Important: This is an open book test. You can use any books, notes, or paper but no electronic device. *Do not log into the computer during the test, or use any electronic or communications device. Change your cell phones to silent mode.* Any calculations and rough work can be done on the back side of the test pages. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem. Please write legibly; if I cannot read what you wrote, I'll give you a zero. You will lose five points for not writing your name.

1. [6 pt] System calls are called just like other functions. Why cannot I write code to duplicate the functionality of, or redefine, a system call?

2. [6 pt] Consider the following code:

```
for ( i = 0; i < 20; i++ )
    for ( j = 0; j < 10; j++ )
        a[i] = a[i] * j;
```

   (a) Give one example of the spatial locality in the code.

   (b) Give one example of the temporal locality in the code.

3. [6 pt] How do you ensure that machine-level instructions are executed atomically?

4. [6 pt] In the beginning, we considered the two terms *kernel* and *operating system* interchangeably. However, there is a fine distinction between the two. Enumerate at least two points that distinguish between a kernel and an OS.

5. [6 pt] What is expected for an operating system to be classified as real-time OS? What is the expectation for an OS for an embedded system?

6. [10 pt] Consider the following program running on a unicore machine:

```
const int n = 50;
int tally;
void total()
{
    for ( int count = 0; count < n; count++ )
        tally++;
}
%
void main()
{
    tally = 0;
    parbegin
        total();
        total();
    parend;
    printf ( "%d\n", tally );
}
```

(a) Determine the proper lower bound and upper bound on the final value of the shared variable `tally` output by this concurrent program. Assume processes can execute at any relative speed and that a value can only be incremented after it has been loaded into a register by a separate machine instruction.

(b) Suppose that an arbitrary number of processes are permitted to execute in parallel under the assumptions of part (a). What effect will this modification have on the range of final value of `tally`?

7. [6 pt] Linux allows any user to write code that will grant them kernel mode privilege. Why does it need two modes if any user can execute code in kernel mode?

8. [6 pt] In a number of early computers, an interrupt caused the register values to be stored in fixed locations associated with the given interrupt signal. Under what circumstances is this a practical technique? Explain why it is inconvenient in general.

9. [6 pt] Demonstrate the correctness of Peterson's algorithm to solve the critical section problem.