* Operating System - It is an system program that acts as an intermediary between the user & the computer hardware & controls the execution of all kinds of program.

* objectives or goals of operating System -
① Convenience ② Efficiency ③ Ability to Evolve

* functions of operating System -
① memory managment
② Processor managment
③ Device management
④ File management.
⑤ Security
⑥ Job accounting.

* Types of OS -
① Batch operating System -
(a) It is one of the oldest operating System.
(b) In batch operating system, Jobs with similar need batched together by operator & run as a group on a computer System.
(c) batch operating System is one where programs & data are collected together in a batch before processing starts.
(d) Advantages -
(i) Huge amount of data can be processed efficiently.
(ii) the execution of Job becomes fast & well managed.
(e) Disadvantage -
(i) It is costly.
(ii) It is difficult to debug program.

2) Spooling -

(a) Spooling stands for Simultaneous Peripheral Operation on Line.

(b) It refers to putting data of various I/O jobs in a buffer.

(c) The most common spooling ~~op~~ application is print spooling.

(d) Spooling is also used for processing data at remote sites.

(e) Advantage = The spooling operation uses a disk as a very large buffer.

(f) Disadvantage = There is an extra overhead of maintaining table of card image.

3) Multiprocessor OS -

(a) A multiprocessor os means the use of two or more processors within a single computer system.

(b) Multiprocessing means multiple CPUs perform more than one job at one time.

(c) Types of multiprocessor systems -

(i) Asymmetric

(ii) Symmetric

(d) Advantage - It increased throughput by increasing the number of processors.

(e) Disadvantage - multiprocessor ~~as~~ systems are expensive.

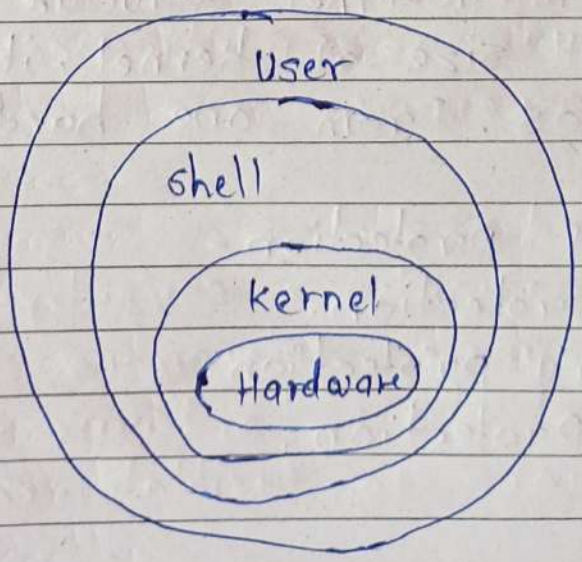
4) Distributed OS -

(a) It is basically a computer network in which two or more autonomous computers are connected their hardware & software interactions to facilitate communication.

(b) In a distributed OS, the users access remote resources in the same manner.
(c) Advantages- Reliability, speed, performance
(d) Disadvantages- Troubleshooting, networking security.

5) Network OS-
(a) A NOS runs on a server & provides the server the capability to manage data, user, security.
(b) The purpose of NOS is to allow shared file & printer access among multiple computers in a network.
(c) peer-to-peer NOS is an OS in which all the nodes are functionally & operational equal to each other.
(d) client-server- NOS operates with a single server & multiple client computers in the network.
(e) Advantage- Security is server managed.
(f) Disadvantage- High cost of buying & running a server.

* operating System structure-



(i) the kernel is the innermost layer & is the central controlling part of the operating system. It is the core of the operating system.

(ii) the shell is the next layer to the kernel A shell is software that provides an interface for users of an operating system.

(iii) A user interact with programs in the user interface typically with the command, interpreter to request use of resources.

(iv) the hardware consists of CPU the main memory, I/o devices, etc. provides the basic computing resources.

(v) (a) Simple structure.
(b) Layered structure.
(c) microkernel structure
(d) monolithic structure.

