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

*P*1 40 20 0

*P*2 30 25 25

*P*3 30 25 30

*P*4 35 15 60

*P*5 5 10 100

*P*6 10 10 105

Write a C code to

a. Show the scheduling order of the processes using a Gantt chart.

b. What is the turnaround time for each process?

c. What is the waiting time for each process?

d. What is the CPU utilization rate?

Description:

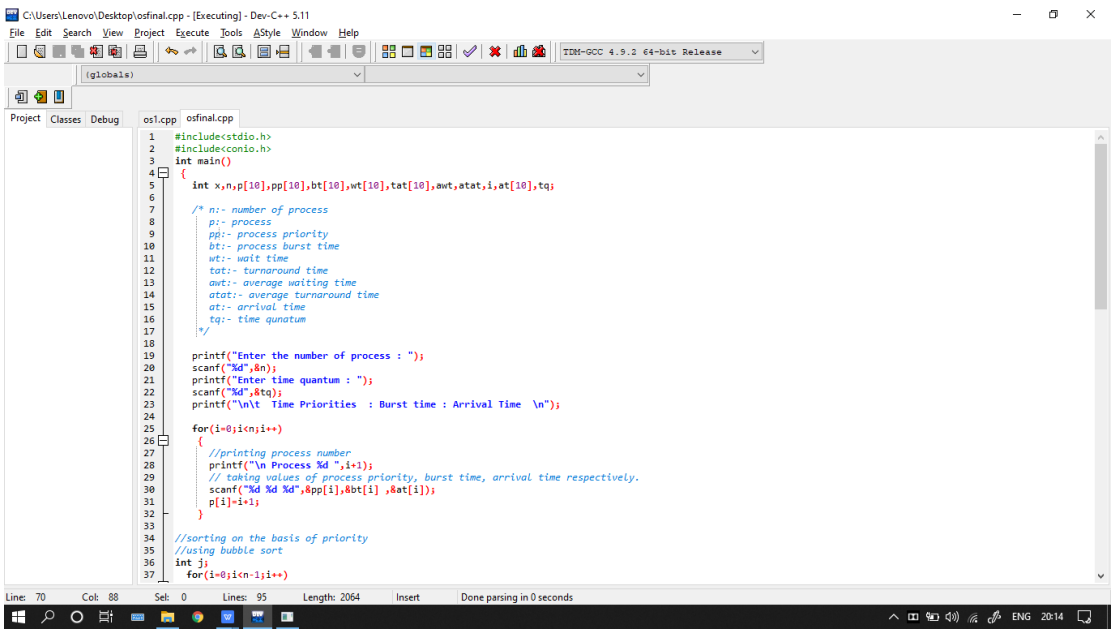
In the given problem priority are given to each process, and the premption should be done in order of priority only. If the process gets prempted by higher priority process then we will place that process at the end of queue. The round robin scheduling

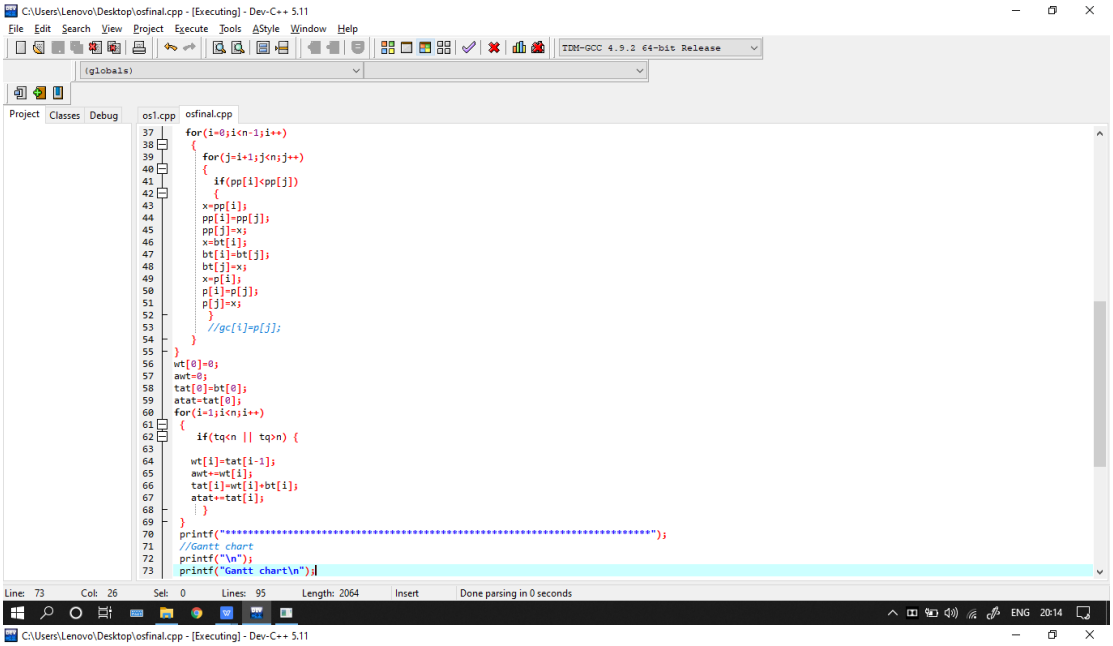
Algorithm:

