Experiment No : 3
Experiment Name : Priority scheduling of Operating system implemented in C++
Theory :
Priority Scheduling is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority.The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority depends upon memory requirements, time requirements, etc.

Priority scheduling divided into two main types:
1.Preemptive Scheduling
In Preemptive Scheduling, the tasks are mostly assigned with their priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.
2.Non-Preemptive Scheduling
In this type of scheduling method, the CPU has been allocated to a specific process. The process that keeps the CPU busy, will release the CPU either by switching context or terminating. It is the only method that can be used for various hardware platforms. That's because it doesn't need special hardware (for example, a timer) like preemptive scheduling.

Characteristics of Priority Scheduling
a)A CPU algorithm that schedules processes based on priority.
b)It used in Operating systems for performing batch processes.
c)If two jobs having the same priority are READY, it works on a FIRST COME, FIRST SERVED basis.
d)In priority scheduling, a number is assigned to each process that indicates its priority level.
e)Lower the number, higher is the priority.
f)In this type of scheduling algorithm, if a newer process arrives, that is having a higher priority than the currently running process, then the currently running process is preempted.
Example of Priority Scheduling
Consider following five processes P1 to P5. Each process has its unique priority, burst time, and arrival time.

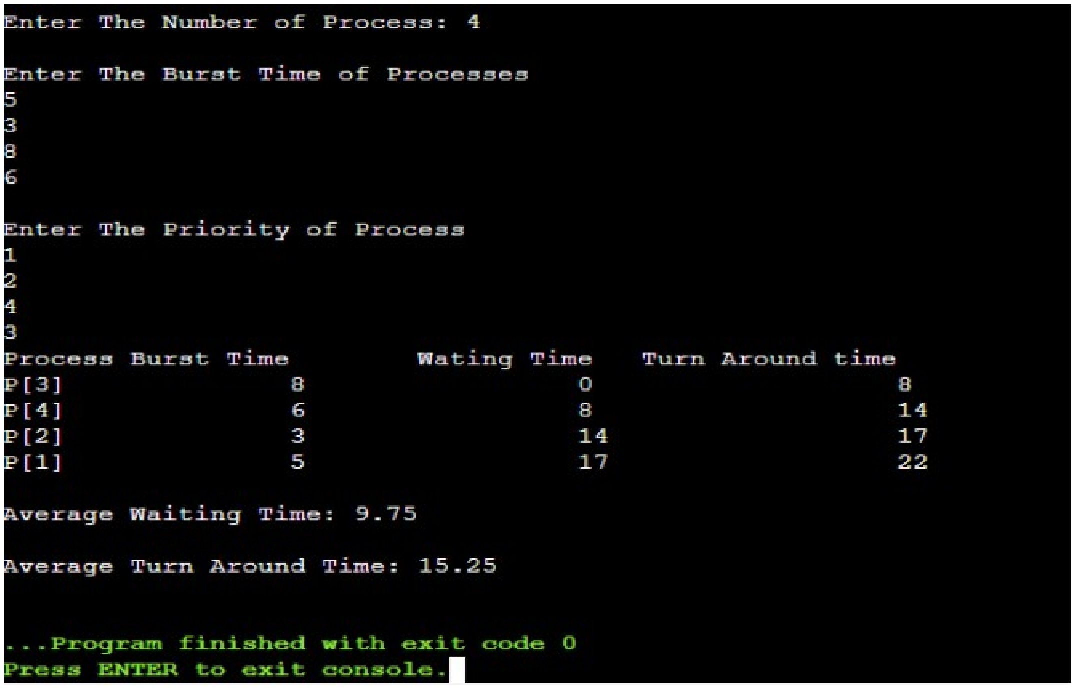
Code :

```
#include<stdio.h>



int main()

{

    int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;

    printf("Enter Total Number of Process:");

    scanf("%d",&n);
```

```
  printf("\nEnter Burst Time and Priority\n");
  for(i=0;i<n;i++)
  {
      printf("\nP[%d]\n",i+1);
      printf("Burst Time:");
      scanf("%d",&bt[i]);
      printf("Priority:");
      scanf("%d",&pr[i]);
      p[i]=i+1;           //contains process number
  }
 //sorting burst time, priority and process number in ascending order using selection sort
  for(i=0;i<n;i++)
  {
      pos=i;
      for(j=i+1;j<n;j++)
      {
          if(pr[j]<pr[pos])
              pos=j;
      }
 temp=pr[i];
      pr[i]=pr[pos];
      pr[pos]=temp;
temp=bt[i];
      bt[i]=bt[pos];
      bt[pos]=temp;
       temp=p[i];
```

```
        p[i]=p[pos];

        p[pos]=temp;

    }

 wt[0]=0; //waiting time for first process is zero

    //calculate waiting time

    for(i=1;i<n;i++)

    {     wt[i]=0;

        for(j=0;j<i;j++)

            wt[i]+=bt[j];

   total+=wt[i];

    }  avg_wt=total/n;      //average waiting time

    total=0;

 printf("\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time");

    for(i=0;i<n;i++)

    {

        tat[i]=bt[i]+wt[i];     //calculate turnaround time

        total+=tat[i];

        printf("\nP[%d]\t\t  %d\t\t    %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);

    }

  avg_tat=total/n;     //average turnaround time

    printf("\n\nAverage Waiting Time=%d",avg_wt);

printf("\nAverage Turnaround Time=%d\n",avg_tat);

return 0;

}
```

Output :



```
Enter The Number of Process: 4

Enter The Burst Time of Processes
5
3
8
6

Enter The Priority of Process
1
2
4
3
Process Burst Time        Wating Time    Turn Around time
P[3]             8                  0                   8
P[4]             6                  8                  14
P[2]             3                 14                  17
P[1]             5                 17                  22

Average Waiting Time: 9.75

Average Turn Around Time: 15.25


...Program finished with exit code 0
Press ENTER to exit console.
```

Discussion & Conclusion: After doing this experiment, we learned about the Priority algorithm.Then we learned how to calculate the waiting time and the turnaround time andalso average waiting and turnaround time for each process without considering the arrivaltime for both preemptive and non preemptive.And last we have implemented the coding of this algorithm successfully.

Department of Electrical and Computer Engineering

Course Title : Operating system Sessional

Course Code: ECE 3222

Submission Date : 17-12-2022

Submitted To:

Hafsa Binte Kibria

Lecturer, Department of ECE

RUET

Submitted By:

Tamim Hasan

Roll : 1810044

Dept of ECE

Session : 2018-19

RUET