

LAB 2

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

- FCFS
- SJF

FCFS Scheduling:

Code:

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

```
void waitingtime(int proc[],int n,int burst_time[],int wait_time[]){
    wait_time[0]=0;
    for(int i=1;i<n;i++){
        wait_time[i] = burst_time[i-1] + wait_time[i-1];
    }
}

void turnaroundtime(int proc[],int n,int burst_time[],int wait_time[],int tat[]){
    for(int i=0;i<n;i++){
        tat[i] = burst_time[i] + wait_time[i];
    }
}

int avgtime(int proc[], int n, int burst_time[]) {
    int wait_time[n], tat[n], total_wt = 0, total_tat = 0;

    waitingtime(proc, n, burst_time, wait_time);
    turnaroundtime(proc, n, burst_time, wait_time, tat);

    printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");

    for (int i = 0; i < n; i++) {
        total_wt = total_wt + wait_time[i];
        total_tat = total_tat + tat[i];
        printf("%d\t\t%d\t\t%d\t\t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
    }

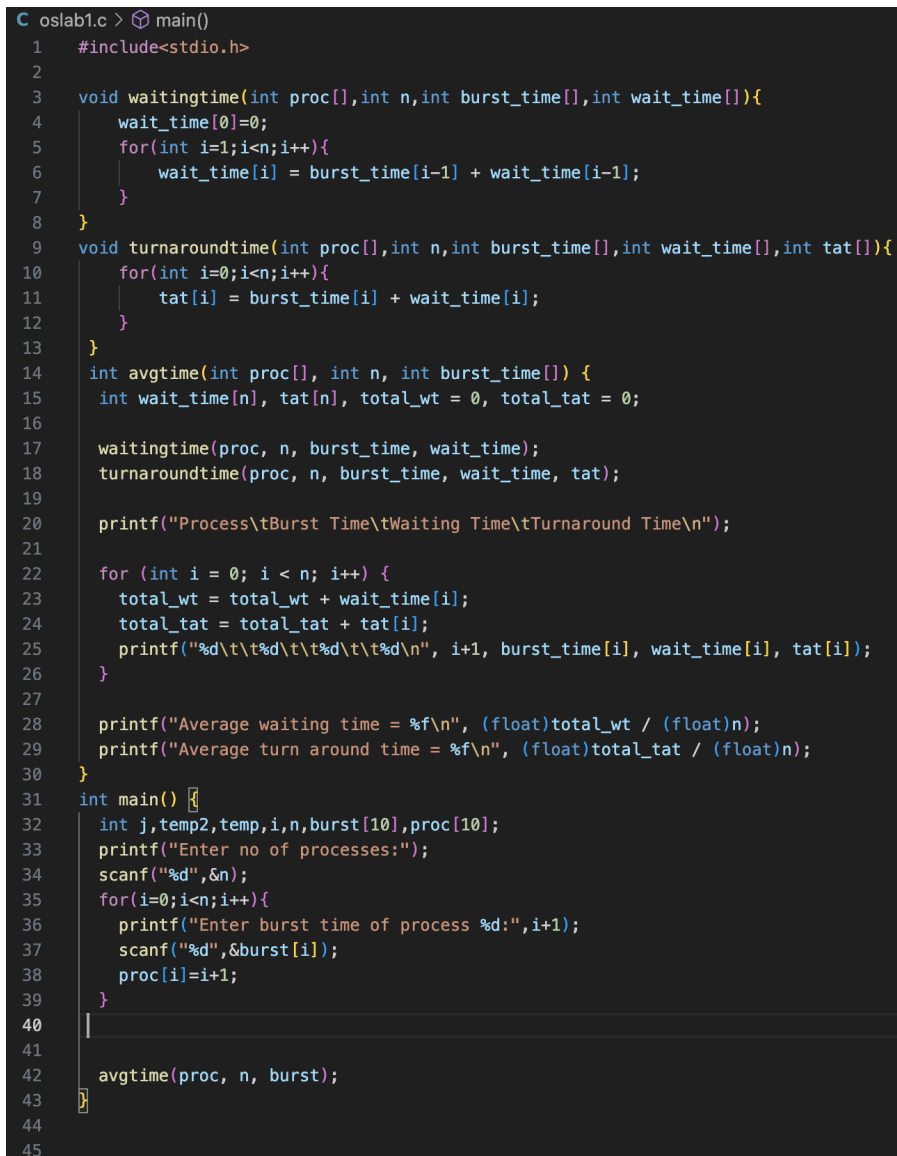
    printf("Average waiting time = %f\n", (float)total_wt / (float)n);
    printf("Average turn around time = %f\n", (float)total_tat / (float)n);
}
```

```

int main() {
    int j,temp2,temp,i,n,burst[10],proc[10];
    printf("Enter no of processes:");
    scanf("%d",&n);
    for(i=0;i<n;i++){
        printf("Enter burst time of process %d:",i+1);
        scanf("%d",&burst[i]);
        proc[i]=i+1;
    }
    avgtime(proc, n, burst);
}

