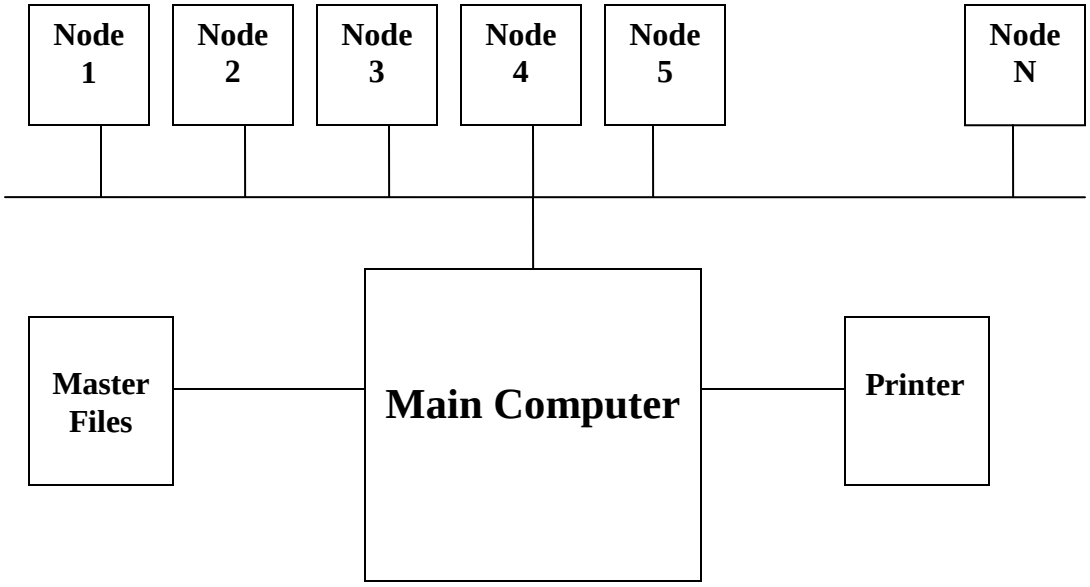
Real Time Operating System

It is a type of special purpose operating system. This type of operating system is used when there is a rigid time requirement for the completion of any job or in other words there is a fixed deadline, job must be completed on that deadline. In this sensor bring data to the computer. Then the computer must analyze the data and modify the input. These operating system has a well defined and fixed time constraint and processing must be done under this constraint, otherwise, the system will fail.

Any computer system designed to respond immediately, as event occurred in the real world. Computer system used in airplane automatic pilots, patient monitoring system, process control, or traffic control system are all real time operating system.

Multi-User Operating System

This operating system takes the best advantage of CPU. Here, many users are connected through different terminals. Each user work at his own terminal. the operating system of the computer allows many user to work simultaneously by assisting user a part of RAM and it divides the computer time among various user. Here the computer gives a user his time and then goes to the other user giving him his quota of time and then shift to the other user and this process will go on. As the environment is multi user, there is a great amount of security provided, so that no one can tamper each others data.



3.2.1.2 System Development Software

It assists in the creation of application programs. An example of system development software is Language Processor.

Language Processor

When a program written in a language other than the machine language of computer, the computer will not understand it. Hence, the program written in other language must be translated into the machine language of the computer. Such translation is done with the aid of software. This type of software is known as language processor.

A Compiler

A compiler is a computer program or set of programs that translates text written in a computer language i.e. in the source language into another computer language i.e. the target language. The original sequence is usually called the source code and the output called object code. The most common reason for wanting to translate source code is to create an executable code. The name "compiler" is primarily used for programs that translate source code from a high level language to a lower level language or machine language. A program that translates from a low level language to a higher level one is a decompiler. A compiler is likely to perform many or all of the following operations: lexing, preprocessing, parsing.

Early computers did not use compilers. Compilers had not yet been invented because early computers had very little memory and programs were necessarily quite short. Users often entered the decimal or binary machine code for a program. With the evolution of programming languages and the increasing power of computers, compilers are becoming more and more complex to bridge the gap between problem-solving modern programming languages and the various computer.

A compiler is itself a computer written in some implementation language. Early compilers were written in assembly language. The first self-hosting compiler which is capable of compiling its own source code in a high-level language — was created for Lisp.

- A program that translates from a low level language to a higher level one is a decompiler.
- A program that translates between high-level languages is usually called a language translator.

Most compilers are classified as either self-compilers or cross compilers. If a compiler run on a computer for which it produces the object code, then it is known as self-compiler. If a compiler run on a computer other than that for which it produces the object code, then it is known as cross compiler.

Compiled versus interpreted languages

Many people divide higher-level programming languages into compiled languages and interpreted languages. However, there is rarely anything about a language that requires it to be compiled or interpreted. Compilers and interpreters are *implementations* of languages, not languages themselves. The categorization usually reflects the most popular or widespread implementations of a language -- for instance, BASIC is thought of as an interpreted language, and C a compiled one, despite the existence of BASIC compilers and C interpreters.

There are exceptions; some language specifications spell out that implementations must include a compilation facility (eg, Common Lisp), while other languages have features that are very easy to implement in an interpreter, but make writing a compiler much harder; for example, SNOBOL4, and many scripting languages are capable of constructing arbitrary source code at runtime with regular string operations, and then executing that code by passing it to a special evaluation function. To implement these features in a compiled language, programs must usually be shipped with a runtime environment that includes the compiler itself.

