OPERATING SYSTEMS

Dr. Leili Farzinvash

Assignment 1

Due: April 19, 2025

Email: zamaninikan@gmail.com

**Note: Submit your assignment via Quera by 11:59pm on the due date. make sure your answer is typed and not handwritten.*

Questions

1. **Preemptive Priority Scheduling Analysis**

   Consider a preemptive priority scheduling system where:

   - Processes begin with priority 0 (higher numbers = higher priority)
   - Waiting processes (ready queue) have priorities changing at rate α
   - Running processes have priorities changing at rate β

   Analyze these cases and what algorithm they are equal to:

   (a) What scheduling approach emerges when $\beta > \alpha > 0$?

   (b) What scheduling approach emerges when $\alpha < \beta < 0$?

2. **CPU Scheduling Algorithm Relationships**

   CPU scheduling algorithms often exist as parameterized sets:

   - RR requires a time slice parameter
   - Multilevel feedback queues need parameters for:
     - Queue count
     - Per-queue scheduling methods
     - Process migration rules between queues

   These algorithm sets may overlap (e.g., FCFS = RR with infinite quantum). Examine relationships between:

   (a) Priority vs SJN

   (b) Multilevel feedback queues vs FCFS

   (c) Priority vs FCFS

   (d) RR vs SJN

3. **C Program Output Analysis**

   Determine the complete output of this C program:

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int value = 11;

int main() {
    pid_t pid;
    pid = fork();
    value += 5;

    if (pid == 0) { /* child */
        printf("%d", value);
        value += 9;
        return 0;
    }
    else { /* parent */
        value += 2;
        wait(NULL);
        printf("%d", value);
        return 0;
    }
}
```

4. **Layered OS Design Alternative**

   Standard layered OS designs enforce strict hierarchy (each layer only uses immediate lower layer). Propose an alternative where performance-critical operations can bypass layers, then discuss the trade-offs of this approach.

5. **CPU Utilization Calculation**

   Given:

   - Average CPU burst time: 6ms
   - Context switch time: 0.5ms
   - Round Robin quantum: 4ms

   Calculate the theoretical maximum CPU utilization achievable under these conditions.

6. **Process Creation Analysis**

   For this code segment:

```c
for (int i = 0; i < 3; i++) {
    fork();
    if (i % 2 == 0) {
        fork();
    }
}
```

   Determine the total number of processes created (including the original), illustrating your answer with a process tree derivation.

7. **Multithreading Performance Analysis**

   Provide:

   1. Two programming scenarios where multithreading improves performance compared to single-threaded implementations
   2. Two scenarios where multithreading decreases performance compared to single-threaded implementations

Good luck!