OPERATING SYSTEM

Written By: VIKAS KUMAR

DEDICATED TO MY PRECEPTOR

“AJIT KUMAR”

- **Operating system (OS):** An operating system is a kind of system software which consists of set of programs. It manages all the computer hardware and also provides a basis for application program and acts as an interface between user and computer hardware.

  An OS provides an operating environment in which a user can easily interface with the computer to execute program. The OS controls and co-ordinates hardware and various application programs for various users.

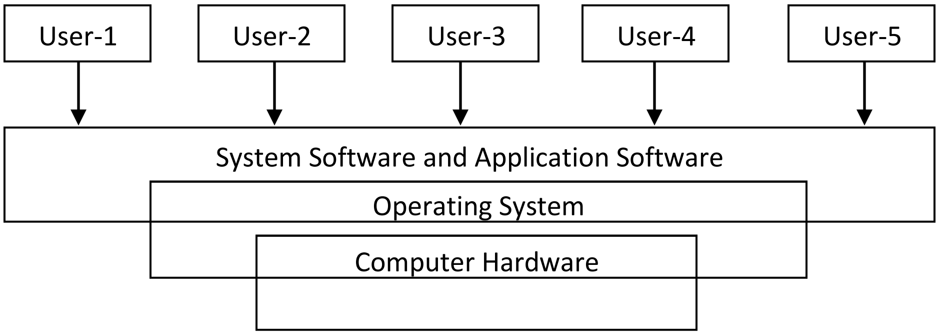


Fig: Abstract view of component of computer

An operating system loads itself into memory and the loading of OS in memory is called “Booting”. The booting process starts the moment computer switched on.

A firmware called bootstrap program stored in ROM is responsible for loading the OS into memory. The moment computer is switched on , the software on memory chip is executed to run the device is called firmware.

There are two-way to interact with OS.

a) By using direct command

b) By using system call

An OS provides the commands the lets the user to communicates with OS. For example Disk Operating System(DOS) provides command like DIR, CLS, RD, MD, CD etc. that directly executed by user. Similarly UNIX/LUNIX OS provides command like clear, cd, mkdir, cp etc. to command direct with OS.

- **System call:** It is a set of functions defined in low level language which act as interface in between application program and OS.

  A computer program request the service from the kernel of OS through system call.

  In other words, system call provides services to application program. It provides and interfaces between a process and OS. All program requiring resources must use system cell.

  There are 5-categories of system cell.

  I. Process control
  II. File management
  III. Device management
  IV. Information management
  V. Communication

❖ Classification of operating system:

An OS in which broadly classified into 2-categories:

a). Single user OS
b). Multiuser OS

- **Single user OS: -** An OS which can handle the request of only one user and can manage one set of I/O device, processor, memory etc. is called single user OS. Single user OS performs the task of only one user at a time and it is specially used to run a stand-alone computer.

- **Subdivision of single user OS:**

  a). Single user Single tasking

  b). Single user multitasking

- **Single user single tasking:** A single user OS which does only one task of a user is called single user single tasking OS. This OS has the ability to load only one program to memory at a time. Ex: MS-DOS (Microsoft disk operating system).

- **Single user multitasking:** A single user OS which performs multiple tasks at a time of a single user is called single user multitasking OS.

  This OS has the ability to load multiple programs into memory at a time.

  When one program waits for some I/O then the OS starts executing another program loaded in memory. This way a user gets illusion that the multiple task being done simultaneously. All GUI OS are single user multitasking OS. Ex: MS-Windows 95/98/XP/vista/07/08/10, MS-Windows NT Workstation, MS-Windows 2003 professional etc.

- **Multiuser OS:** An OS having the ability to handle the request of multiple users is called multiuser operating system.

  A multiuser OS can support more then 1-set of I/O devices , processors memory etc.

  A multiuser OS is also called Network OS because it is used in network environment where multiple computers connected together. A computer network which are multiple OS running is called server. All computers connected to server are called workstation or terminals.

  The multiple user send the request to server and server send back the result to each user after processing.

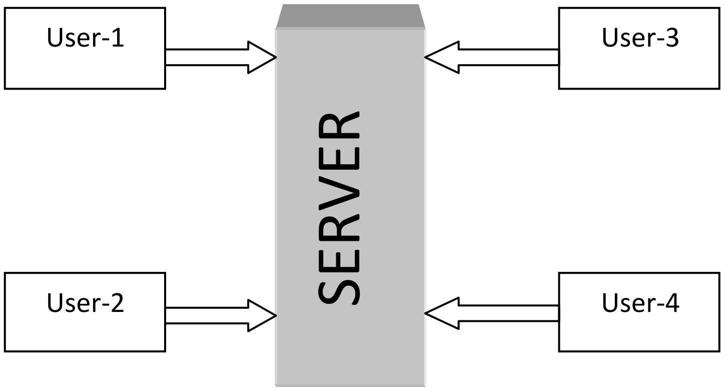


Fig: Multiuser os

Ex:

- MS-windows NT server
- MS-Windows 2000 server
- MS-Windows 2003 server
- Novel Network
- Linux
- Unix

➢ Function of OS:

- CPU management/processor
- Process management
- Memory management
- Disk management
- I/O management

➢ **Evolution of OS:** An OS may process the task serially or concurrently. Based on this concept, there are following 3-evaluation of OS:

a). Serial processing
b). Batch processing
c). Multi-programming

i. **Serial processing:** A serial processing system the instruction and data are entered into computer serially. The process of development and preparation of a program in such an environment is slow and cumbersome due to serial processing and manual processing.

ii. **Batch processing:** In batch processing the utilization of computer resources got improve. In this processing, jobs of similar nature are collected time to time and entered into the computer as a batch/group. The computer then processes these jobs one by one without any user intervention (interrupt).

A small program is called "resident monitor" reside in memory which automatically does the job of sequence from are task to another.

The resident monitor is develop through a language called JCL(Job Control Language).

In batch processing, following technique are use to improve the system performance.

- Buffering
- Spooling
- DMA

✓ **Buffering:** Buffering is a method of overlapping input, output and processing of a single job. After the data having entered in CPU starts processing it. The input device is then instructed to start the next input immediately. This way input devices and CPU are then both busy. This way it improves the system performance.

✓ **Spooling (Simultaneous peripheral operation online):** This technique is more efficient than buffering.

Buffering overlaps I/O and processing of a single job where as spooling allows CPU to overlaps input of one job with computation and output of another job. Therefore, this approach is better than buffering.

✓ **DMA (Direct Memory Access):-** DMA is a memory chip which directly moves the block of data from its own buffer to main memory without intervention by CPU. While CPU is executing, DMA can transfer data between high speed I/O device and main memory. This way it increases the throughput and system performance.

- Throughput:- The amount of work completed in there unit time.

iii. **Multiprogramming:** Multiprogramming offers more efficient approach to increase the system performance by keeping CPU or I/O device busy all the time.

Multiprogramming approach allows to loads more than one programs (jobs) into memory at a time.

