Systems Programming and Operating Systems	**Assignment 2**
CS414, Summer 1996	**July 17, 1996; Due on July 25, 1996**

Problem 1 (24 points)

Using the approach shown on the last page of class notes 4, prove that in the final solution for the two process critical section problem satisfies:

- No two processes are in the critical section at the same time

Prove also that

- There is no starvation
- If two processes are trying to enter the critical section at the same time, only finite time elapses before one of them enters.
- A process not in the critical section cannot prevent another from entering the critical section.

Problem 2 (24 points)

Let x and y be integers such that x = 20 and y = 30. Let there be 2 semaphores s_x and s_y initialized to 0 and 1 respectively. Each of the following problems has an example of two processes (P1 and P2) running in a system at the same time. In each case, show the final values possible for x. In each case, explain how you arrived at the possible answers.

- P1: x = x+1, P2: x = y+1
- P1: signal (s_x), x = x+1, wait (s_x), P2: signal (s_x), x = y+1, wait (s_x)
- P1: wait (s_x), x = x+1, signal (s_x), P2: wait (s_x), x = y+1, signal (s_x)

Problem 3 (12 points)

- If we have a single processor with no interrupts and no preemption, then how should the solutions to the critical section be modified? why?
- If your critical sections were very short, would you use a busy waiting mechanism or a sleep-wakeup mechanism? What if your critical sections were very long? Why?