Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 09

Course code: ICT-3110

Course title:Operating Systems Lab

Date of Performance:13-09-2020

Date of Submission:19-09-2020

Submitted by

Name: Sabikun Nahar Piya

ID:IT-18020

3rd year 1st 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

Theory:

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.

Implementation:

step 1: Declare the array size.

step 2: Get the number of elements to be inserted.

step 3: Get the priority for each process and value.

step 4: Start with the higher priority process from it's initial position let other process to be queue.

step 5: Calculate the total number of burst time.

step 6: Display the values.

Working process:

Code for priority scheduling algorithm:

#include<stdio.h>

int main()

{

```c
    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;
        }
```

```c
      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++)
```

```c
{
    tat[i]=bt[i]+wt[i];

    total+=tat[i];

    printf("\nP[%d]\t\t %d\t\t   %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);

}

avg_tat=total/n;

printf("\n\nAverage Waiting Time=%d",avg_wt);

printf("\nAverage Turnaround Time=%d\n",avg_tat);

return 0;
}
```

Output:

```
Enter Burst Time and Priority

P[1]
Burst Time:6
Priority:3

P[2]
Burst Time:2
Priority:4

P[3]
Burst Time:14
Priority:1

P[4]
Burst Time:6
Priority:4

Process        Burst Time          Waiting Time     Turnaround Time
P[3]              14                    0                   14
P[1]              6                     14                  20
P[2]              2                     20                  22
P[4]              6                     22                  28

Average Waiting Time=14
Average Turnaround Time=21

Process returned 0 (0x0)    execution time : 73.149 s
Press any key to continue.
```

Discussion:

This is 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 algorithm,the chances of indefinite blcking or starvation. But this works successfully and it is easy to implement.