CS 162 Operating Systems and System Programming
Spring 2025

Midterm 3

INSTRUCTIONS

Please do not open this exam until instructed to do so. Do not discuss exam questions for at least 24 hours after the exam ends, as some students may be taking the exam at a different time.

GENERAL INFORMATION

This is a **closed book** exam. You are allowed 3 pages of notes (both sides). You have 80 minutes to complete as much of the exam as possible.

Write all of your answers directly on this paper. *Make your answers are as concise as possible.* If there is something about the questions that you believe is open to interpretation, please ask us about it!

Name

Student ID

Please read the following honor code: "I understand this is a closed-book exam. I promise the answers I give on this exam are my own. I understand that I am allowed to use three 8.5x11, double-sided, handwritten cheat-sheets of my own making, but otherwise promise not to consult other people, physical resources (e.g. textbooks), or the internet in constructing my answers."

Write your full name below to acknowledge that you've read and agreed to this statement.

Problem	**Possible**
1	12
2	15
3	19
4	16
5	18
6	1 (EC)
Total	80

PCIe 7.0 specification is coming out in 2025, offering up to 128 GT/s bitrate!

1. **(12.0 points) True/False**

   Please explain your answer in **TWO SENTENCES OR FEWER**. Answers without an explanation or that just rephrase the question will **not receive credit**.

   (a) **(2.0 pt)** Once the CPU programs the DMA controller, it typically polls its status registers to see if the DMA request has been completed.

   ○ True. Explain how this allows high-speed data transfer to a periphery device.

   ● False. Explain how the CPU knows if the DMA request has been completed.

   DMA engine interrupts the CPU. You can typically poll them, but it defeats the purpose.

   (b) **(2.0 pt)** In FFS, directories and regular files have the same maximum size in bytes.

   ● True. Explain why both directories and files have the same max file size.

   Both are structured as inodes and thus are limited by the structure of the inode.

   ○ False. Explain why directories and files have different max file sizes.

   (c) **(2.0 pt)** In NTFS, reading very small files requires a minimum of two disk accesses: one access for the Master File Table, and one for the file data.

   Note: Assume nothing starts off cached.

   ○ True. Explain how accessing small files works in NTFS.

   ● False. Explain how accessing small files works in NTFS.

   Small fies can store their data entirely in the Master File Table.

(d) **(2.0 pt)** In a journaling file system, you find a transaction with the following entries: `[Begin transaction, Allocate inode, Allocate direct pointer]`.

Given the logged operations, it is possible that these operations have been applied to disk.

○ True. Explain why these operations **may** have been reflected on disk.

● False. Explain why these operations would **not** have been reflected on disk.

Transactions are atomic, which means that if it isn't committed, it is aborted. When COMMIT is written, the data is written back async.

(e) **(2.0 pt)** Updates to files stored in NFS are immediately broadcasted to all other clients.

○ True. Explain how NFS broadcasts updates to clients.

● False. Explain how clients observe file updates in NFS.

NFS clients periodically poll the server.

(f) **(2.0 pt)** In Spark, data from RDDs is not permanently lost upon machine failure.

● True. Explain what mechanism allows Spark to recover the in-memory partition.

You can rebuild RDDs from lineage (the last surviving in-memory RDD or the on-disk beginning dataset)

○ False. Explain how hardware can help mitigate this issue.

2. (15.0 points) Multiple Select

(a) (2.5 pt) Select all true statements about HDDs vs SSDs.

□ SSDs allow remapping of bad (e.g. worn-out) sectors, but HDDs do not.

■ HDDs perform best when doing sequential reads rather than random reads.

■ Writing to a page is typically slower than reading from a page in SSDs.

■ Modern HDD controllers have internal RAM that can buffer prefetched reads.

■ Reading from outer tracks of a modern HDDs offer more bandwidth than from the inner tracks.

(b) (2.5 pt) Select all true statements.

■ Most of the code in Linux comes from device drivers.

■ Device controllers contain a special set of registers that you can interact with through MMIO.

■ Device drivers provide support for the POSIX file API (open/read/write).

■ The PCI bus needs arbitration logic to ensure only 1 device is driving data through a channel at a time.

□ MMIO requires ISA support for special instructions, such as x86 `in` or `out`.

(c) (2.5 pt) The BSD 4.2 optimizations for FFS:

□ Optimized the Fast File System based on the architecture of SSDs.

□ Divided inodes into different block groups than data blocks for the same file.

□ Replaced the block bitmap with a free list to optimize for sequential block allocation.

■ Reserved 10% free space on disk to maximize sequential runs for extending files.

■ Improved locality for metadata and data, as well as small and large files.

(d) (2.5 pt) Select all true statements regarding the FAT file system:

□ Each entry in the FAT contains a file name and file number.

