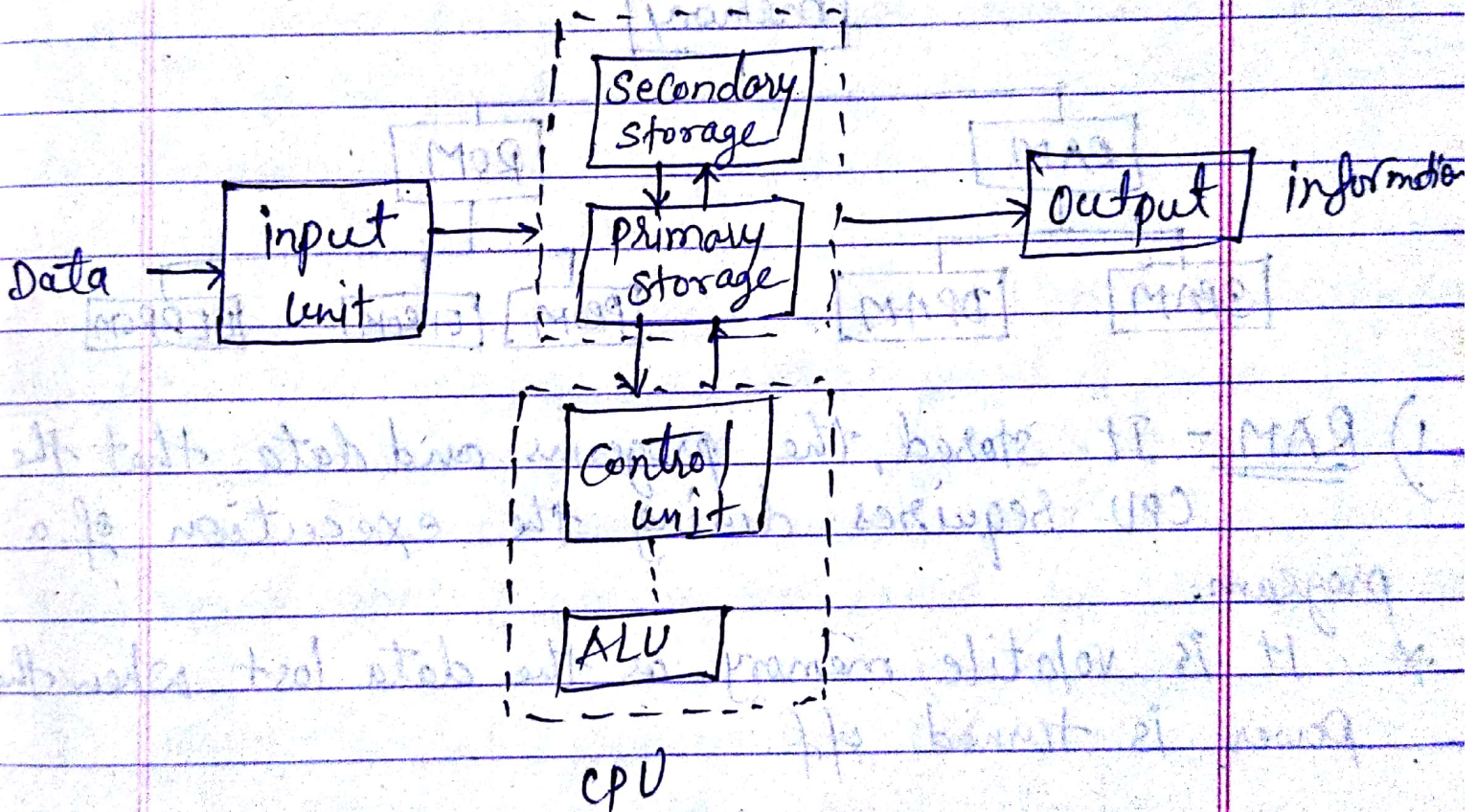
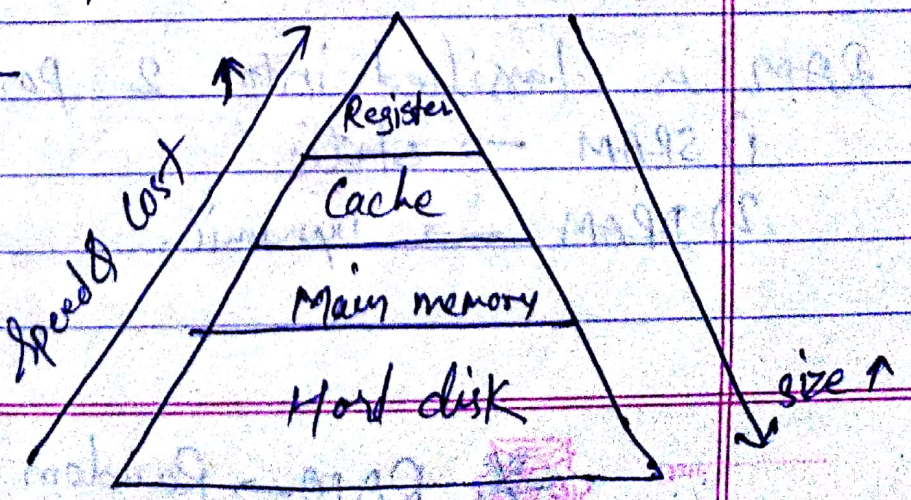
# Operating System

