Your name:

CSE 30341 Operating Systems: Module 2 Exam

OPEN BOOK, OPEN NOTES, CLOSED ELECTRONIC SEARCHES

INDIVIDUAL EFFORT

DURATION: 30 MINUTES

All questions carry equal weight

---

1. Describe a reason why you would prefer a single threaded program in a multi-processor machine. Describe a reason why you prefer a multi-threaded program in a single processor machine.

2. Consider the following program:

```
global int matrix[100][100];
Thread 1: {
    for (int row = 0; row < 50; row++)
        for (int col = 0; col < 50; col++)
            matrix[row][col] = matrix[row][col]*2;
}
Thread 2: {
    for (int row = 50; row < 100; row++)
        for (int col = 0; col < 50; col++)
            matrix[row][col] = matrix[row][col]*2;
}
Thread 3: {
    for (int row = 0; row < 50; row++)
        for (int col = 50; col < 100; col++)
            matrix[row][col] = matrix[row][col]*2;
}
Thread 2: {
    for (int row = 50; row < 100; row++)
        for (int col = 50; col < 100; col++)
            matrix[row][col] = matrix[row][col]*2;
}
```

Suppose you were running this program on a machine with a single processor. Do you need to use any synchronization primitives for the correct execution of this program? Does your answer change if you ran this program on a 16 processor machine? Explain.

Your name:

3. Suppose an operating system was configured to minimize average waiting time. Would you prefer the random scheduling policy or shortest-next-CPU-burst algorithm? Why?

4. Some systems provide reader-writer locks. On systems that do not provide reader-writer locks, can we safely replace these locks with regular mutex locks? We define safety by the ability to provide the same critical section functionality as “reader-writer” locks.

5. Critical sections are not a concern in single processor systems that use non-preemptive scheduling. True or False? Explain.

6. Round-robin schedulers maintain a *ready list* of all runnable threads, with each thread listed at most once in the list. Listing a thread twice on the list would allow the thread to get twice the amount of CPU without creating any additional problems. True or False? Explain.