Multilevel Feedback Queue (MLFQ) Scheduling Simulation

Description

This program simulates **Multilevel Feedback Queue (MLFQ)** scheduling using **pthreads** in C. The program takes user input for the total number of jobs and the total CPU time required for five processes and executes them through multiple scheduling algorithms in a loop until all jobs are completed. The scheduling algorithms used are:

- **Round Robin (RR)**
- **First Come First Serve (FCFS)**
- **Priority Queue**
- **Shortest Job First (SJF)**

The program calculates and displays the **turnaround time** for each job and the **average turnaround time** once all jobs are executed.

Compilation and Execution

To compile and run the program, follow these steps:

1. **Compile the code:**

gcc MLFQ.c

2. **Run the program:**

./a.out

Input Specification

When you run the program, it prompts you to enter the **total number of jobs and the CPU time** needed for each process. You will be asked to enter the time for five processes one by one.

Example Input:

Enter the number of jobs (max 256): 5

For process A Total CPU time needed: 300

For process B Total CPU time needed: 150

For process C Total CPU time needed: 250

For process D Total CPU time needed: 350

For process E Total CPU time needed: 450

Execution Flow

The program executes the jobs in multiple scheduling queues, repeating the process until all jobs are fully executed.

Example Execution Output:

Round Robin

Job A executed for 5 ms, remaining: 295 ms

Job B executed for 5 ms, remaining: 145 ms

Job C executed for 5 ms, remaining: 245 ms

Job D executed for 5 ms, remaining: 345 ms

Job E executed for 5 ms, remaining: 445 ms

First Come First Serve

Job A executed for 5 ms, remaining: 290 ms

Job B executed for 5 ms, remaining: 140 ms

Job C executed for 5 ms, remaining: 240 ms

Job D executed for 5 ms, remaining: 340 ms

Job E executed for 5 ms, remaining: 440 ms

Priority Queue

Job B executed for 5 ms, remaining: 135 ms

Job C executed for 5 ms, remaining: 235 ms

Job A executed for 5 ms, remaining: 285 ms

Job D executed for 5 ms, remaining: 335 ms

Job E executed for 5 ms, remaining: 435 ms

Shortest Job First

Job B executed for 5 ms, remaining: 130 ms

Job C executed for 5 ms, remaining: 230 ms

Job A executed for 5 ms, remaining: 280 ms

Job D executed for 5 ms, remaining: 330 ms

Job E executed for 5 ms, remaining: 430 ms

Moving back to queue0....

This process continues until all jobs are completed.

Final Output

Once all jobs finish executing, the program displays the **turnaround times** for each job and the **average turnaround time**.

Example Final Output:

Jobs Turnaround Times:

Job A Turnaround Time: 1290 ms

Job B Turnaround Time: 730 ms

Job C Turnaround Time: 1135 ms

Job D Turnaround Time: 1395 ms

Job E Turnaround Time: 1500 ms

Average Turnaround Time: 1210.00 ms

Explanation

- **Turnaround Time**: The total time taken from the job's submission until its completion.
- **Average Turnaround Time**: The sum of all jobs' turnaround times divided by the number of jobs.

Key Features

- Uses **multithreading** with pthread to simulate concurrent execution of jobs.
- **Mutex locking** ensures thread-safe execution.
- Implements multiple scheduling algorithms for job execution.
- Displays individual and average turnaround times.