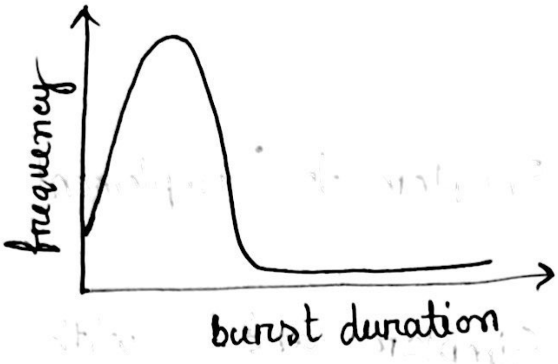
Chap 5

CPU Scheduling

⊞ Histogram of CPU Burst Time

Burst Time → Amount of CPU Time the process requires to complete execution.



⊞ **CPU Scheduler:** A CPU schedular selects from among the processes in ready queue and allocate a CPU core to one of them.

□ • CPU Scheduling decisions may take when a process,

(I) switches from running to waiting state

(II) " " " " ready "

(III) " " waiting " " "

(IV) Terminates.

- for (I),(IV) ⇒ CPU scheduler must pick another process from ready queue. No choice here, a new process must be scheduled.
- for (II),(III) ⇒ CPU scheduler ~~mus~~ may choose to schedule another process, but it's upto the system. There is choice. The current process must continue or a new one could be scheduled.

Scanned with CamScanner

Difference between Preemptive and Non-preemtive Scheduling :

	Preemptive Scheduling	Non-Preemptive scheduling
Interrupts	OS can interrupt or preempt a running process.	The process runts to completetion
Complexity	More complex to implement and manage	Simpler to implement & manage
Efficiency	Can be more efficient in handling higher priority tasks.	Simpler and' with lower overhead.
Context Switching	Frequent, as processes can be preempted at any time.	Less frequent, as processes run until the finish or block.
	Windows, MacOS, Linux. and UNIX use this algorithm	

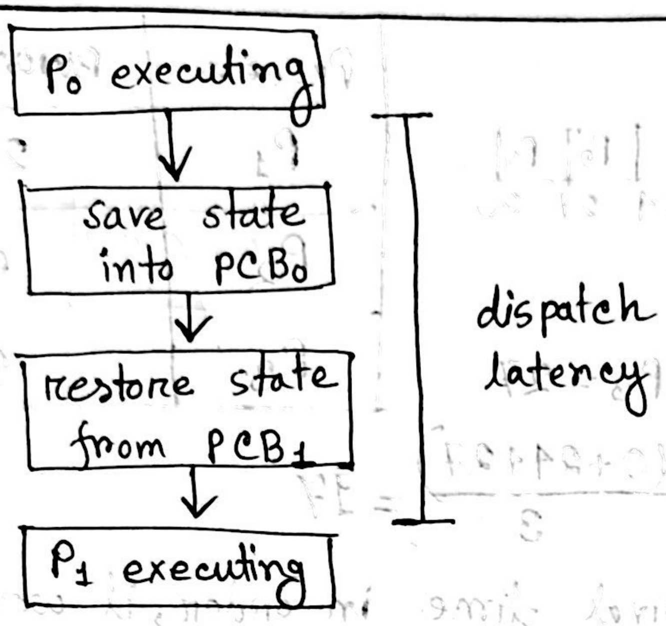
Dispatcher: The dispatcher is responsible for giving ontrol of the CPU to the process chosen by CPU scheduler

(I) Switching context

(II) Switching to user mode

(III) Jumping to the proper location in the user program to restant that program.

- **Dispatch Latency** — is the time it takes for the dispatcher to stop one process and start another running.



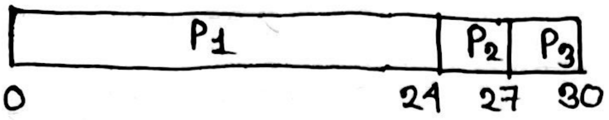
Scheduling Criteria :

- CPU utilization — keeps the CPU as busy as possible
- Throughput — No. of processes that complete their execution per time unit.
- Turnaround Time — amount of time to execute a particular process.
- Waiting Time — amount of time process has been waiting in the ready queue.
- Response Time — amount of time it takes from when a request was submitted until the first response is produced.

