CSE321

OPERATING SYSTEMS

LAB ASSIGNMENT 04

SHAIANE PREMA BAROI

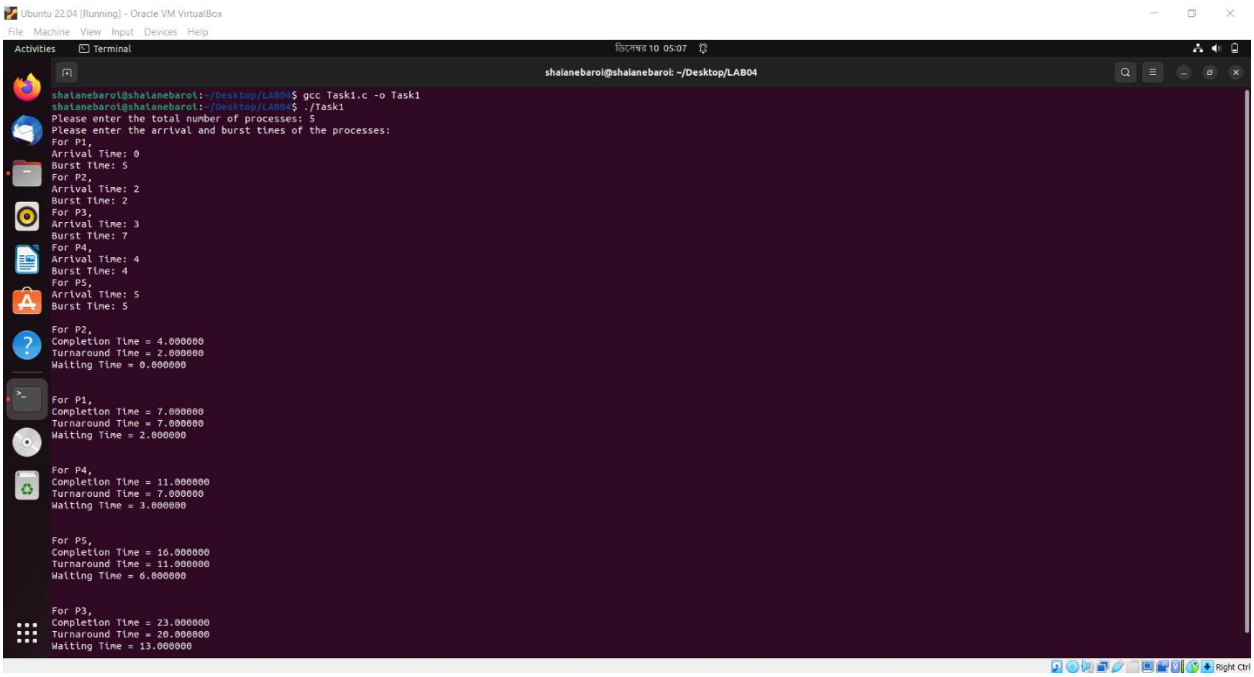
ID: 21101098

SECTION: 07

TASK1

```
#include <stdio.h>

int main() {

    //TOTAL NUMBER OF PROCESSES
    int num;
    printf("Please enter the total number of processes: ");
    scanf("%d", &num);

    //ARRIVAL AND BURST TIMES OF THE PROCESSES
    int arrival_time[num], burst_time[num + 1], temp[num + 1];
    printf("Please enter the arrival and burst times of the
processes: \n");
    for(int count = 0; count < num; count++) {
        printf ("For P%d, \n", count+1);

        //ARRIVAL TIMES
        printf("Arrival Time: ");
        scanf("%d", &arrival_time[count]);

        //BURST TIMES
        printf("Burst Time: ");
        scanf("%d", &burst_time[count]);

        //COPYING BURST TIMES IN A TEMP ARRAY
        temp[count] = burst_time[count];
    }

    int array_size = sizeof(burst_time)/sizeof(burst_time[0]);

    double sum_waiting_time = 0, sum_turnaround_time = 0,
completion_time = 0;
    float average_waiting_time, average_turnaround_time;

    int count = 0;
    burst_time[array_size - 1] = 9999;
    for(int time = 0; count != num; time++) {
        //FINDING THE SHORTEST JOB
        int smallest = array_size - 1;
        for(int count = 0; count <= num; count++) {
              if(arrival_time[count] <= time &&
burst_time[count] < burst_time[smallest] && burst_time[count] >
0) {
                    smallest = count;
              }
```

```
        }

        //GANTT CHART
        //printf("P%d \n", smallest+1);

        burst_time[smallest]--;

        //WHEN A PROCESS COMPLETES
        if (burst_time[smallest] == 0) {

            printf ("\nFor P%d,", smallest + 1);
            count++;

            completion_time = time + 1;
            printf("\nCompletion Time = %lf", completion_time);

            double turnaround_time = completion_time -
arrival_time[smallest];
            printf("\nTurnaround Time = %lf", turnaround_time);
            sum_turnaround_time += turnaround_time;

            double waiting_time = turnaround_time -
temp[smallest];
            printf("\nWaiting Time = %lf \n\n", waiting_time);
            sum_waiting_time += waiting_time;
        }
    }
    /*
    //AVERAGE WAITING TIME
    average_waiting_time = sum_waiting_time / num;
    printf("\nAverage Waiting Time: %lf",
average_waiting_time);

    //AVERAGE TURNAROUND TIME
    average_turnaround_time = sum_turnaround_time / num;
    printf("\nAverage Turnaround Time: %lf",
average_turnaround_time);
    */
    return 0;
}
```

