

MLFQ Scheduler Implementation

Overview

This program implements a Multilevel Feedback Queue (MLFQ) scheduler with four queues:

1. **Q0**: Round Robin scheduling with 5ms time quantum
2. **Q1**: First-Come-First-Served scheduling (5ms allocation)
3. **Q2**: Priority Scheduling (priority = inverse of remaining time, 5ms allocation)
4. **Q3**: Shortest Job First scheduling (5ms allocation)

Processes move through these queues until completion, with unfinished processes cycling back to Q0.

Features

- Supports both default and custom process configurations
- Thread-safe implementation using pthreads
- Detailed scheduling visualization
- Calculates individual turnaround times and average turnaround time

Requirements

- Linux/Unix environment
- GCC compiler
- pthread library

Compilation

To compile the program, run:

```
gcc mlfq.c -o mlfq -lpthread
```

Usage

Run the compiled program:

```
./mlfq
```

When run, the program will prompt you to choose between:

1. Using default processes (A=300ms, B=150ms, C=250ms, D=350ms, E=450ms)
2. Entering custom processes

For custom processes, you'll need to specify:

- Number of processes (1-10)
- For each process:
 - Single-character ID
 - Total CPU time needed

Expected Output for Default Configuration

Copy

===== Final Results =====

Process Total Time Turnaround Time

A	300	1290
B	150	730
C	250	1135
D	350	1395
E	450	1500

Average Turnaround Time: 1210.00

Implementation Details

- Uses pthreads for process simulation and scheduling
- Implements thread-safe queues with mutex locks
- Simulates CPU execution with usleep()
- Tracks process state through multiple queue levels
- Provides detailed logging of scheduling decisions

Cleanup

The program handles all memory cleanup and mutex destruction automatically.

Notes

- The scheduling output can be verbose as it shows all queue transitions
- For accurate timing, run on a Linux system with minimal background processes
- The default configuration should produce an average turnaround time of 1210ms