Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 07

Course code: ICT-3110

Course title: Operating Systems Lab

Date of Performance:

Date of Submission: 15/09/2020

Submitted by

Name: Md Ahadul Haque

ID:IT-18045

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.

Theory :

First come, first served (FCFS) is an operating system process scheduling algorithm and a network routing management mechanism that automatically executes queued requests and processes by the order of their arrival. With first come, first served, what comes first is handled first; the next request in line will be executed once the one before it is complete.

Implementation:

first step: Enters the system it starts its execution immediately and runs till it completes its execution.

Second step: As other processes enter the system, they are put at the end of the queue and wait to get the CPU.

Third step: When a process finishes executing, it releases the CPU, is removed from the queue and the CPU is allocated to next process at the head of the queue.

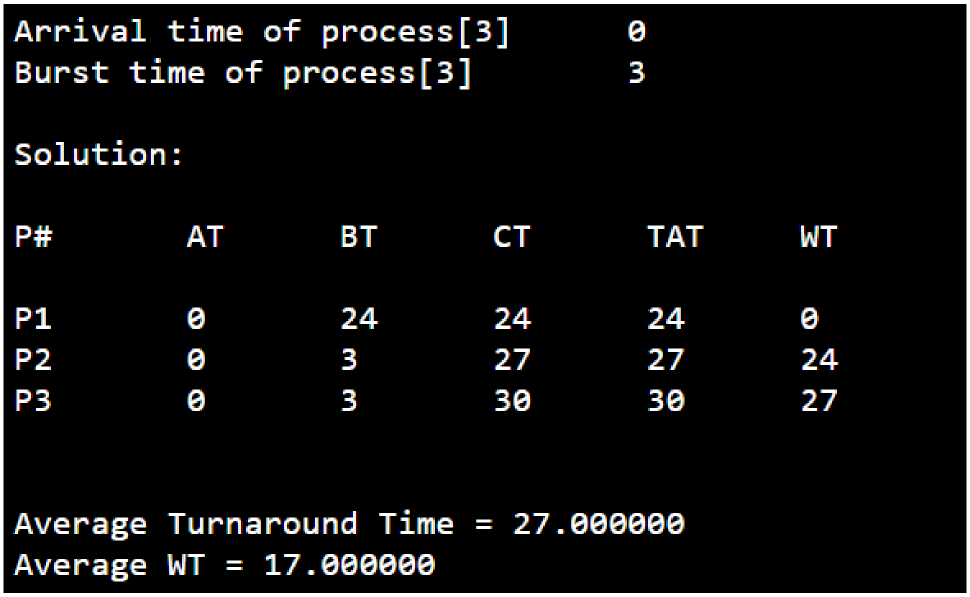
Working Process :

Code for FCFS Scheduling Algorithm –

```
#include<stdio.h>

int main(){

        int bt[10]={0},at[10]={0},tat[10]={0},wt[10]={0},ct[10]={0};
        int n,sum=0;
        float totalTAT=0,totalWT=0;

        printf("Enter number of processes          ");
        scanf("%d",&n);

        printf("Enter arrival time and burst time for each process\n\n");

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

                printf("Arrival time of process[%d]          ",i+1);
                scanf("%d",&at[i]);

                printf("Burst time of process[%d]            ",i+1);
                scanf("%d",&bt[i]);

                printf("\n");
        }

        //calculate completion time of processes

        for(int j=0;j<n;j++)
        {
                sum+=bt[j];
                ct[j]+=sum;
        }

        //calculate turnaround time and waiting times

        for(int k=0;k<n;k++)
        {
                tat[k]=ct[k]-at[k];
                totalTAT+=tat[k];
        }
```

```
        for(int k=0;k<n;k++)
        {
                wt[k]=tat[k]-bt[k];
                totalWT+=wt[k];
        }

        printf("Solution: \n\n");
        printf("P#\t AT\t BT\t CT\t TAT\t WT\t\n\n");

        for(int i=0;i<n;i++)
        {
                printf("P%d\t %d\t %d\t %d\t %d\t
%d\n",i+1,at[i],bt[i],ct[i],tat[i],wt[i]);
        }

        printf("\n\nAverage Turnaround Time = %f\n",totalTAT/n);
        printf("Average WT = %f\n\n",totalWT/n);

        return 0;
}
```

Output :



```
Arrival time of process[3]      0
Burst time of process[3]        3

Solution:

P#       AT       BT       CT       TAT      WT

P1       0        24       24       24       0
P2       0        3        27       27       24
P3       0        3        30       30       27


Average Turnaround Time = 27.000000
Average WT = 17.000000
```

Discussion :

In the above code, the demonstration of the first come first serve scheduling algorithm is shown. The user is asked to enter the number of processes. On entering the number of processes, we have to enter the burst times for each of the processes.

The waiting time is calculated first. First, the waiting time of the first process is zero.

```
for(i=1;i<n;i++)
 {
    wt[i]=0;
    for(j=0;j<i;j++)
       wt[i]+=bt[j];
}
```

Calculation of the waiting time is done by adding the burst time of the previous process.