CS 351 Exam 3, Fall 2010

Your name: _____

Rules

- You may use one handwritten 8.5 x 11" cheat sheet (front and back). This sheet and the attached diagram are the only resources you may consult during this exam.
- Include explanations and comments in your answers in order to maximize your partial credit. However, you will be penalized for giving extraneous incorrect information.
- You may use the backs of these pages if you need more space, but make it clear where to find your answer to each question.
- Unless otherwise specified, you do not need to work out the arithmetic on math problems. Just do enough algebra to set up an answer of the form: Answer = [arithmetic expression] [units]

Grade (instructor use only)

	Your Score	**Max Score**
Problem 1		25
Problem 2		15
Problem 3		15
Problem 4		15
Problem 5		15
Problem 6		15
Total		100

Problem 1: 25 points (5 each).

Answer the following questions. Be sure to explain your answers in order to receive partial credit.

For the multiple-choice questions, **a statement is *true* if it is true for all existent computer systems, even if it could theoretically be false.**

---

a) Which of the following statements is true?

A. A system's L1 cache is smaller than its main memory.
B. Main memory is smaller than the L1 cache.
C. Main memory and the L1 cache are equal in size.
D. Main memory could be smaller than, larger than, or equal in size to the L1 cache.

b) Which of the following statements is true?

A. A system's virtual addresses have more bits than its physical addresses.
B. Physical addresses have more bits than virtual addresses.
C. Physical and virtual addresses must have the same number of bits.
D. Physical addresses could be smaller than, larger than, or equal in size to virtual addresses.

c) Explain what cache block size and write policy would be best for the following snippet of code:

```
int indices[100];
int A = new int[100000];

// Set indices to random numbers
for (int i=0; i < 100; i++)
        indices[i] = rand() % 100000;

while (true)
        for (int i=0; I < 100; i++)
                A[indices[i]]++;
```

d) How could you design a cache to completely avoid conflict misses?

e) List the steps that occur when a process requests a legitimate virtual address that is not currently mapped to physical memory. What is this situation called?

Problem 2: 15 points.

Answer these questions about virtual machines.

---

a) [9 points] Our startup company is trying to virtualize the Schmintel x68 architecture, whose ISA includes the following instructions:

ADD reg1, reg2, reg3
Adds general-purpose registers reg2 and reg3 and stores the answer in reg1

SETPT const
Sets the page table register to *const* if running in privileged mode. If running in user mode, causes an illegal instruction exception.

GETMODE reg1
Reads the hardware privilege bits to find out the current privilege level, and stores the answer in reg1. Any code is allowed to read these bits.

How should our virtual machine monitor handle each of these instructions when run by a guest OS?

b) [3 points] A virtual machine monitor is running two guest operating systems that are attempting to map virtual pages to the same page in physical memory. How does the VMM handle this conflict?

c) [3 points] List one advantage and one disadvantage (other than the cost of buying the VM software) of using virtual machines.

Problem 3: 15 points.

Assume the following memory hierarchy:

- Level 1 instruction cache: 99% hit rate, 1-cycle access time.
- Level 1 data cache: 92% hit rate, 1-cycle access time.
- Level 2 cache: 86% hit rate, 5-cycle access time.
- Main memory: 100% hit rate, 100-cycle access time.

---

a) [12 points] If 25% of instructions are loads or stores, how many cycles does the average instruction spend accessing memory? (Memory = any level of the memory hierarchy, not just main memory)

b) [3 points] If L2 is bigger than L1, how is it possible for it to have a lower hit rate? How realistic do you think this is?

Problem 4: 15 points.

A byte-addressable machine with 32-bit memory addresses has a cache with the following properties:

- 16-byte cache blocks
- 16KB of data in the cache
- 4-way set-associative
- Write-back

---

a) [6 points] On average, how many bits of metadata are required per block? Explain what each bit is for.

b) [2 points + 1 bonus if the whole class gets this right] How many bits are in a byte?

c) [7 points] How big is the entire cache (in bytes)?

Problem 5: 15 points.

For each of the following caches, show how memory location 100111 maps to the cache. Include any necessary metadata. The main memory has 6-bit addresses.

---

a) [5 points] The cache has:

- 4-byte blocks
- Direct mapping
- 8 entries

Entry #	Data from address(es)	Metadata
000		
001		
010		
011		
100		
101		
110		
111		

b) [5 points] The cache has:

- 1-byte blocks
- Direct mapping
- 8 entries

Entry #	**Data from address(es)**	**Metadata**
000		
001		
010		
011		
100		
101		
110		
111		

c) [5 points] The cache has:

- 1-byte blocks
- 2-way set associativity
- 8 entries

Entry #	**Data from address(es)**	**Metadata**
00 (a)		
00 (b)		
01 (a)		
01 (b)		
10 (a)		
10 (b)		
11 (a)		
11 (b)		

Problem 6: 15 points.

A byte-addressable memory system has:

- 2 protection bits
- A 64-entry TLB
- 48-bit virtual addresses
- 48-bit physical addresses
- 8 KB pages

The system can run up to 1024 processes at once.

---

a. [3 points] How many virtual pages does the system have?

b. [3 points] How many physical pages does the system have?

c. [4 points] How many bits are needed for each page table entry (data and metadata)?

d. [5 points] How many bits (data and metadata) are needed for the TLB?