Rajshahi University of Engineering & Technology

Course No.: CSE 3202
Course Title: Sessional Based on CSE 3201

Submitted To:

Mohiuddin Ahmed
Lecturer
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology

Submitted By:

Name: Mondol Mridul Provakar
Roll: 1803062
Section: B
Department: Computer Science & Engineering

Date of submission: 05-12-2022

Problem Name: Implementation of first come first serve algorithm.

Theory:

In this algorithm the job which will come first at the ready queue that job will get cpu first and will be first served. For implementation the given algorithm is needed.

Algorithm of first come first serve cpu scheduling is given below-

1. Input the processes along with their burst time.

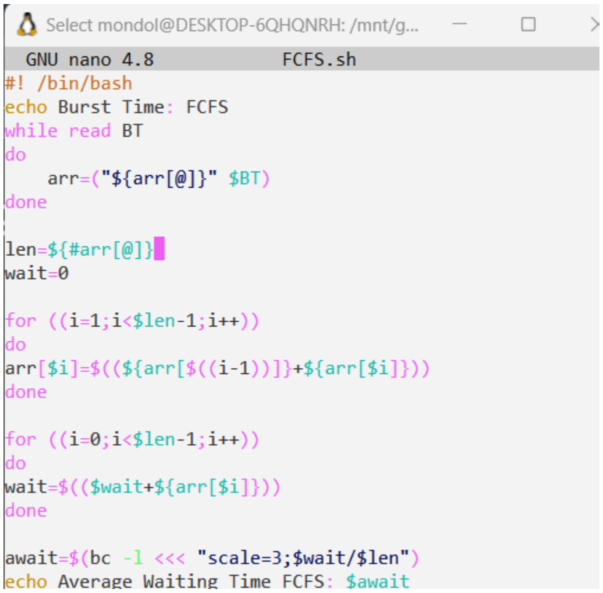
2. Find waiting time (wt) for all processes.

3. As the first process that comes need not to wait so waiting time for process 1 will be 0 that means wt[0] = 0.

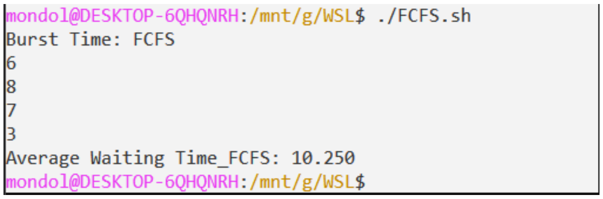
4. Find waiting time for all other processes. For all processes wt[i] = wt[i-1]+ burst_time[i-1] .

5. Find average waiting time = total_waiting_time / no_of_processes.

Source code & Output:



```
  GNU nano 4.8              FCFS.sh
#! /bin/bash
echo Burst Time: FCFS
while read BT
do
    arr=("${arr[@]}" $BT)
done

len=${#arr[@]}
wait=0

for ((i=1;i<$len-1;i++))
do
arr[$i]=$((${arr[$((i-1))]}+${arr[$i]}))
done

for ((i=0;i<$len-1;i++))
do
wait=$(($wait+${arr[$i]}))
done

await=$(bc -l <<< "scale=3;$wait/$len")
echo Average Waiting Time FCFS: $await
```



```
mondol@DESKTOP-6QHQNRH:/mnt/g/WSL$ ./FCFS.sh
Burst Time: FCFS
6
8
7
3
Average Waiting Time_FCFS: 10.250
mondol@DESKTOP-6QHQNRH:/mnt/g/WSL$
```

Discussion: First Come First Serve (FCFS) is an operating system scheduling algorithm that automatically executes queued requests and processes in order of their arrival. It is the easiest and simplest CPU scheduling algorithm.

The experiment was implemented successfully.

Problem Name: Implementation of shortest job first algorithm.

Theory:
In this algorithm the job which has the shortest burst time will get cpu first and will be first served. All jobs get cpu according to their burst time. For implementation the given algorithm is needed.

