SYSTEM SOFTWARE LABORATORY

7. Design, develop and implement a C/C++/Java program to simulate the working of Shortest remaining time and Round Robin (RR) scheduling algorithms. Experiment with different quantum sizes for RR algorithm.

```
#include<stdio.h>
int main()
{
	int count,j,n,time,flag=0,time_quantum,ch=0;
	int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
	int endTime,i,smallest;
	int finished=0,sum_wait=0,sum_turnaround=0;
	printf("1.Round Robin \n2.SRTF \n");
	scanf("%d",&ch);
	printf("Enter no of Processes : ");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		printf("Enter arrival time for Process P%d : ",i);
		scanf("%d",&at[i]);
		printf("Enter burst time for Process P%d :",i);
		scanf("%d",&bt[i]);
		rt[i]=bt[i];
	}
	switch(ch)
	{
		case 1:
		printf("Enter Time Quantum:\t");
			break;
```

case 2:

```
finished=0;
printf("\n   Process   |    Turnaround Time    |    Waiting Time    \n");
rt[9]=9999;
for(time=0;finished!=n;time++)
{
        smallest=9;
        for(i=0;i<n;i++)
                if(at[i]<=time && rt[i]<rt[smallest] && rt[i]>0)
                        smallest=i;
        rt[smallest]--;
        if(rt[smallest]==0)
        {
                finished++;
                endTime=time+1;
/*    TurnAroundTime = EndTime – ArrivalTime
     WaitingTime = EndTime – ArrivalTime – BurstTime    */

printf("\nP[%d]\t|\t%d\t|\t%d",smallest,endTime-at[smallest],   endTime-bt[smallest]-at[smallest]);
                printf("\n");
                sum_wait+=endTime-bt[smallest]-at[smallest];
                sum_turnaround+=endTime-at[smallest];
        }
}
printf("\nAverage waiting time = %f\n",sum_wait*1.0/n);
printf("Average Turnaround time = %f",sum_turnaround*1.0/n);
break;
default:
printf("Invalid\n");
```

```
        }
        return 0;
}
```

Consider the set of 5 processes whose arrival time and burst time are given below-

Process Id	Arrival time	Burst time
P1	3	1
P2	1	4
P3	4	2
P4	0	6
P5	2	3

If the CPU scheduling policy is SJF preemptive, calculate the average waiting time and average turn around time.

Solution-

Gantt Chart-

Gantt Chart

- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

Process Id	Exit time	Turn Around time	Waiting time
P1	4	4 – 3 = 1	1 – 1 = 0
P2	6	6 – 1 = 5	5 – 4 = 1
P3	8	8 – 4 = 4	4 – 2 = 2
P4	16	16 – 0 = 16	16 – 6 = 10
P5	11	11 – 2 = 9	9 – 3 = 6

Now,

- Average Turn Around time = (1 + 5 + 4 + 16 + 9) / 5 = 35 / 5 = 7 unit
- Average waiting time = (0 + 1 + 2 + 10 + 6) / 5 = 19 / 5 = 3.8 unit

---

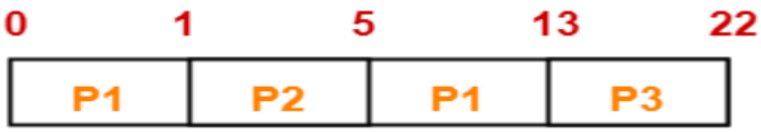
Consider the set of 3 processes whose arrival time and burst time are given below-

Process Id	Arrival time	Burst time
P1	0	9
P2	1	4
P3	2	9

If the CPU scheduling policy is SRTF, calculate the average waiting time and average turn around time.

Solution-

Gantt Chart-



Gantt Chart

Now, we know-

- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

Process Id	Exit time	Turn Around time	Waiting time
P1	13	13 – 0 = 13	13 – 9 = 4
P2	5	5 – 1 = 4	4 – 4 = 0
P3	22	22- 2 = 20	20 – 9 = 11

- Average Turn Around time = (13 + 4 + 20) / 3 = 37 / 3 = 12.33 unit
- Average waiting time = (4 + 0 + 11) / 3 = 15 / 3 = 5 unit

Process Id	Exit time	Turn Around time	Waiting time
P1	13	13 – 0 = 13	13 – 5 = 8
P2	12	12 – 1 = 11	11 – 3 = 8
P3	5	5 – 2 = 3	3 – 1 = 2
P4	9	9 – 3 = 6	6 – 2 = 4
P5	14	14 – 4 = 10	10 – 3 = 7

Now,

- Average Turn Around time = (13 + 11 + 3 + 6 + 10) / 5 = 43 / 5 = 8.6 unit
- Average waiting time = (8 + 8 + 2 + 4 + 7) / 5 = 29 / 5 = 5.8 unit