Government Engineering College Thrissur

CS331 – System Software LabDocumentation
Exp1 – CPU Scheduling Algorithm

Date of Submission:21 August 2020

Submitted By,
Rejath T M
Roll No 50

TCR18CS050

GECT CSE S5

Experiment 1:

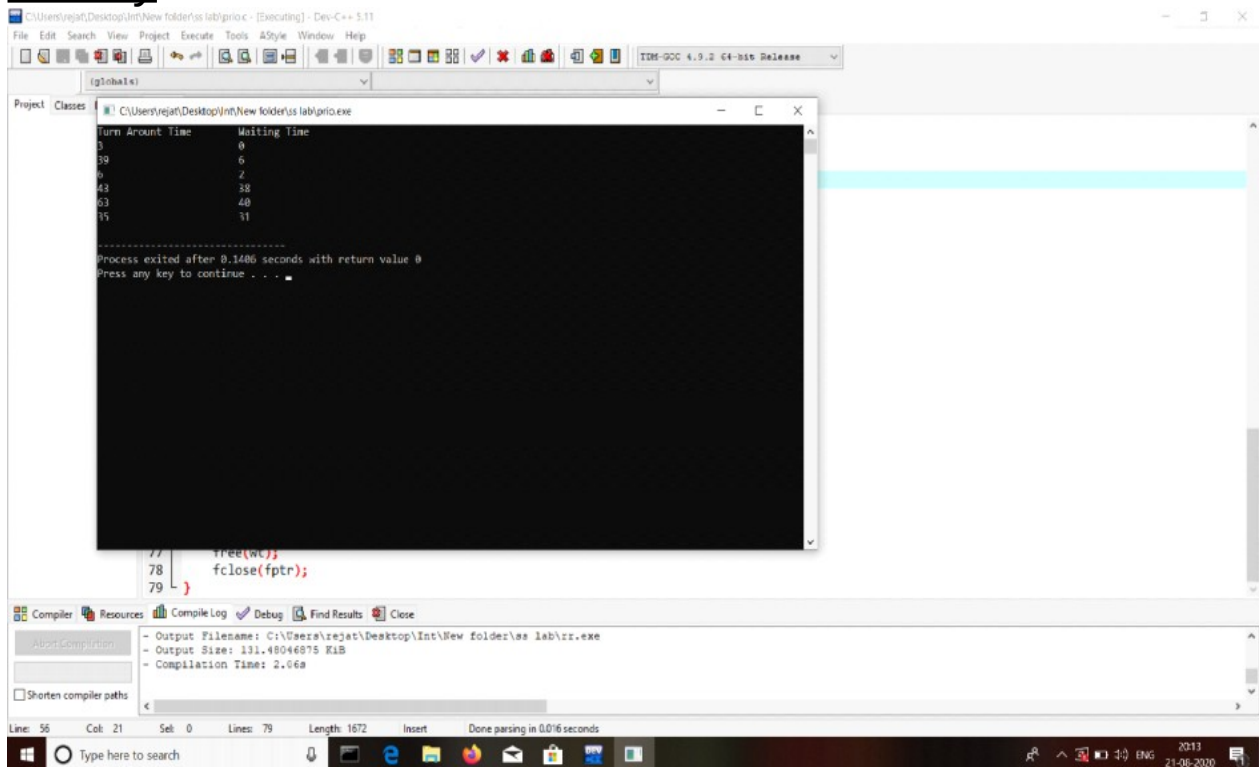Simulate the following non-preempitive CPU scheduling algorithms to find turnaround time and waiting time.

1. FCFS   2. SJF 3. Round Robin(pre-empitive) 4. Priority

Compilation of CodePrerequisite

- The code is provided in the **program.c** file along with its documentation. You just need to run it on any compiler.
- As output you will first see the input. If you want to change input the change in the **input.txt** file int format Number<tab>number<tab>number as AT,BT and Priority respectively.
- Out put of the file will be stored in the **output.txt** file as well as terminal or compiler in which it is executed.
-

Output Screenshots

Priority

Round Robin

Round Robin
Turn Around Time   Waiting Time
3   0
71   58
21   17
18   13
54   31
19   15

Process exited after 0.1451 seconds with return value 0
Press any key to continue . . .

```
        int a, b , c , i = 0;

        FILE *fptr;
        if ((fptr = fopen("input.txt", "r")) == NULL) {
            printf("Error! opening file");
            exit(1);
```


FCFS
Turn Around Time   Waiting Time
3   0
35   2
39   35
39   34
59   36
63   59

Process exited after 0.148 seconds with return value 0
Press any key to continue . . .

```
        findFcfsTime(n, at, bt, tat,wt);
        printf("FCFS\n");
        printf("Turn Around Time\tWaiting Time\n");
        for (i=0; i < n; i++){
            printf("%d\t\t%d\n", *(tat + i), *(wt + i));
        }
```

FCFS

SJF

C:\Users\rejaf\Desktop\int\New folder\ss lab\sjf.exe

```
SJF
Turn Arcount Time        Waiting Time
3                        0
71                       38
6                        2
6                        1
38                       7
7                        3

--------------------------------
Process exited after 0.1621 seconds with return value 0
Press any key to continue . . .
```

4.9.2 64-bit Release

```
42    while (!feof(fptr)){
43        fscanf(fptr, "%d\t%d\t%d", &a, &b, &c);
44        n++;
45    }
46    rewind(fptr);
47
48    at = (int*) malloc(n * sizeof(int));
49    bt = (int*) malloc(n * sizeof(int));
```

Type here to search