Mawlana Bhashani Science and Technology University

Lab-Report

Report No:07

Course code:ICT-3110

Course title:Operating Systems Lab

Date of Performance:07-09-2020

Date of Submission:16-09-2020

Submitted by

Name:Meraz Ahmed

ID:IT-18005

3rd year 1st semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment No: 07

Experiment Name: Implementation of FCFS Scheduling Algorithm.

First-Come-First-Served (FCFS) Scheduling Algorithm:

First-Come-First-Served algorithm is the simplest scheduling algorithm is the simplest scheduling algorithm. Processes are dispatched according to their arrival time on the ready queue. Being a non-preemptive discipline, once a process has a CPU, it runs to completion. The FCFS scheduling is fair in the formal sense or human sense of fairness but it is unfair in the sense that long jobs make short jobs wait and unimportant jobs make important jobs wait. FCFS is more predictable than most of other schemes since it offers time. FCFS scheme is not useful in scheduling interactive users because it cannot guarantee good response time. One of the major drawback of this scheme is that the average time is often quite long. The First-Come-First-Served algorithm is rarely used as a master scheme in modern operating systems but it is often embedded within other schemes.

Algorithm:

1. Start the process
2. Declare the array size
3. Get the number of processes to be inserted
4. Get the value
5. Start with the first process from it's initial position let other process to be in queue
6. Calculate the total number of burst time
7. Display the values
8. Stop the process

PsuduCode:

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

```
void main()
{
   int n,a[10],b[10],t[10],w[10],g[10],i,m;
   float att=0,awt=0;
   for(i=0; i<10; i++)
   {
      a[i]=0;
      b[i]=0;
      w[i]=0;
      g[i]=0;
   }
   printf("enter the number of process:");
   scanf("%d",&n);
   printf("enter the burst times:");
   for(i=0; i<n; i++)
      scanf("%d",&b[i]);
   printf("\nenter the arrival times:");
   for(i=0; i<n; i++)
      scanf("%d",&a[i]);
   g[0]=0;
   for(i=0; i<10; i++)
      g[i+1]=g[i]+b[i];
   for(i=0; i<n; i++)
   {
      w[i]=g[i]-a[i];
      t[i]=g[i+1]-a[i];
      awt=awt+w[i];
```

```
    att=att+t[i];
  }
  awt =awt/n;
  att=att/n;
  printf("\n\tprocess\twaiting time\tturn arround time\n");
  for(i=0; i<n; i++)
  {
    printf("\tp%d\t\t%d\t\t%d\n",i,w[i],t[i]);
  }
  printf("the average waiting time is %f\n",awt);
  printf("the average turn around time is %f\n",att);
}
```

Output:

```
enter the number of process:3
enter the burst times:6 9 7

enter the arrival times:1 2 3

        process waiting time    turn arround time
        p0              -1              5
        p1              4               13
        p2              12              19
the average waiting time is 5.000000
the average turn around time is 12.333333

Process returned 42 (0x2A)   execution time : 78.417 s
Press any key to continue.
```