BCA 203 : Operating System

Unit-I

Necessity of Operating system. Operating system terminology, Evolution of Operating Systems (multiprogramming systems, batch systems, timesharing system, Process control and Real-time system). Factors in OS Design (performance protection and security, correctness, maintainability application integration, portability and interoperability).

Device Management : General device characteristics, I/O Programming concepts, device controllers device drivers Interrupts Driven I/O Memory Mapped I/O, Direct Memory Access Buffering, Device Management Scanarios (serial communications, sequentially accessed storage devices, radomly accessed devices).

Chapter

1

Necessity of Operating System

1.1 Definitions and Function

An operating system (OS) is an integrated set of programs that is used to manage the various resources and overall operations of a computer system. It is a program that acts as an intermediary between the 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 in a convenient and efficient manner.

An operating system is an important part of almost every computer system. It is so ubiquitous in computer operations that one hardly realizes its presence most likely you must have already interacted with our or more different operating systems. The names like DOS, UNIX etc. should not be unknown to you. These are the names of very popular operating systems. Like a manager of a company, it is responsible for the smooth and efficient operation of the entire computer system. Moreover, it makes the computer system user friendly. That is, it makes it easier for people to interface with and make use of the computer. It has many names, depending on the manufacturer of the computer like monitor, executive, supervisor, controller and master control programs. Here are some factors, thats why operating system design.

Most operating system perform the following functions :

1. **Processor management :** It is a assignment of processor for different tasks performed by PCs.
2. **Memory Management :** Allocation of main memory and other storage areas to the system programs like user programs & data.
3. **Input/Output Management :** It is a coordination and assignment of tasks between different input/output devices when one or more programs are being executed.
4. **File Management :** That is the storage of files on various storage devices and transfers data. You can modify data in between transfer through text editor.
5. **Job priority :** That is, it determines and maintains the order in which jobs are to be executed in computer system.
6. **Automatic transition :** It transfers control job to job by special control statement.
7. **Interpretation :** It interprets of commands and insteractions.
8. **Coordinator :** coordination & assignment of compilers, assemblers, utility programs & other software to various users of computer system.
9. **Data security :** It keeps program & data in such a manner that they do not interface

with each other. It also provide data integrity to protect itself from being destroyed by any user.

10. **Message generator :** It produce dumps, traces, error messages and other debugging and error detecting aids.
11. **Maintenance :** It maintain internal time clock and log of system usage for all users.
12. **Easy communication :** It facilitates easy communication between the computer system & the computer operator.

1.2 Necessity of Operating system

An operating system is a large collection of software which manager resources of the computer system. It keeps track of the status of each resource and decides who will have a control over computer resources for how long and when. The position of operating system in all over computer system is shown. From the diagram, it is clear that operating system directly controls computer hardware resources, other programs rely on facilities provided by the operating system to gain access to computer system resources. There are two ways one can interact with operating system-

(i) By means of operating system call in a program.

(ii) Directly by means of operating system commands.

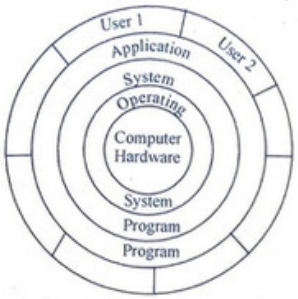


Figure-1.2.-1 : Components of Computer system

1.2.1 System call

System calls provide the interface to a running program and the operating system. User program receives operating system services through the set of system calls. Earlier these calls were available in assembly language instructions but now a days these features are supported through high level language like c, pascal etc., which replaces assembly language for system programming.

1.2.2 Operating system commands

Apart from system calls, users may interact with operating system directly by means of operating system commands. Like If you want to list files or sub-directories in MS-DOS, you invoke **dir** command.

1.3 Evolution of Operating System

Due of the first operating system was developed in the early 1950s for the IBM 701 computers. It was not so powerful like todays operating system. Since then, lot of research work has been carried out in this direction. The main aim of all the researcher was to devise way to minimize the idle time of the computer system and to use the computer system in the most efficient and economical way.

In early days of computers, job to job transition was not automatic. For each and every job to be executed by the computer, the operator had to clear the main memory to remove any data remaining from the previous job, load the program and data of the current job from the input devices, set the appropriate switches and finally run the job to obtain the results from the output devices. After the completion of one job, the same process had to be repeated for the next job by the computer operator. Because of the manual transition from one job to another, lot of time was wasted. In order to reduce this idle time, a method of automatic job to job transition was devised. With this facility, when one job is finished, the system control is automatically transferred back to the operating system for the next job.

But still these was another scope for reducing the idle time of the CPU. The speed of CPU is much more as compared to the speed of Input outputdevices. Hence, the CPU was normally idle while a particular job was busy with some input/output operations. So the next attempt by operating system developers was to over come this speed mis match by executing more than one program at the same time. In this method, while one program was busy with some input/output operation, the CPU time was utilized for processing another job.

In a similar manner, there have been many improvements in the operating system of early days. A modern OS is very sophisticated and does much more than what we have discussed above.

1.3.1 Batch Processing

Batch processing is one of the oldest methods of running programs that is still being employed by many data processing center for processing their jobs. It is a logical step in the evolution of operating system was to automate the sequencing of operations involved in program execution and in the mechanical aspects of program execution and in the mechanical aspects of program development. It is based on the idea of automatic job to job transition facility provided by almost all operating system. Jobs with similar requirements were batched together and run through the computer as a group. For example suppose the operator received one FORTAN program, one COBOL program and another FORTRAN program. If he runs them in that order, he would have to set up for FORTRAN program environment (loading the FORTRAN compiler tapes) then set up COBOL program and finally FORTRAN program again. If the runs the two FORTRAN program as a batch, however he could set up only once for FORTRAN thus saving operator's time.

Batch processing is also known as serial, sequential, off line, or stacked job processing. The method of batch processing reduces the idle time of a computer system because transition from one job to another does not require operator intervention. Moreover, it is the most appropriate method of processing for many types of applications such as payroll or preparation of customer statement where it is not necessary to update information (records) on daily basis. However batch processing suffers from several problems. For example - when a job is stopped, the operator would has to notice that fact by observing the console, determine why the program stopped and then load the card reader or paper tape reader with the next job and restart the computer. During this transition from one job to the next the CPU set idle.

