## Week - 2 (15 June 2023) **Experiment - 2**

## **Ouestion:**

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 (Non-pre-emptive)

## **Program:**

```
#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[i];
  printf("\nProcess\t\tBurst Time\t\tWaiting Time\t\tTurnaround Time");
  for(i=0;i< n;i++)
    Turn around time[i]=Burst time[i]+Waiting time[i];
    avg Waiting time+=Waiting time[i];
    avg Turn around time+=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("\nAverage Waiting Time:%.2f",avg Waiting time);
  printf("\nAverage Turnaround Time:%.2f\n",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];
  avg Waiting time=(float)total/n;
  total=0;
  printf("\nProcess\t\tBurst Time\t\tWaiting Time\t\tTurnaround Time");
  for(i=0;i< n;i++)
    Turn around time[i]=Burst time[i]+Waiting time[i];
    total+=Turn around time[i];
    avg Turn around time=(float)total/n;
  printf("\n\nAverage Waiting Time=%f",avg Waiting time);
  printf("\nAverage Turnaround Time=%f\n",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----\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!!!");
  return 0;
Output:
Enter the total number of processes:3
Enter Burst Time:
P[1]:5
P[2]:12
P[3]:19
----MAIN MENU----
1. FCFS Scheduling
2. SJF Scheduling
Enter your choice:1
Process
                 Burst Time
                                            Waiting Time
                                                                       Turnaround Time
P[1]
                 5
                                                                                5
P[2]
                 12
                                            5
                                                                                17
P[3]
                 19
                                            17
                                                                                36
Average Waiting Time:7.33
Average Turnaround Time:19.33
----MAIN MENU----
1. FCFS Scheduling
2. SJF Scheduling
Enter your choice:2
                                            Waiting Time
Process
                 Burst Time
                                                                       Turnaround Time
P[1]
                                            0
                                                                                5
```

5

17

17

36

P[2]

P[3]

12

19

Average Waiting Time=7.333333 Average Turnaround Time=19.333334

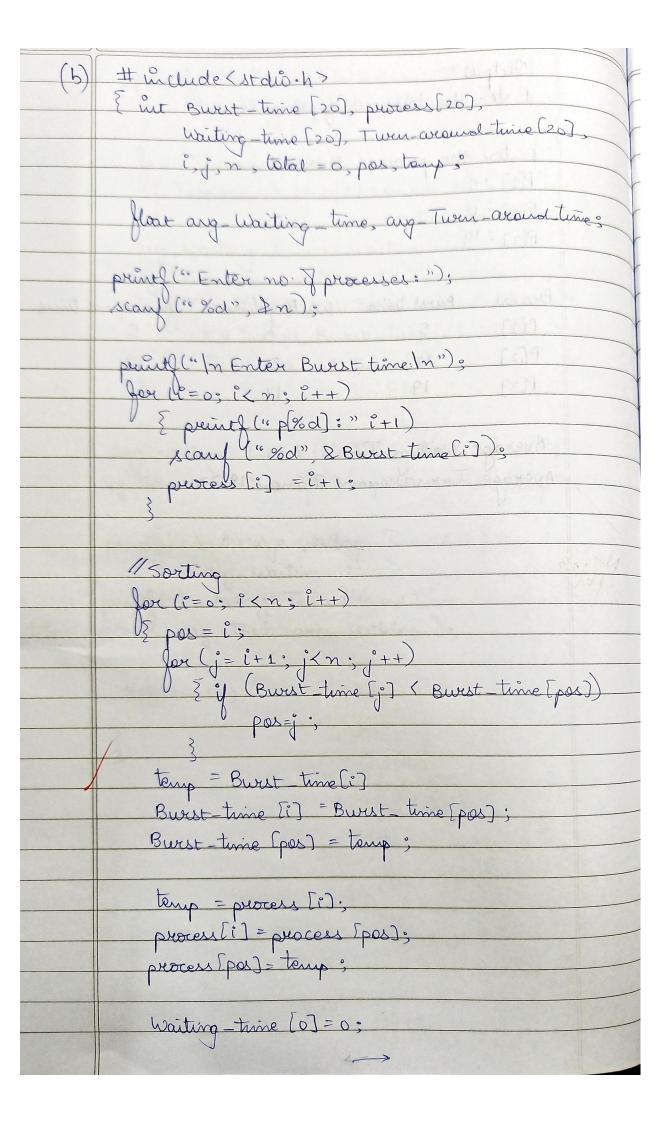
Enton the total	number of processes.2		
Enter the total	number of processes:3		
Enter Burst Time	a•		
P[1]:19	•		
P[2]:5			
P[3]:12			
MAIN MENU-			
1. FCFS Schedul:	ing		
2. SJF Scheduli	ng		
45			
Enter your choice	ce:1		
Process			Turnaround Time
P[1]		0	19
P[2]		19	24
P[3]		24	36
Average Waiting			
Average Turnaro	und Time:26.33		
MAIN MENU-			
1. FCFS Schedul:			
2. SJF Scheduli			
Z. Bor Bonoddir	9		
Enter your choice	ce:2		
-			
Process	Burst Time	Waiting Time	Turnaround Time
P[2]	5	0	5
P[3]	12	5	17
P[1]	19	17	36
Average Waiting	Time=7.333333		