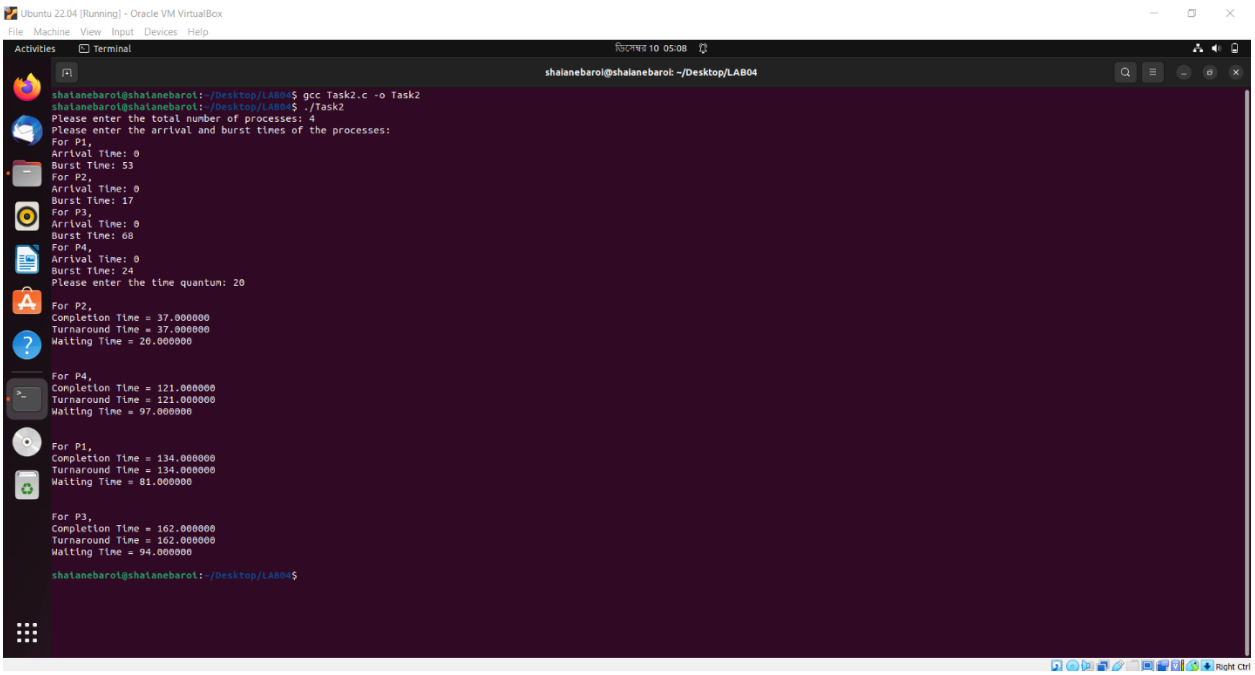

Ubuntu 22.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
ডিসেম্বর 10 05:07
shaianebarol@shaianebarol: ~/Desktop/LAB04
shaianebarol@shaianebarol:~/Desktop/LAB04$ gcc Task1.c -o Task1
shaianebarol@shaianebarol:~/Desktop/LAB04$ ./Task1
Please enter the total number of processes: 5
Please enter the arrival and burst times of the processes:
For P1,
Arrival Time: 0
Burst Time: 5
For P2,
Arrival Time: 2
Burst Time: 2
For P3,
Arrival Time: 3
Burst Time: 7
For P4,
Arrival Time: 4
Burst Time: 4
For P5,
Arrival Time: 5
Burst Time: 5
For P2,
Completion Time = 4.000000
Turnaround Time = 2.000000
Waiting Time = 0.000000
For P1,
Completion Time = 7.000000
Turnaround Time = 7.000000
Waiting Time = 2.000000
For P4,
Completion Time = 11.000000
Turnaround Time = 7.000000
Waiting Time = 3.000000
For P5,
Completion Time = 16.000000
Turnaround Time = 11.000000
Waiting Time = 6.000000
For P3,
Completion Time = 23.000000
Turnaround Time = 20.000000
Waiting Time = 13.000000
Right Ctrl

TASK2

```
#include<stdio.h>

int main() {

    //TOTAL NUMBER OF PROCESSES
    int num;
    int arrival_time[10], burst_time[10], temp[10];
    printf("Please enter the total number of processes: ");
    scanf("%d", &num);

    //COPYTING TOTAL NUMBER OF PROCESSES IN ANOTHER VARIABLE
    int num2 = num;

    //ARRIVAL AND BURST TIMES OF THE PROCESSES
    printf("Please enter the arrival and burst times of the
processes: \n");
    for(int count = 0; count < num; count++) {
        printf ("For P%d, \n", count+1);

        //ARRIVAL TIMES
        printf("Arrival Time: ");
        scanf("%d", &arrival_time[count]);

        //BURST TIMES
        printf("Burst Time: ");
        scanf("%d", &burst_time[count]);

        //COPYING BURST TIMES IN A TEMP ARRAY
        temp[count] = burst_time[count];
