6

Operating System

Date __/__/____

Operating System

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of program. eg: Linux, windows os, VMS, etc.

Functions of operating system :-

* Memory Management
* Processor Management
* Device Management
* File Management
* security
* control over system performance
* Job accounting
* Error detecting aids
* coordinate between other software and users

* **Multiprocessing :-** It is an ability of a computer to use two or more processors (multiprocessors) for computer operation. eg: UNIX is one of the most widely used multiprocessing systems.

* **Multiprogramming :-** It is a technique in which several programs are run at the same time on a uniprocessor. It is a form of parallel processing.

Page N

Date ___/___/_____

* Time-sharing :- In computing, time sharing is the sharing of a computing resource among many users at the same time by means of multiprogramming and multitasking.

* Memory Swapping :- Swapping is a memory management scheme in which any process can be temporarily swapped from main memory to secondary memory so that the main memory can be made available for other processes.
It is used to improve main memory utilization. In secondary memory, the place where the swapped-out process is stored is called swap space.

→ Swappin has two more parts -

① Swap-out :- It is a method of removing a process from RAM and adding it to the hard disk.

② Swap-in :- It is a method of removing a program from a hard disk and putting it back into the main memory or RAM.

Buffer – The temporary storage area in main memory (RAM) is called Buffer.

Buffering – The overlaping of a input and output of a single job is known as Buffering.

* Types of Buffer :-

(i) Single buffer
(ii) double buffer
(iii) circular buffer

SPOOL – Simultaneous peripherals operation on line

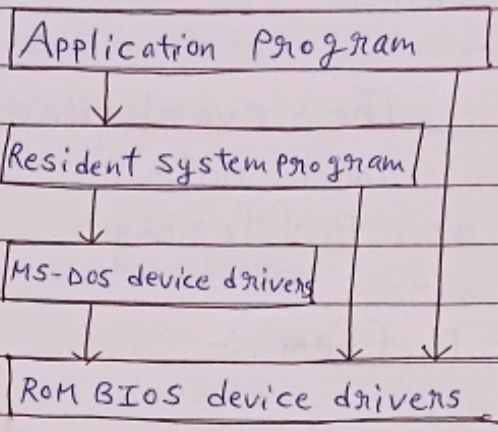
Spooling – The overlaping of input of a number of jobs and output of a single job is known as spooling.

structures of operating system :-

1) Simple structure – such operating systems do not have well defined structure and are small, simple and limited systems. The interfaces and levels of functionality are not well separated.
eg: MS-DOS.
These types of OS cause the entire system to crash if one of the user programs fails.

Date __/__/____

Diagram of the structure of MS-DOS is shown below.



* Advantages of simple structure :-

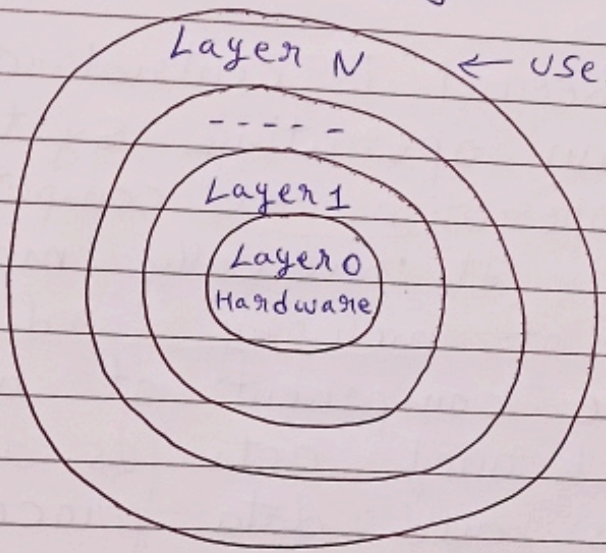
- It delivers better application performance because of the few interfaces between the application program and the hardware.
- Easy for kernel developers to develop such an operating system.

* Disadvantages of simple structure :-

- The structure is very complicated as no clear boundaries exists between modules.
- It does not enforce data hiding in the ~~oper~~ operating system.

2.) Layered structure - In this structure OS is broken into number of layers. The bottom layer (layer 0) is the hardware and the topmost layer

(layer N) is the user interface. These layers are so designed that each layer uses the functions of the lower level layers only. This simplifies the debugging process as if lower layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower layers have already been debugged.



* Advantages of Layered structure :-

- Layering makes it easier to enhance the operating system as implementation of a layer can be changed easily without affecting the other layers.
- It is very easy to perform debugging and system verification.

Date __/__/____

* Disadvantages of Layered structure :-

- In this structure the application performance is degraded as compared to simple structure.
- It requires careful planning for designing the layers as higher layers use the functionalities of only the lower layers.

Kernel - Kernel is central component of an operating system that manages operations of computer and hardware. It basically manages operations of memory and CPU time. It is core component of an operating system. Kernel acts as a bridge between applications and data processing performed at hardware level using inter-process communication and system calls.

→ objectives of kernel :-

* To establish communication between user level application and hardware.
* To decide state of incoming process.
* To control disk management, memory management and task management.

1.)
2.)
3.)
4.)
5.)

Page No.

