**Operating System:** A program that acts as an intermediary between a user of a computer and the computer hardware.

**Interrupt Handling:**  An interrupt can be generated by a device or a program to inform the os to halt its current activities and focus on something else.

**Direct Memory Access:** Device controller transfers block of data from buffer storage directly to main memory without cpu intervention.

**RAM, ROM , DRAM, ,Main Memory , Secondary Storage , Magnetic disks**

**Disk Controller:** determines the logical interaction between the device and the computer.

**Storage Hierarchy:** Based on Speed, Cost,Volatility.

**Bootstrap:** When it is powered up or rebooted it needs to have an initial prgm to run. [stores -ROM]

**Types of OS:**

* Batch: Do not interact with comp directly. Operator takes similar jobs, group them in batch.

Advantages: Multiple users can share batch system, idle time is very less.

Disadvantage: Batch systems are hard to debug, costly

* Multi programmed: Sharing the processor, when two or more programs reside in memory at the same time.

Advantage: High and efficient cpu utilization

Disadvantage: cpu scheduling is required. ,Management is required.

* Timesharing: [Multi tasking] Each task is given some time to execute. So works smoothly.

Advantages: each task get equal opportunity, cpu idle time reduced, No duplication- sw

Disadvantages: Reliability problem , Take care of security and data , Data com probem.

* Multiprocessor: refers to use two or more cpu with a single computer system.

Advantage: More reliable, Enhanced throughput, more economic system

Disadvantage: more expence , complicated os required, large main memory required.

* Distributed: Independent systems possess their own memory unit and cpu. [loosely coupled

Advantage: Failure of one system does not effect other , increases data exchange speed, load on host comp reduces, delay in data processing reduces.

Disadvantages: Failure of main network will stop entire communication

* Network : These system run on server and provide the capability to manage data and other networking functions.

Advantage: Highly stable, Security

Disadvantage: servers are costly , maintenance and update are required.

* Real-time : These type of OS’s serves the real time systems – hard RTS , Soft RTS

Advantage: error free , focus on running application , task shifting

Disadvantage: limited tasks, use heavy system resources.

**Operating System services:** User interface, prog execution, i/o operations , file-system manipulation, communication , error detection

**System calls:** process control, file manipulation, device manipulation, communication, security/protection.

**OS structure:** simple/ monolithic structure, layered approach, microkernel, modules.

**PROCESS:** a program in execution.

**Program:** passive entity – file containing list of instructions stored on disk i.e. executable file

* a program is a passive entity and process is active entity[with a program counter specifies next instruction to execute].

**Process State:** The state of a process is defined in part by current activity of the process.

1. New: the process is being created.
2. Running: instructions are being executed.
3. Waiting: process is waiting for some event to occur.
4. Ready: the process is waiting to be assigned to a process.
5. Terminated: finished execution

**Process control block: [**task control block**]**

**Process Scheduling: objective-**process running all the time🡪 to increase cpu utilization.

**Scheduling Queues:**

1. Job Queue: As the process enters the system, they put in job queue. This keeps are process in system.
2. Ready Queue: The processes that are residing in main memory and are ready and waiting to execute are kept on a list.
3. Device Queue: The list of processes waiting for a particular i/o device .

**Schedulers:**

1. Long-term : [job scheduler] – it selects processes from the queue and loads them into memory for execution.
2. Short-term: [CPU scheduler] – selects the processes that are ready to execute & allocate the cpu to one of them
3. Medium-term: the process is swap-out and later swap-in.

**Context Switching:** **Context Switching** involves storing the **context** or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier.

**Thread:** flow of execution. 🡪 User level thread , kernel level thread.

**Multithreading:** **multithreading** is the ability of a processor to provide **multiple threads** of execution concurrently.

**CPU Scheduling:**

1. CPU – I/O burst cycle
2. CPU Scheduler: Whenever the CPU becomes idle , the OS must select one of the processes in the ready queue to be executed.

Selection carried out by short term scheduler (CPU scheduler)

1. Preemptive Scheduling:  processor stops the execution of the process and switches to another process irrespective of I/O

Operation i.e. *running state to ready* (or) *waiting state to ready*

round-robin (RR), priority, sjf, SRTF (shortest remaining The time first).

