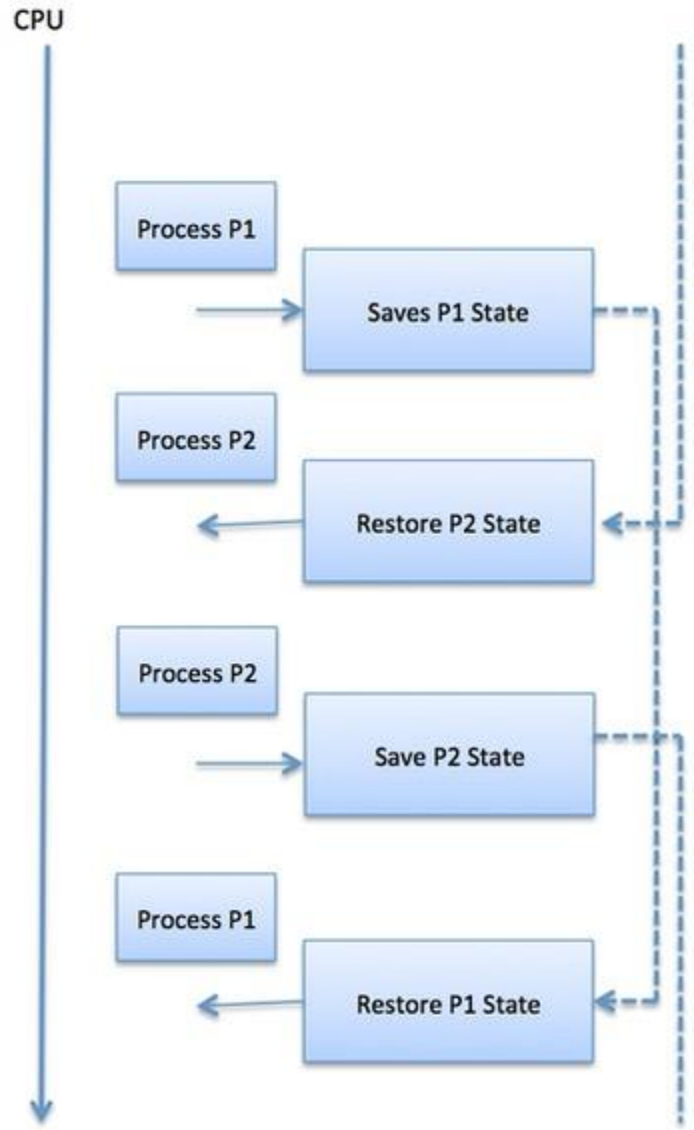
PROCESS SCHEDULING

When one process has to wait, the OS takes the CPU and gives it to another process