Scanned with CamScanner

FCFS Scheduling

Gantt chart:



Process	Burst Time
P_1	24
P_2	3
P_3	3

Waiting Time for
$P_1 = 0$, $P_2 = 24$, $P_3 = 27$

Avg waiting Time $\frac{(0+24+27)}{3} = 17$

if we sort arrival time in order, it would be much better than the previous case.

SJF – Shortest Job First Scheduling:

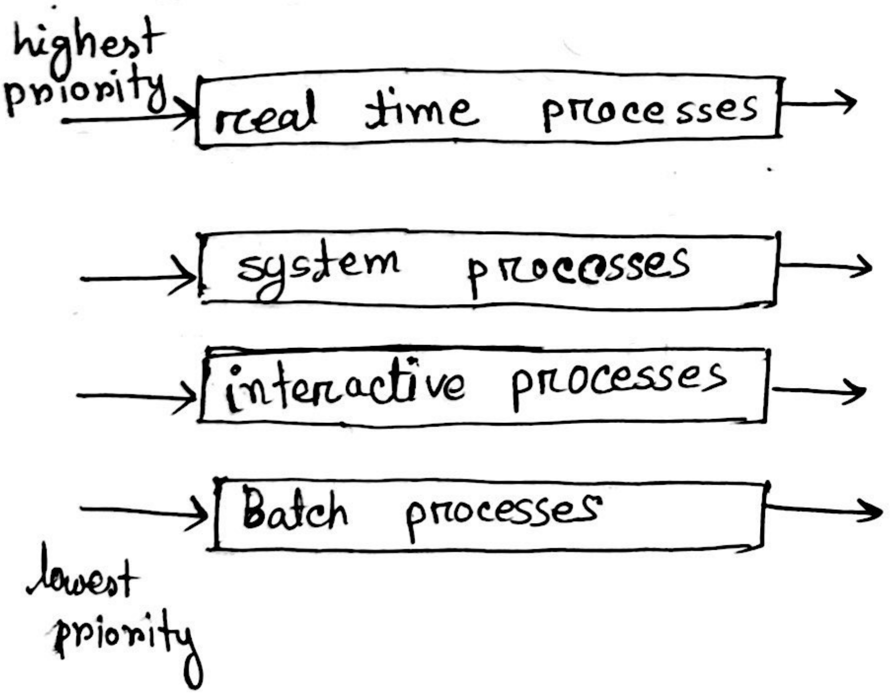
Scanned with CamScanner

Multilevel Queue:

The ready queue consists of multiple queues.

Multilevel Queue scheduling is a CPU scheduling method where processes are divided into different queues based on priority or type. Each queue has its own scheduling program(like FCFS. or Round Robin High - priority queues are executed first, while lower-priority queues wait. This approach ensures efficient handling of diverse process types but can cause starvation for lower-priority processes

priority based upon process type -



Scanned with CamScanner

Multilevel Feedback Queue : ~~This~~ It is a scheduling algorithm that allow processes to move between different queues based on their behaviour and requirements.

- Number of queues ——
- Scheduling algorithm for each queue
- to determine when to upgrade/demote a process
- Used to determine which queue a process will enter ~~the~~ when that process needs service
- Aging is applied to prevent starvation.

Advantages: Prevents starvation for lower-priority ~~queues on~~ processes that have been waiting for a long time. Dynamically adjust priority of processes and Balancees responsiveness for short processes and fairness for long processes.

- Example – Slide 32 Page

Scanned with CamScanner

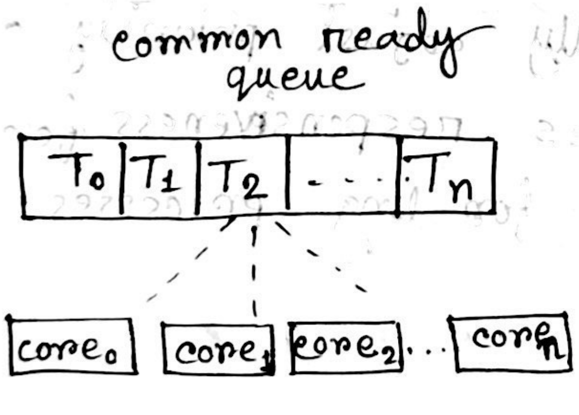
□ Multiple process scheduling

CPU schedule becomes more complex when multiple CPU's are available.

