CPU Scheduling Algorithms

CPU Scheduling algorithms :-

CPU Scheduling algorithm is an algorithm which is used to assign system resources to processes in a computing system. Consider the case where you are using two apps namely a game like Fortnite and a desktop application like Evernote. Both with require the use of a graphics processor and but only one can use it at a time. It is the CPU scheduling algorithms which manages which process will use a given resource at a time. The focus of such algorithms is to maximize CPU resources usage and minimize waiting time for each process.

The different CPU algorithms are:

- First Come First Serve
- Shortest Job First
- Shortest Remaining Time First
- Round Robin Scheduling
- Priority Scheduling
- Multilevel Queue Scheduling
- Multilevel Feedback Queue Scheduling

And we will discuss some this algorithms in this report

Key terms to understand different algorithms:

- **Arrival Time :** Time at which any process arrives in ready queue.
- **Burst Time :** Time required by CPU for execution of a process. It is also called as *Running Time* or *Execution Time*.
- **Completion Time :** Time at which process completes execution.
- **Turn Around Time :** Time difference between Completion Time and Arrival Time (*Completion Time - Arrival Time*)
- **Waiting Time :** Time difference between Turn Around Time and Burst Time (*Turn Around Time - Burst Time*)
- **Response Time :** Time after which any process gets CPU after entering the ready queue.
- **Preemptive Scheduling :** It is used if there is process switching from running state to ready state or from ready state to waiting state.
- Resources allocated to a process are for a limited time.
- Process can be interruped in between.
- In case, high priority process arrives, low priority processes may starve.
- It is flexible in nature.
- **Non-Preemptive Scheduling :** It is used if any process terminates or there is process switching from running to waiting state.
- Resources allocated to a process arehold untill it terminates or it switches to waiting state.
- Process can't be interrupted in between.

- In case, process with high burst time is running, other processes may starve.
- It is rigid in nature.

Different Scheduling Algorithms

1) **First Come First Serve(FCFS)** : It is a simple scheduling algorithm. The idea is that the process that comes first must use the resource first.

   It schedules according to the arrival time of the process. It states that process which request the CPU first is allocated the CPU first.

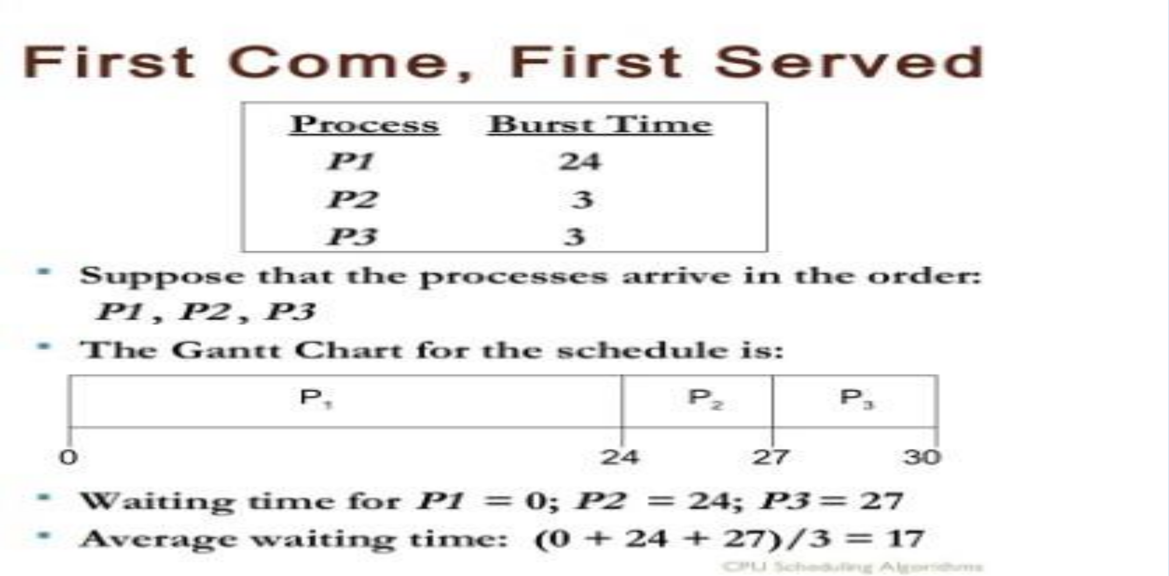
   It is implemented by using FIFO (First Come First Serve) queue and is a Non preemptive scheduling algorithm.

Advantages of FCFS:

- Simple , Easy, useful and understandable □ First come, first served.

Disadvantages of FCFS :

- Because of non-preemptive scheduling, the process will continue to run untill it is finished.
- As the scheduling is non-preemptive so short processes which are at end of ready queue have to wait for a larger time making them starve leading to problem of starvation.
- Throughput is not efficient.



