

ASSESSMENT - 2

NAME: ALOK SINHA

REG. NO.: 17BCE2380

SLOT: L7 + L8

Q)

Execute the following scheduling algorithms using C:

1. FCFS
2. SJF
3. Priority
4. Round Robin
5. Pre-emptive Priority
6. SRTF

CODE

```
#include <stdio.h>
```

```
#include<stdio.h>
```

```
struct priority_preemptive
```

```
{
```

```
    char pn;
```

```
    int at, bt, ct, wt, tat, priority;
```

```
    int s;
```

```
}pq[10];
```

```
int limit;
```

```
void at_Sorting()
```

```
{
```

```

struct priority_preemptive temp;

int i, j;

for(i = 0; i < limit - 1; i++)
{
    for(j = i + 1; j < limit; j++)
    {
        if(pq[i].at > pq[j].at)
        {
            temp = pq[i];
            pq[i] = pq[j];
            pq[j] = temp;
        }
    }
}

int main()
{
    char c; int x;

    do{

        printf("1.FCFS\n2.PRIORITY\n3.SJF\n4.ROUND ROBIN\n5.PRE-EMPTIVE PRIORITY\n6.SRTF\n");
        scanf("%d",&x);

        switch(x)
        {
            case 1:
                {
                    int at[10],bt[10],gnc[10],wt[10],tat[10];

                    float awt=0;

                    float atat=0;

                    int n,i,j;

```

```

for(i=0;i<10;i++)
{
    at[i]=0;
    bt[i]=0;
    gnc[i]=0;
    tat[i]=0;
    wt[i]=0;
}

printf("enter the number of process that you want to proceed \n");
scanf("%d",&n);

printf("enter all the process \n");
int p[10];
for(i=0;i<n;i++)
scanf("%d",&p[i]);

printf("enter all the arrival time \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&at[i]);
    }

printf("enter all the burst time \n");
for(i=0;i<n;i++)
{
    scanf("%d",&bt[i]);
}

gnc[0]=0;
for(i=0;i<10;i++)
gnc[i+1]=gnc[i]+bt[i];
for(i=0;i<n;i++)
{
    wt[i]=gnc[i]-at[i];
    tat[i]=gnc[i+1]-at[i];
}

```

```

        awt=awt+wt[i];
        atat=atat+tat[i];
    }
    awt =awt/n;
    atat=atat/n;
    printf("\n\tprocess\tarrival time\tburst time\twaiting time\tturn around time\n");
    for(i=0;i<n;i++)
    {
        printf("\tp%d\t\t%d\t\t%d\t\t%d\t\t%d\n",i,at[i],bt[i],wt[i],tat[i]);
    }

    printf("the average waiting time is %f\n",awt);
    printf("the average turn around time is %f\n",atat);
    break;
    }
case 2:
    {
        int bt[20],prty[20],wt[20],tat[20];
        int n,i,j,pos,temp;
        float awt,atat,total=0;
        printf("Enter Total Number of Process:");
        scanf("%d",&n);
        printf("enter all the process \n");
        int p[10];
        for(i=0;i<n;i++)
        scanf("%d",&p[i]);

        printf("enter all the burst time \n");
        for(i=0;i<n;i++)
        {
            scanf("%d",&bt[i]);
        }
    }

```

```

    printf("enter all the priority time \n");
for(i=0;i<n;i++)
{
    scanf("%d",&prty[i]);
}
for(i=0;i<n;i++)
{
    pos=i;
    for(j=i+1;j<n;j++)
    {
        if(prty[j]<prty[pos])
            pos=j;
    }

    temp=prty[i];
    prty[i]=prty[pos];
    prty[pos]=temp;

    temp=bt[i];
    bt[i]=bt[pos];
    bt[pos]=temp;

    temp=p[i];
    p[i]=p[pos];
    p[pos]=temp;
}

wt[0]=0;
    for(i=1;i<n;i++)
{
    wt[i]=0;

```

```

        for(j=0;j<i;j++)
            wt[i]+=bt[j];

        total+=wt[i];
    }

    awt=total/n;

    total=0;
    printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("\nP[%d]\t\t %d\t\t %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
    }
    atat=total/n;
    printf("\n\nthe average waiting time=%f",awt);
    printf("\naverage turnaround time=%f\n",atat);
    break;
    }
case 3:{
    int bt[20],wt[20],tat[20],i,j,n,pos,temp;
    float awt,atat,total=0;
    printf("Enter number of process:");
    scanf("%d",&n);
    printf("enter all the process \n");
    int p[10];
    for(i=0;i<n;i++)
        scanf("%d",&p[i]);