□ Reading from a random offset in a file is typically faster in FAT than FFS.

■ The file number corresponds to the index of the first block for that file.

■ FAT is supported by all Linux, Windows, and MacOS.

■ A file's metadata (i.e. file owner) is stored in its parent directory's entries.

(e) (2.5 pt) Select all true statements about MapReduce and Spark as presented in lecture.

□ MapReduce assumes that the coordinator machine will frequently experience failures.

□ If a worker fails while doing a reduce task, the coordinator will reassign that task and all map tasks.

■ The output of a completed map task is stored in a worker's local storage.

■ Spark is more performant than MapReduce for problems that reuse working sets of data.

□ Applying a transformation to an RDD immediately starts materializing the result.

(f) (2.5 pt) Select all true statements about VFS.

□ VFS is a file system for virtual machines that can be mounted on a host machine.

■ VFS allows you to mount external storage media formatted in a different file system.

■ VFS allows you to transparently access a remote file system through the network.

□ VFS exposes a unique syscall interface for every mounted filesystem (e.g. POSIX read vs NFS readAt).

■ VFS abstracts away details about the underlying mounted filesystem from the rest of the kernel.

3. (19.0 points) Short Answer

(a) (3.0 pt) What is the copy-on-write mechanism for the FTL (Flash Translation Layer), and what is its primary benefit? (Two sentences max for each.)

Explain the mechanism here.

Writes out new version of a block to another block and remaps it through the Flash Translation Layer.

Explain its benefit here.

Avoids wear out of a single block that gets repeatedly written.

(b) (3.0 pt) You used to have to click "Eject" on an external storage media (e.g. a flash drive) before removing it, even when you are not actively accessing it at the point of removal, otherwise you could leave the entire file system in it in an inconsistent state. What is a potential explanation for this?

OS will buffer disk writes in memory (buffer cache). If the flash drive is suddenly ejected, some of the changes to the flash drive's file system may not have been flushed, leading to an inconsistent state.

(c) (3.0 pt) Consider the End-to-End principle. Suppose we can fully implement a checksum for the whole packet in the IP layer to make sure it was not corrupted in transit. What are two reasons that may dissuade you from doing so?

Reason 1

It imposes a performance overhead on applications that do not require a checksum (e.g. video streaming).

Reason 2

Replicates same functionality as layers above (transport, e.g. TCP, and applications e.g. Torrent

(d) **(8.0 pt)** RPC and System Design

i. **(2.0 pt)** Consider a heterogeneous system where machines interpret a network byte stream differently (e.g. different endianness).

What mechanism of RPC enables data to be sent across the network and processed by two heterogeneous machines?

Name the mechanism: **Marshaling/Serialization**

Explain this mechanism in 1 sentence:

It converts the function arguments into a canonical representation

ii. **(2.0 pt)** Identify and explain what piece of autogenerated code allows a client to both initiate an RPC request and interpret the RPC result sent from the server.

1. Client invokes client stub which marshals the arguments to send across the network, and unmarshals the return values.

iii. **(2.0 pt)** Suppose the connection between the client and the server can drop messages or delay them for an arbitrary amount of time. What should an NFS client do if it doesn't hear an acknowledgement of its request from the server?

Client should resend this message after some fixed timeout duration.

iv. **(2.0 pt)** Suppose an NFS server receives duplicate `write` operations from a client. What feature of NFS allows the server to handle these duplicate write operations?

Hint: Recall NFS is a **stateless** protocol.

The RPC for write should be idempotent.

(e) **(2.0 pt)** Explain one issue if intermediate results from map tasks were not saved locally, and were transferred directly between map and reduce workers.

Acceptable answers include: (1) A reduce worker could crash while the data is being transmitted to them, and the map task needs to be redone; (2). Each map task needs to send results to all reduce tasks, incurring overhead; (3) Map tasks now need to know addresses of all reduce tasks for sending data; (4) Map tasks need to wait for all reduce workers to be live before streaming. Points were docked for responses describing worker crashes in general, rather than reduce crashes specifically.

4. (16.0 points) File System Efficiency

After purchasing a new HDD, we want to read 5 sectors that are in cylinders 1, 3, 6, 7, and 15 respectively. **Suppose the disk arm is currently placed over cylinder 4.** (Don't include Cylinder 4 in your answer.)

(a) **(2.5 pt)** If we implement SCAN, and the arm is currently moving towards **lower** cylinder numbers, What is the order of cylinders that the disk arm will travel to?

(Write down the cylinder number for each box. Write cylinders serviced earliest in the leftmost box.)

Order of service (Earliest ⟶ Latest)				
3	**1**	**6**	**7**	**15**

(b) **(2.5 pt)** If we implement C-SCAN, and the arm is currently moving towards **higher** cylinder numbers, what is the order of cylinders that the disk arm will travel to?

