2-2-26
Monday

Operating System Interview Questions

Q1, 2, 3, 4, 10, 11, 12, 17, 18, 20, 25, 26, 43, 48, 75

(Q1) What is a process and process table?

(Ans 1) **Process**

→ Program currently running (in execution)
→ Eg:- web browser
→ Operating System (OS) manage all running process.
→ OS gives CPU time to each process to run.
→ OS also provides memory & disk space.

Process Table

→ Keeps track of all running process.
→ Table contains entry for every process.
→ Store info. about resources used by process and current state of the process.

(Q2) Different states of process?

→ 3 States of Process

1) Running State

→ Process has all required requirements to execute.

→ OS has allowed it to use CPU.

→ One process run at a time.

2) Waiting State

→ Process is waiting for external event to happen.

→ Eg:- waiting for user input.

3) Ready State

→ Process has all resources except the CPU.

→ It's waiting for permission from OS to use the processor.

Note

In real(i) ready and waiting state are managed using Queues that store the process in these states.

Q3) What is a

→ Single sequen
a process.

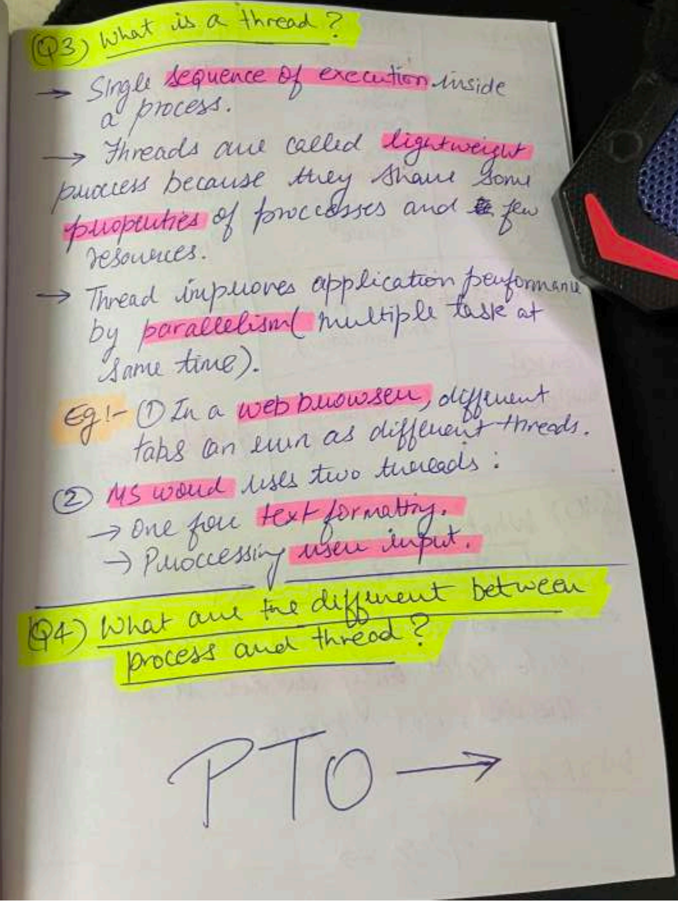
→ Threads a
process becau
properties &
resources.

→ Thread im
by paral
same ti

Eg:- ① In
tabs

② MS wo
→ One
→ Pro

Q4) What
proce



Q3) What is a thread?

→ Single sequence of execution inside a process.

→ Threads are called lightweight process because they share some properties of processes and few resources.

→ Thread improves application performance by parallelism(multiple task at same time).

Eg:- ① In a web browser, different tabs can run as different threads.

② MS word uses two threads:

→ One for text formatting.

→ Processing user input.

Q4) What are the different between process and thread?

PTO →

Aspect	Process	Thread
Execution unit	Independent program under execution.	Smallest unit of CPU execution inside process.
Memory ~~unit~~ usage	has seperate memory space.	Shares memory of the process.
Communication	use IPC (inter-process communication).	Uses shared memory.
Context Switching	Slow/heavy	fast/lightweight

Q10) What is Demand Paging and how it works?