To over come this idle time, a small program called a **resident monitor** was create, which is always resident in the memory. It automatically sequenced one job to another job. Resident monitor acts according to the directives given by a programer through **control cards** which contain informations like marking of job's beginnings and endings, commands for loading and executing programs etc. These commands belongs to **job control language.** These job control language commands are included with user program and data. Here is an example of job control language commands.

$ COB - Execute the COBOL compiler

$ JOB - First card of job

$ END - Las card of a job

$ LOAD - Load program into memory

$ RUN - Execute the user program

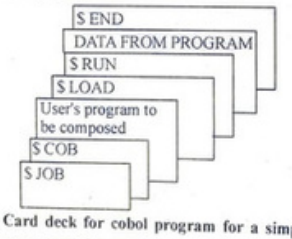


Figure-1.3.1 (a) Card deck for cobol program for a simple batch system

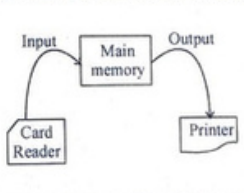
With sequencing of program execution mostly automated by batch operating system, the speed discrepancy between fast CPU and comparatively slow input/output devices such as card readers, printers emerged as a major performance bottleneck. Even a slow CPU works in the microsecond range, with million of instructions per second. But, fast card reader on the other hard, might read 1200 cards per minute. Thus the difference in speed between the CPU and its input/output devices may be three orders of magnitude or more. The relative slowness of input/output devices can mean that CPU is often waiting for input/output.

Over the years of course technology improvement resulted faster input/output devices and CPU speed increased even more faster. Therefore there was a need to increase the thought put and resource utilization by overlapping input/output and processing operations. There was major improvement in this direction in channels peripheral controllers and later dedicated input/output processors. The major development was DMA (Direct memory Access) chip which transfers directly entire block of data from its own buffer to main memory without intervention by CPU. When CPU executing DMA can transfer data between high speed input/output devices and main memory. DMA made interrupt for CPU.

There are two other approaches has to improving system performance by overlapping input, output and processing. They are **Buffering and spooling.**

Buffering is a method of overlapping input, output & processing of a single job. After data has been read and the CPU is about to start operating on it, the input device is instructed to begin the next input immediate CPU & input device are both busy. By the time CPU is ready for the next data item, the input device will have finished reading it. The CPU can then begin processing the newly read data, while input device starts to read the following data. Similarly, this can be done for output. In this case, the CPU creates data that is put into a buffer until an output device can accept it.

If CPU is much faster than an input device, buffering will be little use. If the CPU always faster, then it always finds an empty buffer and have to wait for the input device. For output, the CPU can proceed of full speed until, eventually all system buffers are full. Then CPU must wait for output device. This situation occurs with **input/output bound** jobs where the amount of input/output relation to computation is very high. In this situation, execution is controlled by input/output device, not by speed of CPU. The more sophisticated form of input/output buffering called **SPOOLING** (Simultaneous peripheral operation on line) essentially use the disk as a very large buffer for reading and for storing output files.



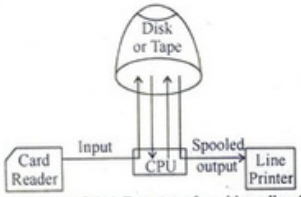


Figure-1.3.1 (b) Data transfer without spooling **Figure-1.3.1(c) Data transfer with sooling facility**

Buffering overlaps input, output and processing of a single job whereas spooling allows CPU to overlap the input of one job with the computation and output of other jobs. This job is better than buffering because spooling provides a very important data structure of job pool. Spooling will generally result in several jobs that have already been read as waiting on disk, ready to run. A pool of jobs on disk allows the O/S to select with job is to be run next in order to increase CPU utilization.

1.3.2 Multiprogramming

Buffering and spooling improve system performance by overlapping the input, output and computation of a single job, but both of them have their limitations. A single user cannot always keep CPU or Input output devices busy at all times.

Multiprogramming offers better efficient approach to increase CPU performance. In order to increase the resource utilization system approach multiprogramming which allows more than one job to utilize CPU time at any moment. Number of programs competing for system resources, better will be resource utilization. Here is an example of main memory of a system that contains more than one program.

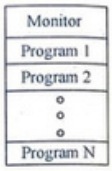


Figure-1.3.2 Primary Memory

The operating system picks one of the program and start executing. During execution process program 1 need some Intput Output operation to complete.

In a sequential execution environment (figure a below), the CPU would sit idle.

In a multiprogramming system, operating system will simply switch over to the next program. (program 2) (figure b)

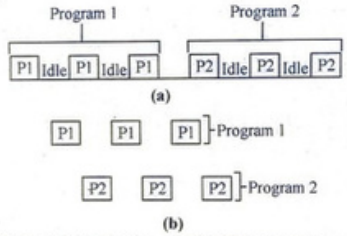


Figure- 1.3.2: (a) Sequential execution (b) Multiprogramming environment execution

Where that program needs to wait for source input/output operation, it suitable over to program 3 and so on. If there is no other new program left in main memory, the CPU will pass its control back to the previous program.

Multiprogramming has traditionally been employed to increase the resource utilization of a computer system and to support multiple simultaneously interactive users (terminals). Compared to operating system which supports only sequential execution, multiprogramming system requires source forms of CPU and memory management strategies.

1.3.2.1. Serial Processing : Programming in 1's and 0's (machine language) was quite common for early computer systems. Programs used to be started by loading the program computer register with the address of the first instruction of program and its result (program) used to be examined by the contents of various register and memory locations of the machine.

Program started being coded into programming language are first changed into object code (binary code) by translator and then automatically gets loaded into memory by a program called **loader.** After transferring a control to the loader to the loaded program, the execution of a program begins and its result gets displayed or printed. Once in memory, the program may be re-run with a different set of input data.

The process of development and preparation of a program in such environment is slow and cumber some due to serial processing and numerous manual processing.

In above flowchart, a typical sequence first the editor is called to create a source code of uses program written in programming language, translator is called to convert a source code into binary code and then finally loader is called to load executable program into main memory for execution. If syntax errore are detected the whole process must be restarted from the beginning. While there was a definite improvement over machine language approach, the serial mode of operation is obviously not very efficient. This result in low utilization of resources.