First Come, First Served

Process	Burst Time
P1	24
P2	3
P3	3

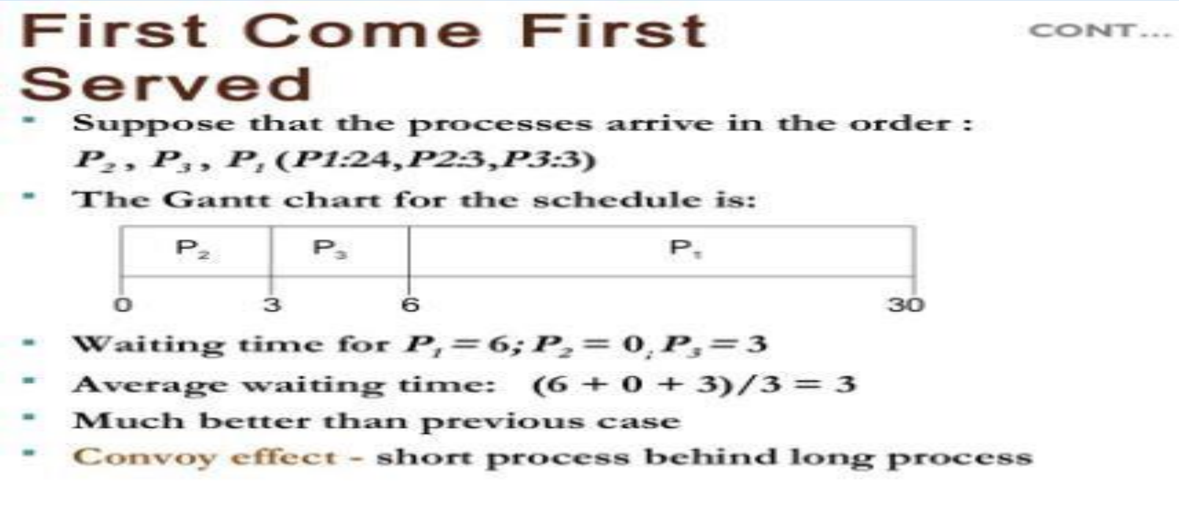
- Suppose that the processes arrive in the order:
  P1 , *P2* , *P3*
- The Gantt Chart for the schedule is:

P_1	P_2	P_3

0 24 27 30

- Waiting time for *P1* = 0; *P2* = 24; *P3* = 27
- Average waiting time: (0 + 24 + 27)/3 = 17

CPU Scheduling Algorithms



First Come First Served

CONT...

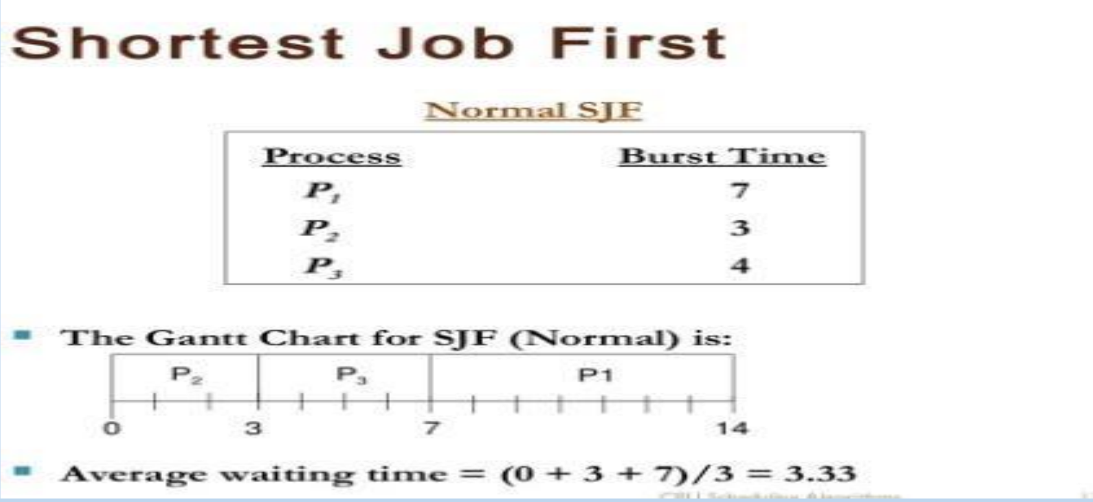
- Suppose that the processes arrive in the order :
  P_2 , P_3 , P_1 (*P1:24,P2:3,P3:3*)
- The Gantt chart for the schedule is:

P_2	P_3	P_1

0 3 6 30

- Waiting time for $P_1 = 6; P_2 = 0_; P_3 = 3$
- Average waiting time: (6 + 0 + 3)/3 = 3
- Much better than previous case
- Convoy effect - short process behind long process

2) **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 nonpreemptive. It significantly reduces the average waiting time for other processes awaiting execution. The full form of SJF is Shortest Job First.



