Final Exam

Due Date: Tuesday, March 20, 2000 at 12:30PM (the end of the reglar final exam period for this class)
If you want to submit this electronically, please email it to me at bishop@cs.ucdavis.edu. Also, please send both PDF/ASCII/Postscript *and* the original format (Word, Framemaker, whatever) as attachments, just in case I cannot read the PDF/ASCII/Postscript.

1. (*70 points*) Consider a network in which the sites are arranged into a single graph for purposes of implementing a distributed mutual exclusion algorithm. (In other words, the network is *not* fully interconnected.) There are n sites, and site k is connected to $k_i \leq n$ other sites. This question asks you to explore some of the issues raised by this organization. For this question, assume each site is running at most one process that will try to gain access to the region of mutual exclusion.
   a. Please generalize Raymond's token-based algorithm to implement mutual exclusion in this network. State any initial conditions that must be met, or any necessary initializations, for your algorithm to work. When you write your algorithm, please use a format such as we did in class, in which each step is numbered and the reader can simply follow the steps. Working an example would help!
   b. Please show that your algorithm correctly satisfies the three requirements for a solution to the critical section problem: mutual exclusion, progress (no deadlock), and bounded wait (no starvation).
   c. Raymond's algorithm solves a special case of this problem (where the graph is a tree structure). Assume now you imposed a tree structure upon this topology and used Raymond's algorithm to implement mutual exclusion for comparison purposes. How does the average number of messages required for a site to acquire the token using your algorithm compare to that of Raymond's algorithm?
2. (*30 points*) Let n processes share m resource units, which can be acquired and released only one at a time. The maximum number of resource units that any single process will need will always be no more than m, and the maximum number of resource units that all n processes will need is less than $m + n$. Prove that deadlock cannot occur under these conditions.