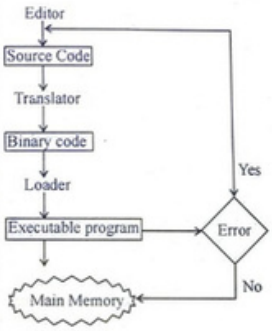


Figure 1.3.2.1 Sequence (serial processing)

1.3.3 Time sharing or Multitasking system

Time sharing or multitasking is a logical extension of multiprogramming. There are multiple jobs are executed by the CPU switching between them, but the switches occurs so frequently that the users way interact with each program while it is running.

In time shared operating system the user share the computer from many users simultaneously. Since each action or command in a time shared system tends to be short, only a little CPU time is needed for each user. As the system switches rapidly from one user to the next, each user is given the impression that it has his own computer, whereas actually one computer is being shared among many users.

The idea of time sharing was demonstrated as early as 1960, but since time shared systems are difficult and expensive to build, they did not become common until the early 1970s. As the popularity of time sharing has grown researches have attempted to merge batch and time shared systems. Many computers systems that were designed as primarily batch system have been modified to create a time sharing subsystem. For example- IBM's OS/360, A BATCH SYSTEM, was modified to support the time sharing option (TSO). At the same time, time sharing systems have often added a batch subsystem. Today, most systems provide both batch and time sharing processing.

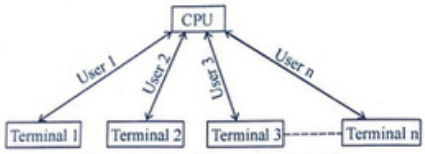


Figure 1.3.3 : Time sharing system

1.3.4 Real time system

Another form of special purpose operating system is the real time system. It is used when there are immediate time requirement on the operation of a processor or the flow of data, and used as a control device in a dedicated application. Examples of such applications are flight control real time simulation military applications.

A primary objective of real time system is to provide quick response times, user convenience and resource utilisation are of secondary concerve to real time system. real time Operating system has well defined, fixed time constraints, Processing must be done within the defined constraints, or the system will fail. It is considered to function correctly only if it returns the correct result within any time containts. It is also used in scientific experiments, medical imaging system, industrial control system automobile engine fual injection system, home appliance controllers and wapon system. A real time operating system (RTOS) is used for the **process control** computer application. It is capable of managing a realtime resource scheduling and control problems in computer based industrial **process control systems.** These process control systems must be able to respond to interrupts from external devices. These interrupts way come from the real time clock input & output devices, the operator terminal etc.

In the real time each process is assigned a certain level of priority according to the relative importance of the evenit process. There are two types of real time systems.

1. A hard real time system, that guarantees that critical tasks complete on time.

2. A soft real time system, where a critical real time taks gets priority over other taks and retains that priority until it completes.

1.4 Types of operating system

In terms of type of O/S we will discuss general properties, types of applications, advantages, disadvantages and basic requirements of different types of operating system.

1.4.1 Batch operating system

Like a batch processing environment it requires grouping of similar jobs which consists of programs data and system commands.

The suitability of this type of processing is in programs with large computation time with no need of user interaction/involvement. some examples of such program include payroll, forcasting statistical analysis and large scientific number crunching programs. User are not required to wait while the job is being processed. They can subuit their programs to operators and return later to collect them. But it has two major disadvantages :

(i) Non-interactive environment : From programers point of view there are source problems with batch system like batch O/S allow little or no interaction between users and executing programs, the turn around time of job subsmission and completion is very high, users have no control over immediate results of a program, It does not create flexibility in software development.

(ii) Off line Debugging : The second disadvantage with this approach is that programs must be debugged which means a programmer cannot correct bugs the moment it occurs.

1.4.2 Multi programming operating system

It is fairly sophisticated then batch operating system. Multiprogramming has a potential for improving system throughput and resource utilisation with very minor differences. Different form of multi programming operating system are-

(i) Multi tasking operating system : A running state of a program is called a process or a task. A multitasking operating system (also called multiprocessing operating system) supports two or more active process simultaneously. It allows then instruction and data from two or more separate process to reside in primary memory simultaneously. Note that multiprogramming implies multi processing or multitasking operation, but multiprocessing operation (or multi tasking) does not imply multiprogramming. Therefore, multitasking operation is one of the machanism that multiprogramming operating system employs in managing the totality of computer related resources like CPU, memory and input output devices.

(ii) Multi user operating system : Multi user operating system allow simultaneous access to a computer system through two or more terminals. Although frequently associated with multiprogramming multi user operating system does not imply multiprogramming or multitasking. A railway reservation system support hundred of terminals under control of a single program is an example of multiuser operating system..Multiprocess operation without multi user support can be found in operating system of some advanced personal computers and in real systems.

(iii) Time sharing system : It is a forms of multiprogramed operating system which operaters in an interactive mode with a quick response time. The user types a request to the computer through a keyboard. The computer process it and a response is displayed on the user's terminal. Most time sharing system user time slice (round robin) scheduling of CPU. In this approach, programs are executed with rotating priority that increases during waiting and drops after the service is granted. In order to prevent a program from monopolising the processor, a program executing longer than the system defined time slice in interrupted by the operating system and placed at the end of the queue of waiting program.

(iv) Real Time System : It is another form of operating system which are used in environments where a large number of event mostly external to computer system, must be accepted and processed

in a short time or within certain deadlines. Higher priority process usually pre-empte execution of lower priority processes. This form of scheduling is called, **Priority based pre emptive scheduling,** is used by a majority of real time systems.

(v) Memory Management : In real time O/S there is little swapping of program between primary & secondary memory. Most of the time, process as remain in primary memory in order to provide quick response, therefore, memory management in real time system is less demanding compared to other types of multi programming system. On the other hand, process in real time system tend to cooperate closely thus providing feature for both protection and sharing of memory.

(vi) input output management : Time critical device management is one of the main characteristics of real time systems. It also provides sophisticated form of interrupt management and input output buffering.

(vii) File Management : The primary objective of file management in real time system is usually the speed of access rather then effcient utilisation of secondary storage. In fact, some embeded real time systems does not have secondary memory. However, where provided file management of real time system must satisfy the same requirment as those found in time sharing and other multi programming systems.