**Computer** - * General purpose device that can execute sequence of instruction presented in a formal format to perform numerical calculations and other tasks.

**Block diagram of Computer**




**Memory hierarchy**-

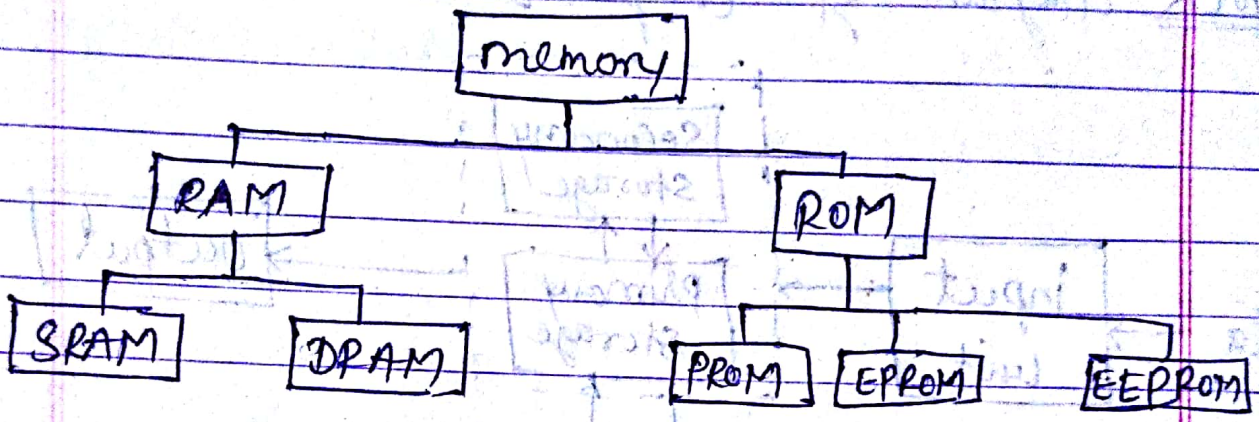


Memory - Two types - → primary memory

1) a) RAM — Random access memory
   b) ROM — Read only memory.

→ Memory is most essential element of a computing system because without it computer can't perform simple task.

2) secondary memory - (hard drive, CD etc)




1) RAM - It stored, the progeams and data that the CPU requires during the execution of a progeam.

* It is volatile memory as the data lost when the power is turned off.

