

**Mawlana Bhashani Science and Technology University**

**Lab-Report**

Report No:09

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:01/09/2020

Date of Submission:10/09/2020

**Submitted by Submitted To**

Name:Md Milon Ahmed

ID:IT-18034

3rd  year 1st semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

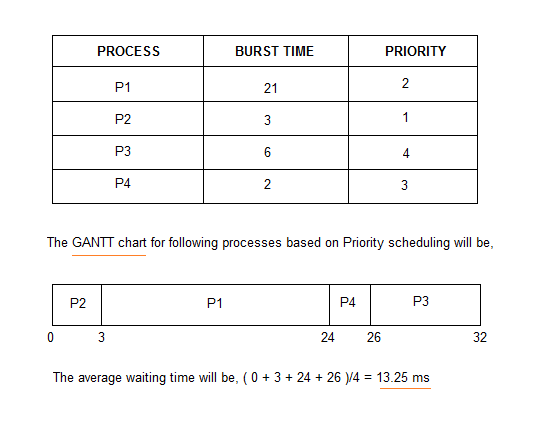
# Experiment no : 09

# Experiment Name : Implementation of Priority Scheduling Algorithm

# 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.



# Working Procedure:

Coding implementation with c++ ….

#include<iostream>

using namespace std;

int main()

{

    int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg\_wt,avg\_tat;

    cout<<"Enter Total Number of Process:";

    cin>>n;

    cout<<"\nEnter Burst Time and Priority\n";

    for(i=0;i<n;i++)

    {

        cout<<"\nP["<<i+1<<"]\n";

        cout<<"Burst Time:";

        cin>>bt[i];

        cout<<"Priority:";

        cin>>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;

    cout<<"\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];

        cout<<"\nP["<<p[i]<<"]\t\t  "<<bt[i]<<"\t\t    "<<wt[i]<<"\t\t\t"<<tat[i];

    }

    avg\_tat=total/n;     //average turnaround time

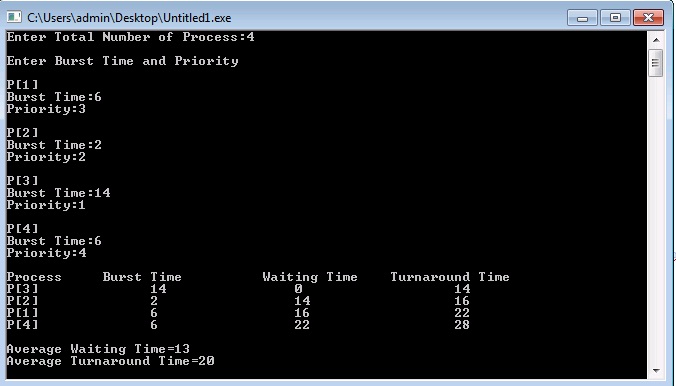
    cout<<"\n\nAverage Waiting Time="<<avg\_wt;

    cout<<"\nAverage Turnaround Time="<<avg\_tat;

    return 0;

}

Output:



# Discussion:

We learn from this lab

1.[What is Priority scheduling?](https://www.guru99.com/priority-scheduling-program.html" \l "1)

2.[Types of Priority Scheduling](https://www.guru99.com/priority-scheduling-program.html" \l "2)

3.[Characteristics of Priority Scheduling](https://www.guru99.com/priority-scheduling-program.html" \l "3)

4.[Example of Priority Scheduling](https://www.guru99.com/priority-scheduling-program.html" \l "4)

5.[Advantages of priority scheduling](https://www.guru99.com/priority-scheduling-program.html" \l "5)