Shortest Job First

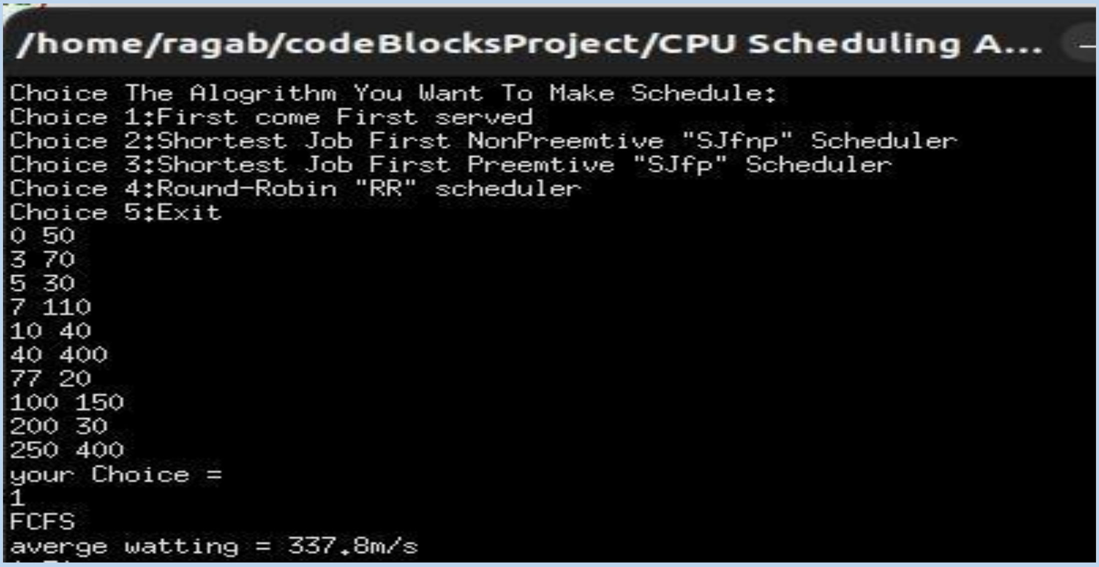
Normal SJF

Process	Burst Time
P_1	7
P_2	3
P_3	4

- The Gantt Chart for SJF (Normal) is:

P_2
P_3
P1
0
3
7
14

- Average waiting time = (0 + 3 + 7)/3 = 3.33

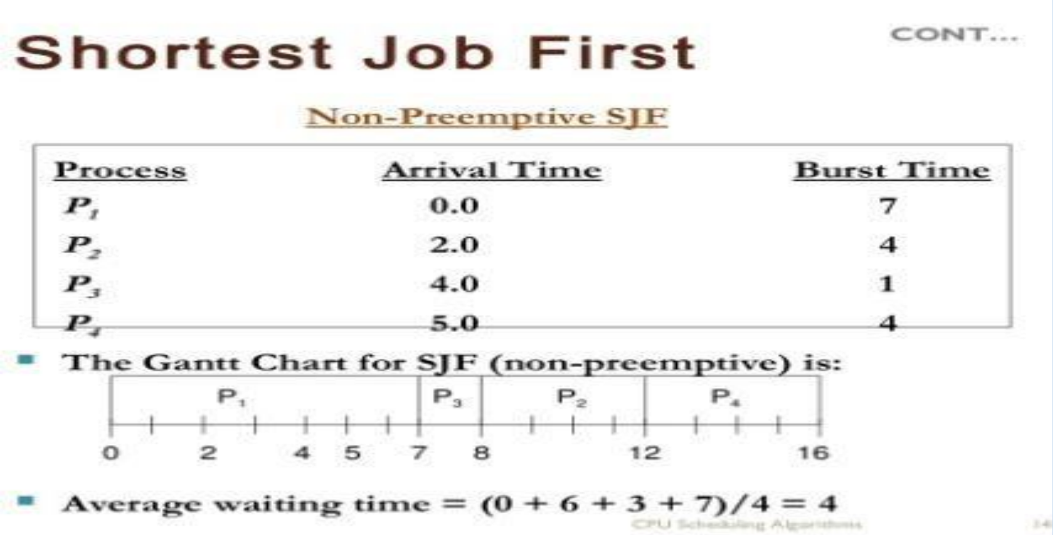


/home/ragab/codeBlocksProject/CPU Scheduling A...

```
Choice The Alogrithm You Want To Make Schedule:
Choice 1:First come First served
Choice 2:Shortest Job First NonPreemtive "SJfnp" Scheduler
Choice 3:Shortest Job First Preemtive "SJfp" Scheduler
Choice 4:Round-Robin "RR" scheduler
Choice 5:Exit
0 50
3 70
5 30
7 110
10 40
40 400
77 20
100 150
200 30
250 400
your Choice =
1
FCFS
averge watting = 337.8m/s
```

There are basically two types of SJF methods:

- Non-Preemptive SJF



Shortest Job First

CONT...

