ਪੰਜਾਬੀ ਯੂਨੀਵਰਸਿਟੀ ਪਟਿਆਲਾ

ਵਿਦਿਆ ਵੀਚਾਰੀ ਤਾਂ ਪਰਉਪਕਾਰੀ

PUNJABI UNIVERSITY PATIALA

POST-GRADUATE DIPLOMA IN COMPUTER APPLICATIONS

PAPER : PGDCA-2
OPERATING SYSTEM

SECTION-A

LESSON NO. :

Department of Distance Education
Punjabi University, Patiala

PGDCA **Paper : PGDCA-2**
Operating System

Lesson No. : 1.1 **Author: Arun Bansal**

Operating Systems Concepts

1.1.0 Objective:

In this lesson the students will become familiar about the basic concepts of Operating systems. The basic concepts such as processes, files, system calls, the shell of the operating system, booting process are dicussed in the lesson. A simple structure of operating system is given. The factors that must be considered for measuring system performance. In the end brief history of operating systems and the features of the modern operating system is given.

1.1.1 Introduction

As we have already seen how a computer uses all its components and peripherals to carry on computing and various other operations. These devices, which are used are hardware devices, which usually require instructions to work as they cannot work by themselves. The memory has to be managed, input/output operations have to be done and various other functions have to be done, which are accomplished by giving necessary instructions, which are given in machine understandable form. Each machine is different from other machines in configuration. Therefore, it becomes quite difficult and nearly impossible task for the programmers to write instructions for the above operations. If still programmers manage to do this, they will be wasting their precious time and effort by concentrating more on making the computer work rather than getting their tasks done. Therefore emerged the search for a common program which is capable to do these common tasks thereby leaving only the task of programming for specific tasks, which resulted in the development of operating system, a system software, which will do all the above tasks as well give an environment to work with computer. The operating systems act as an extended machine as it acts in the similar way as hardware, which helps the user to program his task. The operating systems delink user from hardware, they help the programmer to get his tasks done by acting as mediator between the user and the hardware.

Operating system is the software that provides the interface between the user and hardware. It is the most important program that runs on a computer. Every general purpose computer must have an operating system to run the programs. **Operating system (OS) can be defined as "Set of programs that are capable to manage resources as well as the operations of a computer"**. Operating System hides the hardware virtually from the user, it isolates the hardware from the user. Till the functions are performed properly, user is not concerned about the hardware specifications. We can also say that operating system is a system software that manages Computer Hatdware, software resources and provides comman services for computer programs.

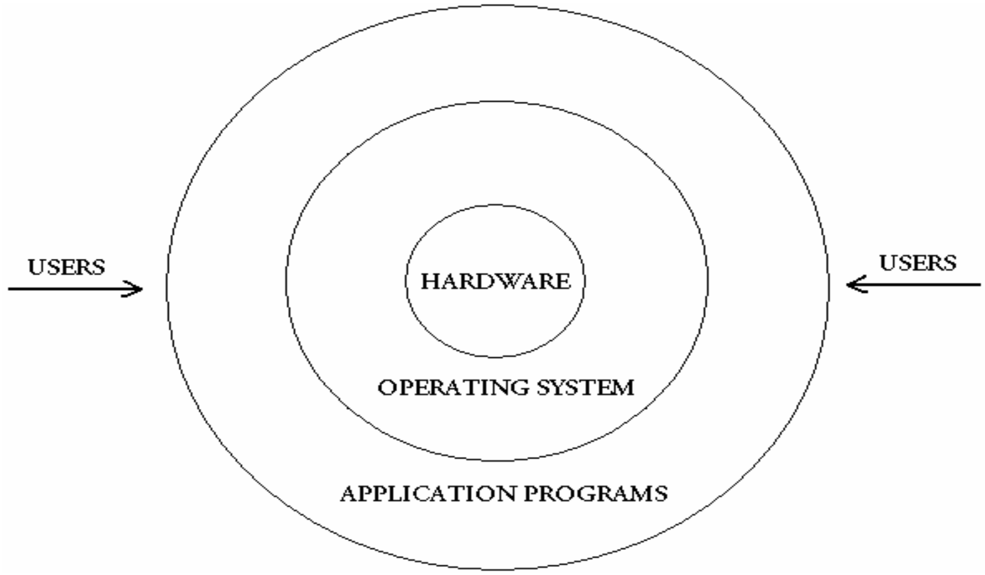


Figure 1.1: The figure depicts how operating system isolates hardware from the users.

Operating systems have been on the evolution spree through the years. This evolution was for the development of more and more better operating systems with better features.

The basic functions which are carried by any operating system are:

i. Assigning processors for performing tasks.
ii. Allocating memory, managing memory and other storage areas.
iii. Command interpretation.
iv. Handling job to job transitions.
v. Maintaining internal clock.
vi. Scheduling of various jobs/tasks.
vii. Establishing and enforcing priorities for different jobs.
viii. Co-ordinating and managing peripheral devices.

ix. Input output management.

x. Establishing data security and integrity.

1.1.2 Operating System Concepts

User programs interact with operating system using set of extended instructions. These instructions are called “System Calls”. These system calls are used to create, delete and use various software objects that are managed by the operating systems. The most commonly used and important system calls are processes and files. The following features are common in any operating system.

1.1.2.1 Processes

Any running program is known as process. It is one of the most vital concepts of an operating system. **We can define a process as a program in execution**. A process is composed of an executable program, the data used by the program and the stack, the program counter, registers, stack pointer and any other information which is required for the execution of the program.

To understand concept of process easily, let us discuss about the time sharing operating systems, where the CPU time is shared by many processes, where a process after being given its share of time is suspended and another process gets the time share. Once its time share is completed, it comes back to the suspended process, which as a matter of common sense, should start from the position where it was suspended and this operation cycle goes on. To keep track of ongoing processes most of the operating systems maintain a process table, which usually is an array where the information is stored.

The basic system calls for process management are those which deal with the creation and termination of process. For example consider a process like shell or command interpreter, which reads the given commands from a terminal. Once a user gives a command, shell has to create a new process which is going to execute the given command and once the execution is complete, the process has to be terminated There are many other processes like memory management, child process management, message management etc. which makes an operating system efficient and workable.

1.1.2.2 Files

These are another broad category of system calls which are used to work with file system. As we have already seen that one of the basic functions of an operating system is to hide the hardware from the user, so that he is not bothered about any interaction with hardware. As these interactions are taken care by the operating system, therefore when a user works with the file system, calls are needed for the creation of files, maintenance of files, reading from the files, writing to the files etc., as well as opening of a file before any operation and closing the file after the usage.

The concept of directory, i.e. folder of files, where a user can group his files together, has to be provided for which system calls are required. In a directory, a user can have either files or other directories. This concept of directory is provided so that a user can group his files or directories which are similar in one or the other aspects, which makes his search and access easier.

Both the processes and files follow the hierarchies, which are organized as trees. This is where processes and files are similar to each other. Usually process hierarchies are not very deep, there is usually at most three levels of hierarchies where as files can have hierarchies which are very deep and they can go up to any levels.

1.1.2.3 System Calls

For every task to be done, user has to interact and communicate with the computer via operating system. **All the user programs communicate with the operating system for various services and tasks by making system calls.** Every system call has a library procedure that user programs can call. These procedures put the parameters of the system calls at specified areas, such as registers etc. and issue a TRAP instruction, which is a protected procedure call, to start the operating system. The basic purpose of the library procedure is to hide the TRAP instruction details so that the system calls look like ordinary procedure calls. After the receipt of the procedure call, the operating system checks the parameters for validity and if valid, performs the work requested. After the requested task is finished, the operating system updates the status code in the register, stating whether the given task has been executed successfully or whether it failed and then executes an instruction to return from the TRAP to the library procedure. Later this library procedure returns to the user the function value and other values as parameters. Examples of system calls are read,write, close, wait, ener fork, exit kill etc.

1.1.2.4 The Shell

"A user interacts with the operating system via one or more user applications and often through a special application called a shell or command interpreter." We have already seen that it is the code of the operating system which carries the system calls. Along with operating system other system software like Editors, compilers, Assemblers, Linkers, Command interpreters etc. also come as helping hand and are quite important. The shell is command interpreter of the UNIX operating system. We will look into this concept while discussing UNIX in the following sections. Even though Shell is not a part of the operating system, it uses many operating system features and is one of the best examples of how system calls can be used. Shell is also the primary interface between the users and the operating system, which starts as soon as user logs on to the system.

Operating system is a collection of programs which are used to control the computer system i.e. the peripherals and other utilities. The operating system is partly stored in the secondary memory storage devices like hard disk, magnetic tape etc. and is partly stored permanently in the ROM.

1.1.2.5 The booting Process

Booting process can be taken as the process of starting the computer. This process is important as it is this process which loads the operating system into the computer's internal memory. This is done by executing a program in ROM, this program reads the operating system into the RAM, which is then executed by the computer.

A quick look at the various categories of the program which constitute the operating system, which have their predefined roles to play in the functionality of the operating system, gives the following categories.

i. Boot Strap Loader

This is the program which is usually stored in ROM and is responsible for the start of the computer system. This program reads the portion of the operating system, which is stored from the secondary memory devices into the internal memory, so that they can be executed. This program starts working once the computer is switched on.

ii. Check Programs

These are the programs that are used for testing the operation of various system components. These are also known as Diagnostic tests, as they diagnose any problem associated with the operation of the system components. Some of the common checks performed by these programs are checking the working of disk drives, checking of RAM, checking of Memory, checking the status etc. Some of these programs are stored in the ROM, which are executed automatically when the system is first turned on. For other programs users must load the check programs from the secondary memory and execute them. Thus these programs are used for various checks.

iii. Monitor Program

This is the program which is used to control the activities of the system. This is also called as operating system Executive. When a user requests for execution of programs, this monitor program schedules the execution of programs and assigns various tasks to the hardware devices. When the computer is switched on, this program is read into RAM, so that it takes control of the system. Most of the current day computers allow execution of several programs simultaneously, which is given a shape by monitor programs which are very sophisticated and the operating systems which are capable to do this type of processing are known as multi-tasking operating system.

iv. Basic Input/Output System (BIOS) Programs

These programs are used for controlling the most rudimentary functions of the hardware components and devices of the computer system. For controlling these hardware components, they use low level routines. The following are some of the basic control functions that are done by these BIOS programs, so that the basic input and output operations can be given a proper shape:

a. Reading a key press i.e., a character from input device like keyboard.
b. Writing a character on to the video display unit or other output devices.
c. Checking whether the output devices are busy.
d. Retrieval of data from disks.
e. Update/Erase of the data.

Apart from above, BIOS perform various other functions which have a direct link to the input and output devices. The user or the programmer need not worry about these functions as they are usually stored in ROM, thus available forever to the user.

v. Support or Utility Programs

The operating system apart from carrying basic functions, should also help user by giving him support by providing him with various housekeeping utility programs or support programs, so that the task of the user becomes easy. Some of the functions which are controlled by these utility/support functions are as following:

a. Display of the contents of a disk.
b. Formatting a disk so that it can store data.
c. Copying data from one disk to another.
d. Creating a backup for the hard disk contents in order to tackle any untoward damages to the hard disk.
e. If hard disk crashes, it has to be restored using backup copy.
f. Checking the memory space available on the disk.
g. Password authentication check, so as to give security to the system etc.

vi. File Maintenance Programs

User has to store data, programs and other information in files and all the operations on the data are done usually through files. Therefore, these files have to be maintained and managed so that they are consistent. The file maintenance programs help users to maintain files by providing them facility to create, read and write files. These operations are done with the help of routines, which have to be called from the user programs.

1.1.3 Definition and need

It is the software that controls the execution of the computer programme and may proivde various services or an Operating System (OS) is an integrated set of programs that controls the resources(CPU, memory, Input and Output devices , files etc.) of a computer system and provides its users with an interface or virtual machine that is more convenient to use than the bare machine (machine without an operating system). Since it manages all the resources of the computer system, it is also known as an overall resource manager. Operating system is considered as a vital component of a computer system, without which we consider that computer to be useless. A computer without an operating system is like a car without petrol.

OS makes the hardware hospitable to use. Bare hardware is extremely inhospitable. To use it efficiently, in absence of an operating system, the programmer should have intimate knowledge about the hardware complexities. Whereas OS takes care of all such complexities and do the needful to help programmer/users.

OS is a large group of different types of software without which a computer cannot perform any operation. The operating system ensures that all the hardware components of a computer work in coordination with each other for the proper functioning of the computer. It acts as an interface between the users and the hardware of a computer system.

The operating system ensures that each application gets the necessary resources. It also ensures that application continue to run when hardware upgrades and updates occurs.

Window 95, Windows 98, Linux, UNIX, DOS, OS/2, Macintosh are good examples of operating systems. These operating systems can run on hardware provided from thousand of vendors. They can accommodate thousand of different printers, disk drives and special peripherals in any possible combination.

1.1.4 System Structure

A system as large and complex as a modern operating system must be engineered carefully if it is to function properly and to be modified easily. A common approach is to partition the task into small components, rather than have one monolithic system. Each of these modules should be a well-defined portion of the system, with carefully defined inputs, outputs, and function.

1.1.4.1 Simple Structure

There are numerous commercial systems that do not have a well-defined structure. Frequently, such operating systems started as small, simple and limited systems, and then grew beyond their original scope. MS-DOS is an example of such a system. It was originally designed and implemented by a few people who had no idea that it would become so popular. It was written to provide the most functionality in the least space, because of the limited hardware on which it ran so it was not divided into modules carefully. Figure 1.2 shows its structure.

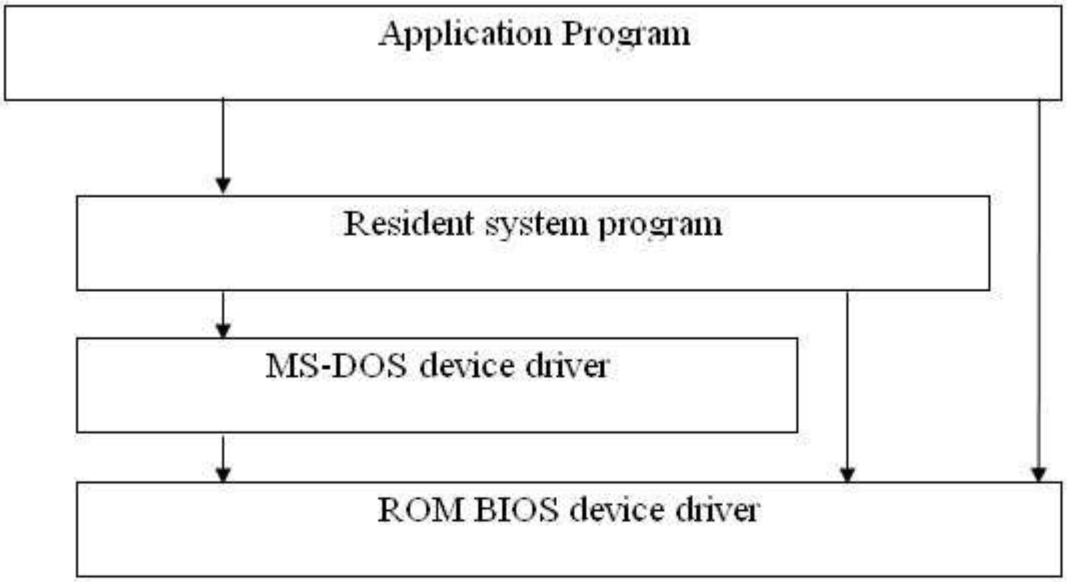


Figure 1.2 depicting the simple structure of MSDOS layer structure

In MS-DOS, the interfaces and levels of functionality are not well separated. For instance, application programs are able to access the basic I/O routines to write directly to the display and disk drives. Such freedom leaves MS-DOS vulnerable to errant (or malicious) programs, causing entire system crashes when user programs fail. Of course, MS-DOS was also limited by the hardware of its era. Because the Intel 8088 for which it was written provides no dual mode and no hardware protection, the designers of MS-DOS had no choice but to leave the base hardware accessible.

1.1.4.2 Layered Approach

These new UNIX versions are designed to use more advanced hardware. Given proper hardware support, operating systems may be broken into smaller, more appropriate pieces than those allowed by the original MS-DOS or UNIX. The operating system can then retain much greater control over the computer and the applications that make use of that computer. Implementers have more freedom to make changes to the inner workings of the system. Familiar techniques are used to aid in the creation of modular operating systems. Under the top-down approach, the overall functionality and features can be determined and separated into components. Information hiding is also important, leaving programmers free to implement the low-level routines as they see fit, provided that the external interface of the routine stays unchanged and the routine itself performs the advertised task.

The modularization of a system can be done in many ways; the most appealing is the layered approach, which consists of breaking the operating system into a number of layers {levels), each built on top of lower layers. The bottom layer (layer 0) is the hardware; the highest (layer N) is the user interface.

An operating-system layer is an implementation of an abstract object that is the encapsulation of data and operations that can manipulate those data. A typical operating-system layer - say layer M - is depicted in Figure 1.3 It consists of some data structures and a set of routines that can be invoked by higher-level layers. Layer M, in return, can invoke operations on lower-level layers.

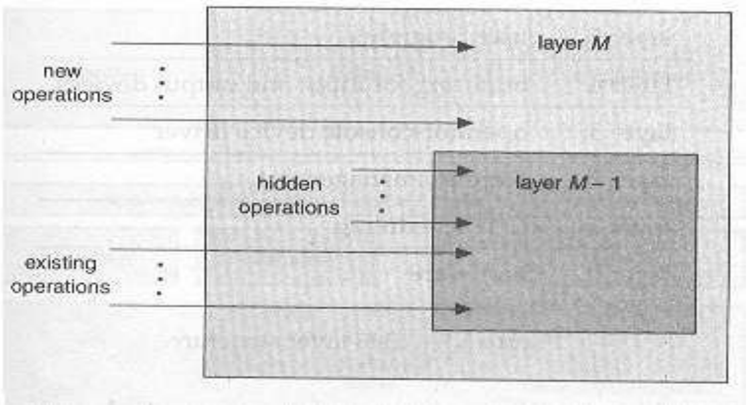


Figure 1.3 depicting an operating-system layer

The main advantage of the layered approach is ***modularity***. The layers are selected such that each uses functions (operations) and services of only lower level layers. This approach simplifies debugging and system verification. The first layer can be debugged without any concern for the rest of the system, because, by definition, it uses only the basic hardware (which is assumed correct) to implement its functions. Once the first layer is debugged, its correct functioning can be assumed while the second layer is worked on and so on. If an error is found during the debugging of a particular layer, we know that the error must be on that layer, because the layers below it are already debugged. Thus, the design and implementation of the system is simplified when the system is broken down into layers.

Each layer is implemented using only those operations provided by lower level layers. A layer does not need to know how these operations are implemented; it needs to know only what these operations do. Hence, each layer hides the existence of certain data structures, operations and hardware from higher-level layers.

The major difficulty with the layered approach involves the appropriate definition of the various layers. Because a layer can use only those layers that are at a lower level, careful planning is necessary.

1.1.5 Operating System Services

Desired functionalities of OS depend on outside factors like users & application's "Expectations" and ``Technology changes" in Computer Architecture (hardware). *OS must adapt:*

- Change abstractions provided to users
- Change algorithms to change these abstractions
- Change low-level implementation to deal with hardware

The operating system provides certain services to programs and to the users of the program. The specific services provided differ from system to system, but common functions performed by the operating systems are :

i) Process Management:

Any program in execution is called a process. The operating system of a computer manages the creation and deletion of processes, scheduling of various system resources to the different processes requesting them and provides mechanism for synchronization and communication among processes.

ii) Memory Management :

The operating system manages the main memory. It takes care of the allocation and deallocation of memory space to various programs in need. It keeps track of free memory and allocates the memory if required and deallocates memory if not required.

iii) File Management

The operating system takes care of file related activities such as organization, storing, retrieval, naming, sharing and protection of files. It allows programmers to use set of operations that characterize the file abstraction and free the programmers from the concerns about the details of space allocation and utilization of secondary storage device.

iv) Device Management

The operating system manages all the I/O devices. It keeps track of requests of processes from I/O device, issues commands to the I/O devices and ensures that correct data transmission to and from I/O devices takes place. It also provides an interface between the devices and the rest of the system.

v) *Security* & *Protection*

Operating system always protects the resources and information of a computer system against destruction and unauthorized access.

vi) Job Sequencing

Helps in Automatic job sequencing. It is the duty of the operating system to ensure that jobs entered in the computer are executed automatically in the provided sequence.

vii) Error Handling

Handles errors when they occur and correction routes to remove them.

viii) Monitoring:

System software keeps track of everything that is going on inside the computer. Some people have the impression that everything they do on the computer system, especially on larger systems is anonymous. No one really knows you were using the computer on Saturday morning because no one was there to see it. Wrong! The system software program knows that you accessed the computer, for how long and what you did. This part of the program monitors, and if programmed correctly, records everything that was done on the system.

ix) Others

a) Helps user to interact with machine. The various systems program written in assembly language help to interact with different components of the machine.

b) It acts as an interface between the computer system and the user. It facilitates easy Communication between the computer system and the user.

c) It maintains internal time clock and log of system usage for all the users

Hence, the operating system controls the overall functioning of the computer system.

