



## **UNIVERSITY OF ASIA PACIFIC**

**Course Title:** Operating System Lab

**Course Code:** CSE 406

**Date of Submission:** 21 October,2025

**Submitted by:**

Name: Md.Sarjil Hasan

Reg: 22101168

Sec : D

**Submitted to:**

Atia Rahman Orthi

Lecturer

University of Asia Pacific

# First come first serve(FCFS) job scheduling algorithm

## Problem Statement:

Given a set of processes, each with a specific arrival time and burst time, design a scheduling algorithm using First Come First Serve (FCFS). The goal is to compute for each process:

Process ID	Arrival Time	Burst Time
P1	0	3
P2	1	2
P3	2	1
P4	3	4

Completion Time (CT)

Turnaround Time (TAT) = CT – Arrival Time(AT)

Waiting Time (WT) = TAT – Burst Time(BT)

The algorithm should also compute and display the average waiting time and average turnaround time. Processes should be executed in the order of their arrival time.

## Objective:

- Apply FCFS algorithm for the given processes.
- Take process ID, arrival time and burst time as inputs.
- Work out the Completion Time, Turnaround Time and Waiting Time for the processes.
- Lastly also calculate the average waiting time and average turnaround time of the processes.

## Code:

```
fcfs.cpp > main()
3  int main() {
4      int n, i;
5      cout << "Enter the number of processes: ";
6      cin >> n;
7      int bt[n], wt[n], tat[n];
8      float avgwt = 0, avgtat = 0;
9      cout << "Enter the Burst Time for each process:\n";
10     for(i = 0; i < n; i++) {
11         cout << "P" << i+1 << ": ";
12         cin >> bt[i];
13     }
14     wt[0] = 0; // waiting time for first process is 0
15     for(i = 1; i < n; i++) {
16         wt[i] = 0;
17         for(int j = 0; j < i; j++)
18             wt[i] += bt[j];
19     }
20     cout << "\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n";
21     for(i = 0; i < n; i++) {
22         tat[i] = bt[i] + wt[i];
23         avgwt += wt[i];
24         avgtat += tat[i];
25         cout << "P" << i+1 << "\t" << bt[i] << "\t\t" << wt[i] << "\t\t" << tat[i] << endl;
26     }
27     avgwt /= n;
28     avgtat /= n;
29     cout << "\nAverage Waiting Time = " << avgwt;
30     cout << "\nAverage Turnaround Time = " << avgtat << endl;
31     return 0;
32 }
```

Process	Burst Time	Waiting Time	Turnaround Time
P1	3	0	3
P2	2	3	5
P3	1	5	6
P4	4	6	10

Average Waiting Time = 3.5

Average Turnaround Time = 6

PS C:\Users\Sarjil\Desktop\lab7&8>

## **Result:**

The FCFS CPU Scheduling Algorithm was implemented successfully. The program correctly calculated Waiting Time, Turnaround Time, and their averages.

## **Discussion:**

The FCFS algorithm is simple but can be inefficient when long processes arrive before short ones — this causes the convoy effect, increasing average waiting time. It's suitable for small, non-critical systems where simplicity is preferred over efficiency.

## **Conclusion:**

The experiment demonstrates how FCFS scheduling works based on process arrival order. While easy to implement, it may not always be optimal in terms of performance.

## **Git Link**

<https://github.com/Sarjil-SarZzz/CSE406-LAB/blob/main/cfs.cpp>