Date __/__/____

Types of Kernel :-

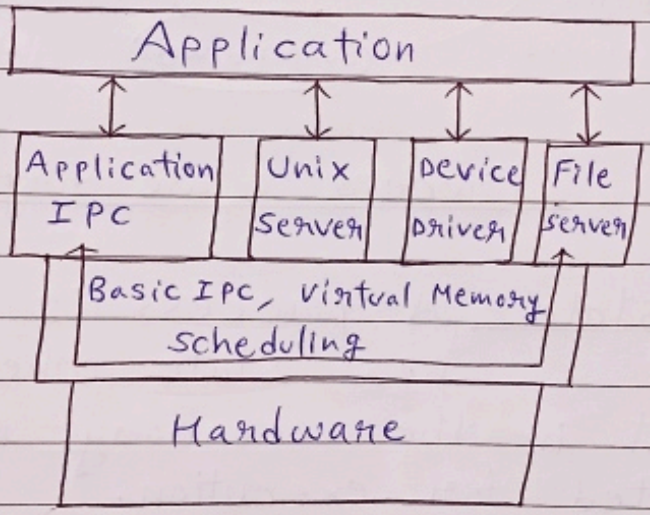
1.) Monolithic Kernel - Unix, Linux, etc.

2.) Micro Kernel - Mach, Minix, etc.

3.) Hybrid Kernel - Windows NT, Netware, etc.

4.) Exo Kernel - Nemesis, ExOS, etc.

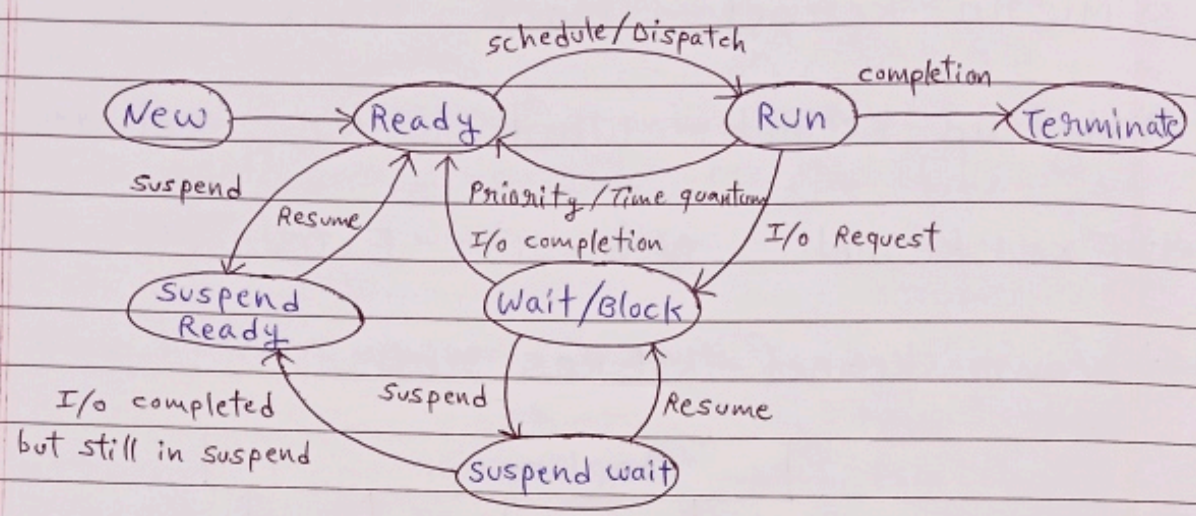
5.) Nano Kernel - EROS, etc.



Process - In computing, a process is the instance of a computer program that is being executed by one or many threads. ~~dep~~ Depending on the operating system, a process may be made up of multiple threads of execution that execute instructions concurrently.

Process states in operating system -

Each process goes through different states in its life cycle -



Process state diagram

1.) **New state** - A process is said to be in new state when a program present in the secondary memory is initiated for execution.

2.) **Ready state** - A process moves from new state to ready state after it is loaded into the main memory and is ready for execution.

3.) **Run state** - A process moves from ready state to run state after it is assigned the cpu for execution.

4.) **Terminate state** - A process moves from run state to terminate

state after its execution is completed.

5.) Block or Wait state - A process moves from run state to block or wait state if it requires an I/O operation or some blocked resource during its execution.

6.) Suspend Ready state - A process moves from ready state to suspend ready state if a process with higher priority has to be executed but the main memory is full.

7.) Suspend wait state - A process moves from wait state to suspend wait state if a process with higher priority has to be executed but the main memory is full.

~~she~~ Schedulers - Schedulers are special system software which handle process scheduling in various ways. Their main task is to select the jobs to be submitted into the system and to decide which process to run.