1.4.3 Network operating system

A network operating system is a collection of software and associated protocols that allows a set of autonomous computers which are inter connected by a computer network to be used together in a convenient and cost-effective manner. In it, the user are aware of existence of multiple computersand can log in to remote machines and copy files from one machine to another machine. Some characteristics of network operating systems are-

1. Each computer has its own private operating system instead of running part of a global system wide operating system.

2. Each user normally works on his/her own system; using a different system requires some kind of remote login.

3. 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.

Network operating system offers many capabilities-

(i) Allow users to access the various resource of network hosts.

(ii) Controlling access so that only users in the proper authorisation are allowed to access particular resource.

(iii) Making the use of remote resources appear to be identical to the use of local resources.

(iv) Providing up to the minute network documentation on line.

The system has little or no fault tolerance, if 5% of the personnel computers crash, only 5% of the users are out of business. Network operating system examples are UNIX, NOVell, lantastic, MSLAN manager, windows NT etc. Network operating system can be specialized to serve as peer to peer operating system or as client/Server operating system.

1.4.4. Distributed operating system

A distributed operating system is one that looks to its users like an ordinary contralized operating system but runs on multiple independent CPUs. In a true distributed system, users are not aware of where their programs are being run or where their programe are being run or where their files are residing; they should all be handled automatically and efficiently by the operating system.

Distributed processing is thus, a form of on line processing that allows a single transaction to be composed of application programs that access one or more databases on one or more computer across a network. This type of transaction in which multiple applications running, is called a distributed transaction.

There are variety of reasons for building distributed systems-Resource sharing, computation speed up, reliability, communication.

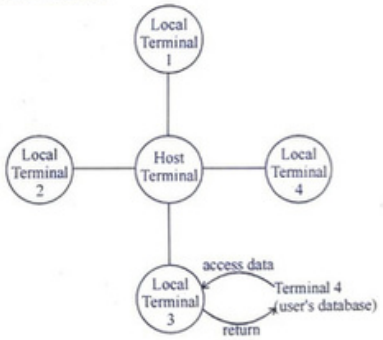


Figure-1.4.4 : Distributed Process

Exercise

Part I (Very Short Answer)

1. Operating system is required to manage ? **[Raj. Univ. 2008]**
2. Distributed system should be for...? **[Raj. Univ. 2008]**
3. Interrupt is generally originated by.....? **[Raj. Univ. 2008]**
4. Operating system is a ..... ? **[Raj. Univ. 2007]**
5. DMA stands for :

Part II (Short Answer)

1. What is the basic aims in time sharing ? **[Raj. Univ. 2007]**
2. Operating system works as a resource manager ? Justify the above statement. **[Raj. Univ. 2011]**
3. What is the difference between kernel mode and user mode ?
4. Explain multiprogramming operating system ?
5. Explain Real Time system.

Part III (Long Answer)

1. What is an operating system ? Explain its functions. **[Raj. Univ. 2005]**
2. What is device management ? Explain the various techniques used for device management. **[Raj. Univ. 2008]**
3. What is distributed system ? Where it is used ? State applications diagram ? **[Raj. Univ. 2010]**
4. Discuss :
   (a) Serial Processing
   (b) Parellel processing
5. What is Batch processing ?

■■■

Chapter

2 Device Management

There are variety of input output devices that can be attached to a computer system. Let us see what are the elements of an input output devices and how they interact with the computer and what are the responsibilities of an operating system. As the two main jobs of a computer are input output and processing, therefore, it becomes essential to know the role of an operating system in managing and controlling the input output operations and input output devices.

Computers operate on many kinds of devices. General types include storage devices (disks tapes), transmission devices (network cards, modem), human interface devices (screen keyboard, mouse), character devices (transferring a single character at a time) or block devices (transferring a chunk of character at a time), multimedia devices (sound card MIDI devices) etc. OS manager device communication via their respective drivers. operating system does the following activities for device management-

- Keep tracks of all devices, program responsible for this task is known as the input output controller.
- Allocates the device in the efficient way.
- De-allocates devices acid allocates the devices.

⇒ On command level, when input output command has been executed and the device has been temporarily released.

⇒ On process level, when process has terminated and the device has been permanently released.

2.1 General Device characteristics

There are some general characteristics of devices, which are following :

- **Character stream/block :** A character-stream device transfers bytes in one by one fashion, where as a block device transfers a complete unit of bytes.
- **Sequential/random access :** A sequential device transfers data in a fixed order determined by the device, random access device can be instructed to seek position to any of the available data storage locations.
- **Synchronous/asynchronous :** A synchronous device performs data transfer with known response time where as an asynchsonous device shows irregular or unpredictable response time.
- **Sharable/dedicated :** A sharable device can be used concurrently by several processes or threads but a dedicated device cannot be used.

- **Speed of operation :** Device speeds may range from a few bytes per second to a few gigabytes per second.
- **Read-write read only or writes only :** Some devices perform both input and output, but others support only one.
- **Efficiency :** Most input output devices are extremely slow compared with the processor and the main memory and buffering is one way to deal with this issue.
- **Generality :** It is desirable to handle all devices in a uniform and consistent manner. It applies for both the way the user processes see the devices, and the way the operating system manager the input output devices and operations.

2.2 Operating system structure

Since operating system is a very large & complex software, supports large number of functions. It should be developed as a collection of several smaller modules with carefully defined input, outputs and functions rather than a single piece of software.

2.2.1 Layered structure Approach

The operating system architecture based on layered approach consists of number of layer (levels), each built on top lower layeres. The bottem layer is the hardware; the highest layer is the user interface.

The first system constructed in this way was the THE system built by E.W. Dijkestra (1968) and his students. The THE system was a simple batch O/S which had 32 K of 27 bit words.

5	User Programs
4	Buffering for I/O devices
3	Devices Driver
2	Memory Manager
1	CPU Scheduling
0	Hardware

Figure 2.2.1 : The layered structure of the operating system

As shown, layer O dealt with hardware, the higher layer. Layer 1 handled allocation of jobs to processor. The next layer implemented memory management. The memory management scheme was virtual memory. Level 3 contains the device driver for the operator's console. By placing it, as well as input output buffering at level 4, above memory management, the device buffers could be placed in virtual memory. The input output buffering was also above the operator's console, so that input output error conditions could be output to the operator's console.

