SRES's Sanjivani College of Engineering, Kopargaon
(*An Autonomous Institute*)
Department of Computer Engineering

SPOS Lab Manual

Assignment No. 09

AIM:
Write a program to implement scheduling algorithms.

PROBLEM DEFINITION:
Write a program to implement following scheduling algorithms : FCFS, SJF (Non Preemptive), Priority (Non Preemptive), and Round Robin (Preemptive).

OBJECTIVES:
1. To understand different scheduling algorithms
2. To understand the difference between preemptive and nonpreemptive scheduling

INPUT:
- Number of Processes
- Process Arrival Times
- Process Burst Times
- Priority of Each Process (for Priority Scheduling only)
- Time Quantum (for Round Robin Scheduling Only)

OUTPUT:
- Waiting Time of Each Process
- Average Waiting Time of all Processes
- Turnaround Time of Each Process
- Average Turnaround Time of all Processes

A] FCFS Scheduling:

THEORY:

First Come First Serve (FCFS) is an operating system scheduling algorithm that automatically executes queued requests and processes in order of their arrival. It is the easiest and simplest CPU scheduling algorithm. In this type of algorithm, processes which requests the CPU first get the CPU allocation first. This is managed with a FIFO queue. The full form of FCFS is First Come First Serve.

As the process enters the ready queue, its PCB (Process Control Block) is linked with the tail of the queue and, when the CPU becomes free, it should be assigned to the process at the beginning of the queue.

Characteristics of FCFS method

- It supports non-preemptive and pre-emptive scheduling algorithm.
- Jobs are always executed on a first-come, first-serve basis.
- It is easy to implement and use.
- This method is poor in performance, and the general wait time is quite high.

Example of FCFS scheduling

A real-life example of the FCFS method is buying a movie ticket on the ticket counter. In this scheduling algorithm, a person is served according to the queue manner. The person who arrives first in the queue first buys the ticket and then the next one. This will continue until the last person in the queue purchases the ticket. Using this algorithm, the CPU process works in a similar manner.

How FCFS Works? Calculating Average Waiting Time

Here is an example of five processes arriving at different times. Each process has a different burst time.

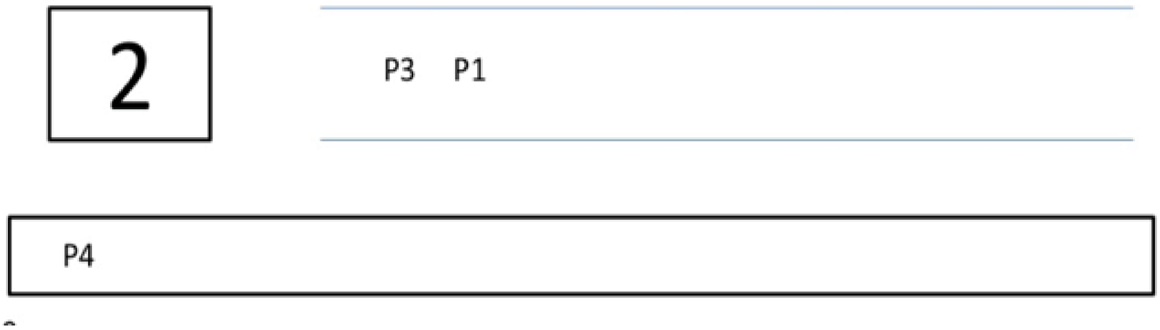
Process	**Burst time**	**Arrival time**
P1	6	2
P2	3	5
P3	8	1
P4	3	0
P5	4	4

Using the FCFS scheduling algorithm, these processes are handled as follows.

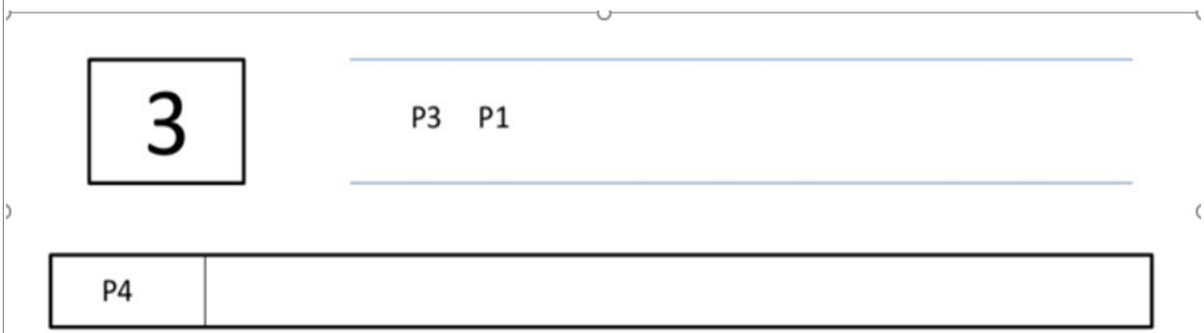
Step 0) The process begins with P4 which has arrival time 0

Step 1) At time=1, P3 arrives. P4 is still executing. Hence, P3 is kept in a queue.