    }

    //TIME QUANTUM
    int time_quantum;
    printf("Please enter the time quantum: ");
    scanf("%d", &time_quantum);



    int sum_waiting_time, sum_turnaround_time;
    float average_waiting_time, average_turnaround_time;

    int time = 0;
    int count = 0;
    int status;
```

```
    while (num2 != 0) {
        //WHEN BURST TIME IS LESS THAN TIME QUANTUM
        if(burst_time[count] <= time_quantum &&
burst_time[count] > 0) {
              time += burst_time[count];
              burst_time[count] = 0;
              status = 1;
              //printf("At time = %d, \n", time);
              //printf("P%d \n", count+1);
        }

        //WHEN BURST TIME IS GREATER THAN TIME QUANTUM
        else if(burst_time[count] > time_quantum &&
burst_time[count] > 0) {
              burst_time[count] = burst_time[count] -
time_quantum;
              time += time_quantum;
              //printf("At time = %d, \n", time);
              //printf("P%d \n", count+1);
        }

        //WHEN A PROCESS COMPLETES
        if (burst_time[count] == 0 && status == 1) {
          printf("\nFor P%d,", count + 1);
            num2--;

            double completion_time = time;
            printf("\nCompletion Time = %lf", completion_time);

            double turnaround_time = completion_time -
arrival_time[count];
            printf("\nTurnaround Time = %lf", turnaround_time);
            sum_turnaround_time += turnaround_time;

            double waiting_time = turnaround_time -
temp[count];
            printf("\nWaiting Time = %lf \n\n", waiting_time);
            sum_waiting_time += waiting_time;

            status = 0;
        }

        //RETURNING TO THE FIRST PROCESS AFTER LAST PROCESS IS
EXECUTED
        if(count == num - 1) {
              count = 0;
        }
```

