Scheduling in OS

Bhupendra Pratap Singh

ACTS, CDAC, Pune

What is CPU scheduling?

- CPU scheduling is a process which allows one process to use the CPU while the execution of another process is on hold(in waiting state) due to unavailability of any resource like I/O etc, thereby making full use of CPU.
- The aim of CPU scheduling is to make the system efficient, fast and fair.
- Whenever the CPU becomes idle, the operating system must select one of the processes in the **ready queue** to be executed. The selection process is carried out by the CPU scheduler).
- The scheduler selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them.

What is Dispatcher?

- The dispatcher is the module that gives control of the CPU to the process selected by the **scheduler**. This function involves:
- Switching context
- Switching to user mode
- Jumping to the proper location in the user program to restart that program from where it left last time.
- The time taken by the dispatcher to stop one process and start another process is known as the **Dispatch Latency**.

When CPU scheduling can/may take place?

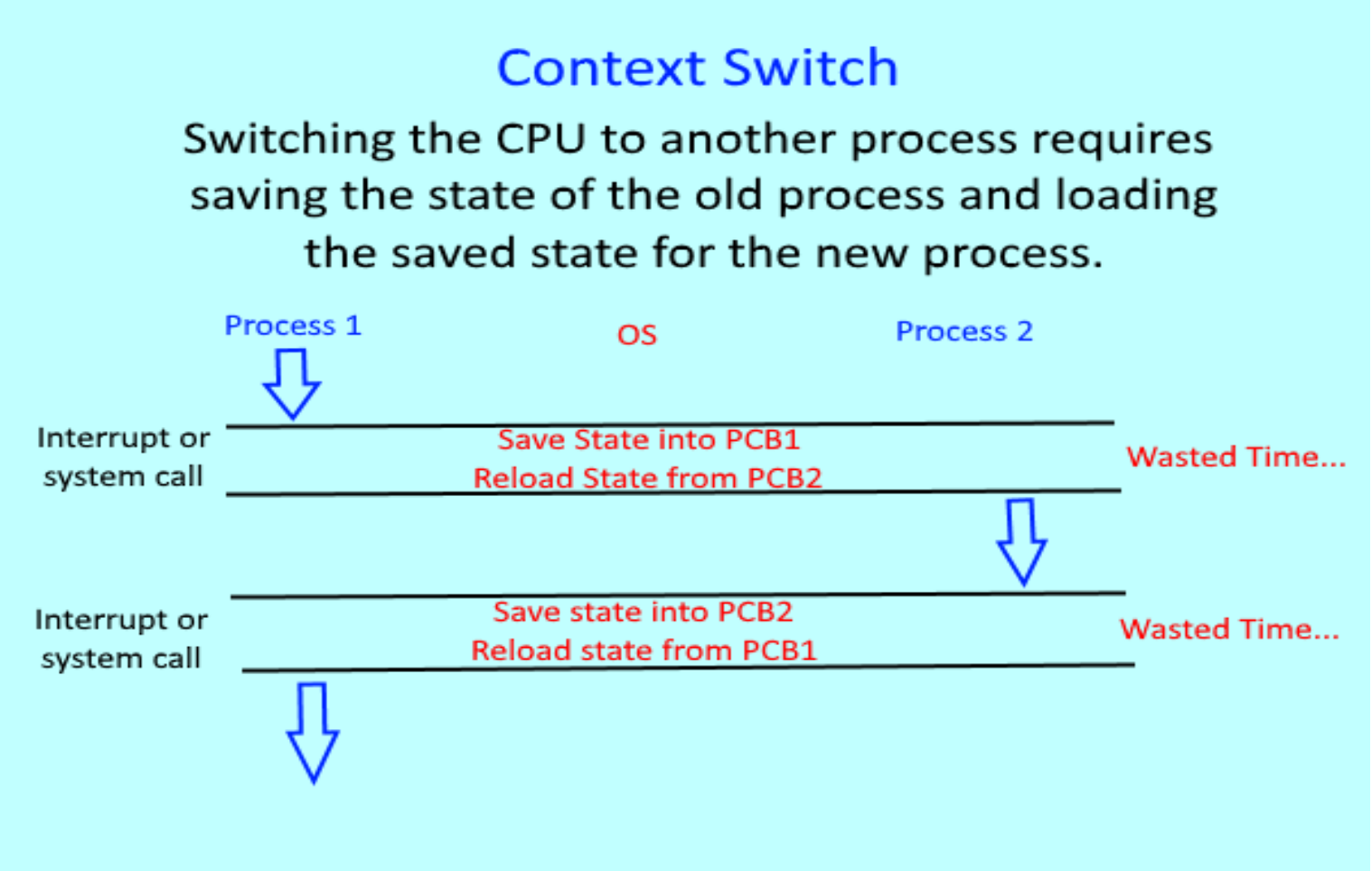
- When a process switches from the **running** state to the **waiting** state(for I/O request or invocation of wait for the termination of one of the child processes).
- When a process switches from the **running** state to the **ready** state (for example, when an interrupt occurs).
- When a process switches from the **waiting** state to the **ready** state(for example, completion of I/O).
- When a process **terminates**.
- 1st and last, there is no choice in terms of scheduling. A new process(if one exists in the ready queue) must be selected for execution.

