Ch 5
CPU Scheduling

By

Lecturer Ameen A.Noor

Scheduling Concepts

- Scheduling is a fundamental operating system function. Almost all computer resources are scheduled before use. The CPU scheduling is central to operating systems.

CPU-I/O Burst Cycle

- The success of CPU scheduling depends on the following observed property of processes: process execution consists of a cycle of CPU execution and I/O wait. Processes alternate between these two states. Process execution begins with a CPU burst. That is followed by I/O burst, then another CPU burst and so on. The last CPU burst will end with a system request to terminate execution.

CPU Scheduler

- 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 short term scheduler (CPU scheduler). The scheduler selects from among the processes in memory that are ready to execute and allocates the CPU to one of them.

Scheduling Schemes

- There are two scheduling schemes can be recognized:
  - Preemptive scheduling
  - Non-preemptive scheduling
- Under the **non-preemptive scheduling**, once the CPU has been allocated to a process, the process keeps the CPU until it release the CPU either by terminating or by switching to the waiting state. On the other hand **Preemptive scheduling** occur when the CPU has been allocated to a process and this process is interrupted by higher priority process. At this moment the executing process is stopped and returned back to the ready queue, the CPU is allocated to the higher priority process.

Dispatcher

- It is the module that gives control of the CPU to the process selected by the CPU scheduler. This function involves:
  - Switching Context
  - Switching to user mode
  - Jumping to the proper location in the user program to restart the program.

Scheduling Criteria

- Many criteria have been suggested for comparing CPU scheduling algorithms. The criteria include the following:
  - CPU Utilization
  - Throughput
  - Turnaround Time
  - Waiting time
  - Response time

Scheduling Algorithm

- Here we will mention some of the CPU scheduling algorithms that are used in different operating systems:
  1. First Come First Served (FCFS)
  2. Shortest Job First Scheduling (SJF)
  3. Priority Scheduling Algorithm
  4. Round Robin Scheduling Algorithm

First Come First Served (FCFS)

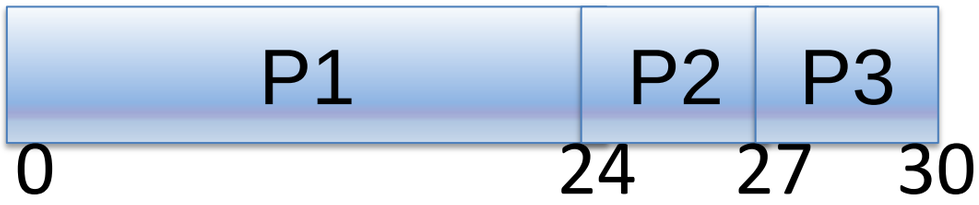
- Simplest CPU scheduling algorithm.
- Non preemptive.
- The process that requests the CPU first is allocated the CPU first.
- The implementation of the FCFS policy is easily managed with FIFO queue.
- When the CPU is free, it is allocated to the process at the head of the queue.
- The average waiting time under the FCFS policy is often quite long.

Example: three processes arrive in order P1, P2, P3.

Process	Burst time
P1	24
P2	3
P3	3

- Waiting Time:
  - P1: 0
  - P2: 24
  - P3: 27
- Completion Time:
  - P1: 24
  - P2: 27
  - P3: 30

The Gantt Chart is as follows:



- Average Waiting Time: (0+24+27)/3 = 17 M.s
- Average Completion Time: (24+27+30)/3 = 27 M.s

- What if their order had been P2, P3, P1?

Process	Burst time
P1	24
P2	3
P3	3

The Gantt Chart is as follows:

- Waiting Time
  - P1: 6
  - P2: 0
  - P3: 3
- Completion Time:
  - P1: 30
  - P2: 3
  - P3: 6
- Average Waiting Time: (0+3+6)/3 = 3
- Average Completion Time: (3+6+30)/3 = 13

Shortest Job First Scheduling (SJF)

- Associate with each process the length of its next CPU burst.
- When the CPU is available, it is assigned to the process has the smallest next CPU burst.
- If two processes have the same length, FCFS scheduling is used to break this tie.
- The SJF is either preemptive or non-preemptive.

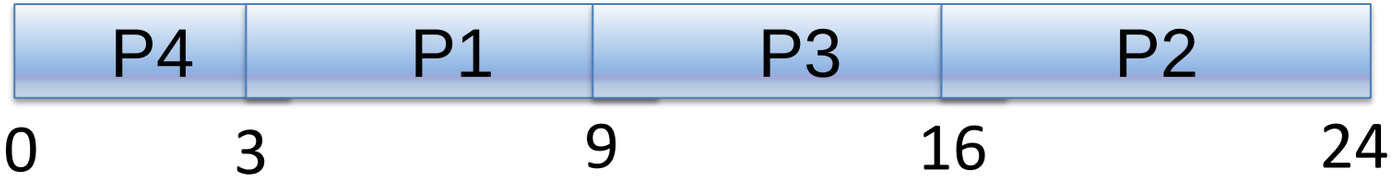
- ***Non preemptive:*** once CPU given to the process it cannot be preempted until completes its CPU burst.

- ***Preemptive:*** if a new process arrives with CPU burst length less than remaining time of current executing process, preempt. This scheme is known as the **Shortest Remaining Time First (SRTF)**.