The advantage of layered approach is modularity which helps in debugging & verification of the system easily. The layers are designed in such a way that it uses operation and services only of a layer below it. A higher layer need not know these operations are implemented, only what these operations do. Hence each layer hides implementation details from higher level layers. Any layer can be debugged without any concers. The main difficilty with the layered approach is definition of a new level that is how to differentiate one level from another since a layer can use services of a layer below it, it should be designed carefully like-device driver for secondary memory

must be at a lower level than the memory management routines since memory management requires the ability to use the backing store.

2.2.2 Kernel Approach

Kerval is that part of operating system which directly makes interface with hardware system, its main functions are :

- To provide a machanism for creation and deletion of processes.
- To provide processor scheduling, memory management and input output management.
- To provide mechanism for synchronisation of processor so that processes synchronize their actions.
- To provide machanism for inter process communications.

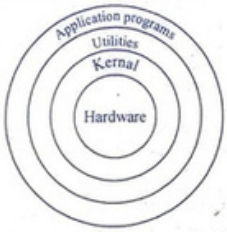


Figure 2.2.2 : "Layered Architecture"

The Linux operating system is based on kernel approach. It consists of two separatable parts : (i) Kernel (ii) system programs.

As shown in diagram, kernel is between system program and hardware. The kernel supports the file system processor scheduling, memory management and other O/S functions through system calls. LINUX O/S supports a large no of system calls for process management and other O/S functions. Through these system calles program utilises the services of O/S (kernel).

2.3 Computer System Operation

2.3.1 Input ouput Concept

In computer system operation basically some input output devices make our modern general purpose computer system. Rather than CPU, device controller, system bus and source specific type of devices executes concurrently competing for memory cycles. Here a description of different parts of computer are-

1. CPU : It is the brain of the omputer. All information works through the CPU to be processed. The CPU can do many types of operations-

(a) It retrieving data from memory.
(b) It gives output data to memory or ports
(c) It performs logical operations such as AND, OR, NOR
(d) can also do mathematical operations.
(e) Comparisone such as equal to, grater than, less than.

(f) Program control operations like jump, skip to next instructions and "if-then" statements.

(g) Respons to special external impulses ("interrupts")

2. Memory : Memory is for the information is stored. Memory is organized in little cubbyholes, each of which needs an address. The various parts of memory is :

(a) RAM : Random access memory stores programs and data as it is used. Information can be written to & read from RAM. Today's computers has 32 to 128 MB RAM.

(b) ROM : Read only memory stores start up and basic operating information can be read from, but not written to ROM. computers generally have 400-500 KB of ROM.

(c) Boot strap Program : It is an initial program stored in ROM which initializes all aspects of the system i.e. CPU registers, device controllers and memory etc. It is responsible for forwading the operating system into memory.

(d) Basic input output system : The BIOS a collection of little programs to run the most basic things in the computer, such as draw a character on the screen, check the memory, read the disk, lead the keyboard etc.

(e) Video memory : It holds the information that is presently being displayed on the monitor. Most computers have 2 to 8 MB of video memory.

3. Disks : Disks are used to store large amount of information even when the power is off. The date stored in disk is stored as regions of changing magnetism an hard disks & floppy disks and as pits in plastic for CDROM's some "heads" move just above the surface of the rapidly spinning disks to read the data.

(a) Floppy disks : Floppy disks comes in two sizes 5.25 inches & 3.5 inches. We can store 360 KB to 1.44 MB. They can be remoned from the computer and the data can be taken to another machine. The drawback is they do not hold very much information.

(b) Hard disks : HD are not removable, it holds more information. HD are available to hold 2 GB to 5 GB of Data. It have read/write capability.

(c) CD ROMs : Compact Disk Read only memory is a read only device, useful for storing huge database or video and audio data. A CD ROM holds about 650 MB data and is removable.

(d) Disk Controller : It is a electronic board in the compute that controls the disk. It tells the disk where to put the heads & what to do (r/w).

4. Input/Output (input output) : Input Output components allow a computer to communicate with the outside world. Every input output components in computer has its own address.

(a) Keyboard : Is used to enter information. from the user side.

(b) Monitor : Are used to display information that the computer tells.

(c) Video controller : It controls the monitor & translate data in video memory.

(d) Parallel/serial parts : Allow the computer to send data and receive data from input output devices.

(e) Mouse/Josticks : Are used to input positional information to the computer.

(f) Network Interface card : A NIC connector the computer to local area network.

5. Bus lines : To move information from one spot to another spot, bus lines are used they are data highways build in computer inside.

(a) Data bus actually carrys the information inside the computer.

(b) Address bus carry the address of where the data is coming from or where it is going.

(c) Control bus carry information as to what to do with data such as read or write or what kind of data it is. They also carry timing info.

When a computer started boot strap program (initial program when computer powered up or reboot) initialings of all aspects of system and finally load the operating system in memory.

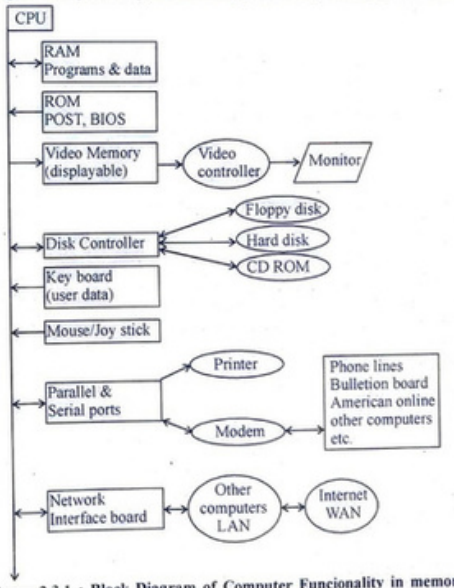


Figure 2.3.1 : Block Diagram of Computer Funcionality in memory

The operating system starts executing with "init" process and waits for the accurrence of some event. The occurance of this event is indicated by an interrupt from either the hardware or the software.

The event can also be signed by the presence of trap. Trap & Interrupt are related terms with a slight difference. **A Trap** (or an exception) is a software generated interrupt caused either by an error (like division by zero, innvalid memory access etc.), or by a specific request for an operating system sevice generated by a user program. Tap is sometimes called exception. Hardware & software can generate these interrupts.