1.1.6 Measuring system performance:

The efficiency of an operating system and the overall performance of a computer system are usually measured in terms of the following:

a) **CPU utilization** : We want to keep the CPU as busy as possible, since CPU is the costliest component in the computer system. CPU utilization may range from 40% to 90%.

b) **Turnaround Time** – As far as simple job is concerned, the important factor is turnaround time. **Turnaround time is the interval from instant of submission to the instant of completion**. Turnaround time is the sum of following five components.

- The time it spends waiting for entry into the system.
- Total time it spends in the ready queue.
- Total time it spends in the device queue.
- Actual execution time of the CPU.
- Total time it spends in doing actual I/O.

c) **Throughput – Throughput is a measure of the work done by the CPU. It is expressed in terms of number of jobs done in a given unit of tim**e. It is measured as the number of processes that are completed by the system per unit time. It is important to note that value of throughput does not depend only on the capability of the system, but also on the nature of the jobs being processed by the system. For long processes, throughput may be one process/ hour and for short processes throughput may be 100 processes per hour.

d) **Response time :** Turnaround time is usually not a suitable measure for the interactive system, because in an interactive system, a process can produce some output early during its execution and can continue execution while previous results are being output to the user. Hence, another measure used in case of interactive system is response time, which is the interval from the time of submission of a job to the system for processing to the time of first response for the job is produced by the system.

For a system performance to be higher we prefer maximum CPU utilization, maximum throughput, minimum turnaround time and minimum response time.

Different operating systems like batch processing operating system, Multiprogramming operating system, Time sharing operating system, Multi-tasking operating system, Distributed operating systems, Networking operating systems and Real time operating systems etc. have been developed. All these will be discussed in 2nd chapter.

1.1.7 History of the Operating Systems (Early Systems)

Historically operating systems have been tightly related to the computer architecture, it is good idea to study the history of operating systems from the architecture of the computers on which they run.

Operating systems have evolved through a number of distinct phases or generations which corresponds roughly to the decades.

a) **The 1940's First Generations**

The earliest electronic digital computers had no operating systems. Machines of the time were so primitive that programs were often entered one bit at time on rows of mechanical switches (plug boards). Programming languages were unknown (not even assembly languages). Operating systems were unheard of .

b) The 1950's - Second Generation

By the early 1950's, the routine had improved somewhat with the introduction of punch cards. The General Motors Research Laboratories implemented the first operating systems in early 1950's for their IBM 701. The system of the 50's generally ran one job at a time. These were called single-stream batch processing systems because programs and data were submitted in groups or batches.

c) The 1960's - Third Generation

The systems of the 1960's were also batch processing systems, but they were able to take better advantage of the computer's resources by running several jobs at once. So operating systems designers developed the concept of multiprogramming in which several jobs are in main memory at once; a processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use.

For example, on the system with no multiprogramming, when the current job paused to wait for other I/O operation to complete, the CPU simply sat idle until the I/O finished. The solution for this problem that evolved was to partition memory into several pieces with a different job in each partition. While one job was waiting for I/O to complete, another job could be using the CPU.

Another major feature in third-generation operating system was the technique called **spooling** (Simultaneous Peripheral Operations On-line). In spooling, a high-speed device like a disk interposed between a running program and a low-speed device involved with the program in input/output. Instead of writing directly to a printer, for example, outputs are written to the disk. Programs can run to completion faster and other programs can be initiated sooner when the printer becomes available, the outputs may be printed.

Note that spooling technique is much like thread being spun to a spool so that it may be later be unwound as needed.

Another feature present in this generation was time-sharing technique, a variant of multiprogramming technique, in which each user has an on-line (i.e., directly connected) terminal. Because the user is present and interacting with the computer, the computer system must respond quickly to user requests, otherwise user productivity could suffer. Timesharing systems were developed to multiprogram large number of simultaneous interactive users.

d) Fourth Generation

With the development of LSI (Large Scale Integration) circuits, chips, operating system entered in the system entered in the personal computer and the workstation age. Microprocessor technology evolved to the point that it became possible to build desktop computers as powerful as the mainframes of the 1970s. Two operating systems have dominated the personal computer scene: MS-DOS, written by Microsoft, Inc. for the IBM PC and other machines using the Intel 8088 CPU and its successors, and UNIX, which is dominant on the large personal computers using the Motorola 6899 CPU family.

1.1.8 Features of the modern operating systems and future

- Hardware becomes cheaper and users more sophisticated. People need to share data and information with other people. Computers become more information transfer, manipulation and storage devices rather than machines that perform arithmetic operations. Networking becomes very important and as sharing becomes an important part of the experience so does security. Operating systems become more sophisticated. Start putting back features present in the old time sharing systems (OS/2, Windows NT, even Unix).
- Rise of network. Internet is a huge popular phenomenon and drives new ways of thinking about computing. Operating system is no longer interface to the lower level machine - people structure systems to contain layers of middleware. So, a Java API or something similar may be the primary thing people need, not a set of system calls. In fact, what the operating system is may become irrelevant as long as it supports the right set of middleware.
- Network computer. Concept of a box that gets all of its resources over the network. No local file system, just network interfaces to acquire all outside data. So have a slimmer version of OS.
- In the future, computers will become physically small and portable. Operating systems will have to deal with issues like disconnected operation and mobility. People will also start using information with a psuedo-real time component like voice and video. Operating systems will have to adjust to deliver acceptable performance for these new forms of data.

1.1.9 Summary

An operating system is a program that acts as an intermediary between a user of a computer and the computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs. The primary goal of an operating system is thus to make the computer system convenient to use and a secondary goal is to use the computer hardware in an efficient manner. We can well understand that even without an operating system our computer can work but it will really be a very-very tedious task for the programmers and users to work on a machine without an operating system. The major functions of operating systems are memory management, processor management, device management and file management. Along with these tasks the other functions provided by operating systems are job sequencing, error handling and providing security and protection from un-authorized users.

A system as large and complex as a modern operating system must be engineered carefully if it is to function properly and to be modified easily. A common approach is to partition the task into small components, rather than have one monolithic system. Each of these modules should be a well-defined portion of the system, with carefully defined inputs, outputs and function. There are numerous commercial systems that do not have a well-defined structure. Frequently, such operating systems started as small, simple, and limited systems and then grew beyond their original scope. MS-DOS is an example of such a system. Given proper hardware support, operating systems may be broken into smaller, more appropriate pieces than those allowed by the original MS-DOS or UNIX. The operating system can then retain much greater control over the computer and the applications that make use of that computer.

The efficiency of an operating system and the overall performance of a computer system are usually measured in terms of the factors such as CPU utilization, turn around time, throughput and response time. For a system performance to be higher we prefer maximum CPU utilization, maximum throughput, minimum turnaround time and minimum response time.

Operating systems have evolved through a number of distinct phases or generations, which corresponds roughly to the decades. The 1940's First Generations computers Operating systems were not heard of. The 1950's - Second Generation Computers have the batch processing operating systems. The 1960's - Third Generation have the facility of multiprogramming and spooling (simultaneous peripheral operations on line). In 1980's Fourth Generation computers personnel operating systems were developed. In the future, computers will become physically small and portable. Operating systems will have to deal with issues like disconnected operation and mobility. Operating systems will have to adjust to deliver acceptable performance for these new forms of data.

1.1.10 Keywords

Process	:	A program in execution
File	:	Collection of records is called a file.
System calls	**:**	Provides an interface between a process and the operating system.
Shell	:	The shell is command interpreter of an operating system.
RAM	:	Random Access Memory (Also known as main memory of the computer.

ROM	:	Read Only Memory
Bare Machine	:	A machine without an operating sytem
I/O	:	Input / Output

1.1.11 Self-Check Exercise

1. What do mean by Operating System?
2. What do you mean by System Call?
3. What is Basic Input/Output System (BIOS)?
4. What do you mean by booting?
5. Discuss various functions performed by an operating system.
6. Discuss the structure of an operating system in brief.
7. What factors are considered for measuring system performances?
8. Discuss in brief about various generations of operating system.

1.1.12 Suggested Readings

1. "Fundamentals of Information Technology", by Dr. Hardeep Singh, S.k. Kakar, by Lakahanpal Publication.
2. "Operating Systems concepts", by Silverschatz, Galvin by Personed Education.
3. "Computer Fundamentals" , by P.K. Sinha, by BPB publication
4. "Operating Systems", by Gary Nutt, by Pearsoned Education
5. "Operating System Principles" by Hensen and Per Brinch, Prentice Hall.

PGDCA **Paper : PGDCA-2**
Operating System

Lesson No. : 1.2 **Author: Arun Bansal**

Basic Operating System Types

1.2.0 Objective:

In this lesson the students are made familiar with various types of basic operating systems such as Batch processing, multiprogramming, multitasking multiprocessing and time sharing operating systems.

1.2.1 Introduction

An operating system may perform a task serially (sequentially) or concurrently (multiple tasks simultaneously). Thus, all the resources of a computer system may be allocated to multiple programs. So on this basis the operating systems can be categorized as follows.

1.2.2 Batch Processing System

In the early days the most common way of using computer had been operating a program punched into a punch cards, the user did not interact directly with the system, but the job was prepared by the user that contained program data and other information related to job. This job was then submitted by the operator. The key punch operator used to receive jobs from many users. A number of jobs were stacked one after the other depending upon the nature of the job. At that time the operating system was very simple. Its major task was to transfer control automatically from one job to next. The operating system was always (resident) in memory.

At that time the compilers of different languages were available on magnetic tapes, which needs to be physically mounted on the tape reader and the compiler is moved to the memory through card reader. Thereafter the job of those languages can be compiled and executed. If the jobs submitted in the sequence order are not of the same languages, then most of the time is wasted in physical movement of compilers to memory and vice versa. To speed up the process, the jobs which have similar requirements were to be batched together for execution. (i.e. the jobs of the same languages). So in batch processing systems, jobs were typically executed in the following manner.

i) Programmers would prepare their programs and data on deck of cards or paper tapes and submit them at the reception counter of the computer center.

ii) The operators would periodically collect all the submitted programs and would batch them together and then load them all into the input devices of the system at one time.

iii) The operator would then give a command to the system to start executing the jobs.

iv) The jobs were then automatically loaded from the input device and executed by the system one by one without any operator intervention. That is, the system would read the first job from the input device, execute it, print out its result on the printer and then repeat these steps for each subsequent job until all jobs in the submitted batch of jobs were over.

v) When all the jobs in the submitted batch were processed, the operator would separate and keep the printed output of each job at he reception counter for the programmers to collect them latter.

The batch processing is one of the oldest methods of running programs. This method reduced the idle time of a computer system because transaction of one program or job to another does not required operator's intervention. The main motive of using bater processing system is to decrease the setup time while submiting the similer jobs to CPU. This method of processing was mostly used in payroll applications or preparation of customer statements because it was not necessary to update records on daily basis.

Advantages of Batch System:

- Reduces idle time of a computer system because transaction from one job to another does not require operator Intervention.
- Increased performance since it was possible for the job to start as soon as previous job finished.

  One of the disadvantages of batch processing is that the job is prepared and submitted but the results are received with delay. Also in these systems the CPU remains idle for most of the times. The reason behind this thing is that CPU speed is very fast as compared to electromechanical I/O devices. A CPU can execute thousand of instructions per second whereas a card reader may read 1200 cards per minute.

Disadvantages of Batch System:

- Turnaround time can be large from user standpoint. Large time is required to accumulate data in batches.
- More difficult to debug program.
- Due to lack of protection scheme, one batch job can affect pending jobs.
- A job could enter into infinite loop.

- It is difficult to provide desired priority scheduling.

1.2.3 Single User Operating System

This operating system is made for systems, which are called single user machines i.e., one person can use the machine at a time and operating system is available to the single user at a time. Here only one program can be executed at a time. Next program is executed only after program execution stops. This operating system is very popular because of its smaller size ease of use and lesser cost. In this operating system no sharing is allowed i.e. different users working on different machines cannot share each other's information. It is supposed to be more secure and personnel in use. This operating system has gained a lot of popularity particularly among Micro category i.e. PC users.

Examples of this Operating System are Disk Operating System called DOS, which is provided by many vendors now as MS-DOS, PCDOS, NOVELL DOS etc. DOS will be discussed in detail in the later lessons. Another single user operating system is WINDOWS.

Though now Windows-95 has come out with network features and can be used as multi-user Operating system.

Advantages of single user

- It is very simple in use.
- More secure and personnel in use.
- Very small in size.
- Requires very less main memory.

Disadvantages of Single user operating system

- CPU utilization is very low so takes much longer to complete a given task.
- Throughput of the system is very low.

1.2.4 Multi-User Operating System

This operating system is made for machines which are more commonly called sharable machines, i.e. **data from one machine can be shared by other users at other machines and vice versa**. In this operating system, system security and integrity becomes very important and it is responsibility of the operating system to carry it out. This operating system will be more complex in nature and bigger in size as compared to single user Operating system. It is also more costly as compared with Single user Operating System. Earlier it did not gain much popularity but now because of its flexibility and multiple functions it has gained a lot of popularity and many organizations are switching to multi-user Operating System.

Its examples are UNIX, ZENIX, SOLARIS, Redhat Linux, Window NT, Window 2000, OS/2 Mac OS X, Windows XP Solaris,FreeBSD, NetBSD etc.

1.2.5 Multiprogramming Operating System:

As far as batch processing was concerned a number of programs were loaded in a sequence in the main memory and program remains the occupant of main memory until the execution of program gets completed. This leads to under utilization of system-resources such as CPU and main memory.

In order to get rid of this problem the concept of multiprogramming was introduced.

Multiprogramming refers to execution of two or more different and independent programs by the same computer. In this two or more programs reside in main memory and are executed concurrently. The basic principle of the multiprogramming is based on the concept, which is read as " the overall speed of the system is always governed by the slowest devices in the system". The speed of the input/output devices is very-2 slow as compared to the speed of the processor.

Normally any job has to handle various resources like input/output devices, memory and processor. It was observed that a job does not need the CPU for entire duration of the its processing. This is because in addition to doing computation (for which CPU is needed), a job often needs to perform I/O operation (such as read/write operation) during the course of processing. In fact, depending upon the CPU utilization during the course of processing, jobs are badly classified into the following types.

CPU-bound jobs : These jobs mostly perform numerical calculation with little I/O operation. They are so called because they heavily utilize the CPU during the course of their processing. Programs used for scientific and engineering computations usually falls in this category.

I/O-bound jobs: These jobs perform very little computation and most of the time they perform I/O operations. Hence, their CPU utilization is very low. Programs used for commercial data processing applications usually fall in this category of jobs.

In case of multiprogramming more than one job are being loaded into the main memory of the system. These jobs must be intermixed i.e. few jobs should be CPU bound and a few jobs should be I/O bound. The number of jobs loaded into the main memory is called **degree of multiprogramming**.

In case of multiprogramming more than one jobs could utilize the CPU at any moment As more number of jobs competed for system resources, it resulted in better resource utilization. If one of the programs was busy in I/O operations the operating system may switch over to the next Program. When this program needed to wait for an I/O operation again the operating system switched over to the next program. Multiprogramming supported multiple interactive users and increased resource utilization. So more is the degree of multiprogramming more is the resource utilization.

Here Job A and Job B are executing simultaneously and Job C is waiting for its turn. The Job A is busy in performing Input/Output operating on the hard disk and the other Job B is utilizing CPU to execute its instructions. As soon as Job B is completed or Job B requires to do I/O operations, the CPU will start executing program C. Similarly after Job A completes I/O operation it will wait for CPU for its turn. Thus in multiprogramming CPU is almost busy and has very little idle time.

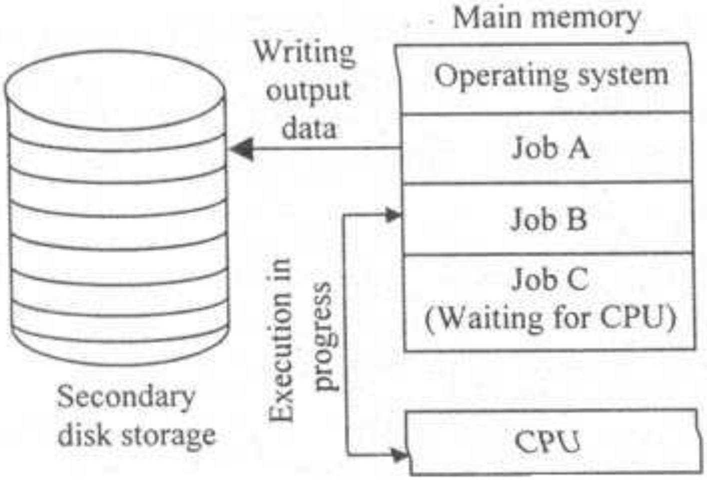


Figure 2.1 : A scenario of jobs in the multiprogramming system

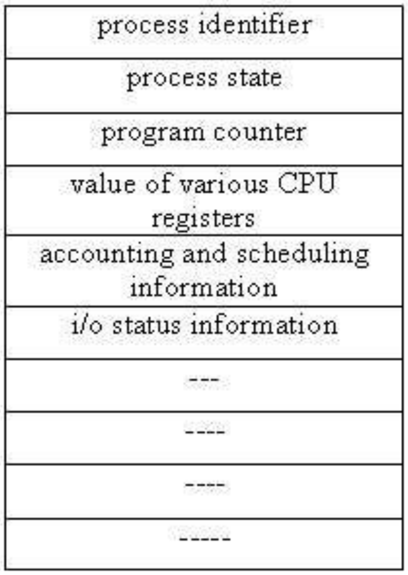
Requirements of Multiprogramming Systems

Multiprogramming systems have better throughput than uni-programming systems because the CPU idle time is drastically reduced. However, multiprogramming systems are sophisticated because they require the following additional hardware and software features:

i) **Large Memory:** For multiprogramming to work satisfactorily, large main memory is required to accommodate a good number of user programs along with the operating system.

ii) **Memory protection :** Computers designed for multiprogramming must provide some type of memory protection mechanism to prevent a job in one memory partition from changing information or instruction of a job in another memory partition. For example, in Figure 2.1 we would not want job A to inadvertently destroy something in the completely independent job B or job C. In a multiprogramming system, this is achieved by the memory protection feature, a combination of hardware and software, which prevents one job from addressing beyond the limits of its own allocated memory area.

iii) **Job status preservation:** In multiprogramming, when a running job is blocked for I/O processing, the CPU is taken away from this job and given to another job that is ready for execution. Later, the former job will be allocated the CPU to continue its execution. Notice that this requires preserving of the job's complete status information when the CPU is taken away from it and restoring this information back, before the CPU is given back to it again. To enable this, the operating system maintains a process control block (PCB) for each loaded process. A typical process control block is shown in Figure 2.2 With this arrangement, before taking away the CPU from a running process, its status is preserved in its PCB, and before the process resumes execution when the CPU is given back to it later, its status is restored back from its PCB. Hence, the process can continue execution without any problem.



process identifier
process state
program counter
value of various CPU registers
accounting and scheduling information
i/o status information

Figure 2.2 : Depicting a typical Process Control Block (PCB)

iv) **Proper job mix:** A proper job mix of I/O-bound and a CPU-bound job is required to effectively overlap the operations of the CPU and I/O devices. If all the loaded jobs need I/O at the same time, the CPU will again be idle. Hence, the main memory should contain some CPU-bound and some I/O-bound jobs so that at least job is always ready to utilize the CPU.

v) **CPU scheduling :** In a multiprogramming system, often there will be situations in which two or more jobs will be in the ready state, waiting for CPU to be allocated for execution. When more than one process is in the ready state and the CPU becomes free, the operating system must decide which of the ready jobs should be allocated the CPU for execution. The part of the operating system concerned with this decision is called the CPU scheduler and the algorithm it uses is called the CPU scheduling algorithm.