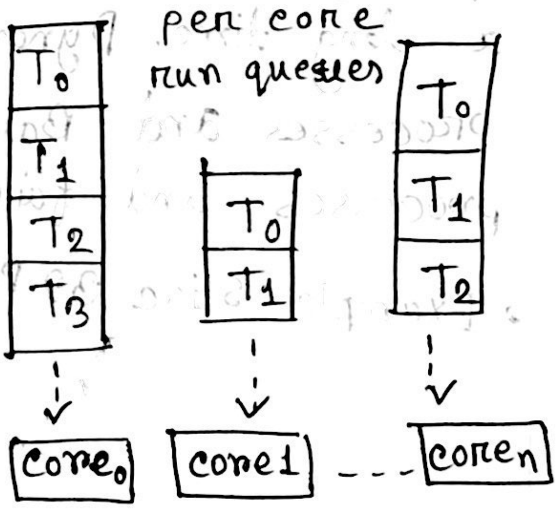
So, Multiprocess scheduling can be ~~any~~ of the following architecture —

- Multicore CPU's
- Multithreaded cores
- NUMA systems
- Heterogeneous multiprocessing.

□ Symmetric multiprocessing (SMP) is where each processor is self processing.



All threads may be in a common ready queue



Each processor may have its own private queue of threads

Multicore processores :-

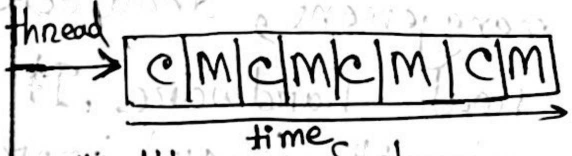
A computer processors that have two or more independent processing units (cores) on a single physical chip.

- faster and consumes less power (Energy & Time efficie
- Handle multiple tasks or threads at the same time.
- Used in gaming, multitasking, data processing and heavy computation.

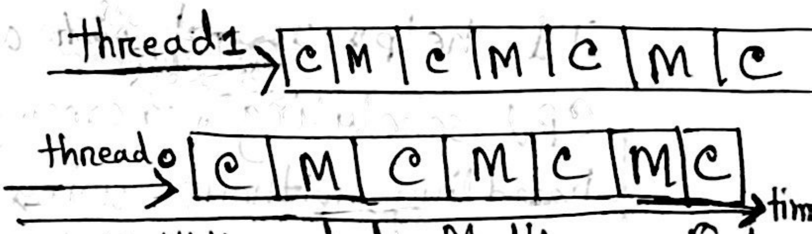
Multiple threads per core also ~~gaining~~ take advantage of memory stall to progress on another thread while memory retrieve happens.

MultiThreaded Multicore System :-

Each core has greater/more than 1 hardware threads. If one thread has a memory stall, switch to another thread.



Multicore System



Multithreaded Multicore System

Little's Formula

n = average queue length

W = Average waiting time in queue

λ = " arrival rate into queue

Little's law — In steady state, processes leaving queue must equal processes arriving.

$$n = \lambda \times W$$

Example: $\lambda = 7$

$n = 14$

$$W = \frac{n}{\lambda} = \frac{14}{7} = 2 \text{ seconds}$$

Simulations

Simulation in OS is using software to mimic how an operating system works. It helps to test and compare things like CPU scheduling, memory management, deadlock handling without using real hardware. It is useful for learning, testing algorithms and improving

Scanned with CamScanner

system performance safely.

- Queueing models are limited in accuracy But simulations are more accurate using programmed computer models.

- It gather statistics to evalute algorithm performance by using probabilities, distribution and trace tapes.

Implementation / Application :

It has high cost, high risk. Though Flexible schedulers can be modified per-system or API but they work differently in different environment.

Scanned with CamScanner