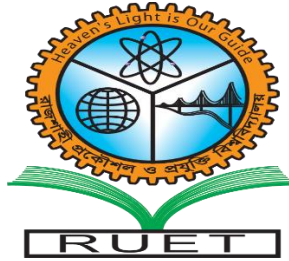


Heaven's Light is Our Guide



Computer Science And Engineering

Rajshahi University of Engineering and Technology

Course No: CSE3202

Course Title: Operating system

Date of Submission: 28-11-22

Submitted To	Submitted By
Mohiuddin Ahmed Lecturer, Department Of Computer Science And Engineering Rajshahi University Of Engineering And Technology	Nazmul Haque Roll: 1803109 Section: B Department Of Computer Science And Engineering Rajshahi University Of Engineering And Technology

i) First Come First Serve(fcfs): always that process get cpu time which come first in ready queue.

At first take some burst time of a process and then find out waiting time and average waiting time. Such as,

Process	Burst time(bt)
P1	24
P2	3
P3	4

Gannt chart of fcfs:

P1	P2	P3
0ms	24ms	27ms

Code:

```
nazmul1803109@DESKTOP-VMBU6LB: /mnt/d/Study/3-2 Semester/CSE-3202/Lab-3(31-10-22)
GNU nano 6.2 avg.sh
#!/bin/bash
echo "Enter bt(Burst time) : "
read -a bt
n=${#bt[@]}
n=$((n-1))
wt=(0 0 0 0 0 0 0)
for ((i=1;i<=$n;i++));
do
    wt[i]=$((bt[i-1]+wt[i-1]))
done
total=0
for t in ${wt[@]}
do
    total=$((total+t))
done
n=$((n+1))
echo "Average Waiting time : "
echo "scale=3;$total/$n"|bc
```

Input and Output:

```
nazmul1803109@DESKTOP-VMBU6LB:/mnt/d/Study/3-2 Semester/CSE-3202/Lab-3(31-10-22)$ nano avg.sh
nazmul1803109@DESKTOP-VMBU6LB:/mnt/d/Study/3-2 Semester/CSE-3202/Lab-3(31-10-22)$ ./avg.sh
Enter bt(Burst time) :
24 3 4
Average Waiting time :
17.000
```

ii) Shortest job first (sif) : always that process get cpu time which has less burst time.

First of all take burst time of all processes . then sort the burst time in increasing order. After sorting find out waiting time of all process and also find out the average time of all process.

For example:

Process	Burst time(bt)
P1	24
P2	3
P3	4

Gannt chart of sjf:

P2	P3	P1
0ms	3ms	7ms

Code:

```

nazmul1803109@DESKTOP-VMBU6LB: /mnt/d/Study/3-2 Semester/CSE-3202/Lab-3(31-10-22)
GNU nano 6.2 sjf2.sh
#!/bin/bash
echo "Enter bt(burst time) : "
read -a bt
bt=($(printf '%s\n' "${bt[@]}" | sort -n))
n=${#bt[@]}
n=$((n-1))
wt=(0 0 0 0 0 0 0 0)
for ((i=1;i<=$n;i++));
do
    wt[i]=$((bt[i-1]+wt[i-1]))
done
total=0
for t in ${wt[@]}
do
    total=$((total+t))
done
n=$((n+1))
echo "Average Waiting time : "
echo "scale=3;$total/$n"|bc

```

Input and Output:

```
nazmul1803109@DESKTOP-VMBU6LB:/mnt/d/Study/3-2 Semester/CSE-3202/Lab-3(31-10-22)$ nano sjf2.sh
nazmul1803109@DESKTOP-VMBU6LB:/mnt/d/Study/3-2 Semester/CSE-3202/Lab-3(31-10-22)$ ./sjf2.sh
Enter bt(burst time) :
24 3 4
Average Waiting time :
3.333
```

Discussion:

All of the above code run successfully. But I faced a problem for using syntax. Shell coding is almost similar like other high level programming language(such as c,c++,python etc).

Main observation from this two algorithm is that we must use shortest job first algorithm because less average waiting time required in it than first come first serve algorithm.

To do implement this algorithm at first we take an array which contain the burst time of the processes and find out waiting time for all processes then find out average waiting time(this process is for fcfs). Same process follow in shortest job first but we need sort the burst time of all processes at first.