



Green University of Bangladesh

Dept. of CSE

Course Title: Operating System Lab

Lab Report-02

Submitted By:		Submitted To:	
Md. Tanvir Ahmed		Most. Rokeya Khatun)	
ID No : 191015102		Designation: Lecturer	
Batch No: ED		Dept. of CSE	
Semester : 8th		Green University Of Bangladesh	
Performance Date:6/18/2021		Submission Date: 6/25/2021	

Experiment No: 02

Experiment Name: Program for First Come First Serve Algorithm.

Introduction: The FCFS, which stands for First Come First Serve Scheduling Algorithm. The FCFS is a non-preemptive scheduling algorithm in which a process is automatically queued and processing occurs according to an incoming request or process order implementation of the FCFS policy is managed with a FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue.

Algorithm:

Step 1: Input the processes along with their arrival time (AT) and burst time (BT).

Step 2: Find completion time for all processes.

Step 3: calculate turnaround time for all processes.

Step 4: calculate waiting time for all processes.

Step 6: Find average waiting time = total waiting time/ no of process.

Step 7: Find average turnaround time = total turnaround time/no of process

Code:

```
#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");
    int i, j, k;
    for(i=1; i <= n ; i++) {
        printf("\nArrival Time of Process[%d]\t",i);
        scanf("%d",&at[i]);
        printf("Burst Time of Process[%d]\t",i);
        scanf("%d",&bt[i]);
    }
    ct[1] = bt[1];
    for(i=2; i <= n ; i++) {
        ct[i] = ct[i-1] + bt[i];
    }
    tat[1] = bt[1];
    totalTAT = totalTAT + tat[1];
    for(i=2; i <= n ; i++) {
```

```

    tat[i] = tat[i-1] + bt[i] - 1;
    totalTAT = totalTAT + tat[i];
}
wt[1] = 0;
for(i=2; i <= n ; i++) {
    wt[i] = tat[i-1] - 1;
    totalWT = totalWT + wt[i];
}
printf("Solution:\n\n");
printf("P#\tAT\tBT\tCT\tTAT\tWT\n\n");
for(i=1; i <= n; i++) {
    printf("P%d\t %d\t %d\t %d\t %d\t %d\n",i,at[i],bt[i],ct[i],tat[i],wt[i]);
}
printf("\nAverage Turnaround Time: %f\n",totalTAT/n);
printf("Average Waiting Time: %f\n",totalWT/n);
return 0;
}

```

Output:

```
D:\codeblocks\code\L_Report_02\bin\Debug\L_Report_02.exe
Enter Number of Processes: 5
Enter Arrival Time and Burst Time for Each Process

Arrival Time of Process[1]      0
Burst Time of Process[1]       3

Arrival Time of Process[2]      1
Burst Time of Process[2]       4

Arrival Time of Process[3]      2
Burst Time of Process[3]       6

Arrival Time of Process[4]      3
Burst Time of Process[4]       3

Arrival Time of Process[5]      4
Burst Time of Process[5]       2
Solution:

P#    AT    BT    CT    TAT   WT
P1     0     3     3     3     0
P2     1     4     7     6     2
P3     2     6    13    11     5
P4     3     3    16    13    10
P5     4     2    18    14    12

Average Turnaround Time: 9.400000
Average Waiting Time: 5.800000

Process returned 0 (0x0)   execution time : 25.792 s
Press any key to continue.
```

Discussion:

- (i) FCFS provides an efficient, simple and error-free process scheduling algorithm that saves valuable CPU resources.
- (ii) We have learnt different time calculation and its formula.
- (iii) We have learnt about FCFS Algorithm in detail.

(iv) Thus the FIFO process scheduling program was executed and verified successfully. After all this class was very interactive for us.

(v) FCFS provides an efficient, simple and error-free process scheduling algorithm that saves valuable CPU resources.

Thank You