



Mawlana Bhashani Science and Technology University

Lab-Report

Lab Report No: 09

Lab Report Name: Implementation of Priority Scheduling algorithm.

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance: 17-09-2020

Date of Submission:

Submitted by

Name: Md Masud Rana

ID: IT-18002

3rd year 1st semester

Session: 2017-18

Dept. of ICT.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

Experiment No: 09

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.

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.

Priority scheduling can be of two types:

1. **Preemptive Priority Scheduling:** If the new process arrived at the ready queue has a higher priority than the currently running process, the CPU is preempted, which means the processing of the current process is stopped and the incoming new process with higher priority gets the CPU for its execution.
2. **Non-Preemptive Priority Scheduling:** In case of non-preemptive priority scheduling algorithm if a new process arrives with a higher priority than the current running process, the incoming process is put at the head of the ready queue, which means after the execution of the current process it will be processed.

Code-

```
Open ▾  *priority_code.c  
~/program Save     
#include <stdio.h>  
int main()  
{  
  
    int bt[20],wt[20],p[20],tat[20],priority[20];  
    float avwt=0,avtat=0;  
  
    int i,j,n,temp,key;  
  
    printf("\nEnter the number of the processes: ");  
  
    scanf("%d",&n);  
  
    for(i=0;i<n;i++)  
    {  
  
        printf("\nEnter the burst time and priority of the process P[%d]: ",i);  
        scanf("%d",&bt[i]);  
        scanf("%d",&priority[i]);  
        p[i]=i;  
  
    }  
  
    for(i=0;i<n;i++)  
    {  
        key=i;  
        for(j=i+1;j<n;j++)  
        {  
            if(priority[j]<priority[key])  
            {  
                key=j;  
            }  
        }  
        temp=bt[i];  
        bt[i]=bt[key];  
        bt[key]=temp;  
  
        temp=priority[i];  
        priority[i]=priority[key];  
        priority[key]=temp;  
  
        temp=p[i];  
        p[i]=p[key];  
        p[key]=temp;  
    }  
}
```

```

wt[0]=0;
tat[0]=bt[0];
avtat=tat[0];

for(i=1;i<n;i++)
{
    wt[i]=wt[i-1]+bt[i-1];

    tat[i]=tat[i-1]+bt[i];

    avwt+=wt[i];

    avtat+=tat[i];
}

avwt=avwt/n;
avtat=avtat/n;

printf("\n\nPROCESS\t\twaiting time\tburst time\tTurnaround time\n");

printf("\n");

for(i=0;i<n;i++)
{
    printf("P[%d]\t\t%d\t\t%d\t\t%d\n",p[i],wt[i],bt[i],tat[i]);
}

printf("\n\nAverage waiting time: %.2f",avwt);
printf("\n\nAverage Turn around time is: %.2f",avtat);
printf("\n");

return 0;

```

```

}
```

Output:

```
masud@masud-VirtualBox: ~/program
File Edit View Search Terminal Help
masud@masud-VirtualBox:~/program$ gcc priority_code.c -o priority_code.out
masud@masud-VirtualBox:~/program$ ./priority_code.out

Enter the number of the processes: 4

Enter the burst time and priority of the process P[0]:
10 1

Enter the burst time and priority of the process P[1]:
20 2

Enter the burst time and priority of the process P[2]:
30 3

Enter the burst time and priority of the process P[3]:
40 4

PROCESS          waiting time    burst time    Turnaround time
P[0]              0              10            10
P[1]             10              20            30
P[2]             30              30            60
P[3]             60              40           100

Average waiting time: 25.00
Average Turn around time is: 50.00
masud@masud-VirtualBox:~/program$
```

Discussion: We have implemented the priority scheduling algorithm in C programming language. We can realize that the priority scheduling algorithm executes the processes depending upon their priority. Each process is allocated a priority and the process with the highest priority is executed first. This lab work helps us to understand about the priority scheduling algorithm.

