

WEEK 1

Write a C program to simulate the following non-pre-emptive CPU scheduling algorithm to find turnaround time and waiting time.

(a) FCFS

(b) SJF

2.1.2 Code:

```
#include<stdio.h>
int n, i, j, pos, temp, choice, Burst_time[20], Waiting_time[20], Turn_around_time[20],
process[20], total=0;
float avg_Turn_around_time=0, avg_Waiting_time=0;
int FCFS()
{
Waiting_time[0]=0;
for(i=1;i<n;i++)
{
Waiting_time[i]=0;
for(j=0;j<i;j++)
Waiting_time[i]+=Burst_time[j];
}
printf("Process\tBurst Time\tWaiting Time\tTurnaround Time");
for(i=0;i<n;i++)
{
Turn_around_time[i]=Burst_time[i]+Waiting_time[i];
avg_Waiting_time+=Waiting_time[i];
```

4

```
avg_Turn_around_time+=Turn_around_time[i];
printf("P[%d]\t%d\t%d\t%d\t%d\t\t",i+1,Burst_time[i],Waiting_time[i],Turn_around_time
[i]);
}
avg_Waiting_time =(float)(avg_Waiting_time)/(float)i;
avg_Turn_around_time=(float)(avg_Turn_around_time)/(float)i;
printf("Average Waiting Time:%.2f",avg_Waiting_time);
printf("Average Turnaround Time:%.2f",avg_Turn_around_time);
return 0;
}
int SJF()
{
//sorting
for(i=0;i<n;i++)
{
```

```

pos=i;
for(j=i+1;j<n;j++)
{
if(Burst_time[j]<Burst_time[pos])
pos=j;
}
temp=Burst_time[i];
Burst_time[i]=Burst_time[pos];
Burst_time[pos]=temp;
temp=process[i];
process[i]=process[pos];
process[pos]=temp;
}
Waiting_time[0]=0;

for(i=1;i<n;i++)
{
Waiting_time[i]=0;
for(j=0;j<i;j++)
Waiting_time[i]+=Burst_time[j];
total+=Waiting_time[i];
}

4
avg_Waiting_time=(float)total/n;
total=0;
printf("Process\tBurst Time\tWaiting Time\tTurnaround Time");
for(i=0;i<n;i++)
{
Turn_around_time[i]=Burst_time[i]+Waiting_time[i];
total+=Turn_around_time[i];
printf("\nP[%d]\t%d\t%d\t%d\t",process[i],Burst_time[i],Waiting_time[i],Turn_ar
oun
d_time[i]);
}
avg_Turn_around_time=(float)total/n;
printf("\n\nAverage Waiting Time=%f",avg_Waiting_time);
printf("\nAverage Turnaround Time=%f",avg_Turn_around_time);
}
int main()
{
printf("Enter the total number of processes:");
scanf("%d",&n);
printf("\nEnter Burst Time:\n");

```

```
for(i=0;i<n;i++)
{
printf("P[%d]",& i+1);
scanf("%d",& Burst_time[i]);
process[i]=i+1;
}
while(1)
{ printf("\n----MAIN MENU----\n");
printf("1. FCFS Scheduling\n2. SJF Scheduling\n");
printf("\nEnter your choice:");
scanf("%d",& choice);
switch(choice)
{
case 1: FCFS();
break;
case 2: SJF();
break;

default: printf("Invalid Input!!!\n");
}
}
return 0;
}
```

OUTPUT

```
D:\Codes\c\OS_Lab>gcc "FCFS(CPU scheduling).c"
```

```
D:\Codes\c\OS_Lab>.\a.exe
Enter the number of processes: 4
Enter the burst times of 4 processes:
-----
Enter the burst times Process1:4
Enter the burst times Process2:5
Enter the burst times Process3:2
Enter the burst times Process4:7
```

The details of the processes are as below:

Process	Burst Time	Turn Around Time	Waiting Time
1	4.000000	4.000000	0.000000
2	5.000000	9.000000	4.000000
3	2.000000	11.000000	9.000000
4	7.000000	18.000000	11.000000

The average waiting time is: 6.000000

The average turn around time is: 10.500000

```
D:\Codes\c\OS_Lab>gcc "SJF(Non-Premptive).c"
```

```
D:\Codes\c\OS_Lab>.\a.exe
Enter the number of processes: 4
Enter the burst times of 4 processes:
-----
Enter the burst times Process1:4
Enter the burst times Process2:5
Enter the burst times Process3:2
Enter the burst times Process4:7
```

The details of the processes are as below:

Process	Burst Time	Turn Around Time	Waiting Time
1	2.000000	2.000000	0.000000
2	4.000000	6.000000	2.000000
3	5.000000	11.000000	6.000000
4	7.000000	18.000000	11.000000