* RAM is classified into 2 part
   1) SRAM → static
   2) DRAM → Dynamic.

* RAM → Random Access Memory OR internal OR volatile OR Main memory OR primary memory OR read-write memory or physical memory

| DRAM | SRAM |
|---|---|
| 1) ~~Constructed~~ used for main memory | 1) Used for cache. |
| 2) Inexpensive | 2) expensive |
| 3) Use less power | 3) use more power |
| 4) Generate less heat | 4) Generate more heat |
| 5) Slower than SRAM | 5) Faster than DRAM |

2) ROM - Store crucial information essential to operate the system, like the program essential to boot the computer.

* Always retains its data.
* Used in embedded system, calculators, peripheral device
* ROM is classified into 3 types -
  1) PROM → (Programmable Read-only memory.)
     ↳ It can be programmed by the user. Once programmed, data & instruction cannot be changed.

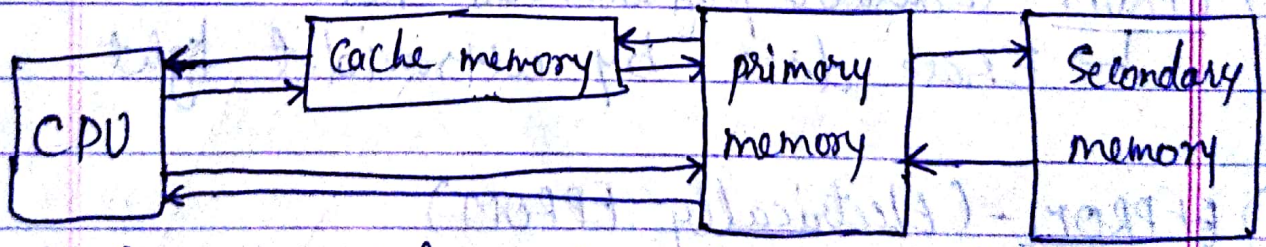
  2) EPROM - (Erasable programmable ROM)
     ↳ Erase data by ultraviolet light.

  3) EEPROM - (Electrically EPROM)
     ↳ Erase data by electricfied. only erase chip portion.

| RAM | ROM |
|---|---|
| 1) Temporary storage. | 1) permanent storage. |
| 2) Store data in MBs. | 2) Store data in GBs. |
| 3) Volatile | 3) Non-volatile |
| 4) Used in normal operation. | 4) Used for startup process of computer. |
| 5) Writing data is faster. | 5) writing data is slower. |

**Cache Memory** — It is special very high speed memory. It is used to speed up and synchronizing with high-speed CPU.

* Costlier than main memory or disk memory.
* Acts as buffer b/w RAM and the CPU.
* It holds frequently requested data and instruction so that they are immediately available to the CPU when needed.
* used to reduce the average time to access data from the main memory.

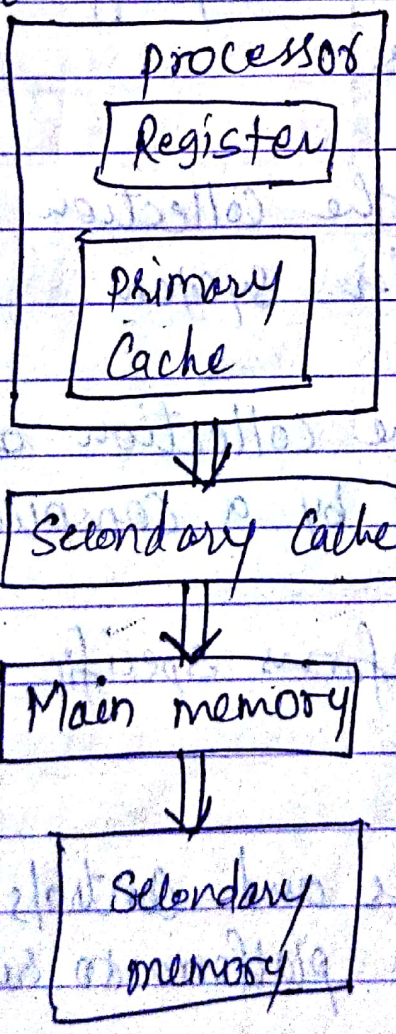



* The performance of cache memory is frequently measure in term of a quantity called **Hit ratio**.