Advantages of Multiprogramming:

- Throughput of system is increased as idle time of CPU is utilized.
- Response time can also be reduced by recognizing the priority of job

1.2.6 Multitasking Operating System:

It is the ability to do more than one thing at any given time. **While referring to computers multitasking is having more than one application open at a time** for example at some time you might be downloading some thing from Internet, as well as you are writing mail to your friend and listening music also. In real life sometimes we talk on mobile while driving means doing more than one task exactly at the same time is called multitasking.

To multitask efficiently a computer needs a good amount of processing power and more importantly a memory.

The term multitasking and multiprocessing are often used interchangeably, although multiprocessing some times implies that more than one CPU is involved. In multitasking, only one CPU is involved, but it switches from one program to another so quickly that it gives the appearance of executing all the programs at the same time.

There are two basis types of multitasking:

1. Preemptive multitasking
2. Co-operative or non-preemptive multitasking.

In preemptive multitasking, we can run several jobs simultaneously even if the previous job has not finished. It forces a job in execution to release the processor so that some other more important program can be executed.

In non-preemptive multitasking, a scheduled program or job is always completed before another scheduling decision is made. In this jobs are executed in the order in which they are scheduled.

1.2.7 Multiprocessing Operating System:

Like multiprogramming, the multiprocessing also improves the speed of computer by employing two or more than two independent CPU that have the ability to simultaneously execute several programs into a computer system. Thus if one of the CPU breaks down, the other CPU will automatically takes over its job. In some multiprocessing systems, each CPU performs a specific type of application. One may handle batch processes and other may handle on-line jobs.

The different CPU's inside a computer system may either have a common main memory or separate memories. It depends on the amount of work and expenses involved. The basic organization of a multiprocessing system is shown in figure 2.3.

Advantages of Multiprocessing

- **Improves Performance-** By allowing parallel processing of segments of programs it improves performance of computer systems. Thus it results in increased throughput and lower turn around time.
- **Provides backup-** If one CPU fails other CPU is brought into use. Thus complete breakdown is rare
- Helps in proper utilization of all other devices
- Increased reliability

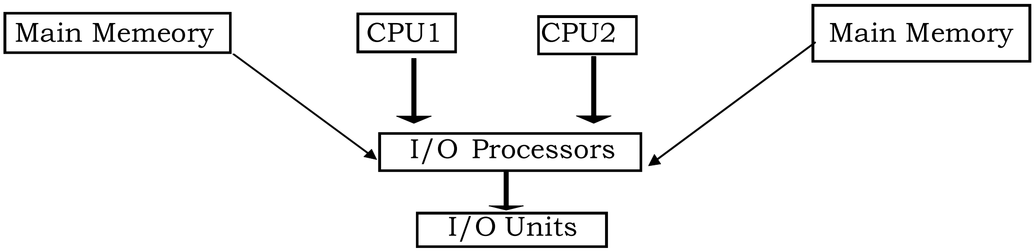


Figure 2.3

Disadvantages of Multiprocessing :

- Large amount of main memory is required.
- Sophisticated operating system is required to handle the processing of multiple CPU's.
- Use of multiple CPU's makes it very expensive.
- If one processor fails then it affects the speed

There are two types of Multiprocessing System

i) Symmetrical Multiprocessing System

ii) Asymmetric

In Symmetrical Multiprocessing systems, all the processors are of same type and they perform the same function. These types of processors are called homogeneous processors.

Advantage of Symmetrical Multiprocessing System

- Any processor can initiate an I/O operation and can run any process in the system.
- All the processors are available to handle whatever needs to be done next.

Asymmetric Multiprocessing System

In these systems all processors are not of same type i.e. there is some difference between processors. These types of processors are called heterogeneous processors. Thus, in Asymmetric multiprocessing system, different processors do different things. In such system, there may be the case that one processor controls the rest of the processors i.e. it is the supervisor of the others. The supervisor processor is known as master processor and the other processors are known as slave processors. This situation is known as master-slave. Some times, processor is responsible for doing one thing or task in the system (like handling the task view, sound I/O).

Advantages of Asymmetric Multiprocessing System

- In some situations, executing I/O operation as application program may be faster because, many processors may be available for a single job.

Disadvantages

- Complexity of the system is more due to the heterogeneous nature of processors.
- In this system, if the supervisor process handling a specific work fails, the entire system will go down

Differences between Multiprogramming and Multiprocessing:

Multiprogramming is the interleaved execution of two or more processes by a single CPU computer system. On the other hand, multiprocessing is the simultaneous processing of two or more processes by a computer system having more than one CPU. To be more specific, multiprogramming involves executing a portion of the one program, then a segment of another, etc., in brief consecutive periods. Multiprocessing, however, makes it possible for the system to simultaneously work on several program segments of one or more programs.

1.2.8 Timesharing Operating System

Timesharing is a technique of allocation of computer resources in a time dependent fashion to several programs/users simultaneously. Thus it helps to provide a large number of users direct access to main computer. In timesharing the CPU is divided among different users on a scheduled basis. Thus each user is given a brief share of the CPU time unlike multiprogramming where CPU is allocated to programs on priority basis. This very brief share of CPU time is called the time slice, time slot or quantum, which may vary from a few milliseconds to a few seconds. (Normally the time slice is of a few milliseconds only.) Since each user-gets some time of CPU and time for next time slice to be allotted is after few seconds so the user gets an illusion that he' is alone using the computer.

In timesharing systems the users will get the CPU one by one in the circular fashion, i.e. CPU will switch over to user2 after serving user1 and then to user3 as shown in fig below. CPU will switch over to another user in the following conditions:

i) Time slice for that user/program is over.

ii) That particular user/programs has encountered an I/O operation.

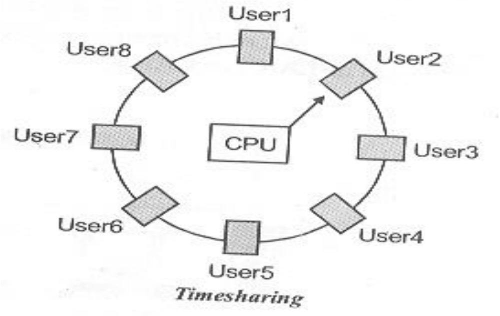


Figure 2.4

Requirements of the Time-sharing Systems:

Time-sharing systems typically require the following additional hardware and software features:

- A number of terminals simultaneously connected to the system, so that multiple users can simultaneously use the system in interactive mode.
- A relatively large memory
- Memory protection mechanism to prevent one job's instructions and data from other jobs in a multiprogramming environment
- Job status prevention mechanism to preserve a job's complete status information when the CPU is taken away from it and restoring this information back, when the CPU is allocated again to it.
- A special CPU scheduling algorithm, which allocates the CPU for a short period one-one one to user process in a circular fashion.

- An alarm clock mechanism to send an interrupt signal to the CPU after every time slice.

Advantages of Time Sharing

- Effective utilization and sharing of the resources. The main memory, disk storage, CPU is fully utilized and shared by different users.
- Reduces CPU idle time. The CPU is busy most of time as it switches from one program to another is rapid succession. This increases throughput and lowers turn around time
- Avoids duplication of software which are used by most of the users. These softwares are stored in the system libraries.

Disadvantages of Time Sharing:

- Security and Integrity of user programs and data may not be maintained as large number of users access the system simultaneously.
- Data transmission charges are very high in comparison to other methods.
- Since regular swapping needs to be done, so when a large number of users are accessing the CPU then overhead may be involved. This results in a poor response.

1.2.9 Summary

An operating system may perform a task serially (sequentially) or concurrently (multiple tasks simultaneously). Thus, all the resources of a computer system may be allocated to multiple programs. Batch processing allowed automatic job sequencing by resident operating systems and greatly improved the overall utilization of the computer. The computer no longer had to wait for human operation. CPU utilization was still low, however, because of slow speed of the I/O devices related to that of CPU.

The operating system designed for the working of a single user is known as single user operating system. MS-DOS is a single user operating system. The operating system in which data from one machine can be shared by other user at some other machine and vice-versa is called multi-user operating system. System security and integrity are important features of this system.

To improve the overall performance of the system, developers introduce the concept of multiprogramming. With multiprogramming, several jobs are kept in memory one at time, CPU is switched back and forth among them to increase the CPU utilization, and to decrease the total time needed to execute the jobs.

Multiprogramming, which was developed to improve performance, also allows time-sharing. Time-shared operating system allows many user (from one to several hundred) to use a computer system interactively at the same time.

1.2.10 Keywords

CPU-Bound : The jobs which needs CPU operatin maximum

I/O Bound : The jobs which requires I/O operation maximum

Degree of multiprogramming : No. of jobs loaded into the main memory

Process Control Block (PCB) :	Status information of each process
Homogeneous processors :	Processors of exactly same type
Heterogeneous processors :	Processors which are of different type and capacity

1.2.11 Self-Check Exercise

1. List various types of basic operating systems.
2. What do you mean by Batch processing?
3. Give the merits and demerits of Batch Processing.
4. What do you mean by Multiprogramming?
5. List the advantages and disadvantages of multiprogramming.
6. What do you mean by Multitasking?
7. How multitasking is different from multiprogramming?
8. Differentiate between single user operating system and multi-user operating system.
9. What do you mean by time-sharing operating system?
10. Compare time-sharing with multiprogramming operating system.

1.2.12 Suggested Readings

1. "Fundamentals of Information Technology", by Dr. Hardeep Singh, S.K. Kakar, by Lakahanpal Publication.
2. "Operating Systems concepts", by Silberschatz, Galvin by Personed Education
3. "Computer Fundamentals" , by P.K. Sinha, by BPB publication
4. "Operating Systems", by Gary Nutt, by Pearsoned Education
5. “System Programming & Operating System by D.M. Dhamdhere, Tata McGraw Hill.

PGDCA **Paper : PGDCA-2**

Operating System

Lesson No. : 1.3 **Author: Arun Bansal**

Modern Operating Systems

1.3.0 Objective

In this lesson the students are made familiar with various types of modern operating systems such as Client Server system, Distributed systems, Network operating system, real time operating systems and parallel systems.

1.3.1 Introduction

In lesson two the basic types of operating systems such as multiprogramming, multitasking, batch processing and timesharing were discussed. To improve the CPU utilization and catering the modern needs, the operating systems can be further categorised as client-server, distributed, Network operating systems, Real time and parallel systems. All these are discussed in this lesson.

1.3.2 Client -Server Systems

The client-server systems involve the splitting of an application into tasks and putting each task on the computer where it can be handled most efficiently. The computers and the operating systems of the clients and the servers may be different. It is a hierarchical strategy in which individual computers share the processing and storage workload with a central server. The client server software can be used on LAN's or WAN's.

Client/Server software is valuable to large modern organizations because it distributes processing and storage workloads among resources efficiently. The users can access the information they need much faster.

The most common example of client/server computing is a database that needs to be accessed by many different computers on the network. The server portion of the database contains all the data and tables on which various operations by various clients are being performed. The clients only run a portion of the DBMS. When a request is send by the client to the server then the server searches the database, collects relevant information and sends it back to the client. Different clients can give the same request to the server. The server receives and processes request of each client one by one.

1.3.3 Distributed Systems

A distributed computer system is a collection of autonomous computer systems capable of communication and cooperation via their hardware and software interconnections. Distributed computer systems are generally characterized by the absence of shared memory or a clock. Instead, each processor has its own local memory. The processors connected communicate with each other through various communication lines such as high-speed buses or telephone lines.

A distributed operating system is one that looks to its users like an ordinary centralized operating system but runs on multiple independent CPUs. A distributed operating system governs the operation of a distributed computer system and provides a virtual-machine system abstraction to its users. The key objective of a distributed operating system is transparency. Ideally, component and resource distribution should be hidden from users and application programs unless they explicitly demand otherwise. That is, the network becomes “invisible,” analogous to the way that virtual memory hides the secondary storage Another way of expressing the same idea is to say that user views the system as virtual uniprocessor but not as a collection of distinct machines. In a true distributed system, users are not aware of where their programs are being run or where their files are residing; they should all be handled automatically and efficiently by the operating system. The allocation of jobs to processors and files to disks, movement of files and all other system functions are automatic. The processors in a distributed system may vary in size and function. These processors are usually referred as sites, nodes or computers etc.

Distributed operating systems have many aspects in common with centralized ones but they also differ in certain ways. Distributed operating system, for example, often allow programs to run on several processors at the same time, thus requiring more complex processor scheduling (scheduling refers to a set of policies and mechanisms built into the operating systems that controls the order in which the work to be done is completed) algorithms in order to achieve maximum utilization of CPU’s time.

Fault-tolerance is another area in which distributed operating systems are different. Distributed systems are considered to be more reliable than uniprocessor based system. They perform even if certain part of the hardware is malfunctioning. This additional feature supported by distributed operating system has enormous implications for the operating system.

The distributed systems are different from computer networks. In case of network one has to explicitly log onto the machine, explicitly handle all the network management personally. But in case of distributed system user is not aware of other nodes connected. In fact, a distributed system is a software system built on the top of network. It has high degree of transparency and cohesiveness. Internet is an example of distributed systems and LAN, WAN's are examples of Networks.

In a distributed operating system, the users access remote resources in the same manner as they do local resources. Data computation and process migration is under the control of distributed operating system.

Linux (apeche services), Alpha kernel, Amoeba, Angle, Chorus, Windows Server 2003, 2008, 2012, Ubuntu, Mach are few examples of distributed operating systems.

1.3.3.1 Data Migration

Suppose that a user on site A wants to access data (such as a file) that reside at site B. There are two basic methods for the system to transfer the data. One approach is to transfer the entire file to site A. From that point on, all access to the file is local. When the user no longer needs access to the file, a copy of the file (if it has been modified) is sent back to site B. Even if only a modest change has been made to a large file, all the data must be transferred. This mechanism can be thought of as an automated FTP system.

The other approach is to transfer to site A only those portions of the file those are actually, necessary for the immediate task. If another portion is required later, another transfer will take place. When the user no longer wants to access the file, any part of it that has been modified must be sent back to site B.

Clearly, if only a small part of a large file is being accessed, the latter approach is preferable. If significant portions of the file are being accessed, it is more efficient to copy the entire file.

Note that it is not sufficient merely to transfer data from one site to another. The system must also perform various data translations if the two sites involved are not directly compatible (for instance, if they use different character-code representations or represent integers with a different number or order of bits).

1.3.3.2 Computation Migration

In some circumstances, it may be more efficient to transfer the computation, rather than the data across the system. For example, consider a job that needs to access various large files that reside at different sites to obtain a summary of those files. It would be more efficient to access the files at the sites where they reside and then to return the desired results to the site that initiated the computation. Generally, if the time to transfer the data is longer than the time to execute the remote command, the remote command should be used.

Such a computation can be carried out in a number of different ways. Suppose that process P wants to access a file at site A. Access to the file is carried out at site A and could be initiated by a remote procedure call *(*RPC*).*

Alternatively, process P can send a message to site A. The operating system at site A would then create a new process Q whose function is to carry out the designated task. When process Q completes its execution, it sends the needed result back to P via the message system. Note that, in this scheme, process P may execute concurrently with process Q, and in fact, may have several processes running concurrently on several sites.

Both methods could be used to access several files residing at various sites. One remote procedure call might result in the invocation of another remote procedure call, or even in the transfer of messages to another site. Similarly, process Q could, during the course of its execution, send a message to another site, which in turn would create another process. This process might either send a message back to Q or repeat the cycle.

1.3.3.3 Process Migration

A logical extension to computation migration is process migration. When a process is submitted for execution, it is not always executed at the site in which it is initiated. It may be advantageous to execute the entire process or parts of it, at different sites. This scheme may be used for several reasons:

- **Load balancing**: The processes (or sub processes) may be distributed across the network to even the workload.
- **Computation speedup**: If a single process can be divided into a number of sub processes that may run concurrently on different sites then the total process turnaround time can be reduced.
- **Hardware preference:** The process may have characteristics that make it more suitable for execution on some specialized processor (such as matrix inversion on an array processor rather than on a microprocessor).
- **Software preference:** The process may require software that is available at only a particular site and either the software cannot be moved, or it is less expensive to move the process.
- **Data access:** Just as in computation migration, if the data being used in the computation are numerous, it may be more efficient to have a process run remotely, rather than to transfer all the data locally.

There are basically two complementary techniques, which can be used to move processes in a computer network. In the first, the system can attempt to hide the fact that the process has migrated from the client. This scheme has the advantage that the user does not need to code his program explicitly to accomplish the migration. This method is usually employed for achieving load balancing and computation speedup among homogeneous systems, as they do not need user input to help them execute programs remotely.

The other approach is to allow (or require) the user to specify explicitly how the process should migrate. This method is usually employed in a situation when the process must be moved to satisfy a hardware or software preference.

Advantages of Distributed Operating Systems:

1. **Major breakthrough in microprocessor technology**: Microprocessors have become very much powerful and cheap, compared with mainframes and minicomputers, so it has become attractive to think about designing large systems consisting of small processors. These distributed systems clearly have

a price/performance advantages over more traditional systems.

2. **Incremental Growth**: The second advantage is that if there is a need of 10 per cent more computing power, one should just add 10 per cent more processors. System architecture is crucial to the type of system growth, however, since, it is hard to give each user of a personal computer another 10 per cent.
3. **Reliability:** Reliability and availability can also be a big advantage; a few parts of the system can be down without disturbing people using the other parts; On the minus side, unless one is very careful, it is easy for the communication protocol overhead to become a major source of inefficiency.
4. **Communication -** The various nodes connected can communicate with each other. The information can be exchanged between them. In case of email, the user can exchange files with one another. The windows systems, which are example of distributed systems, also share and transfer data between displays.
5. **Resource sharing -** The different resources, which are available over the system, can be shared. For example, user at node A may use printer available at node B.
6. **Computation speed up -** The various computations, which can be portioned into sub computations can run concurrently and thus increasing the computation speeds.
7. **Performance** improvements come mostly from the potential for parallel operation when multiple nodes cooperate on solving a single problem. Response time, a measure of performance, may be improved in a distributed system by virtue of the ability to place frequently used data close to their users.

Other potential advantages of distributed systems include reduced costs greater capacity than a single processor can deliver and better ability to parallel real-world organizational structure due to local control of local data.

Disadvantages of distributed Operating System:

- Reduced ability to pool memory and processor resources among distinct nodes
- Increased dependence on network performance and reliability
- Security weaknesses due to sharing
- More complex system administration and maintenance
- some messages can be lost in network system

1.3.4 Network Operating System

A network operating system is a collection of software and associated protocols that allows a set of autonomous computers, which are interconnected by a computer network to be used together in a convenient and cost-effective manner. In a network operating system, the users are aware of existence of multiple computers and can log into remote machines (remote login) and copy files from one machine to another machine (remote file transfer). The system has little or no fault tolerance; if 5% of the personnel computers crash, only 5% of the users are out of business. The two major features of the Network operating system are as follows.

1.3.4.1 Remote Login

An important function of a network operating system is to allow users to log in remotely on other computers. TELNET is a software that allows us to access remote computers. That is, it allows us to log on to another machine - a computer that might be on campus or on the other side of the world - from the terminal we are using. For example, if you were in New Delhi you could Telnet from a machine there to log in to your account here at Patiala. Telnet can help you to keep in touch with your account and with other technological resources no matter where you — or they — are located.

Telnet is short for "computing at a distance" and is what the Internet was originally set up to do. Assuming you know the login code and the password, Telnet allows you to log in to remote computer sites and access the information contained inside those systems. Many computer systems on the Internet allow for such access.

Telnet is a worldwide Internet protocol for remote logins. Its general use is one of the reasons why the Internet is so powerful. One function of the Telnet software on your PC is to convert the PC in front of you into a terminal that can be connected to a remote computer. Exactly what any terminal does with the signals that come to it from another(remote) computer or from the keyboard is determined by hidden control codes. Different manufacturers employ different control codes.

Telnet for Windows comes with a selection of menus and a scroll bar at the side. The scroll bar is useful for viewing what has rolled off the top of the screen. You will seldom need any of the menus, except the one shown on the slide for connecting to your chosen remote computer. Of course most computers won't let you log on to them unless you have an account and a password. The library computer is an exception. Its name is simply library as far as Telnet is concerned and you do not need a password. You will find out at the end of the afternoon how you can login to almost any University library catalogue in the country.

Telnet is based on client/server model. It means you run a piece of software on your PC (also as client) to use the resources of a distant server computer. The distant computer is called the host. Host allows many different clients to access its resources at the same time, when you contact the host; the distant computer and your computer decide which terminal emulation will be used. Terminal emulation determines how your keyboard will transit information to the distant computer and how information will be displayed on your screen.