```

```
printf("\nEnter Burst Time:\n");
```

```
for(i=0;i<n;i++)
```

```
{
```

```
    scanf("%d",&bt[i]);
```

```
}
```

```
for(i=0;i<n;i++)
```

```
{
```

```
    pos=i;
```

```
    for(j=i+1;j<n;j++)
```

```
    {
```

```
        if(bt[j]<bt[pos])
```

```
            pos=j;
```

```
    }
```

```
    temp=bt[i];
```

```
    bt[i]=bt[pos];
```

```
    bt[pos]=temp;
```

```
    temp=p[i];
```

```
    p[i]=p[pos];
```

```
    p[pos]=temp;
```

```
}
```

```
wt[0]=0;
```

```
for(i=1;i<n;i++)
```

```
{
```

```
    wt[i]=0;
```

```

        for(j=0;j<i;j++)
            wt[i]+=bt[j];

        total+=wt[i];
    }

    awt=total/n;
    total=0;

    printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("\np[%d]\t\t %d\t\t %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
    }

    atat=total/n;
    printf("\n\nAverage Waiting Time=%f",awt);
    printf("\nAverage Turnaround Time=%f\n",atat);
    break;
}

case 4:
    {

int i,j,n,time,remain,d=0,qt;
int wt=0,tat=0,at[10],bt[10],t[10];
printf("Enter Total Process:\t ");
scanf("%d",&n);
remain=n;
for(i=0;i<n;i++)

```



```

{
printf("Enter Arrival Time and Burst Time for Process Process Number %d :",i+1);
scanf("%d",&at[i]);
scanf("%d",&bt[i]);
t[i]=bt[i];
}
printf("Enter Time Quantum:\t");
scanf("%d",&qt);
printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
for(time=0,i=0;remain!=0;)
{
if(t[i]<=qt && t[i]>0)
{
time+=t[i];
t[i]=0;
d=1;
}
else if(t[i]>0)
{
t[i]-=qt;
time+=qt;
}
if(t[i]==0 && d==1)
{
remain--;
printf("P[%d]\t|\t%d\t|\t%d\n",i+1,time-at[i],time-at[i]-bt[i]);
wt+=time-at[i]-bt[i];
tat+=time-at[i];
d=0;
}
if(i==n-1)

```

```

i=0;
else if(at[i+1]<=time)
i++;
else
i=0;
}
printf("\nAverage Waiting Time= %f\n",wt*1.0/n);
printf("Avg Turnaround Time = %f",tat*1.0/n);
break;
    }
    case 5:

{

        int i, time = 0, bt = 0, largest;
        char c;
        float wait_time = 0, tat = 0, average_wt, average_tat;
        printf("\nEnter Total Number of Processes:\t");
        scanf("%d", &limit);
        for(i = 0, c = 'A'; i < limit; i++, c++)
        {
            pq[i].pn = c;
            printf("\nEnter Details For Process[%C]:\n", pq[i].pn);
            printf("Enter Arrival Time:\t");
            scanf("%d", &pq[i].at );
            printf("Enter Burst Time:\t");
            scanf("%d", &pq[i].bt);
            printf("Enter Priority:\t");
            scanf("%d", &pq[i].priority);
            pq[i].s = 0;
            bt = bt + pq[i].bt;

```

```

}

at_Sorting();

pq[9].priority = -9999;

printf("\nProcess Name\tArrival Time\tBurst Time\tPriority\tWaiting Time");

for(time = pq[0].at; time < bt;)
{
    largest = 9;
    for(i = 0; i < limit; i++)
    {
        if(pq[i].at <= time && pq[i].s != 1 && pq[i].priority > pq[largest].priority)
        {
            largest = i;
        }
    }
    time = time + pq[largest].bt;
    pq[largest].ct = time;
    pq[largest].wt = pq[largest].ct - pq[largest].at - pq[largest].bt;
    pq[largest].tat = pq[largest].ct - pq[largest].at;
    pq[largest].s = 1;
    wait_time = wait_time + pq[largest].wt;
    tat = tat + pq[largest].tat;

    printf("\n%c\t\t%d\t\t%d\t\t%d\t\t%d", pq[largest].pn, pq[largest].at, pq[largest].bt,
pq[largest].priority, pq[largest].wt);

}

average_wt = wait_time / limit;
average_tat = tat / limit;

printf("\n\nAverage waiting time:\t%f\n", average_wt);
printf("Average Turnaround Time:\t%f\n", average_tat);

break;}

