Homework 2

Operating systems Fall 25

Roya Hosseini

HW2: CPU Scheduling Simulation (Round Robin and Multilevel Feedback Queue)

Objective:
To understand and implement CPU scheduling algorithms using simulation techniques. Students will gain insight into process management, time sharing, and fairness in CPU scheduling.

Part 1 – Round Robin Scheduling (50 points)

1. Write a program (in C, C++, or Python) that simulates **Round Robin Scheduling**.
2. Input:
    - Process IDs, burst times, and arrival times.
    - Time quantum (e.g., 4 units).
3. Output:
    - A Gantt chart showing process execution order.
    - Average waiting time and turnaround time.

Example Input:

```
PID   Arrival   Burst
P1    0         5
P2    1         3
P3    2         8
P4    3         6
Quantum: 4
```

Example Output:

```
Gantt Chart: P1 | P2 | P3 | P4 | P1 | P3 | P4 | P3
Average Waiting Time: 7.25
Average Turnaround Time: 14.5
```

Hints:

- Use a ready queue and manage process arrival dynamically.
- Keep track of remaining burst times.
- When a process's remaining time becomes 0, record its completion time.

Part 2 – Multilevel Feedback Queue (50 points)

Extend your simulator to implement **Multilevel Feedback Queue (MLFQ)** Scheduling.

Algorithm Design:

- Use 3 queues:
  - Q1 → Time Quantum = 4
  - Q2 → Time Quantum = 8
  - Q3 → FCFS (First Come, First Served)
- New processes always enter Q1.
- If a process doesn't finish in its time quantum, it is moved down one queue level.
- Processes in higher queues have priority over lower queues.

Output:

- Text-based Gantt chart showing CPU usage order.
- Average waiting and turnaround times.
- Table showing which queue each process finished in.

Example:

```
Queue 1: P1 | P2 | P3
Queue 2: P3 | P4
Queue 3: P4
Average Waiting Time: 6.9
Average Turnaround Time: 12.7
```

Deliverables:

- Source Code File: `hw2_rr_mlfq.c` or `hw2_rr_mlfq.py`
- Report (`hw2_report.pdf`) including:
  - Algorithm explanation.
  - Screenshots of input/output.
  - Observations comparing RR and MLFQ.

Extra Credit (+10 points)

Create a simple ASCII or matplotlib visualization of the Gantt chart.

Submission Instructions:

- **This is an individual assignment. Collaboration is not allowed.**
- Compress all deliverables into `hw2_cpu_scheduling_<yourname>.zip`.
- Submit via iCollege by the due date.

Grading Rubric:

Component	Description	Points
Round Robin Implementation	Accurate algorithm, correct calculations	25
MLFQ Implementation	Proper multi-level logic and queue transitions	25
Output Clarity	Gantt chart and metrics displayed clearly	15
Report Quality	Explanation, screenshots, analysis	25
Code Quality	Readability, comments, organization	10
Total		**100**

Note: Work individually. Include your full name at the top of your report and in the submission folder name.