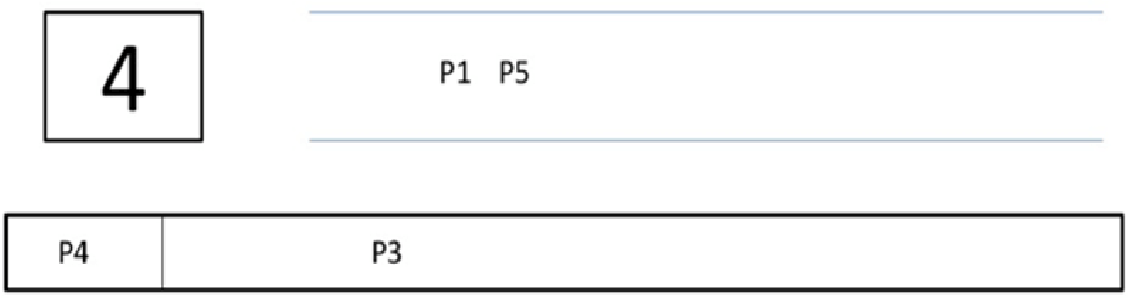
Step 2) At time= 2, P1 arrives which is kept in the queue.



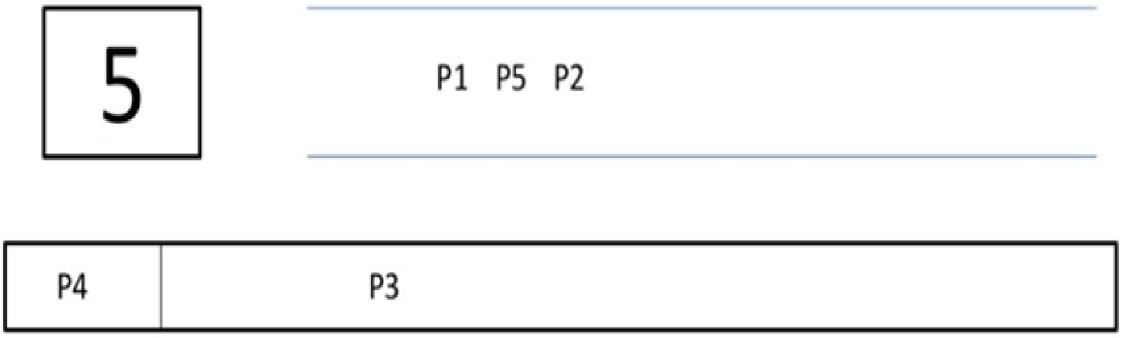
Step 3) At time=3, P4 process completes its execution.



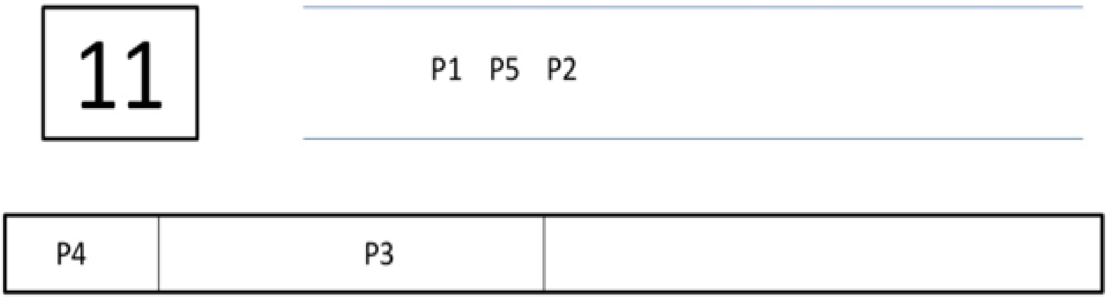
Step 4) At time=4, P3, which is first in the queue, starts execution.



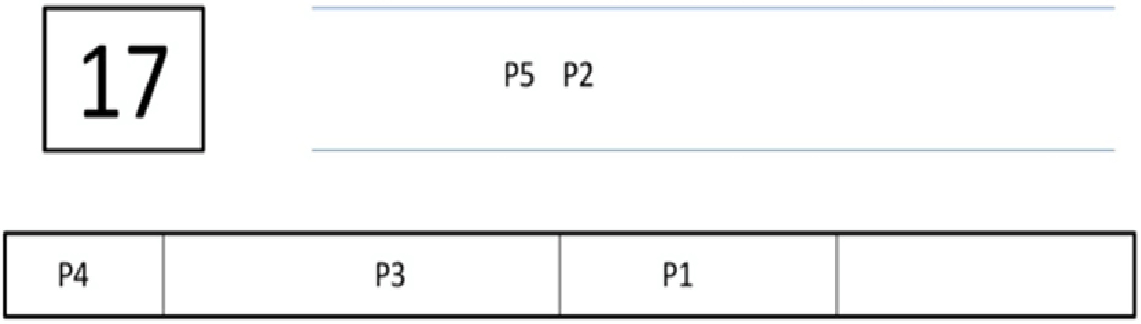
Step 5) At time =5, P2 arrives, and it is kept in a queue.