Telnet is a terminal emulation program for TCP/IP networks. The telnet programs runs on a computer and connect PC to the server on the network. Commands can be executed through the Telnet program as if you were entering them directly on the server console.

To access the Telnet a host name or IP address of machine is required to identify the computer to which you want to connect. To start a telnet session, login to server by entering a valid user name and password. The machine from which telnet commands are executed is the local computer (client program). The machine to which connection is made is the remote machine Once local computer is connected to the remote server, client and server communicate through Telnet protocol. The Telnet protocol assumes that each end of the connection the client and the server is a Network Virtual Terminal NVT. Each (NVT) has a virtual "keyboard". Keyboard sends data from one NVT to the other. When you type text on your keyboard, you are using the NVT keyboard the printer receives and displays the data on the computer screen. When a distant Telnet connection sends the data and displays it on your screen, it is the printer that displays the information. Packets have to go through many Internet routers in each direction between your computer and the host, there might be delay between the time you send a command and the time you see the results on your own computer screen.

Telnet is used under following circumstances:

1. When you want to login on a remote computer and want to do some work on that host
2. When you want to use one of the Internet's client/server applications but don't have client program installed on your machine.
3. To access a stand alone application that is installed on another computer.

Telnet allows to login in to a remote computer. Services and resources can be shared from one another. Telnet is accessed by typing telnet followed by the domain name or IP address of the remote system or IP address followed by port number form:

1. Telnet domain name, e. g. telnet abc. de. edu
2. Telnet IP address, e. g. telnet 198. 162.0.4
3. Telnet domain-name port number e.g. telnet abc. de. edu. 400

For Window Users:

To open a Telnet window click on the start button, then select Run. Instead of supplying the name of the remote computer IP address of the computer can also be used. Click on the button and a telnet windows should appear. Type username for the "Login" and hit return. After type in your "Login". Enter "password", note that password will not appear on the screen. There after the valid telnet commands can be used.

1.3.4.2 Remote File Transfer FTP- (File Transfer Protocol)

One of the most popular use of Internet is to download files-that is, transfer files from a computer on the Internet to your computer. These files can view sounds and music to listen, texts the Internet. The Internet's File Transfer Protocol (FTP) is also used to upload files from your computer to another computer on the Internet.

File Transfer Protocol FTP is used to copy or transfer programs to and from other computers. These computers may be at the same site or at different site thousands of miles apart. It is a general protocol that works on Unix system as well as a variety of other systems. This simple protocol let to do many things: download s/w, upload web pages transfer information between home and work machine. For this purpose, the local machine refers to the machine initially logged into the one on which you type the ftp command. FTP acts as interpreter on the remote machine. Note that several FTP exist on the Internet that store different types of files. Severs may be maintained by universities or research institutions or government agencies or individuals. Different types of files-text, audio image or web file cab be transferred to and from the computer copying files from local machine to the remote machine is called uploading where as copying files from remote machine to local machine is called downloading.

Anonymous FTP:

Anonymous FTP allow to copy files from a remote machine which do not have login name when the remote machine asks for login name type anonymous, for password enter your e-mail address. This allows the remote site to keep records of the anonymous FTP requests. When you are logged in, then you are in the anonymous directory for the remote machine. This contains a number of public files and directors. Files should be moved around in these directories. Files can only be copies from the remote machine to the local machine but not able to write on the remote machine or to delete any files there. Systems that allow anonymous FTP sessions and called anonymous FTP sites and the collection of files they make available are called FTP archies.

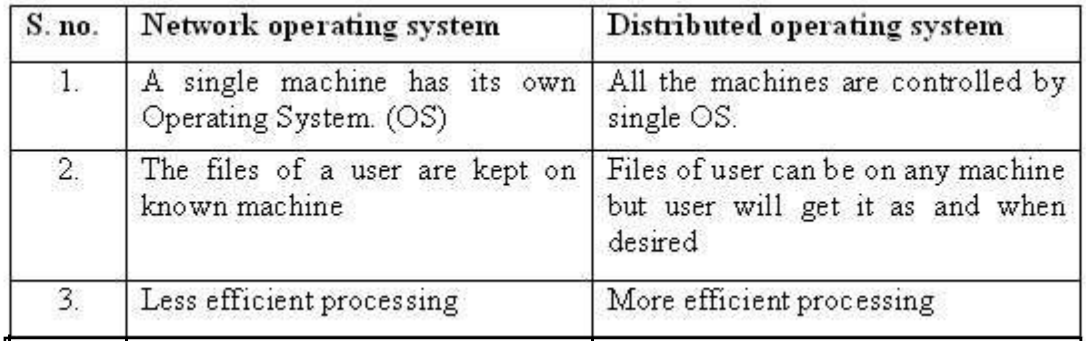
When you log on to the FTP server, a connection called command link is opened up between your computer and the server. The link is used for sending commands to the server and sending messages and information back form the server to your computer when a command is issued to download a file, Data link is established. This link can be opened in one of two modes:- ASCII mode or the binary mode. ASCII mode is used for sending text files and binary mode is used for sending binary files.

Advantages of Network operating system:

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

Differences between Network operating systems and distributed operating systems:

- Each computer has its own private operating system instead of running part of a global system wide Operating system. .
- Each user normally works on his/her own system; using a different system requires some kind of remote login, instead of having the operating system dynamically allocate processes to CPUs.
- Users are typically aware of where each of their files are kept and must move file from one system to another with explicit file transfer commands instead of having file placement managed by the operating system.



S. no.	Network operating system	Distributed operating system
1.	A single machine has its own Operating System. (OS)	All the machines are controlled by single OS.
2.	The files of a user are kept on known machine	Files of user can be on any machine but user will get it as and when desired
3.	Less efficient processing	More efficient processing
4.	Architecture model is client server model	Architecture model is master-slave model

1.3.5 Real Time Systems

The operating systems in which a large number of processes (program in execution) external to the computer systems are performed quickly in a short instance of time within a certain deadline are known as real time processing systems. A real time system is used when there are rigid time requirements on the operation of a processor or the flow of data, and is thus often used as control device in a dedicated application. A real time system has well defined, fixed time constraints. These systems involve a very large number of techniques, which are to be executed immediately. The data which is received from some external environment, is processed and returned back as processed output back to the system in a short span of time. The updated files along with the transactional data is returned immediately back to the system. Real time systems are basically online processing systems with the advantage of inquiry processing. The response of the system to the inquiry itself is used to control the activity. Real Time Systems operate in multiprocessing and multiprogramming environment. Both these processes increase the availability and reliability of the working systems.

A primary objective of the real time systems is to provide quick event-response times, and thus meet the scheduling deadlines. User convenience and resource utilization are of secondary concern to real time systems designers. It is not uncommon for a real-time system to be expected to process burst of thousands of interrupts per second without missing a single event. Such requirements usually cannot be met by multiprogramming alone and real-time operating systems usually rely on some specific policies and techniques for doing their job.

A real time system is often used as a central device in a dedicated application like those that have been listed here :

System that cantrol scientific experiments, Medical imaging system, Robotics, Fuel-injection systems, Air-traffic control, Control system.

On the basis of time taken to process the processes the real-time processing systems can be basically classified into two categories

(a) Hard Real Time Systems

(b) Soft Real Time Systems

The Hard Real Time Systems guarantee that critical tasks complete on time. The goal requires that all delays in the system be bounded to the retrial of stored data to the time that it takes operating system to finish any request made to it. In hard real time applications time delay of milliseconds between execution of processes and returning back to the system is also very important. These applications employ the multiprogramming and multiprocessing to execute their processes. The response of the system to an inquiry itself is done on a very short span of time to control the activity. Consider an example of balloons provided by car makers along with handle in the driver's seat. When the driver applies brakes at a particular instance the balloon grows and prevents the driver's head to hit the handle. Had there been delay even of milliseconds then it would have resulted in an accident. This is an example of hard real time processing systems. In practicality hard real time systems are very rare.

In Soft Real Time Systems, a critical real-time task gets a priority over the other tasks, and retains that priority until it completes. Soft real time is an achievable goal that is amenable to mixing with other type of systems. Given their lack of deadline support, they are risky to use for industrial control and robotics. In Soft Real Time Systems the processes, which are to be performed quickly in a short span of the time, can have time difference of few seconds in their execution and returning of processed data back to the system. This is not the case in hard time applications where time lapse in milliseconds is also very important. The examples of Soft Real Time Processing are availability of flight seats, wholesale supplier and manufacturer information on availability of stocks.

The use of Real Time Processing techniques is increasing day by day. These techniques, which usually operate in multiprogramming and multiprocessing, increase the reliability of the systems. Typical examples of realtime systems include Air trafic cantrol systems, Network Multimedia Systems, Command Cantrol systems etc.

1.3.6 Parallel Systems

Parallel processing systems are designed to perform simultaneous data processing of different jobs for increasing the computation speed of the computer system. These systems are multiprocessor systems and such systems have more than one processor in close communication, sharing the computer bus, the clock, and sometimes memory and peripheral devices. These systems are referred to as tightly coupled systems. In these systems the instructions may not be executed in a sequential manner they are executed as the space is available for them.

There are several reasons for building such systems. One advantage is increased throughput. By increasing the number of processors, we hope to get more work done in a shorter period of time. The speed-up ratio with n processors is not **n,** however, but is rather less than n. When multiple processors cooperate on a task, a certain amount of overhead is incurred in keeping all the parts working correctly. This overhead, plus contention for shared resources, lowers the expected gain from additional processors. Similarly, a group of n programmers working closely together does not result in n times the amount of work being accomplished.

Multiprocessors can also save money compared to multiple single systems because the processors can share peripherals, cabinets and power supplies. If several programs are to operate on the same set of data, it is cheaper to store that data on one disk and to have all the processors share them, rather than to have many computers with local disks and many copies of the data.

Another reason for multiprocessor systems is that they increase reliability. If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, but rather will only slow it down. If we have 10 processors and one fails, then each of the remaining nine processors must pick up a share of the work of the failed processor. Thus, the entire system runs only 10 percent slower, rather than failing altogether. This ability to continue providing service proportional to the level of surviving hardware is called graceful degradation. Systems that are designed for graceful degradation are also called fault-tolerant.

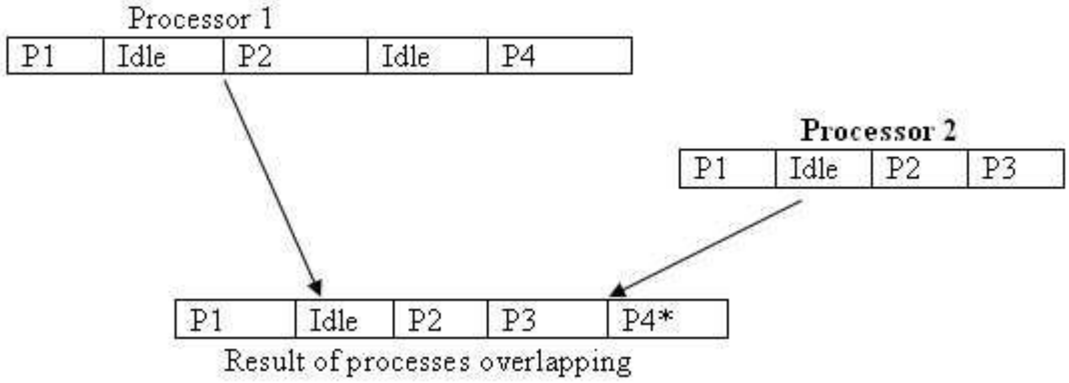


Figure : 3.1

Suppose processor 1 is running the following processes PI, P2, P4 and processor 2 is running the PI, P2 and P3 process. When their processes are made to overlap the result is better resource utilization.

Continued operation in the presence of failures requires a mechanism to allow the failure to be detected, diagnosed, and corrected (if possible). The Tandem system uses both hardware and software duplication to ensure continued operation despite failures. The system consists of two identical processors, each with its own local memory. A bus connects the processors. One processor is the primary, and the other is the backup. Two copies are kept of each process; one on the primary machine and the other on the backup. At the fixed checkpoints in the execution of the system, the state information of each job (including a copy of the memory image is copied from the primary machine to the backup. The major advantage of this scheme is that even if one processor goes faulty the other will continue to work and status information about the faulty processor is also displayed, so that necessary action for correction can be taken. So at extra cost we are getting more reliable machine. These types of systems are used in defence.

There are a number of ways to classify the parallel processing systems. They can be classified on the basis of number of instructions and the data item that can be manipulated simultaneously. Flynn gave such a classification. According to him we could classify parallel processing systems into four categories.

1. Single Instruction Stream, Single Data Stream (SISD)
2. Single Instruction Stream, Multiple Data Stream (SIMD)
3. Multiple Instruction Stream, Single Data Stream (MISD)
4. Multiple Instruction Stream, Multiple Data Stream (MIMD)

Parallel processing is employed in supercomputers. This technique is also useful in areas like weather forecasting, Artificial Intelligence, Expert systems, Image processing, Medical diagnosis, Flight simulations, petroleum explorations etc. A very important example of parallel system is to perform matrix for a matrix addition of large size. In the matrix addition only corresponding cells have to be added, so the matrix can be divided into two or four parts and the corresponding cells are added separately by different processors. Finally the results are combined.

Advantages of Parallel System:

1. As there are a number of processors so a job can be completed in a shorter span of time.
2. Secondly it saves money as multiprocessor systems share peripherals, cabinets and power supplies.
3. Thirdly multiprocessor systems provides more reliability. Due to multiple processors the system will never stop functioning if one of the processor fails because the remaining processors will share the job off ailed processor.

Disadvantages of Parallel System *:*

1. The amount of main memory required is quite large.
2. An operating system, which can perform scheduling job of handling multiple no. of CPU's is required.
3. Due to large memory requirements and multiple number of CPUs' presence the initial expenses are quite high.

1.3.7 Summary

The client server involves the splitting of an application into tasks and putting each task on the computer where it can be handled more efficiently. Each process is under the control of central server.

A distributed system is a collection of processes that share memory or clock. Instead, each processor has its own local memory and the processors communicate with one another through various communications lines. A distributed system provides the user with access to the various resources located at remote sites.

A network operating system is a collection of software and associated protocols that allows a set of autonomous computers, which are interconnected by a computer network to be used together in a convenient and cost-effective manner.

The parallel processing system is a multiprocessing system in which different processes share the computer bus, the clock memory and peripheral devices. This results in increase throughput.

A real time system is used when there are rigid time requirements on the operation of a processor. A hard-real time system is often used as control device in a dedicated application. A hard real time operating system has well defined, fixed time constraints. Processing must be done within the defined constraints, or the system will fail. Soft real time systems have less stringent timing constrains and do not support deadline scheduling.

1.3.8 Keywords

LAN	:	Local Area Network
WAN	:	Wide Area Network
DBMS	:	Data Base Management System
Uniprocessor	:	A system with a single processor
RPC	:	Remote Procedure Call

1.3.9 Self-Check Exercise

1. List various types of Modern operating systems.
2. What do you mean by Distributed Operating System?
3. Give the merits and demerits of Distributed Operating System.
4. What do you mean by Network Operating System?
5. List the advantages and disadvantages of Network Operating System.
6. Differentiate between Network operating system and Distributed Operating system.
7. What do you by real systems? How they can be categorized?
8. List the differences in the working style of hard real time systems and soft real time systems.
9. What do you mean by parallel systems?
10. Give the merits of parallel systems.

1.3.10 Suggested Readings

1. “Fundamentals of Information Technology”, by Dr. Harddep Singh, S.k. Kakar, by Lakahanpal Publication.
2. “Operating Systems concepts”, by Silberschatz, Galvin by Personed Education
3. “Scripting Language and Wed Designing”, by Singh, Verma, by “Kalyani Publisher.
4. “Operating Systems”, by Gary Nutt, by Pearsoned Education
5. “System Programmings Operating Systems” by D.M. Dhamdhere, Tata McGraw Hill.

Post Graduate Diploma in Computer Applications — **Paper : PGDCA-II Operating System**

Lesson No. 1.4 — **Author : Arun Bansal**

Introduction to DOS

1.4.0 Objective

In this lesson the students are made familiar about the basic concepts of DOS (Disk Operating System). The brief history of DOS is given and later on the students are made familiar with basic terminology of the DOS i.e program, file, file naming, directory, path and system prompt i.e command prompt. At the end of the lesson various features of DOS are given.

1.4.1 Introduction

MS-DOS **stands for** Microsoft Disk Operating System**. Microsoft is the name of the company, which owns DOS.** It was the first widely installed operating system for personal computers. It is a single-user operating system used normally in PC's**. MS-DOS is a command line user interface while MS-DOS is not commonly used by itself today, it can still be accessed from windows 95, windows 98 or windows XP by clicking start/run and typing command or cmd.**

With the production of first personal computer by MITS, MEXICO in 1975, which used 8-bit Intel 8080 CPU and had a memory of 256 bytes, began the revolution in the PC market. This PC was called Altair. Bill Gates wrote a program in BASIC for this PC to make it work, it was this program which made Bill Gates, a billionaire.

Within a next few years, more and more companies started manufacturing these PC, which were based on 8080 Intel chip. Almost every PC was working with on operating system, produced by Digital Research Co., of California. This operating system used by these personal computers (also known as Micro

Computers) was known as CP/M. The PC was used basically for entertainment purposes.

IBM, which in 1980's dominated the computer industry made plans to make the PC manufacturing its area and started developing its own personal computers. The problem was, if it starts from the scratch to manufacture the PC, it would take long time therefore it decided to assemble various components manufactured by other manufacturers and make a PC. Intel had already launched its newer versions of the chip i.e., 16-bit 8086 and 8088. IBM choose 8088 as its chip. IBM, as it was concentrating on hardware was not at all interested in software. Therefore it approached Bill Gates, who had started his company, the Microsoft, to use the BASIC interpreter, which he had designed for their PCs and even asked for an operating system if possible.

As Microsoft was selling AT & T UNIX, which needed about 100K of memory for operating system itself and also a hard disk and the IBMs machine had altogether 64K of memory and no hard disk, Bill Gates referred them to use Digital Research's CP/M-86 Operating System. When IBM approached Digital Research, it said that it would take long time for CP/M-86 as it was behind schedule. As IBM could not wait, it re-approached Bill Gates to write an operating system, similar to the CP/M. Then Bill Gates to save time, bought an operating system known as 86-DOS which was the product of Seattle Computer Products, which was used to test memory boards manufactured by it. He also hired Tim Paterson, the author of 86-DOS and it was upgraded and renamed as MS-DOS (Micro Soft-Disk Operating System). Thus MS-DOS had its birth. This Operating System was given to IBM as an operating system, then IBM came with PCs which had MS-DOS as operating system.

The basic virtue of the MS-DOS as perceived by IBM and other companies was that the MS-DOS was capable of running almost every software, which was running on CP/M with 8080 chip with as little changes as possible. This happened in August 1981. The MS- DOS version, which was released with IBM-PCs, was known as version 1.0. The memory occupied by it was as low as 12K. It supported 5.25 -inch diskette with 160K of capacity.

1.4.2 History of DOS

MS-DOS was the operating system from Microsoft before the first version of Windows. MS DOS (Microsoft Disk Operating System), a product of Microsoft Corporation of USA., is the most popular operating system for PCs. Another operating system available in the market is the PC DOS, a product of IBM, which is almost similar to MS DOS. Whether you use MS DOS or PC DOS, the basic commands of DOS remain the same. There were many limitations to DOS including that you could only do one thing at a time. Also, DOS was a text-based system. The user required to type commands on the prompt.

The following is a brief history of DOS. The following summarizes all versions of DOS:

- DOS 1.0 - Released in 1981 to supplement the IBM-PC IBM-PC. First version of DOS. Supported 16K of RAM, single-sided 5.25 inch 160K Floppy.
- DOS1.1 - Fixed many bugs from 1.0 Double-sided floppy support for 320K drives.
- DOS 2.0 - Supplemented the release of IBM's XT in 1983. more than twice the commands of 1.x hard disk support (very small, around 5MB).
- DOS 2.1 - Supplemented the release of IBM's PC. Some minor improvements were added.
- DOS 3.0 - Designed to support newer IBM-AT Hardware. A few LAN features added, (hardly any).
- DOS 3.1 - More LAN features and support added.
- DOS 3.2 - Added support for 3.5 inch floppy drive (720K).
- DOS 3.3 - Added support for IBM's PS/2 and the new 3.5-inch or 1.44MB floppy drive. New international character set was added with support for 17 countries.
- DOS 1.4.0 - DOS Shell added, some minor changes and bug fixes.
- DOS 5.0 - Implemented in 1991, including superior memory management features and tools Macro support, Shell enhancements.
- DOS 6.x - Support for Microsoft Windows, disk defrag, file compression, backups, anti-virus, Memmaker, etc.