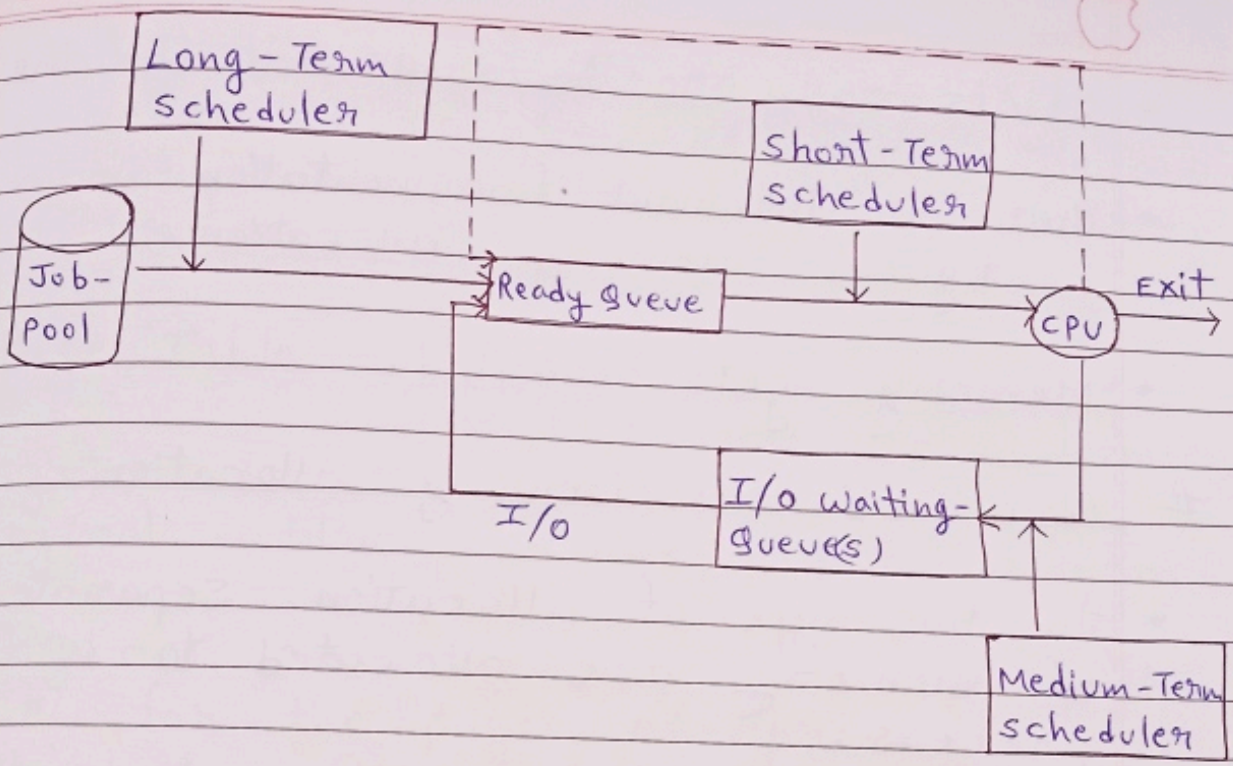
→ Types of schedulers :-

(i) Long-Term scheduler
(ii) short-Term scheduler
(iii) Medium-Term scheduler

1.) Long-Term scheduler – It is also known as Job scheduler. Long-term scheduler regulates the programs which are selected to system for processing. In this the programs are setup in the queue and as per the requirement the best one job is selected and it takes the processes from job pool.

2.) short-Term scheduler – It is also known as the CPU scheduler. It decides which of the ready, in memory processes is to be executed after a clock interrupt, an I/O interrupt, an operating system call or another ~~from~~ form of signal.

3.) Medium-Term scheduler – It involves swapping out a process from main memory. The process can be swapped in later from the point it stopped executing. This can also be called as suspending and resuming the process and is done by the medium-term scheduler.



Memory Management – It is the functionality of an operating system which handles or manages primary memory and moves processes back and forth between main memory and disk during execution. It checks how much memory is to be allocated to processes. It decides which process will get memory at what time.

Contiguous Memory Allocation –

* In this allocation type, the consecutive blocks of memory is allocated to a file/ process.

* It executes quickly in comparision to non-contiguous memory.

* It is easy to be controlled by the os.
* There is internal fragmentation in contiguous memory allocation.
* Memory gets wasted.

Non-contiguous Memory Allocation –

* In this type of allocation, separate blocks of memory are allocated to a file/ process.
* It executes slowly in comparison to contiguous memory.
* It is difficult to be controlled by the os.
* It includes paging.
* It includes segmentation.
* No memory is wasted.
* External fragmentation occurs in ~~the~~ this type of allocation.

Paging – In operating Systems, Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages. The main idea

Date ___/___/_____

behind the paging is to divide each process in the form of pages. The main memory will also be divided in the form of frames.

Page Table – It is a data structure used by the virtual memory system to store the mapping between logical addresses and physical addresses.

Logical addresses are generated by the CPU for the pages of processes and Physical addresses are the actual frame address of the memory.

Segmentation – In operating systems, Segmentation is a memory management technique in which the memory is divided into the variable size parts. Each part is known as a segment which can be allocated to a process. The details about each segment are stored in a table called a segment table.

Virtual Memory – It is a storage mechanism which offers user an illusion of having a very big main memory. It is done by treating a part of secondary memory as the main memory. In virtual memory, the user can store processes with a

Page No.

bigger size than the available main memory.

Differences b/w paging and segmentation -

Paging	segmentation
(i) In paging, program is divided into fixed size pages.	In segmentation, program is divided into variable size sections.
(ii) For paging operating system is accountable.	For segmentation, compiler is accountable.
(iii) Page size is determined by hardware.	Here, the section size is given by the user.
(iv) It is faster than segmentation.	It is slow.
(v) It could result in internal fragmentation.	It could result in external fragmentation.
(vi) It is invisible to user.	It is visible to user.
(vii) In this protection is hard to apply.	Easy to apply protection in segmentation.

Demand Paging – It is a technique used in virtual memory systems where the pages are brought in the main memory only when required or demanded by CPU. Hence it is also named as Lazy swapper because the swapping of pages is done only when required by the CPU.