An OS picks one program from memory and starts executing it. When the current executing program waits for some I/O then the CPU switch over to another program for execution. This way it kept the CPU busy all the time

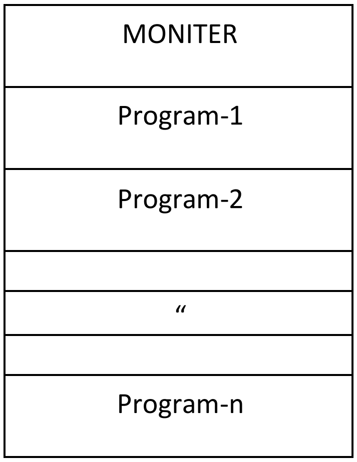


Fig: Memory layout in Multi-programming

- **Operating system architecture:-** An OS is a large and complex software which supports large no of functions. Therefore, it is developed as a collection of modules where each module performs a particular task. The design of OS is refers to as OS architecture. OS architectures are:-

  a) Layered structure approach

  b) Kernel approach

c) Virtual machine

d) Client server model

- **Layered structure approach:** An OS architecture based on layered approach consist of no. of layers(Level) where each layer built on the top of lower layers.

  The bottom layer is the hardware and the highest layer is user interface. The first OS designed on layered approach "THE" operating system which consist 6-layers as shown bellow:-

  The bottom layer-0 deals with hardware. The layer-1 handles the allocation of jobs to CPU. The layer-2 handles the task in memory management such as: creating virtual memory, swapped in, swapped out etc.

Layer-5	User program
Layer-4	Buffering for I/O device
Layer-3	Device driver
Layer-2	Memory management
Layer-1	CPU scheduling
Layer-0	Hardware

Fig: Layered structure of THE OS

  The layer-3 is responsible for handling and running a specific device connected to system. And the next layer-4 does the buffering the I/O device the highest layer-5 user program providing interface to user communicated with system.

  The layers are designed in such a way that it uses operation and services of layer below it. The main advantage of layer approach is the modularity that helps in debugging and verification of system easily. The

Disadvantage of layer approach is the definition of new layers.

- **Kernel approach:** In this approach an OS is basically divided into 2-parts: kernel and shell.

Kernel is a part of OS which directly communicates with hardware. The kernel performs the following function:

I. To provides a mechanism for creating and deleting of process.

II. To provides processor scheduling, memory management and I/O management.

III. To provides mechanism for synchronization of process.

IV. To provide mechanism for inter-process communication.

The shell is a part of OS which act as interface between user and kernel. It accepts the command for user and convey to kernel for execution.

UNIX operating system is designed on kernel approach as shown bellow:-

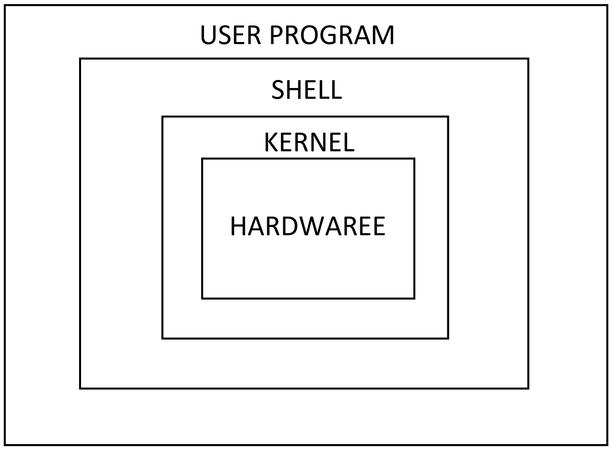


Fig: UNIX operating system

- **Virtual machine:**- Virtual machine is a concept which creates illusion of a real machine. It is created by virtual machine OS which makes a single real machine to appear as several machine in following figure.

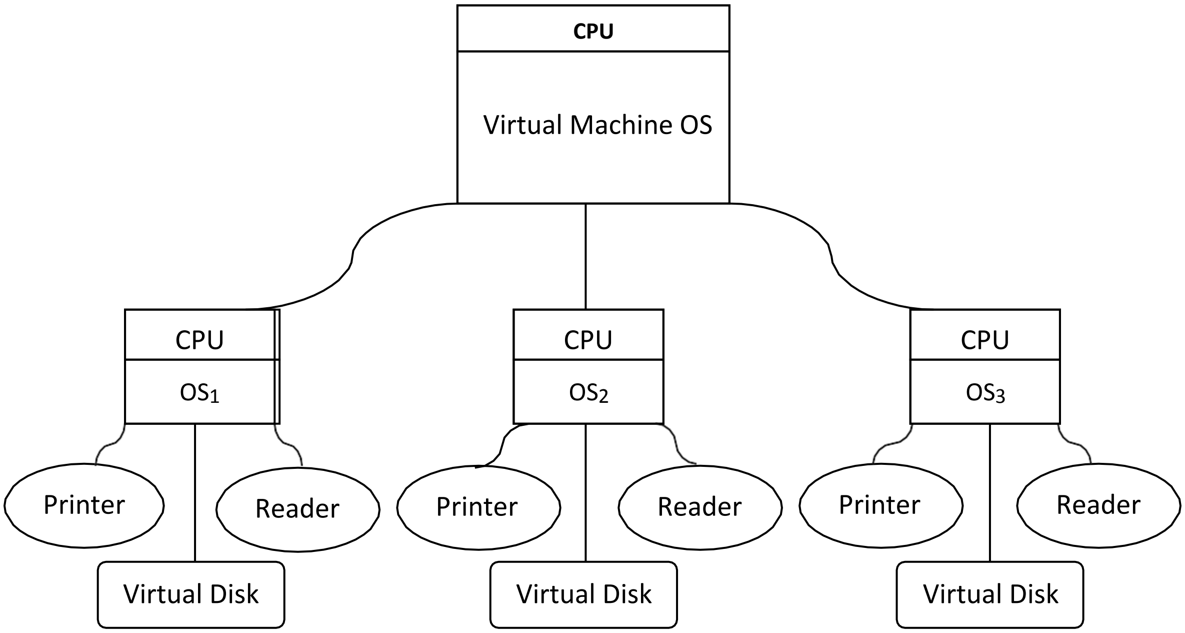


Fig: Virtual Machine

From user point of view virtual machine can be made to appear similar to a real machine can be internally different. Another important accepts is that each user can run OS of his own choice.

- **Client/Server model:** This is the commonly used architecture of OS. In this architecture, an OS is divided into two parts as : "client process" and "server process".

The client process sets the request to server process and the server process sends back the result to client process.

In this model kernel handles the communication between client and server and also manages memory service, file service Terminal service, memory service etc.

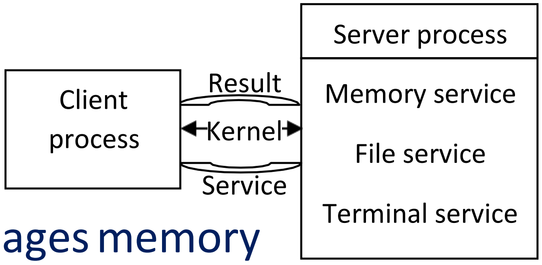


Fig: Client/ Server Model

➢ Type of Operating System:

1) Batch Process:- An OS which support the batch process environment is called batch OS.

The similar jobs are grouped and entered into system.

There are some disadvantages of batch OS:-

a. The time between job submission and job completion is very high. In other word, the output is delayed.
b. The programmer can't correct the error the moment it occurs.
c. The jobs are processed in the order of submission.

2) Multiprogramming OS:- Multiprogramming OS is more sophisticated than batch OS. The multiprogramming OS has a potential to improve system throughput by proper utilization of resources.