Non-Preemptive SJF

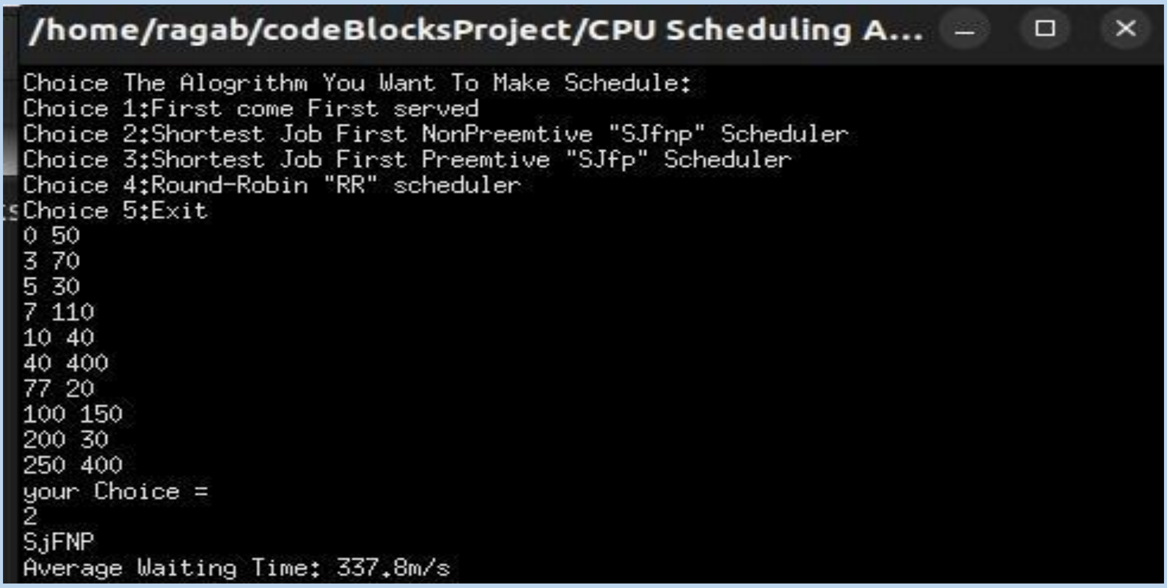
Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- The Gantt Chart for SJF (non-preemptive) is:

P_1
P_3
P_2
P_4
0
2
4
5
7
8
12
16

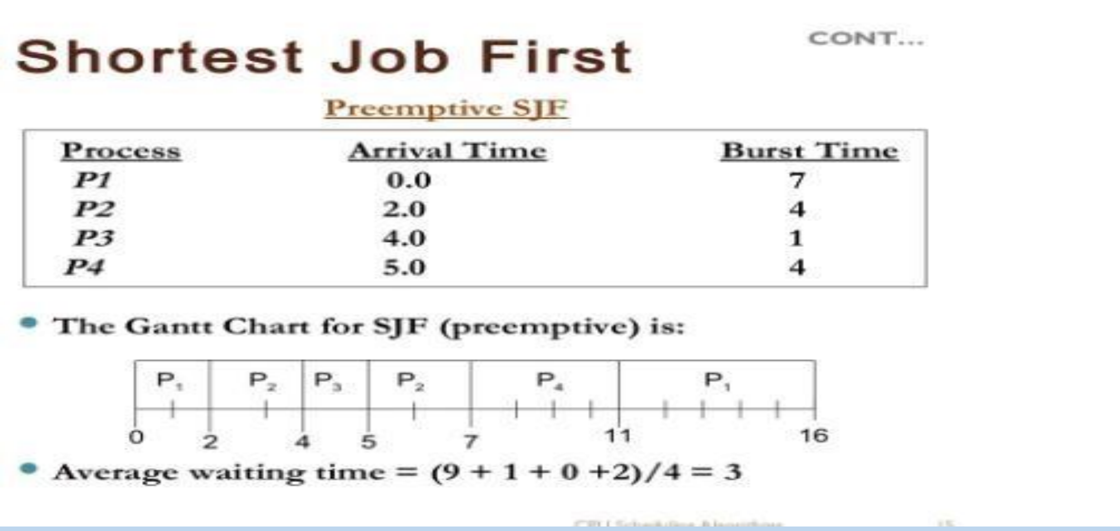
- Average waiting time = (0 + 6 + 3 + 7)/4 = 4

CPU Scheduling Algorithms



```
Choice The Alogrithm You Want To Make Schedule:
Choice 1:First come First served
Choice 2:Shortest Job First NonPreemtive "SJfnp" Scheduler
Choice 3:Shortest Job First Preemtive "SJfp" Scheduler
Choice 4:Round-Robin "RR" scheduler
Choice 5:Exit
0 50
3 70
5 30
7 110
10 40
40 400
77 20
100 150
200 30
250 400
your Choice =
2
SjFNP
Average Waiting Time: 337.8m/s
```