The accurrence of an event is usually signaled by an **Interrupt** from either the hardware or the software. In term of sending a signal to the CPU, a hardware way trigger an interrupt at any time, by way of system bus. Software way trigger an Intrrupt by executing a special operation called a **sytem call** (monitor call).

2.3.2 Input/Output structure

The devie controllers all use a common bus for communication. Each controller control specific types of devices e.g. tape drive, line printer, disk drive etc. A device controller has a local buffer storage and a set of purpose resisters like input register output register control register and status register. The devic controller will be moving data between the peripheral device and its local buffer storage. The variouse steps involved in input output operation-

2.4 Input/Output Interrupts

2.4.1. Asynchronous & synchronouse input output

CPU execution waits while input output proceeds in such case, there is a possibility that at most one input output request is outstand at a time. This is known as synchronous input output. It achieved by polling, when CPU starts an input output operation and continuously polls (checks) that device until the input output operation finishes. Thus synchronous input output keeps CPU idle in the sense that it cannot execute any other process, while an input output operation is going on.

Asynchronous input output is more advantageous as it increaserh system efficiency. Asynchronous input output is the best way to compensate the speed differences between slow input output devices and fast processor.

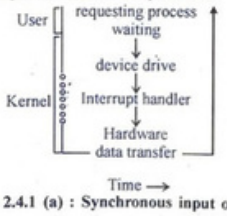


Figure 2.4.1 (a) : Synchronous input output

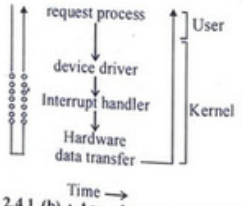


Fig 2.4.1 (b) : Asynchronous Input Output

2.4.2. Direct Memory Access

DMA is another advance mechanism where input output device is given full freedom to transfer

block of data to from memory without going turough the CPU. This is known to increase the speed of overall computer operation. A specific position in memory is used for DMA (e.g. in ISA bus standard, up to 16 MB of memory can be addressed for DMA. The EISA and micro channel architectures standard allow access to the full range of memory addresses.)

- **Necessity of DMA :** If the INPUT OUTPUT operation becomes very fast than CPU faces a situation in which it receives more and more interrupts within a given time interval and thus CPU is left with hardly any time to execute a process.

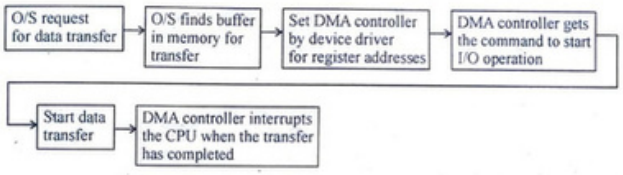


Figure 2.4.2 : input output operations using DMA

DMA is used for high speed input output devices, the device controller transfers an entire block of data directly to or from its local buffer to memory without any intervertion of CPU. Only one interrupt is generated per block rather than one interrupt per byte generated for the law speed devices.

2.5 Storage Structure

Storage structure consists of registers, main memory and magnetic disk is only one of many possible storage system. There are also cache CD ROM, magnetic takes and so on. Each storage system based one storing system and also for holding until it is retrieved at a later time. Differences of many storage device is based on speed, cost, size and volatility.

2.5.1 Main Memory

Main memory and the register built into the processor itself are the only storage that the CPU can access directly. As operating system starts up, it divided the RAM into two broad sections. It reserves for itself, a zone or partition of memory known as the system partition. The system partition always begins at the lowest addressable byte of memory (memory address O) and extends upward. The system partition contains a system leap and a set of global variables.

In below diagram of memory where several applications are open at the same time. The system partition occupies the lowest position in memory. Application partition occupies part of the remaining space application partitions are loaded into the top part of memory first.

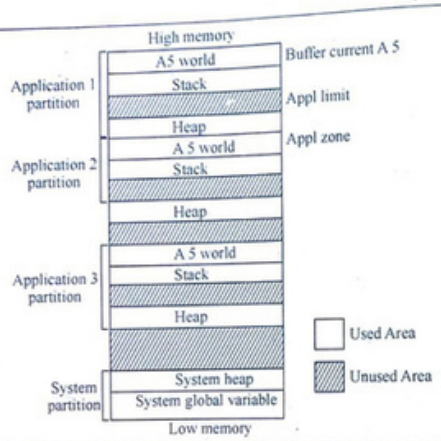


Figure 2.5.1 : Memory Organization when multiple systems are open on the system

2.5.2 Organization of memory in System partition

(a) The system Heap : The main part of the system partition is a area of memory know as the system heap. For exclusive use of operating system the system heap is reserved and for other system software components, which load into its various items such as system resources, system code segments and system data structure. All system buffer and queues, for example are allocated in the system heap.

It is also used for code and other resources that do not belong to specific application, such as code resources that add features to the operating system or that provide control of special purpose peripherals equiment. Device drivers are loaded into the system heap. Most application do not need to load anything in the system heap.

(b) The system Global variable : The lowest part of memory isoccupied by a couection of global variables called system global variables (or low memory). The operating system usesthese variables to maintain different kinds of information about the operating environment. For example the Ticks global variable contains the number of ticks (sixtieths of a second) that have elapsed since the system was most recently.started up similar variables contain, for example, the height of the menubar (M bar height) etc. Most low memory global variables are of this variably : they contain information that is generally useful only to the operating system or other system software components.

Other low memory global variable contain information about the current application. For example the Applzone global variable contains the address of the first bytes of the active applications partition. The Apple Limit global variable contains the address of the last byte the active application's heap can expand to include. The current A5 global variables contains the address of the boundary between the active application's global variables and its application parameters. Because these global variables contain information about the active application, the O/S changes the values of these variables wherever switching from one application to another takes place.

2.5.3 Organization of Memory in are Application Partition

When your application is launched the operating system allocates for it a partition of memory called its application partition. That partition contains required segments of the application's code as well as other data associated with the application. Appplication partition is divided into three major parts :

- The application stack : The stack is an area of memory in your application partition that can grow or shrink at one end while the other end remains fixed. This means that space on the stack is always allocated and released in LIFO (last in first out) order.
- The application heap : An application heap is the area of memory in your application partition in which space is dymanically allocated and relesed on demand.
- The application global variables & A5 world : Our applications global variables are stored in are area of memory near the top of your application partition known as the application A5 world. The A5 world contain 4 kinds of data :

→ application global variables
→ application quick draw global variables
→ application parameters
→ The application's jumps table

2.5.4 Temporary Memory

In a multitasking environment each application is limited to a particular memory partition. The size of your application's partition's partitions places certain limits on the size of your application heap and hence on the size of the buffers and other data structures that your application uses. In general you specify an application partition size that is large enough to hold all the buffers, resoures, and other data that your application is likely to need during its execution.