Hit ratio = $\frac{\text{hit}}{(\text{hit} + \text{miss})} = \frac{\text{no. of hits}}{\text{total accesses}}$

Register Memory - Register memory is the smallest and fastest memory in a computer.

* It is not a part of main memory.
* It is located in the CPU in the form of registers.
* A register temporarily holds frequently used data, instructions and memory address that are used by CPU.
* All data is required to pass through registers before it can be processed.
* Register hold a small amount of data around 32 bits to 64 bits.



| Data | Information |
|---|---|
| 1) Data is collection of raw, unorganised facts. | 1) Information is the processed, organised and structured data |
| 2) Data is individual unit. | 2) Information is a group of data. |
| 3) Data doesn't depend on information. | 3) Information depends on data. |
| 4) Raw data alone is insufficient for decision making. | 4) Information is sufficient for decision making. |

* Computer science is the study of computer system and computing processes.

* Computer Hardware is the collection of all physical elements of the computer system.

* Computer system is the collection of all program stored in and executed by a computer system.

* Application software performs specific task for the user.

* System software operates and controls the computer system, and provides a platform to run application software.

* First OS - ATLAS
(manchester University.
1950s - 1960s

Operating System - An Operating System is an interface between a computer user and computer hardware.

An Operating System -
- manages the computer hardware
- facilitates execution of application programs
- acts as an intermediary between the user and the computer hardware
- designed to be convenint and efficient


| Applicatn programs |
|---|
| Operating system |
| Computer hardware |

* An Operating system acts as an - Resource allocator
- Control program

OS Generation - 1) First Generation (1945-1955)
→ vaccum tubes and plugboards

2) Second Generation - (1955-1965)
- Transistors and Batch System.

3) Third generation - (1965-1980)
- Integrated Circuits and multiprogramming.

4) Fourth Generation - (1980 - present)
- Personal Computer

# Types of Operating System-

1)- Batch OS - Some computer process are very lengthy and time-consuming. To speed the same-process, a job with a similar type of needs are batched together and run as a group.

* Users of a batch OS never directly interacts with the computer.

2) Multi-Tasking / Time-sharing OS:- Time sharing OS enables people located at a different terminal (shell) to use a single computer system at same time. * Response time minimal.
* switching is very fast.

3) Network OS:- Network OS runs on a Server. It provides the capability to serve to manage data, user, groups, security, application, and other networking functions.

4) Multiprogramming OS - It increases CPU utilization by keeping multiple jobs in the memory so that the CPU always has one to execute

After one page →

micro kernel →
example - Mach, L4, K42 etc.

Kernel - A Kernel is that part of the operating system which interacts directly with the hardware and performs the most crucial tasks.

* Main layer b/w os & hardware
* It is a central component of an OS that manages operations of computer and hardware.
* Kernel loads first into memory when an OS is loaded and remains into memory until OS is shut down again.
* It decides which process should be allocated to processor to executed and which process should be kept in main memory to execute.

Objective of Kernel - 1) To establish communication between user level application and hardware.
2) To decide state of incoming processes.
3) To control disk managmnt, memory & task management.

Types of Kernel - 1) Monolithic Kernel - It is a type of Kernel where all os service operate in Kernel space. It has dependencies b/w system components. It has huge lines of code which is complex.
e.g. Unix, Linux etc.

2) Micro-Kernel - smaller in size than a conventional Kernel and supports only the core os functionalities.
* It has virtual memory and thread scheduling.