Various form of multiprogramming OS are:-

a) Multitasking OS
b) Multiprocessing OS

c) Multiuser OS/Network OS
d) Time sharing OS
e) Real time OS

I. **Multitasking OS:**- It is a form of multiprogramming OS that performs multiple task simultaneously. It allows to reside multiple programs into memory. When one program waits for some I/O operation, an OS submits another program to CPU for execution. This way, a user gets illusion that his/her multiple tasks are being completed. Hence, it is also called serial multitasking and it is different from multiprocessing.

II. **Multiprocessing OS:-** It is a type of multiprogramming OS which supports parallel processing where more than one processes are executed concurrently by the multiple computational unit (ALU). Hence, multiprocessing OS is very fast as compared to multitasking OS.
Multitasking OS is based on single CPU whereas multiprocessing is based on multiple CPU. This type of OS is used in fast and complex system like whether forecasting, email processing, export system, artificial intelligence etc.

III. **Multiuser OS/Network OS:**- It is type of multiprogramming OS which has the ability to handle the request of multiple users. It allows simultaneous access to a computer system called server through two or more terminals. It is used in network environment and also called Network Operating System (NOS). A network OS provides many capability like:

a) Allow user to access the various resources of the network host.
b) Controlling access so that only authorized user can access the network resources.
c) Provides up to the minute network documentation online.

IV. **Time sharing OS:** In the time sharing OS allows many user to use a particular computer system at the same time.

The CPU time is shared among multiple users. It is a form of multiprogramming OS in which CPU switches rapidly from one user to another user in a very small fraction of time (Time slot) and this way each user is given the impression that his task only is being processed by the computer.

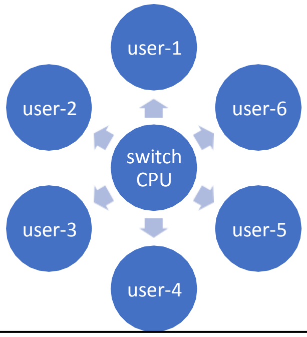


Fig: Time sharing OS

The main advantage of time sharing OS is that if reduce the CPU idle time and provides quick response.

V. **Real time OS:** A real time system is defined as a data processing system in which the data processing and response time is very fast. The time taken to respond and input and display of output is called response time.

The response time is very less as compare to online processing.

It is used where fast response is needed. for example, industrial control system, scientific experiment etc.

The major drawback of real time system is that there is a problem of data security and low volume of data processing.

3) **Distributed Operating System:-** The distributed OS use multiple CPU for executing users program. The data processing jobs are distributed among the processor.

The use of multiple processors is invisible to user i.e. the user view the system as a uni-processor but not as a collection of different machine.

The user's data may get processed on any CPU but user are not aware of where programmer are being run or where there files are store.

The distributed system is considered to be more reliable then uni-processor based on system. Another advantage of distributed system is the incremental growth means more processor added will give the more processing power.

❖ **Process and process scheduling :-** A program in running state is called process. In other words, a program is in passive state whereas process is an active state

- **Process hierarchy:-** An OS creates and kill processes. When a process is created, it creates another process which intern creates some more processes and so on and this form process tree or hierarchy.
- **Process states:-** The life time of process is divided into various stages called states. In other words, the state of a processes changes during its execution, Each process may be in one of the following states:
  a. **New:-** The process has been created
  b. Ready: The process waiting for the allocation to a CPU (processor) for execution.
  c. Running state:- The process being executed by CPU is called said to be running state.
  d. Suspended/Waiting state: The process waiting for some I/O and its execution is temporarily paused/stops is said to be in suspended/waiting states.
  e. Terminate: A process is said to be in terminate state when its execution is finished/completed.

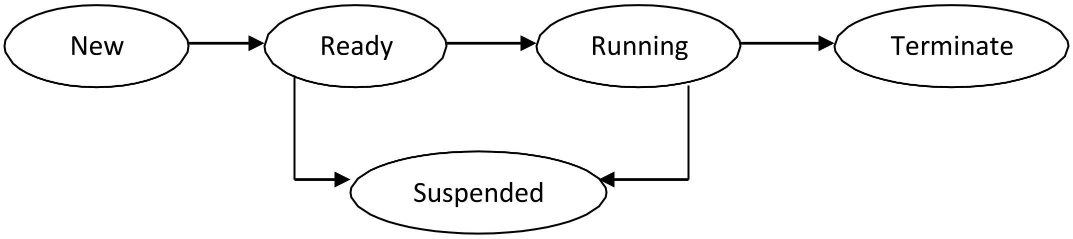


Fig: Process state

/*Process Control Block(PCB*/

An OS groups all information that it needs about a particular process in a data structure called Process

Control Block (PCB). When a process is created, the OS create a corresponding PCB and when the process terminated, the PCB also get released from memory.

A PCB contains the following piece of information about a particular process.

- ✓ Process state:- Each process may be in states like new, ready, running, waiting, terminate.
- ✓ Process number:- Each process is identified by a unique number called process id.
- ✓ Program counter(PC): It indicates the address of instruction to be executed next.
- ✓ I/O- status information: It includes the information about the I/O device allocated to a process.

/*Process Scheduling*/

Scheduling refers to the set of policies and mechanism supported by OS that determines the order in which the process/jobs/task will be completed. It is one of the main functions of an OS. All computer resources are scheduled before use.

A part of OS (Operating System) which performs job of scheduling is called scheduler. A scheduler selects an available process from set of processes for execution by CPU. i.e. the selection of the process is carried out by the scheduler. The main objective of scheduling is to increase the CPU utilization and improve the overall efficiency of computer.

- **Type of scheduler:** There are 3-type of scheduler
  - i. Long term scheduler
  - ii. Medium term of scheduler
  - iii. Sort term scheduler

1. **Long term scheduler:-** It is also called job scheduler. The long term scheduler selects the process from mass storage device and loads them into memory in a ready queue.

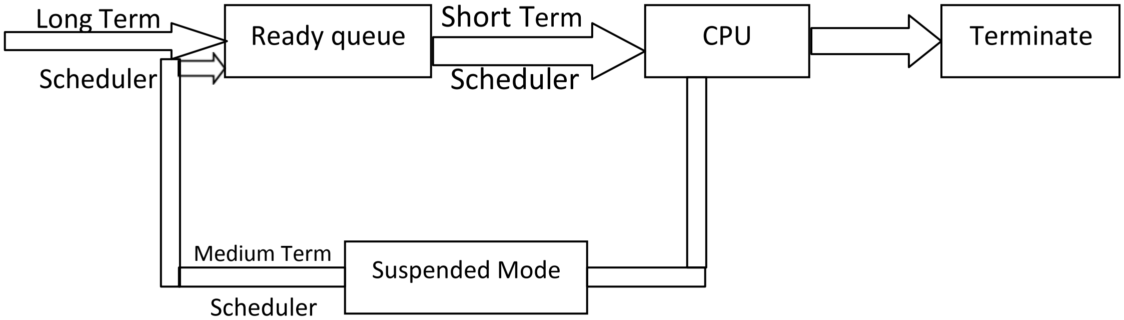


Fig: Different scheduler task

2. **Medium term scheduler: -** The medium term scheduler is a scheduler which makes the suspended process to ready state by loading into ready queue once the suspended criteria is fulfilled.
3. **Short term scheduler:-** It is also called CPU scheduler or dispatcher. It selects the process from ready queue and admits to CPU for immediate processing.

