Department of Computer Science
Qualifying Examination
Operating Systems
Spring, 2002

The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish. Please note the following instructions.

- show your work whenever appropriate. There can be no partial credit unless I see how answers were obtained.
- be succinct. I will take points off for facts that, while true, are not relevant to the question at hand.

1. (30 points) Consider a network-based multiprocessor computer system using distributed shared memory, with four processors. Initially, variable a is on processor P1 and b is on P2; these are also the variables' home processors (and, while the variables can migrate as needed, their homes are fixed).

   Suppose the processors execute the C code shown in the following table. Event 1 is completely executed before event 2 begins, and so forth:

Event	Processor	Code
1	P2	`a = 3;`
2	P3	`b = a + 2;`

   (a) Give a possible sequence of messages which could be used by the DSM system to implement strict memory consistency. We don't just want the data transfers, we also want the required requests for locks and data.

   (b) Translate the code for use in a message-passing system, assuming you have the functions

   `send(`*`processor`*`, `*`variable`*`)` to send a value to another processor, and

   `variable` = `receive(`*`processor`*`)` to receive a variable from another processor.

2. (20 points) The following is almost a description of Peterson's Algorithm for two-process mutual exclusion. The parameter `i` is the process number; the two processes are numbered 0 and 1.

```
shared int flags[2] = {0, 0};
shared int turn;
enter(int i)
{
    turn = 1-i;
    flags[i] = 1;
    while ((flags[1-i] == 1) && (turn == (1-i))
        {/*empty*/}
}
leave(int i)
{
    flags[i] = 0;
}
```

   (a) How does this algorithm differ from Peterson's Algorithm? *Note: we're looking for a substantive difference in the algorithm, not a difference in presentation like using* `int` *where some presentations would use a boolean type.*

   (b) Show an interleaving of two processes attempting to enter their critical sections that will fail with this modified algorithm. "Fail" may mean either that both processes succeed in entering the critical section at the same time, or that both processes get stuck in their `while` loops and neither will ever enter.

3. (20 points) The following table gives the arrival times, processing times, and deadlines for five processes in a preemptive real-time system. Schedule the processes so they all meet their deadlines.

Process	Arrival Time	Processing Time	Deadline
1	0	10	20
2	5	5	10
3	10	15	45
4	20	5	30
5	20	10	35

4. (20 points) Suppose you had a network of workstations implementing a shared file system. The file system's data is distributed so that block i is located on workstation $i \, mod \, n$ where n is the number of workstations: block 0 is on workstation 0, block 1 is on workstation 1, and so forth until block $n-1$ is on workstation $n-1$; then block n is on workstation 0 again and so on.

   How could you use one extra workstation to gain some fault tolerance: in particular, so that if you lost the data from any one of the workstations (including the new one), it would be possible to reconstruct the file system from the remaining data? You can assume that the workstations either provide good data or fail completely; they will not provide corrupt data. *Hint: think about parity.*

5. (10 points) "Scientific computing" typically involves performing operations such as convolutions or matrix multiplications on very large arrays. Explain why a Most Recently Used page replacement strategy is likely to be a better approximation to the optimal replacement algorithm than LRU is in this case.