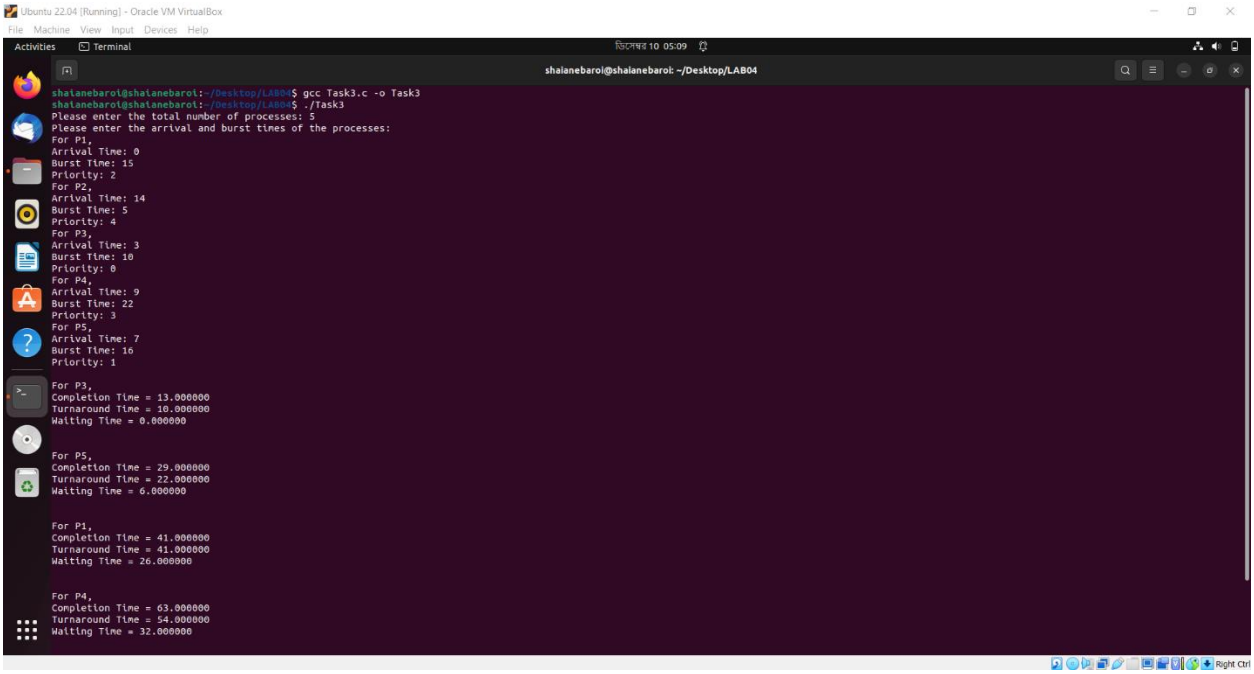
```
        //EXECUTING THE PROCESSES SERIALLY UNTIL LAST PROCESS
IS EXECUTED
        else if(arrival_time[count + 1] <= time) {
            count++;
        }

        //FOR ALL OTHER CASES
        else {
            count = 0;
        }
    }
    /*
    //AVERAGE WAITING TIME
    average_waiting_time = sum_waiting_time / num;
    printf("\nAverage Waiting Time: %lf",
average_waiting_time);

    //AVERAGE TURNAROUND TIME
    average_turnaround_time = sum_turnaround_time / num;
    printf("\nAverage Turnaround Time: %lf",
average_turnaround_time);
    */
    return 0;
}
```