Non- preemptive: processor execute the entire process and terminates . process which is from Running to waiting state

1. Dispatcher: The dispatcher is the module that gives control of the CPU to the process selected by short term scheduler

**Scheduling criteria:**

* CPU Utilization:
* Throughput: The no. of process that are completed per unit time.
* Turnaround time: The interval from the time of submission of the process to the time of completion i.e. process waiting to get into memory **[TA=CT-AT]**
* Waiting Time: sum of periods spent waiting in the ready queue**.[WT=TAT-BT]**
* Response Time: The time from the submission of the request until first response produced.
* AT : time at which the process arrives in the ready queue.
* CT: time at which the process computes its execution.
* BT: time required by a process for CPU execution.

**CPU Scheduling Algorithms:**

First Come First Serve Scheduling :  the process which arrives first, gets executed first

Disadvantage: **Convoy Effect** is a situation where many processes, who need to use a resource for short time are blocked by one process holding that resource for a long time.

poort utilization of resources and hence poor performance.

Shortest Job First scheduling: works on the process with the shortest **burst time**  -- Non Pre-emptive &Pre-emptive

Advantage: It gives minimum avg waiting time. \* Disadvantage: **Starvation**, where a shorter process has to wait for a long time until the current longer process gets executed.

Priority Scheduling: each process is associated with priority, the CPU is allocated to process with highest priority. – both

Disadvantage: **indefinite blocking** or **starvation**.--- **aging**

Round Robin Scheduling: A fixed time is allotted to each process, called **quantum**, for execution.

Once a process is executed for given time period that process is preemptied and other process executes for given time period.

multi-level queue scheduling algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority, or process type

Processes do not move between queues. This setup has the *advantage* of low scheduling overhead, but the disadvantage of being inflexible.

Multilevel feedback queue scheduling: allows a process to move between queues.  If a process uses too much CPU time, it will be moved to a lower-priority queue. Similarly, a process that waits too long in a lower-priority queue may be moved to a higher-priority queue. This form of aging prevents starvation.

**Inter process communication:** is a mechanism that allow the cooperative process to exchange data. Shared memory & message passing

Independent process: a process which cannot affect by other process executing in system , Cooperative process: which effects

**Synchronization:** Process **Synchronization** means sharing system resources by processes in a such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data. [it ensures only 1 process acquire resource at a time]

It controls the execution of processes running concurrently to ensure that consistent results are produced.

**Critical section** is a section of the program where a process access the shared resources during its execution.

**Race condition** is a situation where-The final output produced depends on the execution order of instructions of different processes.

**Critical Section Problem:**If multiple processes access the critical section concurrently, then results produced might be inconsistent.

**Synchronization mechanisms** allow the processes to access critical section in a synchronized manner to avoid the inconsistent results.

**Criteria For Synchronization Mechanisms-**

-Mutual Exclusion 🡪 No other process can enter the critical section until the process already present inside it completes.

-Progress🡪 A process can freely enter inside the critical section if there is no other process present inside it.

-Bounded Wait🡪 The wait of a process to enter the critical section is bounded.

A process gets to enter the critical section before its wait gets over.

-Architectural Neutral 🡪 There is no dependency on the architecture.

**Semaphores in OS-**

A semaphore is a simple integer variable. 2 standard atomic operations wait() and signal().

It is *used to provide synchronization* among multiple processes running concurrently.

1. Counting Semaphores : A counting semaphore has two components-

An integer value

An associated waiting list (usually a queue)

1. Binary Semaphores or Mutexes : A binary semaphore has two components-

An integer value which can be either 0 or 1

An associated waiting list (usually a queue)

**Classic Problems of Synchronization:**

1. Bounded Buffer (Producer-Consumer) Problem

where a **finite** buffer pool is used to exchange messages between producer and consumer processes.

Because the buffer pool has a maximum size, this problem is often called the **Bounded buffer problem**.

1. Dining Philosophers Problem

There are five philosophers sitting around a table, in which there are five chopsticks/forks kept beside them and a bowl of rice in the centre, When a philosopher wants to eat, he uses two chopsticks - one from their left and one from their right. When a philosopher wants to think, he keeps down both chopsticks at their original place.

