CS 376	**Operating Systems**	Test 2
Name:	Winter 2002	Max Pts: 55

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] In almost every operating system, the interrupts are handled by kernel. Yet, we saw in our class projects that you can handle interrupts within the code. In fact, it is recommended that shell scripts perform some cleanup when an interrupt is received. Do you see a problem with what we say and what we do? Explain your answer.

2. [8 pt] Consider a system $\langle \sigma, \pi \rangle$ with $\sigma = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ and $\pi = \{p_0, p_1, p_2\}$. State changes are:

$$p_0(\sigma_0) = \{\sigma_4\}, p_0(\sigma_1) = \Omega, p_0(\sigma_2) = \Omega, p_0(\sigma_3) = \{\sigma_0, \sigma_1, \sigma_2\}, p_0(\sigma_4) = \{\sigma_1, \sigma_3\}$$
$$p_1(\sigma_0) = \{\sigma_1, \sigma_2, \sigma_4\}, p_1(\sigma_1) = \{\sigma_0, \sigma_2\}, p_1(\sigma_2) = \{\sigma_0, \sigma_2\}, p_1(\sigma_3) = \Omega, p_1(\sigma_4) = \{\sigma_0, \sigma_3\}$$
$$p_2(\sigma_0) = \Omega, p_2(\sigma_1) = \Omega, p_2(\sigma_2) = \Omega, p_2(\sigma_3) = \Omega, p_2(\sigma_4) = \{\sigma_1, \sigma_2\}$$

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	8	3
p_2	2	6
p_3	6	9
p_4	8	9

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. [12 pt] Assume a system with five resource types, $C = \langle 5, 8, 5, 12, 9 \rangle$ (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	0	2	3	5	4
p_1	3	4	5	0	3
p_2	1	1	1	0	2
p_3	5	4	1	5	1
p_4	2	1	1	4	3

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	2	1	5	0
p_1	3	1	0	0	3
p_2	0	0	1	0	2
p_3	1	4	1	4	1
p_4	0	1	1	3	3

(a) Draw a process resource graph to show the state of the system described above.

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

5. [2 pt] What is the gcc flag to generate shared objects in Unix?

6. [3 pt] Consider the memory management schemes based on paging, segmentation, and paged segmentation. Which of these schemes will benefit from partial memory compaction if available?

7. [10 pt] You have a memory of 32 frames, with each frame being 2K bytes. Current free-frame list (in order) is: 3, 11, 1E, A, 16, 6, 2, 18, 14, 5, 1D, and 1F (hexadecimal numbers). You just scheduled a process that requires 8 frames for execution. Can you allocate the frames to this process? Show the resulting page table, free frame list, and how the pages are allocated into frames by drawing a picture. Show the translation of logical addresses 0X061B and 0X1701 into physical addresses using your page table.