CS 450 Exam 3 — Mon. 5/16/2016

Name: _____

Rules and Hints

- You may use one handwritten 8.5 × 11″ cheat sheet (front and back). This is the only additional resource you may consult during this exam. *No calculators.*
- Include step-by-step explanations and comments in your answers, and show as much of your work as possible, in order to maximize your partial credit.

Grade

	Your Score	**Max Score**
Problem 1: Short answer		20
Problem 2: Concurrency		13
Problem 3: Page replacement		25
Problem 4: Virtual memory and address translation		28
Problem 5: Filesystems		14
Total		**100**

Problem 1: Short answer (20 points)

For each of the following concepts, explain its effect on functionality, correctness, and/or performance.

Memory-mapped I/O:

Direct memory access:

Virtual memory:

Translation lookaside buffer (TLB):

Problem 2: Concurrency (13 points)

You are designing a Sudoku validator with separate threads to check each row, column, and subgrid. The main thread will print a summary of the results. Explain any synchronization between threads that your program will require, and what data structure(s) you will use to communicate among threads (including between child threads and the main thread).

Problem 3: Page replacement (25 points)

Consider a process that has virtual pages A through Z and is allocated 4 physical pages, all of which are initially invalid.

Trace the sequence of references in the table below. For each reference, state:

- Whether it is a page hit (Hit) or a page fault (PF)
- If it is a page fault, which virtual page will be evicted, if any (e.g. "PF (A)")

The algorithms you will trace are FIFO replacement; LRU (least recently used); LFU (least frequently used), with LRU as a tiebreaker; a clock that only runs on a page fault; and optimal replacement.

	FIFO	LRU	LFU	Clock	Optimal
A					
A					
B					
C					
D					
E					
E					
E					
D					
B					
A					
B					
C					

Problem 4: Virtual memory and address translation (28 points)

State one benefit and one drawback of dividing memory into fixed-size pages (e.g. 4 KB).

When a process tries to access a particular virtual address, is it possible for its entry in the process page table to be invalid? If not, explain why not. If so, what does this indicate, and what happens next?

The rest of this problem is about a hilariously tiny memory system with 16-bit virtual addresses, 16-bit physical addresses, and 4 KB (2^{12} B) pages.

Here are 4 current virtual-to-physical address translations for some process. Assume that any virtual pages not shown here are currently invalid. The 16-bit addresses are separated into groups of 4 bits for legibility:

- Virtual address 0101 0111 1110 0110 becomes physical address 0001 0111 1110 0110.
- Virtual address 1100 0111 0101 0000 becomes physical address 1010 0111 0101 0000.
- Virtual address 0101 1000 0001 1010 becomes physical address 0001 1000 0001 1010.
- Virtual address 0110 0000 1111 1110 becomes physical address 1000 0000 1111 1110.

Draw a single-level, flat page table for this process.

Draw a two-level page table for this process. Assume that the primary and secondary page numbers are the same number of bits.

Problem 5: Filesystems (14 points)

In a system with 4 KB disk blocks (or clusters), a file takes up the following 14 blocks in order:
16, 14, 13, 12, 11, 10, 9, 8, 1, 2, 3, 4, 5, 7

If the filesystem is FAT32, show the relevant portion of the file allocation table:

Is the FAT32 file allocation table per-file or systemwide?

If the filesystem is an inode-based system with 12 direct pointers, 1 indirect pointer, 1 double indirect pointer, and 1 triple indirect pointer, show the inode structure:

Is the inode per-file or systemwide?

Have a great summer!