Example: SJF (non-preemptive)

Process	Burst Time
P1	6
P2	8
P3	7
P4	3

The Gantt Chart is as follows:



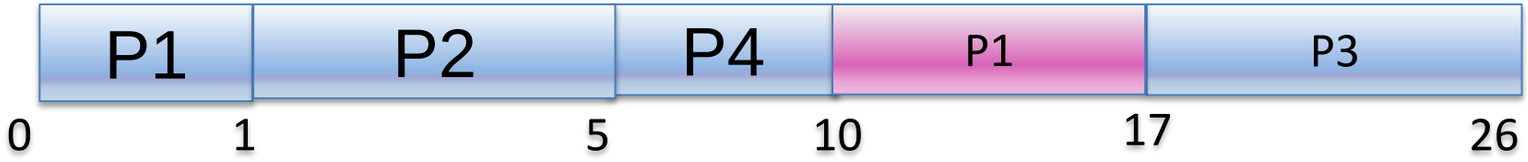
P1 waiting time: 3
P3 waiting time: 9
P2 waiting time: 16
P4 waiting time: 0

The total Execution time is: 28 M.s
The average waiting time (AWT): (0+3+9+16)/ 4 = 7 M.s
The average completion time = (9+24+16+3)/4 =13 milliseconds

SRTF Example: SJF (preemptive)

Process	Arrival Time	Burst time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

The Gantt Chart is as follows:



P1 waiting time: 10-1=9
P2 waiting time: 0
P3 waiting time: 17-9=8
P4 waiting time: 5-3=2

The total Execution time is: 26 M.s
The average waiting time (AWT): (9+0+8+2)/ 4 = 4.75 M.s

Priority Scheduling Algorithm

- A priority number (integer) is associated with each process.
- The CPU is allocated to the process with the **highest priority**

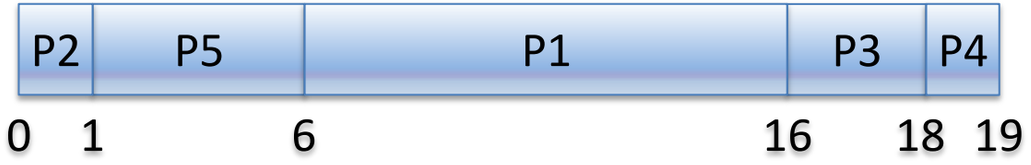
(smallest integer ≡ highest priority).

- Can be **preemptive** (compares priority of process that has arrived at the ready queue with priority of currently. running process) or **non-preemptive** (put at the head of the ready queue).
- Problem ≡ Starvation – low priority processes may never execute.
- Solution ≡ Aging – as time progresses increase the priority of the process.

Priority Example:

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The Gantt Chart is as follows:



P1 waiting time: 6
P2 waiting time: 0
P3 waiting time: 16
P4 waiting time: 18
P5 waiting time: 1

The total running time is: 19
The average waiting time (AWT): (6+0+16+18+1)/5 = 8.2 M.s

Round Robin Scheduling Algorithm

- The Round Robin is designed for time sharing systems.
- The RR is similar to FCFS, but **preemption** is added to switch between processes.
- Each process gets a small unit of CPU time (time quantum or time slice), usually 10-100 milliseconds.
- The ready queue is treated as a circular queue ,the CPU scheduler goes around the ready queue allocating the CPU to each process for a time interval of up to 1 time quantum.

Round Robin Scheduling Algorithm

- One of two things will then happen :

A. The process may have a CPU burst of less 1 TQ . In this case the process itself will release the CPU voluntarily . the scheduler will then proceed to the next process in the ready queue.

B. If the CPU burst of the currently running process is longer than 1 time quantum , the timer will go off and will cause an interrupt to the O.S. A context switch will be executed and the process will be put at the tail of the ready queue .

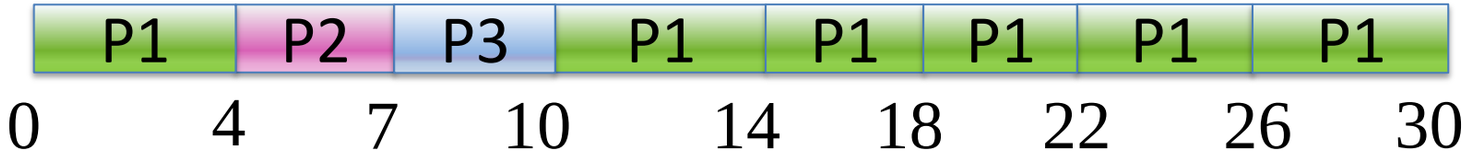
- The CPU scheduler will then select the next process in the ready queue. The A.W.T in RR is often quite long.

Example of RR: with Time Quantum = 4

Process	Burst time
P1	24
P2	3
P3	3

- Waiting Time:
  - P1: (10 - 4) = 6
  - P2: (4 - 0) = 4
  - P3: (7 - 0) = 7

The Gantt Chart is as follows:



- Completion Time:
  - P1: 30
  - P2: 7
  - P3: 10

- Average Waiting Time: (6 + 4 + 7) /3 = 5.67 M.s
- Average Completion Time: (30+7+10) /3 =15.67 M.s

Comparison among Scheduling Algorithms

Algorithms	Policy type	Disadvantages	Advantages
FCFS	Non preemptive	Average waiting time is often quite long.	Easy to implement.
SJF	Non preemptive Or preemptive	Need to know the length of the next CPU request.	Gives minimum average waiting time.
Priority	Non preemptive Or preemptive	Blocking or starvation.	1-Simplicity. 2-support for priority.
R.R	preemptive	If the set time is too long, then the system may become unresponsive, time wasting and would emulate First Come First Served.	It is easy to implement in software