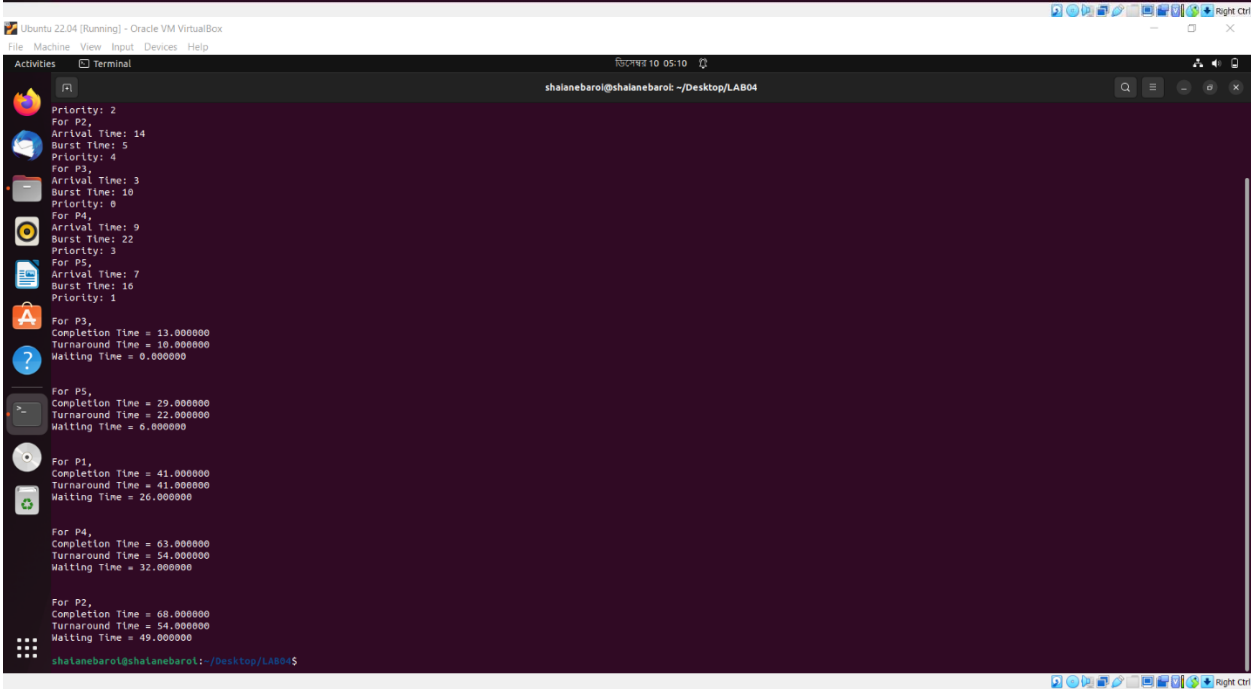


TASK3

```
#include <stdio.h>

int main() {

    //TOTAL NUMBER OF PROCESSES
    int num;
    printf("Please enter the total number of processes: ");
    scanf("%d", &num);

    //ARRIVAL TIMES, BURST TIMES AND PRIORITY OF THE PROCESSES
    int arrival_time[num], burst_time[num], temp[num],
priority[num+1];
    printf("Please enter the arrival and burst times of the
processes: \n");
    for(int count = 0; count < num; count++) {
        printf ("For P%d, \n", count+1);

        //ARRIVAL TIMES
        printf("Arrival Time: ");
        scanf("%d", &arrival_time[count]);

        //BURST TIMES
        printf("Burst Time: ");
        scanf("%d", &burst_time[count]);

        //COPYING BURST TIMES IN A TEMP ARRAY
        temp[count] = burst_time[count];

        //PRIORITY
        printf("Priority: ");
        scanf("%d", &priority[count]);
    }

    int array_size = sizeof(priority)/sizeof(priority[0]);

    double sum_waiting_time = 0, sum_turnaround_time = 0,
completion_time = 0;
    float average_waiting_time, average_turnaround_time;

    int count = 0;
    priority[array_size - 1] = num + 1;
    for(int time = 0; count != num; time++) {
        //FINDING THE SMALLEST(HIGHEST) PRIORITY
        int smallest = array_size - 1;
        for(int count = 0; count <= num; count++) {
```

```
            if(arrival_time[count] <= time && priority[count]
< priority[smallest] && burst_time[count] > 0) {
                    smallest = count;
            }
        }

        //GANTT CHART
        //printf("P%d \n", smallest+1);

        burst_time[smallest]--;

        //WHEN A PROCESS COMPLETES
        if (burst_time[smallest] == 0) {

            printf ("\nFor P%d,", smallest+1);
            count++;

            completion_time = time + 1;
            printf("\nCompletion Time = %lf", completion_time);

            double turnaround_time = completion_time -
arrival_time[smallest];
            printf("\nTurnaround Time = %lf", turnaround_time);
            sum_turnaround_time += turnaround_time;

            double waiting_time = turnaround_time -
temp[smallest];
            printf("\nWaiting Time = %lf \n\n", waiting_time);