- Preemptive SJF



Shortest Job First

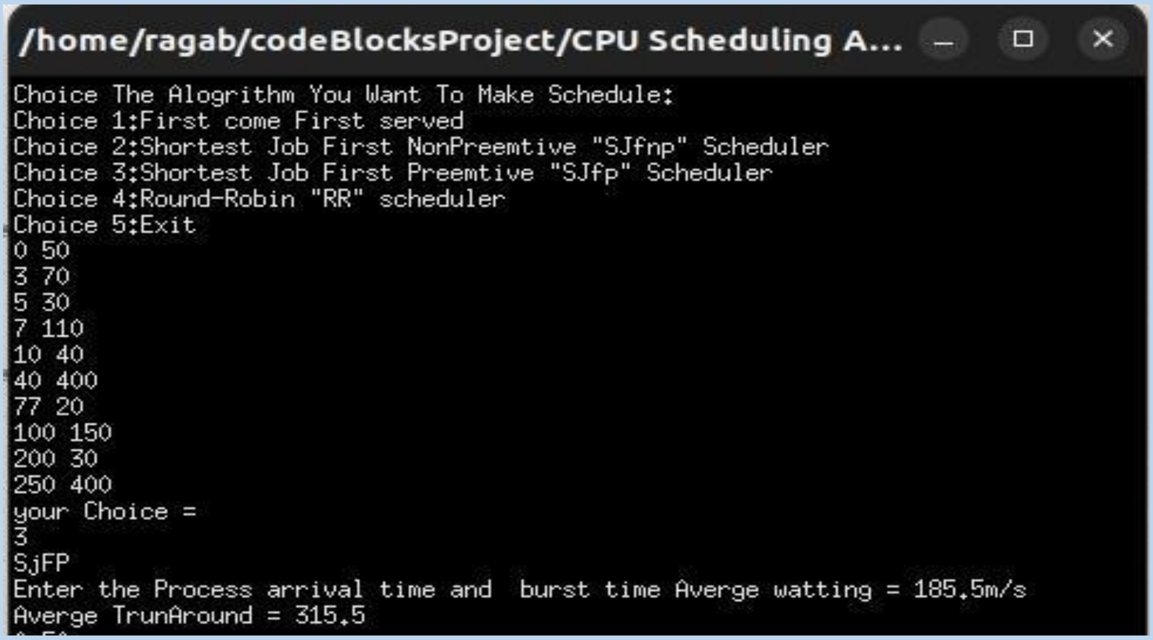
CONT...

Preemptive SJF

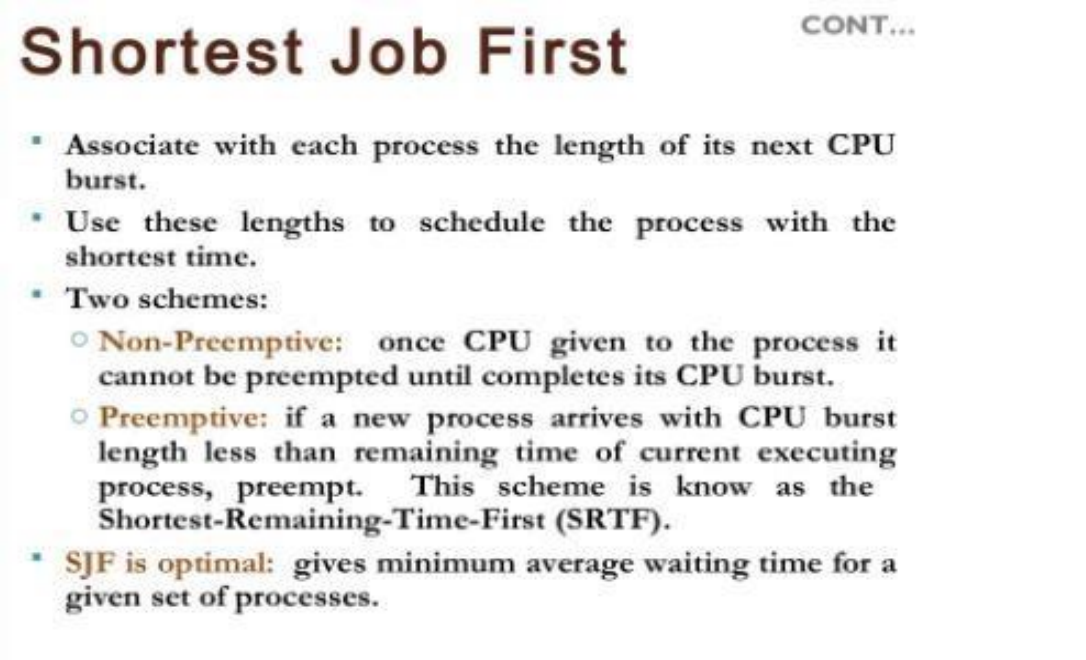
Process	Arrival Time	Burst Time
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4

- The Gantt Chart for SJF (preemptive) is:

- Average waiting time = (9 + 1 + 0 +2)/4 = 3



```
Choice The Alogrithm You Want To Make Schedule:
Choice 1:First come First served
Choice 2:Shortest Job First NonPreemtive "SJfnp" Scheduler
Choice 3:Shortest Job First Preemtive "SJfp" Scheduler
Choice 4:Round-Robin "RR" scheduler
Choice 5:Exit
0 50
3 70
5 30
7 110
10 40
40 400
77 20
100 150
200 30
250 400
your Choice =
3
SjFP
Enter the Process arrival time and  burst time Averge watting = 185.5m/s
Averge TrunAround = 315.5
```