CPU Scheduling Algorithms

CPU scheduling – It is a process of determining which process will own CPU for execution while another process is on hold. The main task of CPU scheduling is to make sure that whenever the CPU remains idle, the OS at least select one of the processes available in the ready queue for execution. The selection process will be carried out by the CPU schedular.

Types of CPU scheduling :-

1. Preemptive scheduling
2. Non-Preemptive scheduling

1.) Preemptive scheduling - In this, the tasks are mostly assigned with their priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running.

2.) Non-Preemptive scheduling - In this, the CPU has been allocated to a specific process. The process that keeps the CPU busy will release the CPU either by switching context or terminating.

Important CPU scheduling Terminologies :-

- Burst time / Execution time - It is a time required by the process to complete execution. It is also called running time.
- Arriva Time - when a process enters in ready state.
- Finish Time - when process complete and exit from a system.
- Multiprogramming - A number of programs which can be present in memory at the same time.

Date __/__/____

- **Jobs** – It is a type of program without any kind of user interaction.
- **Users** – It is a kind of program having user interaction.
- **Process** – It is the reference that is used for both job and user.
- **CPU/IO burst cycle** – characterizes process execution, which alternates between CPU and I/O activity. CPU times are usually shorter than the time of I/O.

CPU scheduling criteria :-

A CPU scheduling algorithm tries to maximize and minimize the following:

→ **Maximize**
CPU utilization
Throughput

→ **Minimize**
Turnaround Time
waiting Time
Response Time

- **CPU utilization** – It is the main task in which the operating system needs to make sure that CPU remains as busy as possible.
- **Throughput** – The number of processes that finish their execution

Date __/__/____

- **waiting time** – It is an amount of time that specific process needs to wait in the ready queue.
- **Response time** – It is an amount of time in which the request was submitted until the first response is produced.
- **Turnaround time** – It is an amount of time taken to execute a particular process, i.e. the interval from time of submission of the process to the time of completion of the process.

Turnaround Time = Processing Time + Waiting Time

Types of CPU scheduling algorithm :-

1. First come First serve (FCFS)
2. Shortest-Job-First (SJF)
3. Shortest Remaining Time (SRT)
4. Priority scheduling
5. Round Robin (RR)
6. Multilevel Queue scheduling

1.) First come First serve (FCFS) –

In this algorithm, Jobs are executed on first come first serve basis. The average waiting time is high so poor in performance.

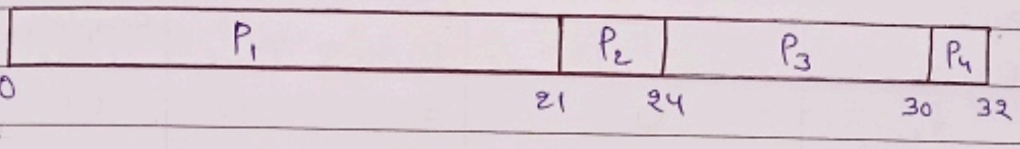
2.)

eg:

Process	Burst Time	Waiting Time
P_1	21	0
P_2	3	21
P_3	6	24
P_4	2	30

Average waiting time = (0 + 21 + 24 + 30) / 4
= 18.75

GANTT chart



2.) Shortest - Job - First (SJF) -

In this algorithm, process with shortest execution time should be selected for execution next. This is the best approach to minimize the waiting time.

eg:

Process	Burst Time
P_1	21
P_2	3
P_3	6
P_4	2

Process	Burst Time	Waiting Time
P_4	2	0
P_2	3	2
P_3	6	5
P_1	21	11

Average waiting Time = (0 + 2 + 5 + 11) / 4
= 4.5

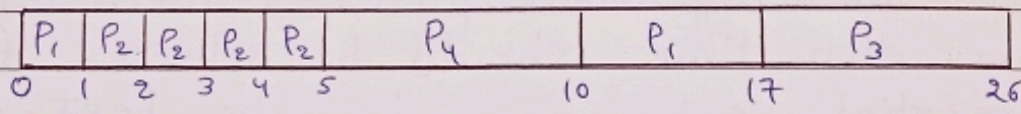
Date ___/___/___

3.) Shortest Remaining Time (SRT) –

It is also known as SJF preemptive scheduling. In this method, the process will be allocated to the task, which is closest to its completion. This method prevents a newer ready state process from holding the completion of an older process.

eg:

Process	Arrival Time	Burst Time	Completion Time	TAT	WT
P_1	0	~~8 7 6 5 4 3 2 1 0~~	17	17	9
P_2	1	~~4 3 2 1 0~~	5	4	0
P_3	2	~~9 8 7 6 5 4 3 2 1 0~~	26	24	15
P_4	3	~~5 4 3 2 1 0~~	10	7	2



* TAT = CT – AT
* WT = TAT – BT

4.) Priority scheduling –

It is a method of scheduling processes that is based on priority. The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round robin or FCFS basis. Priority depends upon memory requirements, time requirements, etc.

5.)

