Department of Computer Science
Operating Systems Qualifying Exam
August, 2002

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. Please note the following instructions.

- show your work. The reasoning you use to get your answers is far more important than the answers themselves; an answer with no justification is **wrong**.
- be succinct. I will take off points for facts that, while true, are not relevant to the question at hand

1. (40 points) In a distributed shared memory system using lazy release consistency, the two underlying syncronization primitives are `Acquire` and `Release`: `Acquire(`*object*`)` obtains access to an object, and `Release(`*object*`)` gives up that access. Suppose we were to add a new primitive, `ReleaseTo`. `ReleaseTo(`*object*`,` *proc*`)` would both release the object, and include a hint that the programmer expects *proc* to `Acquire` it next.

   Why could we expect the addition of `ReleaseTo` to reduce the number of messages transferred between processors? Your answer should include a discussion of the messages required to implement `Acquire` and `Release`, and show how `ReleaseTo` will eliminate the need for some of them.

2. (20 points) Suppose the following piece of code is executed on two processors:

```
shared int i=0, j=0;
i = i + 1;
j = i + 1;
```

   (a) Show that the final value of j can be either 2 or 3, depending on how reads and writes are interleaved. Use the following notation:

```
P1:  R(x)0
-----------------------------
P2:
```

   (b) Insert synchronization operations of your choice which will guarantee the final value of j is 3. The synchronizations may be inserted in different places in the code executed by the two processors.

3. (40 points) The following table gives the number of units and maximum allocation of resources in a computer system (the Total row is not the sum of the other three, it reflects the total number of units of the resource in the system).

	Resource 1	Resource 2	Resource 3	Resource 4
Process 1	1	3	2	1
Process 2	3	0	1	1
Process 3	1	2	3	1
Total	4	3	5	1

The next table shows the current allocation of resources.

	Resource 1	Resource 2	Resource 3	Resource 4
Process 1	1	2	1	1
Process 2	2	0	1	0
Process 3	0	0	3	0

(a) Is the system in a safe state?

The next table shows the currently pending resource requests.

	Resource 1	Resource 2	Resource 3	Resource 4
Process 1	0	1	1	0
Process 2	1	0	0	0
Process 3	1	0	1	0

(b) Is the system deadlocked?

(c) Which of the outstanding requests should be granted?