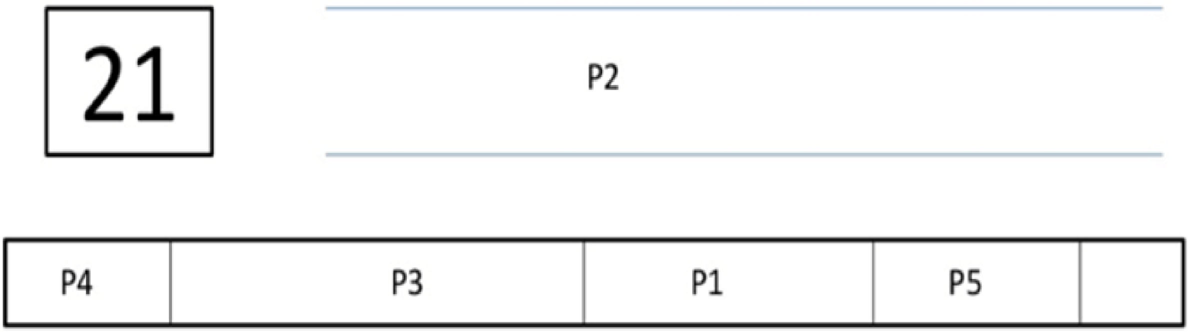
Step 6) At time 11, P3 completes its execution.



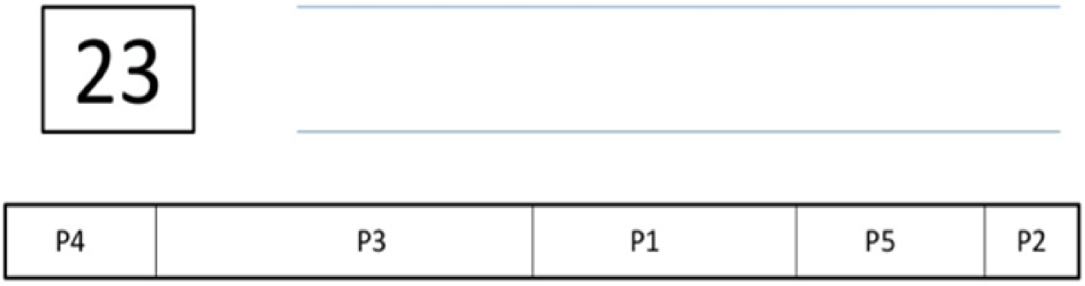
Step 7) At time=11, P1 starts execution. It has a burst time of 6. It completes execution at time interval 17



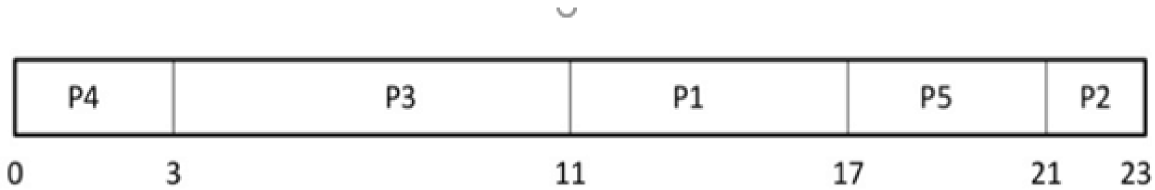
Step 8) At time=17, P5 starts execution. It has a burst time of 4. It completes execution at time=21



Step 9) At time=21, P2 starts execution. It has a burst time of 2. It completes execution at time interval 23



Step 10) Let's calculate the average waiting time for above example.



Waiting Time = Start Time – Arrival Time

- P4 = 0 - 0 = 0
- P3 = 3 – 1 = 2
- P1 = 11 – 2 = 9
- P5 = 17 – 4 = 13
- P2 = 21 – 5 = 16

Average Waiting Time = (0+2+9+13+16) / 5 = 40/5 = 8

Turnaround Time = Finish Time – Arrival time

- P4 = 3 – 0 = 3
- P3 = 11 – 1 = 10
- P1 = 17 – 2 = 15
- P5 = 21 – 4 = 17
- P2 = 23 – 5 = 18

Average Turnaround Time = (3+10+15+17+18)/5 = 63/5 = 12.6

Advantages of FCFS

Here, are pros/benefits of using FCFS scheduling algorithm:

- The simplest form of a CPU scheduling algorithm
- Easy to program
- First come first served

Disadvantages of FCFS

Here, are cons/ drawbacks of using FCFS scheduling algorithm:

- It is a Non-Preemptive CPU scheduling algorithm, so after the process has been allocated to the CPU, it will never release the CPU until it finishes executing.
- The Average Waiting Time is high.
- Short processes that are at the back of the queue have to wait for the long process at the front to finish.
- Not an ideal technique for time-sharing systems.
- Because of its simplicity, FCFS is not very efficient.