B Assembler

A program that converts an assembly language program into machine language so that computer can run the program. As you know that assembly language program can be written by making use of mnemonic code (symbolic code). The use of symbolic

references is a key feature of assemblers, saving tedious calculations and manual address updates after program modifications. Assemblers are available since the 1950s. An assembler, which runs on the computer for which it produces the object code, is called self-assembler. The assembler runs on the computer other than that for which it produces the object code is called cross assembler. There is two types of assembler, first is one pass assembler, it is an assembler which read the program once and assign addresses to the labels used in assembly language program. Second is Two pass assembler, which goes to the program twice, in first pass it will assign address to the labels and in the second pass it will convert each assembly language instruction into machine language instruction.

More sophisticated High-level assemblers provide language abstractions such as:

- Advanced control structures.
- High-level procedure/function declarations and invocations.
- High-level abstract data types, including structures/records, unions, classes, and sets.
- Sophisticated macro processing.

C Interpreter

An interpreter is a program which translates a high level language rogram into machine level language. It translates one instruction of a program at a time. If it is correct then only it proceeds towards the next instruction. It reads the instruction, translate it,and after that it executes the instruction. An interpreter is a smaller program as compared to the compiler. It occupy less memory space. Interpreting code is slower than running the compiled code because the interpreter must analyse each statement in the program each time it is executed and then perform the desired action whereas the compiled code just performs the action. Access to variables is also slower in an interpreter because the mapping of identifiers to storage locations must be done repeatedly at run-time rather than at compile time. The IBM 550 Numeric Interpreter and IBM 557 Alphabetic Interpreter are typical examples of the interpreter.

Difference Between Compiler & Interpreter

Compiler	Interpreter
Scans the entire program first and translate it in to machine code.	Translates the program line by line.
Compiler produces object code.	During conversion, the interpreter does not produce its object code.
Converts the entire program to machine code; when all the syntax errors are removed, execution takes place.	Each time the program is executed, every line is checked for syntax error and then converted to equivalent machine code.

Slow for debugging (removal of mistakes from the program).	Good for fast debugging
Execution time is less	Execution time is more.

3.2.1.3 System Support Software

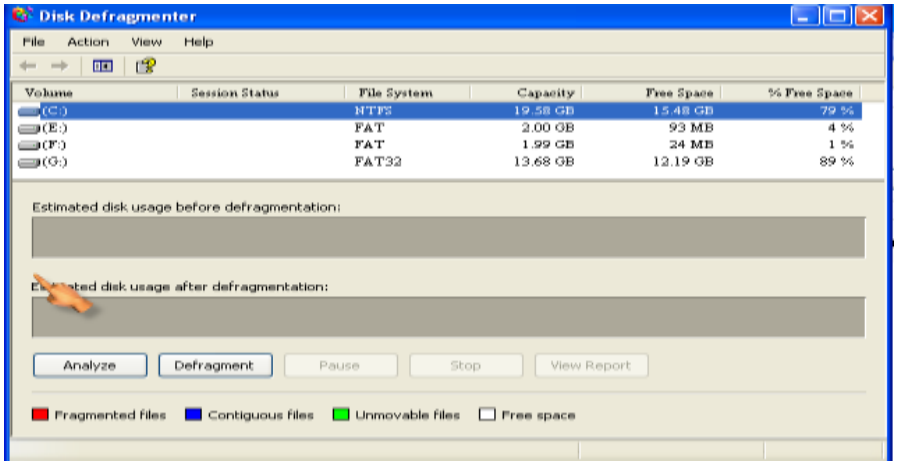
System support program provide routine service functions to the other computer programs and computer users. Example of this type of software is Utility software.

Utility Software

It is a set of program, that supports the operating system by providing the additional services that the operating system does not provide. There are many task which are performed by utility programs are hard disk backup, disk optimization, file recovery, safe formatting and resource editing.

Utility software is also known as service program, or utility routine. It is specifically designed to help in managing and tune the computer hardware, operating system or application software, and perform a single task or a small range of tasks.Some important utilities are bein discussed below:

- **Disk defragmenters/Disk management tools** Disk defragmenter can detect computer files whose contents have been stored on the hard disk in disjointed fragments, and move the fragments together to increase efficiency. There is a Disk checker which can scan the contents of a hard disk to find files or areas that are corrupted in some way, or were not correctly saved, and eliminate them for a more efficiently operating hard drive. A Disk cleaner can find files that unnecessary to computer operation, or take up considerable amounts of space. Disk cleaner helps the user to decide what to delete when his hard disk is full. To start disk defragmentation we have to click start>Programs>Accessories>System tools>Disk Defragmentation. It will show like this



- **Virus scanners/Antivirus**. Virus Scanners scan for computer viruses among files and folders. Virus is a program intended to damage your computer sytem without your knowledge and belief. A virus may itself attach to another programon your hard disk and when the date passes, or a certain event occurs, the virus is trigerred into action. The most famous virusis jerusalem virus, which is also known as Friday the 13^{th}, first seenat the university of jerusalem in july 1987. The virus slow down the system. There are several precautions that you can take to protect yoursef from infection, including backing up your system on the regular basis , and you can also buy and run the commercially available virus detecting program.

  Antivirus is an application program that can detect and eliminate the computer virus. Some antivirus program can detect suspicious activity on your computer as it happens. While, the other must be run periodically. The antivirus program locates and identifies a virus by looking some of the characteristics, like unexpected disk access. It recognizes the virus by comparing the information from the system against the database of known virus that is kept n the disk. There are several simple precautions that you can take to minimize your chances of contracting a virus:

  1. Back up your hard disk regularly.
  2. Do not install software if you don't know where it's been or where it came from.
  3. Write protect disks as soon as you get them.