```

case 6:

```
{
    int at[10], bt[10], temp[10];
    int i, smallest, c = 0, time, n;
    double wt = 0, tat = 0, end;
    float awt, atat;
    printf("\nEnter the Total Number of Processes:\t");
    scanf("%d", &n);
    printf("\nEnter Details of %d Processes\n", n);
    for(i = 0; i < n; i++)
    {
        printf("\nEnter Arrival Time:\t");
        scanf("%d", &at[i]);
        printf("Enter Burst Time:\t");
        scanf("%d", &bt[i]);
        temp[i] = bt[i];
    }
    bt[9] = 9999;
    for(time = 0; c != n; time++)
    {
        smallest = 9;
        for(i = 0; i < n; i++)
        {
            if(at[i] <= time && bt[i] < bt[smallest] && bt[i] > 0)
            {
                smallest = i;
            }
        }
        bt[smallest]--;
        if(bt[smallest] == 0)
```

```

        {
            c++;

            end = time + 1;

            wt = wt + end - at[smallest] - temp[smallest];

            tat = tat + end - at[smallest];

        }
    }

    awt = wt / n;
    atat = tat / n;

    printf("\n\nAverage Waiting Time:\t%f\n", awt);
    printf("Average Turnaround Time:\t%f\n", atat);
break;
    }

    {
default:
        printf("invalid choice");

        break;

    }

    }

    printf("to continue press 'Y' or 'y' else any letter to exit\n");
    scanf("%c",&c);

    }

    while(c=='Y' || c=='y');
}

```

1.FCFS

```
1.FCFS
2.PRIORITY
3.SJF
4.ROUND ROBIN
5.PRE-EMPTIVE PRIORITY
6.SRTF
1
enter the number of process that you want to proceed
5
enter all the process
1
2
3
4
5
enter all the arrival time
0
1
2
3
4
enter all the burst time
4
3
1
2
5

    process arrival time    burst time    waiting time    turn around time
    p0           0           4           0           4
    p1           1           3           3           6
    p2           2           1           5           6
    p3           3           2           5           7
    p4           4           5           6          11

the average waiting time is 3.800000
the average turn around time is 6.800000
to continue press 'Y' or 'y' else any letter to exit
```

```
root@ALOK-SINHA:~# vi exr.c
root@ALOK-SINHA:~# gcc exr.c -o f1
root@ALOK-SINHA:~# ./f1
enter the number of process that you want to proceed
5
enter all the process
1
2
3
4
5
enter all the arrival time
0
1
2
3
4
```

root@ALOK-SINHA: ~

```
enter all the arrival time
```

```
0
1
2
3
4
enter all the brust time
4
3
1
2
5
```

```
enter all the brust time
```

	process	arrival time	brust time	waiting time	turn arround time
	p0	0	4	0	4
	p1	1	3	3	6
	p2	2	1	5	6
	p3	3	2	5	7
	p4	4	5	6	11

```
the average waiting time is 3.800000
```

```
the average turn around time is 6.800000
```

2.PRIORITY

```
1.FCFS
2.PRIORITY
3.SJF
4.ROUND ROBIN
5.PRE-EMPTIVE PRIORITY
6.SRTF
2
Enter Total Number of Process:5
enter all the process
1
2
3
4
5
enter all the burst time
10
1
2
1
5
enter all the priority time
3
1
4
5
2

Process      Burst Time      Waiting Time      Turnaround Time
P[2]          1                0                 1
P[5]          5                1                 6
P[1]         10                6                16
P[3]          2                16                18
P[4]          1                18                19

the average waiting time=8.200000
average turnaround time=12.000000
to continue press 'Y' or 'y' else any letter to exit
```

```
root@ALOK-SINHA:~# gcc exr1.c -o f2
root@ALOK-SINHA:~# ./f2
Enter Total Number of Process:5
enter all the process
1
2
3
4
5
enter all the burst time
10
1
2
1
5
enter all the priority time
3
1
4
5
2

Process      Burst Time      Waiting Time      Turnaround Time
P[2]          1                0                 1
P[5]          5                1                 6
P[1]         10                6                16
P[3]          2                16                18
P[4]          1                18                19

the average waiting time=8.200000
average turnaround time=12.000000
root@ALOK-SINHA:~#
```


3.SJF

```
1.FCFS
2.PRIORITY
3.SJF
4.ROUND ROBIN
5.PRE-EMPTIVE PRIORITY
6.SRTF
3
Enter number of process:5
enter all the process
1
2
3
4
5

Enter Burst Time:
7
5
1
2
8

Process      Burst Time      Waiting Time      Turnaround Time
p[3]          1                0                 1
p[4]          2                1                 3
p[2]          5                3                 8
p[1]          7                8                15
p[5]          8                15                23

Average Waiting Time=5.400000
Average Turnaround Time=10.000000
to continue press 'Y' or 'y' else any letter to exit
```

```
root@ALOK-SINHA:~# vi sjf.c
root@ALOK-SINHA:~# gcc sjf.c -o sjff
root@ALOK-SINHA:~# ./sjff
Enter number of process:5
enter all the process
1
2
3
4
5

Enter Burst Time:
7
5
1
2
8

Process      Burst Time      Waiting Time      Turnaround Time
p[3]          1                0                 1
p[4]          2                1                 3
p[2]          5                3                 8
p[1]          7                8                15
p[5]          8                15                23

Average Waiting Time=5.400000
Average Turnaround Time=10.000000
root@ALOK-SINHA:~#
```

4 ROUND ROBIN

```
1.FCFS
2.PRIORITY
3.SJF
4.ROUND ROBIN
5.PRE-EMPTIVE PRIORITY
6.SRTF
4
Enter Total Process:      4
Enter Arrival Time and Burst Time for Process Process Number 1 :0
9
Enter Arrival Time and Burst Time for Process Process Number 2 :1
5
Enter Arrival Time and Burst Time for Process Process Number 3 :2
3
Enter Arrival Time and Burst Time for Process Process Number 4 :3
4
Enter Time Quantum:      5

Process |Turnaround Time|Waiting Time
P[2]    |          9    |          4
P[3]    |         11     |          8
P[4]    |         14     |         10
P[1]    |         21     |         12

Average Waiting Time= 8.500000
Avg Turnaround Time = 13.750000to continue press 'Y' or 'y' else any letter to exit
```

```
root@ALOK-SINHA:~# vi rr.c
root@ALOK-SINHA:~# gcc rr.c -o rrb
root@ALOK-SINHA:~# ./rrb
Enter Total Process:      4
Enter Arrival Time and Burst Time for Process Process Number 1 :0
9
Enter Arrival Time and Burst Time for Process Process Number 2 :1
5
Enter Arrival Time and Burst Time for Process Process Number 3 :2
3
Enter Arrival Time and Burst Time for Process Process Number 4 :3
4
Enter Time Quantum:      5

Process |Turnaround Time|Waiting Time
P[2]    |          9    |          4
P[3]    |         11     |          8
P[4]    |         14     |         10
P[1]    |         21     |         12

Average Waiting Time= 8.500000
root@ALOK-SINHA:~#
```

5. PRE-EMPTIVE PRIORITY

```
1.FCFS
2.PRIORITY
3.SJF
4.ROUND ROBIN
5.PRE-EMPTIVE PRIORITY
6.SRTF
5

Enter Total Number of Processes:      4

Enter Details For Process[A]:
Enter Arrival Time:      0
Enter Burst Time:      10
Enter Priority: 2

Enter Details For Process[B]:
Enter Arrival Time:      5
Enter Burst Time:      20
Enter Priority: 3


Enter Details For Process[C]:
Enter Arrival Time:      2
Enter Burst Time:      5
Enter Priority: 1

Enter Details For Process[D]:
Enter Arrival Time:      3
Enter Burst Time:      2
Enter Priority: 0

Process Name   Arrival Time   Burst Time   Priority   Waiting Time
A              0              10           2           0
B              5              20           3           5
C              2              5            1          28
D              3              2            0          32

Average waiting time:  16.250000
Average Turnaround Time:  25.500000
to continue press 'Y' or 'y' else any letter to exit
```

6 SRTF

 "F:\c programs\scheduling.exe"

```
1.FCFS
2.PRIORITY
3.SJF
4.ROUND ROBIN
5.PRE-EMPTIVE PRIORITY
6.SRTF
6

Enter the Total Number of Processes:    6

Enter Details of 6 Processes

Enter Arrival Time:    0
Enter Burst Time:      7

Enter Arrival Time:    1
Enter Burst Time:      5

Enter Arrival Time:    2
Enter Burst Time:      3

Enter Arrival Time:    3
Enter Burst Time:      1

Enter Arrival Time:    4
Enter Burst Time:      2

Enter Arrival Time:    5
Enter Burst Time:      1

Average Waiting Time:  4.000000
Average Turnaround Time:      7.166667
to continue press 'Y' or 'y' else any letter to exit
```

root@ALOK-SINHA: ~

```
root@ALOK-SINHA:~# vi sr.c
root@ALOK-SINHA:~# vi srtf1.c
root@ALOK-SINHA:~# gcc srtf1.c -o srtf
root@ALOK-SINHA:~# ./srtf
```

Enter the Total Number of Processes: 6

Enter Details of 6 Processes

Enter Arrival Time: 0

Enter Burst Time: 7

Enter Arrival Time: 1

Enter Burst Time: 5

Enter Arrival Time: 2

Enter Burst Time: 3

Enter Arrival Time: 3

Enter Burst Time: 1

Enter Arrival Time: 4

Enter Burst Time: 2

Enter Arrival Time: 5

Enter Burst Time: 1

Average Waiting Time: 4.000000

Average Turnaround Time: 7.166667

root@ALOK-SINHA:~#