Page No. ☐

eg:

Process	Burst Time	Priority
P_1	10	2
P_2	1	1
P_3	2	3
P_4	1	4

Process	Burst Time	Waiting Time
P_2	1	0
P_1	10	1
P_3	2	11
P_4	1	13

Average waiting Time = $(0+1+11+13)/4 = 6.25$

5.) Round-Robin scheduling –

It is the preemptive version of FCFS. The algorithm focus on time sharing. In this, every process gets executed in a cyclic way. A certain time slice is defined in the system which is called Time quantum. Each process present in the ready queue is assigned the CPU for that time quantum, if the execution of the process is completed during that time then the process will terminate else the process will go back to the ready queue and waits for the next turn to complete the execution.

Date __/__/____

eg: Time Quantum = 2

Process	Arival Time	Burst Time	CT	TAT	WT
P_1	0	5 3 1	12	12	7
P_2	1	4 2 0	11	10	6
P_3	2	2 0	6	4	2
P_4	3	1 0	9	6	5

Ready queue | P_1 | P_2 | P_3 | P_1 | P_4 | P_2 | P_1 |

GANTT chart | P_1 | P_2 | P_3 | P_1 | P_4 | P_2 | P_1 |

0 2 4 6 8 9 11 12

Average waiting time = (7+6+2+5)/4
= 5

6.) Multilevel Queue scheduling –

This algorithm separates the ready queue into various separate queues. In this method, processes are assigned to a queue based on a specific property of the process, like the process priority, size of the memory, etc.

It needs to use other types of algorithms in order to schedule the jobs. Every queue may have its separate scheduling algorithms.

* CPU scheduling improve its efficiency and allocate resources among competing processes.

Page No.

Page Replacement Algorithms :-

This algorithm helps to decide which pages must be swapped out from the main memory in order to create a room for the incoming page. This algorithm wants the lowest page-fault rate.

* Page Fault - A page fault happens when a running program accesses a memory page that is mapped into the virtual address space, but not loaded in physical memory.

* Various Page Replacement algorithms :-

1. FIFO Page Replacement algorithm
2. LRU Page Replacement algorithm
3. Optimal Page Replacement algorithm

1.) FIFO Page Replacement algorithm -

It is very simple way of page replacement and is referred to as First in First out. This algorithm mainly replaces the oldest page that has been present in the main memory for the longest time. This algorithm is implemented by keeping the track of all the pages in the queue.

eg:

Reference string: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 1, 2, 0

f_3	7	0	1	1	1	1	0	0	0	3	3	3	3	2	2			
f_2		0	0	0	0	3	3	3	2	2	2	2	1	1	1			
f_1	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0			
	*	*	*	*	Hit	*	*											

eg:

Reference string:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 1, 2, 0

f_3			1	1	1	1	0	0	0	3	3	3	3	2	2			
f_2		0	0	0	0	3	3	3	2	2	2	2	1	1	1			
f_1	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0			
	*	*	*	*	Hit	*	*	*	*	*	*	Hit	*	*	Hit			

Page faults = 12

Page fault ratio = $\frac{12}{15} = \frac{4}{5}$

Page fault % = $\frac{4}{5} \times \frac{20}{100} = 80\%$

Page hits = 3

Page hit ratio = $\frac{3}{15} = \frac{1}{5}$

Page hit % = $\frac{3}{15} \times \frac{20}{100} = 20\%$

2.) LRU Page Replacement algorithm –

This algorithm stands for "Least recent used" and this algorithm helps the OS to search those pages that are used

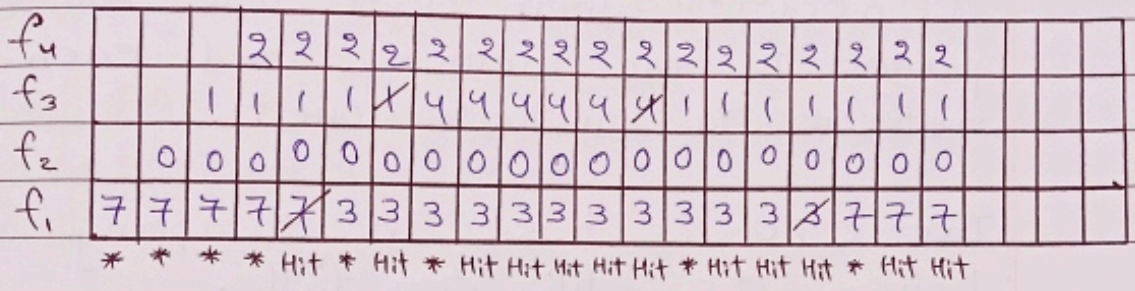
3.)

over a short duration of time frame. In this algorithm, the page that has not been used for the longest time in the main memory will be selected for replacement.

eg:

Reference string:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1



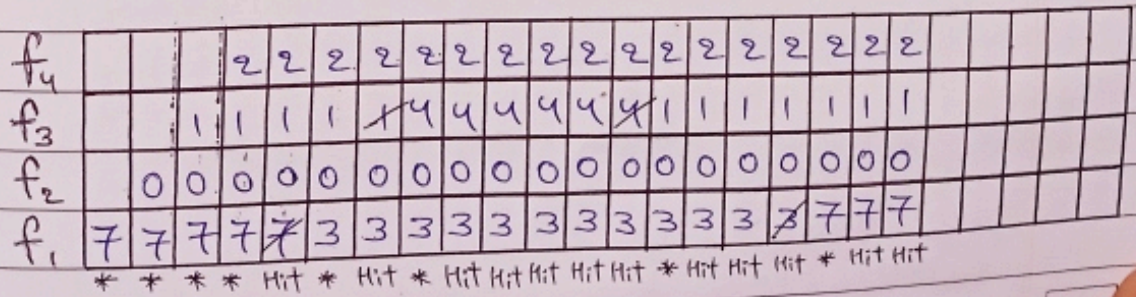
Page Faults = 8 Page Hits = 12

3) Optimal Page Replacement algorithm –

This algorithm mainly replaces the page that will not be used for the longest time in the future. The practical implementation of this algorithm is not possible. This algorithm leads to less number of page faults and thus is the best-known algorithm.