* Advantages of microkernel -
(a) the microkernel structure provides high security & reliability.
(b) the microkernel structure is portable due to small size of kernel, it is easy to port os from one hardware to other.

* kinds of protection -
(a) I/o protection.
(b) Memory protection.
(c) CPU protection.

* open source OS.
① An open source OS is an operating system whose code has been made publicly & freely available to anyone.
② open source OS are available in source code format rather than compiled binary code.
③ GNU/Linux is most popular open source operating system.
④ FreeBSD is a free open source OS.

* Booting - Booting the system is done by loading the kernel into main memory & starting its execution.

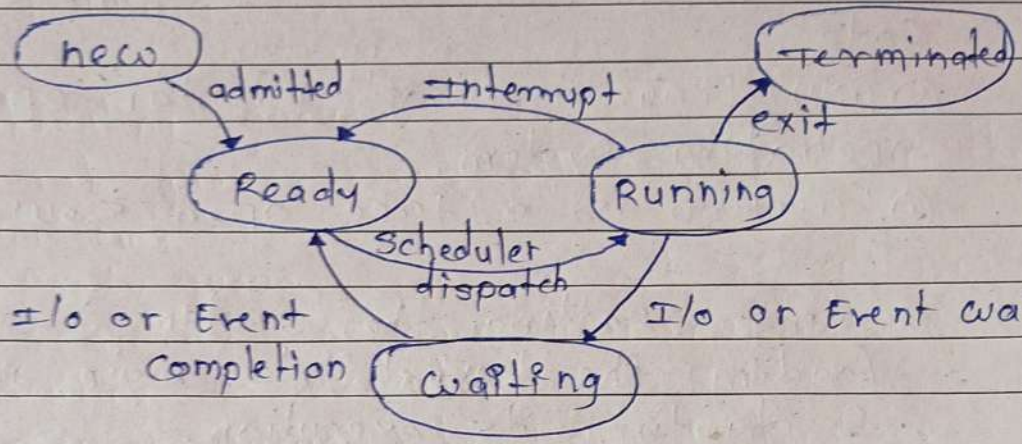
* A small piece of code known as the bootstrap Loader.

* System Call - System call provides an interface between a running program & an operating system.

* Types of System Calls -
① Process control - examples of these ② System calls are end, abort, Load, execute, create, new, process, etc.
② File management - creating files, delete, files, read, write, etc.
③ Device management - Call request device, release device, read & write device, etc.
④ Communication - creating & deleting Connection, send & receive messages.

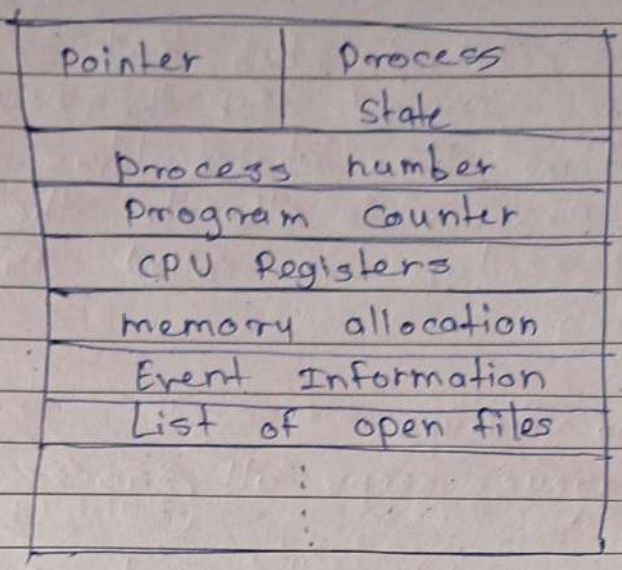
* processes - An entity which represents the basic unit of work to be implemented in the system.

* process states -



① new - A process is said to be new state if it is being created.
② Ready - A process is said to be ready if it is ready for the execution.
③ Running state - A process is said to be in running state if the CPU has been allocated to it & it is currently being executed.
④ Waiting or blocked - A process is said to be waiting if it has been blocked by some event.
⑤ Terminated. It has been terminated abnormally by the OS because of some error.

* Process control block (PCB)



① Pointer - It has an address of the next PCB, whose process state is ready

② Process state - It specifies the current state of the process.

③ Process Number - Each process is identifies by its process number. called PID

④ Priority - the process is assigned the priority at the time of its creation.

⑤ program Counter - It is a pointer indicates the address of the next instruction to be executed.

⑥ CPU registers - various CPU registers like accumulators, index, Stack pointers & general - purpose register, etc.

⑦ CPU scheduling Information - this includes process priority & other scheduling information.

⑧ memory management information - this includes the information of page table, memory Limits, etc.

⑨ File management - It includes information about all open files, access rights, etc.

* process scheduling: when two or more processes compete for the CPU at the same time then choice has to be made which process to allocate the CPU next. this procedure is called Process scheduling.

* Scheduling queues-
  1. Job queue- keeps all processes in the System.
  2. Ready queue- keeps a set of all processes residing in main memory, ~~ready~~ & waiting to execute
  3. Device queue- the list of processes waiting for a pereticuler I/o devices is called device queue.

* Types of scheduler-

Long Term scheduler	short term Scheduler	Medium term scheduler
1. It is a Job scheduler.	It is a CPU scheduler	It is process swapping scheduler
2. It controls the degree of multi-programming.	It provide lesser control over degree of multiprogramming	It reduces the degree of multi-programming.
3. speed is lesser than short term scheduler.	speed is fastest	Speed is in between both short & long term scheduler.
4. It deals with main memory for loading process	It deals with CPU	It deals with main memory for removing processes & reloading whenever required.

* Context switch

① The context switch is an essential features of a multitasking operating system.

② Switching the CPU to another process requires saving the state of the old process & loading the saved state for new process. This task is known as context switch.

③ A context switch is the mechanism to store & restore the state or context of a CPU in process control Block (PCB) so that the ~~resume~~ process execution can be resumed from the same point at a later time.

④ Using context switch technique, a context switcher enables multiple processes to share a single CPU.

* Thread scheduling-

A thread is a flow of execution through the process code, with its own program counter, system registers & stack.

*

	Process	thread
①	It is heavy weight	It is light weight
②	An executing instance of a program is called process	A thread is a subset of process
③	processes have considerable overhead.	threads have almost no overhead.
④	runs in seperate memory spaces.	runs in shared memory spaces.
⑤	It does not share signal handling.	It shares signal handling.

User level threads	Kernel level thread.
① User level threads are faster to create & manage.	Kernel level thread are slower to create & manage
② Implemented by a thread library at the user level.	operating system support directly to kernel threads.
③ User level thread can run on any operating system.	kernel level thread are specific to the operating system.
④ Multithread application cannot take advantage of multiprocessing.	Kernel routines themselves can be multithreaded.

* Benefits ~~of~~

→ ① Resource sharing
② Responsiveness
③ proper utilization of multiprocessor Architecture.
④ Economical
⑤ Efficient communication.

* multithreading models -

① one-to-one model

advantages -

② this model provides more concurrency than the many to one model.
③ It supports multiple threads to execute in parallel on microprocessors.

Disadvantage -

④ Each user thread, the kernel thread is required.
⑤ Creating a kernel thread is overhead.

(2) many to one model-

Advantage-

(a) one kernel thread controls multiple user threads.

(b) Used in Language systems, portable Libraries.

Disadvantage-

(a) Entire process will block if a thread makes blocking System call.

(b) Multiple threads are not able to run in parallel.

(3) many to many model-

Advantage-

(a) many threads can be created as per user's requirement.

(b) provides the best accuracy on concurrency.

Disadvantage-

(a) Low performance.

(b) True concurrency cannot be achieved.

* Thread Libraries- POSIX pthreads, Win32 threads & Java threads.