If for some reason you need more memory than is currently available in your application heap you can ask the O/S to let you use any available memory that is not yet allocated to any other applicatin. This memory, known as temporary memory, is allocated from the available unused RAM; usually that memory is not contiguouse with the memory in your application's zone.

Your application should use temporary memory only for occasional short term purposes that could be accomplished in less space, though perhaps less efficiently.

2.6 Storage Hierarchy

The hierarchy of various kind of memory available in a computer system used for operating system, application programs and other system programs loaded in a system, according to speed and cost. The memory higher levels are expensive, but they are fast. As per Hierarhy, the cost per bit generally decreases, whereas the access time generally increases. There is always a tradeoff between size speed of the memory. As the size grows, the memory becomes slow in its speed. Other concept is storage volatile, main memory is a volatile memory. Its contents are last when power supply to the system is stopped. So the non volatile memory are useful for taking backups.

Caching is a technology used in memory subsystem allow to do tasks more rapidly, when computer is slow down the cache will accelarate the system while keeping the data. It is used for faster but smaller memory type to boost up slower but large memory type (main memory), when we use cache, we must check the cache to see if an item is in there, caches have limited size, cache management is an important design problem. Careful selection of the cache size and of a replacement policy can result in 80 to 99 percent of all accesses being in the cache, greatly increasing performance.

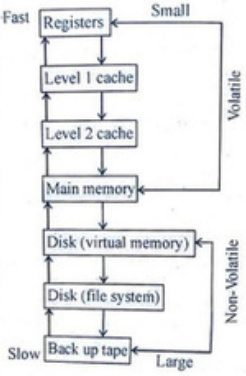


Figure 2.6 : Storage Hierarchy

2.6.1 Coherency of Data

Copy of the some data is maintained by the cache and main memory and may be by the hard disk also. What can happen, if our copy of this data (in cache) is updated and rest remain unchanged ? There is a possibility that stale (not current) data is obtained by an application. Thus, there arises a necessity to keep the data "at all levels", consistent, this is known as cache coherency problem. Cache coherancy is critical for multiprocessing environment. In this case, any change in the data should be immediately notified to the other caches.

2.6.2 Hardware Protection

The operating system gained more & more control over the system with passage of time. With this hold of the operatingy system over the computer system arised the necessity that an operatingy system must ensure that an in correct program do not cause incorrect execution of the other programs. i.e. there should not be possiblity of an erroneous program to modify another program or data of some other program or the operatingy system itself. MS-DOS, macintosh

operating system, both are the examples of operatingy system, Hardware can detect many programming errors. These errors are handled by the operatingy system. In case of an error like trying to jump to an invalid address or executing an illegal instructions, hardware generates trap to the operatingy system. The trap, just like an interrupt, transfer control to the operatingy system. In order for an operating system to serve as an adequate controller of the hardware, the hardware must also provide some kind of support.

2.7 Input output Device Controllers

A general purpose computer system consists of a CPU and a number. of device controller that are connected through a common bus. Each device controller is in charge of a spacific type of device. Depending on the controller, there may be more than one attached device. A device controller maintains some local buffer storage and a set of special purpose registers. The device controller is responsible for moving the data between the peripheral devices that it controls and its local buffer storage. The size of the local buffer within a device controller varies from one controller to another depending on the particular device being controlled. Like the size of the buffer of a disk controller is the same as or a multiple of the size of the smallest addressable portion of a disk, calleda sector, which is usually 512 bytes. A computer system contains a many types of input output devices and their respective controllers.

- Network card
- graphics adaptor
- disk controller
- DVD ROM controller
- serial port
- USB
- Sourd card

2.8 Device Drivers

A device driver is specialized software which specifically written to manage communication with are identified class of devices. For instance a device driver is specially written for a printer with knwon characteristics. Different make of devices may differ in some respect and therefore, shall requirs different drivers. More specifically the devices of different makes may differ is speed, the sizes of buffer and the interface characteristics etc. nevertheless device drivers present a uniform interface to the OS. This is so even while managing to communicate with the devices which have different characterstics.

An application process uses a device by issuing commands and exchanging data with the device driver responsibilities :

- Implement communication APIS that abstract the functiorality of the device.

- Provide device dependent operations to implement functions defined by the API
- API should be similar across different device drivers, reducing the amount of information an application programmer needs to know to use the device.
- Since each device controller is specific to a particular device, the devicedriver implementation will be device specific.
- Provid correct commands to the controller.
- Interpret the controller status register (CSR) correctly.
- Transfer data to and from device controller data registers as required for correct device operations.

2.9 Interrupts driver Input Output

Interrups simplify the software's responsibility for detecting operation completion. Device management is implemented through the interaction of a device driver and interrupts routine. If the CPU always waits for Input Output completion, at most one Input Output request is outstanding at a time. Thus when ever an Input Output interrupt occurs the, Operating System knows exactly which device is interrupting. This approach excluds Input Output operations to several devices and also excludes the possibility of a overlapping useful computation with Input Output. An Input Output device intterpts when it need service.

When an interrupt occurs, the Operatingy system first determines which Input Output device caused the interrupt. If then indexes into the Input Output device table to determine the status of that device and modifies the table entry to reflect the ocurrence of the interrupt. For most devices, an interrupt signals completion of an Input Output request. It there are additional requests waiting in the queue for this device, the O/S starts processing the next request.

Finally, control is returned from the Input Output interrupt. If a process was waiting for this request to complete (as recorded in the device status table), we can now return control to it. Otherwise, we return to whatever we were doing before the Input Output interrupt : to the execution of the user program (the programs started a Input Output operation and that operation has now finished, but the program has not yet waited for the operation to complete) or to the wait loop (the program started two or more Input Output operations and is waiting for a particular one to finish, but the interrupt was from one of the others).

