Homework 3
Due: in-class, Oct 17th 2012

Problem 1: (20 Points)
Given a 512-processor system in which each node is visible to the directory has 8 processors and 1 GB of main memory and a cache block is of size 64 bytes, what is the directory memory overhead for (a) a full bit vector scheme and (b) Dir,B, with i = 3?

Problem 2: (20 Points)
Instead of the doubly linked list used in the SCI protocol, it is possible to use a singly linked list. What is the advantage? Describe the modifications would be needed to make the following operations possible with a singly linked list were used:

(a) Replacement of a cache block that is in the sharing list
(b) Write to a cache block that is in a sharing list

Problem 3: (20 Points)
Why is write atomicity more difficult to provide with update protocols than with invalidation-based protocols in directory-based systems? How would you solve the problem? Does the same difficulty exist in a bus-based system?

Problem 4: (20 Points)
Consider the following program fragment running on a cache-coherent multiprocessor, assuming all values to be 0 initially,

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>u=A</td>
<td>w=A</td>
<td></td>
</tr>
<tr>
<td>v=A</td>
<td></td>
<td>x=A</td>
<td>A=2</td>
</tr>
</tbody>
</table>

There is only one shared variable (A). Suppose that a writer magically knows where the cached copies are and sends to them directly without consulting a directory node. Construct a situation in which write atomicity may be violated assuming an update-based protocol.

(a) Show the violation of sequential consistency that occurs in the results.
(b) Can you produce a case where coherence is violated as well? How could you solve these problems?
(c) Can you construct the same problems for an invalidation-based protocol?
(d) Can you construct them for update protocols on a bus?

Problem 5: (20 Points)
Read the following papers and write a critique: