Green University of Bangladesh
Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering
Semester: (Spring , Year:2024), B.Sc. in CSE (Day)

LAB REPORT NO #03

Course Title: Operating System Lab

Course Code: CSE - 310 **Section: 213_D5**

Lab Experiment Name: CPU Scheduling Algorithms to find Turnaround Time and Waiting Time.

Student Details

Name	ID
MD Dulal Hossain	213902116

Lab Date : **08 - 05 - 2024**
Submission Date : **10 - 05 - 2024**

Course Teacher's Name : **Md. Solaiman Mia**

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**
Comments:..... **Date:.....**

Task :

Title :

Implement the following problem using Scheduling Algorithms (Priority > SJF > FCFS).

Table 1: **Sample Input**

Process	Burst Time	Priority
P1	10	4
P2	13	1
P3	7	3
P4	15	2
P5	6	3
P6	10	4

Table 2: **Sample Output**

Process	Burst Time	Priority	Waiting Time	Turnaround Time
P2	13	1	0	13
P4	15	2	13	28
P5	6	3	28	34
P3	7	3	34	41
P1	10	4	41	51
P6	10	4	51	61

Algorithms :

1. Input:
    - The script prompts the user to input the number of processes (n), burst time, and priority for each process.
2. Process Sorting:
    - The script sorts the processes based on priority and burst time.
    - It iterates over each process and compares its priority with the others. If the priority is lower, swaps their positions. If priorities equal,it compares burst times.
    - This sorting ensures that processes with higher priority or lower burst time are processed first.
3. Waiting Time Calculation:
    - It initializes the waiting time array wt with the first element as 0 (since the first process has no waiting time).
    - For each subsequent process, it calculates the waiting time by summing up the burst times of all previous processes.
4. Turnaround Time Calculation:
    - It calculates the turnaround time for each process by adding its burst time and waiting time.
5. Average Waiting Time and Turnaround Time Calculation:
    - It calculates the total waiting time and total turnaround time by summing up the respective values for all processes.
    - Then, it computes the average waiting time and average turnaround time by dividing the total values by the number of processes (n).
6. Output:
    - Finally, it displays the details of each process including burst time, priority, waiting time, and turnaround time.
    - It also displays the average waiting time and average turnaround time.

Code write in Hand :

```
echo "Enter the number of processes : "
read n
echo " "
for((i=0 ; i<n ; i++))
 do
  echo "Enter burst time for process $((i+1)): "
    read bt[$i]

  echo "Enter priority for process $((i+1)): "
    read pr[$i]
done
 for ((i=0; i<n; i++))
 do
      pos = $i
      for((j= i+1; j<n; j++))
       do
          if [ ${pr[$j]} - lt ${pr[$pos]} ]; then pos=$j
          else if [ ${pr[$j]} - eq ${pr[$pos]} ] ; then
           if [ ${bt[$j]} - lt ${bt[$pos]} ]; then pos = $j
           fi
        fi
      done
 temp = ${pr[$i]}
  pr[$i] = ${pr[$pos]}
  pr[$pos] = $temp
  temp = ${bt[$i]}
  bt[$i] = ${bt[$pos]}
   bt[$pos] = $temp
   temp = ${p[$i]}
   p[$i] = ${p[$pos]}
    p[$pos] = $temp
done
wt[0] = 0
```

```
for (( i=1 ; i<n; i++ ))
do
    wt[$i]=0
    for (( j=0 ; j<i; j++ ))
    do
        wt[$i]=$((wt[$i]+bt[$j]))
    done
done
total_wt=0
total_tat=0
for (( i=0; i<n; i++ ))
do
    tat[$i]=$((bt[$i]+wt[$i]))
    total_wt=$((total_wt+wt[$i]))
    total_tat=$((total_tat+tat[$i]))
done
avg_wt=$(echo "scale=2; $total_wt/$n" | bc)
avg_tat=$(echo "scale=2; $total_tat/$n" | bc)
echo ""
echo "process  BrustTime  Priority  waitingtime  Turnaround Time"
for (( i=0 ; i<n; i++))
do echo "p ${p[$i]}  ${bt[$i]}  ${pr[$i]}
        ${wt[$i]}  ${tat[$i]}"
done

echo ""
echo "Average waiting Time : $avg_wt"
echo "Average Turnaround Time : $AVG_tat"
```

Source Code in write :

```
#!/bin/bash

echo "Enter the number of processes: "
read n
echo ""

for ((i = 0; i < n; i++))
do
    echo -n "Enter Burst Time for process $((i + 1)): "
    read bt[$i]
    echo -n "Enter Priority for process $((i + 1)): "
    read pr[$i]
    p[$i]=$((i + 1)) # Contains process number
done

for ((i = 0; i < n; i++))
do
    pos=$i
    for ((j = i + 1; j < n; j++))
    do
        if [ ${pr[$j]} -lt ${pr[$pos]} ]; then
            pos=$j
        elif [ ${pr[$j]} -eq ${pr[$pos]} ]; then
            if [ ${bt[$j]} -lt ${bt[$pos]} ]; then
                pos=$j
            fi
        fi
    done

    temp=${pr[$i]}
    pr[$i]=${pr[$pos]}
    pr[$pos]=$temp

    temp=${bt[$i]}
    bt[$i]=${bt[$pos]}
    bt[$pos]=$temp

    temp=${p[$i]}
    p[$i]=${p[$pos]}
    p[$pos]=$temp
done
```

```
wt[0]=0

# Calculate waiting time

for ((i = 1; i < n; i++))
do
    wt[$i]=0
    for ((j = 0; j < i; j++))
    do
        wt[$i]=$((wt[$i] + bt[$j]))
    done
done

# Calculate turnaround time

total_wt=0
total_tat=0
for ((i = 0; i < n; i++))
do
    tat[$i]=$((bt[$i] + wt[$i]))
    total_wt=$((total_wt + wt[$i]))
    total_tat=$((total_tat + tat[$i]))
done

avg_wt=$(echo "scale=2; $total_wt / $n" | bc)
avg_tat=$(echo "scale=2; $total_tat / $n" | bc)

echo ""
echo "Process       Burst Time     Priority      Waiting time
Turnaround Time"

for ((i = 0; i < n; i++))
do
    echo "P${p[$i]}            ${bt[$i]}          ${pr[$i]}
${wt[$i]}              ${tat[$i]}"
done

echo ""
echo "Average Waiting Time: $avg_wt"
echo "Average Turnaround Time: $avg_tat"
```

Output :

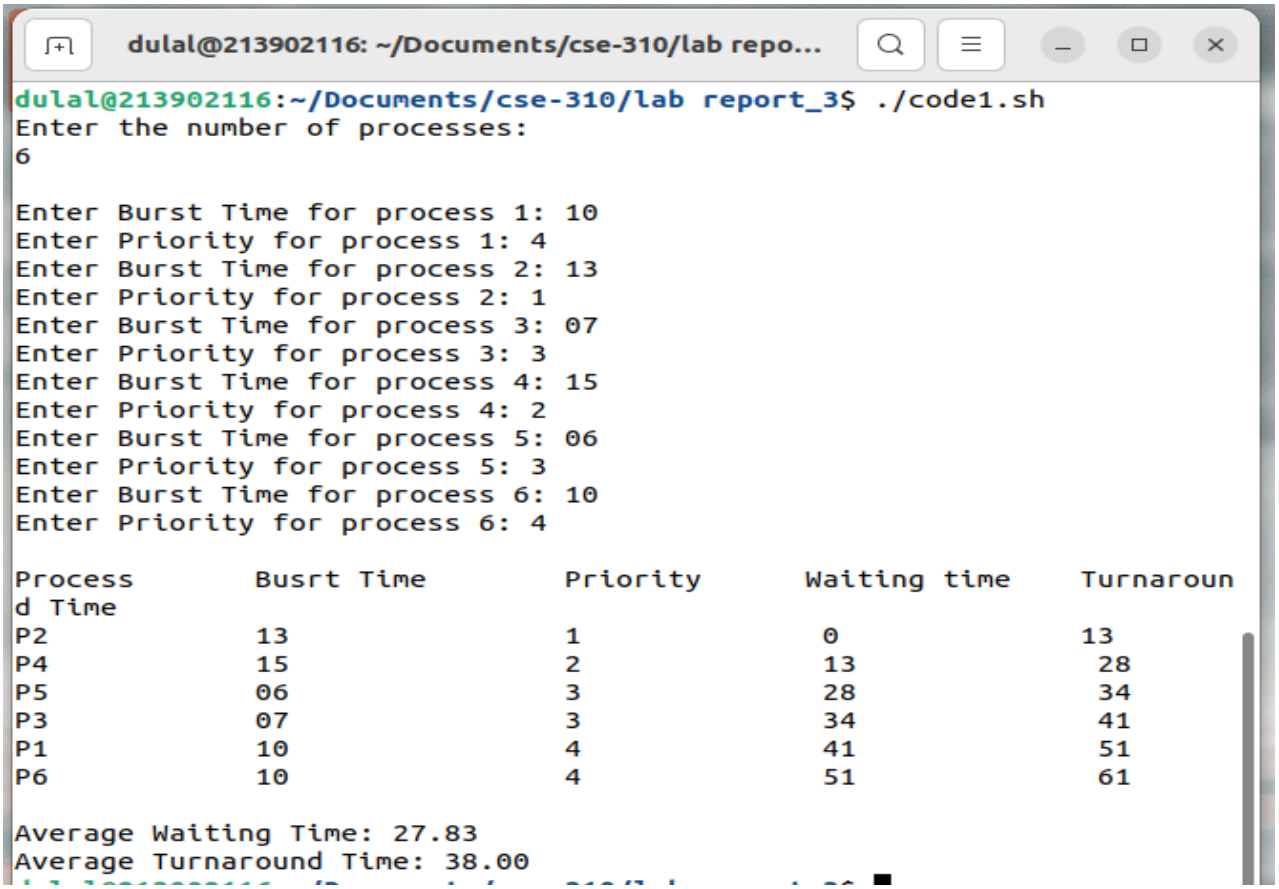

Figure 1.1 : Output in show Successfully.

Explain Output :

1. Input:
   - The script prompts the user to enter the number of processes.
   - For each process, it asks for the Burst Time and Priority.
2. Sorting:
   - The script sorts the processes based on their priority and burst time.
   - If two processes have the same priority, it further considers their burst time to break the tie.
3. Waiting Time Calculation:
   - It calculates the waiting time for each process.
   - The waiting time for process i is the sum of burst times of all processes with a lower priority than i.
4. Turnaround Time Calculation:
   - Turnaround time for each process is the sum of its burst time and waiting time.
5. Display:
   - The script displays a table with process details:
   - Process number (P1, P2, P3, P4, P5, P6)
   - Process Burst time Priority Waiting time Turnaround time
   - It also shows the average waiting time and average turnaround time.