Name: Sahin Nayak **Enrollment Number : 12023006015086**

ASSIGNMENT-3

1. Write a C programme to simulate the following non-preemptive CPU scheduling algorithms to find the turnaround time and waiting time for the above problem. A. FCFS B. SJF C. Priority

A. FCFS cpu scheduling algorithm.. a. For the FCFS scheduling algorithm, read the number of processes/jobs in the system, and their CPU burst times. b. The scheduling is performed based on the arrival time of the processes, irrespective of their other parameters. c. Each process will be executed according to its arrival time. . d. Each process will be executed according to its arrival time.

B. SJF cpu scheduling algorithm.. a. For the SJF scheduling algorithm, read the number of processes/jobs in the system, and their CPU burst times. b. Arrange all the jobs in order with respect to their burst times. c. Two jobs may be in queue with the same execution time, and then the FCFS approach will be performed. d. Each process will be executed according to the length of its burst time. e. Then calculate each process's waiting time and turnaround time accordingly.

C. PRIORITY cpu scheduling algorithm.. a. For the priority scheduling algorithm, read the number of processes/jobs in the system, their CPU burst times, and the priorities. b. Arrange all the jobs in order with respect to their priorities. c. There may be two jobs in queue with the same priority, and then FCFS approach will be performed. d. Each process will be executed according to its priority. e. Calculate the waiting time and turnaround time of each of the processes accordingly.

```
#include <stdio.h>
struct Process {
   int id;
   int burst_time;
   int arrival_time;
   int priority;
   int waiting_time;
   int turnaround_time; };
void sortByArrivalTime(struct Process proc[], int n) {
   for (int i = 0; i < n - 1; i++) {
      for (int j = 0; j < n - i - 1; j++) {
         if (proc[j].arrival_time > proc[j + 1].arrival_time) {
            struct Process temp = proc[j];
            proc[j] = proc[j + 1];
            proc[j + 1] = temp; }}}}
void sortByBurstTime(struct Process proc[], int n) {
   for (int i = 0; i < n - 1; i++) {
      for (int j = 0; j < n - i - 1; j++) {
         if (proc[j].burst_time > proc[j + 1].burst_time) {
            struct Process temp = proc[j];
            proc[j] = proc[j + 1];
            proc[j + 1] = temp;}}}}
void sortByPriority(struct Process proc[], int n) {
   for (int i = 0; i < n - 1; i++) {
      for (int j = 0; j < n - i - 1; j++) {
         if (proc[j].priority > proc[j + 1].priority) {
            struct Process temp = proc[j];
            proc[j] = proc[j + 1];
            proc[j + 1] = temp; }}}}
void calculateTimes(struct Process proc[], int n) {
   int total_waiting_time = 0, total_turnaround_time = 0;
      proc[0].waiting_time = 0;
   for (int i = 1; i < n; i++) {
      proc[i].waiting_time = proc[i - 1].waiting_time + proc[i - 1].burst_time; }
   for (int i = 0; i < n; i++) {
      proc[i].turnaround_time = proc[i].waiting_time + proc[i].burst_time;
      total_waiting_time += proc[i].waiting_time;
      total_turnaround_time += proc[i].turnaround_time;}
   printf("Process ID | Burst Time | Arrival Time | Priority | Waiting Time | Turnaround Time\n");
   for (int i = 0; i < n; i++) {
      printf("  %d     |     %d     |     %d     |     %d     |     %d     |     %d\n", proc[i].id, proc[i].burst_time, proc[i].arrival_time, proc[i].priority,
proc[i].waiting_time, proc[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);}
int main() {
   int n, choice;
   printf("Enter the number of processes: ");
   scanf("%d", &n);
   struct Process proc[n];
   for (int i = 0; i < n; i++) {
      printf("Enter burst time for process %d: ", i + 1);