B] Non Preemptive SJF Scheduling:

What is Shortest Job First Scheduling?

Shortest Job First (SJF) is an algorithm in which the process having the smallest execution time is chosen for the next execution. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution. The full form of SJF is Shortest Job First.

Characteristics of SJF Scheduling

- It is associated with each job as a unit of time to complete.
- This algorithm method is helpful for batch-type processing, where waiting for jobs to complete is not critical.
- It can improve process throughput by making sure that shorter jobs are executed first, hence possibly have a short turnaround time.
- It improves job output by offering shorter jobs, which should be executed first, which mostly have a shorter turnaround time.

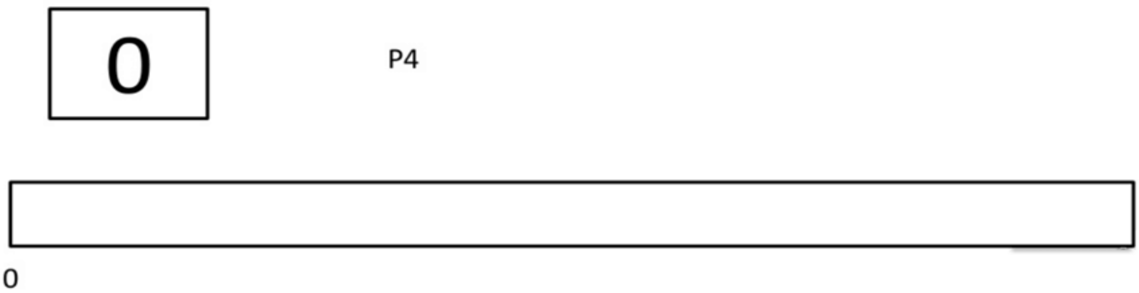
Non-Preemptive SJF

In non-preemptive scheduling, once the CPU cycle is allocated to process, the process holds it till it reaches a waiting state or terminated.

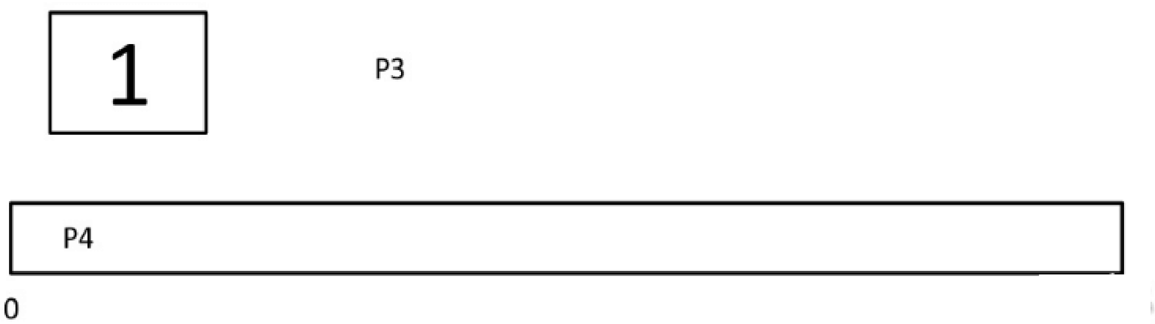
Consider the following five processes each having its own unique burst time and arrival time.

Process	**Burst time**	**Arrival time**
P1	6	2
P2	2	5
P3	8	1
P4	3	0
P5	4	4

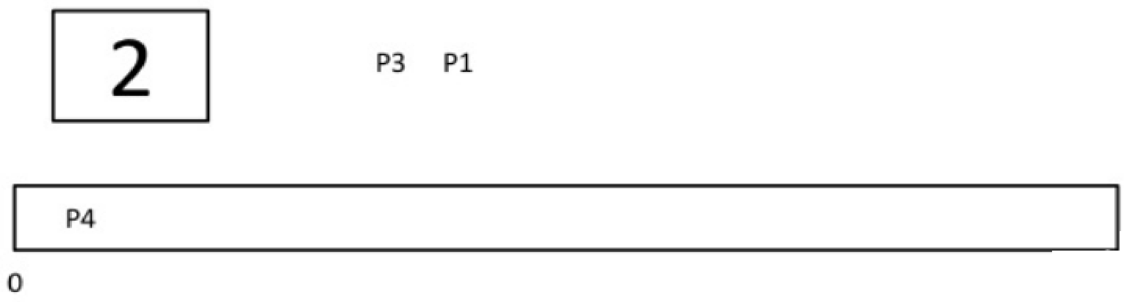
Step 0) At time=0, P4 arrives and starts execution.