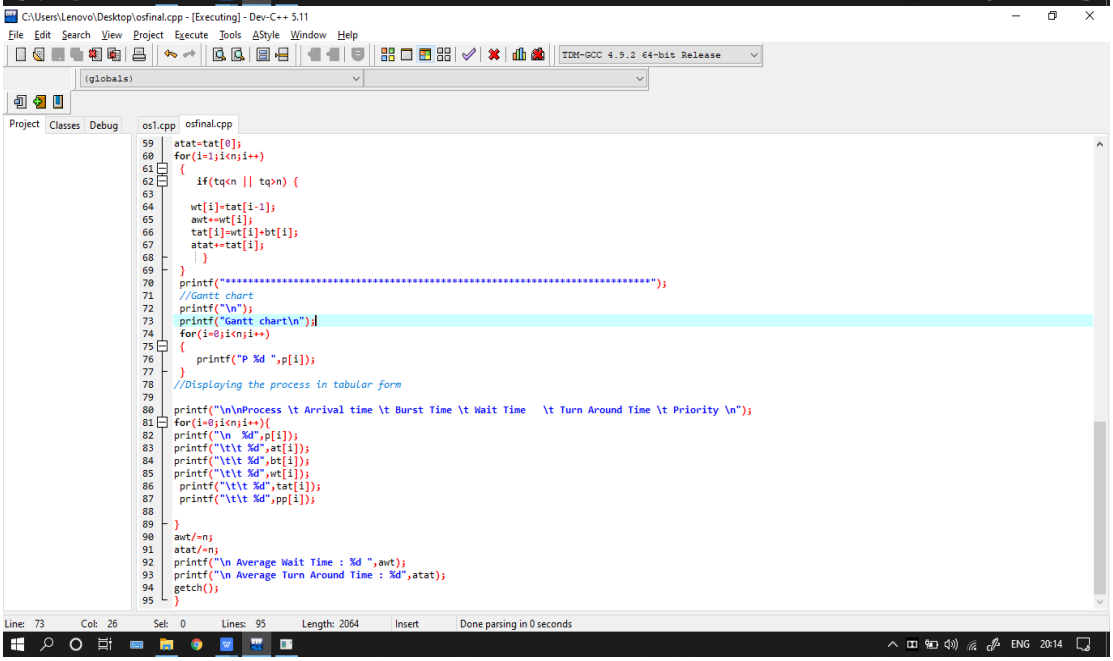
For sorting Bubble sort is used.

