

Operating System Report Assignment Simulation Based 3rd Question

Student Name:-Rathor Mukeshkumar

Section:-EE034

Roll Number:-B-45

Email Address:-mukeshrathodm999@gmail.com

Github Link:- <https://rathor-830.github.io/Assignment/>

Que.3 Consider a scheduler which schedules the job by considering the arrival time of the processes where arrival time if given as 0 is discarded or displayed as error. The scheduler implements the shortest job first scheduling policy, but checks the queue of the processes after every process terminates and time taken for checking and arranging the process according to the shortest job is 2-time unit. Compute the waiting time, turnaround time and average waiting time and turnaround time of the processes. Also compute the total time taken by the processor to compute all the jobs.

The inputs for the number of requirements, arrival time and burst time should be provided by the user.

Consider the following units for reference.

| Process | Arrival time | Burst Time |
|---------|--------------|------------|
| 1 | 0 | 6 |
| 2 | 3 | 2 |
| 3 | 5 | 1 |
| 4 | 9 | 7 |
| 5 | 10 | 5 |
| 6 | 12 | 3 |
| 7 | 14 | 4 |
| 8 | 16 | 5 |
| 9 | 17 | 7 |
| 10 | 19 | 2 |

Solution:-

Code for the Burst Time:-

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int bt[10],p[10],n,temp,i,j,wt[10],sum=0;
    float avg;
```

```

printf("Enter total no of proces:");
scanf("%d",&n);
printf("\n Enter burst time for each process:-");
for(i=0;i<n;i++)
{
    printf("\nBurst time of process P%d:",i);
    scanf("%d",&bt[i]);
    p[i]=i;
}
for(i=0;i<n-1;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(bt[i]>bt[j])
        {
            temp=bt[i];
            bt[i]=bt[j];
            bt[j]=temp;
            temp=p[i];
            p[i]=p[j];
            p[j]=temp;
        }
    }
}
wt[0]=0;
for(i=1;i<n;i++)
{
    wt[i]=wt[i-1]+bt[i-1];
}
for(i=0;i<n;i++)
{
    sum+=wt[i];
}
avg=(float)sum/n;
printf("\n Waiting time for each process:-");
for(i=0;i<n;i++)
{
    printf("\n Waiting time for process P%d is %d sec.",p[i],wt[i]);

}
printf("\n Average waiting time is %f sec.",avg);
getch();
return 0;
}

```

Description:-

Burst Time:-Time required by a process for CPU execution.

Output:-

Test Case-1

```
Enter total no of proces:10

Enter burst time for each process:-
Burst time of process P0:6

Burst time of process P1:2

Burst time of process P2:1

Burst time of process P3:7

Burst time of process P4:5

Burst time of process P5:3

Burst time of process P6:4

Burst time of process P7:5

Burst time of process P8:7

Burst time of process P9:2

Waiting time for each process:-
Waiting time for process P2 is 0 sec.
Waiting time for process P1 is 1 sec.
Waiting time for process P9 is 3 sec.
Waiting time for process P5 is 5 sec.
Waiting time for process P6 is 8 sec.
Waiting time for process P7 is 12 sec.
Waiting time for process P4 is 17 sec.
Waiting time for process P0 is 22 sec.
Waiting time for process P8 is 28 sec.
Waiting time for process P3 is 35 sec.
Average waiting time is 13.100000 sec.
```

Code for the Arrival Time:-

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int bt[10],p[10],n,temp,i,j,wt[10],sum=0;
    float avg;
    printf("Enter total no of proces:");
    scanf("%d",&n);
    printf("\n Enter Arrival time for each process:-");
    for(i=0;i<n;i++)
    {
        printf("\nArrival time of process P%d:",i);
        scanf("%d",&bt[i]);
        p[i]=i;
    }
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(bt[i]>bt[j])
            {
                temp=bt[i];
                bt[i]=bt[j];
                bt[j]=temp;
                temp=p[i];
                p[i]=p[j];
                p[j]=temp;
            }
        }
    }
    wt[0]=0;
    for(i=1;i<n;i++)
    {
        wt[i]=wt[i-1]+bt[i-1];
    }
    for(i=0;i<n;i++)
    {
        sum+=wt[i];
    }
    avg=(float)sum/n;
    printf("\n Waiting time for each process:-");
    for(i=0;i<n;i++)
    {
        printf("\n Waiting time for process P%d is %d sec.",p[i],wt[i]);
    }
}
```


```
printf("\n Average waiting time is %f sec.",avg);
getch();
return 0;
}
```

Description:-

Arrival Time:-Time at which the process arrives in the ready queue.

Output:-

Test case-2

 F:\arrival time.exe

```
Enter total no of proces:10

Enter Arrival time for each process:-
Arrival time of process P0:0

Arrival time of process P1:3

Arrival time of process P2:5

Arrival time of process P3:9

Arrival time of process P4:10

Arrival time of process P5:12

Arrival time of process P6:14

Arrival time of process P7:16

Arrival time of process P8:17

Arrival time of process P9:19

Waiting time for each process:-
Waiting time for process P0 is 0 sec.
Waiting time for process P1 is 0 sec.
Waiting time for process P2 is 3 sec.
Waiting time for process P3 is 8 sec.
Waiting time for process P4 is 17 sec.
Waiting time for process P5 is 27 sec.
Waiting time for process P6 is 39 sec.
Waiting time for process P7 is 53 sec.
Waiting time for process P8 is 69 sec.
Waiting time for process P9 is 86 sec.
Average waiting time is 30.200001 sec.
```