- **Compression** utilities output a shorter stream or a smaller file when provided with a stream or file. A file that has been processed by a special utility program so that it occupies a little hard disk space. When the file is needed, the same program decompresses the file back into its original form so that it can be read by the computer.

  Two kind of file compression program are available those program that can compress more than one file at a time such as winzip. Those program that can compress al the files on a specific disk. Any method of encoding data so that it occupies less space than its original form.

- **File Management tools** A file manager or file browser is a computer program that provides a user interface to work with file systems. They are very useful for speeding up interaction with files. The most common operations on files are create, open, edit, view, print, play, rename, move, copy, delete, attributes, properties, search/find, and permissions.

- **Encryption** utilities use a specific algorithm to produce an encrypted stream or encrypted file when provided with a key and a plaintext.

3.1.2 Application Software Packages

Application software allows humans to accomplish one or more specific tasks.It is a set of program that is necessary to carry out operation for a specified application. Typical applications include industrial automation, business software, educational software, medical software, databases. Almost every field of human activity now uses some form of application software. It is used to automate all sorts of functions.

Application software or Applications are what most people think of when they think of software. Application software is often purchased separately from computer hardware. Sometimes applications are bundled with the computer, but that does not change the fact that they run as independent applications. Applications are almost always independent programs from the operating system

Customized Application software: The software which is developed to meet all the requirement specified by the user. Custom-made programs are usually composed to meet the processing needs of a specific organization or an individual. The need for custom-made programs arises when the specific needs of users are not matched with the capabilities of any of the packages available in the market. The main advantage of such programs is that they can be tailored to the user's exact specifications.

General Application software: The software which is developed by keeping all the general requirements in mind for carrying out a specific task.These are those software which are developed by the group of people or an individual to be use by the other.For e.g. word processing software, Electronic spread sheet etc.

Application packages

The key to making the computer useful is to combine it with software in a particular application. An application is the job a user wants the computer to perform. An application package is a set of programs that allow the computer to perform a specific data-processing job for the user. Application software helps the user to work faster, more efficiently and more productively.

Application packages may perform:

(i) **single functions**: These can be divided into

- **special purpose packages:** are packages designed for a specific class of users. For instance, medical diagnostic packages.
- **General purpose packages:** are packages that can be applied to a variety of tasks. For instance, word processing packages, data manager, electronic spreadsheet graphics.

(ii) integrated functions: these combine several general purpose functions in a single product. They include most of the following functions.

- **Processing words**

- **Manipulating spreadsheet**
- **Managing database or files**
- **Preparing graphics**

An integrated package makes it possible to use a common set of instructions to share data and move data among several applications. For instance, word processing program of an integrated package would allow a user to use the keyboard and display screen of a computer for creating and editing a text of a report.

3.1.2.1 Application software consist of Packages and Business software.

A Packages

- Word Processor
- Electronic Spreadsheet
- Database Management System
- Desktop publishing

B Business Software

- Inventory Management System
- Payroll System
- Financial Accounting

3.3 Programming Language

The process of designing, writing, testing, debugging, documenting and maintaining a program is known as programming. A language is a medium, which is used for understanding each other ideas. Communicating medium should be such that one has to understand, what we are expressing i.e., medium of communication should be common between the two. Similarly we need computer languages for giving instructions to the computer. A language used to write a program that a computer can execute. Almost, 200 different programming languages exist. Some are well suited to all sorts of computing task. However there are certain kind of tasks particularly those involving artificial intelligence, process control, or very high mathematical applications that can be benefit from a more specific language.

Programming languages provide various ways of specifying programs for computers to run. Unlike natural languages, programming languages are designed to permit no ambiguity and to be concise. They are generally either translated into machine language before being run by a compiler or an assembler or translated directly at run time by an interpreter. Programming languages are classified as follows:

- Machine Language.
- Assembly Language.

- High Level Language.

A. Machine Language

Machine language is a system of instructions and data which is directly understandable by a computer's central processing unit. In this instructions are patterns of bits. Every CPU model has its own machine code, although there is considerable overlap between some. If CPU A understands the full language of CPU B it is said that A is compatible with B. CPU B may not be compatible with CPU A, as A may know a few codes that B does not. Machine language is built up from discrete statements or instructions. Depending on the processing architecture, a given instruction may specify:

- Particular registers for arithmetic, addressing, or control functions
- Particular memory locations or offsets
- Particular addressing modes used to interpret the operands

Some operations which is available in most instruction sets include:

1. **Moving**
    a. set a register to a fixed constant value
    b. move data from a memory location to a register, or vice versa. This is done to obtain the data to perform a computation on it later, or to store the result of a computation.
    c. read and write data from hardware devices
2. **Computing**
    a. add, subtract, multiply, or divide the values of two registers, placing the result in a register .
    b. compare two values in registers

The program written in these language are non-portable. These are the language which are understood by the machine directly as it consist of a binary digit. Instruction written in machine language has two parts, first is opcode which tells the computer what operation has to be performed. The second part of the instruction is operand that specify the address of the data.

Advantages Of machine Language

1. The machine easily understands these languages.
2. Processing is faster.

Limitations Of Machine Language

1. It is machine dependent; it is different from one computer to another.
2. All the instructions are to be given into binary digits, therefore it is very difficult to program.
3. It is difficult to rectify a program written in machine language.

4. It is very time consuming and tedious task to write a program in machine language.

B. Assembly Language

Writing the programs in machine language is a very tedious and boring job. To solve this problem, assembly languages is developed which makes use of alphanumeric symbols for writing the set of instruction instead of 0 and 1. Assembly languages are specific to a given microprocessor, and therefore, it is not portable. In this program written for one type of processor must be rewritten before they can be used for another type of processor. A program written in assembly language consists of a series of instructions in symbolic code i.e. mnemonics that correspond to a stream of executable instructions, when translated by an assembler, that can be loaded into memory and executed.

The equivalent assembly language representation is easier to remember (more mnemonic):

- mov al, 061h

This instruction means, Move the hexadecimal value 61 (97 decimal) into the processor register named "al".

The mnemonic "mov" is an operation code or opcode, and was chosen by the instruction set designer to abbreviate "move." A comma-separated list of arguments or parameters follows the opcode; this is a typical assembly language statement. Transforming assembly into machine language is accomplished by an assembler, and the reverse by a disassembler.

Advantages Of Assembly Language

1. Easy to understand because of mnemonic code.
2. Easy to detect and correct error.
3. Eassily modified
4. It saves time and reduce complexity than machine language program.
5. The computation time of assembly anguage program is less.

Limitations

1. The assembly language is machine oriented. This means that program must have the detailed knowledge of the structure of the computer.
2. Assembly language program contains more instruction as compared to high-level languages.

C. High Level Language

To overcome the difficulties associated with assembly language, high level language has been developed. It is a problem oriented language. A high-level programming language is a programming language that, in comparison to low-level programming languages, may be more abstract, easier to use, or more portable across platforms. Such languages often abstract away CPU operations such as memory access models and management of scope. high-level languages deal with variables, arrays and complex arithmetic or boolean expressions. In addition, they have no opcodes that can directly compile the language into machine code, unlike low-level languages like Assembly language. In general, high-level languages make complex programming simpler, while low-level languages tend to produce more efficient code. In a high-level language, complex elements can be broken up into simpler.

The advantage of high level language program is that they are portable from computer to computer. But the computer can only understand the machine language, so we have to translate the instruction from high level language to machine language. For that, we need language translator which is interpreter and compiler.Language translator is a program which is used to translate a language into a low-level programming language for which native code compilers are already widely available.

Compiler: The name "compiler" is primarily used for programs that translate source code from a high level language to a lower level language or machine language.

Interpreter: An interpreter is a program, which translates a high-level language rogram into machine level language. But, it translates one instruction of a program at a time.

Advantages Of High Level Language

1. It is a machine independent language.
2. They are easier to learn and understand.
3. Program written in this anguage are portable and easier to maintain.
4. In this program can be easily written because natural language English is used.

Limitations

1. The high level language program take more time to run.
2. Computer are unable to understand instruction given in high level language.

Difference Between Low Level, Middle Level & High Level Language.

Low Level Language	Middle Level Language	High Level Language
Eassily understood by the machine	Partially Understood by the machine	Not understood by the machine
Processing is faster	Processing is little slower	Processing is very slower
Machine dependent	Machine oriented language	Machine independent

language		language.
Difficult ot program because it make use of only 0 & 1.	Easy to program because it make use of mnemonic code.	Very easy to program because it uses simple English word.
Difficult to detect error.	Eassy to detect error.	Errors are detected are very eassily.
No translator is required	Assembler is required	Compiler & Interpreter are used as a translator.
It is very time consuming to write programs in it.	It is less time consuming to write programs.	Programs are easier & Faster to write.

3.4 Generation Of Languages:

First Generation Languages: A first-generation programming language is a machine-level programming language. It consists of 1s and 0s. Originally, no translator was used to compile or assemble the first-generation language. The first-generation programming instructions were entered through the front panel switches of the computer system. The main benefit of programming in a first-generation programming language is that the code a user writes can run very fast and efficiently, since it is directly executed by the CPU. 1GL is mainly used on now very ancient computers, machine level programming still finds a use in several areas of modern programming.

Second Generation Languages: These languages were developed between late 1950's and early 1960's. These language introduce many new concepts into the way and style of programming. The concept of data types are developed, the structured programming concept developed, which laid a strong foundation for current languages. ALGOL-60, COBOL, LISP etc are the popular languages of this generation have introduced sophisticated programming construct, both conditional and iterative constructs.

Third Generation Languages: Late 1960 and early 1970 witnessed the evolution of third generation languages. The languages of this generation introduced the concept of record structure and classes. The programming language become simple and one of the easiest task, with more and more features for better and efficient programming like the concept of arrays, pointers, storage class, exception handling etc. Simula 67, BASIC, SNOBOL, ALGOL-68 etc were some of the popular languages for this generation.

Fourth Generation Languages: A fourth-generation programming language(1970s-1990) (abbreviated 4GL) is a programming language or programming environment designed with a specific purpose in mind, such as the development of commercial business software. In the evolution of computing, the 4GL followed the 3GL in an upward trend toward higher abstraction and statement power. The 4GL was

followed by efforts to define and use a 5GL. These languages interact with Database management system tools for storing, manipulating and retrieving data.

All 4GLs are designed to reduce programming effort, the time it takes to develop software, and the cost of software development. They are not always successful in this task, sometimes resulting in inelegant and unmaintainable code.

Types

A number of different types of 4GLs exist:

- Report generators take a description of the data format and the report to generate and from that they either generate the required report directly or they generate a program to generate the report.
- Similarly, forms generators manage online interactions with the application system users or generate programs to do so.
- Data management 4GLs such as SAS, SPSS and Stata provide sophisticated commands for data manipulation, file reshaping, case selection and data documentation in the preparation of data for statistical analysis and reporting.

Some successful fourth-generation languages

- Database query languages
  - FOCUS
  - SQL

- Report generators
  - BuildProfessional
  - Oracle Reports
  - PostScript
  - Progress 4GL Query/Results

- Data manipulation, analysis, and reporting languages

  - Clarion Programming Language
  - Ab Initio
  - ABAP
  - ADS/Online (plus transaction processing)
  - PL/SQL
  - Synon

- Data-stream languages
  - APE
  - AVS
  - Iris Explorer

- GUI creators
    - Borland Delphi
    - MATLAB's GUIDE
    - Progress 4GL AppBuilder
    - Visual Basic's form editor
    - Windows Forms (part of the .NET Framework)
- Database driven GUI Application Development
    - Powerbuilder

Fifth Generation Languages: A fifth-generation programming language (abbreviated 5GL) is a programming language based around solving problems using constraints given to the program, rather than using an algorithm written by a programmer. Most constraint-based and logic programming languages and some declarative languages are fifth-generation languages.

While fourth-generation programming languages are designed to build specific programs, fifth-generation languages are designed to make the computer solve the problem for you. This way, the programmer only needs to worry about what problems need to be solved and what conditions need to be met, without worrying about how to implement a routine or algorithm to solve them. Fifth-generation languages are used mainly in artificial intelligence research. Prolog, OPS5, and Mercury are the best known fifth-generation languages.

5GL or fifth-generation programming language is programming that uses a visual or graphical development interface to create source language that is usually compiled with a 3GL or 4GL language compiler. These types of languages were also built upon Lisp, many originating on the Lisp machine.

3.5 Computer processing techniques

A. Batch processing:

Batch processing is the execution of a series of programs ("jobs") on a computer without human interaction, when possible. Batch jobs are set up so they can be run to completion without human interaction, so all input data is preselected through scripts or commandline parameters. This is in contrast to interactive programs which prompt the user for such input.

Batch processing has these benefits:

- It allows sharing of computer resources among many users
- It shifts the time of job processing to when the computing resources are less busy
- It avoids idling the computing resources without minute-by-minute human interaction and supervision

- It is used on expensive classes of computers to help amortize the cost by keeping high rates of utilization of those expensive resources.

Batch processing has historically been synonymous with mainframe computers. Since this class of computer was so expensive, batch processing was used for the reasons listed above. Also, in the early days of electronic computing, interactive sessions with computer terminal interfaces (and later Graphical user interfaces) were not yet widespread.

Batch processing has grown beyond its mainframe origins, and is now frequently used in UNIX environments, where the cron and at facilities allow for scheduling of complex job scripts. Similarly, Microsoft DOS and Windows systems refer to their command-scripting language as batch files and Windows has a job scheduler. There are several disadvanmtage nto batch processing:

- It involves accumulating data into batches which causes delay in job.
- Since it takes a fixed time interval before current data is added, it is not timely. Therefore a system that utilizes

B. On-line Processing:

OLAP is an acronym for **On Line Analytical Processing**. It is an approach to quickly provide the answer to analytical queries that are dimensional in nature. It is part of the broader category business intelligence, which also includes Extract transform load (ETL), relational reporting and data mining. The typical applications of OLAP are in business reporting for sales, marketing, management reporting, business process management (BPM), budgeting and forecasting, financial reporting and similar areas. The term OLAP was created as a slight modification of the traditional database term OLTP (**On Line Transaction Processing**). Databases configured for OLAP employ a multidimensional data model, allowing for complex analytical and ad-hoc queries with a rapid execution time. The output of an OLAP query is typically displayed in a matrix (or pivot) format. The dimensions form the row and column of the matrix; the measures, the values.

C. Real time processing (Transaction processing):

In computer science, **transaction processing** is information processing that is divided into individual, indivisible operations, called *transactions.* Each transaction must succeed or fail as a complete unit; it cannot remain in an intermediate state.

Transaction processing is designed to maintain a database in a known, consistent state, by ensuring that any operations carried out on the database that are interdependent are either all completed successfully or all cancelled successfully.

For example, consider a typical banking transaction that involves moving $500 from a customer's savings account to a customer's checking account. This transaction is a single operation in the eyes of the bank, but it involves at least two separate operations in

computer terms: debiting the savings account by $500, and crediting the checking account by $500.

Transactions are processed in a strict chronological order. If transaction *n+1* touches the same portion of the database as transaction *n*, transaction *n+1* does not begin until transaction *n* is committed. Before any transaction is committed, all other transactions affecting the same part of the database must also be committed; there can be no "holes" in the sequence of preceding transactions.

1. Rollback

Transaction-processing systems ensure database integrity by recording intermediate states of the database as it is modified, then using these records to restore the database to a known state if a transaction cannot be committed

2. Rollforward

It is also possible to keep a separate journal of all modifications to a database (sometimes called *after images);* this is not required for rollback of failed transactions, but it is useful for updating the database in the event of a database failure, so some transaction-processing systems provide it. If the database fails entirely, it must be restored from the most recent back-up.

3. Deadlocks

In some cases, two transactions may, in the course of their processing, attempt to access the same portion of a database at the same time, in a way that prevents them from proceeding. For example, transaction A may access portion X of the database, and transaction B may access portion Y of the database. If, at that point, transaction A then tries to access portion Y of the database while transaction B tries to access portion X, a *deadlock* occurs, and neither transaction can move forward. different order, automatically, so that the deadlock doesn't occur again.

D. Multiprogramming:

In the early days of computing, CPU time was expensive, and peripherals very slow. When the computer ran a program that needed access to a peripheral, the CPU would have to stop executing program instructions while the peripheral processed the data. This was deemed very inefficient.

The first efforts to create multiprogramming systems took place in the 1960s. Several different programs in batch were loaded in the computer memory, and the first one began to run. When the first program reached an instruction waiting for a peripheral, the context of this program was stored away, and the second program in memory was given a chance to run. The process continued until all programs finished running.

Multiprogramming doesn't give any guarantee that a program will run in a timely manner. Indeed, the very first program may very well run for hours without needing access to a peripheral. As there were no users waiting at an interactive terminal, this was no problem: users handed a deck of punched cards to an operator, and came back a few hours later for printed results. Multiprogramming greatly reduced the waiting.

E. Multiprocessing:

Multiprocessing is a generic term for the use of two or more central processing units (CPUs) within a single computer system. There are many variations on this basic theme, and the definition of multiprocessing can vary with context, mostly as a function of how CPUs are defined (multiple cores on one die, multiple chips in one package, multiple packages in one system unit, etc.).

Multiprocessing sometimes refers to the execution of multiple concurrent software processes in a system as opposed to a single process at any one instant. However, the term multiprogramming is more appropriate to describe this concept, which is implemented mostly in software, whereas multiprocessing is more appropriate to describe the use of multiple hardware CPUs. A system can be both multiprocessing and multiprogramming, only one of the two, or neither of the two.

F. Time sharing

Time-sharing refers to sharing a computing resource among many users by multitasking.Because early mainframes and minicomputers were extremely expensive, it was rarely possible to allow a single user exclusive access to the machine for interactive use. But because computers in interactive use often spend much of their time idly waiting for user input, it was suggested that multiple users could share a machine by using one user's idle time to service other users. Similarly, small slices of time spent waiting for disk, tape, or network input could be granted to other users.

Other historical timesharing systems, some of them still in widespread use, include:

- IBM CMS (part of VM/CMS)
- IBM TSS/360 (never finished; see OS/360)
- IBM Time Sharing Option (TSO)
- KRONOS (and later NOS) on the CDC 6000 series
- Michigan Terminal System
- Multics
- MUSIC/SP

3.6 Introduction To DOS

DOS commonly refers to the family of closely related operating systems which dominated the IBM PC compatible market between 1981 and 1995. This is a single user, single task systems. MS-DOS from Microsoft was the most widely used. The first

version, PC-DOS 1.0, was released in August, 1981. It supported up to 256 kB of RAM and two 160 kB 5.25" single sided floppy disks.

In May 1982, PC-DOS 1.1 added support for 320 kB double-sided floppy disks. PC-DOS 2.0 and MS-DOS 2.0, released in March 1983, were the first versions to support the PC/XT and fixed disk drives. Floppy disk capacity was increased to 180 kB (single sided) and 360 kB (double sided) by using nine sectors per track instead of eight.

Important Terms That You Are Using with DOS

Program: It is a set of instruction written in computer languages. These instructions are stopred in filers ad tells your computer to perform a task.

File: A file is a collection of related information. File on disk can also contaion letters, memos, data, program etc. there are three types of files:

Text Files
Command Files
Application Program Files

- **Text files** are data files that contain characters, numbers and symbols.
- A set of DOS commands put together to perform a specific task which can laterbe stored in a text file called **command files.**
- An A**pplication program files** such as word processor, it stores our work, such as documents in data files,

Directory: A directory contain an entry for each file that gives the file name, extension,size, date & time it was created or last modified.

File Name: A file name consist of two parts i.e. the name of the file & its extension. A file name can be up to eight characters long and can be typed into upper case or lower case letter. DOS automatically converts the lower case letter into upper case letter.

Filename.Extension

3.6.1 Types Of Command

In MS-DOS all the command are categorized into two main category. They are:

Internal Command
External Command

Internal Command

These are the commands which resides in a portion of the computer memoryand are loaded along with the operating system into the memory. These files are always available for execution.

The internal commands are part of DOS' COMMAND.COM file. If you delete COMMAND.COM, you can no longer command DOS to do anything at the command line. Internal command is a command that has the capability to run within DOS at all times, such as Dir, Copy, Del, Ren, Type and Cls. The other kind of commands DOS can execute are external commands. Each external command, such as Format, Xcopy and Backup, resides in its own file in the DOS directory. If any of those files are deleted accidentally or purposefully, you can still execute all of DOS' internal commands and any external commands that are still left on disk.

External Command

These are the commands which have to be loaded from the disk into the memory of the computer before we want to execute them. A separate utility program that comes with DOS, such as Format, Diskcopy, XCopy, Tree, Backup and Restore, but is not resident within DOS, such as Copy and Dir. The directory that contains these programs should be on the path so that you can run them no matter which directory you're in.

Internal DOS Command

1. MD and MKDIR

This command is used to Make a Directory. You type MD followed by a file name. You can nest the directory i.e. Make a directory within a directory up to 16 directories on most DOS only systems. Some will let you nest up to 32. Syntax for this is:

C:\> md directory-name

For Eg:

C:\> md windows

2. CD and CHDIR

This command lets you change directories. The syntax for this is:

C:\> cd directory-name

For Eg:

Lets say you are at root (Just a "C:\>" prompt) and you want to get into a directory named WINDOWS. Type the following:

C:\>cd windows
You will get a prompt like this:
C:\WINDOWS>

You need to get back to your root directory, and quick! Type "CD.." and you will be magically transported back to root.

3. COPY

This command is used to copy a file from one place to another. The syntax for this command is:

Copy Source-filename Destination-filename

For Eg:

C:\copy mystuff.doc A:

This example will copy "mystuff.doc" to your A: drive.

C:\copy mystuff.doc A:\stuff.doc

This copies "mystuff.doc" to the A: drive and renames it to "stuff.doc".

You want to combine two plain text files into one big happy file. This will accomplish that:

C:\copy mystuff.doc+herstuff.txt C:\house\ourstuff.yea

Certain options that are used in the copy command are:

/A: The switch /A treats a file like an ASCII text file. This means that if the file has an End-Of-File character in the middle of it everything up until the [E-O-F] character will be copied. Anything after that will be chopped off.

/B: The switch /B treats the file like a binary file and will copy EVERY THING up the specified file length to the destination. Using the /B switch will ignore all control characters including End Of File markers.

/V: Switch /V makes COPY VERIFY if it correctly made a copy to the destination.

4. TYPE

This command will dump the contents of a text file to your screen. When you use the TYPE command, the file is displayed with limited on-screen formatting. Tabs are expanded and generally displayed as eight spaces wide. If you display files that

contain special (non-text) characters, these characters may have unpredictable effects on your display. Wild card characters (? and *) cannot be used with this command in either the filename or the extension. Syntax for this is:

TYPE [d:][path]filename

For Eg:

C:\>type bill.txt
Bills stuff document.
Bill has 2 things:
1 basket ball
1 used sock
C:\>

This should mostly be used on plain ASCII text files. While you can use TYPE to print out an executable program to the screen, it will be a bunch of junk and you'll get a lot of annoying beeps. TYPE will stop printing a file to your screen when it encounters an End-Of-File character.

5. DEL/ERASE

DEL and ERASE do the exact same thing. This command is used to delete a file. We can also erase one or more files at a time using the wild cards. But one has to be very careful while using wild cards because it will erase all the files mentioned in the specification.

Syntax:

DEL[d:][path]filename.ext

For Eg:

C:\> abc.txt
DOS immediately replies with
ARE YOU SURE (Y/N)

6. CLS

This command clears the screen. It also places your cursor at the top left-hand corner of your screen.

Syntax:

CLS

7. VOL

VOL will tell you the volume label of your hard drive or floppy disk. On DOS 5 and up it will also give you a serial number too.

Syntax:

Vol[d:]

Eg:

Vol E:

8. Ver

VER will tell you what version of DOS you are using. VER/R will tell you some more information like the revision letter and if DOS is in HIGH memory or not.

Syntax:

VER

9. REN

This command is used to rename a file.

Syntax:

REN[d:][path] oldfile-name newfile-name

For Eg:

REN ABC XYZ

10.RD

RD will Remove a Directory. Use RD followed by the name of the directory you wish to delete. You must empty the directory first or you will just get an error message from DOS. Instead of RD we can also use RMDIR.

Syntax:

RD[path] filename

For Eg:

C:\> RD Windows

11.Date

This command lets you set your systems date.

C:\>date
Current date is Wed 03-11-98
Enter new date (mm-dd-yy):
C:\>

12.Time

This lets you set your systems time.

C:\time
Current time is 1:46:11.30p
Enter new time:
C:\>

Notice that at the end of the second line. That means it's P.M. You MUST put at the end of your new time or else your computer will be set to A.M. You can put an there if you want an A.M. time, or you can just leave it blank.

13.Prompt

This is used to set your command prompt. If you use it by itself your prompt will change to C>. Your prompt looks like this most of the time:

C:\>

You can change it by typing PROMPT followed by any text that you want as your new prompt. There are also some switches you can add to spice up your boring old prompt.

$$ adds a $ to your prompt
$t states the time
$d states the date
$p lists your current directory and drive letter
$v adds your DOS version (Or Windoze 95 version)
$n lists just your current drive
$g the > character
$l the < character
$b the | character
$q the = character

$h a backspace, it deletes the last letter of your prompt
$e the escape character, can be A LOT of fun
$_ does a carriage return after listing your prompt

14.DIR

This command gives a listing of most of the files and directories on a disk. DIR also displays both the volume name and amount of free storage space on the disk (if there are files stored in the current directory).

Syntax:

DIR [d:][path][filename] [/A:(attributes)] [/O:(order)] [/B][/C][/CH][/L][/S][/P] [/W]

/W - gives the directory listing wide across your screen without times, dates, and sizes listed

/P - pauses the output of the DIR command if there are more files than can be listed on your screen at once.

/B - (Bare format) Displays only file names.

/C - Displays the compression ratio of files compressed using DBLSPACE. This option is available with DOS Version 6.

CH - Displays the compression ratios of files on a DoubleSpace volume.

/L- Information is displayed in lowercase letters.

/S- Displays file entries in the specified directory and all subdirectories located below it hierarchically.

For Eg:

Dir D:

15.Path

This command is used to set the path for search. When you type the name of a program, like FORMAT, DOS will look through it's list of INTERNAL COMMANDs. Since FORMAT isn't an internal command DOS will then look through your current directory for that programs name. If it doesn't find FORMAT in your current directory it will look through all of the directories listed in your PATH environment variable. You can see what your current PATH is by just typing PATH.

```
C:\>path
PATH=C:\DOS;C:\WINDOWS;C:\UTIL;D:\GAMES;

C:\>
```

To change your path just use PATH followed by the new PATH directories. Notice the semi-colon between directories, this separates entries. The equal sign is optional when you are setting the PATH statement. Be careful, when you set the PATH you will replace the previous one.

16. Copy Con

This command is used to create files and saving them to the disk . once you give the command then you can type the contents of the file, once finished press CTRL+Z and then press enter.

Syntax:

COPYCON [d:] [path] filename

For Eg:

COPYCON D1.TXT

HELLO! HOW ARE YOU.

17.MOVE

This command is used to move the files from one directory to another. It is also used to rename a directory. This is done by specifying the old directory named as source and target would be new directory name.

Syntax:

MOVE [d:] [path] DIRECTORY1 DIRECTORY2.