Non-Preemptive & Preemptive scheduling

- Under non-preemptive scheduling, once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or by switching to the waiting state.
- Preemptive scheduling: the tasks/process/threads are usually assigned with priorities. At times it is necessary to run a certain tasks/process/threads that has a higher priority before another tasks/process/threads although it is running. Therefore, the running task is interrupted for some time and resumed later when the priority task has finished its execution.

Context Switching

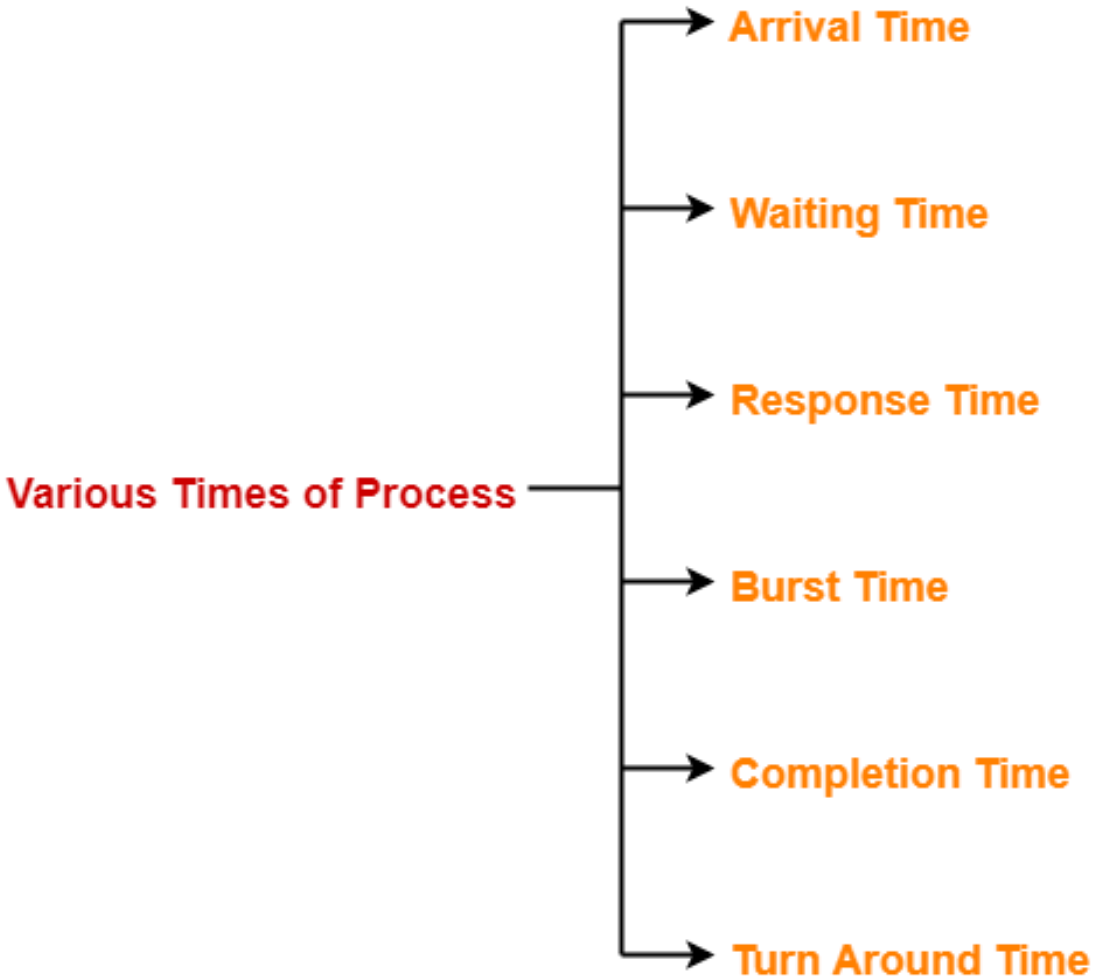
Switching of CPU from one process to another is known as Context Switching



