

Operating System

1) O.S = piece of software that manages all resource of computer.

2) function of OS :-

- 1) resource management
- 2) acts as an interface
- 3) abstraction

goal :-

- 1) max CPU utilization
- 2) ^{no} process starvation
- 3) high priority job exe

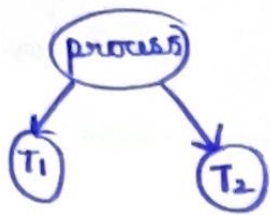
3) Types of OS :-

- single process - ~~no~~ one process executes at one time, no CPU utilization, starvation. X ✓ X
- Batch - dividing into batch wise X ✓ X
- multiprogramming - single CPU, context switching
- multitasking - single CPU, C.S, time sharing
- multiprocessing - more than one CPU, C.S, T.S
- distributed - many interconnected computer
- Real time OS - less error chance and fast execution

4) program :- exe file with set of instruction

process :- program under exe, RAM

⑥ Thread:- process divided into segments and each is indep.



parallel, multiple CPU

⑦ Multitasking

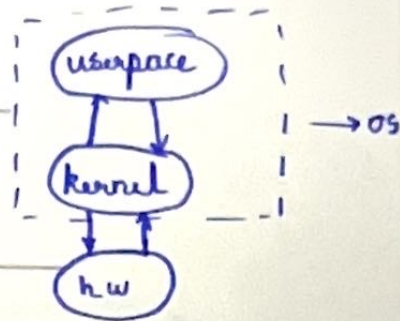
- more than 1 process
- isolation ~~there~~ is provided

Multithreading

- ~~not~~ more than 1 thread
- not there

⑧ Kernel:- interacts with hardware, heart of OS

⑨ userspace:- no hardware access, app runs here

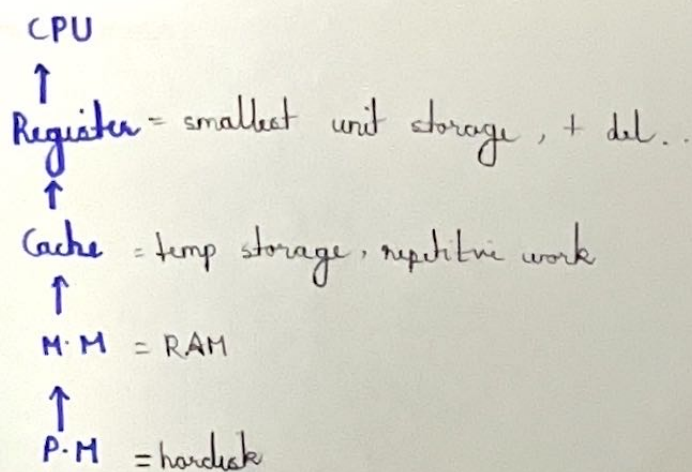


⑩ func of kernel:- ~~to manage~~

- process management
- memory "

⑪ types of OS:- 1) Monolithic - every process present in kernel
2) Micro kernel - major func are present in kernel

⑫ types of storage:-



⑬ Primary Memory

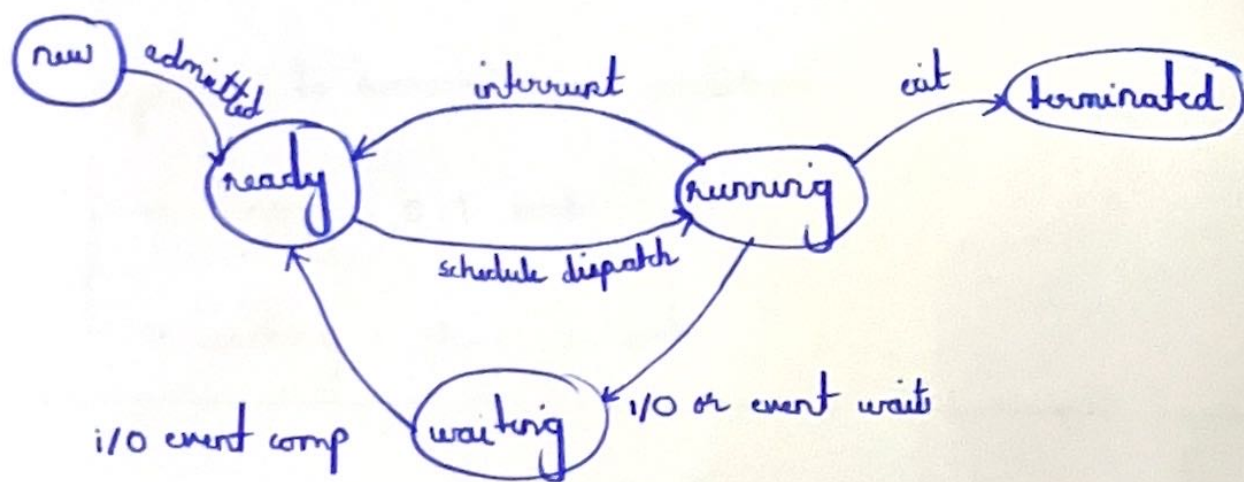
- non volatile

Main Memory

- volatile

⑭ Process state:-

- new :- program to process
- ready state :- process is in memory ready queue
- running state :- CPU allocate
- waiting :- wait for I/O operation
- termination :- process finishes



⑮ orphan and zombie process

16) 32 bits vs 64 bit :-

→ stores less addl
(4 byte)

→ good performance, can do many calculations
(8 byte)

1 byte = 8 bit

17) **Convoy effect** :- smaller process for the big process to get off from the CPU. avg waiting time increases.

18) **critical section** :- multiple process comes together and ~~data~~ dis section is critical section. ~~raise~~ condition comes (one process goes another car) = data inconsistency

19) **Semaphore** :- to remove ~~raise~~ condition

→ Binary - 0, 1 lock

→ counting - 0 --- n lock

20) **Deadlock** :- situation in which two or more process & unable to proceed because each of them is waiting for another one to do something.

Conditions :- 1) mutual exclusion :- no two process can access ~~same~~ one resource at same time

2) ~~hold and wait~~

2) hold and wait :- a process holds a resource and wait for second resource

3) No preemption of resource :- if goes in d process then will comp and come, time quantum doesn't work.

4) Circular wait :- process depends on each other in circle form

Deadlock Handling :-

1) D.L prevention - prevent before occurring

2) D.L avoidance

3) D.L Ignorance X

4) D.L detection & recovery X

D.L prevention :- it prevents before occurring, one of necessary D.L cond becomes false

D.L avoidance :-

Bankers algo :- when a process req set of res d sys must determine whether allocating these resource will lv d sys in safe state. If yes then allocated.

→ deadlock free

Paging :- Block of mem called frame
Block of process is called page

→ **Scheduling algo** :-

① **SJF (shortest job first)** :- Burst time, non preemptive

$$TAT = T - AT$$

$$WT = TAT - BT$$

↓
one process given to
CPU, it will execute fully

② **priority scheduling** :- priority, preemptive

③ **round robin** :- time quantum^{A.T}, preemptive + queue

* preemptive mean bas ek time quantum add.

④ **FCFS** :- A.T