```

```
    scanf("%d", &proc[i].burst_time);
    printf("Enter arrival time for process %d: ", i + 1);
    scanf("%d", &proc[i].arrival_time);
    printf("Enter priority for process %d: ", i + 1);
    scanf("%d", &proc[i].priority);
    proc[i].id = i + 1;}
printf("\nChoose the scheduling algorithm:\n");
printf("1. First Come First Serve (FCFS)\n");
printf("2. Shortest Job First (SJF)\n");
printf("3. Priority\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice) {
    case 1:
        sortByArrivalTime(proc, n);
        printf("\nFCFS Scheduling\n");
        break;
    case 2:
        sortByBurstTime(proc, n);
        printf("\nSJF Scheduling\n");
        break;
    case 3:
        sortByPriority(proc, n);
        printf("\nPriority Scheduling\n");
        break;
    default:
        printf("Invalid choice\n");
        return 1;}
calculateTimes(proc, n);
return 0;}
```

```
sahin@sahin-VirtualBox:~$ gedit cpuScheduling.c
sahin@sahin-VirtualBox:~$ gcc cpuScheduling.c -o cpuScheduling
sahin@sahin-VirtualBox:~$ ./cpuScheduling
Enter the number of processes: 3
Enter burst time for process 1: 5
Enter arrival time for process 1: 0
Enter priority for process 1: 2
Enter burst time for process 2: 3
Enter arrival time for process 2: 1
Enter priority for process 2: 1
Enter burst time for process 3: 8
Enter arrival time for process 3: 2
Enter priority for process 3: 3

Choose the scheduling algorithm:
1. First Come First Serve (FCFS)
2. Shortest Job First (SJF)
3. Priority
Enter your choice: 1

FCFS Scheduling
Process ID | Burst Time | Arrival Time | Priority | Waiting Time | Turnaround Time
    1      |     5      |      0       |    2     |      0       |       5
    2      |     3      |      1       |    1     |      5       |       8
    3      |     8      |      2       |    3     |      8       |       16

Average Waiting Time: 4.33
Average Turnaround Time: 9.67
```

```
sahin@sahin-VirtualBox:~$ ./cpuScheduling
Enter the number of processes: 3
Enter burst time for process 1: 6
Enter arrival time for process 1: 0
Enter priority for process 1: 3
Enter burst time for process 2: 2
Enter arrival time for process 2: 1
Enter priority for process 2: 2
Enter burst time for process 3: 8
Enter arrival time for process 3: 2
Enter priority for process 3: 1

Choose the scheduling algorithm:
1. First Come First Serve (FCFS)
2. Shortest Job First (SJF)
3. Priority
Enter your choice: 2

SJF Scheduling
Process ID | Burst Time | Arrival Time | Priority | Waiting Time | Turnaround Time
    2      |     2      |      1       |    2     |      0       |       2
    1      |     6      |      0       |    3     |      2       |       8
    3      |     8      |      2       |    1     |      8       |       16

Average Waiting Time: 3.33
Average Turnaround Time: 8.67
sahin@sahin-VirtualBox:~$
```

```
sahin@sahin-VirtualBox:~$ ./cpuScheduling
Enter the number of processes: 3
Enter burst time for process 1: 4
Enter arrival time for process 1: 0
Enter priority for process 1: 3
Enter burst time for process 2: 5
Enter arrival time for process 2: 1
Enter priority for process 2: 2
Enter burst time for process 3: 2
Enter arrival time for process 3: 2
Enter priority for process 3: 1

Choose the scheduling algorithm:
1. First Come First Serve (FCFS)
2. Shortest Job First (SJF)
3. Priority
Enter your choice: 3

Priority Scheduling
Process ID | Burst Time | Arrival Time | Priority | Waiting Time | Turnaround Time
    3      |     2      |      2       |    1     |      0       |       2
    2      |     5      |      1       |    2     |      2       |       7
    1      |     4      |      0       |    3     |      7       |       11

Average Waiting Time: 3.00
Average Turnaround Time: 6.67
sahin@sahin-VirtualBox:~$
```