جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الانظمة الطبية الذكية

Intelligent Medical Systems Department

Lab (4)

CPU scheduling algorithms

Subject: OPERATING SYSTEMS

Class: Second

Lecturer: Dr.Ahmad almhanna

Eng .Jumana altahier

Programar :Ahyab hashim

Study Year: 2023-2024

Write a C prgram simulate the following CPU scheduling algorithms:

c) Round Robin

DESCRIPTION

Assume all the processes arrive at the same time.

ROUND ROBIN CPU SCHEDULING ALGORITHM

For round robin scheduling algorithm, read the number of processes/jobs in the system, their CPU burst times, and the size of the time slice. Time slices are assigned to each process in equal portions and in circular order, handling all processes execution. This allows every process to get an equal chance. Calculate the waiting time and turnaround time of each of the processes accordingly.

HARDWARE REQUIREMENTS: Intel based Desktop Pc RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C/ Borland C.

THEORY:

Round Robin:

Example of RR with time quantum=3

Process	Burst time
aaa	4
Bbb	3
Ccc	2
Ddd	5
Eee	1

ALGORITHM

1. Start

2. Declare the array size

3. Read the number of processes to be inserted

4. Read the burst times of the processes

5. Read the Time Quantum

6. if the burst time of a process is greater than time Quantum then subtract time quantum form the burst time Else Assign the burst time to time quantum.

7.calculate the average waiting time and turn around time of the processes.

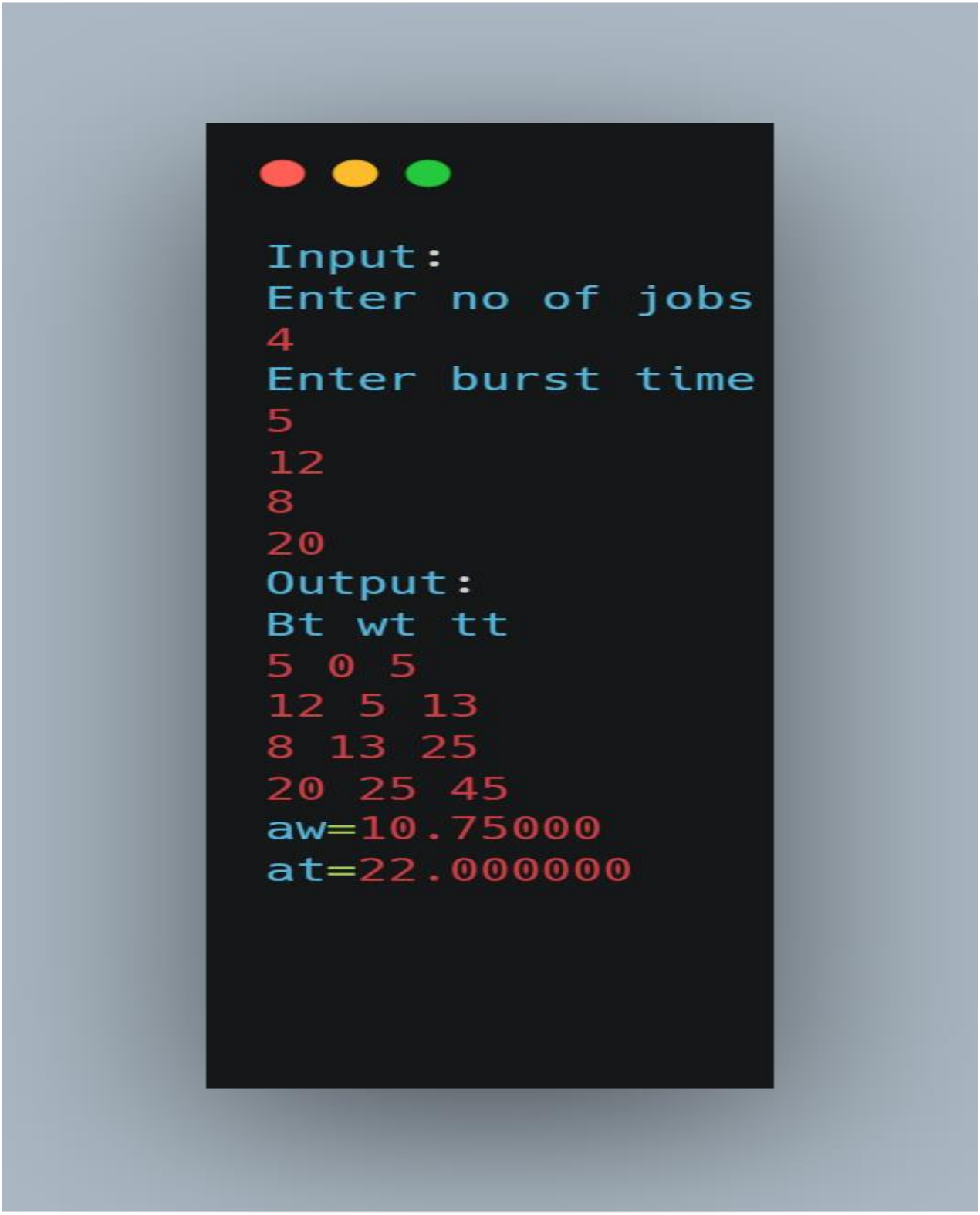
8. Display the values

9. Stop

PROGRAM:

```c
#include<stdio.h>
void main()
{
int st[10],bt[10],wt[10],tat[10],n,tq;
int i,count=0,swt=0,stat=0,temp,sq=0;
float awt=0.0,atat=0.0;
printf("Enter number of processes:");
scanf("%d",&n);
printf("Enter burst time for sequences:");
for(i=0;i<n;i++)
{
scanf("%d",&bt[i]);
st[i]=bt[i];
}
printf("Enter time quantum:");
scanf("%d",&tq);
while(1)
{
for(i=0,count=0;i<n;i++)
{
temp=tq;
if(st[i]==0)
{
count++;
continue;
}
if(st[i]>tq)
st[i]=st[i]-tq;
else
if(st[i]>=0)
{
temp=st[i];
st[i]=0;
}
sq=sq+temp;

tat[i]=sq;
}
if(n==count)
break;
}
for(i=0;i<n;i++)
{
wt[i]=tat[i]-bt[i];
swt=swt+wt[i];
stat=stat+tat[i];
}
awt=(float)swt/n;
atat=(float)stat/n;
printf("Process_no Burst time Wait time Turn around time");
for(i=0;i<n;i++)
printf("\n%d\t %d\t %d\t %d",i+1,bt[i],wt[i],tat[i]);
printf("\nAvg wait time is %f Avg turn around time is %f",awt,atat);
}
```


Input:
Enter no of jobs
4
Enter burst time
5
12
8
20
Output:
Bt wt tt
5 0 5
12 5 13
8 13 25
20 25 45
aw=10.750000
at=22.000000


THANK YOU!