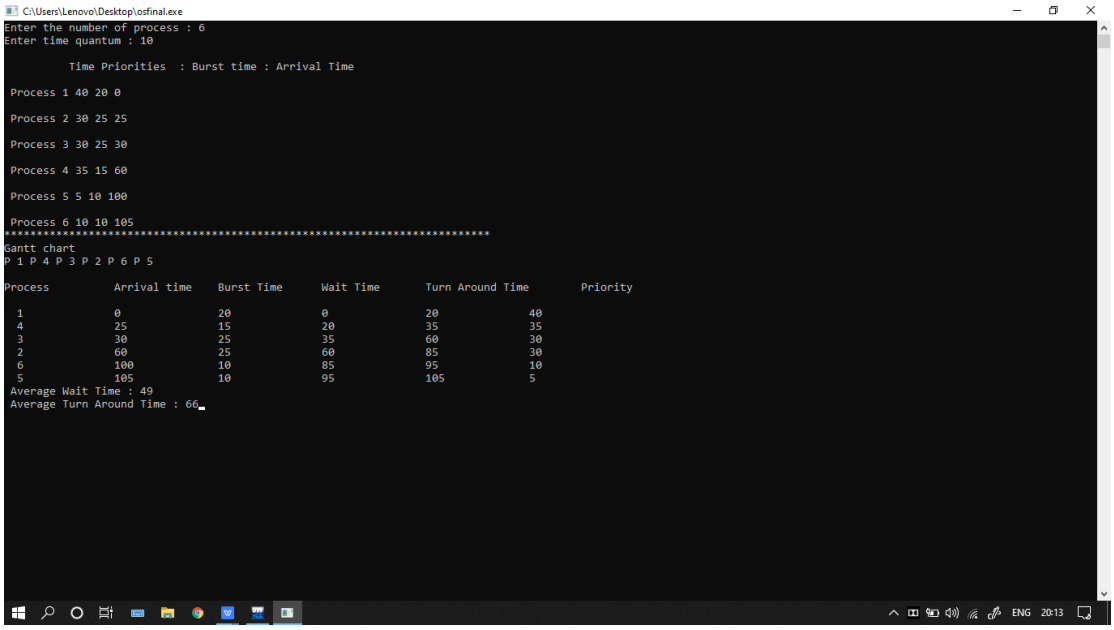
Complexity:- O(n^2).

Code:




C:\Users\Lenovo\Desktop\osfinal.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
os1.cpp osfinal.cpp
for(i=0;i<n-1;i++)
{
for(j=i+1;j<n;j++)
{
if(pp[i]<pp[j])
{
x=pp[i];
pp[i]=pp[j];
pp[j]=x;
x=bt[i];
bt[i]=bt[j];
bt[j]=x;
x=p[i];
p[i]=p[j];
p[j]=x;
}
//gc[i]=p[j];
}
}
wt[0]=0;
awt=0;
tat[0]=bt[0];
atat=tat[0];
for(i=1;i<n;i++)
{
if(tq<n || tq>n) {
wt[i]=tat[i-1];
awt+=wt[i];
tat[i]=wt[i]+bt[i];
atat+=tat[i];
}
}
printf("**");
//Gantt chart
printf("\n");
printf("Gantt chart\n");
Line: 73 Col: 26 Sel: 0 Lines: 95 Length: 2064 Insert Done parsing in 0 seconds
ENG 20:14


C:\Users\Lenovo\Desktop\osfinal.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
os1.cpp osfinal.cpp
atat=tat[0];
for(i=1;i<n;i++)
{
if(tq<n || tq>n) {
wt[i]=tat[i-1];
awt+=wt[i];
tat[i]=wt[i]+bt[i];
atat+=tat[i];
}
}
printf("**");
//Gantt chart
printf("\n");
printf("Gantt chart\n");
for(i=0;i<n;i++)
{
printf("P %d ",p[i]);
}
//Displaying the process in tabular form
printf("\n\nProcess \t Arrival time \t Burst Time \t Wait Time \t Turn Around Time \t Priority \n");
for(i=0;i<n;i++){
printf("\n %d",p[i]);
printf("\t\t %d",at[i]);
printf("\t\t %d",bt[i]);
printf("\t\t %d",wt[i]);
printf("\t\t %d",tat[i]);
printf("\t\t %d",pp[i]);
}
awt/=n;
atat/=n;
printf("\n Average Wait Time : %d ",awt);
printf("\n Average Turn Around Time : %d",atat);
getch();
}
Line: 73 Col: 26 Sel: 0 Lines: 95 Length: 2064 Insert Done parsing in 0 seconds
ENG 20:14



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 \t%d\n" ,p[i],pt[i],pr[i],wt[i]);
    }
    printf("total waiting time=%d\navg waiting time=%f",totwt,avgwt);
    getch();
}
```