praveenkumar.R
192324012

Experiment-3: Design a CPU scheduling program with C using First Come First Served technique with the following considerations.
a. All processes are activated at time 0.
b. Assume that no process waits on I/O devices

Aim:

To simulate the CPU scheduling using the First Come First Served (FCFS) technique.

Procedure:

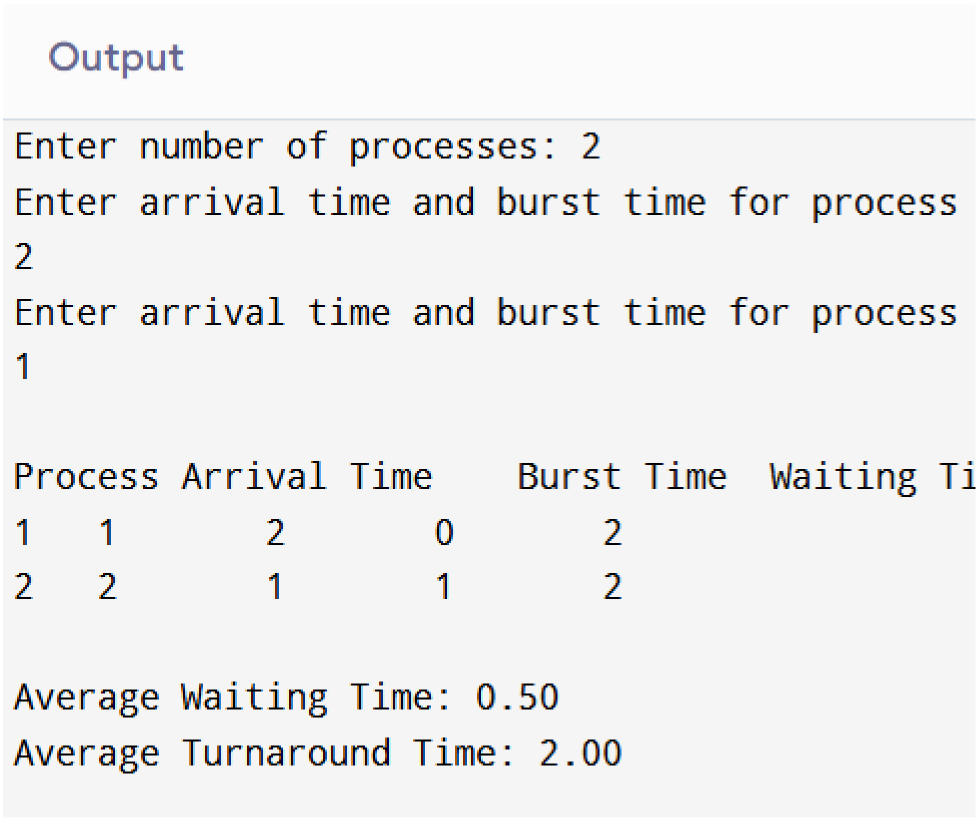
1. Input the arrival time and burst time for each process.
2. Calculate the completion time, turnaround time, and waiting time for each process.
3. Display the scheduling table.

C Program:

```
#include <stdio.h>


struct Process {

   int id;

   int arrival_time;

   int burst_time;

   int completion_time;

   int waiting_time;

   int turnaround_time;

};

int main() {

   int n;

   printf("Enter number of processes: ");

   scanf("%d", &n);


   struct Process processes[n];
```

```
    int total_waiting_time = 0, total_turnaround_time = 0;


    for (int i = 0; i < n; i++) {
        processes[i].id = i + 1;
        printf("Enter arrival time and burst time for process %d: ", i + 1);
        scanf("%d %d", &processes[i].arrival_time, &processes[i].burst_time);
    }
    processes[0].completion_time = processes[0].arrival_time + processes[0].burst_time;
    for (int i = 1; i < n; i++) {
        processes[i].completion_time = processes[i-1].completion_time + processes[i].burst_time;
    }
    for (int i = 0; i < n; i++) {
        processes[i].turnaround_time = processes[i].completion_time - processes[i].arrival_time;
        processes[i].waiting_time = processes[i].turnaround_time - processes[i].burst_time;
        total_waiting_time += processes[i].waiting_time;
        total_turnaround_time += processes[i].turnaround_time;
    }
    printf("\nProcess\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time\n");
    for (int i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", processes[i].id, processes[i].arrival_time,
processes[i].burst_time, processes[i].waiting_time, processes[i].turnaround_time);
    }
    printf("\nAverage Waiting Time: %.2f\n", (float)total_waiting_time / n);
    printf("Average Turnaround Time: %.2f\n", (float)total_turnaround_time / n);


    return 0;
}
```

Output:



```
Enter number of processes: 2
Enter arrival time and burst time for process
2
Enter arrival time and burst time for process
1

Process Arrival Time    Burst Time  Waiting Ti
1    1        2        0        2
2    2        1        1        2

Average Waiting Time: 0.50
Average Turnaround Time: 2.00
```