Starvation and Aging

- Starvation or indefinite blocking is phenomenon associated with the Priority scheduling algorithms, in which a process ready to run for CPU can wait indefinitely because of low priority. In heavily loaded computer system, a steady stream of higher-priority processes can prevent a low-priority process from ever getting the CPU.
- Aging is a condition which is used to reduce starvation of low priority tasks. It is a process which gradually increases the priority of task depending on waiting time. It ensure that jobs in the lower level queues will eventually complete their execution.

Various Process Timings



Various Process Timings

Arrival Time- Arrival time is the point of time at which a process enters the ready queue.

Waiting Time- Waiting time is the amount of time spent by a process waiting in the ready queue for getting the CPU.

Waiting time = Turn Around time – Burst time

Response Time- Response Time: Response time is the amount of time after which a process gets the CPU for the first time after entering the ready queue.

Response Time = Time at which process first gets the CPU – Arrival time

Various Process Timings

Burst Time: Burst time is the amount of time required by a process for executing on CPU.

- It is also called as execution time or running time.
- Burst time of a process can not be known in advance before executing the process. It can be known only after the process has executed

Completion Time: Completion time is the point of time at which a process completes its execution on the CPU and takes exit from the system. It is also called as exit time.

Various Process Timings

Turn Around Time:

- Turn Around time is the total amount of time spent by a process in the system.
- When present in the system, a process is either waiting in the ready queue for getting the CPU or it is executing on the CPU.
- Turn Around time = Burst time + Waiting time

or

Turn Around time = Completion time – Arrival time

FIRST COME FIRST SERVE SCHEDULING(FCFS)

- The process which arrives first in the ready queue is firstly assigned the CPU.
- In case of a tie, process with smaller process id is executed first.
- It is always non-preemptive in nature.

Advantages-

- It is simple and easy to understand.
- It can be easily implemented using queue data structure.
- It does not lead to starvation.

Disadvantages-

- It does not consider the priority or burst time of the processes.
- It suffers from **convoy effect**.

What is convoy effect?

- Consider processes with higher burst time arrived before the processes with smaller burst time.
- Then, smaller processes have to wait for a long time for longer processes to release the CPU.

Shortest Job First Scheduling

- Out of all the available processes, CPU is assigned to the process having smallest burst time.
- In case of a tie, it is broken by **FCFS Scheduling**.
- Non Preemptive in nature.

Shortest Job First Scheduling

Advantages-

- SRTF is optimal and guarantees the minimum average waiting time.
- It provides a standard for other algorithms since no other algorithm performs better than it.

Disadvantages-

- It can not be implemented practically since burst time of the processes can not be known in advance.
- It leads to starvation for processes with larger burst time.
- Priorities can not be set for the processes.
- Processes with larger burst time have poor response time.

Longest Job First

- Out of all the available processes, CPU is assigned to the process having largest burst time.
- In case of a tie, it is broken by FCFS Scheduling.
- Non Preemptive in nature.

Longest JOB First

Advantages-

- No process can complete until the longest job also reaches its completion.

Disadvantages-

- The waiting time is high.
- Processes with smaller burst time may starve for CPU.

Round Robin Scheduling

- CPU is assigned to the process on the basis of FCFS for a fixed amount of time.
- This fixed amount of time is called as time quantum or time slice.
- time quantum expires, the running process is preempted and sent to the ready queue & processor is assigned to the next arrived process.
- **It is always preemptive in nature.**

Round Robin Scheduling

Advantages-

- It gives the best performance in terms of average response time.
- It is best suited for time sharing system, client server architecture and interactive system.

Disadvantages-

- It leads to starvation for processes with larger burst time as they have to repeat the cycle many times.
- Its performance heavily depends on time quantum.
- Priorities can not be set for the processes.

Round Robin Process Selection Rules

With round-robin, the scheduler always selects the process that's had the most time since it was last run, and *new processes coming in are always assigned a last run time of 1 time unit before the oldest last run time in the current list*.

Thank you ☺

Any Questions??