Systems Q Exam (January 1999)
Computer Science Dept., Indiana University

Operating Systems

1. (20 pts) Assume preemptive scheduling, i.e., assume a timer interrupt may occur between any two machine instructions, and that timer interrupts may cause context switches. Initially, address x contains 2, and address y contains 10. Assembly-language pseudo-code is given for two threads, T1 and T2, which execute concurrently.

```
T1:  Load x, r1    // load contents of address x into register r1
     Load y, r2    // load contents of address y into register r2
     Add r2, r1    // r1 := r1+r2
     Store r1, x   // store contents of register r1 into address x

T2:  Load x, r1    // load contents of address x into register r1
     Load y, r2    // load contents of address y into register r2
     Add r2, r1    // r1 := r1+r2
     Store r1, y   // store contents of register r1 into address y
```

What are the possible pairs of final values of x and y, i.e., the values after both threads have finished? (Just to clarify the English, the possible pairs of values might be something like: (x=1 and y=2) or (x=2 and y=3).)

2. (25 pts) a. Describe the way(s) in which DMA is typically used in implementations of operating systems (including file systems and network protocols).

b. Describe the interactions (including synchronization issues), if any, between DMA and virtual memory.

3. (25 pts) Describe how to implement swapping. Include descriptions of when and why processes get swapped in or out. Discuss the interaction, if any, between swapping and CPU scheduling.

4. (30 pts) Consider a traditional UNIX fileystem (with inodes, indirect blocks, etc.). Describe how to implement the system call CAwrite(fd, buf, pos, len), which writes len bytes of data from buf into the file specified by file descriptor fd, starting at position pos in the file; furthermore, this write is guaranteed to be *crash-atomic*, i.e., if the computer crashes during this operation, then when the computer reboots after the crash, the file contains all of the new data or none of it. If your implementation of CAwrite affects the data structures or algorithms used by other system calls, describe the effects. You can assume that writing a single disk sector is crash-atomic (in a similar sense). You can assume that the sizes of inodes, sectors, and blocks are powers of two. Of course, len might be larger than the sector size! You are free to make additional assumptions, provided they are reasonable and are stated explicitly.

Computer Architecture

```
1. a. ( 5) Define a RISC and a superscalar computer architecture.
   b. ( 5) Can a RISC architecture be superscalar, or vice versa?
           Explain, giving specific reasons.

2. a. ( 5) List the steps in the instruction execution cycle.
   b. (10) Explain what occurs in each step.

3. a. ( 5) List the four data hazards.
   b. (10) Explain why each can or cannot be ignored in the design
           of a pipelined computer architecture.
```

c. (15) For each hazard that cannot be ignored, if any, name two techniques used to eliminate the hazard.

4. a. (5) List five operand addressing modes.
   b. (15) Explain how each operates, assuming there is only one ALU in the CPU, and that instruction memory and data memory are not kept separately and are accessed using a single bus.
   c. (20) Make a table that compares the relative speed of each addressing mode, assuming one clock cycle for each use of the ALU, each register fetch, each MAR and MDR load/store, and each memory read and write.