The average waiting time is: 4.750000

The average turn around time is: 9.250000

```
D:\Codes\c\OS_Lab>gcc "SJF(premptive).c"

D:\Codes\c\OS_Lab>.\a.exe
Enter the number of processes: 4
Enter the burst times of 4 processes:
-----
Enter the burst times and Arrival time Process1:4 0
Enter the burst times and Arrival time Process2:3 0
Enter the burst times and Arrival time Process3:5 1
Enter the burst times and Arrival time Process4:6 2
Processes      Burst time      Waiting time     Turn around time
 1              4                  3                  7
 2              3                  0                  3
 3              5                  6                  11
 4              6                 10                 16
Average waiting time = 4.750000
Average turn around time = 9.250000
```

Lab 02

To write a C program to simulate the CPU scheduling algorithm, first come first serve (FCFS) & also for shortest job first

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int at[20], bt[20], tat[20], i, n;
    float waitavg, fwtavg; int sum = 0
    printf("Enter no of process -- ");
    scanf("%d", &n);
    for (i=0; i<n; i++)
    {
        printf("Enter AT & CPU time\n");
        scanf("%d", &at[i]);
        scanf("%d", &bt[i]);
        tat[i] = (sum + bt[i]) - at[i];
        sum += bt[i];
        wft[i] = tat[i] - bt[i];
    }
    for (i=0; i<n; i++)
    {
        printf("Turn around time for process no %d : %d\n", i+1, tat[i]);
        fwtavg += tat[i];
    }
    printf("Average turn around time = %f\n", fwtavg/n);
    printf("Average waiting time = %f\n", waitavg/n);
}
```

case 2 : sjfb(n),
break;

case 3 : s2tf(n)
break;

{

Void sjfc(int n)

{ int b[20], tat[20], wt[20];

float aout[20]; atat = 0, sum-burst = this

int sum = 0, i, j, smallest;

for (i=0; i < n; i++) & Cput[i] = Cput(i)

Sum burst time += Cput[i]; }

Print ("At process (+ waiting time + turnaround time) ",

Cput[9] = 99.99);

while (sum < sum - burst - time) {

smallest = 9;

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

{

Sum - burst - time += Cput[i];

}

Cput[9] = 9999;

while (sum < sum - burst - time)

{ smallest = 9;

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

{ if (at[i] <= sum + Cput[i])

put[i] < Cput[smallest])

```
    cout << n, i++ )  
    {  
        cout (" Prod %d %d %d %d " )  
        cout (i), cur (i), w1 (i) + a (i))  
    }
```

```
    cout (" Average existing items %f of n ", awt ),  
    cout (" Average turnaround time lost at cat ),  
    getch );  
}
```

```
void Setf (const)  
int remaining (t) = 0,  
int here = 0,  
int comb = 0,  
for (int i = 0; i < n; i++)
```

```
{ remaining (i) = comb (i);
```

```
while (temp != n)
```

```
{
```

```
    int shortest = -1
```

```
    int min = 1000;
```

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

```
{ if (at (i) <= awt )
```

```
    remaining (v) > 0
```

```
    remaining (v) < min )
```

```
{ shortest = i,
```

(Continue),
 Run +
 if (remaining [shortes] == 0)
 if [remaining [shortes]] == 0]
 { atat [shortes] = arr - at [shortes])
 cur [shortes] = atat [shortest]

Output

Enter the number of processes
 Enter arrival time & CPU time for each process respectively

0	3
1	6
4	4
8	2

Menu

- 1) FCFS
- 2) SJF (non preemptive)
- 3) SRTF (preemptive)
- 4) Exit

Process	AT	CPU Time	Waiting time	Turnaround time
P ₀	0	3	0	3
P ₁	1	4	2	8
P ₂	4	6	7	9

Process	AT	CPU TIME	WWT	WT
0	0	3	0	3
1	1	6	8	14
2	4	4	0	4
3	6	2	2	4

Avg WWT -- 2.5
 Avg TT -- 6.25

FCFS

P ₀	P ₁	P ₂	P ₁	P ₁	P ₂	P ₃
0	1	3	4	6	9	13 15
P ₀₍₂₎	P ₀₍₂₎		P ₁₍₅₎		P ₂₍₄₎ P ₃₍₂₎	
			P ₂₍₄₎		P ₃₍₂₎	
			P ₁₍₃₎			
			P ₂₍₄₎			
			P ₃₍₂₎			

2) SJF

P ₀	P ₀	P ₁	P ₁	P ₁	P ₃	P ₂
0	1	3	4	6	9	13 15
P ₀₍₃₎	P ₀₍₆₎		P ₁₍₃₎	P ₂₍₄₎ P ₃₍₄₎		
			P ₃₍₄₎	P ₃₍₂₎		
			P ₃₍₂₎			

3) SRTI

P ₀	P ₁	P ₂	P ₃	P ₁