3) Hybrid Kernel - It is the combination of both monolithic kernel and microkernel. It has speed and design of monolithic kernel and modularity and stability of microkernel.
e.g - Window NT, Netware etc.

4) Exo - kernel - It is the type of kernel which follows end-to-end principle. It has fewest hardware abstractions as possible. It allocates physical resources to applications.
e.g. - EXOS, Nemesis etc.

5) Nano kernel - It offers hardware abstraction but without system services.
* Micro-Kernel and Nano Kernel have become analogous.
e.g. - EROS etc.

# Shell - Also known as a command interpreter, is that part of the os that receives command from the users and gets them executed.

Remaining -

5) Time sharing OS - Time-sharing system require interaction with the user to instruct the OS to perform various task. The OS responds with an output. The instruction are usually given through an input device like the keyboard.

Date
Page No

5) Real time OS - Real-Time OS are usually built for dedicated system to accomplish a specific set of tasks within deadlines.

Process - A process is a program under execution. The value of the program counter indicates the address of the next instruction of the process being executed. Each process is represented by a process Control Block (PCB).

Process Scheduling : is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.

1) Arrival Time - Time at which the process arrives in the ready queue.

2) Completion Time - Time at which process completes its execution.

3) Burst Time - Time required by a process for CPU execution.

4) Turn Around Time - Time difference b/w turn around time and burst time.

5) Waiting Time (WT) - Time difference b/w turn around time and burst time.

Waiting Time = Turnaround Time - Burst Time.

Date

Page No

**Thread** - A thread is a lightweight process and forms the basic unit of CPU utilization. A process can perform more than one task at the same time by including multiple threads.

* A threads has its own program counter, register set, and stack.
* A thread shares resources with other threads of the same process the code section, the data section, files and signals.

**Note** -

A new thread, or a child process of a given process, can be introduced by using the fork() system call. A process with n fork() system call generates $2^n - 1$ child processes.

There are two types of threads:

* User threads (user threads are implemented by user)
* Kernel threads (Kernel thread are implemented by OS)

**Scheduling Algorithms** - Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms.

→ There are six popular process scheduling algorithms.

1) First come first serve
2) Shortest job first/Next
3) Shortest Remaining Time
4) Round Robin scheduling.
5) priority scheduling.
6) Multiple-level Queues.

(Non-preemptive sch. algo)

1) First Come First Serve (FCFS): Simplest scheduling algorithm that schedule according to arrival times of processes.
→ FCFS scheduling algorithm state that the process that request the CPU first is allocated the CPU first.
→ It is implemented by using the FIFO queue.
(Non-preemptive)

2) Shortest Job First (SJF) - Process which have the Shortest burst time are scheduled first.
→ If two processes have the same bust time then FCFS is used to break the tie. ~~It is a non-~~

3) Shortest Remaining Time First (SRTF): It is a preemptive mode of SJF algorithm in which jobs are scheduled according to the shortest remaining time.

4) Round Robin (RR) scheduling :- Each process is assigned a fixed time in cyclic way.
→ The ready queue is treated as a circular queue.
→ To implement Round Robin scheduling, we keep the ready queue as a FIFO queue of processes.
→ New processes are added to the tail of the ready queue.

Date
Page No

(Non-preemptive)

5) Priority Based Scheduling :- In this scheduling, processes are scheduled according to their priorities, i.e., highest priority process is scheduled first.
→ If priorities of two processes match, then scheduling is according to the arrival time.

6) Multilevel-Queue scheduling (MLQ)- According to the priority of the process, processes are placed in the different queues.
→ Generally high priority processes are placed in the top level queue. Only after completion of processes from the top level queue, lower level queued processes are scheduled.

7) Highest Response Ratio Next (HRRN): In this scheduling, ~~processes are scheduled according to their priorities, i.e., the~~ with the highest response ratio are scheduled. This algorithm avoids starvation.

