Lab Exercise – 10

AIM :: Implement Round Robin Scheduling On Linux Using Shell Scripting.

Theory ::

- **Round Robin (RR) Scheduling** is a **preemptive** CPU scheduling algorithm.
- Each process is assigned a fixed **time quantum** or slice.
- Processes are executed in a cyclic order, ensuring **fairness** by giving each process an equal share of CPU time.
- When a process's time quantum expires, it is placed at the back of the **ready queue**.
- This scheduling method reduces **starvation** and is ideal for **time-sharing** systems.
- Performance depends on the size of the time quantum: too small increases context switching, while too large reduces responsiveness.
- Round Robin is simple to implement and widely used in **multitasking** environments.
- It balances **CPU utilization** across processes by distributing CPU time fairly.
- It offers good **response time** for interactive systems but can lead to **higher turnaround times** if many processes have long burst times.

Source_Code ::

```
echo -e "\n# Process Scheduling Simulation\n"

echo -e "Amit Singhal - 11614802722 (5C6)\n"


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


echo -n "Enter time quantum: "
read tq


for ((i = 0; i < n; i++)); do
    echo -n "Enter arrival time & burst time for process P$((i+1)): "
    read arrival[$i] burst[$i]
    p[$i]=$i
    temp[$i]=${burst[$i]}
    tat[$i]=0
    wt[$i]=0
```

```
done

gantt_process=()
gantt_time=()
time=0
remain=$n

while ((remain != 0)); do
    for ((i = 0; i < n; i++)); do
        if ((temp[i] > 0)); then
            gantt_process+=(${p[$i]})
            gantt_time+=($time)

            if ((temp[i] <= tq)); then
                time=$((time + temp[i]))
                tat[$i]=$time
                wt[$i]=$((time - burst[$i]))
                temp[$i]=0
                remain=$((remain - 1))
            else
                temp[$i]=$((temp[$i] - tq))
                time=$((time + tq))
            fi
        fi
    done
done
gantt_time+=($time)

echo -e "\n## Process Table\n"
echo -e "| Process | AT | BT | CT | TAT | WT |"
echo -e "|---------|----|----|----|-----|----|"
for ((i = 0; i < n; i++)); do
    ct[$i]=${tat[$i]}
    echo -e "| P$((i+1))      | ${arrival[$i]}  | ${burst[$i]}  | ${ct[$i]}  | ${tat[$i]}   | ${wt[$i]}  |"
done

total_waiting_time=0
```

```
total_turnaround_time=0
for ((i = 0; i < n; i++)); do
    total_waiting_time=$((total_waiting_time + wt[$i]))
    total_turnaround_time=$((total_turnaround_time + tat[$i]))
done
avg_waiting_time=$(echo "scale=2; $total_waiting_time / $n" | bc)
avg_turnaround_time=$(echo "scale=2; $total_turnaround_time / $n" | bc)

echo -e "\nTotal Waiting Time: $total_waiting_time"
echo -e "Average Waiting Time: $avg_waiting_time"
echo -e "\nTotal Turnaround Time: $total_turnaround_time"
echo -e "Average Turnaround Time: $avg_turnaround_time"

echo -e "\n## Gantt Chart\n"
echo -n "+"
for ((i = 0; i < ${#gantt_process[@]}; i++)); do
    echo -n "----"
done
echo -e "+\n"

for ((i = 0; i < ${#gantt_process[@]}; i++)); do
    echo -n "| P$((gantt_process[$i] + 1)) "
done
echo "|"

echo -n "+"
for ((i = 0; i < ${#gantt_process[@]}; i++)); do
    echo -n "----"
done
echo -e "+"

for ((i = 0; i < ${#gantt_time[@]}; i++)); do
    echo -n "${gantt_time[$i]} "
done
echo -e "\n"
```

Output ::

```
singhal-amit@singhal-amit-ThinkPad-T430:~$ vi amit.sh
singhal-amit@singhal-amit-ThinkPad-T430:~$ chmod +x amit.sh
singhal-amit@singhal-amit-ThinkPad-T430:~$ ./amit.sh

# Process Scheduling Simulation

Amit Singhal - 11614802722 (5C6)

Enter number of processes: 4
Enter time quantum: 2

Enter arrival time & burst time for process P1: 0 6
Enter arrival time & burst time for process P2: 1 8
Enter arrival time & burst time for process P3: 2 7
Enter arrival time & burst time for process P4: 3 3

## Process Table

| Process | AT | BT | CT | TAT | WT |
|---------|----|----|----|-----|----|
| P1      | 0  | 6  | 19 | 19  | 13 |
| P2      | 1  | 8  | 22 | 21  | 13 |
| P3      | 2  | 7  | 24 | 22  | 15 |
| P4      | 3  | 3  | 13 | 10  | 7  |

Total Waiting Time: 48
Average Waiting Time: 12.00

Total Turnaround Time: 72
Average Turnaround Time: 18.00

## Gantt Chart

+----+----+----+----+----+----+----+----+----+----+----+----+
| P1 | P2 | P3 | P4 | P1 | P2 | P3 | P1 | P2 | P3 | P2 | P3 |
+----+----+----+----+----+----+----+----+----+----+----+----+
0    2    4    6    8    10   12   14   16   18   20   22   24

singhal-amit@singhal-amit-ThinkPad-T430:~$
```