(Write down the cylinder number for each box. Write cylinders serviced earliest in the leftmost box.)

Order of service (Earliest ⟶ Latest)				
6	**7**	**15**	**1**	**3**

(c) **(5.0 pt)** Suppose we have a hard drive with the following specifications:

- Average seek time: 8 ms
- Rotation speed: 3,000 RPM
- Transfer speed: 5 MB / s
- Sector size: 2 KB

i. **(2.0 pt)** What is the average transfer latency for reading 5 sequential sectors **(in ms)**?

Notes:

- Disregard all other delays (controller, software/queuing, etc.)
- This question is in terms of **MB/KB**, not MiB/KiB.
- **Write down a single numerical value. Do not leave it as a sum of unsimplified terms. For example, 59ms is acceptable, 10 + 40 + 9 ms is not.**

Average Transfer Latency: 20ms

Show your work below. (Credit will only be awarded if work is shown).

3000 rot / min * 1 min / 60 s = 50 rot/s or 0.02 s/rot → 0.01 s/half rot. 8 ms = 0.008 s for seek time. Sector size = 2000 B, Transfer speed = 5 * 10^7 B/s so 1s/(5 * 10^6 B) * 10^3 ms / s * 10 * 10^3 B = 2 ms to transfer data. 2 ms (transfer) + 8 ms (seek) + 10 ms (rot) = 20 ms

ii. **(3.0 pt)** As a follow-up question to part **i**, how many sectors do you need to read in succession to achieve an effective transfer rate of 2 MB/s?

For your convenience, the hard drive specifications from part (i) have been copied below:

- Average seek time: 8 ms
- Rotation speed: 3,000 RPM
- Transfer speed: 5 MB / s
- Sector size: 2 KB

Notes:

- **Effective transfer rate:** (size of data) / (total service time)
- Disregard all other delays (controller, software/queuing, etc.)
- **Write down a single numerical value.**

Number of sectors: 10

Show your work below. (Credit for both boxes will only be awarded if work is shown).

Xfer time = 5 * 10^6 B / s * 1 s / 10^3 ms = 5 * 10^3 B / ms = 5000 B / ms 1/5000 ms/B * 2000 B/sector * x sectors = 0.4x ms for x sectors. 0.4*x + 10 ms (1/2 rot) + 8 ms = (18 + 0.4x ms) / (2000x B) or 2000x B / (0.4x ms + 18) = 2 * 10^6 B / s * 1 s / 10^3 ms = 2000 B/ms. 2000x / (0.4x + 18) = 2000 x / (0.4x + 18) = 1 x = 0.4x + 18 so x = 18 / 0.6 = 30

In the following questions, consider the FS block size to be equivalent to the sector size of the underlying storage medium.

(d) **(6.0 pt)** Consider a variant of FFS with 4 KiB blocks and sectors. An inode contains 8 direct pointers and 1 indirect pointer.

Within the root directory, directory entry for `cs162` is in the 5th data block.

Within the root directory, directory entry for `shamith.txt` is in the 1st data block.

The file `shamith.txt` is 36 KiB in size.

Assume that no data from disk is cached in memory, and the inumber of the root is known.

Determine how many disk accesses it takes to fully read in `"/cs162/shamith.txt"`. You may not need all boxes provided.

1. Read **1** sector(s) to **read in the root inode**
2. Read **5** sector(s) to **find cs162 in root dir**
3. Read **1** sector(s) to **read cs162 inode**
4. Read **1** sector(s) to **find dir entry for shamith.txt in cs162**
5. Read **1** sector(s) to **read in inode for shamith.txt**
6. Read **8** sector(s) to **direct pointers in shamith.txt**
7. Read **1** sector(s) to **indirect pointer/block in shamith.txt**
8. Read **1** sector(s) to **read in one sector from indirect block**

Total of **19** sectors read.

5. (18.0 points) Distributed Systems

(a) **(4.0 pt) MapReduce**

A social media platform called PintOSrest wants to analyze who the most person with most friends is. User A and User B are each other's friends if and only if **both** users follow each other.

The company has a table with key-value records, where the key is the user name, and the value is the name of one of the user's followers. Implement the following MapReduce program to find out how many friends each person has.