- **Various type of scheduling algorithm:-**
  - a. First-Come First Serve(FCFS) Scheduling
  - b. Shortest-Job-First(SJF) Scheduling
  - c. Round-Robin(RR) Scheduling
  - d. Priority Based scheduling

e. Multi-Level Queue(MLQ) Scheduling

1. **FCFS Scheduling:-** It is non pre-emptive type of scheduling algorithm.

   This scheduling algorithm processes the job in the order of their arrival i.e. the job which arrive first will be processed/ executed first. It is based on FIFO (First-In-First-Out) concept. A job has longer processing time then another job has wait for longer. Hence, it has poor performance.

Q1. Find the waiting time and turnaround time for the following process under FCFS Scheduling.

Process	Execution time
P1	5
P2	7
P3	9
P4	4

The process arrives in the order as $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4$

➢

Process	Waiting time	Turnaround time
P1	0	5
P2	5	12
P3	12	21
P4	21	25

Avg. Waiting Time= (0+5+12+21)/4
=38/4
=9.5 unit

2. **SJF-Scheduling:-** It is a non pre-emptive type of scheduling algorithm. In this scheduling algorithm, jobs are processed based on the shortest executing time. The job with equal execution time will be processed based on FCFS algorithm.

Q1: Find the waiting time and turnaround time for following process under SJF.

Process	Execution time
P1	5
P2	7
P3	9
P4	4

The process arrives in the order as P_1→P_2→P_3→P_4 compare FCFS and SJF Who is best.

➢

Process	Waiting time	Turnaround time
P1	4	9
P2	9	16
P3	16	25
P4	0	4

Avg. waiting time= 29/4

=7.25 unit

Conclusion:- Since, Avg. waiting of SJF is less than the avg. time in FCFS and therefore, SJF is more efficient.

3. **Round Robin(RR) Scheduling:-** It is also called Context Switching or Time Sharing scheduling.

It is a pre-emptive type of scheduling algorithm. The CPU time is divided into time slices. Each process is allocated small and equal time slices and CPU switched from one process to another process in the given time slice.

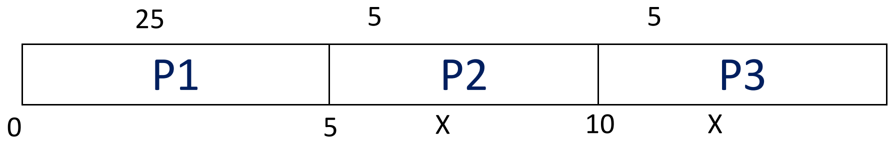
If a process requires more time slices then its waits for next time slice. In this scheduling algorithm each user gets impression that his/her only being processed.

Q1: Calculate waiting time and turnaround time for the following process under RR scheduling (Time Slice: 5unit)

Process	Execution Time
P1	25
P2	5
P3	5

Arrival order: P1→P2→P3

➢



Process	Waiting Time	Turn around Time
P1	10	35
P2	5	10
P3	10	15

Avg. Waiting Time=25/3

=8.33 Unit

4. **Priority Based Scheduling:-** It is non-preemptive type of scheduling algorithm. In this algorithm each-process is assign a priority (A number indicating level/precedence). A

process is allocated to CPU for execution based on the priority i.e. a process with higher priority is allocated to CPU before a process will lower priority.

The process having same priority is executed on FCFS basis.

The process having same priority is executed on FCFS basis.

The major drawback of priority based scheduling is the infinite block of a low priority process this is called starvation. A process is ready to run and waiting for CPU but in the min time a process with higher priority comes in ready queue then waiting process will not be submitted to CPU. This infinite blocking of process of low priority is solved through aging priority.

Aging priority is a technique of increasing the priority of process that was it in the system for long time.

Q1: Calculated waiting time and turnaround time for the following process under priority based scheduling.

Process	Execution Time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	2	5
P5	5	2

Process arrival order: P1→P2→P3→P4→P5

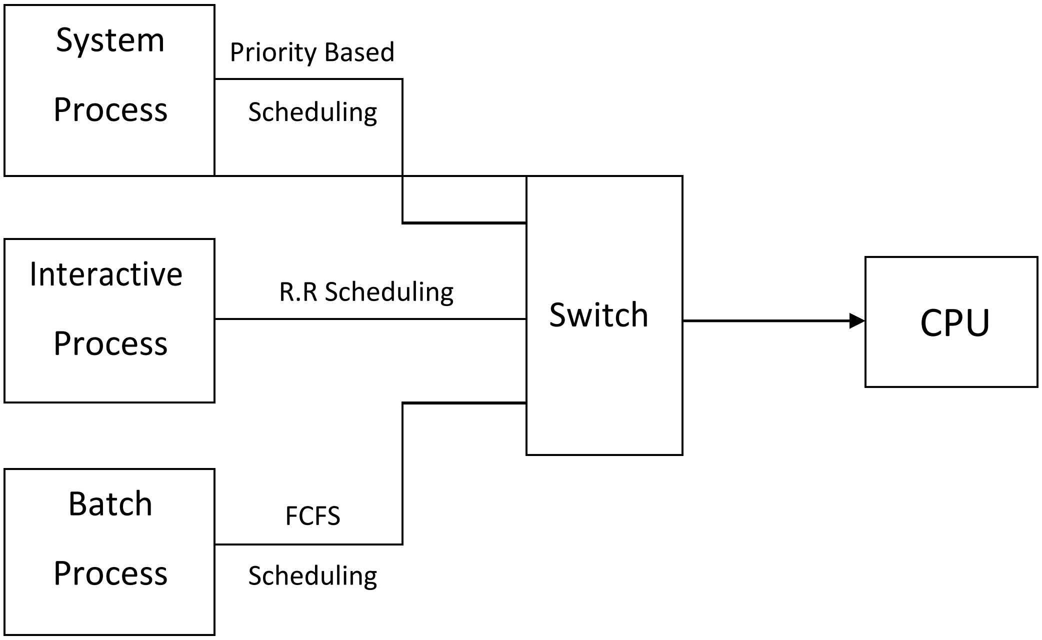
➢

Process	Waiting Time	Turnaround Time
P1	6	16
P2	0	1
P3	16	18
P4	18	19
P5	1	6

Avg. Waiting Time=41/5

=8.1Unit

5. **Multi-Level Queue(MLQ) Scheduling:-** In this scheduling algorithm, processes are classified into different groups. for example, the interactive (foreground) process and batch process(background). These 2-type of process have different response time and so they are scheduled in different manner. This scheduling algorithm partition the ready queue into system processes, interactive process and batch process that create 3 ready queues as shown bellow.



/*Inter-Process Communication*/

In multiprogramming environment multiple processes are executed concurrently (parallel). These concurrent processes also communicate with each other which are called inter-process communication.

The process of synchronization is also needed in case of inter-process communication. Processes are executed with very high speed and therefore 1-process must perform some task before other process deletes it.

Synchronization can be viewed as setup constrains on the ordering the events.

Following are the some of the technique use in process synchronization in multi programming environment:-

1. **Mutual Exclusion:** The processes that are working together often share some common storage may be in main-memory or it may be a shared file. Each process has segment of code called "Critical section" which access shared memory or files. The key issue is to restrict the more than one process from reading and writing the shared data at the same time.

