

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.*
- You may write your answers in the form *[mathematical expression][units]*. There is no need to actually do the arithmetic.
- 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.
- You may use extra scratch paper if you need more space, but make it clear where to find your answer to each question.

Grade

	Your Score	Max Score
<i>Problem 1: Caches</i>		26
<i>Problem 2: Virtual memory</i>		22
<i>Problem 3: Disk performance</i>		12
<i>Problem 4: RAID</i>		11
<i>Problem 5: Flash</i>		5
<i>Problem 6: Reliability</i>		5
<i>Problem 7: Parallelism</i>		13
<i>Problem 8: Cache coherence</i>		11
Total		105

Problem 1: Caches (26 points)

Consider a 2-way set-associative, write-back cache with 2-byte blocks. The cache stores 16B of data (not counting metadata). Memory addresses are 8 bits.

Part A (16 points)

Draw this cache, including all metadata, after the following sequence of references:

- Write address 01001101
- Write address 11001100
- Write address 00000111
- Write address 00010110
- Read address 01011100

Part B (5 points)

How many bytes will be written to memory over the course of these accesses?

Part C (5 points)

Provide a sequence of references (as long as necessary) that will have a higher hit rate with this cache than with a 16B fully-associative cache with 2B blocks.

Problem 2: Virtual memory (22 points)

Given a hilariously tiny memory system with 8B pages, 6-bit virtual addresses, and 8-bit physical addresses, answer the following questions:

Part A (3 points)

How many virtual pages will each process have?

Part B (3 points)

How many physical pages will the system have?

Part C (7 points)

Draw a sample page table for some process. Your only constraint is that exactly 4 of its virtual pages are currently mapped to physical memory.

Part D (8 points)

Fill out the following table to translate each of the virtual addresses below to a physical address for the process whose page table you created in Part C. If a particular access is a page fault, you may map it to any unused physical page.

Virtual Addr.	Virt. Page #	Page fault?	Phys. Page	Phys. Addr.
101001				
100011				

Problem 3: Disk performance (12 points)

You have an array of 4 disks (counting any redundant data). Each disk has an average seek time of 3 ms, a rotation speed of 7200 rpm, and a transfer rate of 80 MB/s.

Choose 3 of the 4 configurations below. For each of your choices, calculate how long it takes to do a 4 MB read. Where applicable, you may assume that 4 MB is an integer multiple of the stripe size.

- JBOD
- RAID 0
- RAID 1
- RAID 5

Problem 4: RAID (11 points)

Part A (4 points)

If you have 5 disks' worth of original, non-redundant data, how many TOTAL will you need for:

- A RAID 1 array?

- A RAID 5 array?

Part B (4 points)

If you have 1,000 disks' worth of original, non-redundant data, how many TOTAL will you need for:

- A RAID 1 array?

- A RAID 5 array?

Part C (3 points)

Of the 4 different configurations discussed in Parts A and B (i.e. RAID 1 and RAID 5, each at two different sizes), which is the least reasonable to implement in practice? Explain.

Problem 5: Flash (5 points)

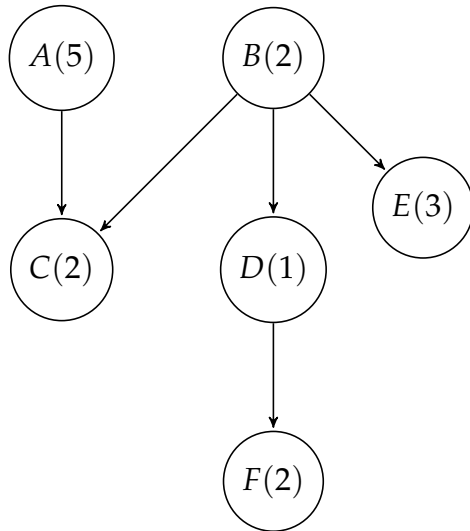
You open up a text file, change one uppercase letter to lowercase, and close the file. If that file is being stored on a flash drive, why will this *not* result in a 1B write operation, and what is likely to happen instead?

Problem 6: Reliability (5 points)

A vendor brags that they've increased the MTBF of their cloud storage platform. Should you be impressed? Explain.

Problem 7: Parallelism (13 points)

Consider the task graph below, where an arrow from X to Y means that Y cannot start executing until X completes. The amount of time (in some unspecified units) that each task takes is in parentheses.



Part A (4 points)

How many threads are necessary to obtain the maximum possible performance from this code? Show the allocation of tasks (graph nodes) to threads.

Part B (5 points)

What is the maximum possible *speedup* of a parallel version, compared to executing all of the tasks in some sequential order?

Part C (4 points)

What is the *efficiency* for the number of threads you identified in Part A?

Problem 8: Cache coherence (11 points)

Consider a memory hierarchy in which two processors have private, write-through L1 caches and a shared L2 cache.

For your reference, we define a coherent memory system as one that meets these conditions:

- If a given processor writes a value Y to memory location X and then reads from memory location X , with no writes to X from other processors occurring in the interim, the read should return the value Y .
- If a processor $P1$ writes a value Y to memory location X , and no other processors write to X in the interim, all other processors' reads from X that happen a sufficient amount of time after the write should return the value Y .
- Consecutive writes to a location X should be seen by all processors in the same order.

Part A (6 points)

If there is no hardware mechanism to guarantee cache coherence, give a sequence of operations that will cause the coherence properties to be violated, starting with:

1. Processor 1 writes the value 5 to location X .

Part B (5 points)

You want to modify the L1 caches to be coherent, but you are not willing to add metadata (so you can only work with the valid bit). Explain what action (if any) Processors 1 and 2 could take in the following situations to ensure coherence. Try to minimize unnecessary coherence and memory traffic, but a solution that correctly supports coherence is better than no solution.

- Processor 1 has an L1 read hit for location X.
- Processor 1 has an L1 read miss for location X.
- Processor 1 has an L1 write hit for location X.
- Processor 1 has an L1 write miss for location X.