For Eg:

MOVE c:\abc c:\xyz

External Dos Command

1. BACKUP

It is used for backing up one or more files from a hard disk on to a floppy disk. To backup the files in all sub directories under the path specified ‘/s’ has to be used.

Syntax:

BACKUP d: [path] filename] d: [/s]

For Eg:

BACKUP C:\abc a:

2. CHKDSK

This command is used for checking the disk and reporting back to the status of the system. It checks the memory and gives the detailed information about the memory. It has following switches:

/F automatically fixes any errors found.

/V shows every file on your disk.

Syntax:

CHKDSK [d:] [filename.txt]

For Eg:

CHKDSK

CHKDSK D:

3. DISKCOPY

This is used to diskette into another.this make an exact copy of the diskette in the first drive on the diskette into the another drive.

Syntax:

DISKCOPY [d:] [d:]

For Example

DISKCOPY d: e:

4. DISKCOMP

This command is used to compare the contents of two diskettes. And here are the two switches you can use with it:

/1 only compares the first side of the disks.

/8 only checks the first 8 sectors of each track.

It compares two diskettes to see if they are identical. It can't be used on hard drives. DISKCOMP will ask you to insert the first diskette into drive A:, then it will scan this disk. It will then ask for the second disk and scan it. If they are identical it will tell you, if they are not it will also tell you.

Syntax

DISKCOMP [d:] [d:]

For Eg:

DISKCOMP d: e:

5. Tree

This is used to display all the directories and also the files if specified. If we want display all the files use '/f'.

Syntax:

TREE [d:] [/f]

For Eg:

TREE b:/F

6. FORMAT

FORMAT is used for- you guessed it- formatting disks. This is necessary so that DOS knows where to put data on a disk. FORMAT writes over every available sector on the disk, putting "place holders" where every bit can go. It also sets up the boot sector, root directory, and FAT. FORMAT also detects bad sectors on your disk and marks them out so DOS wont try to use them. FORMAT gives you a list of all it did after completion.

/S this copies system files to you disk after it's formatted

/Q does a quick format- basically it just overwrites the FAT and root directory, does not check for bad sectors or overwrite the data section of the disk **For Eg:**

Format c:

3.7 Booting Process:

The loading of an operating system into memory, which is usually from a hard disk, although ocassionally from a floppy disk is known as booting. This is an automatic procedure begun when you first turn on the computer or reset your computer. The process is normally strted by a small program known as bootstrap loader. This program is present in the ROM. When you start the computer the instruction contained in the ROM begin executing , first running a series of self test which is known as POST (Power On Self Test), to check wether all devices must be correctly connected to the computer system and are in working order or not. Then locating and loading the operating system , and finally passing the control of the computer over to that operating. When the operating system is fully loaded into the memory, the computer is ready to accept users command. There are two types of booting:

1. **Cold Boot:** The computer starts any process when you turn on the computer. A cold boot might be needed if a program crashes in such a way that you cannot continue.
2. **Warm Boot:** A reboot performed after the operating system has started running.

DOS BOOTING PROCESS

The system boot sequence is the series of steps that the system performs when it is turned on (or rebooted with the reset switch, for example). This always starts with the special boot program software that is in the system BIOS ROM. This program runs the *Power On Self Test (POST)*which checks BIOS,CPU, RAM, Video, Keyboard, drives, etc. Once POST is completed, the next step is to load the *Operating System (OS)*. The steps are described in detail here:

1. BIOS locates the Master Boot Record (MBR) on the hard drive.
2. The master boot code examining the master partition table. It first must determine if there is an extended DOS partition, then it must determine if there is a bootable partition specified in the partition table.
3. Extended partition table is loaded in the memory.
4. Next, the small program in the Master Boot Record will attempt to locate an active (bootable) partition in the hard drives partition table.
5. The partition table find the physical location of the logical boot drive and turns to the boot record of that logical drive
6. The boot record (a very short program) loads two hidden files into memory. These files are IO.SYS and MSDOS.SYS
    1. The **IO.SYS** file contains more BIOS software
    2. The **MSDOS.SYS** contains software to manage files, run applications software and interface with hardware
7. Once these two files are loaded, the boot record program is longer needed and turns control over to a file stored on *MSDOS.SYS*

8. This program looks on the hard drive for a file named **CONFIG.SYS**. This is the first OS file that you, the user, can change. This file contains commands that tell DOS how many files it can open at any one time *(FILE=)* and how many file buffers (a temporary holding area for a file) to create **(BUFFERS=)**. It also contains the commands to load device drivers (small programs that tell your computer how to communicate with devices such as printers) **(DRIVERS=)** and other information. Several drivers can be loaded into memory and CONFIG.SYS puts them anywhere it wants unless the program requests a certain memory location.
9. When CONFIG.SYS is done, MSDOS looks for another file called **COMMAND.COM**. This file consists of 3 parts: more code to manage Input/Output (I/O), internal DOS commands such as COPY and DIR, and a short program that looks for **AUTOEXEC.BAT**.
10. **AUTOEXEC.BAT** stands for "automatically executed batch" program. This file holds a list of DOS commands that are automatically executed each time DOS loads. Two of these commands are:
    1. **PROMPT PG** this instructs DOS to display the current directory name and the current drive name as part of the prompt. C:\Windows instead of C:>.
    2. **PATH** Tells DOS where to look for program files. Example: C:Windows\Driver\CacheFonts
    3. **AUTOEXEC.BAT** also loads TSRs (terminate and stay resident programs).
11. The boot process is completed after AUTOEXEC.BAT has finshed executing. At this point, COMMAND.COM is in charge and you have the command Prompt (C:>).