eg: Reference string:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1



Page Faults = 8 Page Hits = 12

Date ___ / ___ / ____

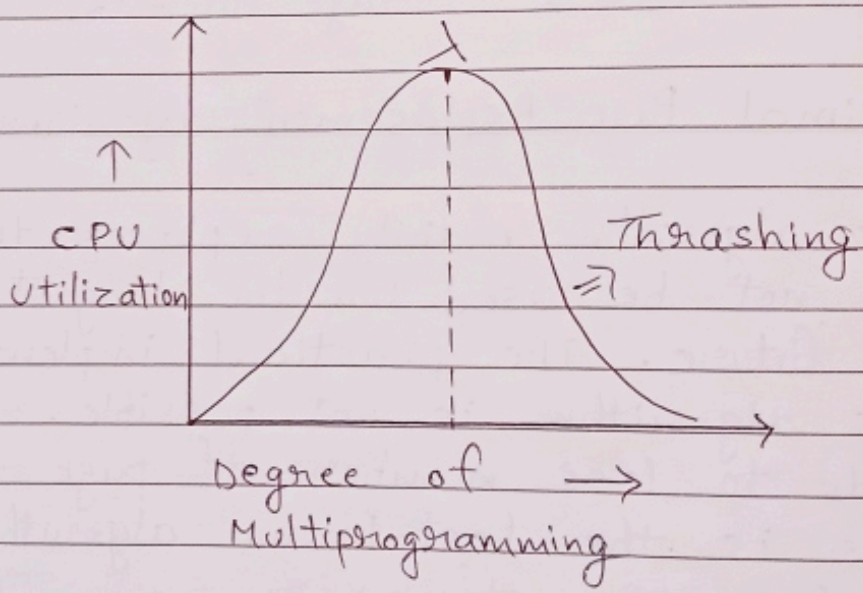
* ~~Thrasing~~

* Thrashing –

A state in which the CPU performs lesser "productive" work and more "swapping" is known as thrashing.

It occurs when there are two many pages in the memory and each page refers to another one.

The CPU is busy in swapping and hence its utilization falls.



Date ___/___/___

Disk structure

Disk sheduling – A process makes the I/o requests to the operating system to access the disk. Disk scheduling Algorithm manages those requests and decides the order of the disk access given to the requests.

These algorithms help in minimizing the seek time by ordering the requests made by the processes.

* Seek Time – It is the time taken by the disk arm to locate the desired track.

* Rotational Latency – The time taken by a desired sector of the disk to rotate itself to the position where it can access the Read/write heads is called Rotational Latency.

* Transfer Time – It is the time taken to transfer the data requested by the processes.

* Disk Access Time – Disk Access Time is the sum of the Seek Time, Rotational Latency and Transfer Time.

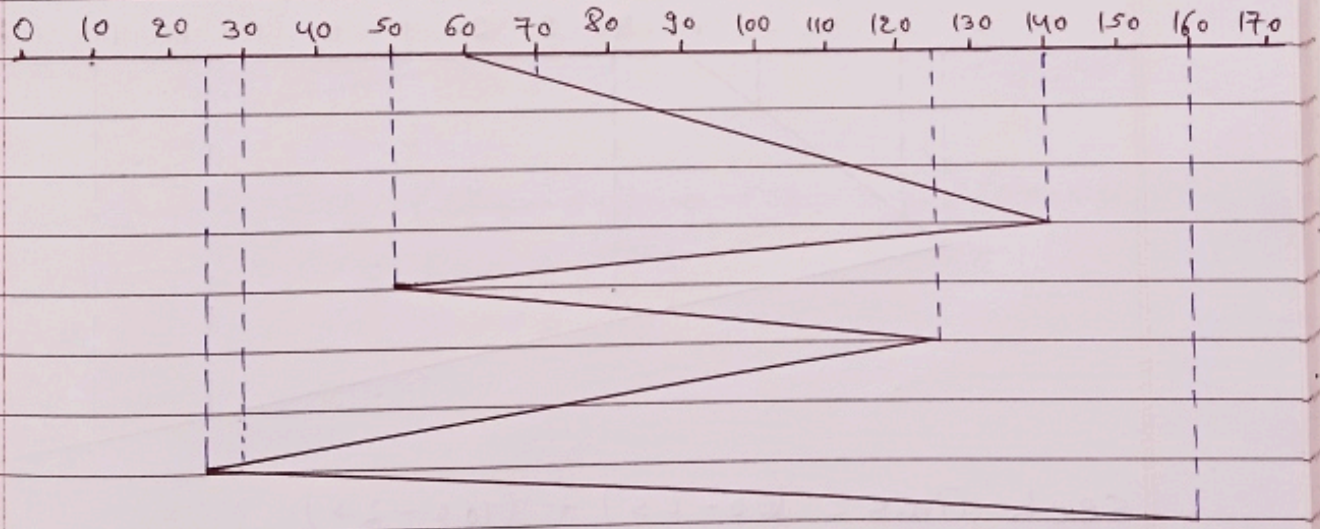
Date ___/___/___

Disk scheduling Algorithms :-

1) First come First serve (FCFS) -

In this algorithm, the requests are served in the order they come. Those who come first are served first. This is the simplest algorithm.