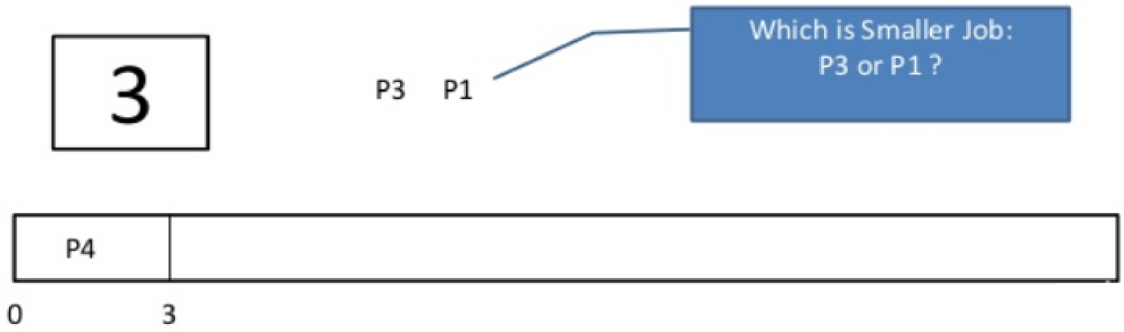
Step 1) At time= 1, Process P3 arrives. But, P4 still needs 2 execution units to complete. It will continue execution.



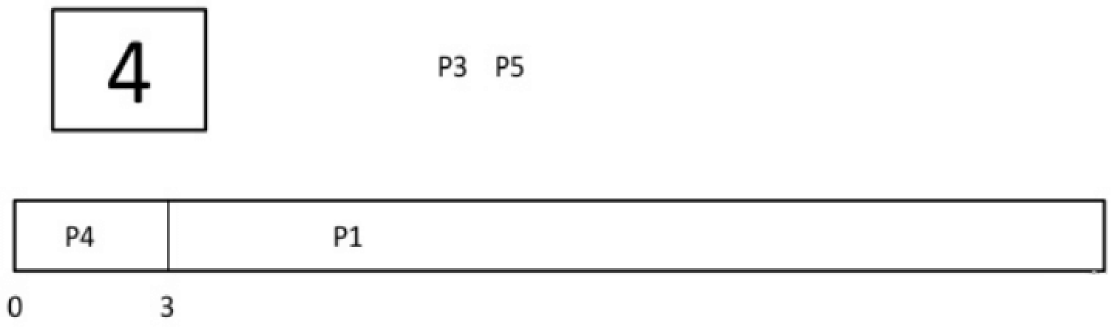
Step 2) At time =2, process P1 arrives and is added to the waiting queue. P4 will continue execution.



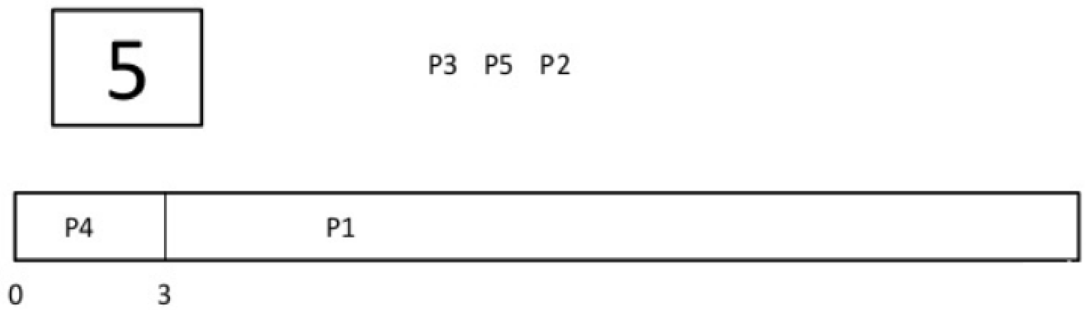
Step 3) At time = 3, process P4 will finish its execution. The burst time of P3 and P1 is compared. Process P1 is executed because its burst time is less compared to P3.



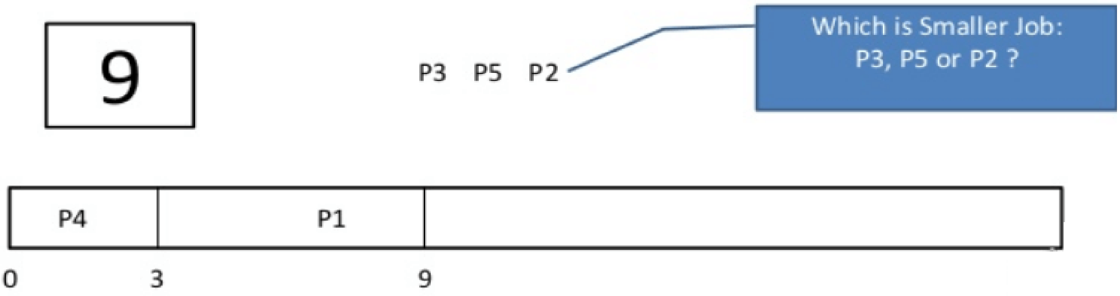
Step 4) At time = 4, process P5 arrives and is added to the waiting queue. P1 will continue execution.



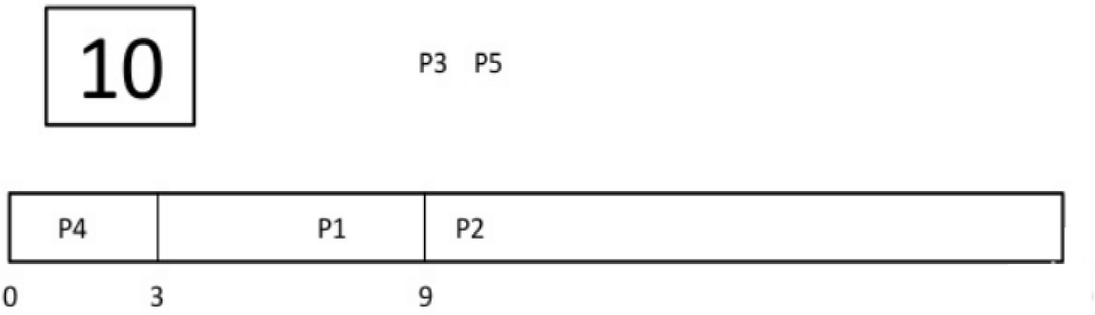
Step 5) At time = 5, process P2 arrives and is added to the waiting queue. P1 will continue execution.



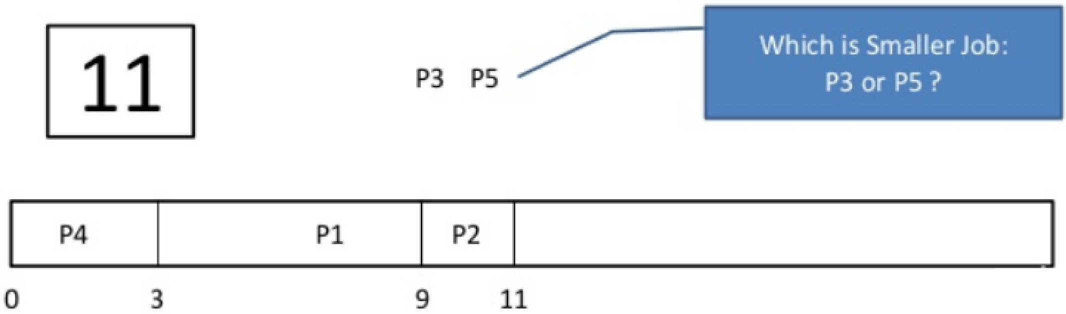
Step 6) At time = 9, process P1 will finish its execution. The burst time of P3, P5, and P2 is compared. Process P2 is executed because its burst time is the lowest.



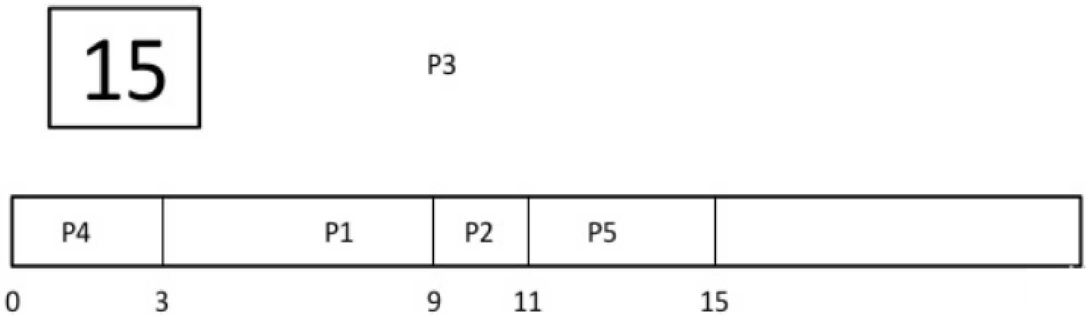
Step 7) At time=10, P2 is executing and P3 and P5 are in the waiting queue.



Step 8) At time = 11, process P2 will finish its execution. The burst time of P3 and P5 is compared. Process P5 is executed because its burst time is lower.

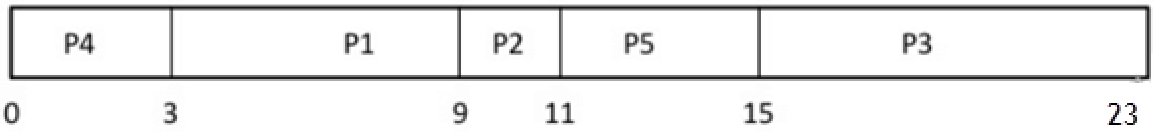


Step 9) At time = 15, process P5 will finish its execution.



