

Student name: Abhay yadav

Student ID: 11811964

Roll.no: 21

Email address: abhay1902200@gmail.com

Githublink:

Q.19. The following processes are being scheduled using a preemptive, round robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an *idle task* (which consumes no CPU resources and is identified as *P_idle*). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

Thread Priority Burst Arrival

P1 40 20 0

P2 30 25 25

P3 30 25 30

P4 35 15 60

P5 5 10 100

P6 10 10 105

Write a C code to

- Show the scheduling order of the processes using a Gantt chart.
- What is the turnaround time for each process?
- What is the waiting time for each process?
- What is the CPU utilization rate?

Description:

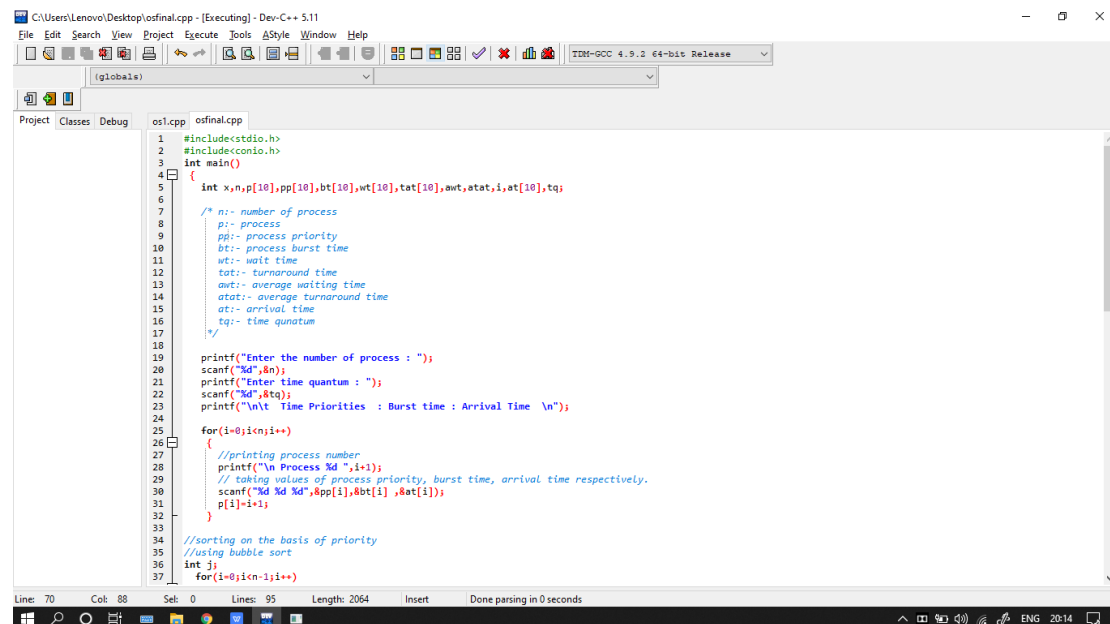
In the given problem priority are given to each process, and the preemption should be done in order of priority only. If the process gets preempted by higher priority process then we will place that process at the end of queue. The round robin scheduling

Algorithm:

For sorting Bubble sort is used.

Complexity:- $O(n^2)$.

