OS — Thread is smallest seq. of programmed inst.
smallest unit of CPU

① Process is program under execution

② PCB :- Process Control Block :- Store data of past process & executing one.

- It is a data structure.

Code	→ code
Data	→ var
Stack	→ funcn.

③ Batch OS :- Make batches of jobs & send it to cpu to execute it.
→ Run on FCFS basis.
→ Increase inter performance, less overhead on operator
→ No intervention.

④ Scheduling OS :-

③ Multitasking OS :- → fast switching in all programs
- User can interact with program
- Need scheduling algo.
- Time shared OS
- memory mang. required

④ Distributed OS ~ ① Multiple CPU with own memory.

⑤ Kernel — 1st program loaded at startup.
Critical part of OS handled by kernel

Two types ① Monolithic :- all ~~pro~~ critical & non critical in kernel. A small bug will crash system.
② Micro :- Critical service in kernel.

★ P1 creates P2 process
P1 → Parent process
P2 → Child process

Info of child process stored in parent process identification code

Program running is called process.

child process is called fork

PAGE NO :
DATE : /

☆ Process state :→
① Start ② Ready ③ Running
④ Waiting ⑤ Terminated

☆ To switch the process from P_1 to P_2, in the CPU is called context switching. At that time P_1 goes from running to block state

☆ Process Synchronization

① Part of code/process where it manipulates the value of global var. / shared resource is called critical section of code.

② At instance only one process should go in critical section.

③ At a time, when one process is allowed to go in critical section is called mutual exclusion

④ There should be sync. in process, to know that already one process is in critical secn.

⑤ When multiple process execute, concurrently, and they access value of global var, and print it, that depend on their execution sequence. It is called Race condition

Characteristic of Process Synch

① Mutual Exclusion
② No starvation
③ Progress :- If no process, in critical secn, but more than 1 process wants to go in it, then from ~~both~~ only 1 process will go in critical secn.

Peterson algo is soln for it

Deadlock : A ...
Request →
→ Edge ...
Cond...

Scanned with CamScanner

Deadlock :- A state of system where processes are waiting for each other and their wait is indefinite

→ Request → Allocate → Release

→ Edge from resource to process → allocate
→ " " process to resource → Request.

Condn for Deadlock :-
→ Mutual exclusion
→ Process hold resources and keep waiting for those other process resources
→ No preemption :- OS cannot take resource from resource forcefully
→ Circular waiting

$P_1 \to P_2 \to P_3 \to P_4$

Prevent Deadlock :-
① Deadlock Prevention - by making anyone condn fails
② Deadlock avoidance :- allocate resource in some order to prevent it
③ Detect and Recover the.
④ ~~Banker's Algo~~

→ Dead lock can be avoided by ~~algo~~ Banker's algo :→
• Safety algo • Resource req. algo.

☆ Memory management :-

① Contiguous Allocation.
- External Fragment :→ First Fit, Worst Fit, Best Fit

When we have memory equal or greater than required but spilled all over → External fragmentation

- Non-contiguous memory allocation

→ Paging :- • Secondary mem divided in equal size called pages.
S.M to Fragment
• Page table made when page is transferred from
• Main memory also divided in equal size of page called fragments.
• From secondary m., all process are bought in fragment.

Page table

Page No.	Fragment No

Page no is offset.
i.e. at this page, I want that no. of instruction in that page.

- Solves external fragmentation
- When we have left over memory in a page, as the process is small, is called internal fragmentation

→ Segmentation :- Logical address space divides in segments.
- Segments are not of fixed size
- Main memory ~~not~~ divided in segments

★ Page Replacement Algo :-

① FIFO

Page is not available in m.m → Page fault.

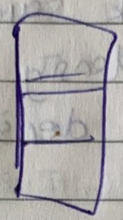
Page fault Rate = $\frac{\text{No of faults}}{\text{total no. of ref.}}$ → i.e. no of pages.

Page Hit Rate = $\frac{\text{No of Hit}}{\text{Total no. of ref.}}$ → i.e. no. of page in m.m

② Belady's Algo :- In FIFO algo, if main m. size is increased, fault rate increases

③ Optimal Page Replac.. Algo :
- Page fault is min
- Replace page that won't use in near fn
- Pratically not possible

MM = 3 ⇒



④ Least Recently Used (LRU) :-
delete the page that is least recently use

☆ Semaphores :- For Process Synch.
- Global shared var. that is used by every process.
- Implement as int and struct type.
- Two functions to manipulate value of semaphore :-

wait (sem_ s) : P(s)
signal (sem- s) : V(s)

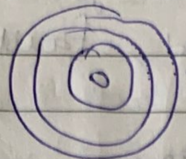
if ($s \leq 0$) → donothin else S-- → $s = 0$

$s = 1$ or $s = 0$

signal (s) → s++ → $s = 1$

If semaphore not implemented correctly, deadlock & starvation occur

☆ Secondary Memory Management:-

Disk:-  → platter

Each concentric circle is called track.

- Constant angular velocity:→
  Every track same capacity.

∴ Inner track has more density.
∴ Outer track has lower density.

- Constant linear velocity:-
  Every track has same density.

∴ Inner track → lower capacity
∴ outer —"— → high capacity.
Velocity increase inner side.

- Disk Access Time:-

→ Seek time = $\frac{\text{Time from inner to outer or vice versa}}{2}$

→ Rotational latency = $\frac{\text{one complete revolution}}{2}$

→ Block Transf. time = $\frac{\text{Block size}}{\text{transf. rate}}$

transf. rate = $\frac{\text{Track capacity}}{\text{1 revolution time}}$

∴ D.A.T = Seek time + Rotal. L + B.T.T.

★ Disk Scheduling algo:-
FCFS, SSTF (shortest seek time first)
SCAN, C-SCAN, LOOK, C-LOOK.

① FCFS | 20 | 0 | 1 | 5 |

No. of track = 100 Current head at 4.
∴ Total head movement = (20-4) + (20-0) +
(1-0) + (5-1).
=

② Shortest Seek Time:-
= (5-4) + (5-1) + (1-0) + (20-0)
=

③ SCAN :- Head move from inner to outer and then
comeback or viceversa

∴ (5-4) + (20-5) + $\underline{\underline{(99-20)}}$ + (99-1)
+ (1-0).
=

④ C-SCAN ⇒ Come back to zero directly.
- Fulfill req. in one direction only

(5-4) + (20-5) + (99-20) + (99-0)
+ (0-0) + (1-0) =

⑤ Look :- Head move from left to right & back till last reqs
= (5-4) + (20-5) + (20-1) + (1-0)
=

⑥ C-look ⇒ Same but serves in only one dir.
(5-4) + (20-5) + (20-0) + (0-0)
+ (1-0)

PAGE NO :

DATE : / /

- RAID stands for Redundant array of independent disk store all same data to improve performance

- External fragmentation occur when variable size allocation

→ Binary semaphore takes only 0 and 1 as value. Implement mutual exclusion and sych. process

- Aging used to avoid starvation.

- Multithreading in a system in which multiple thread are created of a process to increase computing speed

- Round Robin used for time sharing OS.