

Name	Stephen David Vaz
UID	2021700070
Branch	S.E. CSE DS
Experiment	5
Subject	OS
Aim	PREEMTIVE AND NONPREEMTIVE PROCESS SHEDULING

```
{
    avgW += wt[i];
    avgT += bt[i] + wt[i];
}
printf("Avg\t|\t      |\t   %d\t      |\t%d\n", avgW / n, avgT / n);
return 0;
}
```

Result:

```
students@lenovo-ThinkCentre-neo-50s-Gen-3:~/code$ gcc -o fcfs fcfs.c && ./fcfs
No. of processes: 3
Burst Time for P1: 5
Burst Time for P2: 11
Burst Time for P3: 11


| Process | Burst Time | Waiting time | Turnaround time |
|---------|------------|--------------|-----------------|
| P1      | 5          | 0            | 5               |
| P2      | 11         | 5            | 16              |
| P3      | 11         | 16           | 27              |
| Avg     |            | 7            | 16              |


students@lenovo-ThinkCentre-neo-50s-Gen-3:~/code$
```

First Come First Serve (FCFS) with Wait Time

Program:

[illegible]

```

    int bith = 0;
    for (int j = 0; j < i; j++)
    {
        bith += bt[j];
    }

    wt[i] = bith - at[i];
    printf("P%d\t|\t%d\t|\t%d\t\t|\t\t%d\t\t\t|\t%d\n", i + 1, at[i], bt[i],
wt[i], bt[i] + wt[i]);
    avgW += wt[i];
    avgT += bt[i] + wt[i];
}
printf("Avg\t|\t\t\t|\t\t\t\t|\t\t%.2f\t\t\t|\t%.2f\t\n", avgW / n, avgT / n);
return 0;
}

```

Result:

```

* Executing task: /usr/bin/clang /Users/stephen03/Dev/repos/stepDS/osLab/
afcfs.c -o ./excs/afcfs && ./excs/afcfs

No. of processes: 3
Burst and Arrival Time for P1: 5 0
Burst and Arrival Time for P2: 9 3
Burst and Arrival Time for P3: 6 6

```

Process	Arrival Time	Burst Time	Waiting time	Turnaround time
P1	0	5	0	5
P2	3	9	2	11
P3	6	6	8	14
Avg			3.33	10.00

```

* Terminal will be reused by tasks, press any key to close it.

```

Round Robin

Program:

```

#include <stdio.h>

int main()
{
    int i, NOP, sum = 0, count = 0, y, quant, wt = 0, tat = 0, at[10], bt[10],
temp[10];
    float avg_wt, avg_tat;
    printf("No. of processes: ");
    scanf("%d", &NOP);
    NOP = 4;
    y = NOP;
    printf("Time Quantum: ");
    scanf("%d", &quant);
    for (i = 0; i < NOP; i++)
    {

```

```

printf("Burst and Arrival Time for P%d: ", i + 1);
scanf("%d %d", &bt[i], &at[i]);

temp[i] = bt[i];
}
printf("\nProcess Arrival Time Burst Time TAT Waiting Time ");
for (sum = 0, i = 0; y != 0;)
{
    if (temp[i] <= quant && temp[i] > 0)
    {
        sum = sum + temp[i];
        temp[i] = 0;
        count = 1;
    }
    else if (temp[i] > 0)
    {
        temp[i] = temp[i] - quant;
        sum = sum + quant;
    }
    if (temp[i] == 0 && count == 1)
    {
        y--;
        printf("\nP%d\t\t%d \t%d\t%d\t %d", i + 1, at[i], bt[i], sum - at[i],
sum - at[i] - bt[i]);
        wt = wt + sum - at[i] - bt[i];
        tat = tat + sum - at[i];
        count = 0;
    }
    if (i == NOP - 1)
    {
        i = 0;
    }
    else if (at[i + 1] <= sum)
    {
        i++;
    }
    else
    {
        i = 0;
    }
}
avg_wt = wt * 1.0 / NOP;
avg_tat = tat * 1.0 / NOP;
printf("\nAverage Turn Around Time: %2f", avg_wt);
printf("\nAverage Waiting Time: %2f\n", avg_tat);

```

Result:

```
* Executing task: /usr/bin/clang /Users/stephen03/Dev/repos/stepDS/osLab/rr.c -o ./excs/rr && ./excs/rr

No. of processes: 4
Time Quantum: 6
Burst and Arrival Time for P1: 8 0
Burst and Arrival Time for P2: 5 1
Burst and Arrival Time for P3: 10 2
Burst and Arrival Time for P4: 11 3

Process Arrival Time Burst Time TAT Waiting Time
P2          1         5      10         5
P1          0         8      25        17
P3          2        10      27        17
P4          3        11      31        20
Average Turn Around Time: 14.750000
Average Waiting Time: 23.250000
* Terminal will be reused by tasks, press any key to close it.
```

Conclusion: Successfully understood various scheduling algorithms by simulating FCFS and Round Robin in C. Also, understood the calculations of Waiting time and Turn Around time along with their importance in process scheduling.