1.4.3 Loading DOS

To be able to use a computer system, the operating system (DOS) must always be present in its memory. So, as and when a computer system is switched on, the very first thing that is placed in the computer’s memory (i.e. RAM) is DOS. Thus, loading DOS in computer’s memory is called *Booting* the system. The booting happens automatically when the computer is switched on, provided DOS is available on it. DOS can be present on both hard disk as well as the floppy disk. So, when the system is turned on, a search for DOS is done on floppy drive first. If it is not present on the floppy drive, the DOS is searched for on the hard disk. Thus, for booting to happen, the DOS must be present either on the floppy or on the hard disk. So, if DOS it not present on the hard disk, insert the floppy carrying DOS software in the A drive before switching on the system.

The DOS operating system is set of a programs, which is, stored on some secondary storage device normally floppy disk or hard disk. It is then loaded into RAM when required. The DOS software is divided into three parts and stored in three different files on a disk. The disk, which contains these three files, is called a Bootable disk. The three files loaded on this disk are IO.SYS, MSDOS.SYS and COMMAND.COM. The IO.SYS is also called IBMBIO.COM in some systems.

The IO.SYS file contains the device drivers for the standard devices such as keyboard, disk, floppy, printer and monitor. It also contains the basic low-level services such as Time-of-day and System configuration analysis. All these device drivers are often called BIOS (Basic Input Output System). The IO.SYS also contains SYS.INIT file, which loads MSDOS.SYS from hard disk into memory. The IO.SYS is also called IBMBIO.COM in some systems.

The MSDOS.SYS file is also called DOS Kernel. It contains all the modules for process management. These modules are written in machine independent manner so that they could be easily ported. The MSDOS.SYS is also called IBMDOS.SYS in some systems.

The DOS Kernel provides the link between the BIOS and user's application programs, which provides the interface for the application program. It provides more efficient control over the Input Output operations. The DOS kernel performs four basic important functions:

1. **Memory Management** It allocates memory to different application programs on demand.
2. **Process Management** It does the management of the program by loading the program requested by the user from disk in memory and helps in execution of the program. It also helps to recover system files when unwanted termination occurs.
3. **File Management** It does the management of files by reading, writing, displaying, renaming, copying and removing the files.
4. **Provides an interface-** It establishes interface between the user and the hardware. It works as an intelligent agent by allotting various hardware devices to various programs when needed. Programs can be written to have direct interface with the hardware by the user.

MS-DOS provides a command line interface. The MS-DOS shell is known as COMMAND.COM. It has about 70 or more commands that are partitioned into different categories-Internal commands and External commands. Internal commands are those commands that are always present in the resident portion of memory while external commands are those commands, which normally reside on the disk and are loaded into memory whenever they are to be executed. These commands are stored in some external files.

1.4.4 Basic Terminology of DOS:

Let us look at the terminology, which is used by MS- DOS.

i. **Program**

Program is basically a combination of set of various instructions, which are written using any computer language. These instructions are used to make the computer know the tasks, which it has to perform.

ii. **File**

File is collection of records, which are logically related. These are quite similar to the manual files, which we maintain to hold mails, memos etc. MS-DOS supports following types of files:

1. **Text Files**

   These are the files, which are used to store data. The data can be in the form of characters, numbers and any other symbols. These are the files, which are used to store input as well as output data of the programs. These files are also known as data files as they are used to store data.

2. **Batch Files**

   These are the files, which contain set of instructions, which are to be executed as a Batch in a sequential order. These files are usually used to perform a specific task.

3. **Application Program Files**

   These are the files which usually are application specific. They are used to do perform specific tasks, which will enable a user to do his job easily. Word processors, calculators, spreadsheets etc. are some of the application program files.

iii. **File Name**

There are certain rules that have to be followed while giving names to your files. A filename has two parts:

i. Primary Name

ii. Extension or Secondary Name

A dot (.) separates a primary name from an extension. Let us see the two parts of the file named DRAGON.TXT.

A primary name cannot have more than eight characters and similarly an extension can contain only up to three characters. A filename having only the primary name and no extension is absolutely valid because it is absolutely optional to give an extension to a filename. An extension is generally given by the language or software used by the file. For example, if you are entering

BASIC and PASCAL programs, their filenames would have extension BAS and PAS respectively. A filename can contain the following characters:

1. An Alphabet (A-Z) or (a-z)
2. A number (0-9)
3. Special characters such as underscore (_).
   except *, ?, full stop (.) and space

It is a good practice to give meaningful names to your files. However, no two files can have exactly the same name on the disk. Thus, a name given to a file on a disk has to be unique. The rules for naming a directory are the same as that of naming files.

Commonly used file extensions and there meaning.

File Extension	**Meaning**
.c	C source program file
.for	Fortran source program file
.pas	Pascal source program file
.cob	COBOL source program file
.cpp	C++ source program file
.obj	Object file
.exe	Executable file
.com	Executable file
.bat	Batch file
.txt	Text file
.bak	Backup file
.jpg	JPEG graphics file (Joint Photographic Expert Group)
.gif	GIF graphic file
.dat	Data file

iv. Directory

We usually feel and have a need to store all the related files at one place so that our search for a specific file becomes easier. For this reason, the concept of the Directory has emerged. Directory is logical grouping of files. A directory contains the names of the files it holds, along with their extension, the file size, the date of creation or the date of last modification. We can look at directory structure, which is an inverted tree, which has one directory at top level, or highest level, which is known as "root" directory, which is the Home directory for all the directories. Usually this directory happens to be the drive representation like C:\>, D:\>, E:\>, A:\>, B:\> and so on.

We usually use these directories to store related files, therefore we create directories and usually it is advisable to work in directories other than the root/Home directory. The directory in which one is currently working is known as current directory.

iv. Path

The location of directory with respect to the other directory has to be clearly stated if we are working under some other directory. This path can be represented by two methods. One giving the total path starting from the root directory is known as Absolute path.

The other way is to give a path with respect to the current directory and this is known as relative path. Let us look at the following directory structure and understand the absolute and relative path:

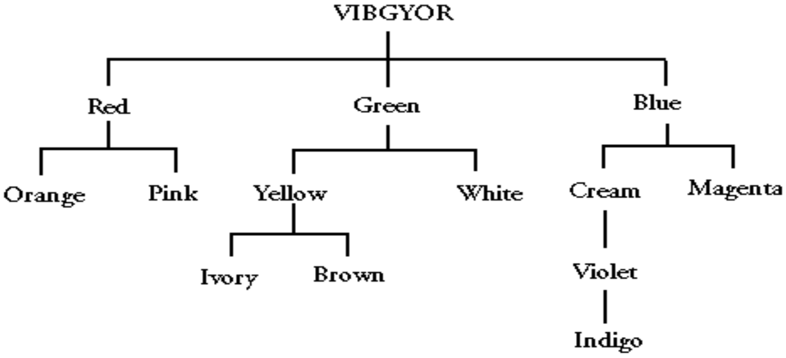


Fig 1.1: Directory Hierarchy

Let us look at the path for "Indigo" directory in both the ways:

a. Absolute Path

For absolute path, we start from root i.e. VIBGYOR, descendant directories separated by slash '\':

VIBGYOR\BLUE\CREAM\VIOLET\INDIGO

b. Relative Path

To understand the relative path, we need to know the following notation:

Current Directory is denoted by "." [dot]

Parent Directory is denoted by ".." [2 Dots]

Let us look at the relative path of "INDIGO", let us assume that we are at white i.e. current directory is white, now let us look at the relative path of "INDIGO" w.r.t. white.

.\..\..\BLUE\CREAM\VIOLET\INDIGO

The first dot represents white i.e. the current directory, next two dots take to the parent directory of white i.e. "Green", next two dots take to the parent directory i.e. VIBGYOR then the path takes to "INDIGO".

The relative path some times will be very easy to use when the destination directory is near the current directory.

V. System Prompt (Command Prompt)

A system prompt is an indication that the system is ready to take input/instructions from the user and give the output to the user. From the prompt we give the instructions to the computer. For MS-DOS, if we are using a hard disk computer, then the DOS is fetched into the memory of the computer from the hard disk. DOS assumes the default disk drive to be drive 'C\ so it gives an initial system prompt as C>. If any other drive is used to load the DOS into the memory, then it becomes the prompt. The prompt plays an important role, as it is an indication to start the working on the system. We can change the prompt from the initial system prompt to any other prompt, which we feel.

1.4.5 FEATURES OF DOS

1. It is a single user operating system and is a command user interface (CUI). It interprets commands typed on the DOS mode.
2. Standardization and openness of MS-DOS architecture has greatly benefited Software and Hardware developers.
3. Very good file, client and PRINT Server options.
4. Most recent development in DOS i.e. MS-DOS 7.0 has provided GUI (Graphics User Interface).
5. A pack of 'housekeeping tools' is available (e.g. listing contents, creating, copying, and deleting. It also repairs errors about file system like cross link and clusters. It also removes physical disk errors.
6. It has a scan disk utility that helps to detect, diagnose and repair disk errors.
7. It has ability to bypass some startup commands when the user switches on the computer.
8. The TREE command helps to show graphical representation of all files and directories present in the system or a particular directory or a sub-directory.
9. It has a facility for memory management. We can manage the memory requirement using, the memmaker command. The enhanced load high and device high commands helps in specifying its Memory Region for loading a Program.
10. User can create, remove and change from one directory to another using md, rd and cd commands respectively.
11. It is very easy to copy the files and directories in DOS. Various DOS commands like copy, move, XCOPY are available. With the help of DISK Copy command we can copy one floppy disk to another.
12. The system's setting can be changed by making appropriate changes in CONFIG.SYS file.

13. Using date and time command the user can modify the system's date and time as per his requirements.
14. It gives full information of files, directories and sub directories in the selected files using the command like dir in DOS.
15. With the help of DIR and TYPE command it can display the compute information of all the directories, subdirectories and files respectively.
16. With the help of EDIT, EDLIN, COPY CON commands we can create and modify the contents of the files. With EDIT command we can make partial changes in the file but with COPY CON Command we can change the contents of file on a whole not partially.
17. MS-DOS has MS-DOS help facility, which provides references about any MS-DOS commands. To check the help for any particular command we have to TYPE the Command Name?
18. The BACKUP Command helps to back up the data by restoring the data which has been stored by this command.
19. The UNDELTREE Command helps in recovering all the deleted files.
20. A special POWER programs helps to conserve the battery POWER when all the application and hardware devices are sitting idle.
21. When the computer starts the systems memory is automatically tested by HIMEM extended memory manager facility.
22. Using Pipe symbol (|) helps to execute more than one command simultaneously. Thus it reduces the command's typing time.
23. With the help of the program Microsoft Defragmenter, a POWER facility of DOS, it can optimize file storage on hard disk.

1.4.6 Summary

MS-DOS stands for Microsoft Disk Operating System. Microsoft is the name of the company, which owns DOS. It is a single-user operating system used normally in PC's.

MS-DOS was capable of running almost every software, which was running on CP/M with 8080 chip with as little changes as possible. This happened in August 1981. The MS- DOS version, which was released with IBM-PCs, was known as version 1.0.

There were many limitations to DOS including that you could only do one thing at a time. Also, DOS was a text-based system. The user required to type commands on the prompt.

Many upgrades of DOS were released such as DOS 1.1, DOS 2.0, DOS 2.1 -------- -- DOS 6.x. In every new version more and more facilities were provided.

The DOS operating system is set of a programs, which is, stored on some secondary storage device normally floppy disk or hard disk. It is then loaded into RAM when required. The DOS software is divided into three parts stored in three different files on a disk. The disk, which contains these three files, is called a Bootable disk. The three files loaded on this disk are IO.SYS, MSDOS.SYS and COMMAND.COM. .

Program is of set of various instructions, which are written using any computer language.

File is collection of records, which are logically related. The files, which are supported by DOS are text file, Batch file and application program file.

A file name has two parts i.e primary name and extension. A dot (.) separates a primary name from an extension.

Directory is logical grouping of files. A directory contains the names of the files it holds, along with their extension, the file size, the date of creation or the date of last modification.

The location of directory with respect to other directory is known as path. Paths are of two types absolute and relative.

A system prompt is an indication that the system is ready to take input/instructions from the user and give the output to the user. From the prompt we give the instructions to the computer.

1.4.7 Keywords:

PCDOS	:	Disk Operating System for PC's
Booting	:	To make the system usable
Program	:	Set of instruction given to the computer in any computer language
File	:	Collection of records
Directory	:	Directory is logical grouping of files.
Path	:	The location of the file w.r.t. root directory or current Directory
System prompt	:	Where the DOS commands can be given

1.4.8 Review Questions :

1. Discuss the History of DOS in detail.
2. What are the important files, which are required for loading DOS?
3. What functions does the kernel of DOS perform?
4. Discuss in brief about program, file, naming of file, path and command prompt.
5. Discuss the salient features of DOS.
6. List various file extensions, which are commonly used.

1.4.9 Suggested Readings :

1. “Fundamentals of Information Technology”, by Dr. Hardeep Singh, S.K. Kakar, by Lakahanpal Publication.
2. “Windows Based Computer Courses”, by Gurvinder Singh and Rachpal Singh by Kalyani Publisher.
3. "Operating System Concepts" by Silverschotz – Pearson Education India

1.4.10Web Resources :

www.easydos.com

Post Graduate Diploma in Computer Applications | **Paper : PGDCA-II Operating System**

Lesson No. 1.5 | **Author : Arun Bansal**

Booting Process of DOS

1.5.0 Objective:

In this lesson the students are made familiar with the booting process of DOS. In the beginning structure of MSDOS disk is given and later on the process of booting is discussed in detail. The important files used in the booting process like autoexec.bat, config. sys and batch file programming is discussed in detail.

1.5.1 Structure of MS-DOS Disk

MS-DOS divides the disk into two parts so that the disk is organized effectively. These two parts of the disk are:

I. System Area

It is that area of the disk, which is used by the MS-DOS operating system to keep track of any vital information about the disk. This system area uses very small portion of the disk, which is about 2% of the total disk capacity. This system area is further divided into:

i. Boot area
ii. Fat area
iii. Root Directory

i. Boot Area

The Boot area, also known as Boot record is the first part of a DOS disk. This area stores a program, which is used to do the task of beginning and loading the MS-DOS operating system into the memory of the computer. This process is called Booting process. This process starts the MS-DOS by reading the primary part of DOS program from the disk into the memory. The Booting process can be of two types:

a. **Cold Boot:** Which is switching the power to the computer system, which enables the computer to start up.

b. **Warm Boot:** It is the process adopted, when instead of switching the power on, when the system hangs up, even when the power is on, it is restarted by pressing Alt+Ctrl+Del keys together from the key board it is called warm boot.

ii. FAT Area

FAT stands for File Allocation Table. It is the next part of the system area of the disk. The MS-DOS to manage the big data portion of the disk uses this area of FAT. This area is used to record the status of each area of the disk. For managing the area of the disk, the MS-DOS operating system divides the total area into logical divisions known as clusters. These clusters store the files, which contain data. The FAT table consists of numbers, with one place in the table assigned for each cluster on the disk. The number which is recorded in every cluster's FAT entry indicates the status of the cluster, i.e. is it free or has a file. If there is zero in the cluster's FAT entry, it indicates that the cluster is free and it can be assigned any file. If there is any other number, it indicates that it is in use.

iii. Root Directory

This is the last part of the system area of the disk. This is the file1 directory, which is the basic, built in directory for the disk, which every disk has. The directory records the files that are stored on the disk. This directory has a fixed size for each disk format.

II. DATA Area

It is the bigger part of the disk, where the data files are stored on the disk. Each file on the disk has an entry in the FAT table of the system area. Once a file in the data area is assigned a cluster it finds the entry into FAT table. The data area nearly forms 98% of the disk capacity. The following figure gives the structure of the MS-DOS disk:

Boot Area	FAT Area	Root Directory Area

Fig. 2.1: Structure of DOS Disk

1.5.2 Booting process of DOS

The POST (POWER on Self Test) begins the booting in your PC. When you turn on your PC, the POWER supplied to the PC wakes up the processor and looks for the Start-up instructions. These start-up instructions are stored in the ROM-BIOS (Read only Memory-Basic Input Output System) chip. The ROMBIOS chip actually carrying the instructions begins the POST. The POST begins and the chip double checks, their own code against the same code stored in memory. If everything is okay then the POST proceeds. It then test for all the expansion slots plus the serial and parallel ports, whether they are working right or not. Then it checks the clock chip. Then it gets into video adapter and sees the message of Video BIOS. It also checks for the keyboard port and checks whether any key is pressed or not. Similarly it looks for other devices attached through different ports. This is called a Standard device check.

Then it writes the data to the RAM chips and reads it back, to double check for the accuracy again. The POST keeps going. It then checks for Floppy drive and Hard drive comparing these with the list in CMOS chip. Finally, all system checks are done.

Now actually the booting process starts. It looks for floppy disk in floppy drive and hard drive. If floppy disk is present, it loads the operating system from it to bootup. If it is not present then it looks for it in hard drive.

The ROM start-up routines called Bootstrap loader loads the contents of side 0, track 0 and sector 1 of the first drive in the system bootup sequence. In case of floppy disk, the disk reads the bootstrap program, but in case of hard disk, the ROM loads the boot sector containing the boot parameters and disk bootstrap program from master boot program. The remaining procedure is same for both of them.

The drive contains two hidden files IO.SYS and MSDOS.SYS. These are hidden files and are not visible.

The IO.SYS loads the file MSDOS.SYS. The MSDOS.SYS file then loads the CONFIG.SYS file that configures the system. The MSDOS.SYS also loads the COMMAND.COM. This file basically contains all the standard commands.

The transient portion of the COMMAND.COM file executes the AUTOEXEC.BAT file. If it finds something in this file then it executes that command. If it doesn't find anything then it just displays the DOS prompt on the screen (C:\>). The resident portion of the COMMAND.COM stores all the internal commands.

1.5.3 Rebooting the Computer

Sometimes, it may happen that your system crashes down because of any reason. In such a situation, hit < Ctrl + Alt + Del > keys together. This will restart your computer. This process is called *Warm Boot or Reset.* This will again get things going. But it is advisable to use it as a last alternative when all other efforts of resuming work fail. At times, Warm Boot also fails to restart the computer. This means the instructions that are given by the keyboard are no longer valid and are not interpreted by the system. In this case, switch off the power supply and wait for a few minutes. Then turn on the power supply and start working. Sometimes, due to unknown reasons the system is unable to boot from the hard disk, in such a situation, you are required to boot the system from the bootable floppy/CD that contains DOS.

1.5.4 AUTOEXEC.BAT

Several commands are generally associated with the AUTOEXEC.BAT file as they are usually only executed once immediately after the system has been booted. These will each be discussed in turn and then used in a sample AUTOEXEC.BAT file.

TIME [*hours*:*minutes*[:*seconds*[.*hundredths*]]]

allows the system clock to be set. If no parameters are supplied on the command line the system will display the current time and prompt the user for the correct time. Defaults to midnight.

DATE [*mm-dd-yy*]

allows the system date to be set. Defaults to 1/1/2005. If no parameters are supplied on the command line, the system will display the current date and prompt the user for the correct date.

CLS

Clears the screen.

PROMPT [*text*][$*character*]...

This command lets you change the command prompt. The most common form is possibly pg which represents the working directory of the default drive followed by the '>' character.

BREAK [ON | OFF]

sets the enhanced <*Ctrl-C*> check for user break on or off. Without a parameter returns the current setting of BREAK. The enhancement extends the check that normally includes keyboard I/O to include disk I/O.

VERIFY [ON | OFF]

sets the file write verification mode on or off. Without a parameter returns the current setting of VERIFY.

SET [*string*=[*string*]]

MSDOS maintains an *environment* that is in effect a set of string variables to which string values have been assigned. These string variables can be accessed by executing programs and their corresponding values used to determine the subsequent behaviour of the programs. If the second string is omitted, the string variable is set to the null string. If no parameters are supplied then MSDOS displays all the environment variables and their corresponding values.

PATH [*drive*1:][*path*1][;[*drive2*:][*path*2] ...]

Whenever anything is entered on the command line, the command interpreter interprets the first name on the line as that of an MSDOS command, a batch (BAT) file or an executable (COM or EXE) code file. If it is not an internal MSDOS command, MSDOS will search the working directory for a file with the corresponding filename and the extension: BAT, COM or EXE that order. If none is found a 'Bad command or file name' error message is issued by MSDOS. The PATH command can be used to set up a search path of directories for MSDOS to search sequentially after searching the working directory unsuccessfully.

The following lines illustrate how these commands may be used in a simple AUTOEXEC.BAT file.

```
ECHO OFF
REM *** REM is a batch file command used for comments.
REM *** ECHO is also a batch file command that can be used
REM to either send text to the console screen or to toggle
REM the echoing of command lines executed from the batch file
REM on or off
BREAK ON
VERIFY ON
SET EDITOR=A:\BIN\E.COM
PATH=A:\;A:\BIN;A:\MSDOS
PROMPT=$P$G
DATE
TIME
CD A:\MYFILES
```

1.5.5 CONFIG.SYS

The CONFIG. SYS file is an ASCII text file that contains instructions to DOS regarding your system configuration. It basically resides on the root directory of the default starting drive and is read only once at the startup time.

Since CONFIG. SYS is an ASCII file you can easily edit its contents using a text editor. Since the contents of CONFIG.SYS are read only at startup time so you must reboot your computer to activate any changes you make to this file.

The CONFIG. SYS commands are basically used for increasing disk-read buffers, loading peripheral device drives software, increasing the number of logical drive letters in your system, increasing the maximum number of files allowed to be open etc. Some of common CONFIG. SYS commands that can be used are

1. Buffers 2. Install 3. Device 4. Files
5. Last Drive 6. Shell 7. DOS

All these commands are discussed below.

1. **Buffers:** This command specifies the size of the disk buffer. The disk buffer is the reserved area of RAM for the disk read and write operations. It reduces the time required to access the disk thus resulting in faster read/write operations on the disk. The general setting for buffers is from 1-199. Each buffer takes about 530 bytes of RAM. So, higher buffer size reduces the available memory for the programs.

The syntax for executing this command is

Buffer = ##, n

where ## represents numbers between 1-99 and n represents read-ahead buffers, which store data just beyond the area of the disk being read.

Example:

Buffers = 20

This would allocate 20 blocks of memory for storage of data.

2. **Install:** This command is used to install some executable programs after starting of the computer. This command speeds installation and minimizes addressing problems, but it will not prevent interrupt conflicts between TSR programs and applications.

Example:

A program abcd.exe can be loaded into memory to execute after booting using command

install = abcd.exe ↵

3. **Device**: This command allows us to install any hardware device that is not supported by DOS. It helps to install the device drives in your computer's memory. The syntax for writing this command is

device = <filename>

Example:

device = keyboard.sys	This will install the keyboard.
device = C:\mouse.com	This will install the mouse.

4. **Files:** This command is used to set and specify the number of files that can be opened simultaneously in DOS. By default the number of files are 8, and the maximum number is 255. The specified number of files is the maximum limit that maximum number of files that can be opened. Every time a file is incremented to this parameter, the capacity of RAM decreases by 128 bytes. So you should set the parameter accordingly.

Example:

Files = 20	Here it specifies that 20 files can be opened at maximum simultaneously.

5. **Last Drive:** This command is used to tell how many drives are there in your computer. By default, the last drive is assumed to be E. Any letter uppercase or lowercase can be used to specify last drive. (A-Z) Each additional drive over E takes up 81 bytes of RAM. .

The syntax for writing this command is :

lastdrive = <drivename>

Example:

lastdrive = F	This command informs DOS that drive F is the last drive on the system. Maximum of 6 logical drive letters are set.

6. **Shell:** DOS contains a file 'COMMAND.COM' which is root directory of the hard disk. The shell command helps us to specify another location for COMMAND.COM. For example, you may put COMMAND.COM on RAM disk to increase performance, speed and reduce disk changes on systems without a hard disk.

7. **DOS:** This command is used to load the operating system in the conventional, extended or reserved memory. To execute this, you must load the device driver of HIMEM.SYS using DEVICE command.

The syntax for executing this command is

DOS = higher low, umbl. noumb

Example:

DOS = HIGH, UMB	This command will load DOS in reserved memory with any reminder placed in extended memory.

DOS = LOW, UMB	This command with load DOS into reserved memory with any remainder placed in conventional memory.

1.5.6 Batch File

A batch file is a text file containing a series of commands intended to be executed by the command interpreter. When a batch file is run, the shell program reads the file and executes its command, normally line by line. We know that it is possible to automate the execution of operations or commands that are normally carried out when a PC system is first switched on. This was done by placing the sequence of commands in a batch file with a specific name, namely AUTOEXEC.BAT. This file must also be located in the root directory of the boot volume. The extension 'BAT' is an abbreviation of the term BATCH, which is used to describe a batch or sequence of commands.

As computer systems are very often used in a way that involves the user in the repetition of a sequence of commands, the concept of a 'BAT' file containing the sequence of commands if a very powerful one. Thus whenever we find ourselves entering the same sequence of commands repeatedly we should consider the creation of a batch file to automate the task. An important benefit of this approach, besides ease of use, is of course the increased reliability associated with automation.

Batch File Commands

In the AUTOEXEC.BAT file described at the end of the previous chapter two batch file commands, namely, REM for remarks or comments and ECHO for controlling output to the screen were used. The REM command may be a preferable method of displaying text on the screen if ECHO is ON as the ECHO command and its output will both be displayed. MSDOS 3.3 has remedied this situation by suppressing the echoing of commands prefixed with an '@' character. We will now consider the rest of the available batch processing commands in turn:

PAUSE [*string*]

will display the message *Strike any key when ready ...* when the command is executed. If ECHO is ON then the command and the optional string will of course also be displayed before the message. A more elegant method would be to use the following code sequence:

```
ECHO OFF
.
.
.
ECHO Insert data disk in Drive B
PAUSE
```

If the amount of displayed text is considerable then creating a standalone file containing the text may be better. This file can then subsequently be displayed by using the TYPE command. This method will also be faster.

Replacement parameters

As most human interaction with a computer requires some form of input this is provided for with batch files. The user of the batch file supplies these values to the batch file by including them on the command line immediately after the batch file name. These values are known as *parameters* and they must be delineated from each other by means of spaces.

Within the batch file the parameters are denoted by means of the place holders %1, %2, ... %9. The place holder %0 has the name of the batch file assigned to it by default. If more than 9 parameters are used, they can be accessed by means of the SWITCH command.

Loops

Very often in a batch file you may want to repeat a command several times in succession, each time using a different parameter or value, from a set of parameters or values. We now consider how this can be done with the FOR command.

FOR %%*c* IN *set* DO *command*

This results in the command being repeated as often as there are values in set, using a different value from the set each time in the order in which the values appear in set. The %%c is known as a loop control variable and the ***c*** can be any character other than 0, 1, ..., 9. It will have assigned to it in turn, for successive iterations of the command, the successive values in set. For each iteration command will be executed with each occurrence of %%c in command assigned the corresponding value from set. For example:

```
FOR %%P IN (CHAP1.TXT,CHAP2.TXT,CHAP3.TXT) DO COPY %%p A:
```

Results in the three files being copied from the current or working directory on the default drive to the current or working directory on the volume in the A: drive. The next example is similar except that %%P will in sequence be assigned the values of the command line parameters %1, %2 and %3 extended by the extension '.TXT' rather than specific values.

```
FOR %%P IN (%1.TXT,%2.TXT,%3.TXT) DO COPY %%P A:
```

If this command has a short coming it would have to be that only one command can be repeated directly.

Branching

It may sometimes be necessary to cause the flow of control to deviate from the normal sequence of commands by jumping to some arbitrary point within the sequence of commands within a batch file. This can be achieved by labelling the destination or target command and then transferring control there by means of the GOTO command. A label is defined as the characters following the GOTO and may not contain any separators other than spaces. Labelled lines within a batch file must start with a ':' followed by the label and are ignored during batch processing. These features are illustrated by the following example:

```
: LABEL IN A BATCH FILE
REM ---- EVOKE A COMMAND USING THE FIRST BATCH FILE
PARAMETER.
DEL %1
REM ---- USE SHIFT TO DISCARD THE CURRENT FIRST PARAMETER
REM ---- REPLACING IT WITH THE CURRENT SECOND PARAMETER
SHIFT
GOTO LABEL IN A BATCH FILE TO REPEAT THE DEL COMMAND
```

Using GOTO is considered bad programming practice as it generally alters the *flow of control* through a sequence of statements in an unstructured manner thereby making them difficult to read and *debug* and consequently, *maintain*. During the course this year you will be introduced to a programming methodology known as *structured programming*. Other more powerful operating systems do support the structured programming constructs within their batch /shell/script programming languages.

Conditional Testing and Branching

Very often in a batch file we only want to execute a specific command or command sequence if a certain condition is satisfied or conversely NOT satisfied. This can be achieved quite easily with the IF command.

IF [NOT] *condition command*

Four different types of tests can be performed by the IF command. These include the existence or nonexistence of parameters and files, the equivalence of strings, and a special test involving a variable known as ERRORLEVEL. We will now consider some examples of the use of the IF command.

```
IF %1 == ECHO Are you sure that you want to do this.
IF "%1" == "" ECHO there are no parameters.
IF %1 == "" ECHO The first parameter is a pair of quotes.
IF %1X == X ECHO There are no parameters.
IF NOT EXIST C:\CONFIG.SYS ECHO No CONFIG.SYS found.
IF ERRORLEVEL==1 ECHO Hard disk done for, run while you can.
```

Used with the GOTO command the IF command allows the batch file programmer to implement several different control structures. In this context it also provides a mechanism for terminating the execution of a loop created with the GOTO label command.

1.5.7 Summary

MS-DOS divides the disk into two parts i.e system area and data area so that the disk is organized effectively.

This system area is further divided into Boot area, Fat area, Root Directory. The boot area stores a program, which is used to do the task of beginning and loading the MS-DOS operating system into the memory of the computer. The process of loading MSDOS into memory is called booting. Booting is of two types i.e cold booting and warm booting.

The MS-DOS to manage the big data portion of the disk uses the FAT area. The FAT area is used to record the status of each area of the disk.

The root directory records the files that are stored on the disk. This directory has a fixed size for each disk format.

The data area is the bigger part of the disk, where the data files are stored on the disk.

The three important files required for the purpose of booting are IO.SYS, MSDOS.SYS AND COMMAND.COM.

The IO.SYS loads the file MSDOS.SYS. The MSDOS.SYS file then loads the CONFIG.SYS, file that configures the system. The MSDOS.SYS also loads the COMMAND.COM. This file basically contains all the standard commands.

The transient portion of the COMMAND.COM file executes the AUTOEXEC.BAT file.

The resident portion of the COMMAND.COM stores all the internal commands.

Several commands are generally associated with the AUTOEXEC.BAT file as they are usually only executed once immediately after the system has been booted. These commands are TIME, DATE, CLS, PROMPT, PATH etc.

The CONFIG. SYS file is an ASCII text file that contains instructions to DOS regarding your system configuration.

A batch file is a file with an extension BAT. It contains normal DOS commands and all these commands can be executed by executing a single batch file.

1.5.8 Keywords:

FAT	:	File Allocation Table
Cold Boot	:	If we switch on the power to the system
Warm boot	:	If we restart the system with Alt+Ctrl+Del keys
Root Directory	:	The main directory on any secondary storage
POST	:	Power On Self Test
Autoexec.bat file	:	A batch file which is automatically executed on booting
Config.sys file	:	File which is automatically executed on booting, and required to configure the system.

1.5.9 Review Questions :

1. What do you mean by booting.
2. How warm booting is different from cold booting?
3. Discuss the structure of DOS disk in detail.
4. Discuss the role of Autoexec.bat file in booting process.
5. Discuss the role of Config.Sys file in booting process.
6. What do you mean by Batch file? Discuss various commands used in the batch file.

1.5.10 Suggested Readings:

1. “Fundamentals of Information Technology”, by Dr. Hardeep Singh, S.k. Kakar, by Lakahanpal Publication.
2. “Windows Based Computer Courses”, by Gurvinder Singh and Rachpal Singh by Kalyani Publishers.
3. "Windows for Dummies" by Andy Rathbone, Pustak Mahal.

1.5.11 Web Resources :

ww.easydos.com

Post Graduate Diploma in Computer Applications

Paper : PGDCA-II Operating System

Lesson No. 1.6

Author : Arun Bansal

DOS Internal Commands

1.6.0 Objective

In this lesson the students are made familiar with the use of wild card characters and how to take help from the system in the DOS Operating system. Commonly used internal commands of DOS with their syntax and examples are given in this lesson.

1.6.1 Introduction

The command generally means an instruction written in computer acceptable language that user types to execute a specific operation on the DOS prompt. DOS offers a variety of commands to perform various functions. With the help of DOS commands, you can display the list of files and directories that are present on the disk, create new files and directories; remove unwanted files and directories and much more. DOS commands can be entered either in uppercase or lowercase letters. The format of a DOS command is called *syntax.* All DOS commands begin with command name. When DOS carries out the instructions given by you, it is called the execution of DOS command.

All DOS commands can be classified into two categories: Internal Commands and External Commands.

a) Internal Commands

Internal commands are those commands that are loaded automatically in the memory when DOS is loaded into memory during booting process. These commands are easier to learn and use and require no external files for their storage as in case of external commands. These commands are always available at the console of the user and are executed must faster than external commands. The internal commands are used for common jobs such as copying, erasing and renaming files. Some common internal commands are copy, dir, cd, md, copy con, time, vol etc.

In other words, those commands, which are frequently used and stored in the resident portion of the memory of the computer are called internal commands.

b) External Commands

The external commands are used less frequently and are stored in some external files, which are stored in some secondary storage device. Whenever an external command is to be executed then the external file in which that particular command is stored in transferred from the secondary storage i.e. disk to main memory i.e. RAM.

All executable files with extensions .com, .exe, and .bat are the external commands. These commands are used for relatively complex jobs like comparing two files, formatting disk etc. For using external commands we need to use the corresponding command file available in DOS.

Some of the common external commands are FORMAT, move, attrib, more, chkdsk, TREE , fdisk, Diskcopy, scandisk and defrag etc. These external commands will be discussed in the next lesson.

All the DOS commands have to be given at the system prompt or the command prompt. The commands are listed here in the following structure.

Command : It is the name of the actual command.

Purpose	:	It is the purpose for which the command is used. It gives the objective of the command.
Format	:	It is the way in which we have to give the command to the system.

This is the syntax for giving any command. The syntax has to be followed while using the commands, else the system will not give the wanted response. It is like the grammar of any language, which has to be followed while using that language without fail. The general format/syntax of giving MS-DOS commands is as follows:

Command [d:] [path] [filenames] [/p] [/w]

The symbols used in the general format of the command are to be interpreted as follows:

[]	-	Indicates that the parameter of the Command inside it is optional, which has to be used if necessary else leave it.
d:	-	It indicates the drive. It can be A:, B:, C: etc.
Path	-	It is the route through the directories to access the file. This has to be specified if the file to be used is not in the current directory.
Filename	-	It indicates the full name of the file along with the extension.
/P	-	It indicates pagewise display.
/w	-	It indicates widthwise display.

Finally it should be noted that:

- At least one space should be used to separate commands from their options.
- The command interpreter (or shell), which is responsible for the interpretation of the commands, is not case sensitive. The command interpreter in fact converts all lowercase input to uppercase to simplify the interpretation process.
- Each command is terminated by the ↵ (ENTER) key.

1.6.2 Wild Card Characters

These are special characters, which are used for specific purposes so that some of our operations with commands become more flexible and useful. These characters are used as indicative and space holders for the characters. These are used when we need to replace a place with any of the characters. The following are the most commonly used wild card characters:

i. * [Asterisk]

This is used to represent multiple spaces and is used in conjunction with a command. This special character is useful in many situations like searching, listing etc.

Examples :

a) Dir *.dot

Displays all the files with have .dot as extension irrespective of the file names.

b) Copy a *.* c:\abc

Copies files with file name starting with "a" to the destination directory.

ii. ? [Question mark]

This character is used to represent single space or any character. Example:

a). Dir ?a.*

Displays all the files whose file name has "a" as second character and has any character as first letter.

b). Del ?c

Delete all files with "c" as second character in file name and any character in first space/place.

1.6.3 To Display Help

To get the help of any command help command is used:

C:\>help dir or C:\>command name/?

We can get help for any command in two ways. As shown above the first command tells how to get help for dir command. To view the help of any command, we can also TYPE the **command** at the DOS prompt followed by / and then followed by?

Suppose we want to get the help of dir command then we would write

C:\>dir/?

It would display all the information about the dir command. Besides telling its usage it will also show us the various switches, which can be used along with it.

1.6.4 Internal commands

The commands, which are a part of the main file of DOS i.e.COMMAND.COM, are known as Internal Commands. They are loaded in the RAM as soon as the computer is switched on.

Below we shall be showing various DOS internal commands, which are frequently used.

1.6.4.1 CLS

It is basically used to clear all the information from the display screen, bearing only the system prompt and a cursor on the upper left corner of the screen.

The syntax for writing CLS command

```
C:\>cls ↲
```

It will clear the entire display screen and no previous commands would be seen. It will display operating system prompt, if any, on the first line of the display.

Note: (↲) refers to enter key

1.6.4.2 COPY CON

This is a command basically used to create a file. The only disadvantage of this command is that the file created by this command cannot be modified. If user tries to do it then a message is displayed whether user wants to overwrite already existing file or not. There is no cursor movement upward or backward by using arrow keys when we use this command to make multiline file. This command is not recommended for storing large text because if while typing some error occurs then it would be very difficult to modify.

The general syntax of this command is

```
C:\>copy con filename. extension ↲
```

to save the contents of file press F6 key or Ctrl+z (ˆZ) keys simultaneously.

```
C:\>copy con mayur ↲
Hello  everybody ˆz 1 file(s)        copied C:\>
```

Here a file named 'mayur' is formed in which the contents 'Hello everybody' is saved after pressing (Ctrl+z) keys simultaneously and message ‘1 file (s) copied’ is displayed. You can check the contents of new file formed with the help of 'TYPE' command.

1.6.4.3 COPY

The basic purpose of this command is:

- To copy the contents of one or more specified files to another disk or same disk with same or different file name.
- This command can also be used to copy several files in one stroke by using various wild cards.
- This command can also be used to concatenate two or more files into a single file.

Note: a) The target PATH is optional. At a minimum, the COPY command requires a source file name.

b) The target PATH should not be same as source PATH. In other words, the COPY command will not copy a file onto itself.

The general syntax for copy command is

```
C:\>copy «source PATH» «target PATH » ↲
```

There are a number of switches, which are available with this command:

/A Indicates an ASCII text file.

/B Indicates a binary file.

/V Verifies that new files are written correctly.

/Y Suppresses prompting to confirm you want to overwrite an existing destination file.

/- Y Causes prompting to confirm you want to overwrite an existing destination file.

Example:

C:\>copy C:\mayur C:\varun file This command copies the contents of 'mayur' to contents of file' atuI'

C:\>copy C:\>mayur D:\arun This command copies contents of file 'mayur' in the C drive to contents of file 'arun' on drive D.

C:\>copy mayur+varun arun This command concatenates the two files
'mayur' and 'varun' and stores the result
in 'arun'

C:\>copy A:\ C:\anil This command copies all the contents of
the floppy drive to that of 'anil' directory in C drive.

C:\>copy abc.txt + *.txt This command will copy all files with extension.txt except abc.txt into abc.txt. Basically all files will be copied but contents of one file cannot be copied onto
itself so contents of abc.txt will not be duplicated in the resulting file.

C:\>copy abc.txt+ This command attaches current time and date to a file without changing it.

1.6.4.4 MD OR MKDIR

It is used when we want to make a directory or a subdirectory. If the subdirectory name is preceded by a backslash(\) then it will be created one level below the root directory. If the name is preceded by a space then it will be created one level below the currently logged subdirectory.

On including an existing subdirectory PATH the new subdirectory will be created one level below the indicated PATH. On including a drive letter, the sub directory will be created on the existing drive.

The syntax of md command is:-

md directoryname ↵

Example:

C:\>md arun ↵	This command would make a directory arun at the root.
C:\md varun ↵	This command creates a subdirectory name VARUN one level below the root directory.
C:\>md D:\ABC\DEF↵	This command creates a subdirectory name DEF on level below the ABC subdirectory on drive D, if D:\ABC already exists.

If we want to make a subdirectory in the arun directory then we would have to perform following steps.

```
C:\ cd arun ↵
C:\arun>md varun ↵
C:\arun>cd varun ↵
C:\arun>varun>
```

Thus 'varun' subdirectory would be created in 'arun' directory.