   Mutual exclusion is a technique that makes sure that if one process is executing critical section and accessing the shared data then the other process will be excluded from doing the same thing. The mutual exclusion need to be enforced only when processes accessed shared data.

There is one drawback of mutual exclusion is that it doesn’t handle a situation when critical section of multiple processes get executed at the same time.

Following are the 4-conditions to handle the problem of critical section:-

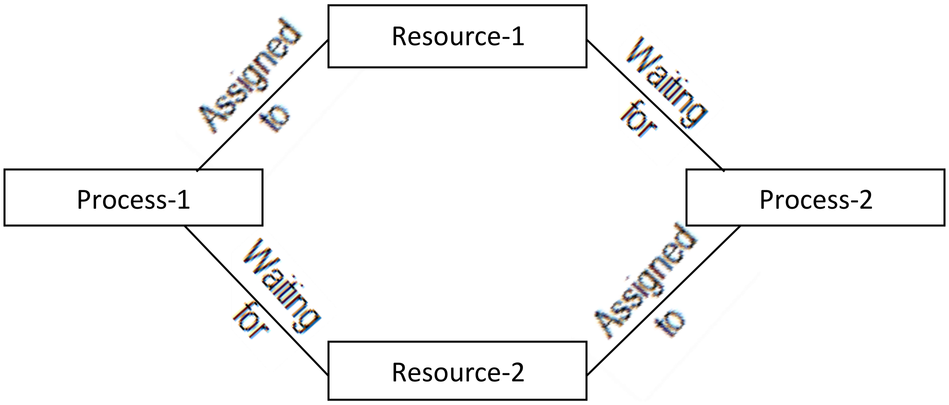
a) No processes may at the sometime enter into critical section.
b) No assumptions are made about relative speed of processes.
c) No processes outside the critical section should block other process.
d) No processes should wait long to enter its critical section.

2. **Semaphore:-** The problem encountered in mutual exclusion is overcome by a synchronization tool called semaphore which was proposed by Dijkestra in 1965.

A semaphore is a variable which accept non negative integer value and it is manipulated through to operations “Wait” and “Signal”. Each process ensure the integrity of its critical section by opening it with a “wait” operation and closing with a “signal operation. This way any number of con-current processes might share the resources provided each of these processes use wait and signal operation. A semaphore called binary semaphore (BSEM) can contain value of 0 and 1. 0 indicates wait and 1 indicates signal. The signal(1) indicates that the resource is available.

❖ **Dead lock:-** Dead lock is a situation where set of processes are blocked because each processes is holding a resource and waiting for another resource which is acquired by some other processes.

Following diagram shows the situation of dead lock.



In the above diagram, process 1 is holding resource 1 and waiting for resource 2 which is acquired by process 2 and process 2 is waiting for resource 1. This situation leads to deadlock where execution of both process is blocked.

Deadlock can arrive with following 4-condition -

a) **Mutual exclusion:-** Only one process at a time can use resources.

b) **Hold and wait:**- A process is holding one resource and waiting for another resource which is currently held by another process.

c) **No pre-emption:-** A resource can't be taken from a process unless the process release the resource, in other words the resources previous granted can't be precisely taken away from a process.

d) **Circular wait condition:-** A set of process are waiting for each other for a resource in circular form.

- **Handling dead lock:-** There are 4-strategies used for handling the dead lock situation.
  a) **Dead lock prevention or avoidance:-** The idea is to not let the system enter into deadlock state.
  b) **Detection and recovery:**- Detect the situation of dead lock and then do pre-emption to handle it.
  c) **Ignore the problem all together:**- If the deadlock is very rare then let it happen and reboot the system.
  d) Prevention by negating one of the 4-nesseary conditions.

/*Memory management*/

The memory management is mainly concerned with allocation of main memory to requested process. No process can even run before a certain amount of memory is allocated to it. Hence, organization and management of main memory has been one of the most important factors to be considered in design of OS. The overall resource utilization and other performance criteria of a computer system are mostly affected by the performance of memory management module.

Two-important feature of memory management function are protection and sharing.

An active process should never attempt to access the content of other process or destroy it. Apart from it , memory management scheme must support sharing of common data.

- **Single process monitor:-** In this memory management, the memory is divided into 2-sections:

  a.) Section for OS

  b.) Section for user Program

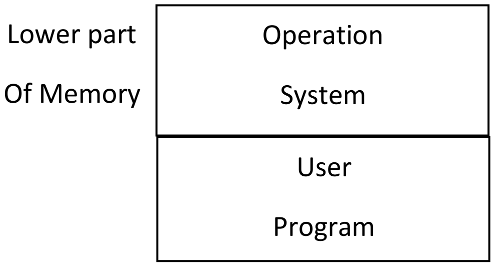


Fig: Memory layout of single process monitor

In this memory management approach, an OS keeps track of 1st and the last location available for allocation of user program.

When one program is completed and terminated, the OS may load another program for execution i.e. only one program is loaded at a time into memory. This type of memory management scheme is commonly used in single process OS such as MS DOS (Microsoft Disk Operating System).

Sharing data in single process environment doesn't make much sense because only one process resides in main memory at a time. Protection is hardly supported since, only one program is residing at a time in memory and so it may not occupy the whole memory. Therefore, memory is under-utilized and also CPU will be sitting idle when a running program require some I/O operation.

/*Multiprogramming with fixed partition /*

In multiprogramming environment, several programs reside in main memory at a time and the CPU passes its control rapidly between these programs. One way to support multiprogramming is to divide in main memory into several sections/partitions where each partition is allocated to a single process. Depending upon how and when partitions are created,

There are two types of memory partitioning. These are following:-

a) Static partitioning

b) Dynamic partitioning

OS
/////(Free)
P1
P2
P3
/////(Free)

a) **Static partitioning: -** In static partitioning, the memory is divided into number of partition and its size is made in the beginning itself which remain fixed thereafter.

b) **Dynamic partitioning:-** In dynamic partitioning the memory is divided into number of partition but the number of partition and its size is decided at run time by the OS.

Each partition will store a single program for execution. The no. of programs that can reside in memory is bounded by the number of partition. When the program terminates the partition is free for another program.

- **PDT (Partition Description Table):-** When partitions are define/ created, an OS keeps track of the status of memory partition into a data structure called "PDT". In PDT, OS maintains the information like starting address of partition, its size and status as free or allocated.

Partition No.	Starting Address	Size of partition	Status
1	OK	200K	Allocated
2	200k	200k	Free
3	400k	200k	Allocated
4	600k	300k	Allocated
5	900k	100k	Free
6	1000k	100k	Free

Fig: Partition Description Table

- **Partition allocation method:-** There are 2-most commonly techniques used to allocated free partition to ready process. These are following.

  a) First Fit

  b) Best Fit

a) **First fit:-** In this partition allocation method, an OS allocates first free partition to a requested process which is large enough to accommodate the process.

b) **Best fit:-** In this partition allocation method, an OS search the entire PDT and allocated the free partition to the requested process which is smallest to fit requirement.

Conclusion:- The first fit method is faster than best fit method but in case of first fit lots of memory get wasted where as best fit method achieve higher memory utilization.

Q1: Consider following given PDT and process to be loaded show the allocation of partition to process in first fit as well as in best fit.

