“lab 11”

COURSE :

OPERATING SYSTEM

SUBMITTED TO :

Sir shehzad

SUBMITTED BY :

Rabia Batool (2022-BSE-064)

SECTION :

B

LAB TASK

- Implement Shortest Job First (Non-Preemptive) CPU Scheduling Algorithm.

Code:

```
#include <stdio.h>

struct process
{
   char name[10];
   int burst_time;
   int arrival_time;
   int waiting_time;
   int turnaround_time;
};

int main()
{
   struct process p[3];

   printf("Enter name, burst time, and arrival time of the processes\n");
   for (int i = 0; i < 3; i++)
   {
      printf("Name of process p%d: ", i);
      scanf("%s", p[i].name);
      printf("Burst time of p%d: ", i);
      scanf("%d", &p[i].burst_time);
      printf("Arrival time of p%d: ", i);
      scanf("%d", &p[i].arrival_time);
   }

   printf("Processes after sorting according to shortest job first\n");
   for (int i = 0; i < 3; i++)
   {
      for (int j = 0; j < 2; j++)
      {
         if (p[j].burst_time > p[j + 1].burst_time)
         {
            // Swap
```

```
            struct process temp = p[j];
            p[j] = p[j + 1];
            p[j + 1] = temp;
          }
        }
    }

    printf("PROCESS NAME   BURST TIME   ARRIVAL TIME\n");
    for (int i = 0; i < 3; i++)
    {
        printf("%s          %d          %d\n", p[i].name, p[i].burst_time, p[i].arrival_time);
    }

    p[0].waiting_time = 0;
    p[1].waiting_time = p[0].burst_time;
    p[2].waiting_time = p[0].burst_time + p[1].burst_time;
    p[0].turnaround_time = p[0].burst_time;
    p[1].turnaround_time = p[0].burst_time + p[1].burst_time;
    p[2].turnaround_time = p[0].burst_time + p[1].burst_time + p[2].burst_time;

    printf("\nImplementation of Shortest Job First Algorithm\n");
    printf("PROCESS NAME   BURST TIME   ARRIVAL TIME   WAITING TIME   TURNAROUND TIME\n");
    for (int i = 0; i < 3; i++)
    {
        printf("%s          %d          %d          %d          %d\n", p[i].name, p[i].burst_time, p[i].arrival_time,
p[i].waiting_time, p[i].turnaround_time);
    }

    int temp = 0, temp1 = 0;
    for (int i = 0; i < 3; i++)
    {
        temp1 += p[i].turnaround_time;
        temp += p[i].waiting_time;
    }

    printf("\nAverage Waiting Time of the Processes = %d\n", temp / 3);
    printf("Average Turnaround Time of the Processes = %d\n", temp1 / 3);

    return 0;
}
```

Output:

```
Average Turnaround Time of the Processes = 0
~$ gcc lab.c
~$ ./a.out
Enter name, burst time, and arrival time of the processes
Name of process p0: p1
Burst time of p0: 24
Arrival time of p0: 0
Name of process p1: p2
Burst time of p1: 3
Arrival time of p1: 0
Name of process p2: p3
Burst time of p2: 2
Arrival time of p2: 0
Processes after sorting according to shortest job first
PROCESS NAME   BURST TIME   ARRIVAL TIME
p3             2            0
p2             3            0
p1             24            0

Implementation of Shortest Job First Algorithm
PROCESS NAME   BURST TIME   ARRIVAL TIME   WAITING TIME   TURNAROUND TIME
p3             2            0            0             2
p2             3            0            2             5
p1             24            0            5             29

Average Waiting Time of the Processes = 2
Average Turnaround Time of the Processes = 12
```