

## Lab-Report

Report No: 9

Course code: ICT-3110

Course title: Operating Systems Lab

Date of Performance:

Date of Submission: 13/09/2020

### Submitted by

Name: Mahfuza Talukdar

ID:IT-18009

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 : 9**

**Experiment Name** : Implementation of Priority Scheduling Algorithm.

### **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.

Also Read: C Program for Shortest Job First (SJF) Scheduling Algorithm

Also Read: C/C++ Program for First Come First Served (FCFS) Scheduling Algorithm

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.

### **Corresponding Code :**

Code for Priority Scheduling Algorithm –

```
#include<stdio.h>

#include<conio.h>

int main()
{
    int x,n,p[10],pp[10],pt[10],w[10],t[10],awt,atat,i;

    printf("Enter the number of process : ");

    scanf("%d",&n);

    printf("\nEnter process : time priorities \n");

    for(i=0;i<n;i++)
    {
        printf("\nProcess no %d : ",i+1);

        scanf("%d %d",&pt[i],&pp[i]);
```

```
    p[i]=i+1;
}
for(i=0;i<n-1;i++)
{
    for(int j=i+1;j<n;j++)
    {
        if(pp[i]<pp[j])
        {
            x=pp[i];
            pp[i]=pp[j];
            pp[j]=x;
            x=pt[i];
            pt[i]=pt[j];
            pt[j]=x;
            x=p[i];
            p[i]=p[j];
            p[j]=x;
        }
    }
}
w[0]=0;
awt=0;
t[0]=pt[0];
atat=t[0];
for(i=1;i<n;i++)
{
    w[i]=t[i-1];
```

```

    awt+=w[i];
    t[i]=w[i]+pt[i];
    atat+=t[i];
}
printf("\n\n Job \t Burst Time \t Wait Time \t Turn Around Time \t Priority \n");
for(i=0;i<n;i++)
    printf("\n %d \t %d \t %d \t %d \t %d \n",p[i],pt[i],w[i],t[i],pp[i]);
awt/=n;
atat/=n;
printf("\n Average Wait Time : %d \n",awt);
printf("\n Average Turn Around Time : %d \n",atat);
getch();
}

```

Output :

```

Enter the number of process :
4

Enter process : time priorities
Process no 1 : 4 5
Process no 2 : 3 6
Process no 3 : 5 7
Process no 4 : 4 9

Job      Burst Time      Wait Time      Turn Around Time      Priority
4         4              0              4              9
3         5              4              9              7
2         3              9              12             6
1         4              12             16             5
Average Wait Time : 6
Average Turn Around Time : 10

```

**Discussion :**

From this lab, we know that how implemented Priority scheduling algorithm using C language. We can solve any problem of this algorithm in future.