Week 2 Report: MLFQ Scheduler Core Implementation

In Week 2, we successfully implemented the core Multi-Level Feedback Queue (MLFQ) scheduler in xv6's proc.c, replacing the default round-robin scheduler with a 4-level priority queue system.

Implementation Details

4-Level Priority Queue Structure

We implemented four distinct priority queues (Q0-Q3) with the following characteristics:

Queue	Priority	Time Slice	Target Workload
Q0	Highest	1 tick	Interactive/I/O-bound
Q1	High	2 ticks	Light CPU usage
Q2	Medium	4 ticks	Moderate CPU-bound
Q3	Lowest	8 ticks	Heavy CPU-bound

Modified Process Structure

Added the following fields to struct proc in proc.h:

- queue_level - Current queue level (0-3)
- time_slices - Number of time slices consumed
- wait_time - Total time spent waiting
- runtime_total - Total CPU time consumed

Scheduler Logic

Implemented in scheduler() function in proc.c:

1. **Queue Selection**: Always checks queues from Q0→Q3, selecting the first RUNNABLE process from the highest non-empty queue
2. **Time Slice Enforcement**: Each process runs for its allocated quantum before being preempted
3. **Demotion Rule**: Process demoted to next lower queue if it exhausts its full time slice
4. **Yield Behavior**: Process remains in current queue if it yields voluntarily before quantum expires

Round-Robin Within Queues

Each queue operates as a circular buffer with round-robin scheduling, ensuring fairness among processes at the same priority level.

Testing Results

Test 1: CPU-Bound Process Demotion

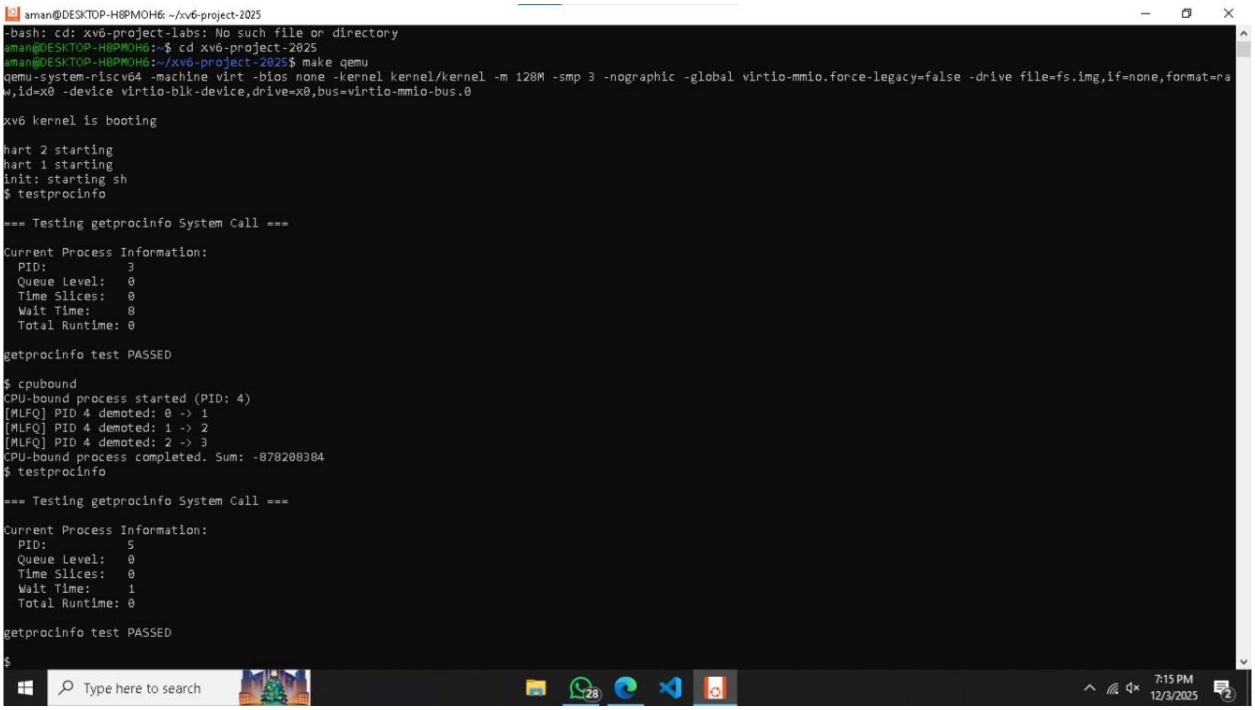
- Created cpubound.c test program with intensive computation loop
- **Observation**: Process started at Q0 and progressively demoted through Q1→Q2→Q3
- **Result**: Correctly penalizes CPU-intensive workloads

Test 2: I/O-Bound Process Behavior

- Created iobound.c test program with frequent I/O operations
- **Observation**: Process remained in Q0/Q1 due to voluntary yields
- **Result**: Rewards interactive processes with higher priority

Test 3: Round-Robin Validation

- Ran multiple processes within the same queue level
- **Observation**: Processes rotated fairly within each level
- **Result**: Confirms proper round-robin implementation



```
-bash: cd: xv6-project-labs: No such file or directory
aman@DESKTOP-H8PMOH6:~$ cd xv6-project-2025
aman@DESKTOP-H8PMOH6:~/xv6-project-2025$ make qemu
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -global virtio-mmio.force-legacy=false -drive file=fs.img,if=none,format=ra
w,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0

xv6 kernel is booting

hart 2 starting
hart 1 starting
init: starting sh
$ testprocinfo

=== Testing getprocinfo System Call ===

Current Process Information:
  PID:           3
  Queue Level:   0
  Time Slices:   0
  Wait Time:     8
  Total Runtime: 0

getprocinfo test PASSED

$ cpubound
CPU-bound process started (PID: 4)
[MLFQ] PID 4 demoted: 0 -> 1
[MLFQ] PID 4 demoted: 1 -> 2
[MLFQ] PID 4 demoted: 2 -> 3
CPU-bound process completed. Sum: -878208384
$ testprocinfo

=== Testing getprocinfo System Call ===

Current Process Information:
  PID:           5
  Queue Level:   0
  Time Slices:   0
  Wait Time:     1
  Total Runtime: 0

getprocinfo test PASSED

$
```

Conclusion

Week 2 deliverables completed successfully. The MLFQ scheduler correctly differentiates between CPU-bound and I/O-bound workloads, applying appropriate prioritization through dynamic queue management.