Algorithm of shortest job first cpu scheduling is given below-

1. Input the processes along with their burst time.

2. Sort the burst time increasingly using bubble sort.

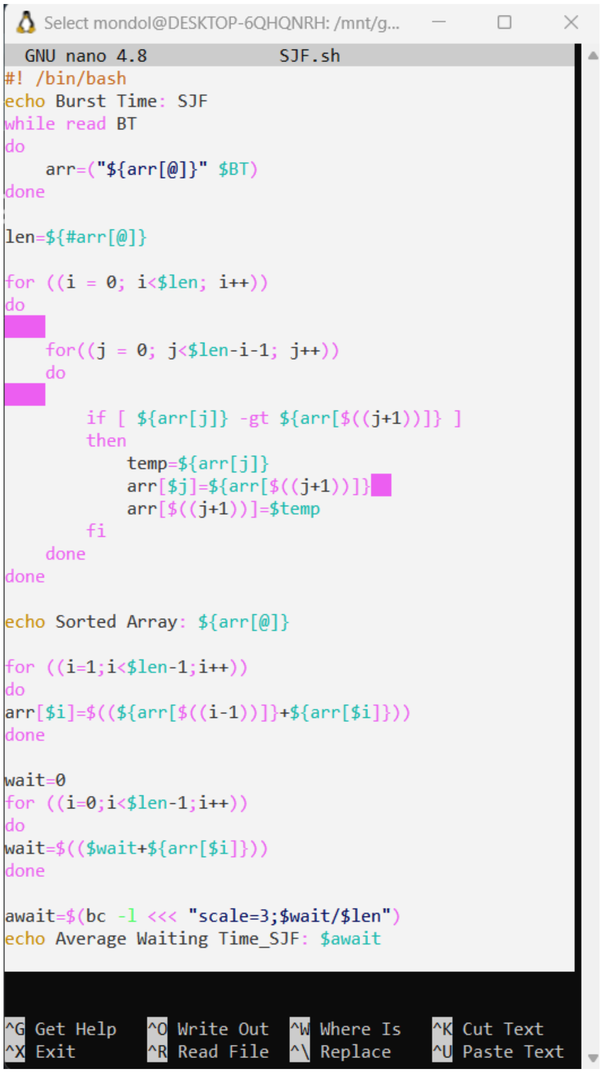
3. Find waiting time (wt) for all processes.

4. As the first process that comes need not to wait so waiting time for process 1 will be 0 that means wt[0] = 0.

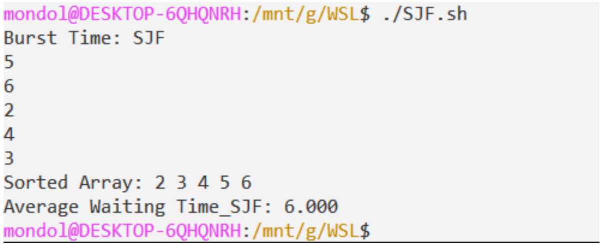
5. Find waiting time for all other processes. For all processes wt[i] = wt[i-1]+ burst_time[i-1].

6. Find average waiting time = total_waiting_time / no_of_processes.

Source code & Output:



```
  GNU nano 4.8                SJF.sh
#! /bin/bash
echo Burst Time: SJF
while read BT
do
    arr=("${arr[@]}" $BT)
done

len=${#arr[@]}

for ((i = 0; i<$len; i++))
do

    for((j = 0; j<$len-i-1; j++))
    do

        if [ ${arr[j]} -gt ${arr[$((j+1))]} ]
        then
            temp=${arr[j]}
            arr[$j]=${arr[$((j+1))]}
            arr[$((j+1))]=$temp
        fi
    done
done

echo Sorted Array: ${arr[@]}

for ((i=1;i<$len-1;i++))
do
arr[$i]=$((${arr[$((i-1))]}+${arr[$i]}))
done

wait=0
for ((i=0;i<$len-1;i++))
do
wait=$(($wait+${arr[$i]}))
done

await=$(bc -l <<< "scale=3;$wait/$len")
echo Average Waiting Time_SJF: $await

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text
```



```
mondol@DESKTOP-6QHQNRH:/mnt/g/WSL$ ./SJF.sh
Burst Time: SJF
5
6
2
4
3
Sorted Array: 2 3 4 5 6
Average Waiting Time_SJF: 6.000
mondol@DESKTOP-6QHQNRH:/mnt/g/WSL$
```

Discussion: Shortest Job First (SJF) is an algorithm in which the process having the smallest execution time is chosen for the next execution. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution.

The experiment was implemented successfully.