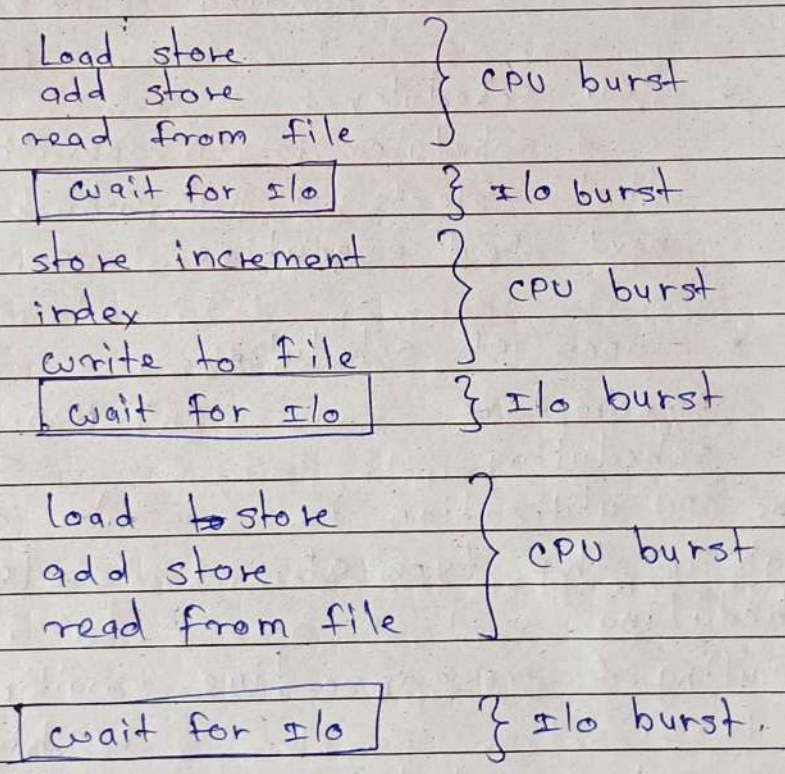
* CPU & I/O Burst Cycle -

① CPU scheduling is greatly affected by how a process behaves during its execution.

② all the processes continue to switch between CPU & I/O devices during their execution.

③ Process execution begins with a CPU burst. It is followed by an I/O burst, which is followed by another CPU burst then another I/O burst & so on.

④ Eventually the last CPU burst will end with a system request to terminate execution, rather than another I/O burst.



* Scheduling Criteria -

① CPU utilization - The percentage of time the CPU is busy in executing processes.

② Throughput - one measure of work is the no. of processes that are completed per time unit called throughput.

③ Turnaround Time (TAT) - the amount of time that has rolled by from the time of creation to the termination of a process.

④ Waiting time - the time spend by a process while waiting in the ready queue.

⑤ Response time - the time elapsed between the moment when a user initiates a request & the instants when the system starts responding to this request.

* CPU scheduler -
A scheduler is an operating system module that selects a job which is to be admitted next for execution.

* Types of scheduling -

Preemptive scheduling	Non-preemptive scheduling
① The CPU utilization is high in this type of scheduling.	The CPU utilization is less efficient.
② waiting & response time is less.	waiting & response time is higher.
③ Preemptive scheduling is flexible.	Non-preemptive scheduling is rigid.
④ Examples are Round Robin & Shortest Remaining Time First.	Examples are First Come First Serve & Shortest Job First.

⑤ the system resources are used efficiently.

the system resources are not used efficiently.

* Dispatcher- It is the module that actually gives control of the CPU to the process selected by the short term scheduler.

* Examples on algorithms FCFS, SJF, priority, RR.

* Race condition.
It is a situation where multiple processes access & manipulate the same data concurrently & the outcome of execution depends on the order in which the instruction executes.

* Critical section problem-
① to avoid race conditions & results, one must identify codes in critical sections in each process.
② Critical section is a code segment that can be accessed by only one process at a time.
③ Critical section contains shared variables which need to be synchronized to maintain consistency of data variables.
④ A system consisting of n processes $\{P_0, P_1, \ldots, P_{n-1}\}$. Each process has a segment of code, called a critical section.
⑤ do
{
Entry section
critical section
Exit section
remainder section
} while (TRUE);

* Peterson solution-
① Peterson's solution is widely used solution to critical section problems.
② this algorithm was developed by a computer scientist Peterson that is why it is named as a peterson's solution.
③ Peterson solution preserves all three conditions.
(a) Mutual Exclusion.
(b) progress.
(c) Bounded waiting.