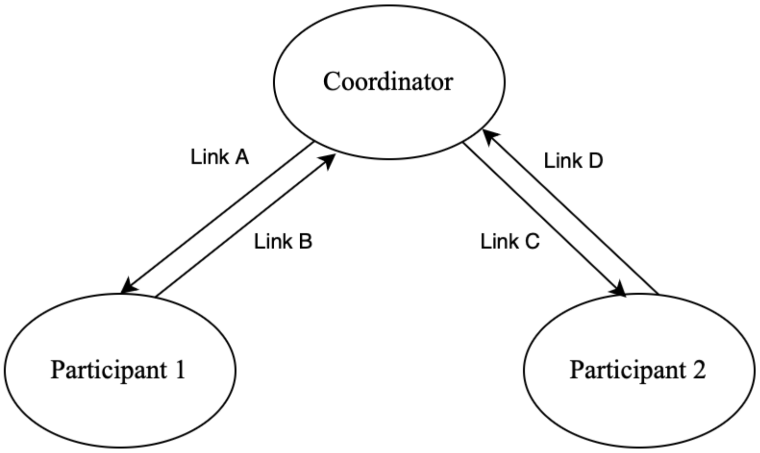
Assume `Emit` can take in values of any type. Do not add any loops or conditionals, including ternary operators.

```
map(char* key, char* value) {
    Emit(key, value);
    Emit(value, key);
}

reduce(char* key, char* values[], size_t n) {
    sort_in_place(values); // <--- Sorts the values in-place.
    int num_friends = 0;
    for (int i = 0; i < n-1; i++) {
        if (strcmp(values[i], values[i+1]) == 0) {
            num_friends++;
        }
    }
    Emit(key, num_friends);
}
```

(b) **(8.0 pt) 2PC**

We have the following 2PC cluster with 2 Participants, with asymmetric communication links:



Answer the following questions, considering each scenario independently of one another.

i. **(2.0 pt)** The Coordinator sends a vote request which **arrives successfully** at Participants 1 and 2. Which of the following failures, considered independently, can lead to the transaction being aborted?

□ Link A

■ Link B

□ Link C

■ Link D

□ None of the above

ii. **(2.0 pt)** Participant 1 recovers from a crash and sees a `GLOBAL-COMMIT` in its log. Which of the following failures, considered independently, can lead to the transaction being aborted?

□ Link A

□ Link B

□ Link C

□ Coordinator

■ None of the above

iii. **(2.0 pt)** Participants 1 and 2 successfully send a `VOTE-COMMIT` to the Coordinator. Which of the following failures, considered independently, can lead to the transaction being aborted?

□ Link A

□ Link C

■ Coordinator

□ Participant 1

□ None of the above

iv. **(2.0 pt)** The Coordinator recovers from a crash and finds nothing in its logs. Under 2PC's assumptions, what are possible scenarios on what the last thing the Coordinator did before the crash?

■ The Coordinator failed before even sending out a `PREPARE` message.

□ The Coordinator failed immediately after receiving `VOTE-COMMIT` from both Participants.

□ The Coordinator sent a `GLOBAL-COMMIT` and failed immediately after.

□ The Coordinator logged a `GLOBAL-COMMIT` and failed immediately after.

□ None of the above

(c) **(6.0 pt) Paxos**

Assume that in this question, all participants share the same time clock, and actions only happen at discrete time units (e.g. $t = 0, 1, 2, 3...$).

Circle the correct response when necessary, and fill in the blank with appropriate answers.

Example:

What must be true about a participant who made a proposal with ballot id 1?

The participant must have [Received / [✓ Sent]] a proposal with ballot id 1.

i. **(2.0 pt)** At $t = 3$, a participant receives a proposal with ballot ID 1, but does not accept this value. Suppose this participant has not initiated any "elections". What had to have happened during t $<$ 3 for the participant to not accept this value?

The participant must have [Received / **Sent**] a **PREPARE** with ballot ID $\geq$ **2**.

ii. **(2.0 pt) This part is independent of part (i).**

After Participant A proposes ballot ID 2 and value 5, suppose Participants A, B, and C accept this value. Suppose the PROPOSE message to Participant D and E gets infinitely delayed.

How do Participants D and E find out about this new value?

Participant A must have [Received / **Sent**] a **COMMIT** with ballot ID 2 and value 5.

iii. **(2.0 pt) Continuing from part (ii)**, suppose Participants D and E do not learn about this message due to network failure. Participants A and B fail.

If Participant D prepares a ballot with ID 3, it will propose value 5 again because ...

Participant C will [Receive / **Send**] a **PROMISE** containing value 5 from ballot ID 2.

6. (Extra Credit: 1.0 pt) What year did Professor Crooks and Professor Zaharia get their PhDs?

Professor Crooks: 2019

Professor Zaharia: 2013

7. **(0.0 pt)** If you found any questions ambiguous, please address your interpretation of the question with its question number here. We will take a look at this during regrades and may award points if we consider it valid.

.

Reference Sheet

```
/* Units */
```

KiB = 2^{10} B

MiB = 2^{20} B

GiB = 2^{30} B

KB = 1000 B

MB = 10^6 B

GB = 10^9 B

```
/* Strings */
char *strcpy(char *dest, char *src);
char *strdup(char *src);
int strcmp(char *s1, char *s2);
```