Phase 1 Report

MLFQ Scheduler (xv6)

1. Introduction

This phase focuses on understanding the default xv6 scheduler, designing the Multi-Level Feedback Queue (MLFQ) scheduling policy, and implementing/testing the getprocinfo system call. The system call allows a user program to retrieve scheduling-related information about any process, essential for validating MLFQ behavior.

2. System Call Implementation: getprocinfo

2.1 Overview

A new system call getprocinfo(int pid, struct procinfo *info) was added, returning PID, state, queue level, time slices, wait time, and total runtime.

2.2 Testing Strategy

(testprocinfo.c)

1. **Testing the current process:**
   The program retrieves its own PID using `getpid()` and calls `getprocinfo()` on it. The printed output includes PID, state, queue level, time slices, wait time, and total runtime. This confirms that the syscall works for normal user processes.
2. **Testing the init process (PID 1):**
   The program then queries PID 1 to ensure that the syscall correctly handles kernel-created processes. This verifies that the system call is not restricted to user processes only.
3. **Testing an invalid PID:**
   Finally, `getprocinfo()` is called with a non-existent PID (e.g., 9999). The program expects the syscall to fail and return an error. This validates that error handling is implemented correctly.

3. Testing & Verification

After running the syscall, results showed:

- valid PIDs return expected info,

- PID 1 behaves correctly,

- invalid PIDs trigger an error.

```
aman@DESKTOP-H8PMOH6:~/xv6-project-2025$ make qemu
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -
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 ===

Process Information:
  PID:           3
  State:         RUNNING
  Queue Level:   0
  Time Slices:   0
  Wait Time:     0
  Total Runtime: 0

Init Process Information:
  PID:           1
  State:         SLEEPING
  Queue Level:   0

 Correctly failed for invalid PID

$
```

4. MLFQ Scheduler Design

4.1 Queue Structure

Three-level MLFQ:

- Q0: highest priority

- Q1: medium

- Q2: lowest

4.2 Time Quantums

- Q0: 1 tick

- Q1: 2 ticks

- Q2: 4 ticks

4.3 Promotion / Demotion Policy

- If a process consumes its full quantum, it is **demoted** to the next queue.
- If it **yields early**, it stays in the same queue.
- Starvation prevention:
  Processes waiting too long in lower queues are promoted back to Q0.

4.4 Rationale

- Ensures responsiveness for interactive tasks.
- Prevents starvation of CPU-bound tasks.
- Keeps high-priority queues short and fast.