1. The Readers Writers Problem

there are some processes(called **readers**) that only read the shared data, and never change it, and there are other processes(called **writers**) who may change the data in addition to reading, or instead of reading it.

Monitor is one of the ways to achieve Process synchronization. The monitor is supported by programming languages to achieve mutual exclusion between processes.

**Deadlock** is a situation where each of the computer process waits for a resource which is being assigned to some another process. In this situation, none of the process gets executed since the resource it needs, is held by some other process which is also waiting for some other resource to be released.

**Necessary conditions for Deadlocks**

**🡪Mutual Exclusion:** It implies, if two process cannot use the same resource at the same time.

**🡪Hold and Wait:** A process waits for some resources while holding another resource at the same time.

**🡪No pre-emption:** The process which once scheduled will be executed till the completion. No other process can be scheduled by the scheduler meanwhile.

**🡪Circular Wait**: All the processes must be waiting for the resources in a cyclic manner so that the last process is waiting for the resource which is being held by the first process.

**Methods for Handling Deadlock:**

1. Deadlock Prevention🡪 me, h&w, pre-emption , circular wait. [atleast one we cannot hold, we can prevent deadlock]
2. Deadlock Avoidance🡪Ensure that system never enter a deadlock state

Single instance of a resource type: resource allocation graph

Multiple instances of a resource type**: Banker’s algorithm** 🡪It follows the safety algorithm to check whether the system is in a safe state or not.

1. Deadlock detection and recovery

Single instance: convert resource – allocation graph to a wait-for graph

Several instance

1. Deadlock ignorance

**File:** A file is a named collection of related information recorded on secondary storage.

**File Attributes:** Name , identifier, type, locations, size, protection, time, data and user identification

**File Operations:** Creating a file, writing a file, Read a file, Repositioning within a file, Delete a file, Truncating a file.

**File Access Method:**

1. Sequential Access: Information in the file, is pressed in order, one record after another.read-next,writing-next,rewind,skin n rec
2. Direct Access: A file is made up of fixed length.Logical records that allow to r & w in no particular order.
3. Indexed Access: It contains key and pointer.

**File Directory:** A physical disk can be broken into multiple partitions or mini- disks in any OS.

**Operations on Directory:** Search for file, Create new file, Delete file, List a directory, Rename file.

* **Single level directory:** All the files are contained in same directory
* **Two-level directory:** create a directory foe each user
* **Tree-level directory:** allows users to create their own sub-directories
* **Acyclic graph Directory:** useful when the same files need to be accessed in more than one place in directory strct.

**File Allocation Methods:**

1. Contiguous Allocation: Each file occupies a set of contiguous addresses on disk.

First-fit🡪 Allocates first hole that is big enough.

Best-fit🡪 Allocates the smallest hole that is big enough.

Worst-fit🡪 Allocates the largest hole

1. Linked Allocation: No external fragmentation. [in sequential access file]
2. Indexed Allocation: All the pointers are briught together into one location called index block

**Memory Management:** Memory management 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.

**Memory protection:** Base reg🡪holds thr smallest legal physical memory address

Limit reg🡪 specifies the size of the range

**Logical address:** address generated by cpu

**Physical address:** address seen by memory unit.

**Logical address space:** set of all logical address generated by program.

**MMU:** The run-time mapping from virtual to physical address is done by hardware device called mmu

**Swapping:** A process however can be swapped temporaily out of memory to a backing store and then brought back into memory for continued execution.

**Fragmentation:** Fragmentation is the inability to reuse memory that is free

**External fragmentation:** occur when enough free memory is available but isn’t contiguous.

**Internal Fragmentation:** arises when a large enough block is allocated but it is bigger than needed.

**Paging:** Paging is a memory management scheme that permits the physical address space of space to be non-contiguous.

Avoid external fragmentation.

**Frames:** Breaking physical memory into fixed sized blocks,

**Pages:** Breaking logical memory into blocks of same size.

🡪address generated by cpu divides into 2 parts: page no. and page offset.