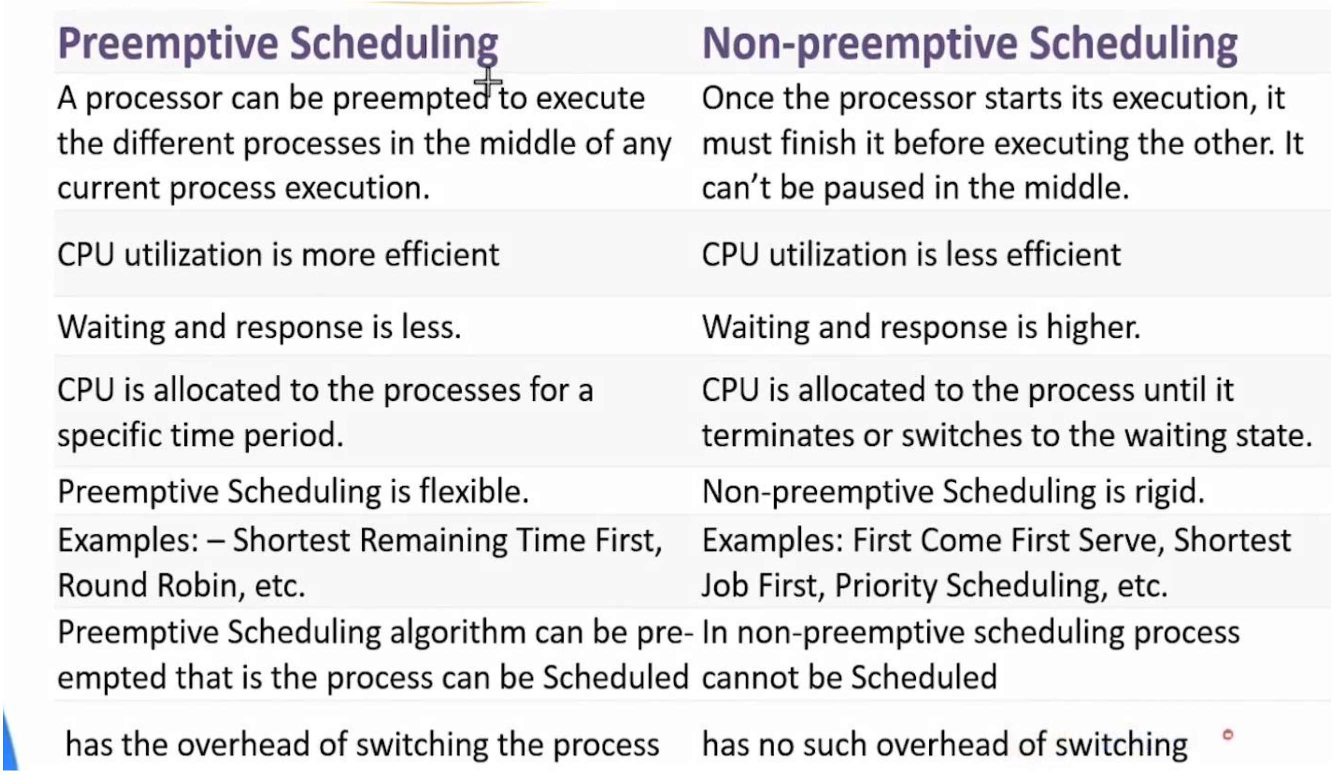


CPU scheduling decision may take place when a process

1) Switches from **running to waiting** state
2) Switches from **running to ready** state
3) Switches from **waiting to ready**
4) **Terminates**

Scheduling under **(1) and (4) is nonpreemptive**

All other scheduling is preemptive



Preemptive Scheduling	Non-preemptive Scheduling
A processor can be preempted to execute the different processes in the middle of any current process execution.	Once the processor starts its execution, it must finish it before executing the other. It can't be paused in the middle.
CPU utilization is more efficient	CPU utilization is less efficient
Waiting and response is less.	Waiting and response is higher.
CPU is allocated to the processes for a specific time period.	CPU is allocated to the process until it terminates or switches to the waiting state.
Preemptive Scheduling is flexible.	Non-preemptive Scheduling is rigid.
Examples: – Shortest Remaining Time First, Round Robin, etc.	Examples: First Come First Serve, Shortest Job First, Priority Scheduling, etc.
Preemptive Scheduling algorithm can be pre-empted that is the process can be Scheduled	In non-preemptive scheduling process cannot be Scheduled
has the overhead of switching the process	has no such overhead of switching

Dispatcher

The dispatcher is done after the scheduler. **It gives control of the CPU to the process selected by the short-term scheduler**. After selecting the process, the dispatcher gives CPU to it.

The dispatcher determines from which location of the process (if the process previously ran and its states were saved then it need to be start from where it left off)

It involves

a) Switching context
b) Switching to user mode
c) Jumping to proper location in the user program resume that program

Dispatch latency

Time it takes for the dispatcher to stop one process and start another process

Scheduling Criteria

CPU utilization

Keep the CPU as busy as possible

Throughput

Number of processes that complete their execution per time unit

Turnaround Time

Amount of time to execute a particular process from submission to completion

Waiting Time

Amount of time a process has been waiting in ready queue

Response Time

Amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing system)

Scheduling Algorithm optimization Criteria

a) Max CPU utilization
b) Max throughput
c) Min turnaround time
d) Min waiting time
e) Min response time