→ Means a page is loaded into RAM only when it is needed, not before.

Working :

PTO →

1) Process St

→ Full

→ Only

2) Page to

→ OS ch

(3) Page

→ if p

occu

(4) Page

→ OS

mem

(5) Pa

→

(6)

(1) Process Start

→ Full page is not loaded into RAM.
→ Only required pages are loaded.

(2) Page table check

→ OS check page table if page in RAM

(3) Page fault Occur

→ if page not in RAM, Page fault Occur.

(4) Page is loaded

→ OS bring page from Secondary memory to RAM.

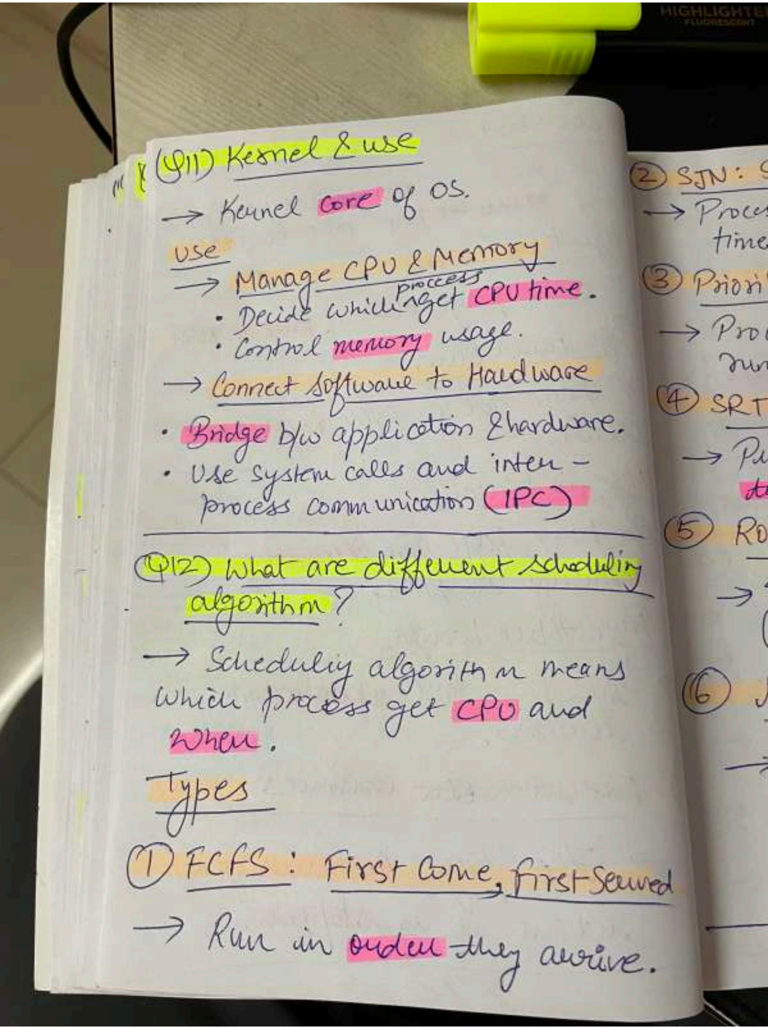
(5) Page table Loaded

→ The page table updated with new location.

(6) Execution ~~Gett~~ Continues

→ The page continues from where it is stopped.

PTO →



(Q11) Kernel & use

→ Kernel core of OS.

Use

→ Manage CPU & Memory

- Decide which process get CPU time.
- Control memory usage.

→ Connect software to Hardware

- Bridge b/w application & hardware.
- Use system calls and inter-process communication (IPC)

(Q12) What are different scheduling algorithm?

→ Scheduling algorithm means which process get CPU and when.

Types

① FCFS : First Come, first Served

→ Run in order they arrive.

② SJN : S

→ Proce
time

③ Priori

→ Pro
run

④ SRT

→ Pu
t

⑤ Ro

→

⑥ J

(2) SJN: Shortest Job Next

→ Process with **shortest** execution time runs first.

(3) Priority Scheduling

→ Process with highest **priority** runs first.

