Department of Computer Science
Operating Systems Qualifying Exam

Fall, 2000

The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish, but **no calculators** are allowed. On all questions, the reasoning you use to obtain your answer is more important than the actual answer; if your answer to any question is simply a number (or a yes/no) without justification, it is wrong (even if the number happens to be correct, or the answer really is ‘‘yes’’ or ‘‘no’’).

1. Here’s a protocol purported to implement mutual exclusion:

   - An array in shared memory has an entry for every process that will want access to the critical section.
   - Exactly one element of the array contains a 1; every other array element contains a 0. The process corresponding to the array element with the 1 is said to be ‘‘holding the token.’’
   - Whenever a process wishes to enter the critical section, it checks to see if it holds the token. If it holds the token, it is able to enter the critical section. If it does not hold the token, it busy−waits until it does have the token. Once it is done with the critical section, it passes the token to the next process by setting its own array element to 0, and setting the next process’s array element to 1.

   Does this protocol meet the complete definition of a mutual exclusion algorithm?

2. Find a scheduling for the following set of real−time processes, which meets all of their deadlines.

Process	*Arrival Time*	*Processing Time*	*Deadline*
P1	0	10	15
P2	5	5	10
P3	10	10	30
P4	10	5	20

3. Suppose that in some message−passing multiprocessor computer using distributed shared memory, memory location M’s home (which does not migrate) is located on Processor 1, and Processor 1 initially has the only valid copy of M. Now suppose Processor 2 and Processor 3 simultaneously attempt to read and then modify M.

   Describe a possible sequence of messages between Processors 1, 2, and 3 in this scenario.

4. Given the following page reference string:

   1, 2, 9, 4, 9, 2, 7, 4, 7, 2, 6, 5, 6, 2, 1

   If the working set is three page frames, and it is initially empty, which of these accesses will cause page faults if an LRU replacement strategy is used?

5. Normally, a program requests operating system services by using a system call instruction (which may go by other names such as software interrupt).

   Would it be possible to just use an ordinary procedure call instruction, trying to call a procedure in the system space (which would not be user−executable), to perform system calls?