**Lab 5: CPU Scheduling - I**

1. **First Come First Serve (FCFS)**

#include<stdio.h>

int main() {

    int n, i, j, arrival\_time[20], burst\_time[20], waiting\_time[20], turnaround\_time[20], total\_waiting\_time = 0, total\_turnaround\_time = 0;

    float average\_waiting\_time, average\_turnaround\_time;

    printf("Enter the number of processes: ");

    scanf("%d", &n);

**// Loop to input the arrival time and burst time for each process**

    for (i = 0; i < n; i++) {

        printf("Enter Arrival time for process P%d : ", i + 1);

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

        printf("Enter Burst time for process P%d : ", i + 1);

        scanf("%d", &burst\_time[i]);

    }

**// Calculating Waiting time and Turnaround time for each process**

    waiting\_time[0] = 0;

    for(i = 1; i < n; i++) {

        waiting\_time[i] = 0;

        for(j = 0; j < i; j++) {

            waiting\_time[i] += burst\_time[j];

        }

        waiting\_time[i] -= arrival\_time[i];

    }

    for(i = 0; i < n; i++) {

        turnaround\_time[i] = burst\_time[i] + waiting\_time[i];

    }

**// Calculating Total Waiting time and Total Turnaround time**

    for(i = 0; i < n; i++) {

        total\_waiting\_time += waiting\_time[i];

        total\_turnaround\_time += turnaround\_time[i];

    }

**// Calculating Average Waiting time and Average Turnaround time**

    average\_waiting\_time = (float)total\_waiting\_time / n;

    average\_turnaround\_time = (float)total\_turnaround\_time / n;

**// Printing the Results**

    printf("\nProcess\t Arrival Time\t Burst Time\t Waiting Time\t Turnaround Time\n");

    for(i = 0; i < n; i++) {

        printf("P%d\t\t %d\t\t %d\t\t %d\t\t %d\n", i+1, arrival\_time[i], burst\_time[i],

waiting\_time[i], turnaround\_time[i]);

    }

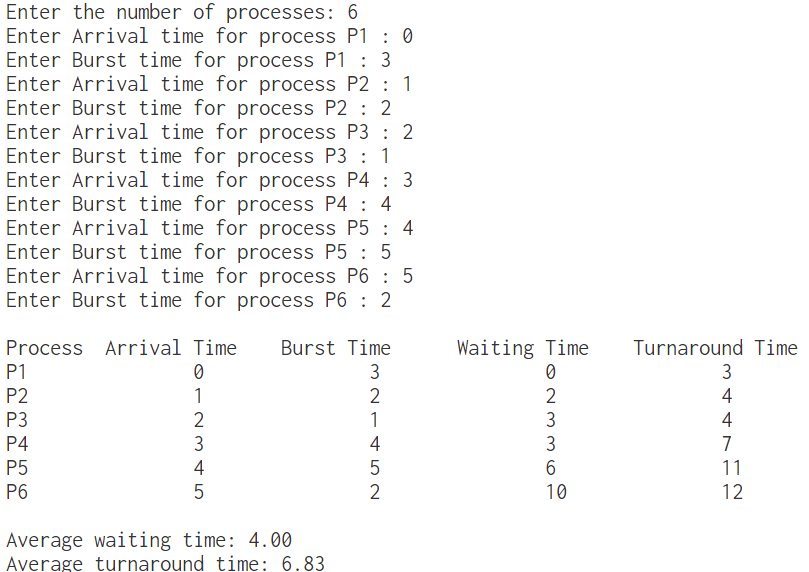
    printf("\nAverage waiting time: %.2f\n", average\_waiting\_time);

    printf("Average turnaround time: %.2f\n", average\_turnaround\_time);

    return 0;

}

* **Output**



1. **Shortest Job First**

#include <stdio.h>

int main() {

    int n, i, j, temp;

    float total\_waiting\_time = 0, total\_turnaround\_time = 0, average\_waiting\_time,

average\_turnaround\_time;

    printf("Enter the number of processes: ");

    scanf("%d", &n);

    int arrival\_time[n], burst\_time[n], waiting\_time[n], turnaround\_time[n], process[n];

**// Loop to input the arrival time and burst time for each process**

    for (i = 0; i < n; i++) {

        printf("Enter Arrival time for process P%d : ", i + 1);

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

        printf("Enter Burst time for process P%d : ", i + 1);

        scanf("%d", &burst\_time[i]);

        process[i] = i+1;

    }

**// Sorting processes by arrival time and burst time using selection sort**

    for(i = 0; i < n-1; i++) {

        for(j = i+1; j < n; j++) {

            if(arrival\_time[i] > arrival\_time[j] || (arrival\_time[i] == arrival\_time[j]

&& burst\_time[i] > burst\_time[j])) {

                temp = arrival\_time[i];

                arrival\_time[i] = arrival\_time[j];

                arrival\_time[j] = temp;

                temp = burst\_time[i];

                burst\_time[i] = burst\_time[j];

                burst\_time[j] = temp;

                temp = process[i];

                process[i] = process[j];

                process[j] = temp;

            }

        }

    }

**// Calculating waiting time and turnaround time**

    int current\_time = 0;

    for(i = 0; i < n; i++) {

        if(current\_time < arrival\_time[i]) {

            current\_time = arrival\_time[i];

        }

        waiting\_time[i] = current\_time - arrival\_time[i];

        current\_time += burst\_time[i];

        turnaround\_time[i] = current\_time - arrival\_time[i];

    }

**// Calculating total waiting time and total turnaround time**

    for(i = 0; i < n; i++) {

        total\_waiting\_time += waiting\_time[i];

        total\_turnaround\_time += turnaround\_time[i];

    }

**// Calculating average waiting time and average turnaround time**

    average\_waiting\_time = total\_waiting\_time / n;

    average\_turnaround\_time = total\_turnaround\_time / n;

**// Printing results**

    printf("Process\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time\n");

    for(i = 0; i < n; i++) {

        printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", process[i], arrival\_time[i], burst\_time[i],

waiting\_time[i], turnaround\_time[i]);

    }

    printf("Average waiting time: %f\n", average\_waiting\_time);

    printf("Average turnaround time: %f\n", average\_turnaround\_time);

    return 0;

}

* **Output**