Eg: Suppose the order of requests are 70, 140, 50, 125, 30, 25, 160 and the initial position of the Read-write head is 60.

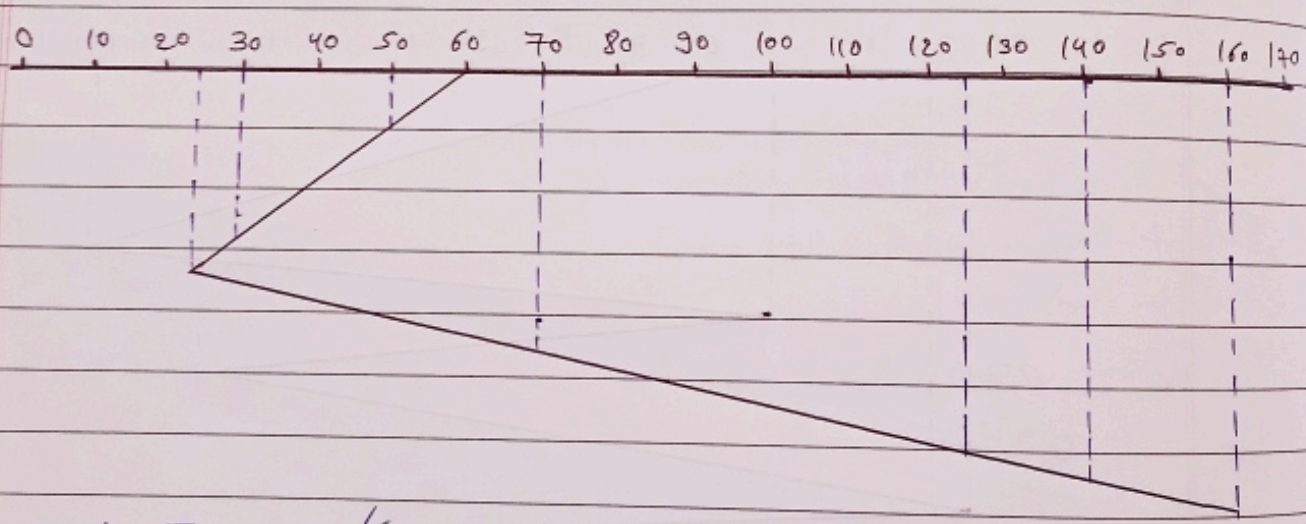


Seek Time = (140 - 60) + (140 - 50) + (125 - 50)
+ (125 - 30) + (160 - 25)
= 80 + 90 + 75 + 100 + 135
= 480

2.) Shortest seek Time First (SSTF) -

In this algorithm, the shortest seek time is checked from the current position and those requests which have the shortest seek time is served first. In simple words, the closest request from the disk arm is served first.

Eg: Suppose the order of requests are 70, 140, 50, 125, 30, 25, 160 and the initial position of the Read-write head is 60.

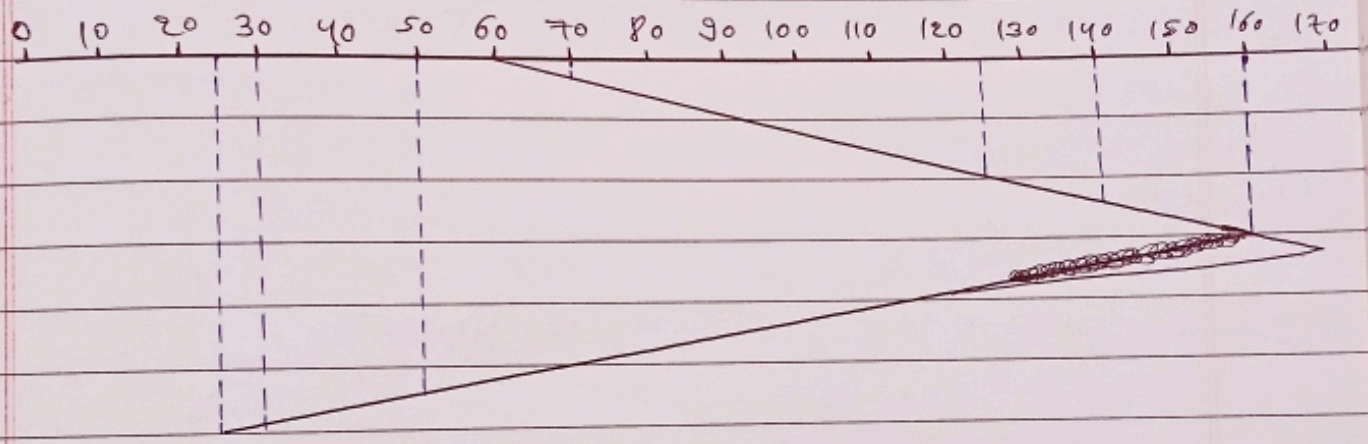


seek Time = (60-25) + (160-25)
= 35 + 135
= 170

3.) Scan - In this algorithm, the disk arm moves in a particular direction till the end and serves all the requests in its path, then it returns to the opposite direction and moves till the last

request is found in that direction and serves all of them.

