

# **Practice Assignment #**

## **02**

**FCFS**

# THEORY

## First Come First Served (FCFS) Algorithm

First-Come-First-Served 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.

## Priority based Scheduling Algorithm

In Priority based Scheduling, each process is assigned a priority. Process with highest priority is to be executed first and so on. Processes with same priority are executed on first come first serve basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

# QUESTION

Simulate the FCFS (First Come First Serve) Processor Scheduling Algorithm in C following the given steps. Also attach the print out of code and snapshot of output window. For compilation of C code, refer to steps discussed in Lab Session # 08.

- a) Define an array of structure process with members' process id, burst time, waiting time & turnaround time.
- b) Get length of the ready queue, i.e., number of process (say n). Also obtain the burst time for each process.
- c) The waiting time for first process is 0.
- d) Compute & display the waiting time and turnaround time for each process.
- e) Compute & display average waiting time 'awat' and average turnaround time 'atur'.
- f) Display GANTT chart for the above scheduling.
- g) Stop.

# Implementation and Algorithm

## **EX.NO: 3)iii)a CPU SCHEDULING: FIRST COME FIRST SERVE WITHOUT ARRIVAL TIME**

**AIM:** to write a c program to implement the first come first serve without arrival **TIME**  
CPU scheduling algorithm

### **PROBLEM DESCRIPTION:**

Cpu scheduler will decide which process should be given the CPU for its execution. For this it uses different algorithm to choose among the process. One among that algorithm is FCFS algorithm.

In this algorithm the process which arrive first is given the cpu after finishing its request only it will allow cpu to execute other process.

# Algorithm



## ALGORITHM:

- Step 1: Create the number of process.
- Step 2: Get the ID and Service time for each process.
- Step 3: Initially, Waiting time of first process is zero and Total time for the first process is the starting time of that process.
- Step 4: Calculate the Total time and Processing time for the remaining processes. Step 5: Waiting time of one process is the Total time of the previous process. Step 6: Total time of process is calculated by adding Waiting time and Service time.
- Step 7: Total waiting time is calculated by adding the waiting time for lack process. Step 8: Total turn around time is calculated by adding all total time of each process. Step 9: Calculate Average waiting time by dividing the total waiting time by total number of process.
- Step 10: Calculate Average turn around time by dividing the total time by the number of process.
- Step 11: Display the result.

# Implementation

## PROGRAM CODING:

```
#include<stdio.h> int
main()
{
    int n,b[10],t=0,i,w=0,r=0,a=0; float
    avg,avg1;
    printf("\nEnter number of processes:");
    scanf("%d",&n);
    printf("\nEnter the burst times : \n");
    for(i=1;i<=n;i++) scanf("%d",&b[i]);
    printf("\n Gantt chart ");
    for(i=1;i<=n;i++)
        printf("P%d\t",i);
    printf("\n\nProcess BurstTime WaitingTime TurnaroundTime\n");
    for(i=1;i<=n;i++)
```

```
{
    t=t+w;
    r=r+b[i];
    printf("P%d\t\t%d\t\t%d\t\t%d\t\t\n",i,b[i],w,r);
    w=w+b[i];
    a=a+r;
}
avg=(float)t/n;
avg1=(float)a/n;
printf("\n Average WaitingTime is %f",avg); printf("\n
Average TurnaroundTime is %f\n",avg1); return(0);}
```

Output

## OUTPUT:

```
[cse6@localhost Pgm]$ cc prog9a.c -o prog9a.out  
[cse6@localhost Pgm]$ ./prog9a.out
```

Enter number of processes : 3

Enter the burst times :

24

5

3

Gantt chart	P1	P2	P3
Process	BurstTime	WaitingTime	TurnaroundTime
P1	24	0	24
P2	5	24	29
P3	3	29	32

Average WaitingTime is 17.666666

Average TurnaroundTime is 28.333334

```
[cse6@localhost Pgm]$
```

## RESULT:

Thus the program is executed

# Task

## **EX.NO: 3)iii) b CPU SCHEDULING: FIRST COME FIRST SERVE WITH ARRIVAL TIME**

### **AIM:**

To write a C program to implement the array representation of the CPU scheduling algorithm first come first serve using arrival time.

### **PROBLEM DESCRIPTION:**

CPU scheduler will decide which process should be given the CPU for its execution. For this it uses different algorithm to choose among the process. One among that algorithm is FCFS algorithm.

In this algorithm the process which arrive first is given the cpu after finishing its request only it will allow cpu to execute other process



## ALGORITHM:

Step 1: Create the number of process.

Step 2: Get the ID and Service time for each process.

Step 3: Initially, Waiting time of first process is zero and Total time for the first process is the starting time of that process.

Step 4: Calculate the Total time and Processing time for the remaining processes.

Step 5: Waiting time of one process is the Total time of the previous process. Step 6:

Total time of process is calculated by adding Waiting time and Service time.

Step 7: Total waiting time is calculated by adding the waiting time for lack process.

Step 8: Total turn around time is calculated by adding all total time of each process.

Step 9: Calculate Average waiting time by dividing the total waiting time by total number of process.

Step 10: Calculate Average turn around time by dividing the total time by the number of process.

Step 11: Display the result.

## Attach Task Output