

Atma Ram Sanatan Dharma College
(University Of Delhi)

Name	Neeraj
Roll no	18088
Course	B.Sc (H) Computer science
<u>Submitted To</u>	<u>Dr. Parul Jain</u>

Q 6. Write a program to implement FCFS scheduling algorithm?

Answer

```
#include <stdio.h>
int tim = 0;

int main()
{
    int n,b[20], i, j, w[20], tw = 0, taround[20], tt = 0;
    float avw, avt;
    printf("\nEnter the number of process: ");
    scanf("%d",&n);

    for(i = 1; i<=n; i++)
    {
        printf("Enter the Burst Time of process %d:" ,i);
        scanf("%d", &b[i]);
    }

    for(i=1; j<=n; i++)
    {
        w[i] = tim;
        for(j=1; j<=b[i]; j++)
        {
```

```

        tim++;
        if(j==b[i])
        {
            taround[i] = tim;
        }
    }
}

```

```

for(i=1; i<=n; i++)
{
    tw = tw + w[i];
}

```

```

avw = (float)tw/n;

```

```

for(i=1; i<=n; i++)
{
    tt = tt + taround[i];
}

```

```

avt = (float)tt/n;

```

```

printf("\nWaiting times and turn around times of the processes");

```

```

printf("\n
n*****
*****
");

```

```

for(i=1; i<=n; i++)
{
    printf("\nProcess: %d waiting time: %d", i, w[i]);
}

```

```
    printf("\nProcess: %d trun around time: %d", i, taround[i]);
}
```

```
printf("\n\nAverage waiting time: %f", avw);
printf("\n\nAverage trun around time: %f\n", avt);
```

```
return 0;
```

```
}
```

Output:-

```
gautam@gautam:~/Desktop/os$ gcc -o a program7.c
gautam@gautam:~/Desktop/os$ ./a

Enter the number of process: 3
Enter the Burst Time of process 1:3
Enter the Burst Time of process 2:2
Enter the Burst Time of process 3:1

Wating times and trun around times of the processes
*****
Process: 1 wating time: 0
Process: 1 trun around time: 3
Process: 2 wating time: 0
Process: 2 trun around time: 0
Process: 3 wating time: 0
Process: 3 trun around time: 0

Average waiting time: 0.000000
Average trun around time: 1.000000
gautam@gautam:~/Desktop/os$
```

Q 7. Write a program to implement Round Robin scheduling algorithm?

Answer

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

```
int main()
```

```
{
```

```
    int count, j,n, time,remain, flag=0, time_quantum;
    int wait_time=0, turnaround_time = 0, at[10], bt[10], rt[10];
```

```
    printf("Enter total processes: ");
```

```

scanf("%d", &n);

remain = n;

for(count =0; count<n; count++)
{
    printf("\nEnter Arrival time and Burst time for the process
%d: ", count+1);
    scanf("%d",&at[count]);
    scanf("%d",&bt[count]);

    rt[count] = bt[count];
}

printf("\nEnter time quantum: ");
scanf("%d",&time_quantum);

printf("\n\nProcess \t|Turn Around time \t\t|waiting time\n\n");
for(time = 0, count = 0; remain!=0;)
{
    if(rt[count]<= time_quantum && rt[count]>0)
    {
        time += rt[count];
        rt[count] = 0;
        flag = 1;
    }

    else if(rt[count]>0)
    {
        rt[count] -= time_quantum;
        time+=time_quantum;
    }
}

```

```

if(rt[count] == 0 && flag == 1)
{
    remain--;
    printf("\nProcess: %d \t\t %d \t\t\t\t %d", count+1, time-
at[count], time-at[count]-bt[count]);
    wait_time += time-at[count]-bt[count];
    turnaround_time += time-at[count];
    flag = 0;
}

if(count==n-1)
    count = 0;

else if(at[count+1]<=time)
    count++;
else
    count=0;
}

printf("\nAverage waiting time: %f",wait_time*1.0/n);
printf("\nAverage turn around time: %f\
n",turnaround_time*1.0/n);
return 0;
}

```

Output:-

```

gautam@gautam:~/Desktop/os$ gcc -o a program8.c
gautam@gautam:~/Desktop/os$ ./a
Enter total processes: 3

Enter Arrival time and Burst time for the process 1: 0 1
Enter Arrival time and Burst time for the process 2: 1 2
Enter Arrival time and Burst time for the process 3: 2 3
Enter time quantum: 3

Process          |Turn Around time          |waiting time
Process: 1       |          1               |          0
Process: 2       |          2               |          0
Process: 3       |          4               |          1
Average waiting time: 0.333333
Average turn around time: 2.333333
gautam@gautam:~/Desktop/os$ █

```

Q 8. Write a program to implement SJF scheduling algorithm?

Answer

```
#include <stdio.h>

int main()
{
    int bt[20],p[20],wt[20],tat[20],i,j,n,total = 0, pos,temp;
    float avg_wt, avg_tat;

    printf("\nEnter number of process: ");
    scanf("%d",&n);

    printf("\nEnter Burst time: ");
    for(i=0; i<n; i++)
    {
        printf("\nProcess%d: ",i+1);
        scanf("%d",&bt[i]);
        p[i] = i+1;
    }

    for(i=0;i<n;i++)
    {
        pos = i;
        for(j =i+1; j<n;j++)
        {
            if(bt[i]<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];
}
```

```
avg_wt = (float)total/n;
total =0;
printf("\nProcess\tBurst Time\twaiting Time\tTurn Around
Time");
for(i=0; i<n; i++)
{
    tat[i] = bt[i]+wt[i];
    total+=tat[i];
    printf("\nProcess: %d\t\t %d \t\t %d \t\t %d",
p[i],bt[i],wt[i],tat[i]);
}
```

```
avg_tat = (float)total/n;
printf("\n\nAverage Waiting time: %f",avg_wt);
printf("\nAverage Turn around time: %f\n",avg_tat);
return 0;
```

}

Output:-

```
gautam@gautam:~/Desktop/os$ gcc -o a program9.c
gautam@gautam:~/Desktop/os$ ./a

Enter number of process: 3

Enter Burst time:
Process1: 1

Process2: 3

Process3: 2

Process|      Burst Time|      waiting Time|      Turn Around Time
Process: 1      |          1      |          0      |          1
Process: 2      |          3      |          1      |          4
Process: 3      |          2      |          4      |          6

Average Waiting time: 1.666667
Average Turn around time: 3.666667
```