(4) SRT: Shortest Remaining time

→ Process with **least** remaining **time** is executed next.

(5) Round Robin (RR)

→ Each process gets **fixed time** (time slice) in circular order.

(6) Multi-level Quue Scheduling

→ Process are divided into **different Queues** based on type.

PTO →

(Q17) Explain FCFS

→ First come first serve

→ Process which comes first gets CPU first.

→ working

- Process are executed in same order they arrive in ready Queue.
- Once process gets CPU it keep's it until finishes or goes for I/O.

→ Disadvantage

- If long process comes all short process must wait.

(Q18) Round Robin (RR) Scheduling

→ Each process gets a fixed time in Circular order.

Working

① Fixed

→ Eac fix

② Procc

→ If mov

③ CP rep

(Q20)

Working

① Fixed time given

→ Each process is given small fixed time.

② Process not finished

→ If process not finished / interupted move to end of queue.

③ CPU moves to next process, repeating the cycle.

(Q20) Banker's Algorithm

→ Deadlock Avoidance Algorithm.

→ Checks giving resources to a process is safe or not before allocation.

→ Working

→ Each process tells maximum resource need.

→ OS simulate resource allocation before giving them.

→ If safe allocated else denied.

(Q25) Basic Function of paging

→ **Paging:** Store a program in memory in non contincous block.

Working

→ Divide memory
- Hard disk → pages
- RAM → frame

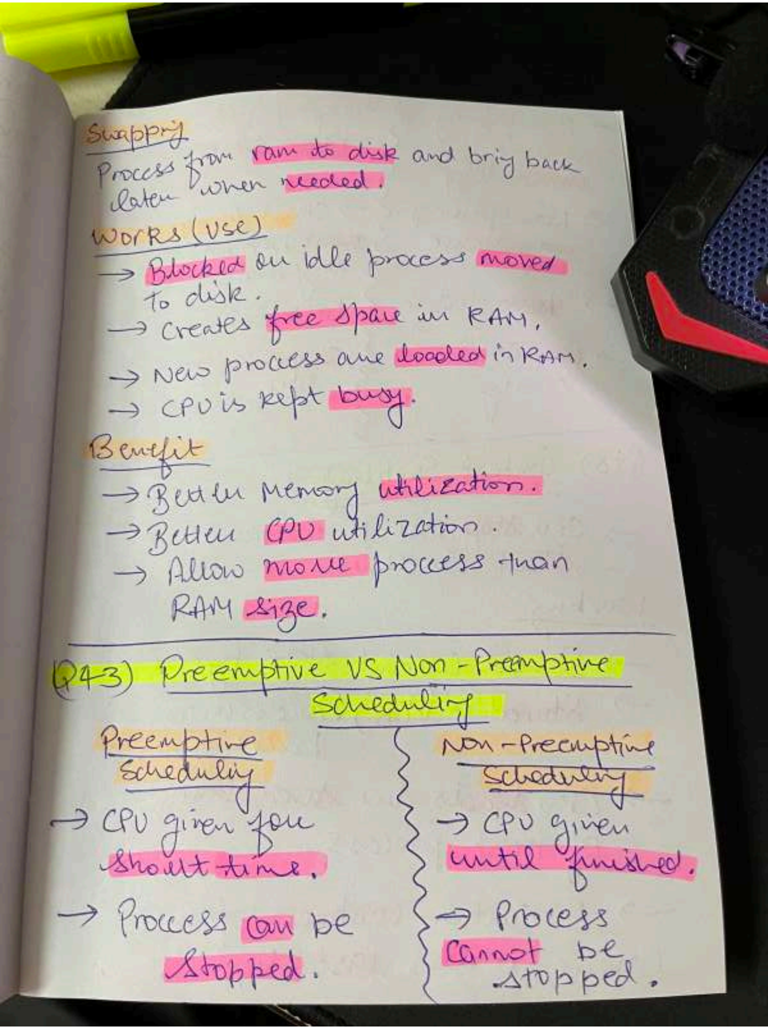
→ Loads Program parts
- Process split into pages
- Pages are placed into free frame in ram.

Benefit

→ Better use of memory.
→ NO Contineous memory space need.
→ Reduce Memory waste.

(Q26) How swapping improves memory management?

Swapping
Process fr
later
works (
→ Blo
To c
→ Cr
→ Ne
→ C
Benef
→ B
→ R
→
(Q43)
Pr
→ C
→



Swapping

Process from ram to disk and bring back later when needed.

Works (use)

- → Blocked or idle process moved to disk.
- → Creates free space in RAM.
- → New process are loaded in RAM.
- → CPU is kept busy.

Benefit

- → Better memory utilization.
- → Better CPU utilization.
- → Allow more process than RAM size.

Q43) Preemptive VS Non-Preemptive Scheduling

Preemptive Scheduling	Non-Preemptive Scheduling
→ CPU given for short time.	→ CPU given until finished.
→ Process can be stopped.	→ Process cannot be stopped.

→ High priority can run immidiately.	→ High priority must wait.
→ Low priority may starve.	→ Short priority wait behind long ones.
→ More flexible.	Less flexible.
→ Complex & costly	→ Simple & low cost.

Q48) Context Switching

→ CPU stop one process and start other.

Working

→ Saved current state by CPU.

→ Stored in (PCB) Process Control block.

→ CPU loads the saved state of next process.

→ Execution continous from where it was last stopped.

→ Why nee

→ Multip

→ impr

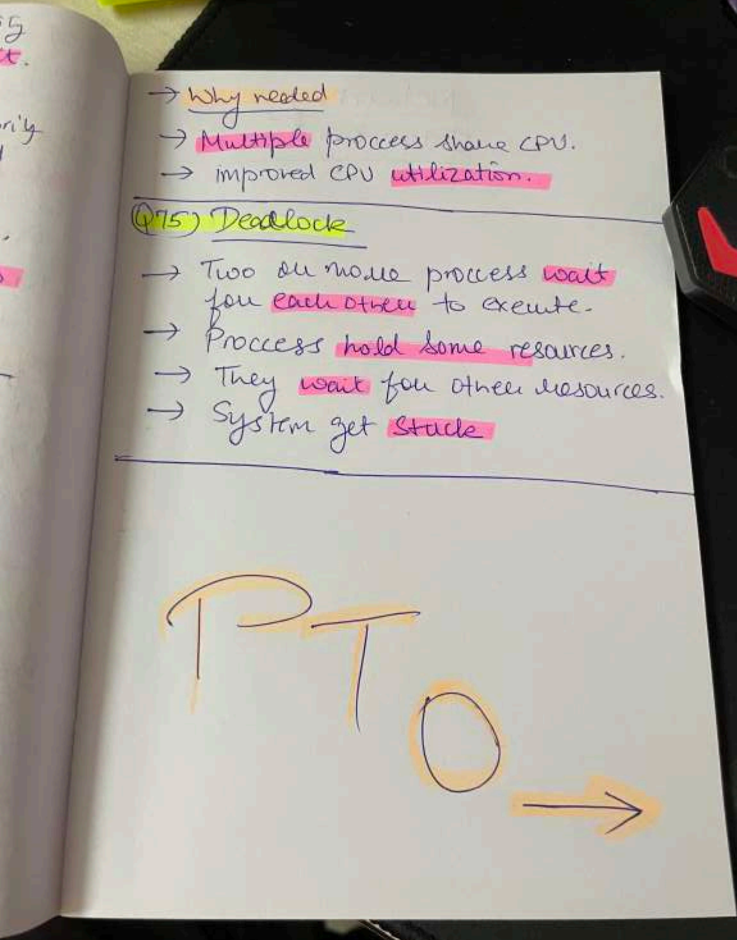
Q75) Dea

→ Two fou

→ Pro

→ Th

→ Sy



→ Why needed

→ Multiple process share CPU.

→ improved CPU utilization.

Q75) Deadlock

→ Two or more process wait for each other to execute.

→ Process hold some resources.

→ They wait for other resources.

→ System get Stuck

PTO →