            sum_waiting_time += waiting_time;
        }
    }
    /*
    //AVERAGE WAITING TIME
    average_waiting_time = sum_waiting_time / num;
    printf("\nAverage Waiting Time: %lf",
average_waiting_time);

    //AVERAGE TURNAROUND TIME
    average_turnaround_time = sum_turnaround_time / num;
    printf("\nAverage Turnaround Time: %lf",
average_turnaround_time);
    */

    return 0;
}
```


Ubuntu 22.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
ডিসেম্বর 10 05:09
shaianebarol@shaianebarol: ~/Desktop/LAB04
shaianebarol@shaianebarol:~/Desktop/LAB04$ gcc Task3.c -o Task3
shaianebarol@shaianebarol:~/Desktop/LAB04$ ./Task3
Please enter the total number of processes: 5
Please enter the arrival and burst times of the processes:
For P1,
Arrival Time: 0
Burst Time: 15
Priority: 2
For P2,
Arrival Time: 14
Burst Time: 5
Priority: 4
For P3,
Arrival Time: 3
Burst Time: 10
Priority: 0
For P4,
Arrival Time: 9
Burst Time: 22
Priority: 3
For P5,
Arrival Time: 7
Burst Time: 16
Priority: 1
For P3,
Completion Time = 13.000000
Turnaround Time = 10.000000
Waiting Time = 0.000000
For P5,
Completion Time = 29.000000
Turnaround Time = 22.000000
Waiting Time = 6.000000
For P1,
Completion Time = 41.000000
Turnaround Time = 41.000000
Waiting Time = 26.000000
For P4,
Completion Time = 63.000000
Turnaround Time = 54.000000
Waiting Time = 32.000000
Right Ctrl


Ubuntu 22.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
ডিসেম্বর 10 05:10
shaianebarol@shaianebarol: ~/Desktop/LAB04
Priority: 2
For P2,
Arrival Time: 14
Burst Time: 5
Priority: 4
For P3,
Arrival Time: 3
Burst Time: 10
Priority: 0
For P4,
Arrival Time: 9
Burst Time: 22
Priority: 3
For P5,
Arrival Time: 7
Burst Time: 16
Priority: 1
For P3,
Completion Time = 13.000000
Turnaround Time = 10.000000
Waiting Time = 0.000000
For P5,
Completion Time = 29.000000
Turnaround Time = 22.000000
Waiting Time = 6.000000
For P1,
Completion Time = 41.000000
Turnaround Time = 41.000000
Waiting Time = 26.000000
For P4,
Completion Time = 63.000000
Turnaround Time = 54.000000
Waiting Time = 32.000000
For P2,
Completion Time = 68.000000
Turnaround Time = 54.000000
Waiting Time = 49.000000
shaianebarol@shaianebarol:~/Desktop/LAB04$
Right Ctrl