1.6.4.5 CD OR CHDIR

cd or chdir stands for change directory This command is used to find out the location of the file in the directory structure or it can also move us from one location in the directory structure to another.

The general syntax of cd command is

C:\>cd directoryname ↵

Suppose 'ankur' is a subdirectory in the directory 'arun' and we are presently at DOS prompt and want to reach the subdirectory. Then

```
Step 1      C:\>cd arun ↵
Step 2      C:\arun>cd ankur ↵
Finally           C:\arun\ankur >
```

On entering

C:\>cd ↵	It displays currently logged directory/subdirectory.
C:\>cd..↵	It moves one level back to the parent directory
C:\>cd\ ↵	It moves to the root directory directly from any logged subdirectory.

1.6.4.6

PATH

This command tells DOS to look for a program in the specified directories if it cannot find that in the current specified directory. Using this command we can either set a PATH view, see a PATH or we can remove the PATH. The entire PATH command can include up to 127 characters. The PATH command creates a search PATH for program files only. Program files have the extensions .EXE, .COM or .SYS.

Example:

C:\>PATH = C:\DOS ; A:\WS; ↵ This command sets PATH to DOS in C and WS directory at drive A. After entering this PATH command, if you enter any command then DOS first looks for the file in the current directory. If it cannot find that file there then it searches for that in C:\DOS and A:\ WS

C:\>PATH ↵ This command helps the user to know which PATH has been set and if no PATH is set it prints message 'No PATH' is found.

C:\>PATH; ↵ This command clears search-PATH settings and instructs DOS to search only the current directory.

1.6.4.7 POWER

POWER command reduces POWER usage when your system is idle. It is intended for users of battery-operated laptop and portable systems.

The syntax for POWER command is

POWER ADV : status/STD/OFF/switches ↵

The status used with this command following codes:

MAX	Maximum POWER conservation.
REG	Balance POWER conservation with device performance.
MIN	Minimum POWER conservation.

Following parameters can also be used with this command

OFF	Turns off POWER conservation
STD	Uses your computer's internal POWER conservation device.

Examples

C:\>POWERADV: MAX

This command enables maximum POWER conservation.

1.6.4.8 PROMPT

PROMPT is an internal command. This command is used to change the MS-DOS command prompt. The users can on/off the prompt by using this command.

The syntax for writing this command is

prompt [text] where, text specifies the new command prompt.

You can also specify certain special effect characters to the prompt string. These characters are not case-sensitive i.e. they can work in either uppercase or lowercase.

$Q	= (equal sign)	$$	$ (dollar sign)
$T	Current time	$D	Current date
$P	Current drive and PATH	$V	Windows version number
$N	Current drive	$G	> (greater-than sign)
$L	< (less-than sign)	$B	| (pipe)
$H	Backspace (erases previous character)	$E	Escape code (ASCII code 27)
$_	Carriage return and linefeed		

Example:

C:\>prompt ↵	This will change the prompt from off state to that in on
C>	state an display C>
C>prompt pg.↵	This will take us back to the DOS prompt.
C:\>	
C:\>prompt/?	By this we can check the various switches available with this command.
C:\>prompt bansal ↵	This will change the prompt to that of the written text.
bansal>	Thus instead of C:\> we would be getting 'bansal' on display screen as the prompt.
C:\>prompt $d $t	This will set the prompt to the current date and time.
Mon 11/12/2006 14:05>	

1.6.4.9 RD OR RMDIR

It is an internal command. This command is used to remove a particular directory or subdirectory from a disk. Only an empty directory or subdirectory can be removed. If directory is not empty, it is to be emptied first by deleting all the files present in it. Directories that contain hidden or system files may appear empty but cannot be removed until the hidden files are deleted or moved from the directory.

The syntax for writing this command is :

rd directoryname ↵

Here 'directory' can be root directory or subdirectory.

Following things should be ensured before using rd command for removing a directory.

1) Make sure that all files in subdirectory of the directory are deleted.

2) We should be one level above the directory or subdirectory we want to delete.

Example:

C:\>atul\abhi>del *.*
C:\>a1ol\abhi>cd... ↵
C:\>atul>rd abhi ↵
	Suppose we want to remove 'abhi' subdirectory from 'atul' directory, then we should first delete all files from abhi subdirectory first and then delete it. Now abhi subdirectory would be deleted from 'atu1' root directory.

If we have a number of subdirectories present in the directory to be removed then it is better to use deltree command (will be discussed in chapter 4) as it saves time.

1.6.4.10 REN

This command is used to change the names of existing file. If specified file does not exist or name is already assigned to some other file, DOS responds with a message "Duplicate file name or file not found."

The syntax for writing REN commands is :

C:\>REN <old filename> <new filename>

Example:

C:\>REN mayur varun.↵ This command results with renaming of file' mayur' with new name 'varun'.

D:\> REN C:\abc.txt mmm.txt On executing this command the file which is stored in C: drive and having name abc.txt will now be named as mmm.txt and could be stored in C: drive.

C:\>D:\arun\ * .BAK * .OLD It will change the names of all files with the extension BAK on the D:\arun subdirectory so that they now have the extension. OLD.

1.6.4.11 TIME

It displays the current system's time. This command allows us to change the time also. If we don't want to change the time then just write the time command and press enter no change would be made.

The syntax for writing time command is :

C:\>time ↵ This command tells us the current time. You can also enter the new time.

Current time is 11:35:30:38 p

Enter new time:

C:\>time/? The various switches that we can use with this command is shown with the help of this command.

C:\>time/t It displays only the current time and does not prompts you to enter new time.

1.6.4.12 TYPE

This command is used to display the contents of a file on the screen and optionally on printer too. The contents of long files move up so rapidly that it is difficult to read them. Then we use filter command more with it. Wild cards characters are not allowed in the file name.

The syntax for writing TYPE command is :

TYPE filename.extension ↵

Example:

C:\>TYPE abc.txt ↵ This command shows us the contents of the file 'abc.txt'

C:\>TYPE abc.txt : more ↵ This command shows us the contents of the file 'abc.txt' page wise if the contents of the file increase then that of full screen area.

C:\>TYPE abc.txt > prn ↵ we can also send the contents of file 'abc.txt' for printing also

1.6.4.13 VER

VER is an internal command. This command is used to see the current version of DOS. By having the knowledge of version we can come to know which commands would be available at the console of the user.

The syntax for Writing this command is :

C:\>VER ↵

MS-DOS version 6.22

1.6.4.14 VOL

It is used for identification purpose. Apart from a physical label DOS allows us to give an electronic label for each disk. This is called volume label. To see volume label we use VOL command.

The syntax for writing VOL command is :

C:\>VOL <drivename> ↵

Example:

C:\>VOL ↵ It is not necessary to mention the drive name

1.6.4.15 DATE

This command is used to view the system date i.e. date as maintained by the DOS. It is also used to change the current date to that entered by the user. Dates are accepted using the FORMAT mm/dd/yy, where mm is the month (1-12), dd is the day (1-31) and yy is the year from 00 to 99. It is useful in case when our computer has no real time clock.

The syntax for writing this command is

```
                C:\>date ↵
Current date is Mon 11-12-06
Enter new date (mm-dd-yy) :
```

If you don't want to change the date press ENTER.

`C:\>date/t.↵` This command will only display the current date and will not ask for a new date.

1.6.4.16 DEL

It is used to remove a file from a disk. To delete or to erase file from the other drive/directory you need to specify the PATH. It can also remove a group of files by specifying wildcards.

The syntax for writing this command:

DEL [drive:][PATH]filename [/P]

/P Prompts for confirmation before deleting each file.

Example:

`C:\>del a:\sample.txt ↵` This command deletes sample.txt from drive A.

`A:\>del *. * ↵` This command deletes all files of the directory where user is working. Don't use this command on the root directory.

`C:\del/p *. *↵` This command will prompt for confirmation before deleting each file.

1.6.4.17 DIR

It stands for directory listing. It is used to display all the files and sub directories stored in the directory along with their size in bytes, the date and time when they were created.

The syntax for writing this command is

DIR [drive:][PATH][filename]/switches ↵

The various switches that can be used with DIR command are:

/P Pauses after each screenful of information.

/W Uses wide list FORMAT.

/A Displays files with specified attributes.

attributes

D	Directories	R	Read-only files
H	Hidden files	A	Files ready for archiving
S	System files	-	Prefix meaning not

/O List by files in sorted order.

Sortorder

N	By name (alphabetic)	S	By size (smallest first)
E	By extension (alphabetic)	D	By date & time (earliest first)
G	Group directories first	-	Prefix to reverse order
A	By Last Access Date (earliest first)		
/S	Displays files in specified directory and all subdirectories.		
/B	Uses bare FORMAT (no heading inFORMATion or summary).		
/L	Uses lowercase.		
/V	Verbose mode.		
/4	Displays year with 4 digits (ignored if /V also given).		

Example:

C:\>dir filename.extension ↵ It displays the information of specified file.

C:\>dir D: ↵	It displays list of directories and files in drive D.
C:\>dir/o/p ↵	It displays list of directories and files pagewise and orderwise.
C:\>dir *.TC ↵	It displays all the files with extension TC
C:\>dir A*.* ↵	It displays all the files starting from first character A
C::\>dir?? U.FOR ↵	It displays all files have extension FOR and having any first two letters and third letter is U.

If there are very large number of directories and files in your computer system then on using directory command all of them would be shown. It would take lot of time. So to exit that stage we use Ctrl+C buttons.

1.6.5 Summary

The command generally means an instruction written in computer acceptable language that user types to execute a specific operation on the DOS prompt. DOS offers a variety of commands to perform various functions. All DOS commands can be classified into two categories: Internal Commands and External Commands. Internal commands are part of command.com and always available. External commands need separate executable file, which is stored on the secondary storage.

These are special characters, which are used for specific purposes so that some of our operations with commands become more flexible and useful. In DOS two types of Wild card characters are used i.e. *** [Asterisk] and ? [Question mark].**

To view the help of any commands, we can also TYPE the **command** at the DOS prompt followed by / and then followed by?

Internal commands are listed below

Command	**Purpose**
CLS	Clearing the screen display
VOL	Displaying the label of the volume in the drive specified
VER	Displays the version of the MS-DOS
PATH	Sets the path for search
DEL	Deletes the specified file(s)
TYPE	Displays the contents of the file on the screen
MD or MKDIR	Creates a sub Directory
CD or CHDIR	Changing the current directory
DIR	Displaying the Contents of Current directory
REN (or) RENAME	To rename files
PROMPT	To change the prompt
COPY	To copy files
RD (Or) RMDIR	To Remove an empty directory
TIME	To display as well set new time
DATE	To display as well as set new date

1.6.6 Keywords :

Internal command : Command which are part of Command.com

External Command : Command which requires a separate file

1.6.7 Review Questions :

1. What do you mean by DOS internal commands?
2. List various internal commands of DOS.
3. What is the use of path command? Give its syntax.
4. What do you mean by wild card characters? How these are used?
5. Discuss any five internal commands of DOS.

1.6.8 Suggested Readings :

1. “Fundamentals of Information Technology”, by Dr. Hardeep Singh, S.k. Kakar, by Lakahanpal Publication.
2. “Windows Based Computer Courses”, by Gurvinder Singh and Rachpal Singh by Kalyani Publisher.

1.6.9 Web Resources :

www.easydos.com

Post Graduate Diploma in Computer Applications — **Paper : PGDCA-II Operating Systems**

Lesson No. 1.7 — **Author : Arun Bansal**

External Commands of DOS

1.7.0 Objective

In this lesson the students are made familiar with the various DOS External Commands. Three type of commands which are required to copy the contents from source to target disk are compared with each other. Finally the concept of redirection to standard input and standard output, pipes and filters are discussed.

1.7.1 Introduction

As discussed in lesson 3, DOS commands are the instructions, which are used to communicate with the computer system. We call these instructions as commands as we issue commands to the system to get our work done. The commands of DOS are divided into two categories, namely: internal commands and external commands. Internal commands were discussed in detail in the previous chapter. In this lesson we will discuss the commonly used external commands.

1.7.2 External Commands

An external command is an MS-dos command that is not included in command.com. **These are the commands, which have to be loaded from the disk into the memory of the computer before we want to execute them.** These commands are usually in the form of files, which reside on the disk. As and when required they will be loaded into memory before execution. The commonly used external commands of DOS operating system are given below.

1.7.2.1 ATTRIB

ATTRIB.EXE file is needed for its execution. This command is used to set attributes like read only, hidden, archive to a normal file. It displays, sets or removes the attributes. The read only, archive and hidden attributes can be assigned to the file.

The general syntax for writing attrib command is

C: \>attrib PATH name attributes ↵ where PATH name is [drive:] [PATH] [filename]

The various attributes that are available with us while using these commands are :

+	Sets an attribute.
-	Clears an attribute.
R	Read-only file attribute.
A	Archive file attribute.
S	System file attribute.
H	Hidden file attribute.
/S	Processes files in all directories in the specified PATH.

Example:

C:\>attrib ↵	This command will show the current attributes of all the files which are present at the root drive (C:)
C:\> attrib atul +H +R. ↵	This hides and make the file 'ATUL' read only. To unhide it and make it read-write use -H -R switches.
C:\>attrib NN +R ↵	This command makes the file 'NN' read only. 'You cannot modify a read only file' message will be generated when you try to modify this file using EDIT command.
C:\abc\def>attib KKK +H ↵	This command will make the file' KKK' hide only. The name of the file will not be displayed if we use DIR command. The file 'KKK' is present in 'def' subdirectory, which is further present in 'abc' directory.

Note: (↵) refers to enter key

1.7.2.2 BACKUP

BACKUP.EXE file is needed for its execution. This command is used to copy the various files from the hard disk to the floppies to create a back up disk to protect it from crashes and damages that occur due to various reasons.

The general syntax for backup command is

C:\>backup <source PATH>/options<Target PATH> ↵

The various options are:

1. /S to copy all subdirectories to the source path.
2. /M to copy all files of source path which have been modified since last backup.
3. /A to copy files to the target disk without erasing the target disk. Without this option, backup command erase all the files of target disk.
4. /D to copy files saved on or after given date.
5. /T to copy files save on or after given time.
6. /F to format the target disk before doing backup.

1.7. /L Creates a file containing time and date of backup file. Extension of the file created is LOG.

Example :

```
C:\> Backup C:|>CCDept A: /S /L
```

This command will copy all the files with all the subdirectories of the DOS directory to the floppy disk by using /S option. It will also create a log file named Backup.log which contains the information about date and time of backup of each file. The log file is created because of /L option.

1.7.2.3 CHKDSK

CHKDSK.exe file is needed for its execution. This command scans the disk in a specified drive for error. If there is any error, it reports that in the status report. It returns information about volume serial number, total disk space, space in hidden files, space in directories, space available on disk, number of bytes in each allocation unit etc.

The general syntax for writing this command is

CHKDSK [drive:][[PATH]filename] [/F] [/V]

[drive:] [PATH]	Specifies the drive and directory to check.
filename	Specifies the file(s) to check for fragmentation.
/F	Fixes errors on the disk.
/V	Displays the full PATH and name of every file on the disk.

Example:

C:\> chkdsk ↵	Checks the current drive in which you are working and displays information about it.
C:\>chkdsk C: ↵	Result you can see on your screen
C:\>chkdsk D: ↵	Checks the drive D: and displays information about it.

1.7.2.4 DELTREE

DELTREE.exe file is needed for its execution. This command is used to delete all the files and directories in the current directory completely. This command prompts to verify the deletion which can be removed by using the switch 'y' .

The syntax for writing 'DELTREE' command is

DELTREE [/Y] [drive:]PATH ↵

/Y	Suppresses prompting to confirm you want to delete the subdirectory.
[drive:] PATH	Specifies the name of the directory you want to delete.

Example:

C:\>deltree/? ↵	The various switches or options available in deltree can be checked by this command
C:\>deltree a:\arun\*.*↵	This command deletes all the files in arun directory of floppy drive.
C:\>deltree/y	This command will quickly remove unwanted directory, plus all files on it and all subdirectories nested below it. The /y switch suppresses the prompt that ask you to confirm that you want to delete.

The only benefit of this command is that instead of deleting one directory at a time when large numbers of subdirectories are present using del and rd commands, we can delete all in a single command using deltree command.

1.7.2.5 DISKCOMP

DISKCOMP.COM file is required for its execution. This command is used to compare two disks on a track-by-track basis. It checks whether two disks are identical or not. The 'DISKCOMP' will only compare floppy disks that are of the same size and of the same data density.

The syntax for writing this command is

DISKCOMP <source> <target>/switches ↵

If you do not specify a second drive for comparison, DOS uses the current default drive for the second drive.

/1	Compare one side of the disk.
/8	Compare first 8 sectors of disk

Example:

C:\>diskcomp A: B: ↵	If a PC contains two floppy drives of the same TYPE then comparison can be performed by this command
C:\>diskcomp A: A: ↵	If a PC contains only one floppy drive then we specify the same name of the source drive and the second drive. DOS first reads the floppy disk in the source drive, then prompt you to remove the disk and insert the second disk into the same drive.

1.7.2.6 DISKCOPY

DISKCOPY.COM file is needed for its execution. This command is used to copy all the files as well as directories from one disk to another on a track-by-track basis. It is useful in keeping the backup of all files, documents etc. stored in a disk. Diskcopy will only copy floppy disks that are of the same size and capacity. You cannot copy a 5.25" floppy disk onto a 3.5" floppy disk. You also cannot copy hard disks or RAM using this command.

While copying using this command if the target disk is unformatted, 'Diskcopy' will FORMAT the disk first automatically. Disks made using DISKCOPY are track-and sector duplicates of each other. If there is a data error on the source disk, it will also be copied onto the target disk as well. Fragmented files on the source disk will be identically fragmented on the target disk.

The syntax for writing DISKCOPY command is :

DISKCOPY [drivel:] drive2:]/switches ↵

Some of the available switches with this command are:

/1	Copies only the first side of the disk.
/V	Verifies that the information is copied correctly.

/M	Make use of RAM to speed up copy access

You may specify the same drive for drive 1 and drive 2.

Example:

C:\>diskcopy A: A: ↵	If a PC contains only one floppy disk drive then disk copy is performed using this command. Here, DOS first reads the floppy disk in the source drive and then prompts you to remove the disk and insert the target disk into the same drive.
C:\>diskcopy A: B: ↵	If the PC contains two floppy drives of same TYPE then we can use this command.

1.7.2.7 DOSKEY

DOSKEY.COM file is needed for its execution. It is used to recall all previous commands.

The syntax for writing this command is

DOSKEY/options ↵

Before using any of the options first install DOSKEY by writing this command on the DOS prompt and pressing enter, on doing so a message DOSKEY Installed would be displayed on the screen. Once it is invoked, this command saves previously invoked DOS commands in a special buffer in RAM. The user can recall these previous commands by scrolling with the help of arrow keys. It normally uses about 4KB of RAM. This capacity can increase by specifying large buffer size of a command line option.

Some options available with DOSKEY command are:

/ECHO:on/off	Enables/disables echo of macro expansions (default on)
/HISTORY	Displays all commands stored in memory
/INSERT	Inserts new characters into line when typing
/LINE:size	Sets maximum size of line EDIT buffer (default 128)
/MACROS	Displays all DOSKEY macros
/OVERSTRIKE	Overwrites new characters onto line when typing (default)
/REINSTALL	Installs a new copy of DOS KEY

DOSKEY does not effect the standard DOS function keys, but it adds function keys of its own.

Esc	clears current command		
F7	displays command history	Alt+F7	clears command history
F9	selects a command by number	Alt+F1O	clears macro definitions

Example:

C:\>DOSKEY/history	This would display the history of all the commands stored in the memory buffer.
C:\>DOSKEY	Installs DOSKEYs

C:\>DOSKEY/ReinstalllBufsize=2048	Installs an additional copy of DOSKEY, clears command line buffer and enlarges it to 1K.

1.7.2.8 FORMAT

FORMAT.Com file is needed for its execution. This command is used to make a disk usable for operating system by dividing the disk into magnetic tracks and sectors. The data lost after using the FORMAT command can be recovered using unformat command.

The syntax for FORMAT command is :

FORMAT drivename:/switches ↵

Some of the available switches are:

/V [:label]	Specifies the volume label.
/S	To make the target disk boot able.
/F: size	Specifies the size of the floppy disk to FORMAT (such as160, 180,320,360, 720, 1.2, 1.44, 2.88).
/T: tracks	Specifies the number of tracks per disk side.
/N:sectors	Specifies the number of sectors per track.
/I	Formats a single side of a floppy disk.
/8	Formats eight sectors per track.
/C	Tests clusters that are currently marked "bad."

