CS 414	**Operating Systems**	Test 2
Name:	Fall 2002	Max Pts: 50

Important: This is an open book test. You can use any books, notes, or paper. *You are not allowed to use any communications device. Do not log into the computer during the test.* Any calculations and rough work can be done on the back side of the test pages. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem. You will lose five points for not writing your name.

1. [5 pt] The code for wait in general semaphore is given as:

```
void P()
{
    mutex.P();
    if ( --count < 0 )
    {
        mutex.V();
        delay.P();
    }
    mutex.V();
}
```

   Why do we have to send a signal on `mutex` twice if the value of `count` is negative?

2. [8 pt] Consider a system $\langle \sigma, \pi \rangle$ with $\sigma = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$ and $\pi = \{p_0, p_1, p_2\}$. State changes are:

$$
\begin{array}{l}
p_0(\sigma_0) = \{\sigma_3\}, p_0(\sigma_1) = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}, p_0(\sigma_2) = \Omega, p_0(\sigma_3) = \{\sigma_0, \sigma_1\} \\
p_1(\sigma_0) = \{\sigma_2\}, p_1(\sigma_1) = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}, p_1(\sigma_2) = \Omega, p_1(\sigma_3) = \{\sigma_0\} \\
p_2(\sigma_0) = \Omega, p_2(\sigma_1) = \Omega, p_2(\sigma_2) = \{\sigma_3\}, p_2(\sigma_3) = \{\sigma_2\}
\end{array}
$$

   Show the possible sequence of state changes. Can you identify any knot in there? Is there a possibility of a deadlock?

3. [15 pt] Assume you have the following jobs to execute with one processor:

Process	Burst time	Arrival time
p_0	9	0
p_1	2	9
p_2	3	11
p_3	13	12
p_4	13	13

Give the average wait time and average turnaround time for each process using the following algorithms. Also compute the percentage of time when the system is busy with user processes.

(a) First in first out

(b) Shortest job next (no preemption)

(c) Shortest remaining time next

(d) Round robin, with a quantum of 3

(e) Round robin, with a quantum of 4 plus context switch time of 1

4. [8 pt] Assume a system with five resource types, $C = \langle 7, 9, 9, 13, 11 \rangle$ (this is the total number of resources in the system, and not what is currently available), and the maximum claim table shown below.

Process	R_0	R_1	R_2	R_3	R_4
p_0	1	6	7	0	0
p_1	0	1	0	9	0
p_2	1	1	2	4	4
p_3	5	5	4	6	5
p_4	4	0	0	7	8

The resource allocator is considering allocating resources according to the following table:

Process	R_0	R_1	R_2	R_3	R_4
p_0	0	0	1	0	0
p_1	0	1	0	6	0
p_2	1	1	0	4	2
p_3	5	5	4	6	5
p_4	4	0	0	6	5

Run the safety algorithm on this system to determine if this state is safe.

5. [4 pt] What are the advantages of early binding compared to late binding? What are the advantages of late binding compared to early binding?

6. [10 pt] You have a physical memory of 64M, starting from address 0. Your operating system requires at least 10M all the time. Consider the arrival of processes as follows:

Process	Burst time	Arrival time	Memory needed
p_0	32	0	16M
p_1	74	13	12M
p_2	36	18	8M
p_3	65	24	20M
p_4	6	24	15M
p_5	8	28	17M
p_6	42	28	18M
p_7	43	37	17M

Show the layout of memory at times 12, 29, 47, 72, and 100.