EXPERIMENT - 1

Operating Systems Lab

AIM

Write a program to implement CPU scheduling for first come first serve.

Syeda Reeha Quasar
14114802719
6C7

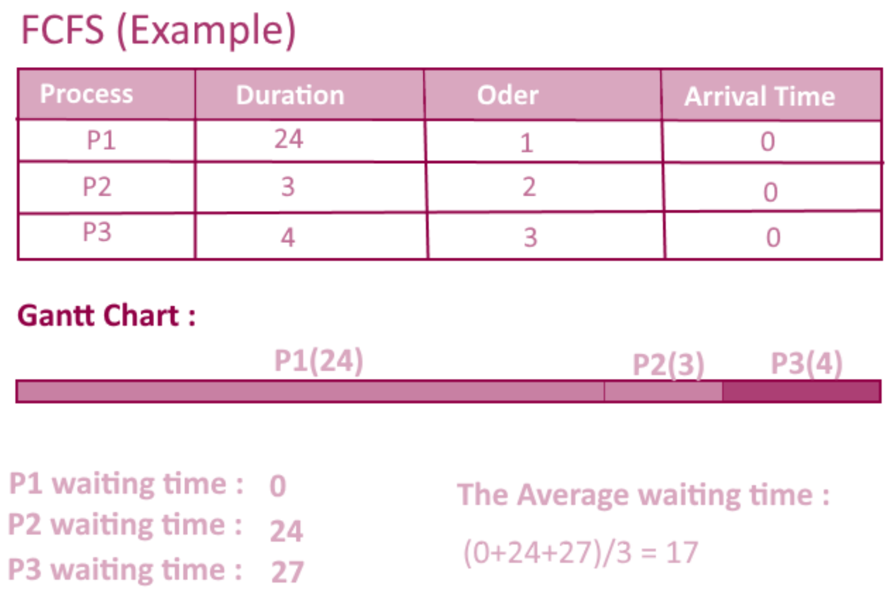
EXPERIMENT – 1

Aim:

Write a program to implement CPU scheduling for first come first serve.

Theory:

The first come first serve (FCFS) scheduling algorithm simply schedules the jobs according to their arrival time. The job which comes first in the ready queue will get the CPU first. The lesser the arrival time of the job, the sooner will the job gets the CPU. FCFS scheduling may cause the problem of starvation if the burst time of the first process is the longest of all the jobs.



FCFS (Example)

Process	Duration	Oder	Arrival Time
P1	24	1	0
P2	3	2	0
P3	4	3	0

Gantt Chart :

P1 waiting time : 0

P2 waiting time : 24

P3 waiting time : 27

The Average waiting time :

(0+24+27)/3 = 17

Computation:

1. Completion Time: Time at which process completes its execution.
2. Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time
3. Waiting Time(W.T): Time Difference between turn around time and burst time. Waiting Time = Turn Around Time – Burst Time

Algorithm:

```
1-  Input the processes along with their burst time (bt).
2-  Find waiting time (wt) for all processes.
3-  As first process that comes need not to wait so
    waiting time for process 1 will be 0 i.e. wt[0] = 0.
4-  Find waiting time for all other processes i.e. for
     process i ->
       wt[i] = bt[i-1] + wt[i-1] .
5-  Find turnaround time = waiting_time + burst_time
```

for all processes.

6- Find **average waiting time** =

total_waiting_time / no_of_processes.

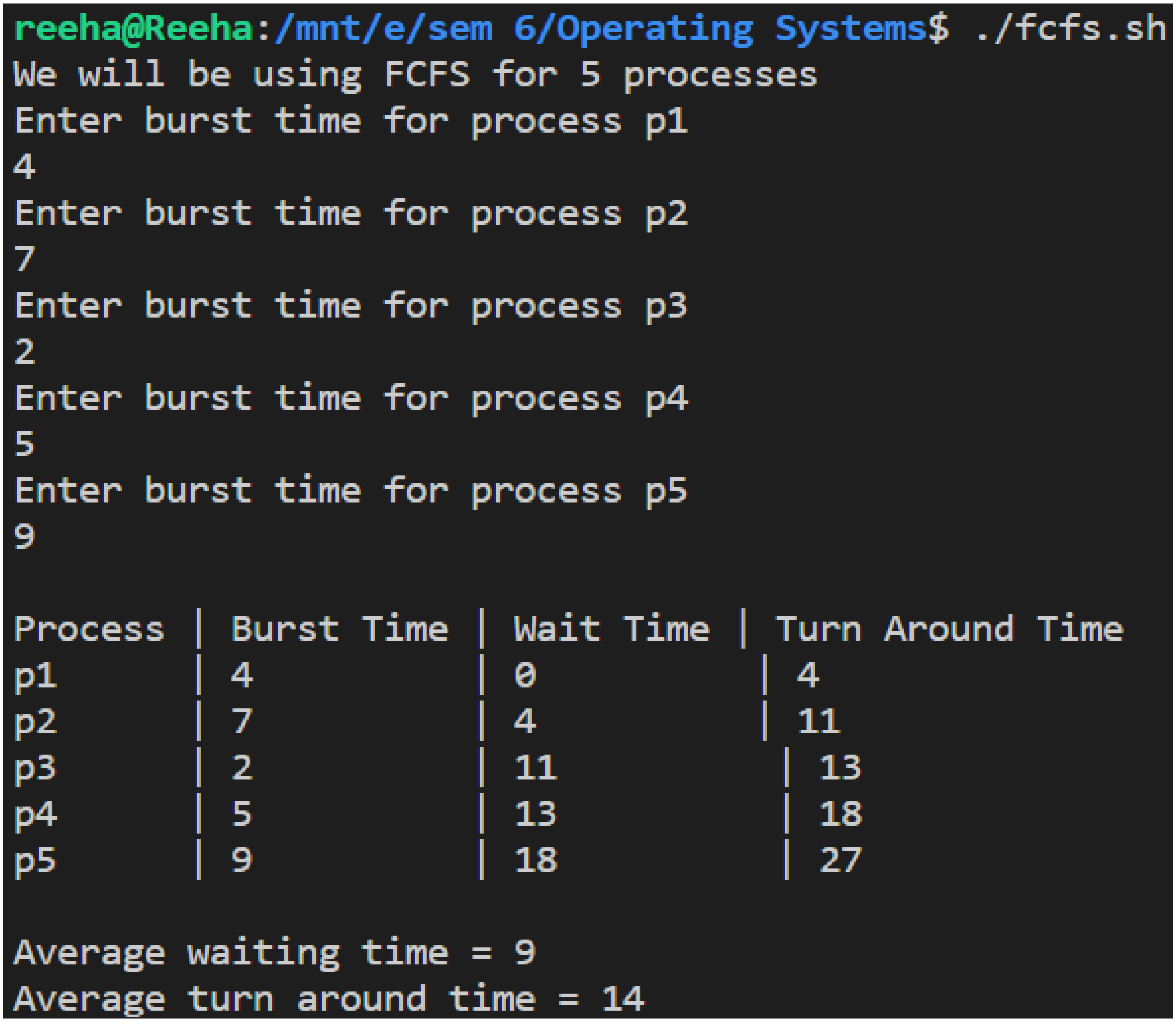
7- Similarly, find **average turnaround time** =

