

Lab-Report

Report No:

Course code: ICT-3110

Course title: Operating Systems Lab

Date of Performance:

Date of Submission:

Submitted by

Name: Hasibul Islam Imon

ID:IT-18047

3th year 1st semester

Session: 2017-18

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Lab report name: Implementation of Priority Scheduling Algorithm

Objectives:

- 1. A CPU algorithm that schedules processes based on priority.
- It used in Operating systems for performing batch processes.
- If two jobs having the same priority are READY, it works on a FIRST COME, FIRST SERVED basis.
- In priority scheduling, a number is assigned to each process that indicates its priority level.
- Lower the number, higher is the priority.
- 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.

Question No.1 What is Priority Scheduling Algorithm?

Answer: 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.

Question NO.2 How to implement in c?

Answer:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<string.h>
```

```
void main()
```

```
{
```

```
int et[20],at[10],n,i,j,temp,p[10],st[10],ft[10],wt[10],ta[10];
```

```
int totwt=0,totta=0;
```

```

float awt,ata;

char pn[10][10],t[10];

//clrscr();

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

scanf("%d",&n);

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

{

    printf("Enter process name,arrivaltime,execution time & priority:");

    //flushall();

    scanf("%s%d%d%d",pn[i],&at[i],&et[i],&p[i]);

}

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

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

        {

            if(p[i]<p[j])

                {

                    temp=p[i];

                    p[i]=p[j];

                    p[j]=temp;

                    temp=at[i];

                    at[i]=at[j];

                    at[j]=temp;

```

```

        temp=et[i];
        et[i]=et[j];
        et[j]=temp;
        strcpy(t,pn[i]);
        strcpy(pn[i],pn[j]);
        strcpy(pn[j],t);
    }
}

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

{

    if(i==0)
    {
        st[i]=at[i];
        wt[i]=st[i]-at[i];
        ft[i]=st[i]+et[i];
        ta[i]=ft[i]-at[i];
    }
    else
    {
        st[i]=ft[i-1];

```

```

        wt[i]=st[i]-at[i];

        ft[i]=st[i]+et[i];

        ta[i]=ft[i]-at[i];

    }

    totwt+=wt[i];

    totta+=ta[i];

}

awt=(float)totwt/n;

ata=(float)totta/n;

printf("\n Pname\t arrivaltime\t executiontime\t priority\t waitingtime\t tatime");

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

printf("\n %s\t %5d\t %5d\t %5d\t %5d\t %5d",pn[i],at[i],et[i],p[i],wt[i],ta[i]);

    printf("\n Average waiting time is:%f",awt);

    printf("\n Average turnaroundtime is:%f",ata);

    getch();

}

```

Output:

```

Enter the number of process:3
Enter process name,arrivaltime,execution time & priority:1 2 1 3
Enter process name,arrivaltime,execution time & priority:3 5 4 1
Enter process name,arrivaltime,execution time & priority:3 4 6 2

Pname   arrivaltime   executiontime   priority   waitingtime   tatime
3        5              4              1          0             4
3        4              6              2          5             11
1        2              1              3          13            14
Average waiting time is:6.000000
Average turnaroundtime is:9.666667

```

Conclusion:

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.