



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 : 25/09/2020

Submitted By:

Name: Md Abdur Rahim

ID: IT-18024

3rd Year 1st Semester

Session : 2017-18

Dept. of ICT

MBSTU

Submitted To:

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU

Experiment No : 09

Experiment Name : Implementation of Priority Scheduling Algorithm.

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

Priority Scheduling Algorithm:

→ Processes with same priority are execute on first come first served (FCFS) basis.

→ Priority scheduling is a non-preemptive algorithm and one of most common scheduling algorithm in batch systems.

→ Each process is assigned a priority process with highest priority is to be execute first and so on.

Algorithm:

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Sort the ready queue according to the priority number.

Step 5: Set the waiting of the first process as $_0$ and its burst time as its turnaround time

Step 6: Arrange the processes based on process priority

Step 7: for each process in the Ready Q calculate

a) Waiting time(n)= waiting time (n-1) + Burst time (n-1)

b) Turnaround time (n)= waiting time(n)+Burst time(n)

Step 8: Calculate

c) Average waiting time = Total waiting Time / Number of process

d) Average Turnaround time = Total Turnaround Time / Number of process Print the results in an order

Step 9: Stop the process.

Code Implementation :

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

```
int main()
```

```
{
```

```
    int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;
```

```
    printf("Enter Total Number of Process:");
```

```
    scanf("%d",&n);
```

```
    printf("\nEnter Burst Time and Priority\n");
```

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

```
    {
```

```
        printf("\nP[%d]\n",i+1);
```

```
        printf("Burst Time:");
```

```
        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 \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",p[i],bt[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);

return 0;

}

```

Output :

```
Enter the num of Process : 5
Enter the burst time & priority time for Process P0 = 10 3
Enter the burst time & priority time for Process P1 = 1 1
Enter the burst time & priority time for Process P2 = 2 4
Enter the burst time & priority time for Process P3 = 1 5
Enter the burst time & priority time for Process P4 = 5 2
```

Process time	priority	burst time	waiting time	turnaround
P1	1	1	0	1
P4	2	5	1	6
P0	3	10	6	16
P2	4	2	16	18
P3	5	1	18	19

```
Average waiting time 8.20
Average turnaround time 12.00
```

Discussion :

This lab helps to learn priority algorithm. Using this algorithm we can find waiting time and turnaround time that depend the priority.