Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 09

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:

Date of Submission:

Submitted by

Name: Moniruzzaman

ID:IT-18007

3rd year 1stsemester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment No :- 09

Experiment Name: Implementation of Priority scheduling algorithm.

Objectives:

i What is Priority Scheduling algorithm
ii How to implementation in C

Theory:

In priority scheduling algorithm each process has a priority associated with it and as each process hits the queue, it is stored in based on its priority so that process with higher priority are dealt with first. It should be noted that equal priority processes are scheduled in FCFS order.

To prevent high priority processes from running indefinitely the scheduler may decrease the priority of the currently running process at each clock tick (i.e., at each clock interrupt). If this action causes its priority to drop below that of the next highest process, a process switch occurs. Alternatively, each process may be assigned a maximum time quantum that it is allowed to run. When this quantum is used up, the next highest priority process is given a chance to run.

Implementation:

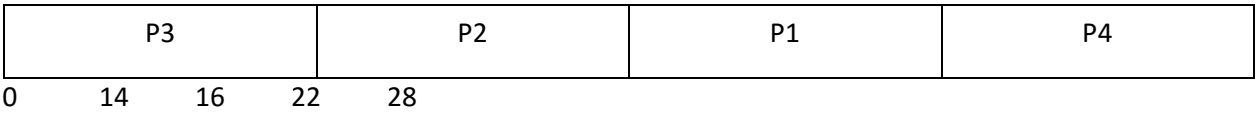
1. Take input burst time and priority of a process.
2. Sort the process according to the priority.
3. Then calculate **waiting time = start time –arrival time**.
4. Then calculate **turnaround time = waiting time + burst time**.

Example:

Process	Burst Time	Priority
P1	6	3
P2	2	2
P3	14	1

P4	6	4

Grant chart:



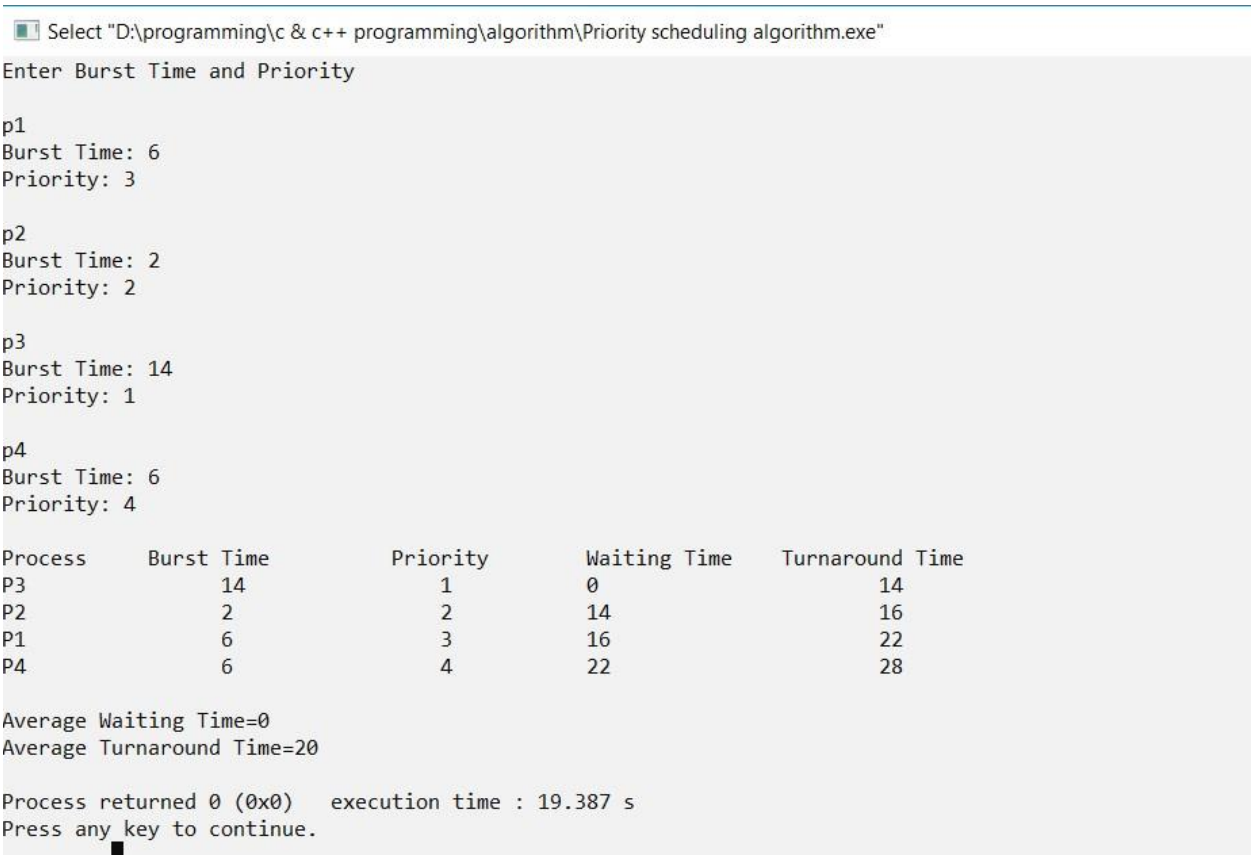
Source code:

```
#include<stdio.h>
int main()
{
	int bt[100],p[100],wt[100],tat[100],pr[100],i,j,n,pos,temp,avg_tat;
double total=0,avg_wt;
	printf("Enter The Number of Process:");
scanf("%d",&n);
	printf("\nEnter Burst Time and Priority\n");
for(i=0;i<n;i++)
	{
		printf("\np%d\nBurst Time: ",i+1);
		scanf("%d",&bt[i]);
printf("Priority: ");
scanf("%d",&pr[i]);
		p[i]=i+1;
	}
	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;
      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;
total=0;
      printf("\nProcess\t    Burst Time    \tPriority\tWaiting Time\tTurnaround Time");
for(i=0;i<n;i++)
      {
         tat[i]=bt[i]+wt[i];
total+=tat[i];
         printf("\nP%d\t\t  %d\t\t    %d\t\t%d\t\t\t\t%d",p[i],bt[i],pr[i],wt[i],tat[i]);
      }
      avg_tat=total/n;
      printf("\n\nAverage Waiting Time=%d",avg_wt);
printf("\nAverage Turnaround Time=%d\n",avg_tat);
}
```

Output:



Conclusion:

In this lab I learn how to implement Priority scheduling algorithm and also run the code and shows the output and output is expected.