**Translation look-aside buffer:** A Translation look aside buffer can be defined as a memory cache which can be used to reduce the time taken to access the page table again and again.[**associative memory**]

**Effective Access Time (EAT):** hit ratio (search TLB + Access time for MM) + (1-hitratio)(search TLB+AMM(PT)+AMM)

**Size of MM = 2n** x size of location

🡪**Heirarichal page table , Hashed page table , Inverted page table**

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

**Virtual Memory:** The virtual memory technique allows users to use more memory for a program than a real memory of a computer.

Virtual memory is a technique that allows execution of the process that are not completely in memory

Demand Paging: lly to paging system with swapping.

**Page fault:** When the process tries to access a page and the page was not brought into memory.

**🡪Page replacement:** a page fault occurs and we need to bring the desired page into memory. There are no free frames

**First in First out**:It replaces the oldest page that has been present in the main memory for the longest time.

**Optimal Replacement Algorithm:** Replace the page that will not be used for the longest period of time.

**Least Recently Used:** Replace the page that has not been used for the longest period of time.

**Allocation of frames:** Fixed allocation🡪 equal allocation , proportional allocation

**Priority allocation🡪**

**Global replacement:** process selects a replacement frame from the set of all frames, one process can take frame from another.

**Local replacement:** each process selects from only its own set of allocated frames.

**Thrashing:** a process is busy swapping pages in and out. If a process does not have enough pages the page fault rate is very high.

I/O hardware:.

**Polling: [**device handshaking**]** The process of periodically checking status of the device to see if it is time for the next I/O operation, is called polling

**Interrupt**: A device controller puts an interrupt signal on the bus when it needs CPU’s attention when CPU receives an interrupt, It saves its current state and invokes the appropriate interrupt handler

**Disk Scheduling:** The algorithms used for disk scheduling are called as **disk scheduling algorithms**.

The purpose of disk scheduling algorithms is to reduce the total seek time.

**FCFS:** this algorithm entertains requests in the order they arrive in the disk queue.

**Advantages-**

It is simple, easy to understand and implement.

It does not cause starvation to any request.

**Disadvantages-**

It results in increased total seek time.

It is inefficient.

**Shortest Seek Time First**. : Select the request with min seek time from current head.

Advantages: It reduces the total seek time as compared to [**FCFS**](https://www.gatevidyalay.com/disk-scheduling-disk-scheduling-algorithms/).

It provides increased throughput.

Disadvantage: Cause starvation for some reqs,There is an overhead of finding out the closest request.

**Scan:** Head starts from one end of the disk and move towards the other end servicing all the requests in between.

After reaching the other end, head reverses its direction and move towards the starting end servicing all the requests in between.

**Advantages-**

It is simple, easy to understand and implement.

It does not lead to starvation.

It provides low variance in response time and waiting time.

**Disadvantages-**

It causes long waiting time for the cylinders just visited by the head.

It causes the head to move till the end of the disk even if there are no requests to be serviced.

**C-Scan:** The head moves from one end of disk to other servicing req as it goes. When it reaches the other end, however it immediately returns to beginning of disk , without servicing any req on return trip.

**Advantages-**

It provides uniform waiting time.

It provides better response time

**Disadvantages-**

It causes more seek movements as compared to SCAN Algorithm.

It causes the head to move till the end of the disk even if there are no requests to be serviced.

**LOOK:**  Head starts from the first request at one end of the disk and moves towards the last request After reaching the last request at the other end, head reverses its direction.

It then returns to the first request at the starting end servicing all the requests in between.

Adavantage: It provides better performance as compared to SCAN Algorithm.

It does not lead to starvation.

Disadvantage: There is an overhead of finding the end requests.

It causes long waiting time for the cylinders just visited by the head.

**C-LOOK**: Head starts from the first request at one end of the disk and moves towards the last request at the other end servicing all the requests in between.

After reaching the last request at the other end, head reverses its direction.

It then returns to the first request at the starting end without servicing any request in between.

**Advantages-**

It does not causes the head to move till the ends of the disk when there are no requests to be serviced.

It reduces the waiting time for the cylinders just visited by the head.

It provides better performance as compared to LOOK Algorithm.

It does not lead to starvation.

It provides low variance in response time and waiting time.

**Disadvantages-**

There is an overhead of finding the end requests.