CS 4760	**Operating Systems**	Test 2
Name:	Spring 2006	Max Pts: 48

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. [6 pt] When a process is interrupted, some information is saved in process control block. Describe the items to be saved. Why do each of them need to be saved? Should they be restored in any particular order? Why?

2. [6 pt] What is the difference between a process control primitive and a system call? Illustrate by using the `create` primitive and `fork` system call.

3. [5 pt] Does a knot in a process resource graph guarantee the existence of a deadlock? Explain your answer.

4. [6 pt] What is the difference between static and dynamic linking? What is the advantage of each of them?

5. [10 pt] Assume a system with four resource types, $C = \langle 8, 5, 7, 8 \rangle$ (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
p_0	3	1	3	4
p_1	4	2	2	1
p_2	4	5	2	2
p_3	2	2	5	1
p_4	2	2	2	2

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

Process	R_0	R_1	R_2	R_3
p_0	2	1	2	3
p_1	2	1	1	1
p_2	1	1	0	1
p_3	0	1	3	1
p_4	0	1	1	1

Run the safety algorithm on this system to determine if this state is safe. If it is safe, give the sequence in which processes can be run. If it is unsafe, enumerate the processes that may get involved in a deadlock.

6. [15pt] Assume that you have the following jobs to be executed with one processor:

Process	Burst time	Arrival time
p_0	6	0
p_1	8	3
p_2	7	6
p_3	5	7
p_4	2	9

Give the average wait time and average turnaround time for each process using the following algorithms. Is the CPU idle at any time in the given algorithms?

(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