DPU

Dr. D. Y. Patil Institute of Engineering & Technology

ASSIGNMENT NO. 1

(*) TITLE : Write a Java program (using OOP features) to implement following scheduling algorithms : FCFS, SJF (Preemptive), Priority (Non-Preemptive) and Round-Robin (Preemptive).

(*) OBJECTIVES : (1) To understand the functions of operating system
(2) To learn and understand process, resource and memory management

(*) SOFTWARE AND HARDWARE REQUIREMENTS :
64 bit open source Linux, OS (ubuntu), Eclipse IDE etc.

(*) THEORY :

(*) SCHEDULING ALGORITHM :

CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. There are many different CPU-scheduling algorithms.

(1) FIRST-COME, FIRST-SERVED SCHEDULING.

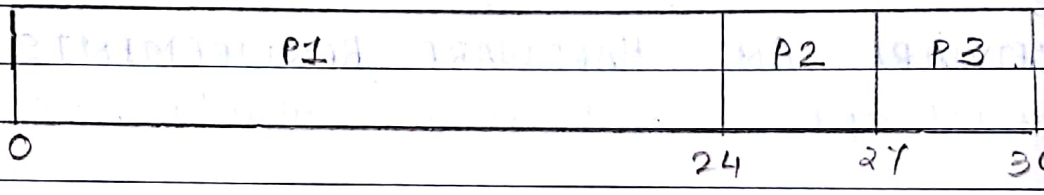
The FCFS is the simplest CPU scheduling algorithm. With this scheme, the process that requests the CPU first is allocated the CPU first. The implementation of FCFS policy is managed with a FIFO queue. When a process enters the ready queue, its PCB is linked onto tail of queue. The running process is then removed

from the queue. The FCFS scheduling algorithm is non-preemptive. once the CPU has been allocated to process, the process keeps CPU until it releases CPU, either by terminating or by requesting I/O.

Example: Consider, (set of processes that arrive at time 0).

Process	Burst Time
P1	24
P2	3
P3	3

GANTT CHART



Waiting time is 0 milliseconds for process P1, 24 milliseconds for process P2 and 27 milliseconds for process P3.

Avg waiting time = $\frac{0+24+27}{3}$ = 17 milliseconds.

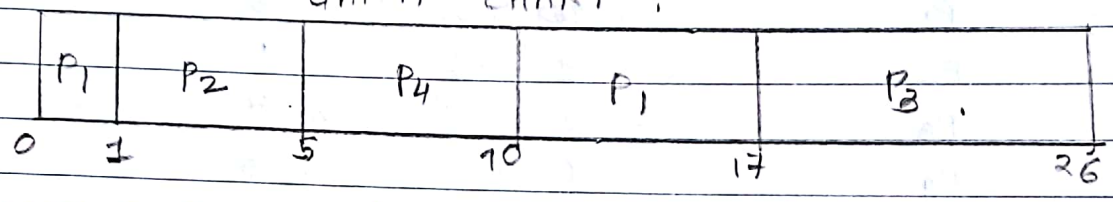
(2) SHORTEST-JOB-FIRST SCHEDULING. [PREEMPTIVE].

This algorithm associates with each process the length of process's next CPU burst. When the CPU is available, it is assigned to process's next CPU burst (smallest). The SJF scheduling algorithm is provably optimal, in that it gives the minimum average waiting time for a given set of processes. It can be either preemptive or non-preemptive. (Preemptive).

Example: Consider the following 4 processes, with the length of CPU burst given in milliseconds.

Process	Arrival time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

GANTT CHART.



Process P_1 is started at time 0, since it is the only process in queue. Process P_2 arrives at time 1. The remaining time for process P_1 (milliseconds) is larger than time required by process P_2 (4 milliseconds), so process P_1 is preempted and process P_2 is scheduled.

$$\text{Avg waiting time} = \frac{[(10-1)+(1-1)+(17-2)+(5-3)]}{4}$$

$$= 26/4 = 6.5 \text{ milliseconds}.$$

(3) PRIORITY (NON-PREEMPTIVE) SCHEDULING ALGORITHM:

The SJF Algorithm is a special case of general priority scheduling Algorithm.

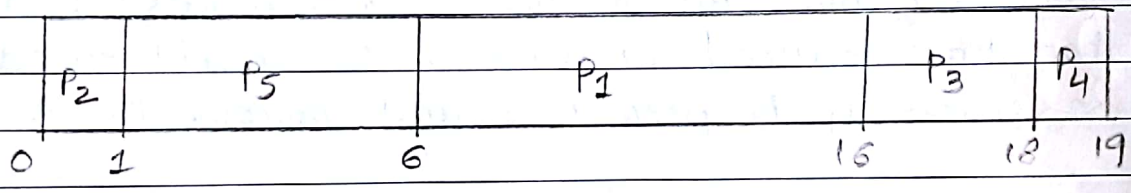
A priority is associated with each process and CPU is allocated to process with highest priority. A non-preemptive priority scheduling algorithm will simply put the new process at head of ready queue.

A major problem with priority scheduling algorithms is indefinite blocking or starvation.

Example: Consider the following set of processes, assumed to have arrived at time 0 in the order $P_1, P_2, \ldots P_5$ with length of CPU burst given in milliseconds.

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

GANTT CHART



The average waiting time is 8.2 milliseconds.

(4) ROUND - ROBIN SCHEDULING (PREEMPTIVE).

The round-robin (RR) scheduling algorithm is designed especially for time sharing systems. It is similar to FCFS scheduling, but preemption is added to enable the system to switch b/w processes.

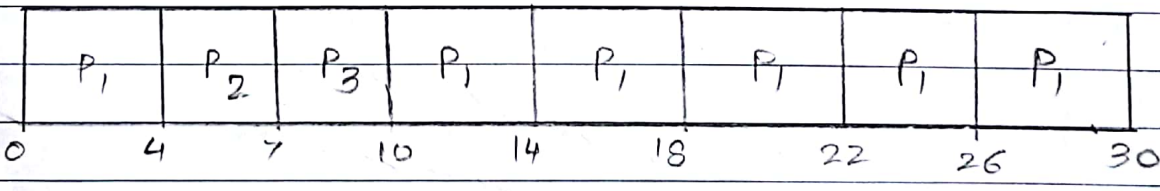
A small unit of time called time quantum or time slice is defined. A time quantum is generally from 10 to 100 milliseconds in length.

Example : consider.
The following set of process that arrive at time 0, with the length of CPU burst given in milliseconds.

Process	BurstTime
P_1	24
P_2	3
P_3	3

If we use a time quantum of 4 msec, P_1 = 4 milliseconds, since it requires another 20 millisec, it is preempted after 1st quantum and CPU is given the next process in queue, P_2. Process P_2 does not need 4 millisec so it quits before it time quantum expires. The CPU is then given to next process, process P_3. Once each process has received 1 time quantum, the CPU is returned to process P_1 for additional time quantum.

GANTT CHART :



Average waiting Time : P_1 waits for 6 milliseconds (10-4), P_2 waits for 4 millisec P_3 waits 7 millisec.

∴ Avg waiting time = 17/3 = 5.66 millisec.

DPU

Dr. D. Y. Patil Institute of Engineering & Technology

(*) CONCLUSION:

Thus we have successfully implemented Java program in scheduling algorithms. [FCFS, SJF (preemptive), priority (Non-preemptive), Round Robin (Preemptive)].

RMK