Homework #1

Due Date: Tuesday, February 8, 2000 at 11:59PM
Points: 100

1. (*20 points*) The following table shows domains and ranges for a system of processes along with known precedence constraints:

process	p_1	p_2	p_3	p_4	p_5	p_6	p_7
domain	v_5	v_1, v_7	v_4	v_4, v_8	v_2	v_3	v_4
range	v_4, v_7	v_5	v_6	v_2	v_1, v_3	v_5	v_5, v_8
preceded by		p_1	p_1	p_2	p_3	p_3	p_4, p_6

   Add the minimum number of precedence constraints to make this system of processes determinate. Do not remove any constraints.

2. (*26 points*) For a semaphore s, define:

   init[*s*] is the initial value of *s*;
   start_down[*s*] is the number of times *down*(*s*) has been started;
   end_down[*s*] is the number of times *down*(*s*) has been completed; and
   end_up[*s*] is the number of times *up*(*s*) has been completed.

   A useful semaphore invariant is:

   $end\_down[s] = \min(start\_down[s], init[s] + end\_up[s])$;

   Show that

   $0 \leq end\_down[empty] - end\_down[full] \leq n$

   for the version of the producer-consumer solution using semaphores in the handout ***Process Synchronization and Communication***, p. 11.

3. (*14 points*) Indicate how each of the following items could be incorporated in the monitor mechanism (a sentence or two for each is sufficient; you do not have to show an implementation).
   a. type of request
   b. times at which the requests were made
   c. request parameters
   d. process information
   e. priority relation
   f. local state of resource
   g. history information
4. (*20 points*) Show that the *ordered request policy* of Havender prevents deadlocks (text, problem 3.4).
5. (*20 points*) Given that the mutual exclusion, hold and wait, and no preemption conditions are in place, consider the following strategy: All processes are given unique priorities. When more than one process is waiting for a resource and the resource becomes available, allocate the resource to the waiting process with the highest priority. Either prove this prevents deadlock or give an example in which this strategy does not prevent deadlock.

Extra Credit

6. (*10 points*) (Tanenbaum) Cinderella and the Prince are getting divorced. To divide their property, they have agreed on the following algorithm. Every morning, each one may send a letter to the other's lawyer requesting one item of property. Since it takes a day for letters to be delivered, they have agreed that if both discover that they have requested the same item on the same day, the next day they will send a letter cancelling the request. Among their property is their dog, Woofer, Woofer's doghouse, their canary Tweeter, and Tweeter's cage. The animals love their houses, so it has been agreed that any division of property separating an animal from its house is invalid, requiring the whole division to start over from scratch. Both Cinderella and the Prince desperately want Woofer. So they can go on (separate) vacations, each spouse has programmed a personal computer to handle the negotiation. When they come back from vacation, the computers are still negotiating. Why? Is deadlock possible? Starvation? Discuss.