Example:

C:\>format a:↵	This command would simply format a disk in A drive.
C:\>format a:/1↵	This command will format a disk in A drive having single sided single density.
C:\>format a:/S↵	This command will format the disk and make the disk bootable.

1.7.2.9. FDISK

FDISK.EXE file is needed for its execution. This command is used to make partition of the high capacity fixed disk.

It is written as:

C:\>fdisk <drive name> ↵

Or

C:\> FDISK ↵

Example: to prepare hard disk in drive assigned C: use the following procedure

A :\> FDISK C:\ ↵ or A:\? FDISK ↵

It will display 4 options

1. Create DOS partition or logical DOS drive
2. Set active partition
3. Delete partition or logical DOS Drive

4. Display partition information

Further on any options there are more options so this command is for experienced users only because misuse of it leads to loss of data from the disk.

1.7.2.10 FIND

FIND.EXE file is needed for its execution. This command is used to search all occurrences of a specified character string in a specified file. This command can also search for a specified file in a given drive. It can be used with filters also. The character string to be searched must be enclosed in quotation marks. It is case sensitive so uppercase letters are different from lower case characters when used as a search string. If a string contains quotation marks, they should be enclosed in another set of quotation marks.

The syntax of this command is:

find "string" <filename> ↲ OR dir|find "filename" ↲

some of the available switches are :

/V Displays all lines NOT containing the specified string.

/C Displays only the count of lines containing the string.

/N Displays line numbers with the displayed lines.

/I Ignores the case of characters when searching for the string.

Example :

C:\>find "this" abc This command searches the string this in the abc file.

C:\>find/? This command is used to check the number of switches available with find command.

C:\> find "this" def.txt/c This command gives the count of lines in 'def.txt' that contain string this.

1.7.2.11 LABEL

LABEL.EXE file is needed for its execution. This command is used to assign, change or delete a volume label to a disk. The name of the label should not be more than 11 characters. It may include spaces and punctuation marks like! - @ # $ % ~ ' - {}

The syntax for this command is :

label <drive> <label name>↲

Example:

C:\>label A:/sales_rep ↲ This command changes the volume label on diskette drive A to sales_rep. If the diskette did not have any volume label the command creates a volume label sales_rep

C:\>label C: This will cause DOS to prompt you to enter a volume lable for a disk in drive C:'

1.7.2.12 MEM

MEM.EXE file is needed for its execution.. This command is used to display the amount of used and file memory in your system.

The syntax for executing this command is

mem /switches

Some of the available switches are :

/CLASSIFY or /C Classifies programs by memory usage. Lists the size of programs, provides a summary of memory in use, and lists largest memory block available.

/DEBUG or /D Displays status of all modules in memory, internal drivers and other information.

/PAGE or /P Pauses after each screen of information

Example:

C:\>mem This command displays total conventional memory (in bytes), space available to MS- DOS (in bytes), size of largest executable program total contiguous extended memory (in bytes), XMS memory (in bytes)

1.7.2.13 MORE

MORE.Com file is needed for its execution. This command is used to display screen output of a command page by page. It is basically a filter command. This means that either it can be used with some command or used with file to display contents pagewise if they exceed the page size.

The syntax of more command is :

C:\>TYPE <filename>: more ↵ or C:\>more <filename> ↵

Example:

C:\>TYPE abc.txt : more ↵ This command would show the contents of the file abc.txt pagewise.

C:\>more def.txt ↵ This command will also show contents of def.txt pagewise.

C:\>dir: more ↵ It would display list of directories page wise on the screen.

C:\>more<abc.txt It would display contents of abc.txt on the screen, pausing each time the screen is full.

1.7.2.14 MOVE

MOVE.EXE file is needed for its execution. This command is used to move the contents of file from one position (i.e. source) to another (i.e: target directory). In this command the contents of the source are copied to the target and then contents are erased from the source. This is analogous to 'cut and paste'. This command can also be used to rename a directory.

The syntax for writing this command is :

move <source PATH> <target PATH> ↵

The switches used with the move command are

/Y Suppresses prompting to confirm creation of a directory or overwriting of the destination.

/-Y Causes prompting to confirm creation of a directory or overwriting of the destination.

Example:

C:\>move A:\*.* C:\atul This command moves all files from the floppy disk to the atul directory on drive C and removes the contents from floppy disk.

1.7.2.15 MODE

MODE.COM file is needed for its execution. This command performs various functions relating to the transfer of data between the processor, screen, printer and the keyboard. It

1. Sets the parallel printer mode, the view mode, key repetition rates, screen length and width, serial communication protocols.
2. Redirects parallel printer output
3. Shifts screen left or right.
4. Display the status of attached devices

The syntax for executing this command is MODE

Example

C:\>MODE CON This command will display the status of all devices on the system.

C:\>MODE/? This command will display help of mode command.

1.7.2.16 PRINT

PRINT.EXE file is needed for its execution. This command is basically used for printing the files. This command is used to set aside some space in RAM for printing the file in background while you continue working in DOS.

The syntax for writing PRINT command is:-

PRINT <filename> ↵ Example:

C:\>PRINT abc.txt Suppose a file named 'abc.txt' is to be printed so the command used would be

C:\>PRINT C:\abc\ *. txt This command will PRINT all files in the C:\abc subdirectory with the .TXT extension.

Thus, we can also PRINT more than one file using PRINT command.

1.7.2.17 RESTORE

RESTORE.EXE file is needed for its execution. This command is used to RESTORE files from disks made using the BACKUP command.

The syntax for executing this command is

RESTORE source drive: target drive: [PATH] [/switches]

This command requires two parameters, a source drive where the backup files are located and a target drive where the backup files are to be stored.

Some of the switches used with this command are :

/A:mm-dd-yy Forces restoration of only those files that were modified on or after the specified date. Files modified before the specified date are not restored.

/D Displays file name(s) on the backup disk but does not restore them

/M Restores only those files that were modified since the last backup was made.

/N Restore only those files that were deleted since last backup.

Example:

C:\>RESTORE A: C: This command will RESTORE files from the backup disk in drive A: that originally resided on the currently logged subdirectory.

1.7.2.18 SCANDISK

SCANDISK.EXE file is needed for its execution. It is used to repair and check various disk errors, file allocation table (FAT) and other disk related errors. Scandisk retains the record of repairing the disk. It also detects various physical disk errors and surface errors. It is more powerful than chkdsk command.

The syntax for writing scandisk command is :

SCANDISK<drive name>/switches ↵

Example:

C:\>SCANDISK A: ↵ This command checks floppy disk for errors and removes them.

1.7.2.19 SORT

SORT.EXE file is needed for its execution. This command basically belongs to the family of filter commands. This command is used to sort the contents of the file numerically and alphabetically in the ascending or descending order. This command SORT input data and writes the result to the screen to a file or the other device.

Example:

C:\>dir \ SORT ↵ It will display all the files and directories in a sorted order.

C:\>TYPE abc.txt: SORT/ ↵ This command sorts the file 'abc.txt' in the reverse order and also PRINT the result on the screen.

1.7.2.20 SYS

SYS.EXE file is needed for its execution. The purpose of this command is to copy DOS system files to a new disk and make the new disk bootable. This command transfers the DOS system files (IO.SYS and MSDOS.SYS) plus COMMAND.COM to a formatted disk without requiring reformatting. The target drive where SYS is to place the operating system files must be specified. If you do not specify a source location, DOS looks for the files on the current logged drive.

The general syntax for writing this command is

SYS [drivel:][PATH] drive2:

Where drive 1 : represents source drive and drive 2 : represents target drive.

Example :- .

C:\> SYS A:	It will copy system files to disk in drive A:
D:\>SYS C: A:	It will copy system files to drive A : from drive C:

1.7.2.21 TREE

TREE.COM file is needed for its execution. It displays all the directories and subdirectories graphically. The pattern, which it shows, is in a form of a TREE.

The syntax for writing TREE command is :

C:\>TREE<PATH> /switches ↵

The switches, which can be used with 'TREE" command, are

/F	Includes the file names in each subdirectory.
/A	Displays the subdirectory using standard ASCII characters rather than graphic characters.

Example:

C:\>TREE C:\arun ↵	This command shows all the directories and subdirectories that are present in arun root directory.
C:\>TREE | more	It displays the subdirectory structure of the currently logged drive, and pauses the screen displays after each screen of information

1.7.2.22 UNFORMAT

UNFORMAT.COM file is needed for its execution. This command is used to RESTORE the data, which is accidentally erased using format command.

The syntax for writing UNFORMAT command is :

UNFORMAT <PATH>/switches ↵

The available switches are :

/L	List all files and subdirectories found on the formatted drive.
/P	Echoes program messages to the standard printing device
/TEST	Processes, but does not write any changes, to the formatted disk.

Example:

C:\>UNFORMAT A:/L	This command would list files and directories while unformatting the disk.
C:\> UNFORMAT D:	This command will RESTORE an accidentally reformatted disk in drive D:

1.7.2.23 XCOPY

XCOPY.EXE file is needed for its execution. It is used to copy files. It copies files from one place to another on the same disk or different disk. It is more flexible and fast for copying multiple files. XCOPY can create the directories on the target PATH if the destination directory does not already exist. This command is basically used for the purpose of backup.

Note: The XCOPY command will not copy a file onto itself.

The syntax for writing XCOPY command is :

C:/>XCOPY <source> <target> /switches ↵

Some of the available switches are :

/A	Copies files with the archive attribute set, doesn't change the attribute.
/M	Copies files with the archive attribute set, turns off the archive attribute.
/P	Prompts you before creating each destination file.
/S	Copies directories and subdirectories except empty ones.
/E	Copies directories and subdirectories, including empty ones.
/W	Prompts you to press a key before copying.
/F	Displays full source and destination file names while copying.
/L	Displays files that would be copied.
/H	Copies hidden and system files also.
/R	Overwrites read-only files.
/U	Updates the files that already exist in destination.
/-Y	Prompts you before overwriting existing files.

Example:

C:\>XCOPY C:\ABC\*.TXT D:	It copies all files in the C:\ABC subdirectory with the extension .TXT to drive D:
C:\>XCOPY C:\DEF\*.TXT/S/P D:	It copies all files in C:\DEF subdirectory as well as any files located in subdirectories listed below C:\DEF with extension TXT. It prompts before copying to drive D:

1.7.2.24 EXE2BIN

EXE2BIN.COM file is needed for its execution. This is used to convert compiled program with .exe extension to command files. The major requirement of this command is that the resulting program must be smaller than 64K, because the maximum size of a file with .Com extension can not be more than 64K.

The syntax for writing EXE2BIN command is :

C:\>EXE2BIN sourcefile.exe destinationfile.com

Example

C:\>EXE2BIN Prime.exe Prime.com This will convert Prime.exe to Prime.com

Note : if the .exe extension is not given in the source file it is taken for granted and if the .com is not given for the destination file , then it will convert the .exe file to .bin file, which further needs to be renamed with .com extension.

1.7.2.25 DEFRAG

Microsoft defrag, or disk defragmenter, is a software utility designed to help chronologically order the data on the hard disk drive. The theory behind defrag is to place each program together instead of scattered throughout the hard disk drive, allowing the read-write head to access the data faster on the hard disk drive. Windows 98 includes an improved version of defrag that is said to place the most commonly accessed data at the beginning of the drive, allowing the read-write head a shorter distance to access the data, making the computer faster. Generally, unless the hard disk drive is seriously fragmented, you will not notice a significant difference after defragging the hard disk drive.

Defrag was first available in MS-DOS 6.0 and was available ever since.

The syntax for writing DEFRAG command is :

DEFRAG ↲

1.7.3 Differences in COPY, XCOPY and DISKCOPY commands

Sno	**COPY**	**XCOPY**	**DISKCOPY**
1.	It is an internal command.	It is an external command and requires XCOPY .EXE file for its execution	It is also an external command and requires DISKCOPY.COM for its execution
2.	We can use wild card characters to copy multiple files. Copy A:/*.* C:	We can use wildcard characters to copy files in current directory or subdirectories present under it. XCOPY C:\ABC*.TXT/S D:	We cannot use wildcard characters. It is used to copy floppy disks only DISK COPY A: A:
3.	It can copy contents of one disk to another of different size or density	It can also copy disks of different capacities of densities. The Target disk must be large enough to accommodate all source files.	It can copy contents of one The target disk to another of same size or density only.
1.7.	Copy can use	XCOPY can be used to copy	DISKCOPY cannot be

	wildcards characters to copy disk across a network.	disks across a network, which contain files according to specified date also.	used to copy disks across a network.
5.	Copy command is there to copy the specified files onto the target location. It cannot copy a file onto itself.	XCOPY can be used to put together fragmented files on the target drive without calling any attention to some data errors. There will be no loss in the copying speed also	DISKCOPY copies track by track sector to sector starting from track O. Data errors if any on the disks are also copied onto the target disk. It copies label also.

1.7.4 Disk related commands

The commands which are related to disk are known as disk related commands. These are further subdivided into following categories:

a) Disk Creation Commands

The commands such as format, sys, diskcopy, diskcomp, and fdisk are called Disk creation commands. All these are discussed earlier in this lesson.

b) Disk File and Directory Management Commands

The commands such as vol, chkdsk, dir, mkdir, rmdir, cd, copy, copy con, xcopy, del and ren are called Disk file and directory management commands. All these are discussed earlier in this lesson and in the previous lesson.

1.7.5 Redirection of Standard Input and/or Output

We have already mentioned that the MSDOS COPY command can be used to copy data between files, files and devices and devices. There is however another method of doing the same thing by allowing the user to redirect the *standard input* and/or the *standard output* of an executing program. By the standard input and output we mean the *keyboard* and *screen* respectively. This allows the user to specify that the executing program's standard input or output should come from or go to a file or device other than the keyboard or screen respectively.

The reserve symbols used for redirection are '<' and '>' for the redirection of input and output respectively. The composite symbol '>>' is used to redirect standard output such that it is *appended* to the destination or file. The process of redirection is best illustrated by means of the following examples each of which is followed by an explanation:

DIR > PRN redirect the directory listing to the printer

DIR /P would not be a good idea. Why?

TYPE AUTOEXEC.BAT > PRN Send a copy of AUTOEXEC.BAT to the printer.

TREE C: /F > CDRIVE.DIR

The TREE command produces a list of the *full* paths of each directory and subdirectory on the specified drive. The optional /F parameter forces the names of the files of each directory to be displayed as well. All this will be copied into a file called CDRIVE.DIR.

All of the examples above can either be entered interactively from the command line or executed from within a batch file. They also only involve the redirection of standard output and do not expect any standard input when they are executed. While the redirection of standard input is not as useful as the redirection of standard output, it does have its uses, especially with the use of batch files.

As an example, we will consider the creation of a batch file that will produce a directory listing of the current directory prefixed with today's date. To do this we place the commands DATE and DIR into a batch file. The only problem with this solution is that the DATE command expects a response from standard input when it is executed. This response is normally either a new date or a *null* response (the latter is represented by entering <Enter> on its own). If the directory listing is to appear on the screen then the DATE prompt will also appear on the screen and the user can respond appropriately. If however the standard output of both commands is redirected to the printer then that is also where the prompt produced by the DATE command will appear where the user may not notice it. The normal response is the null response (i.e. <Enter>). It will be useful if we can supply this response automatically when it is required by the DATE command. This can be done simply by placing the expected response (i.e. <Enter>) in a file called 'CR' and then redirecting the standard input of the DATE to this CR file. Now the following batch file will execute without requiring any user intervention. It has also been modified so that it accepts as an optional parameter the file to which the dated directory listing should be written. In the absence of this parameter the output defaults to the standard output device, namely the CONsole.

```
@ECHO OFF
REM Dated directory listing
DATE < CR
SET LISTDEV=CON
IF "%1A"=="a" GOTO CONT1
SET LISTDEV=%1
:CONT1
DIR > %LISTDEV%
IF NOT %LISTDEV%==PRN GOTO CONT2
COPY FF PRN
:CONT2
```

1.7.6 Pipes

Given that we can redirect standard I/O it may be useful to redirect the standard output of one program to another program as standard input via an intermediate file, "PIPE' as follows:

```
PROG1 > PIPE
PROG2 < PIPE
DEL PIPE
```

The developers of operating systems have however anticipated this need and have provided a mechanism known as a *pipe* to achieve the result directly without the need for an intermediate file. The three statements shown above can now be replaced by the following statement:

```
PROG1 | PROG2
```

The '|' operator represents the pipe connecting PROG1 to PROG2. Any reasonable number of programs can be linked by means of pipes in this manner. This is a very powerful concept as it allows us to connect several commands, each of which alters its input in a specific manner such that the final output is significantly different from the initial input.

1.7.7 Filters

The possible use of pipes just mentioned is implemented by MSDOS commands that behave as *filters*, i.e. they are designed to specifically alter their standard input in some way. We now take a brief look at the three MSDOS filters before using two of them to improve our DATEDIR batch file.

The first filter that we will examine is the MORE command. This is really a ***passive*** filter that displays its standard input on the screen one screen at a time, pausing between screenfuls until the user either requests the next screen or terminates the command with a BREAK, i.e. <*Ctrl-C*>.

MORE < *source* where *source* is a file or device

source | MORE where *source* is a program's standard output

The next filter sorts its standard input into alphabetical (/R for reversed) order starting from a specific position (nth) within each text record.

[*source*] | SORT [/R] [/+*n*] where source is a program's standard output.

SORT [/R] [/+*n*] < *source* where *source* is a file or device.

Finally we examine the filter FIND that searches a file or files for lines containing instances of string and displays these on the standard output device. /V displays all lines not containing string. /C displays only the number of lines containing string. /N precedes each line with its relative line number within the file.

FIND [/V] [/C] [/N] "*string*" [[*drive*:][*pathname*]] ...

We now use SORT to sort the files displayed in the directory listing of DATEDIR into alphabetical order and FIND to tidy the output up by only listing those lines that do not contain an 'e' or a '.'.

```
@ECHO OFF
REM Dated directory listing
DATE < CR FIND /V "E"
SET LISTDEV=CON
IF "%1A"=="a" GOTO CONT1
SET LISTDEV=%1
:CONT1
DIR | FIND /V . | FIND /V "e" | SORT > %LISTDEV%
IF NOT %LISTDEV%==PRN GOTO CONT2
ECHO ^L > PRN
:CONT2
```

The line, "Enter new date ..." is removed from the listing by piping the output of the date command to the FIND command which in turn lists all the lines which do not contain and "E". The lines corresponding to the current and the parent directories are removed in the same manner. So are all the lines containing a lower case "e" before the remaining lines are sorted prior to being written to the output file. If the destination device is the system printer then the file is copied to the printer followed by a *form feed.*

1.7.8 Summary

External command needs file for their execution. Various external commands and their purpose is given below

Command	**Purpose**
Attrib	Used to hide files or change hidden files to normal files
Backup	Take the backup of files stored on any type of disk
Chkdsk	Scanning the disk and printing related information on screen
Deltree	To delete all files and directories in a single command
Diskcomp	To compare two disks track-by-track
Diskcopy	To copy one disk to another of same capacity
Doskey	To recall all previous commands with the help of arrow keys
Format	To format a disk i.e. to make disk usable in the DOS environment
Find	To find the given text in a file
More	To move the contents from one location to another location
Scandisk	To scan the entire disk for various physical errors

Xcopy	To copy contents from one place to another
Exe2bin	To convert an EXE file to COM file.

The reserve symbols used for redirection are '<' and '>' for the redirection of input and output respectively. The composite symbol '>>' is used to redirect standard output such that it is *appended* to the destination or file.

DOS allows a convenient means of combing multiple commands in a single command. This technique is called piping.

Filters are designed to specifically alter their standard input in some way.

1.7.9 Keywords:

Internal command : Command which is part of Command.com

External Command : Command which requires a separate file and is not contained in Command.Com

1.7.10 Review Questions :

1. What do you mean by external commands? Why these are known as external?
2. Discuss any five external commands giving suitable examples.
3. How COPY command is different from XCOPY.
4. What do you mean by redirection of input and output?
5. What do you mean by Pipes? Discuss their role and importance.
6. What are filters?

1.7.11 Suggested Readings :

1. “Fundamentals of Information Technology”, by Dr. Hardeep Singh, S.k. Kakar, by Lakahanpal Publication.
2. “Windows Based Computer Courses”, by Gurvinder Singh and Rachpal Singh by Kalyani Publisher.