**Ex. No.: 6A Roll no:231901002**

**Date:04/03/2025**

**FIRST COME FIRST SERVE (FCFS)**

**Aim:**

To implement First-Come First-Serve (FCFS) scheduling technique.

**Algorithm:**

1. Start the program.
2. Input the number of processes.
3. Read the burst time for each process.
4. Calculate the waiting time for each process:
   * Waiting time of process 0 is 0.
   * For others:

WaitingTime[i] = WaitingTime[i-1] + BurstTime[i-1]

1. Calculate the turnaround time for each process: TurnAroundTime[i] = WaitingTime[i] + BurstTime[i]
2. Calculate the total and average waiting time and turnaround time.
3. Display process details, total and average times.
4. End.

**Program Code (in C):**

#include <stdio.h>

int main() { int n, i;

int burst\_time[20], waiting\_time[20], turn\_around\_time[20]; int total\_wt = 0, total\_tat = 0;

printf("Enter the number of process:\n"); scanf("%d", &n);

printf("Enter the burst time of the processes:\n"); for (i = 0; i < n; i++) {

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

}

waiting\_time[0] = 0;

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

waiting\_time[i] = waiting\_time[i - 1] + burst\_time[i - 1];

}

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

turn\_around\_time[i] = waiting\_time[i] + burst\_time[i]; total\_wt += waiting\_time[i]; total\_tat += turn\_around\_time[i];

}

printf("Process\tBurst Time\tWaiting Time\tTurn Around Time\n"); for (i = 0; i < n; i++) {

printf("%d\t%d\t\t%d\t\t%d\n", i, burst\_time[i], waiting\_time[i], turn\_around\_time[i]);

}

printf("Average Waiting Time is: %.1f\n", (float)total\_wt / n); printf("Average Turn Around Time is: %.1f\n", (float)total\_tat / n);

return 0;

}



**Sample Output:**

Enter the number of process:

3

Enter the burst time of the processes:

24 3 3

Process Burst Time Waiting Time Turn Around Time

| 0 | 24 | 0 | 24 |
| --- | --- | --- | --- |
| 1 | 3 | 24 | 27 |
| 2 | 3 | 27 | 30 |

Average Waiting Time is: 17.0

Average Turn Around Time is: 27.0



**Result:**

The FCFS Scheduling algorithm was successfully implemented. The program calculated the waiting time and turnaround time for each process and displayed the average times.