

**EXP 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.**

```
#include <stdio.h>
```

```
struct Process {
```

```
    int pid;          // Process ID
```

```
    int burst_time;   // CPU Burst Time
```

```
    int waiting_time; // Time process waits in ready queue
```

```
    int turnaround_time; // Time from arrival to completion
```

```
};
```

```
void calculate_times(struct Process p[], int n) {
```

```
    p[0].waiting_time = 0; // First process doesn't wait
```

```
    // Calculate waiting time for each process
```

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

```
        p[i].waiting_time = p[i-1].waiting_time + p[i-1].burst_time;
```

```
    }
```

```
    // Calculate turnaround time for each process
```

```

    for (int i = 0; i < n; i++) {
        p[i].turnaround_time = p[i].burst_time + p[i].waiting_time;
    }
}

void display(struct Process p[], int n) {
    float total_waiting = 0, total_turnaround = 0;

    printf("\nPID\tBurst\tWaiting\tTurnaround\n");
    for (int i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\n", p[i].pid, p[i].burst_time,
            p[i].waiting_time, p[i].turnaround_time);

        total_waiting += p[i].waiting_time;
        total_turnaround += p[i].turnaround_time;
    }

    printf("\nAverage Waiting Time: %.2f", total_waiting / n);
    printf("\nAverage Turnaround Time: %.2f\n", total_turnaround / n);
}

int main() {
    int n;

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

```

```

struct Process p[n];

// Input burst times
for (int i = 0; i < n; i++) {
    p[i].pid = i + 1;
    printf("Enter burst time for Process P%d: ", p[i].pid);
    scanf("%d", &p[i].burst_time);
}

// FCFS Scheduling
calculate_times(p, n);

// Display results
display(p, n);

return 0;
}

```

### **Sample Input and Output:**

```

Enter the number of processes: 5
Enter burst time for Process P1: 7
Enter burst time for Process P2: 8
Enter burst time for Process P3: 5
Enter burst time for Process P4: 7
Enter burst time for Process P5: 8

```

PID	Burst	Waiting	Turnaround
P1	7	0	7
P2	8	7	15
P3	5	15	20
P4	7	20	27
P5	8	27	35

```

Average Waiting Time: 13.80
Average Turnaround Time: 20.80

```