



Mawlana Bhashani Science and Technology University

Lab-Report

Lab Report No: 07

Lab Report Name: Implementation of FCFS Scheduling Algorithm.

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance: 7/9/2020

Date of Submission:

Submitted by

Name: Afra Ibnat Tethye

ID: IT-18055

3rd year 1st semester

Session: 2017-18

Dept. of ICT

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

Experiment Name: Implementation of FCFS Scheduling Algorithm.

Question-01:

What is FCFS Scheduling algorithm?

Answer:

FCFS stands for 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.

Question-02:

How to implemented in C?

Answer:

Implemented code in C is given bellow:

```
#include<stdio.h>

using namespace std;

int main()
{
    int n ,burst[40],wait[30],turnaround[25],avgwait=0,avgturnaround=0,i,j;

    printf("Enter total number of processes(maximum 20):");

    scanf("%d",&n);

    printf("\nEnter Process Burst Time\n");

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

```

{

printf("P[%d]:",i+1);

scanf("%d",&burst[i]);

}


wait[0]=0;

for(i=1; i<n; i++)

{ wait[i]=0;

for(j=0; j<i; j++)

    wait[i]+=burst[j];

}


printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaroundTime");


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

{

turnaround[i]=burst[i]+wait[i];

avgwait+=wait[i];

avgturnaround+=turnaround[i];

    printf("\nP[%d]\t\t%d\t\t%d\t\t%d",i+1,burst[i],wait[i],turnaround[i]);

```

```

    }

    avgwait/=i;

    avgturnaround/=i;

    printf("\n\nAverage Waiting Time:%d",avgwait);

    printf("\n\nAverageTurnaround Time:%d",avgturnaround);

    return 0;

}

```

Output:

```

C:\Users\my\Documents\Untitled1.exe
Enter total number of processes(maximum 20):5
Enter Process Burst Time
P[1]:12
P[2]:18
P[3]:11
P[4]:11
P[5]:22

Process      Burst Time      Waiting Time      TurnaroundTime
P[1]         12              0                12
P[2]         18              12               30
P[3]         11              30               41
P[4]         11              41               52
P[5]         22              52               74

Average Waiting Time:27
AverageTurnaround Time:41
Process returned 0 (0x0)   execution time : 14.025 s
Press any key to continue.

```

Conclusion:

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 sum of the burst time and waiting time gives the turn-around time . The time the processes need to wait for it to get the CPU and start execution is called waiting time.