CS 414	**Operating Systems**	Test 3
Name:	Spring 2003	Max Pts: 60

Important: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer, or use any electronic/communications device (including calculator) during the test.* Any calculations and rough work can be done on the back side of the test pages. You will lose five points for not writing your name.

1. [8 pt] Assume a system with four resource types, $R_i = \langle 6, 9, 6, 4 \rangle$ (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	2	5	0	4
p_1	4	3	4	0
p_2	3	1	6	3
p_3	2	4	2	0

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

Process	R_0	R_1	R_2	R_3
p_0	2	2	0	4
p_1	3	2	3	0
p_2	1	1	3	0
p_3	0	4	0	0

   Run the safety algorithm on this system to determine if this state is safe. If it is safe, give the order of execution of processes. If it is not safe, specify the processes that may participate in a deadlock.

2. [4 pt] What is rollback in deadlock recovery? What problems are solved by it, and what may still left unsolved?

3. [8 pt] In a virtual memory system, 1 in 600 references (on average) causes a page fault. When the page fault is to be serviced, 1 in 40 pages have their dirty bit set. Let the average seek time for the disk be 12 milliseconds, the average latency be 1 millisecond, and the average wait time in device queue be 10 milliseconds. In addition, the transfer time per page is 1 millisecond. Let the memory access time be 40 nanoseconds when there is no page fault. Compute the effective memory access time in this system.

4. [10 pt] You have a memory of 256 frames, with each frame being 1K bytes. Current free-frame list (in order) is: DC, 97, C1, 3D, 4B, 1A, 40, 1F, AA, 34, 4E, and 53 (hexadecimal numbers). You just scheduled a process that requires 11 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 0x4638 and 0x126F into physical addresses using your page table.

5. [18 pt] If FIFO page replacement is used with 4 page frames, how many page faults will occur with the reference string

4 3 0 7 8 1 9 0 1 8 5 1 4 9 6 4 2 9 5

if the frames are initially empty. Now repeat this problem for OPT, LRU, LFU, and second chance algorithm. How will it perform with a window size of 5 under the working-set algorithm (assume unlimited number of frames available for working set algorithm but working set window size is 5)?

6. [4 pt] Unix does not have a file type to support random access of data in files. Instead, it treats all files as a stream of bytes. However, it does not cause a major problem and we can still access data in random order. How? What is the internal data structure that is loaded in memory to achieve this?

7. [8 pt] I have a floppy disk with 1.44MB [unformatted] capacity. The data blocks are 128 bytes each. I am going to create a Unix filesystem on this floppy with an empty boot block of size 1 block. Consider 1 block to be allocated for super block. Let each inode require 1024 bytes. What can be the maximum formatted capacity (total capacity of data blocks) of the floppy? What is the maximum file size that can be stored on this floppy if the system uses 12 direct blocks, 1 single indirect block, and 1 double indirect block?