Step 10) At time = 23, process P3 will finish its execution.

23



Wait time = Start Time – Arrival time

- P4 = 0-0=0
- P1 = 3-2=1
- P2 = 9-5=4
- P5 = 11-4=7
- P3 = 15-1=14

Average Waiting Time= 0+1+4+7+14/5 = 26/5 = 5.2

Turnaround Time = Finish Time – Arrival Time

- P4 = 3 – 0 =3
- P1 = 9 -2 = 7
- P2 = 11 – 5 = 6
- P5 = 15 – 4 = 11
- P3 = 23 – 1 = 22

Average Turnaround Time = (3+7+6+11+22) / 5 = 49/5 = 9.8

Advantages of SJF

Here are the benefits/pros of using SJF method:

- SJF is frequently used for long term scheduling.
- It reduces the average waiting time over FIFO (First in First Out) algorithm.
- SJF method gives the lowest average waiting time for a specific set of processes.
- It is appropriate for the jobs running in batch, where run times are known in advance.
- For the batch system of long-term scheduling, a burst time estimate can be obtained from the job description.
- For Short-Term Scheduling, we need to predict the value of the next burst time.
- Probably optimal with regard to average turnaround time.

Disadvantages/Cons of SJF

Here are some drawbacks/cons of SJF algorithm:

- Job completion time must be known earlier, but it is hard to predict.

- It is often used in a batch system for long term scheduling.
- SJF can't be implemented for CPU scheduling for the short term. It is because there is no specific method to predict the length of the upcoming CPU burst.
- This algorithm may cause very long turnaround times or starvation.
- Requires knowledge of how long a process or job will run.
- It leads to the starvation that does not reduce average turnaround time.
- It is hard to know the length of the upcoming CPU request.
- Elapsed time should be recorded, that results in more overhead on the processor.

C] Non Preemptive Priority Scheduling :

What is Priority Scheduling?

Priority Scheduling is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority.

The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority depends upon memory requirements, time requirements, etc.

Non-Preemptive Scheduling

In this type of scheduling method, the CPU has been allocated to a specific process. The process that keeps the CPU busy, will release the CPU either by switching context or terminating. It is the only method that can be used for various hardware platforms. That's because it doesn't need special hardware (for example, a timer) like preemptive scheduling.

Characteristics of Priority Scheduling

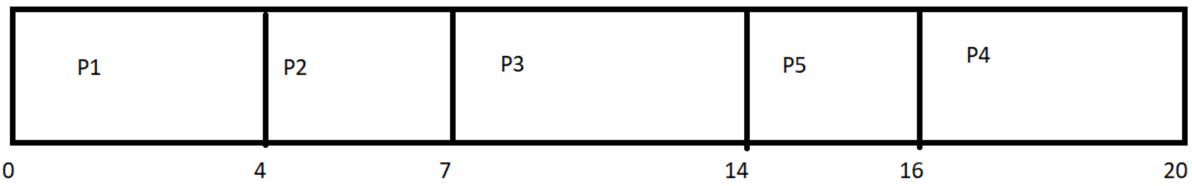
- A CPU algorithm that schedules processes based on priority.
- It used in Operating systems for performing batch processes.
- If two jobs having the same priority are READY, it works on a FIRST COME, FIRST SERVED basis.
- In priority scheduling, a number is assigned to each process that indicates its priority level.
- Lower the number, higher is the priority.
- In this type of scheduling algorithm, if a newer process arrives, that is having a higher priority than the currently running process, then the currently running process is preempted.

Example of Priority Scheduling

Consider following five processes P1 to P5. Each process has its unique priority, burst time, and arrival time.

Process	Priority	Burst time	Arrival time
P1	1	4	0
P2	2	3	0
P3	1	7	6
P4	3	4	11
P5	2	2	12

Gantt Chart:



Waiting Time = Start Time – Arrival Time

- P1 = 0 – 0 = 0
- P2 = 4 – 0 = 4
- P3 = 7 – 6 = 1
- P4 = 16 – 11 = 5
- P5 = 14 – 12 = 2

Average Waiting Time = (0+4+1+5+2)/5 = 12/5 = 2.4

Turnaround Time = Finish Time – Arrival Time

- P1 = 4 – 0 = 4
- P2 = 7 – 0 = 7
- P3 = 14 – 6 = 8
- P4 = 20 – 11 = 9
- P5 = 16 – 12 = 4

Average Turnaround Time = (4+7+8+9+4) / 5 = 32/5 = 6.4

Advantages of priority scheduling

Here, are benefits/pros of using priority scheduling method:

- Easy to use scheduling method
- Processes are executed on the basis of priority so high priority does not need to wait for long which saves time
- This method provides a good mechanism where the relative important of each process may be precisely defined.
- Suitable for applications with fluctuating time and resource requirements.

