

# Lab-Report

Report No: 09

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:

Date of Submission:

## Submitted by

Name: Ashikur Rahman Miran

ID:IT-18014

3<sup>rd</sup> year 1<sup>st</sup> semester

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 implement Priority scheduling algorithm.

### Theory:

Priority scheduling is a scheduling process based on priority. In this method the process works first which priority is high. Process with the same priority are execute based on arrival time.

### 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:

P3	P2	P1	P4
0	14	16	22
			28

## 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%d",p[i],bt[i],pr[i],wt[i],tat[i]);
    }
    avg_tat=total/n;
    printf("\n\nAverage Waiting Time=%d",avg_wt);
    printf("\n\nAverage Turnaround Time=%d\n",avg_tat);
}

```

## Output:

```
Select "D:\programming\c & c++ programming\algorithm\Priority scheduling algorithm.exe"
Enter Burst Time and Priority

p1
Burst Time: 6
Priority: 3

p2
Burst Time: 2
Priority: 2

p3
Burst Time: 14
Priority: 1

p4
Burst Time: 6
Priority: 4

Process      Burst Time      Priority      Waiting Time      Turnaround Time
P3           14              1            0                 14
P2           2              2            14                16
P1           6              3            16                22
P4           6              4            22                28

Average Waiting Time=0
Average Turnaround Time=20

Process returned 0 (0x0)   execution time : 19.387 s
Press any key to continue.
```

## 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.