```

Screenshots:



```

C oslab1.c > main()
1  #include<stdio.h>
2
3  void waitingtime(int proc[],int n,int burst_time[],int wait_time[]){
4      wait_time[0]=0;
5      for(int i=1;i<n;i++){
6          wait_time[i] = burst_time[i-1] + wait_time[i-1];
7      }
8  }
9
10 void turnaroundtime(int proc[],int n,int burst_time[],int wait_time[],int tat[]){
11     for(int i=0;i<n;i++){
12         tat[i] = burst_time[i] + wait_time[i];
13     }
14 }
15
16 int avgtime(int proc[], int n, int burst_time[]) {
17     int wait_time[n], tat[n], total_wt = 0, total_tat = 0;
18
19     waitingtime(proc, n, burst_time, wait_time);
20     turnaroundtime(proc, n, burst_time, wait_time, tat);
21
22     printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");
23
24     for (int i = 0; i < n; i++) {
25         total_wt = total_wt + wait_time[i];
26         total_tat = total_tat + tat[i];
27         printf("%d\t%d\t%d\t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
28     }
29
30     printf("Average waiting time = %f\n", (float)total_wt / (float)n);
31     printf("Average turn around time = %f\n", (float)total_tat / (float)n);
32 }
33
34 int main() {
35     int j,temp2,temp,i,n,burst[10],proc[10];
36     printf("Enter no of processes:");
37     scanf("%d",&n);
38     for(i=0;i<n;i++){
39         printf("Enter burst time of process %d:",i+1);
40         scanf("%d",&burst[i]);
41         proc[i]=i+1;
42     }
43
44     avgtime(proc, n, burst);
45 }

```

Output:

```
ryanthapa@Ryans-MacBook-Air c lang % cd "/Users/ryanthapa/Desktop/c lang/" && gcc oslab1.c -o oslab1 && "/Users/ryanthapa/Desktop/c lang/"oslab1
oslab1.c:30:1: warning: non-void function does not return a value [-Wreturn-type]
}
^
1 warning generated.
Enter no of processes:3
Enter burst time of process 1:10
Enter burst time of process 2:5
Enter burst time of process 3:15
Process Burst Time    Waiting Time    Turnaround Time
1          10           0             10
2           5          10             15
3          15          15             30
Average waiting time = 8.333333
Average turn around time = 18.333334
ryanthapa@Ryans-MacBook-Air c lang %
```

SJF Scheduling:

Code:

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

```
void waitingtime(int proc[],int n,int burst_time[],int wait_time[]){
    wait_time[0]=0;
    for(int i=1;i<n;i++){
        wait_time[i] = burst_time[i-1] + wait_time[i-1];
    }
}

void turnaroundtime(int proc[],int n,int burst_time[],int wait_time[],int tat[]){
    for(int i=0;i<n;i++){
        tat[i] = burst_time[i] + wait_time[i];
    }
}

int avgtime(int proc[], int n, int burst_time[]) {
    int wait_time[n], tat[n], total_wt = 0, total_tat = 0;

    waitingtime(proc, n, burst_time, wait_time);
    turnaroundtime(proc, n, burst_time, wait_time, tat);

    printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");

    for (int i = 0; i < n; i++) {
        total_wt = total_wt + wait_time[i];
        total_tat = total_tat + tat[i];
        printf("%d\t%d\t%d\t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
    }

    printf("Average waiting time = %f\n", (float)total_wt / (float)n);
    printf("Average turn around time = %f\n", (float)total_tat / (float)n);
}

int main() {
    int j,temp2,temp,i,n,burst[10],proc[10];
    printf("Enter no of processes:");
```

```

scanf("%d",&n);
for(i=0;i<n;i++){
    printf("Enter burst time of process %d:",i+1);
    scanf("%d",&burst[i]);
    proc[i]=i+1;
}
for(i=0;i<n;i++){
    for(j=i+1;j<n;j++){
        if(burst[i]>burst[j]){
            temp=burst[i];
            burst[i]=burst[j];
            burst[j]=temp;
            temp2=proc[i];
            proc[i]=proc[j];
            proc[j]=temp2;
        }
    }
}

avgtime(proc, n, burst);
}

```

Screenshots:

```

C oslab1.c > main()
1  #include<stdio.h>
2
3  void waitingtime(int proc[],int n,int burst_time[],int wait_time[]){
4      wait_time[0]=0;
5      for(int i=1;i<n;i++){
6          wait_time[i] = burst_time[i-1] + wait_time[i-1];
7      }
8  }
9  void turnaroundtime(int proc[],int n,int burst_time[],int wait_time[],int tat[]){
10     for(int i=0;i<n;i++){
11         tat[i] = burst_time[i] + wait_time[i];
12     }
13 }
14 int avgtime(int proc[], int n, int burst_time[]) {
15     int wait_time[n], tat[n], total_wt = 0, total_tat = 0;
16
17     waitingtime(proc, n, burst_time, wait_time);
18     turnaroundtime(proc, n, burst_time, wait_time, tat);
19
20     printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");
21
22     for (int i = 0; i < n; i++) {
23         total_wt = total_wt + wait_time[i];
24         total_tat = total_tat + tat[i];
25         printf("%d\t%d\t%d\t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
26     }
27
28     printf("Average waiting time = %f\n", (float)total_wt / (float)n);
29     printf("Average turn around time = %f\n", (float)total_tat / (float)n);
30 }
31 int main() {
32     int j,temp2,temp,i,n,burst[10],proc[10];
33     printf("Enter no of processes:");
34     scanf("%d",&n);
35     for(i=0;i<n;i++){
36         printf("Enter burst time of process %d:",i+1);
37         scanf("%d",&burst[i]);
38         proc[i]=i+1;
39     }
40     for(i=0;i<n;i++){
41         for(j=i+1;j<n;j++){
42             if(burst[i]>burst[j]){
43                 temp=burst[i];
44                 burst[i]=burst[j];
45                 burst[j]=temp;
46                 temp2=proc[i];
47                 proc[i]=proc[j];
48                 proc[j]=temp2;
49             }
50         }
51     }
52
53     avgtime(proc, n, burst);
54 }

```

Output;

```

ryanthapa@Ryans-MacBook-Air:~/Desktop/c lang$ gcc oslab1.c -o oslab1
ryanthapa@Ryans-MacBook-Air:~/Desktop/c lang$ ./oslab1
oslab1.c:30:1: warning: non-void function does not return a value [-Wreturn-type]
30 }
   ^
1 warning generated.
Enter no of processes:3
Enter burst time of process 1:10
Enter burst time of process 2:5
Enter burst time of process 3:2
Process Burst Time    Waiting Time    Turnaround Time
1          2          0           2
2          5          2           7
3          10         7          17
Average waiting time = 3.000000
Average turn around time = 8.666667
ryanthapa@Ryans-MacBook-Air:~/Desktop/c lang$

```