Average Turnaround Time=19.333334

## **Observation Book Pictures:** PAGE NO: DATE: 15/06/2023 Experiment -2 preogram to simulate the following # include < stdio. h > 1 +1 Eint i, j, n, Burst-time [10], Waiting time [10], Turn around time 10): float any Turn-areamd - time =0, ang waiting time = 0; print f (" Enter the total no- of processes: "); print (" Enter the Process Burst Time: In"); or Uli=0; Kn; i++) Scary ( 6 % d", & Burst time [i] waiting time [o] = 0; or ( =1 ; ( n ; i++) Waiting time (i) += Burst-time [i]:

	Drivet (" by Process It It Ruget Time It It		
	premer in reduces to burst time ")		
	print ("In Process It It Burst Time It It  waiting time It It Turn around time");		
Las K	Jan (i = 0; i < n; i++)		
	{ Turnaround_time [i] = Bwest_time [i] +		
	waiting time [i];		
	ang waiting time += waiting time [i];		
	arg Turn-around-time += Turn-around-time[i]:		
	print [" In P[%d] It It %d It It %d It It It		
	%d", i+1, Burst-time [i], waiting-time[i],		
	Turn-around-time [i]);		
Tool	mitima for 7 smit-terus, r. ; 9 +813		
	Total Cold and Indiana west		
	ang waiting time = (float) (ang waiting time) (float) (i); ang Turn-around time = (float) (ang Turn-around time) (float) (i);		
.(1	print (66 \nAverage Waiting Time: %.21		
	print (" \n Average Waiting Time: %.2 f,  ang waiting time);  print (" \n Average Turn Around Time: %.2 f,		
	preint (" \n Average Turn Around Time: %.21		
	arg Turn around time);		
	(++9 on >9 -0=910 mg)		
	return 0; (15 " 16 18 19 ") Hang 3		
	Conflict times "1/20 W Munit times (1) .		
	TO TO TOTAL MINE		

	Ton (1/2)	PAGE NO : DATE :	
	Output:	S. H.	(1)
	Enter the total no of processes: 3	103	(6)
	Dant musica west factoristics		
	Enter the process Burst time:	wat ry	Harry L
	P[1]:5	10.5	
	P(2): 12	loos	
	P[3]: 19	0	
	(Company Francisco)	O-1 SHO	
	Process Burst time waiting time Tu	rnarou	od time
	Process Burst time waiting time Tu P[1] 5	9 5	- Wille
	P(2) ("12/2mil 7/28/19 5/10)		A Paris
	P[3] 19 (++9 17 5°		
	(1+9 ": [b.28] ") Driver		
	Average waiting Time: 7.33		4
	Average Turnareund Time: 19.33.		An.
		7 3	
	The state of the s		
1/12	3	11/201	1 2.49
22/6/	(++9 · x > 1 · 2 · 2	1 40	I no
		1	



for (i=1, i(n, i++) E waiting time (i) = 0 is for (;= 0; ; (;; ; ++) waiting time [i] + = Bwest-time [j]; lotal += waiting time [i] is ang waiting time = (float) total / n; printf (" In Process It Buret Time It waiting time It Turn around time "); er (i=0, i<n, i++) Turnarand time [i] = Buret time [i] waiting time (i) total += Turn-around-time [i]; printf (66 \n P[%d] \t\t %d \t\t %d\t\t\t %d") process [i], Burst-time[i], waiting time[i], Turn-around time [0]); printf ("/n/n Average waiting Time = %)

ang-waiting-time); preint f (" \n Average Turnaround time = % of \n",
ang - Turn - around - time);

	Output:	
	Enter the no of processes: 3	
	iga and the of process	
-1-32	Enter Burst Time:	
Colorador Stan	P(1): 19 and milion + late	
	P(2): 5	
	P[3]: 12	
	en I intat ( +mill) = anit miliar ou	
	Process Burest Time Waiting Time Turna	
	P(2) 5	5
annit	P(3) 1/am T2+1x1/3 1/110 5x9 r/4) (+1)	17
	P[1] : (19 and because 17 )	36
	(++9 0 × 29 -0 =9) real	
4		
- (7	Average Waiting Time: 7.333: Average Turn around Time: 19.333	*
	19 grant la manage ( 82 2 - + 0 det	
" )~ X	4/4/4/16 4/4/16 4/4/5/6/4/4 900/16/4 90/ 16/4/4/E	
( and a	property puret to all waiting	