total_turn_around_time / no_of_processes.

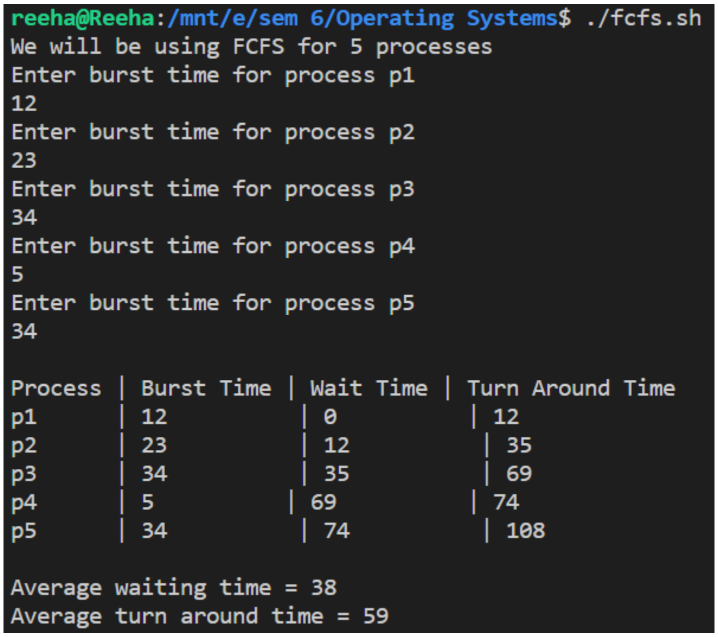
Source Code:

```
process=("p1" "p2" "p3" "p4" "p5")
bt=(0 0 0 0 0)
wt=(0 0 0 0 0)
tat=(0 0 0 0 0)

bt[0]=1
factor=0
wtSum=0
tatSum=0

echo "We will be using FCFS for 5 processes"

for ((i=0; i<5; i++))
do
  echo "Enter burst time for process ${process[i]}"
  read ele
  bt[i]=$ele
  wt[i]=$factor
  factor=$((factor + ele))
  tat[i]=$((bt[i] + wt[i]))
  wtSum=$((wt[i] + wtSum))
  tatSum=$((tat[i] + tatSum))
done

echo ""
echo "Process | Burst Time | Wait Time | Turn Around Time"

for ((i=0; i<5; i++))
do
  echo "${process[i]}      | ${bt[i]}         | ${wt[i]}         | ${tat[i]}"
done

echo ""
echo "Average waiting time = $((wtSum/5))"
echo "Average turn around time = $((tatSum/5))"
```

Output:



```
reeha@Reeha:/mnt/e/sem 6/Operating Systems$ ./fcfs.sh
We will be using FCFS for 5 processes
Enter burst time for process p1
4
Enter burst time for process p2
7
Enter burst time for process p3
2
Enter burst time for process p4
5
Enter burst time for process p5
9

Process | Burst Time | Wait Time | Turn Around Time
p1      | 4          | 0          | 4
p2      | 7          | 4          | 11
p3      | 2          | 11          | 13
p4      | 5          | 13          | 18
p5      | 9          | 18          | 27

Average waiting time = 9
Average turn around time = 14
```



```
reeha@Reeha:/mnt/e/sem 6/Operating Systems$ ./fcfs.sh
We will be using FCFS for 5 processes
Enter burst time for process p1
12
Enter burst time for process p2
23
Enter burst time for process p3
34
Enter burst time for process p4
5
Enter burst time for process p5
34

Process | Burst Time | Wait Time | Turn Around Time
p1      | 12          | 0          | 12
p2      | 23          | 12          | 35
p3      | 34          | 35          | 69
p4      | 5          | 69          | 74
p5      | 34          | 74          | 108

Average waiting time = 38
Average turn around time = 59
```