2.10 Memory Mapped Input Output (MMIO)

Memory mapped Input Output and port Input Output (also called isolated Input Output or portmapped Input Output appreviated PMIO) are complementary methods of performing Input Output between the CPU and peripheral devices in a computer. Memory mapped Input Output uses buses of address to address both memory & Input Output devices. The memory & registers of Input Output devices are mapping together with address values. So when CPU accessed address, it may refer to a portion of physical RAM, but it can also refer to memory of the Input Output

device. Thus, the CPU instruction used to access memory can also be used for accessing devices.

Input Output device monitors the CPU's address bus and responds to any CPU access of an address assigned to that device, connecting the data bus to the destination device's hardware register. To hold the Input Output devices, area of the addresses used by the CPU must be reserved for Input Output and must not be available for casual physical memory. The reservation might be temporary.

2.11 Direct Memory access Buffering

DMA is hardware mechanism that allows peripheral components to transfer their Input Output data directly to and from main memory without the need to involve the system processor. It helps to greatly increase throughput to and from a device, because a great deal of computational overhead is eliminated.

Moving data between the controller and the main memory. Recal that (independent of the issue with respect to DMA) the disk controller when processing a read request pulls the desired data from the disk to its over buffer (and pushes data from the buffer to the disk when processing a write).

Without DMA i.e. with Programmed Input Output (PIO), the CPU then does loads and stores (assuming the controller buffer is memory mapped, or uses Input Output instructions if it is not) to copy the data from the buffer to the desired memory locations. DMA controller, instead writes the main memory itself, without intervention of the CPU. So DMA saves CPU work. But it is no important if the CPU is limited by the memory or by system buses. AND there is less data movement with DMA so the buses are used less and all the operation takes less time.

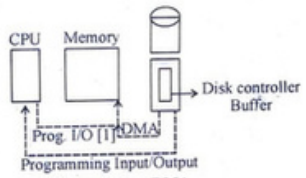


Figure 2.11 : DMA

DMA involves designing of hardware to avoid the CPU perform the transfer of information between the device (controller's data registers) and the memory,

- DMA controller are able to read and write information directly from to primary memory with no software intervertion.
- The Input Output operation has to be initiated by the driver.
- DMA hardware enables the data transfer to be acccomplished without using the CPU at all.
- The DMA controller must include an address register (and address generation hardware) loaded by the driver with a pointer to the relavant memory block; this pointer is used by the DMA hardware to locate the target block in primary memory.

2.12 Device Management Scenarior

Operatingy Systme manager device communication via their respective drivers. As the two main jobs of a computer are Input Output and processing, therefore it becomes essential to know the role of an Operating System in managing and controlling the Input Output operations and Input Output devices.

Devices management functions that must be performed by an O/S are-

- Status of each device
- Use algorithems to decide which process will get a device and for how long.
- Allocate th devices.
- Deallocate the devices at two levels :
- At command level : When Input Output commander has been executed and the devices has been temporarily released.
- At process level : When process has terminated and the device has been permanently released.

2.12.1 Allocating Devices

The devices can be managed and allocated by an Operatingy System in three ways-

(1) Dedicated devices : An Operating system can use a device in a dedicated manner by assigning it to only one process at a time, such devices will remain assigned to only one process at a time and will serve that process till it terminates. There are certain devices that can be managed and used in this manner only. Examples are take drives, plotters etc. These devices demand some kind of allocation Schema as it is difficult for them to be shared. This kind of scheme has some disadvantage.

1. It can be very inefficient scheme especially when device is not used 100 percent of time. System may not always get full utilitization of that device.
2. Another obvious disadvantage of this scheme is that the device cannot be shared among different uses processes. These are sequential accessed storage device.

(2) Shared devices : The Operaingy System can assign a device in such a way that it can be shared among several process like-printers, hard disks etc are of the devices that can be shared. At the same time several process can share their data by interleaved execution of their requests.

The device manager must control this interleaving of process requests. If two requests from two different process arrive at the same time, then such conflicts must be resolved with the help of predetermined policies which can decide that which request should be handled first.

This may be done partially by software (Input Output scheduler and traffic controller) are entirely by hardware (as in some computer with very sophisticated channels and control units). Policy for establishing which process request is to be satisfied first might be based on a priolity list or on the system output (example, by choosing whichever request is nearest to the current positive of the head of the disk). The advantage of this scheme is that the device utilization reaches almost

100 percent if it is shared among several processor and thus kept busy almost at energy instance. These are random accessed devices.

(3) Virtual Devices : Source devices that would normally have to be dedicate (card reader, printers) may be converted into shared devices through techniques such as SPOOLING.

A SPOOLING program can read and copy all card input onto a disk at high speed. Later where a process tries to read a card, the SPOOLING program intercepts the request and converts it to a read from the disk. Since several users may easily share a disk, we can easily convert a dedicated device to a shared device, changing one card reader into many "virtual" ard readers. This technique is equally applicable to a large member of peripheral devices, such as teletypes, printers and most dedicated slow input/output devices.

Exercise

Part I (Very Short Answer)

1. An operating system is a ? **[Raj BCA 2010]**
2. What is the aim of time sharing system ? **[Raj BCA 2011]**
3. Which is a sequential storage device ? **[Raj. BCA 2009]**
4. Interrupt is generally originated by ? **[Raj. BCA 2008]**
5. Which part of an Operating System which parmanently resides in main memory ? **[Raj BCA 2011]**

Part II (Short Answer)

1. Define device driver software and its responsibilities. **[Raj. BCA 2008]**
2. What is Batch Operating System ? Exaplain with suitable example. **[Raj. BCA 2005]**
3. What is real time Operating System ? How does it differ from generally used OS ? **[Raj. BCA 2010]**
4. Differentiate between spooling & Buffering. **[Raj. BCA 2008]**
5. What are the difference between a trap & an interrupt ? **[Raj. BCA 2007]**

Part III (Long Answer)

1. What is an Operating System ? Explain its functions. **[Raj. BCA 2005]**
2. Write short notes on DMA and distributed system. **[Raj. BCA 2006]**
3. What is the main advantage of multi progamming ? **[Raj. BCA 2010]**
4. What is the device management ? Explain the various techniques used for device management. **[Raj. BCA 2008]**
5. What are the main differene between Operating System for mainframe computers and persoual computers ? **[Raj. BCA 20040]**