

Lab Exercise - 7

- AIM :: WAP in shell script to implement CPU scheduling for `first come first serve` (fcfs).

Source_Code ::

```
echo $'\n' "5C6 - Amit Singhal (11614802722)" $'\n'

read -p "Enter the number of processes: " num_processes
echo $'\n' "Enter Arrival Time & Burst Time for $num_processes processes"

# Collect process details
for ((i=0;i<num_processes;i++)); do
    echo -n "P$((i+1)): "
    read arrival_time burst_time
    processes[$i]="$arrival_time $burst_time"
done

# Sort processes by arrival time
IFS=$'\n' sorted_processes=$(sort -n -k1 <<<"${processes[*]}")
unset IFS

# Initialize variables
total_completion_time=0
total_waiting_time=0
total_turnaround_time=0
gantt_chart="0" # Start Gantt chart at time 0

# Display table header
```

```
echo -e "\nProcess  Arrival Time  Burst Time  Completion Time  TurnAround  
Time  Waiting Time"
```

```
# Process all processes
```

```
for ((i=0;i<num_processes;i++)); do
```

```
    current_process=${sorted_processes[$i]}
```

```
    current_arrival_time=${current_process[0]}
```

```
    current_burst_time=${current_process[1]}
```

```
# If the process arrives after the last completion time, idle CPU
```

```
if (( total_completion_time < current_arrival_time )); then
```

```
    idle_time=$((current_arrival_time - total_completion_time))
```

```
    total_completion_time=$current_arrival_time
```

```
    gantt_chart+=" -- XX -- $total_completion_time"
```

```
fi
```

```
# Calculate waiting time
```

```
if (( total_completion_time >= current_arrival_time )); then
```

```
    waiting_time=$((total_completion_time - current_arrival_time))
```

```
else
```

```
    waiting_time=0
```

```
fi
```

```
# Calculate completion time and turnaround time
```

```
completion_time=$((total_completion_time + current_burst_time))
```

```
turnaround_time=$((completion_time - current_arrival_time))
```

```
# Update total values
```

```
total_completion_time=$completion_time
```

```
total_waiting_time=$((total_waiting_time + waiting_time))
```

```
total_turnaround_time=$((total_turnaround_time + turnaround_time))
```

```
# Display process details
```

```
echo -e "P$((i+1))\t\t$current_arrival_time\t\t$current_burst_time\t\t  
$completion_time\t\t $turnaround_time\t\t $waiting_time"
```

```

# Update Gantt chart

gantt_chart+=" -- P$((i+1)) -- $completion_time"

done

# Calculate averages

avg_waiting_time=$(awk "BEGIN {printf \"%.2f\",
$total_waiting_time/$num_processes}")

avg_turnaround_time=$(awk "BEGIN {printf \"%.2f\",
$total_turnaround_time/$num_processes}")

# Display Gantt chart

echo -e "\nGantt Chart:"

echo -e "$gantt_chart"

# Display averages

echo ""

echo "Avg waiting time: $avg_waiting_time"

echo "Avg turnaround time: $avg_turnaround_time"

```

Output ::

```

singhal-amit@singhal-amit-ThinkPad-T430:~/Downloads/_LAB_Wrk/OS/Code$ vi prg_5_fcfs.sh
singhal-amit@singhal-amit-ThinkPad-T430:~/Downloads/_LAB_Wrk/OS/Code$ chmod +x prg_5_fcfs.sh
singhal-amit@singhal-amit-ThinkPad-T430:~/Downloads/_LAB_Wrk/OS/Code$ ./prg_5_fcfs.sh

```

5C6 - Amit Singhal (11614802722)

Enter the number of processes: 4

Enter Arrival Time & Burst Time for 4 processes

P1: 0 2

P2: 1 2

P3: 5 3

P4: 6 4

Process	Arrival Time	Burst Time	Completion Time	TurnAround Time	Waiting Time
P1	0	2	2	2	0
P2	1	2	4	3	1
P3	5	3	8	3	0
P4	6	4	12	6	2

Gantt Chart:

0 -- P1 -- 2 -- P2 -- 4 -- XX -- 5 -- P3 -- 8 -- P4 -- 12

Avg waiting time: 0.75

Avg turnaround time: 3.50