Eg: Suppose the order of the requests are 70, 140, 50, 125, 30, 25, 160 and the initial position of the Read-write head is 60. And it is given that the disk arm should move towards the larger value.



Seek Time = ~~(170 60) + (170 - 25)~~ (170 - 60) + (170 - 25)

= ~~100 + 135~~ 110 + 145

= ~~235~~ 155

File System Structure

File allocation - The allocation methods define how the files are stored in the disk blocks. The main idea behind these methods is to provide:

- Efficient disk space utilization.
- Fast access to the file blocks.

→ There are three main disk space or file allocation methods -

(i) Contiguous Allocation
(ii) Linked Allocation
(iii) Indexed Allocation

1.) Contiguous Allocation - In this scheme, each file occupies a contiguous set of blocks on the disk. For example, if a file requires n blocks and is given a block b as the starting location, then the block assigned to the file will be: b, b+1, b+2, . . . b+n-1. The directory entry for a file with contiguous allocation contains

- Address of starting block.
- Length of the allocated portion.

Date __/__/____

* Advantages of contiguous allocation -

- Both the sequential and direct Accesses are supported by this.
- This is extremely fast since the number of seeks are minimal.

* Disadvantages of contiguous Allocation -

- This method suffers from both internal and external fragmentation.
- Increasing file size is difficult because it depends on availability of contiguous memory at a particular instance.

2.) Linked List Allocation - In this scheme, each file is a linked list of disk blocks which need not be contiguous. The disk blocks can be scattered anywhere on the disk.

The directory entry contains a pointer to the starting and the ending file block. Each block contains a pointer to the next block occupied by the file.

* Advantages of Linked List Allocation -

- File size can be increased easily since the system doesn't have to look for a

Page No.

contiguous chunk of memory.

- This method does not suffer from external fragmentation.

* Disadvantages of linked list allocation -

- Because the file blocks are distributed randomly on the disk, a large number of seeks are needed to access every block individually.
- It does not support random or direct access.
- Pointers required in the linked allocation.

3.) Indexed Allocation - In this scheme, a special block known as the Index block contains the pointers to all the blocks occupied by a file. Each file has its own ~~Idex~~ index block. The directory entry contains the address of the index block.

* Advantages of Indexed Allocation -

- This supports direct access to the blocks occupied by the file and therefore provides fast access to the file blocks.
- It overcomes the problem of external fragmentation.

Date __/__/____

* Disadvantages of Indexed Allocation -

- The pointer overhead for indexed allocation is greater than linked allocation.
- For very small files, the indexed allocation would keep one entire block for the pointers which is inefficient in terms of memory utilization. However, in linked allocation we lose the space of only 1 pointer per block.

Free Space Management

A file system is responsible ~~for~~ to allocate the free blocks to the file therefore it has to keep track of all the free blocks present in the disk.

There are mainly two approaches for managed the free blocks in the disk -

1) Bit vector (Bitmap) - In this approach, the free space list is implemented as a bit map vector. It contains the number of bits where each bit represents each block.

If the block is empty then the bit is 1 otherwise it is 0. Initially all the blocks are empty therefore each bit in the bit map vector contains 1.

2)

2.) Linked List – This approach suggests linking together all the free blocks and keeping a pointer in the cache which points to the first free block.

Therefore, all the free blocks on the disks will be linked together with a pointer. whenever a block gets allocated, its previous free block will be linked to its next free block.

I/O systems

Polling –

- The process in which the CPU constantly checks the status of the device to see if it needs the CPU's attention.
- It is a protocol, in this protocol the CPU services the device.
- CPU polls the devices at regular intervals of time. This waste many of the CPU cycles.

Interrupt –

- It is a process with the help of which the CPU is notified of requiring attention.
- It is considered as a hardware mechanism.

Date __/__/____

- An interrupt handler services/works with the device.
- CPU is used only when a device requires servicing. This saves the CPU cycles.

Direct memory access (DMA) -

It is a feature of computer system that allows certain hardware subsystems to access main system memory (RAM) independently of the CPU.

Types of Operating Systems

Batch OS - It is the first OS for 2nd generation computers. This OS does not ~~directaly~~ directely interact with the computer. Instead, an operator takes up similar jobs and groups them together into a ~~bactch~~ batch and then these batches are executed one by one based on FCFS.

Distributed OS - In a distributed OS various computers are connected through a single communication channel. In this a user can access files that are not present on his system but another connected system.

Page No.

Date ___/___/_____

Multitasking OS – The multitasking OS is also known as the time-sharing OS as each task is given some time so that all the tasks work efficiently. This system provides access to a large number of users, and each user gets the time of CPU as they get in a single system.

Network OS – Network OS are the systems that run on a server and manage all the networking functions. They allow sharing of various files, applications, printers, security and other networking functions over a small network of computers like LAN or any other private network.

Real-Time OS – Real-Time OS serve real time system. These OS are useful where many events occur in a short time or certain deadlines, such as real-time simulations.

Types of the Real-time OS –

- Hard real-time OS
- Soft real-time OS

Mobile OS – It is an OS for smartphones, tablets and PDA's. It is a platform on which other applications can run on mobile devices.

Page No.