Code:



```
1 #include<stdio.h>
2 #include<conio.h>
3 int main()
4 {
5     int x,n,p[10],pp[10],bt[10],wt[10],tat[10],awt,atat,i,at[10],tq;
6
7     /* n:- number of process
8        p:- process
9        pp:- process priority
10       bt:- process burst time
11       wt:- wait time
12       tat:- turnaround time
13       awt:- average waiting time
14       atat:- average turnaround time
15       at:- arrival time
16       tq:- time quantum
17     */
18
19     printf("Enter the number of process : ");
20     scanf("%d",&n);
21     printf("Enter time quantum : ");
22     scanf("%d",&tq);
23     printf("\n\t Time Priorities : Burst time : Arrival Time \n");
24
25     for(i=0;i<n;i++)
26     {
27         //printing process number
28         printf("\n Process %d ",i+1);
29         // taking values of process priority, burst time, arrival time respectively.
30         scanf("%d %d %d",&pp[i],&bt[i],&at[i]);
31         p[i]=i+1;
32     }
33
34     //sorting on the basis of priority
35     //using bubble sort
36     int j;
37     for(i=0;i<n-1;i++)
```

```
C:\Users\Lenovo\Desktop\osfinal.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools Style Window Help
(globals)
Project Classes Debug
os1.cpp osfinal.cpp
37 |
38 | for(i=0;i<n-1;i++)
39 | {
40 |     for(j=i+1;j<n;j++)
41 |     {
42 |         if(pp[i]<pp[j])
43 |         {
44 |             x=pp[i];
45 |             pp[i]=pp[j];
46 |             pp[j]=x;
47 |             x=bt[i];
48 |             bt[i]=bt[j];
49 |             bt[j]=x;
50 |             x=p[i];
51 |             p[i]=p[j];
52 |             p[j]=x;
53 |         }
54 |         //gc[t]=p[j];
55 |     }
56 |     wt[0]=0;
57 |     aut=0;
58 |     tat[0]=bt[0];
59 |     atat=tat[0];
60 |     for(i=1;i<n;i++)
61 |     {
62 |         if(tq<n || tq>n) {
63 |
64 |             wt[i]=tat[i-1];
65 |             aut+=wt[i];
66 |             tat[i]=wt[i]+bt[i];
67 |             atat=tat[i];
68 |         }
69 |     }
70 |     printf("*****");
71 |     //Gantt chart
72 |     printf("\n");
73 |     printf("Gantt chart\n");
Line: 73 Col: 26 Sel: 0 Lines: 95 Length: 2064 Insert Done parsing in 0 seconds

C:\Users\Lenovo\Desktop\osfinal.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools Style Window Help
(globals)
Project Classes Debug
os1.cpp osfinal.cpp
59 | atat=tat[0];
60 | for(i=1;i<n;i++)
61 | {
62 |     if(tq<n || tq>n) {
63 |
64 |         wt[i]=tat[i-1];
65 |         aut+=wt[i];
66 |         tat[i]=wt[i]+bt[i];
67 |         atat=tat[i];
68 |     }
69 | }
70 | printf("*****");
71 | //Gantt chart
72 | printf("\n");
73 | printf("Gantt chart\n");
74 | for(i=0;i<n;i++)
75 | {
76 |     printf("P %d ",p[i]);
77 | }
78 | //Displaying the process in tabular form
79 | printf("\n\nProcess \t Arrival time \t Burst Time \t Wait Time \t Turn Around Time \t Priority \n");
80 | for(i=0;i<n;i++)
81 | {
82 |     printf("\n %d",p[i]);
83 |     printf("\t\t %d",at[i]);
84 |     printf("\t\t %d",bt[i]);
85 |     printf("\t\t %d",wt[i]);
86 |     printf("\t\t %d",tat[i]);
87 |     printf("\t\t %d",pp[i]);
88 | }
89 |
90 | aut/=n;
91 | atat/=n;
92 | printf("\n Average Wait Time : %d ",aut);
93 | printf("\n Average Turn Around Time : %d",atat);
94 | getch();
95 | }
```

```
C:\Users\Lenovo\Desktop>osfinal.exe
Enter the number of process : 6
Enter time quantum : 10

Time Priorities : Burst time : Arrival Time

Process 1 40 20 0
Process 2 30 25 25
Process 3 30 25 30
Process 4 35 15 60
Process 5 5 10 100
Process 6 10 10 105
*****
Gantt chart
P 1 P 4 P 3 P 2 P 6 P 5
Process      Arrival time  Burst Time  Wait Time  Turn Around Time  Priority
1            0             20           0          20             40
4            25            15          20          35             35
3            30            25          35          60             30
2            60            25          60          85             30
6            100           10          85          95             10
5            105           10          95         105             5
Average Wait Time : 49
Average Turn Around Time : 66.5
```

19:

Ten students (s1,s2,s3,s4,s5,s6,s7,s8,s9,s10) are going to attend an event.

There are lots of gift shops, they all are going to the gift shops and randomly picking the gifts.

After picking the gifts they are randomly arriving in the billing counter.

The accountant gives the preference to that student who has maximum number of gifts.

Create a C program to define order of billed students?

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

```
#include<conio.h>
```

```
#include<string.h>
```

```
void main()
```

```
{
```

```
    char p[10][5],temp[5];
```

```
    int i,j,pt[10],wt[10],totwt=0,pr[10],temp1,n;
```

```
    float avgwt;
```

```
    printf("Enter no of girls:");
```

```

scanf("%d",&n);
for(i=0;i<n;i++)
{
    printf("enter girl %d name:",i+1);
    scanf("%s",&p[i]);
    printf("enter process time:");
    scanf("%d",&pt[i]);
    printf("enter no of gifts:");
    scanf("%d",&pr[i]);
}
for(i=0;i<n-1;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(pr[i]>pr[j])
        {
            temp1=pr[i];
            pr[i]=pr[j];
            pr[j]=temp1;
            temp1=pt[i];
            pt[i]=pt[j];
            pt[j]=temp1;
            strcpy(temp,p[i]);
            strcpy(p[i],p[j]);
            strcpy(p[j],temp);
        }
    }
}
wt[0]=0;
for(i=1;i<n;i++)

```

```

{
    wt[i]=wt[i-1]+wt[i-1];
    totwt=totwt+wt[i];
}
avgwt=(float)totwt/n;
printf("G_name\t P_time\t No.Gifts\tW_time\n");
for(i=0;i<n;i++)
{
    printf(" %s\t %d\t %d\t %d\n",p[i],pt[i],pr[i],wt[i]);
}
printf("total waiting time=%d\navg waiting time=%f",totwt,avgwt);
getch();
}

```