* Bakery Algorithm solution-
① It is one of the simplest known solutions to the mutual exclusion problem for the general case of n process.
② It is a critical section solution for n processes. the bakery algorithm preserves the First come First Serve (FCFS)
③ the common data structures are used in Bakery algorithm;
Boolean choosing [n];
int number[n];

* Semaphore- A semaphore is a variable used to control access to a common resource by multiple processes & avoid critical section problem.

* Types of semaphore-
① Binary semaphore.
② Counting semaphore.

* Bounded buffer problem -

① producer consumer problem is a classical synchronization problem. we can solve this problem by using semaphores.

② Bounded - buffer problem is also called Producer - consumer problem.

③ solution to this problem is, creating two Counting Semaphores "full" & "empty" to keep track of the current number of full & empty buffers respectively.

④ producer -

ⓐ Create item & adds to the buffer.

ⓑ Do not want to overflow the buffer.

⑤ Consumer.

ⓐ Remove items from buffer.

ⓑ Do not want to get ahead of producer.

* Dining philosopher problem -

① The Dining philosopher is a popular classic synchronization problem for Concurrency control.

② the problem can be stated as follows -

ⓐ Consider five philosophers spend their lives alternating between thinking & eating.

ⓑ They are seated around a circular table. In the centre of table is a bowl of rice & table is laid with five single chopsticks.

ⓒ Each philosopher has access to the chopsticks at his / her left or right. In order to eat, a philosopher must be in possession of both chopsticks.

ⓓ A philosopher may only pick up one chopstick at a time. Each philosopher

attempts to pick up the left & then the right chopstick.
(e) when done eating, a philosopher puts both chopstricks back down on the table & begins thinking.
(f) Since the philosophers are sharing chopsticks, it is not possible for all of them to be eating at the same time.
(g) A solution of the dining philosopher's problem is to use a semaphore to represent a chopstick.
(h) A chopstick can be picked up by executing a wait() operation on the semaphore & released by executing a signal() operation semaphore.

5-

* Address binding - symbolic addresses generated by program must be mapped into the process address space. this mapping is known as address binding.
Types -
① Compile time address binding
② Load time address binding
③ Execution Time or dynamic address binding.

* logical address - An address generated by the CPU is commonly known as logical address.

* physical address - the address seen by the memory unit that is one loaded into the memory address register of the memory called physical address.

* Dynamic Loading - To obtain better memory utilization, routines can be loaded in main memory as required called dynamic loading.

* Swapping - It is a memory management technique 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.

Two tasks -
1. Swap - in
2. Swap - out

* Memory Allocation - It is a process by which computer programs are assigned memory or space to a process.

Two types of approaches -
1. Static (fixed sized) memory partitioning.
2. Dynamic (variable sized) memory partitioning

* Advantages of MVT -
1. It increases degree of multiprogramming due to which there is no unused space in the memory.
2. In MVT space in the main memory is allocated strictly ~~to~~ according to the requirement of the process.
3. In MVT the size of process can grow or shrink dynamically.

* fragmentation - As processes are loaded & removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes cannot be allocated to memory blocks considering their small size & memory blocks remains unused this problem is called fragmentation.

* Compaction - The collection of free space from multiple non-contiguous blocks into one large free block ~~is~~ in a system's memory is called compaction.

Internal fragmentation	External fragmentation
① It occurs when fixed sized memory blocks are allocated to the processes	It occurs when variable sized memory blocks are allocated to processes.
② It mainly refers to the unused space in the partition that resides in the allocated region.	It mainly refers to the unused blocks of the memory that are not contiguous.
③ It can be eliminated by allocating memory to processes dynamically.	It can be eliminated by compaction technique.

* paging - It is a memory management scheme that permits the physical address space of a process to be non-contiguous

* Examples on page replacement algorithms.
① FIFO ② optimal ③ LRU