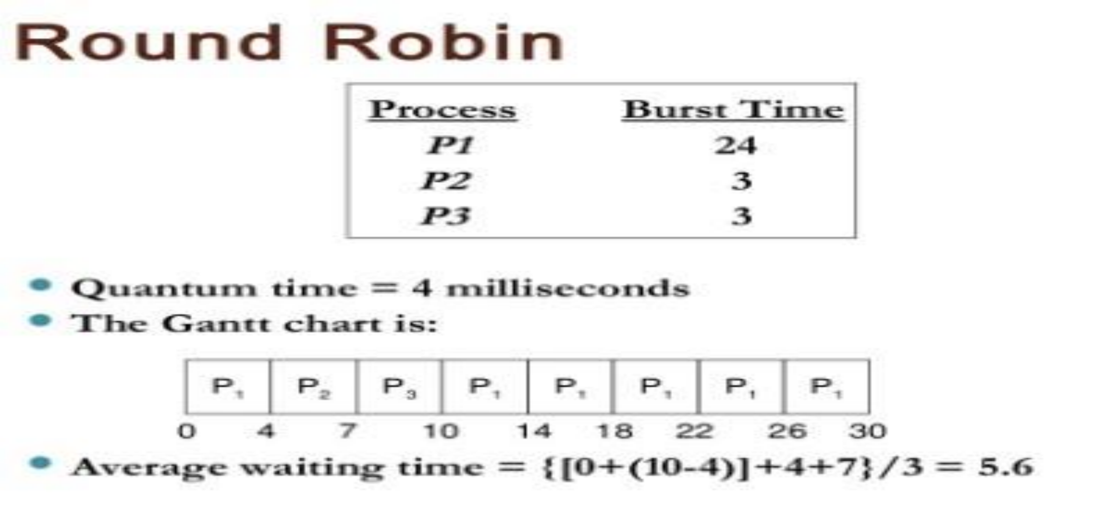
Shortest Job First

CONT…

- Associate with each process the length of its next CPU burst.
- Use these lengths to schedule the process with the shortest time.
- Two schemes:
  - 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 know as the Shortest-Remaining-Time-First (SRTF).
- SJF is optimal: gives minimum average waiting time for a given set of processes.

3) Round Robin Scheduling

Each process is assigned a Time Quantum in a cyclic way. It is designed specially for Time-Sharing system so the execution of ready queue must be in form of circular queue. CPU is alloted to each process for time interval of one time quantum. New processes are added at the end of ready queue. If process has burst time less than a time quantum, the process will release itself of CPU voluntarily after its execution. But, if burst time is greater than a time quantum, then after the time quantum, interrupt will occur and process will be put at the end of ready queue by executing a Context Switching.



Round Robin

Process	Burst Time
P1	24
P2	3
P3	3

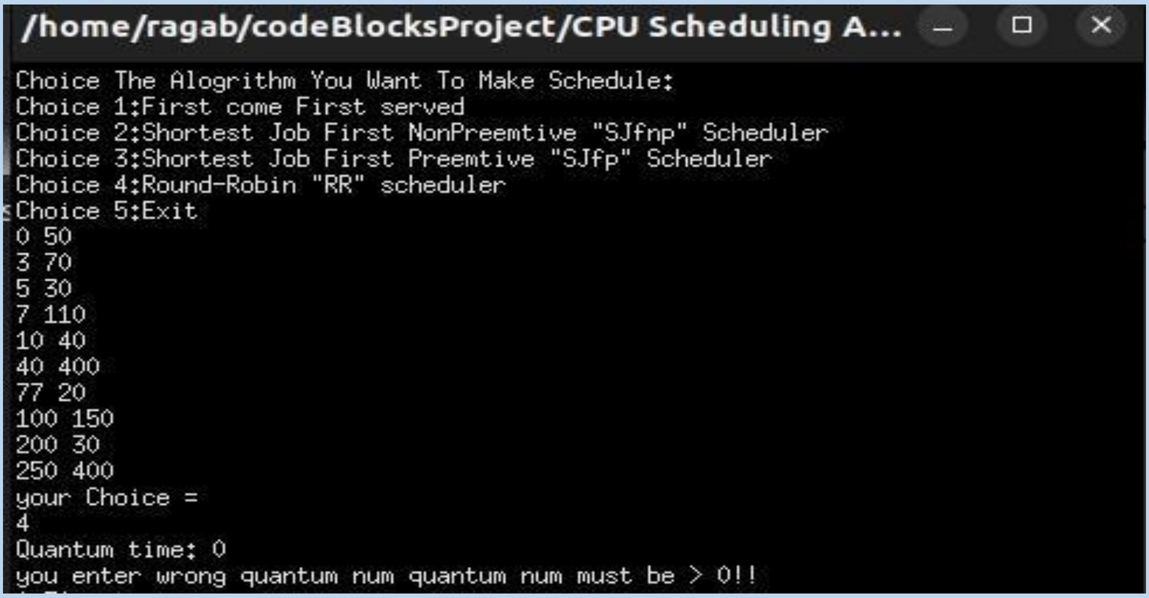
- Quantum time = 4 milliseconds
- The Gantt chart is:

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1

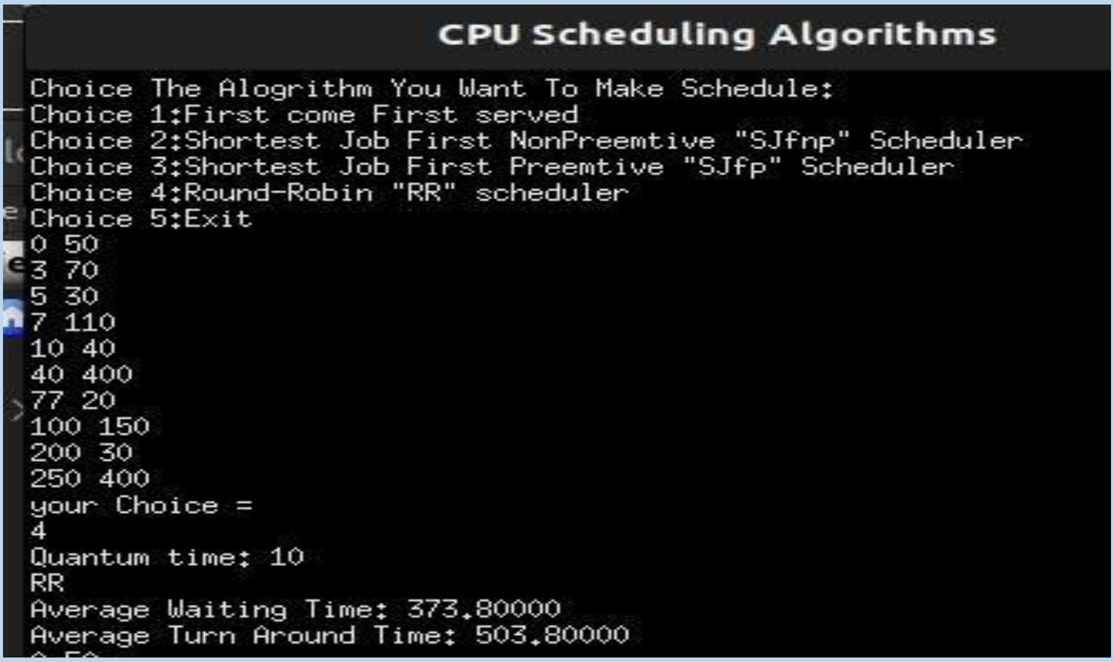
0 4 7 10 14 18 22 26 30

- Average waiting time = {[0+(10-4)]+4+7}/3 = 5.6

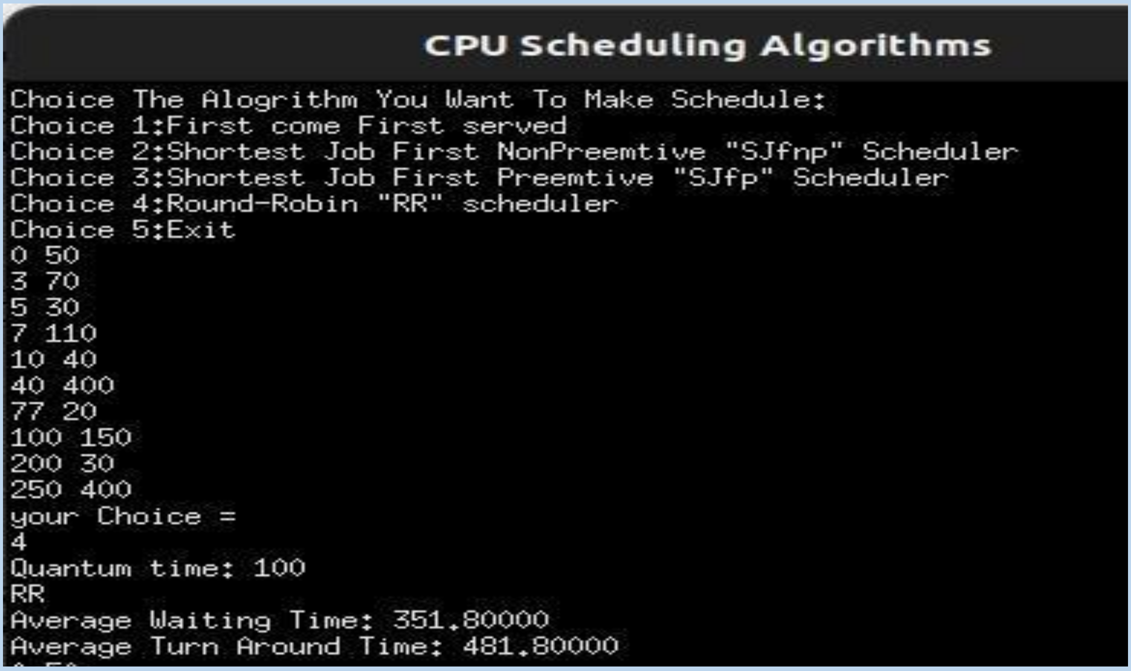
- **Round-Robin with Quantum equal to zero & 10 & 100 & 1000**



```
Choice The Alogrithm You Want To Make Schedule:
Choice 1:First come First served
Choice 2:Shortest Job First NonPreemtive "SJfnp" Scheduler
Choice 3:Shortest Job First Preemtive "SJfp" Scheduler
Choice 4:Round-Robin "RR" scheduler
Choice 5:Exit
0 50
3 70
5 30
7 110
10 40
40 400
77 20
100 150
200 30
250 400
your Choice =
4
Quantum time: 0
you enter wrong quantum num quantum num must be > 0!!
```

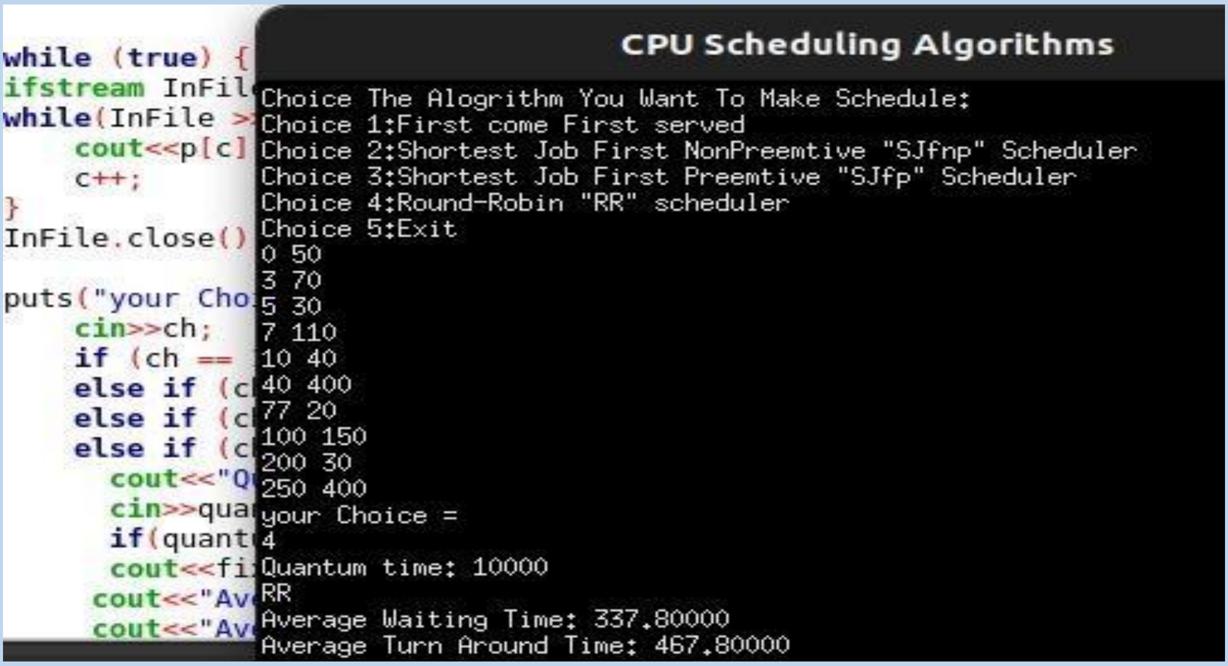


```
Choice The Alogrithm You Want To Make Schedule:
Choice 1:First come First served
Choice 2:Shortest Job First NonPreemtive "SJfnp" Scheduler
Choice 3:Shortest Job First Preemtive "SJfp" Scheduler
Choice 4:Round-Robin "RR" scheduler
Choice 5:Exit
0 50
3 70
5 30
7 110
10 40
40 400
77 20
100 150
200 30
250 400
your Choice =
4
Quantum time: 10
RR
Average Waiting Time: 373.80000
Average Turn Around Time: 503.80000
```



CPU Scheduling Algorithms

```
Choice The Alogrithm You Want To Make Schedule:
Choice 1:First come First served
Choice 2:Shortest Job First NonPreemtive "SJfnp" Scheduler
Choice 3:Shortest Job First Preemtive "SJfp" Scheduler
Choice 4:Round-Robin "RR" scheduler
Choice 5:Exit
0 50
3 70
5 30
7 110
10 40
40 400
77 20
100 150
200 30
250 400
your Choice =
4
Quantum time: 100
RR
Average Waiting Time: 351.80000
Average Turn Around Time: 481.80000
```



```
while (true) {
ifstream InFil
while(InFile >
    cout<<p[c]
    c++;
}
InFile.close()

puts("your Cho
    cin>>ch;
    if (ch ==
    else if (c
    else if (c
    else if (c
      cout<<"Q
      cin>>qua
      if(quant
      cout<<fi
     cout<<"Av
     cout<<"Av
```

CPU Scheduling Algorithms

```
Choice The Alogrithm You Want To Make Schedule:
Choice 1:First come First served
Choice 2:Shortest Job First NonPreemtive "SJfnp" Scheduler
Choice 3:Shortest Job First Preemtive "SJfp" Scheduler
Choice 4:Round-Robin "RR" scheduler
Choice 5:Exit
0 50
3 70
5 30
7 110
10 40
40 400
77 20
100 150
200 30
250 400
your Choice =
4
Quantum time: 10000
RR
Average Waiting Time: 337.80000
Average Turn Around Time: 467.80000
```