CS 376	**Operating Systems**	Test 3
Name:		Maximum Points: 75

Important: This is an open book test. You can use any books, notes, or paper. *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.

1. [16 pt] Consider the following four systems A through D:

	A	B	C	D
Page size (in words)	512	512	1024	1024
Word size (in bits)	16	32	16	32

   For each system, determine

   (a) the size of the page table (number of entries), and

   (b) the size of the virtual memory (number of pages).

   Assume that only one page table exists for the entire system and each virtual address (p, d) occupies on word (16 or 32 bits).

2. [12 pt] You are given the following information about a virtual memory system that employs both paging and segmentation:

   - Word size is 32 bits
   - Page and segment table entries each occupy one word
   - Page size is 512 words
   - A page table occupies at most one page
   - Virtual addresses have the form (s_1, s_2, p, w) where $|s_1|$, $|s_2|$, $|p|$, and $|w|$ denote the lengths (in bits) of each of the four address components

   (a) What is the value of $|w|$?

   (b) What is the maximum number of pages per segment and what is the corresponding value of $|p|$?

   (c) Using the determined values of $|p|$ and $|w|$, which of the following choices for $|s_1|$ and $|s_2|$ is preferable? Explain.

      i. $|s_1| = |w|$ and $|s_2| = 32 - |s_1| - |p| - |w|$.

      ii. $|s_2| = |w|$ and $|s_1| = 32 - |s_2| - |p| - |w|$.

   (d) Would either choice under part (c) result in a larger virtual address space? Explain.

3. [17 pt] Consider the following sequence of virtual memory references generated by a single program in a pure paging system:

$$10, 11, 104, 170, 73, 309, 185, 245, 246, 434, 458, 364, 75, 12$$

(a) Derive the corresponding reference string, assuming a page size of 100 words.

(b) Determine the number of page faults for each of the following page-replacement strategies, assuming that two page frames are available to the program:

i. FIFO replacement

ii. Optimal replacement

iii. LRU replacement

iv. LFU replacement

v. Second chance algorithm

4. [12 pt] Assume the following:

- The disk's outer track is the 0th track and the disk contains 200 tracks per surface.
- A seek takes $10 + 0.1 \times T$ ms, where T is the number of tracks of motion from one request to the next, and 10 is a movement time constant.
- One full rotation takes 8 ms.
- Transfer time is 1 ms.

Compute the total time required to transfer the file using the given strategies[1]. The head is currently at track 100 moving towards track 0. Assume that the requests come in the following order:

Track requested	Sector requested
45	0
132	6
20	2
23	1
198	7
170	5
180	3
78	4
73	5
150	7

(a) FCFS scheduling

(b) SSTF scheduling

(c) SCAN scheduling

(d) LOOK scheduling

(e) C-SCAN scheduling

(f) C-LOOK scheduling

5. [10 pt] Consider a disk with c cylinders, t tracks per cylinder, s sectors per track, and a sector length l. A logical file L with fixed record length r is stored contiguously on this disk starting at location (c_L, t_L, s_L), where c_L, t_L, and s_L are the cylinder, track, and sector numbers, respectively.

[1] You will get good partial credit if you just compute the time to travel across the tracks

(a) Derive a formula to calculate the disk address (i.e., cylinder, track, sector) of a given logical record n, assuming that

   i. $r \leq l$

   ii. $r > l$

(b) How many seek operations are necessary to read an entire file of m logical records?

6. [18 pt] Consider a file currently consisting of 100 blocks. Assume that the file control block (and the index block, in case of indexed allocation) is already in memory. Calculate how many disk I/O operations are required for contiguous, linked, and indexed (single-level) allocation strategies, if, for one block, the following conditions hold. In the contiguous-allocation case, assume that there is room to grow in the end. Assume that the block information to be added is stored in memory.

   (a) The block is added at the beginning.

   (b) The block is added in the middle.

   (c) The block is added at the end.

   (d) The block is removed from the beginning.

   (e) The block is removed from the middle.

   (f) The block is removed from the end.