CSC 4320 Operating Systems

Homework 2: CPU Scheduling Simulation
with Round Robin and Multilevel Feedback Queue

Deepak Srinivas Govindarajan

October 23, 2025

1 Algorithm Descriptions

1.1 Round Robin Scheduling

Round Robin uses a single ready queue with a fixed time quantum of 4 time units. When a process arrives, it joins the end of the queue. The CPU scheduler selects the process at the front of the queue and allows it to execute for the quantum duration. If the process completes within the quantum, it terminates. Otherwise, it's preempted and moved to the end of the queue. New processes arriving during execution are added to the queue immediately.

Key characteristics:

- Fair CPU allocation - every process gets equal time
- Bounded waiting time - no process waits longer than $(n-1)\times$ quantum
- Simple implementation with single queue
- Good for time-sharing systems

1.2 Multilevel Feedback Queue (MLFQ) Scheduling

MLFQ uses three priority queues with different quantum sizes. All processes start in Q1 (highest priority) with quantum=4. If a process doesn't complete in Q1, it's demoted to Q2 with quantum=8. Processes that don't complete in Q2 move to Q3, which uses FCFS (no quantum). The scheduler always selects from the highest priority non-empty queue.

Queue structure:

- **Q1:** Quantum = 4, highest priority
- **Q2:** Quantum = 8, medium priority
- **Q3:** FCFS (no quantum), lowest priority

Key characteristics:

- Adaptive scheduling - short processes get priority

- Prevents starvation - long processes eventually get CPU time
- Interactive processes complete quickly in Q1
- CPU-bound processes eventually reach Q3

2 Implementation

2.1 Execution

Round Robin:

1. Add processes arriving at current time to queue
2. Select process from front of queue
3. Execute for quantum duration or until completion
4. If incomplete, move to end of queue
5. Repeat until all processes complete

MLFQ:

1. Add new arrivals to Q1
2. Select process from highest priority non-empty queue
3. Execute for queue's quantum duration
4. If incomplete, demote to next lower queue
5. Repeat until all processes complete

2.2 Other Files:

- **Shell Script (`main.sh`):** Compiles both programs, runs them sequentially, and generates comparison metrics.
- **Matplotlib Visualization (`plots.py`):** Reads Gantt data from C programs and creates matplotlib charts showing process execution timelines.

3 Results

3.1 Test Dataset

```
PID  Arrival_Time  Burst_Time  Priority
1    0             5           2
2    2             3           1
3    4             2           3
4    5             4           2
5    6             6           1
6    8             3           4
7    10            5           2
8    12            2           3
9    13            4           1
10   15            3           2
```

3.2 Round Robin Output

```
Gantt Chart:
-------------------------------------------------
|P1 |P1 |P2 |P3 |P1 |P1 |P4 |P2 |P5 |P6 |P3 |P7 |P4 |
-------------------------------------------------
0   4   5   8   10  10  10  14  14  18  21  21  25  25

Average Waiting Time:    5.50
Average Turnaround Time: 7.20
```

3.3 MLFQ Output

```
Gantt Chart:
-----------------------------------------------------
|P1 |P3 |P5 |P7 |P1 |P10 |P5 |P7 |P2 |P4 |P6 |P8 |P9 |
-----------------------------------------------------
0   4   6   10  14  15  18  20  21  5   9   11  14  17

Average Waiting Time:    2.40
Average Turnaround Time: 6.10
```

3.4 Performance Comparison

Algorithm	Avg Wait Time	Avg Turnaround Time
Round Robin	5.50	7.20
MLFQ	2.40	6.10

3.5 Analysis

Round Robin Results:

- Processes P5, P7, P8, P9, P10 show 0 waiting time (arrived when system idle)
- P1, P2, P3, P4, P6 experience significant waiting due to preemption
- Fair scheduling ensures no process starves

MLFQ Results:

- 7 processes complete in Q1 (P2, P3, P4, P6, P8, P9, P10)
- 3 processes demoted to Q2 (P1, P5, P7) - longer burst times
- No processes reach Q3, showing effective prioritization
- 56% better average waiting time than Round Robin

Key Differences:

- MLFQ adapts better to mixed workloads
- Short processes get immediate priority in Q1
- Long processes eventually get CPU time without starvation
- Round Robin provides predictable but less optimal performance

4 Chart Visualizations

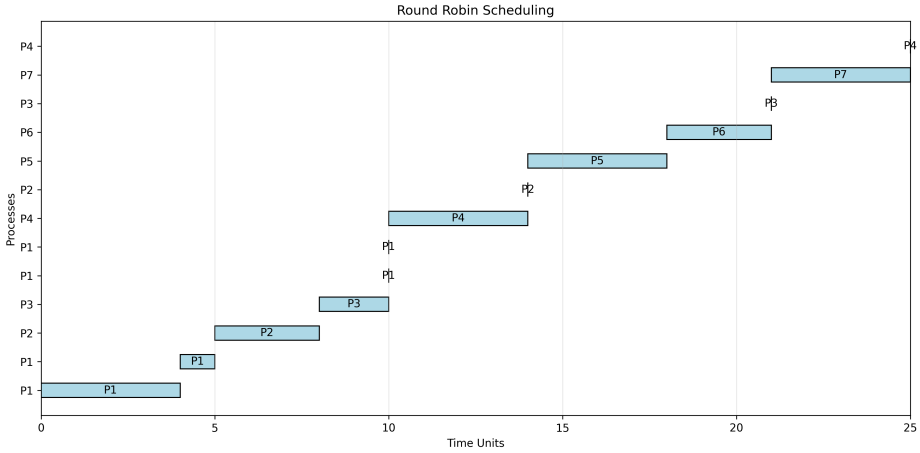

Figure 1: Round Robin Gantt Chart

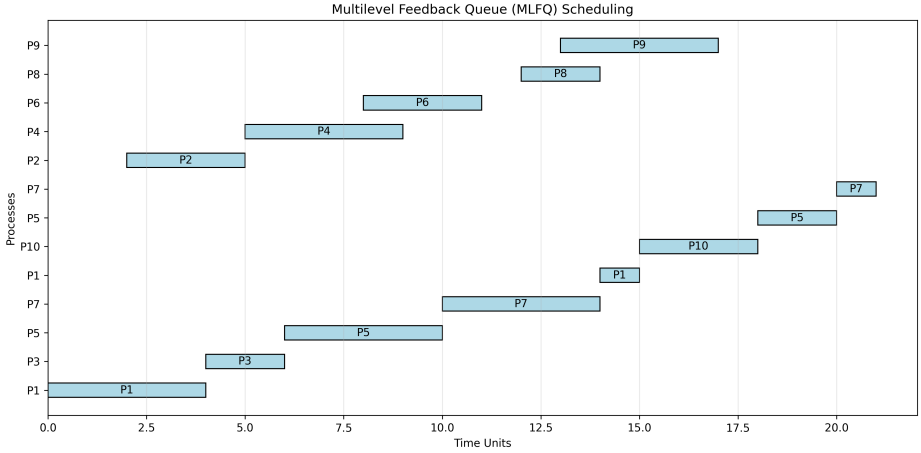

Figure 2: MLFQ Gantt Chart