Homework 3 (due 11/20/2008 at 12:05pm)

via email to ada@cc or hardcopy in class (preferred)

1. Explain the performance of the 'backoff ref' spinlock algorithm observed in Figure 3 in Anderson's Spinlock paper. Make sure to explain why this algorithm performs better/worse compared to other spinlock implementations at low/high loads.

2. Consider the URPC system implemented on a 4-processor SMP. There is a multi-threaded client (with 3 threads T1, T2, and T3) that currently has 2 processors. There is a file server that currently has 1 processor, and a graphics server that currently has 1 processor. T1 and T2 are currently running on the processors assigned to the client. T3 is a compute intensive thread that is always ready to run. The scheduling discipline in the client threads package is: Run a thread until it makes a URPC call; schedule any runnable thread; if no runnable thread then donate the processor to an under-powered server with which one of its thread is having a URPC session.
The following events happen: T1 makes a file (URPC) request at time 2; T2 makes a graphics (URPC) request at time 5; T2 makes a file (URPC) request at time 11. Assume each file request takes 10 time units, and each graphics request takes 5 units of time. Also assume processor transfer takes 5 units of time in either direction (client to server or vice versa). Show a time line of scheduling actions that take place in the system with respect to the above execution.
Note: if you make any assumptions make sure (a) to list them explicitly, (b) that they don't contradict with anything stated above.

3. In the remote procedure call (RPC) paper, in several places, the implementation makes use of the fact that calls are synchronous and a client does not make the next call before the result for the pending call is returned. In many applications, an asynchronous RPC is useful which allows a process to make a call and then collect the results at a later time. Thus, the process is able to execute in parallel with the execution of the call at the server node. Since the process may issue multiple asynchronous calls before the first one returns, it is no longer possible that the runtime deals with at most one call between a client-server pair. (Multithreading can also result in multiple outstanding calls even when calls are synchronous).
Discuss how the design and implementation presented in the paper will have to be changed to accommodate asynchronous RPCs. In particular, discuss the data structures maintained at client and server side and describe the changes that will be necessary.