Partition No	Address	Size	Status
1	100	100k	Free
2	200	500k	Free
3	700	200k	Free
4	900	300k	Free
5	1200	600k	Free

Process	Size
P1	212k
P2	417k
P3	112k
P4	426k

Fig: Process to be loaded

Fig: PDT

➔ Partition allocation in first fit algorithm:-

Process	Size	Allocation
P1	212k	500k
P2	417k	600k
P3	112k	200k
P4	426k	X(Can't be loaded)

➔ Partition allocation in best fit algorithm:-

Process	Size	Allocation
P1	212k	300k
P2	417k	500k
P3	112k	200k
P4	426k	600k

In first fit algorithm, the process P4 does not get loaded as the available partition is not able to fit the process. On the other hand, in case of best fit algorithm, all process gets loaded in memory. Hence, Best fit algorithm utilizes memory efficiently rather first fit.

- **Multiprogramming with dynamic partition:-** The main drawback with fixed size partition is the wastage of memory when the programs are smaller than partitions. This type of wastage of memory is also called internal fragmentation.

  A memory management approach known as dynamic partitions or variable partitions which creates the partition dynamically to meet the requirement of each process. Compared to multiprogramming with fixed partition, in multiprogramming with dynamic partition, the size, location and the no of partition very dynamically as process are created and terminated where as they are fixed in fixed size partition approach.

  In this approach the memory manager allocated partition to the requesting process until all the physical memory is exhausted. Therefore, memory utilization gets improved. The main problem in this approach is the external fragmentation i.e. even if there is enough memory to occupy the process, it can't satisfy a request due to non-contiguous memory allocation. The external fragmentation happens when the storage is fragmented into a smaller number of holes(free space).

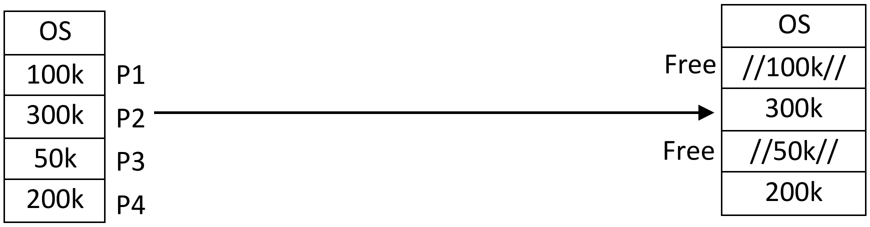


Fig: External Fragmentation

- **Compaction:-** In dynamic partition external fragmentation occurred as the process terminated. This external fragmentation problem is solved through compaction. Compaction is a process of combining all free holes into a large block by pushing all the process downwards as far as possible. Compaction is usually not done because it consume lots of CPU time.

  Following figure shows the compaction of memory.

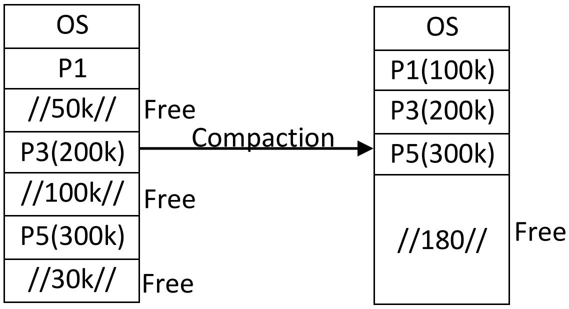


Fig: Compaction of memory

➔ Advantage:

I. Memory utilization is better than fixed size partitioning since, partition is created according to the size of process.

II. Protection and sharing in fixed and dynamic partition are similar.

III. When the process is larger than free partition then OS expends the free area by combining the adjacent free area and move the process into it.

→ Disadvantage:-

I. Dynamic memory management requires lots of OS space, time, complex memory management algorithm.

II. Compaction time is very high.

/*Paging*/

Paging is memory management scheme/technique which allows a process to be store in memory in non-contiguous partition. This way it solves the problem of external fragmentation. The program is divided into logical memory space or virtual address space called pages. Similarly the physical memory is also divided into the fixed/same size blocks called frames or page frames. When a process needs to be executed than its pages are loaded into any frames of physical memory from the disk.

The program generated addresses are called virtual address and the address where actually the page will be stored is called physical address. Hence, address mapping is required to convert the logical address into physical address.

The logical address consists of page no. and offset where as physical address consist of base address and offset.

Following diagram shows the address mapping in paging system.

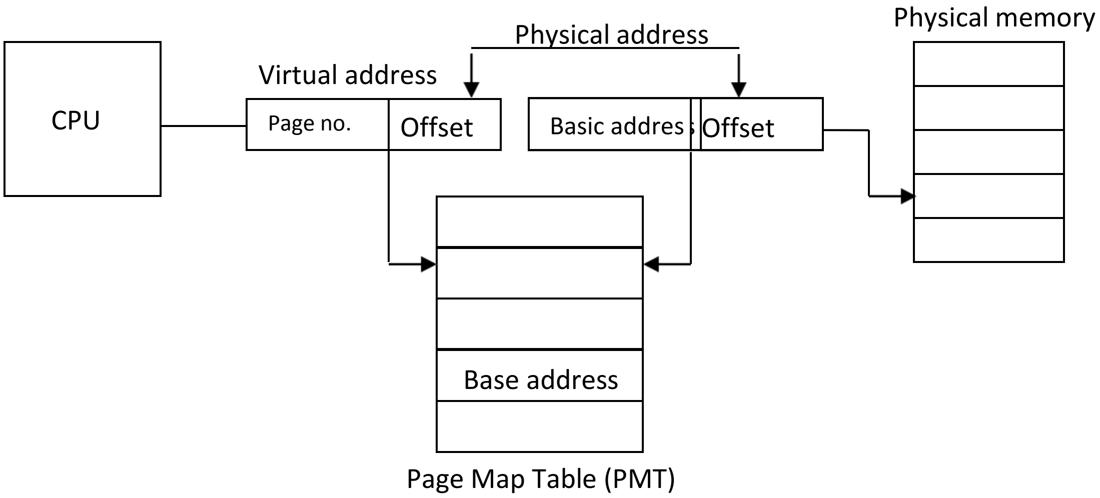


Fig: Address mapping in paging system

Paging system uses tables called Page Map Table (PMT) to store base address which consist of page number and offset.

- **Segmentation:-** Segmentation is a memory management scheme which support user's/programmer's view of memory. In this memory management technique, the process is divided into variable size segment and loaded to the logical memory address space. The logical address space is the collection of variable size segment where each segment has it length and name.

  The address specified by the user contains segment name and the offset and segments are no called segment no. Segment avoids internal fragmentation but leads to external fragmentation.

  Segmentation uses segment table in address mapping system where the virtual address is converted into physical address. The segment table contains segment no. ,base address and the size of segment.

Following diagram shows the address mapping in segment system.

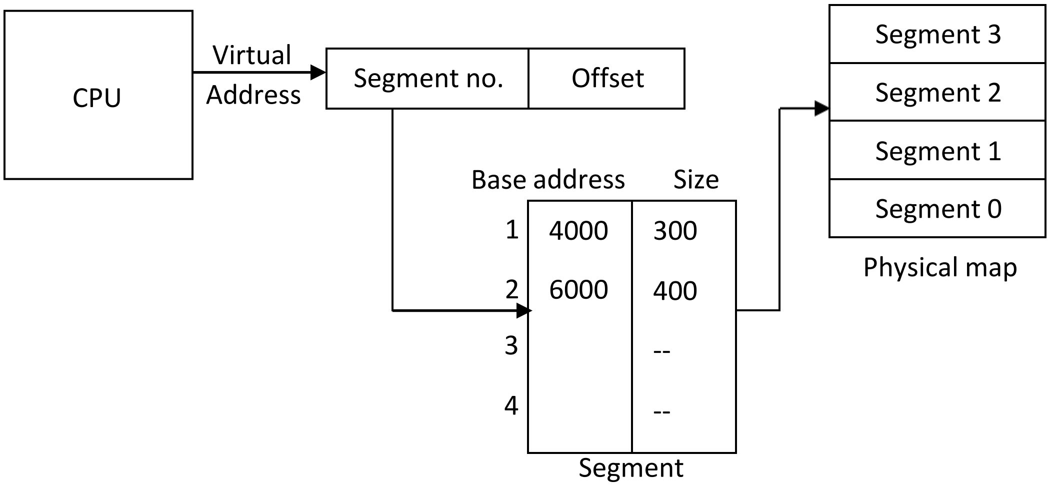


Fig: Virtual to physical memory mapping in segment system

- **Demand Paging:-** Demand paging is another memory management scheme used in paging system when memory capacity is low. In demand paging, the pages are loaded into memory only on the demand and not in advance. It is similar to paging system with a difference that instead of swapping the entire program into memory, only those pages are swapped which are required currently by the system.

  When a program tries to access a page which was not swapped in memory then it is called page fault trap. When Page fault trap occur then valid page is again loaded into memory. An OS handles the page fault in following way

  i) The OS checks whether the memory reference for the missing page was valid or not.

ii) If the memory reference is valid but the page is missing then process of bringing a page into physical memory starts.

iii) Free memory location is identify to bring the missing page.

iv) The page is read from disk and loaded into memory location

v) Page map table is updated with the process/page brought in memory

vi) Recrd the instruction which was intercepted due to missing page.

/*Disk management*/

Disk management is an important function of an OS. The performance of a computer system is also largely depends upon how fast a disk request is serviced. In multiprogramming environment, many processes may be generated request for reading and writing disk record. To service a request, the disk system requires moving the head at desired track then wait for latency and finally transfer the data. When more than track is to be serviced to different processes then the order in which track will be serviced to process depends upon the disk scheduling algorithm.

- **Disk scheduling algorithm:** Following are the various scheduling algorithm for disk services:-

    i. FCFS Scheduling

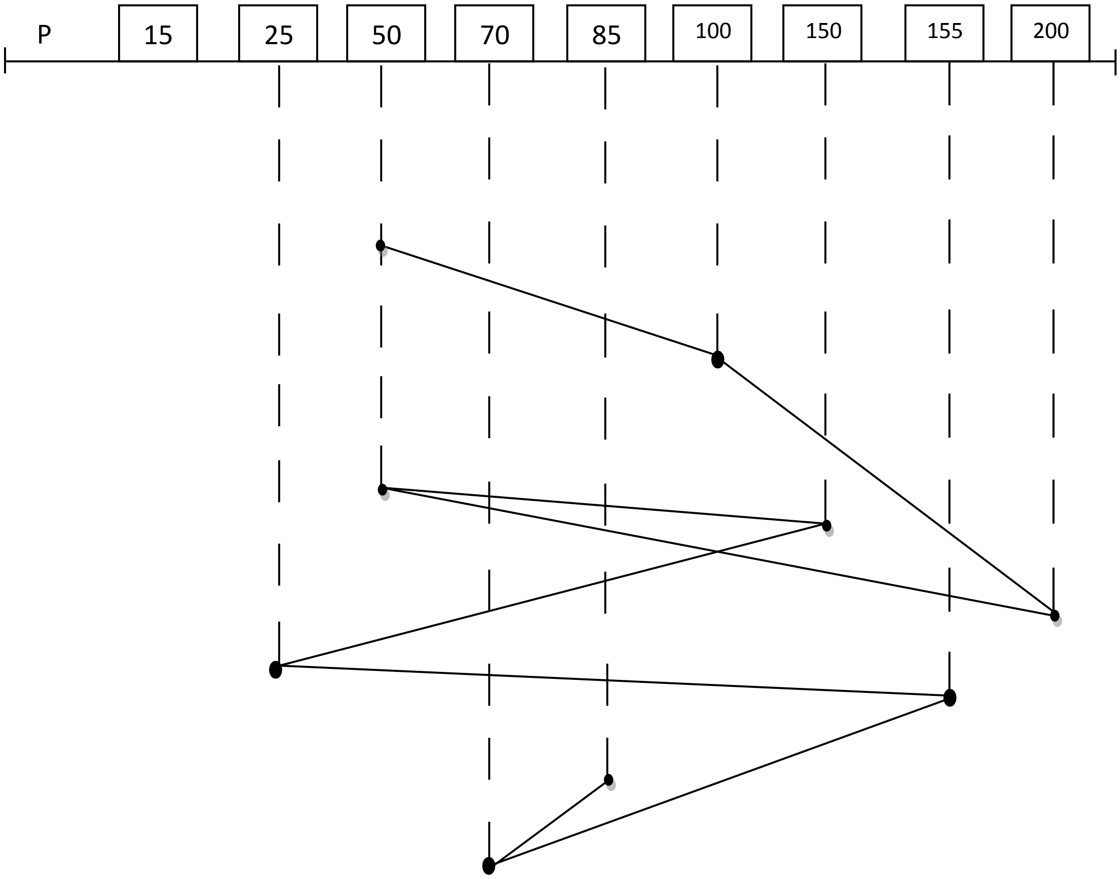
ii. SSTF Scheduling

iii. SCAN Scheduling

1. **FCFS(First Come First Served Scheduling):-** This is the simplest form of disk scheduling in which the first request to arrive is the first one to be serviced. It is easy to implemented but doesn't provides to best services.
   Consider the following track read under FCFS scheduling

   100, 200, 50, 150, 25, 155, 70 and 85

   Starting track to read in 100 and last to read in 85, let the head is initially positioned at track 50 find the total head movement.



Total head movement:- (100-50)+(200-100)+(200-50)+(150-50)+(150-25)+(155-25)+(155-70)+(85-70)
=> 50+100+150+100+125+300+85+15= **755 Ans**

2. **SSTF(Shortest Seek Time First):-** In SSTF scheduling, priority is given to those process which need shortest seek time. In other words, the requested track which are nearer to current head position will be serviced before the tracks which are far away from the head position.

   One of the disadvantage of SSTF scheduling of SSTF scheduling is that the inner-most and outermost tracks will received poor services as compare to need range tracks

3. **Scan scheduling:-** In this scheduling algorithm, the read/ write head starts from one end and moves to another end and services the requested(tracks). Which come on the path of head movement.

   After reaching another end the disk head reverses its path and continuing the request which even comes in the path. This way the head continues oscillates from one end to another end. Hence, this algorithm is also called "Elevator" algorithm.

4. **C-Scan scheduling (Circular):-** In this scheduling algorithm, the read write head moves from one end to another end and services the tracks which comes on the path of head movement. After reaching to another end of disk, the

read/write head reverses its direction to move to another end. It doesn't services the request while reversing the direction.

5. **Look scheduling: -** This disk scheduling algorithm is similar to Scan scheduling with a difference that look scheduling reverses its path of movement of read/write head after services the last track instead of jumping to the last tracks of disk.