Response Ratio = (waiting time + Burst time) / Burst time

Starvation - is a phenomenon associated with the priority scheduling algorithms, in which a process ready to run for CPU can wait indefinitely because of low priority.

8) Multilevel Feedback Queue (MLFQ) Scheduling:
It allows the process to move in between queues. The idea is to separate processes according to the characteristics of their CPU bursts.
→ If a process uses too much CPU time, it is moved to a lower-priority queue.

# The Critical Section Problem:-

1) Critical Section - The portion of the code in the program where shared variables are accessed and/or updated.

2) Remainder Section - The remaining portion of the program excluding the Critical section.

3) Race around Condition - The final output of the code depends on the order in which the variables are accessed. This is termed as the race around condition.

A solution for the critical section problem must satisfy the following three condition -
1) Mutual Exclusion
2) Progress
3) Bounded waiting →

1) Mutual Exclusion - If a process Pi is executing in its critical section, then no other process is allowed to enter into the critical section.

2) progress - If no process is executing in the critical section, then the decision of a process to enter a critical section cannot be made by any other process that is executing in its remainder section. The selection of the process cannot be postponed indefinitely.

3) Bounded Waiting - There exists a bound on the number of times other processes can enter into the critical section after a process has made a request to access the critical section and before the request is granted.

# Synchronization Tools:

1) Semaphore - Semaphore is a protected variable or abstract data type that is used to lock the resource being used. The value of the semaphore indicates the status of a common resource

→ There are two types of semaphores:

Binary Semaphores - Binary semaphores take only 0 and 1 as value and are used to implement mutual exclusion and synchronize concurrent processes.

Counting Semaphores - A counting semaphore is an integer variable whose value can range over an unrestricted domain.

Mutex - A mutex provides mutual exclusion, either producer or consumer can have the key and proceed with their work. As long as the buffer is filled by the producer, the consumer needs to wait, and vice versa.
→ At any point of time, only one thread can work with the entire buffer. The concept can be generalized using semaphore.

Deadlocks - A situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.
Deadlock can arise if following four conditions hold simultaneously (Necessary conditions):

1) Mutual Exclusion
2) Hold and Wait
3) No preemption
4) Circular wait

1) Mutual Exclusion - One or more than one resource is non-sharable (only one process can use at a time)

2) Hold and Wait: A process is holding at least one resource and waiting for resources

3) No preemption - A resource cannot be taken from a process unless the process releases the resource.

4) Circular wait - A set of processes are waiting for each other in circular form.

Methods for handling deadlocks: There are three ways to handle deadlock

1) Deadlock prevention or avoidance: The idea is to not let the system into a deadlock state.

2) Deadlock detection and recovery: Let deadlock occur, then do preemption to handle it once occured.

3) Ignore the problem all together - If deadlock is very rare, then let it happen and reboot the system. This is the approach that both Windows and UNIX take.

# Banker's Algorithm - It is used to avoid deadlock.
→ It is one of the deadlock-avoidance methods. It is named as Banker's algorithm on the banking system where a bank never allocates available cash in such a manner that it can no longer satisfy the requirements of all of its customers.

# Memory Management - These techniques allow the memory to be shared among multiple processes.
→ Overlays - The memory should contain only those instruction and data that are required at a given time.
→ Swapping - In multiprogramming, the instruction that have used the time slice are swapped out from the memory.

#

# Techniques:

(a) Single partition Allocation Schemes - The memory is divided into two parts. One part is kept to be used by the OS and the other is kept to be used by the users.

b) Multiple partition schemes -

1) Fixed partition - The memory is divided into fixed size partitions.

2) Variable partition: - The memory is divided into variable sized partitions.

Note: Variable partition allocation schemes:

1. First Fit - The arriving process is alloted the first hole of memory in which it fits completely.

2. Best Fit - The arriving process is allotted the hole of memory in which it fits the best by leaving the minimum memory empty.