Disadvantages of priority scheduling

Here, are cons/drawbacks of priority scheduling

- If the system eventually crashes, all low priority processes get lost.
- If high priority processes take lots of CPU time, then the lower priority processes may starve and will be postponed for an indefinite time.
- This scheduling algorithm may leave some low priority processes waiting indefinitely.
- A process will be blocked when it is ready to run but has to wait for the CPU because some other process is running currently.
- If a new higher priority process keeps on coming in the ready queue, then the process which is in the waiting state may need to wait for a long duration of time.

D] Round Robin Scheduling:

What is Round-Robin Scheduling?

The name of this algorithm comes from the round-robin principle, where each person gets an equal share of something in turns. It is the oldest, simplest scheduling algorithm, which is mostly used for multitasking.

In Round-robin scheduling, each ready task runs turn by turn only in a cyclic queue for a limited time slice. This algorithm also offers starvation free execution of processes.

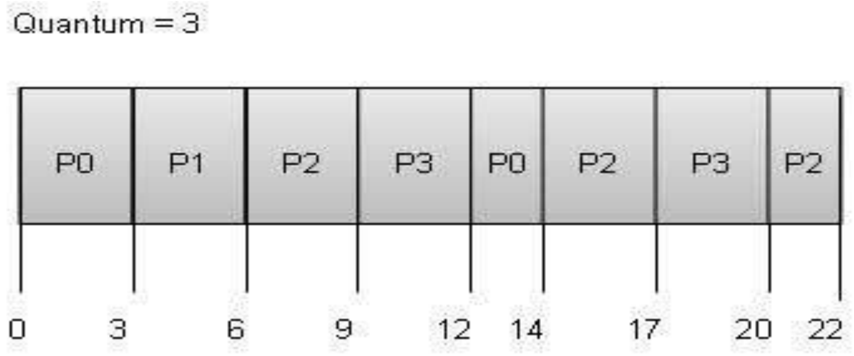
Characteristics of Round-Robin Scheduling

Here are the important characteristics of Round-Robin Scheduling:

- Round robin is a pre-emptive algorithm
- The CPU is shifted to the next process after fixed interval time, which is called time quantum/time slice.
- The process that is preempted is added to the end of the queue.
- Round robin is a hybrid model which is clock-driven
- Time slice should be minimum, which is assigned for a specific task that needs to be processed. However, it may differ OS to OS.
- It is a real time algorithm which responds to the event within a specific time limit.
- Round robin is one of the oldest, fairest, and easiest algorithm.
- Widely used scheduling method in traditional OS.

Example of Round Robin Scheduling:

Process	**Arrival Time**	**Burst Time**
P0	0	5
P1	1	3
P2	2	8
P3	3	6



Waiting Time = Start Time – Arrival Time

- P0 = (0-0) + (12-3) = 9
- P1 = (3-1) = 2
- P2 = (6-2) + (14-9) + (20-17) = 12
- P3 = (9-3) + (17-12) = 11

Average Waiting Time = (9+2+12+11) / 4 = 34/4 = 8.5

Turnaround Time = Finish Time – Arrival Time

- P0 = 14 – 0 = 14
- P1 = 6 – 1 = 5
- P2 = 22 – 2 = 20
- P3 = 20 – 3 = 17

Average turnaround Time = (14+5+20+17) / 4 = 56/4 = 14

Advantage of Round-robin Scheduling

Here, are pros/benefits of Round-robin scheduling method:

- It doesn't face the issues of starvation or convoy effect.
- All the jobs get a fair allocation of CPU.
- It deals with all process without any priority
- If you know the total number of processes on the run queue, then you can also assume the worst-case response time for the same process.
- This scheduling method does not depend upon burst time. That's why it is easily implementable on the system.
- Once a process is executed for a specific set of the period, the process is preempted, and another process executes for that given time period.
- Allows OS to use the Context switching method to save states of preempted processes.
- It gives the best performance in terms of average response time.

Disadvantages of Round-robin Scheduling

Here, are drawbacks/cons of using Round-robin scheduling:

- If slicing time of OS is low, the processor output will be reduced.
- This method spends more time on context switching
- Its performance heavily depends on time quantum.
- Priorities cannot be set for the processes.
- Round-robin scheduling doesn't give special priority to more important tasks.
- Decreases comprehension
- Lower time quantum results in higher the context switching overhead in the system.
- Finding a correct time quantum is a quite difficult task in this system.

CONCLUSION: In this assignment we have studied implementation of different CPU Scheduling algorithms also computation of waiting and turnaround times by each of the algorithm

References: Operating System Concepts By Abraham Silberschatz, Peter B. Galvin, Greg Gagne

Prepared by	**Approved by**
Prof.N.G.Pardeshi	**Dr. D.B.Kshirsagar**
Subject Teacher	**HOD**