6. **C-Look scheduling:-** This disk scheduling algorithm is similar to C-Scan algorithm with a difference that the read/ write head reverse its movement of path after services the last requested and doesn't jump the last track of disk. Like C-scan, it doesn't provides services while reversing its path.

/*Disk space management method*/

Operating System maintains a list of free disk space to keep track of all disk block which are not being used. Whenever a file is to be created, the list of free disk space is searched for and then allocated to new file. The amount of space allocated to file is the removed from the free space list and when file is deleted, the free disk space is added to free space list. Following are the two method use to manage free disk blocks:-

i. Linked list
ii. Bit-map

1. **Linked-list:-** In this method, all free disk blocks are linked together by each free block pointing to the next free block. There is an extra pointer pointing to the first free block of linked list

   Following diagram show the linked list method:-

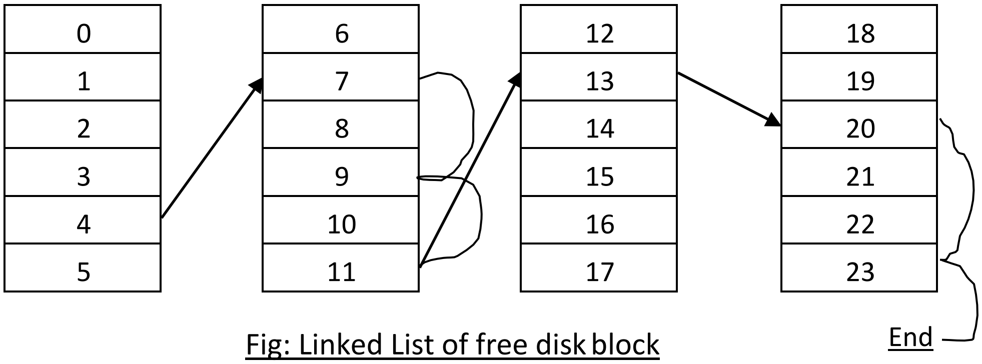


Fig: Linked List of free disk block

❖ Drawback of linked-list method:-

i. To reach at specific free block, traversing starts from the beginning and hence, it tacks substantial time.

ii. The pointer maintains free disk block requires additional disk/memory space.

iii. A single break in the link makes the disk block inaccessible

2. **Bit map:-** In this disk space management method, binary digits are used for indicating free blocks and allocated blocks. The binary digits 0 is used for marking free block whereas 1 is used for marking allocated block

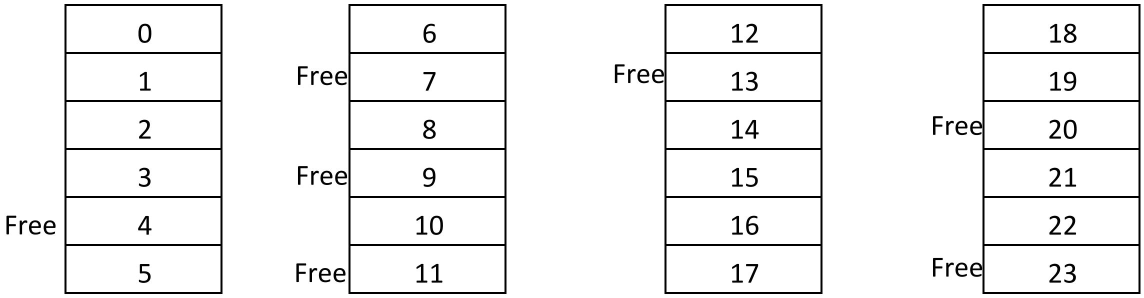


Fig: Allocated and free block disk

In this above figure the block 4, 7, 9, 11, 13, 20 and 23 are free block and rest of the block are allocated to file.

The bit map representation for the above shows free and allocated block will be as follow:

111101101010101111110110

One of the main advantages of this method is that it is simple and efficient to find the free block on the disk but the disadvantage is that it requires extra disk space to store bitmap.

/*Disk allocation method*/

There are 2-popular methods for allocation free disk-block to file these are:-

a) Contiguous allocation

b) Non-contiguous allocation

1. **Contiguous allocation:** In this method, files are assigned to contiguous area of secondary storage. A user specifies the size of area in advance needed to hold a file and if the desired amount of contiguous space is not available the file can't be created.

This method uses following to ways to allocated free block:-

a) First fit
b) Best fit

I. **First fit:-** In this case, as soon as first free block large enough to store file in encountered, it is allocated.

II. **Best fit: -** In this case, the smallest block large enough to accommodate the file is searched and then allocated.

First fit is faster than best fit but best fit does the efficient management of disk space. In case of first fit lots of disk space are wasted.

2. **Non-contiguous allocation:-** In this method files are allocated to non-contiguous free block. It uses following two technique to keep track of non-contiguous blocks

a) Linked allocation
b) Indexed allocation

I. **Linked allocation:-** This method uses linked list for maintaining the allocated non-contiguous block.

Because of non-contiguous allocation there is no external disk fragmentation.

It is simple and doesn't require disk compaction but it doesn't allow direct accessing.

Also there is a problem of reliability and takes extra space for storing pointer.

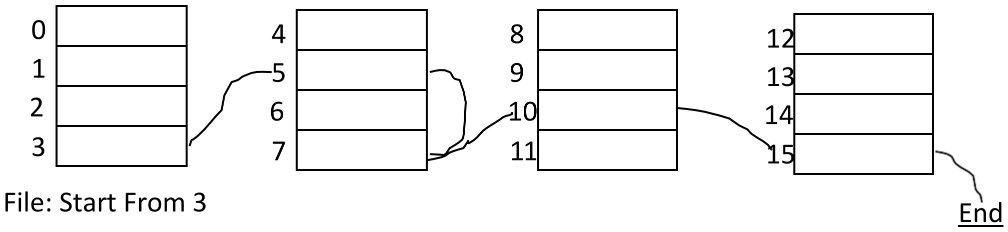
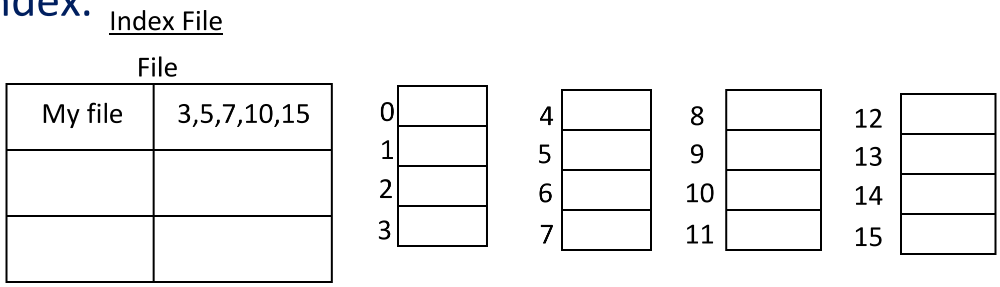


Fig: Linked allocation of disk block

II. **Indexed allocation:-** In this method, the allocated disk block are maintained by creating the index of allocated block. Each indexed points to the allocated block one of the major advantages of indexed allocation is that the index allocation allows direct accessing. The disadvantage of index allocation is that it is more complex and takes more memory for maintaining to index.



My file	3,5,7,10,15

Fig: Index allocation of disk block

THE END

THANK YOU