3. Worst fit - The arriving process is allotted the hole of memory in which it leaves the maximum gap.

Note - * Best fit does not necessarily give the best results for memory allocation.

* The cause of external fragmentation is the condition in fixed partitioning and variable partitioning saying that the entire process should be allocated in a contigous memory location. Therefore paging is used.

1) Paging - The physical memory is divided into equal sized frames. The main memory is divided into fixed size pages. The size of a physical memory frames is equal to the size of a virtual memory frame.

2) Segmentation - Segmentation is implemented to give users a view of memory. The logical address space is a collection of segments. Segmentation can be implemented with or without the use of paging.

# Page Fault - A page fault is a type of interrupt raised by the hardware when a running program accesses a memory page that is mapped into the virtual address space, but not loaded in physical memory.

# Page Replacement Algorithms-

1) First in First Out- This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced, the page in the front of the queue is selected for removal.

E.g.- Consider page reference string 1, 3, 0, 3, 5, 6 and 3 page slots. Initially all slots are empty, so when 1, 3, 0 come they are allocated to the empty slots → 3 page faults. When 3 comes, it is already in memory so → 0 page fault. Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e. 1 → 1 page fault. Finally, 6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 → 1 page fault.

Belady's anomaly: Belady's anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the first in first out page replacement algorithm. e.g:- if we consider reference string (3 2 1 0 3 2 4 3 2 1 0 4) and 3 slots, we get 9 total page faults, but if we increase slots to 4, we get 10 page faults.

2. Optimal Page Replacement - In this algorithm, pages are replaced which are not used for the longest duration of time in the future.

3. Least Recently Used (LRU) - In this algorithm, the page will be replaced with the one which is least recently used.

# Disk Scheduling:- Disk scheduling is done by os to schedule I/O requests arriving for disk. Disk scheduling is also known as I/O scheduling.

1) Seek Time - Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or written.

2) Rotational latency: Rotational Latency is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads.

3) Transfer time - Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transfered.

4) Disk Access Time: Seek Time + Rotational Latency + Transfer Time.

5) Disk Response Time: Response Time is the average of time spent by a request waiting to perform its I/O operation. Average Response time is the response time of all request.

## Disk Scheduling Algorithms -

1) FCFS - FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue.

2) SSTF - (Shortest seek time first), requests having the shortest seek time are executed first. So, the seek time of every request is calculated in advance in a queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first.

3) SCAN - In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of the disk, it reverses its direction and again services the requests arriving in its path. So this algorithm works like an elevator and hence is also known as elevator algorithm.

4) CSCAN: In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.

5) LOOK: It is similar to the SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk

6) CLOOK: As LOOK is similar to SCAN algorithm, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end's last request. Thus, it also prevents the extra delay which occured due to unnecessary traversal to the end of the disk.

* Re-entrancy - It is a very useful memory saving technique that is used for multi-programmed time sharing system. It provide functionality that multiple users can share a single copy of a program during the same period. It has two key aspects: The program code cannot modify itself and the local data for each user process must be stored separately.

* Demand paging - specifies that if an area of memory is not currently being used, it is swapped to disk to make room for an application's need.

* RAID - stand for Redundant Array of Independent Disks. It is used to store the same data redundantly to improve the overall performance. There are 7 RAID levels.

* Fragmentation - is a phenomenon of memory wastage. It reduces the capacity and performance because space is used inefficiently.
1) Internal frag. - It occurs when we deal with the systems that have fixed size allocation units.

2) External fragmentation - It occurs when we deal with system that have variable-size allocation units.

* Spooling - is a process in which data is temporarily gathered to be used and executed by a device, program or the system. It is associated with printing.

# Aging is a technique used to avoid starvation in the resource scheduling system.

# Thrashing is a phenomenon in virtual memory schemes when the processor spends most of its time in swapping pages, rather than executing instructions.

Date .....

Page No .....