Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.1800 Computer Systems Engineering: Spring 2024

Exam 2

There are **13 questions** and **10 pages** in this exam booklet. Answer each question according to the instructions given. You have two hours to answer the questions.

- The questions are organized loosely by topic. They are not ordered by difficulty nor by the number of points they are worth.
- **If you find a question ambiguous, write down any assumptions you make.** Be neat and legible.
- You are not required to explain your answers unless we have explicitly asked for an explanation. You may include an explanation with any answer for possible partial credit.
- Some students will be taking a make-up exam at a later date. **Do not** discuss this exam with anyone who has not already taken it.
- Write your name and kerberos ID in the space below. Write your initials at the bottom of each page.

This is an open-book, open-notes, open-laptop exam, but you may **NOT** use your laptop, or any other device, for communication with any other entity (person or machine).

Turn all network devices, including your phone, off.

Name:

Kerberos ID:

1. [12 points]: Jordan and Phoebe are learning about RAID. They set up a very simple RAID-4 system with two data disks and one parity disk. The first block on the first data disk is `01010101`. The first block on the parity disk is `00100100`.

A. What is the first block on the second data disk?

With that practice out of the way, Jordan and Phoebe are ready to store more data. They each have N terabytes of data to store, and can purchase disks that each store **two** terabytes. You can assume that N is even. Jordan continues using RAID-4, while Phoebe decides to use RAID-1.

B. How many disks will each system need in order to store N TB of data?

Jordan's system: _____ **Phoebe's system:** _____

C. Suppose that one disk in each system fails. How many disks does the system need to read in order to reconstruct the failed disk?

Jordan's system: _____ **Phoebe's system:** _____

D. Phoebe claims her system can recover from two disks failing at the exact same time (not approximately the same time; the *exact* same time). Is she correct? Circle the **best** answer.

(a) Phoebe is entirely correct: her system can recover from any pair of disks failing at the exact same time.

(b) Phoebe is partially correct: her system can recover from *some* pairs of disks failing at the exact same time, but not every pair of disks.

(c) Phoebe is incorrect: her system cannot recover from any two-disk failures.

E. Jordan is considering upgrading his system from RAID-4 to RAID-5. What is the **best** reason for him to do this?

(a) RAID-5 is more reliable than RAID-4 in the sense that it can recover from more disks failing at once than RAID-4 can.

(b) RAID-5 has better performance than RAID-4.

(c) RAID-5 requires fewer disks than RAID-4 to store the same amount of data (and thus it costs less to use RAID-5 than RAID-4).

2. [10 points]: A transaction-based system is using write-ahead-logging. The log is backed by both on-disk cell storage and an in-memory cache. The cache is shared by all transactions. Writes go first to the log, then to the cache. Reads go first to the cache, and then to cell storage if the data is not present in the cache. Assume that all values are initialized to zero in cell storage, and that the cache is large enough to hold all data in question.

Consider the following log snippet.

TID	T1	T1	T1	T2	T3	T3
	UPDATE	UPDATE	COMMIT	UPDATE	UPDATE	COMMIT
OLD	W=0	X=0		Y=0	Z=0	
NEW	W=10	X=20		Y=30	Z=40	

Occasionally, all values in the cache are flushed to cell storage. For the purposes of this question, assume that that happens exactly once: immediately after `T2`'s update (but before `T3`'s update).

Suppose that the system crashes after the last entry in the above log was written (i.e., after `T3` commits).

A. Immediately before the crash, what are the values in the cache?

W: _____ X: _____ Y: _____ Z: _____

B. Immediately before the crash, what are the values in cell storage?

W: _____ X: _____ Y: _____ Z: _____

After the crash, the system goes through the recovery process using the recovery code below (this code is different from the code you saw in lecture).

```
commits = []
for record r in log[len(log)-1] .. log[0]:
   if r.type == COMMIT:
      commits.add(r.tid)
for record r in log[0] .. log[len(log)-1]:
   if r.type == UPDATE and r.tid in commits:
      cell_write(r.var, r.new_value)
```

C. After this recovery code runs, what are the values in cell storage?

W: _____ X: _____ Y: _____ Z: _____

D. True / False After the recovery code runs, this system is correct, in the sense that all committed updates will be read by future transactions, and no uncommitted updates will be.

Initials:

3. [11 points]: Consider the following four transactions (`T1`, `T2`, `T3`, and `T4`). The syntax for these transactions states whether they are reading or writing a particular variable (`x`, `y`, or `z`); the values of the reads and writes do not matter for this question. Each line of each transaction has a unique line number (e.g., `T1.2` denotes the second line of `T1`).

```
     T1             T2             T3             T4
T1.1 read(x)   T2.1 write(x)  T3.1 read(y)   T4.1 write(x)
T1.2 write(y)  T2.2 read(z)   T3.2 write(z)  T4.2 read(y)
                                             T4.3 read(z)
```

Below we show four schedules: an "original" schedule, and three alternative schedules.

Original	Alternative 1	Alternative 2	Alternative 3
`T1.1 read(x)`	`T1.1 read(x)`	`T1.1 read(x)`	`T1.1 read(x)`
`T4.1 write(x)`	`T1.2 write(y)`	`T4.1 write(x)`	`T4.1 write(x)`
`T2.1 write(x)`	`T4.1 write(x)`	`T2.1 write(x)`	`T2.1 write(x)`
`T1.2 write(y)`	`T2.1 write(x)`	`T1.2 write(y)`	`T3.1 read(y)`
`T3.1 read(y)`	`T3.1 read(y)`	`T4.2 read(y)`	`T1.2 write(y)`
`T4.2 read(y)`	`T2.2 read(z)`	`T3.1 read(y)`	`T4.2 read(y)`
`T2.2 read(z)`	`T3.2 write(z)`	`T2.2 read(z)`	`T2.2 read(z)`
`T3.2 write(z)`	`T4.2 read(y)`	`T3.2 write(z)`	`T3.2 write(z)`
`T4.3 read(z)`	`T4.3 read(z)`	`T4.3 read(z)`	`T4.3 read(z)`

A. Draw the conflict graph for the original schedule in the space below.

B. Which of the alternative schedules have conflict graphs that are equivalent to the original schedule's conflict graph? Circle all that apply.

(a) Alternative Schedule 1
(b) Alternative Schedule 2
(c) Alternative Schedule 3
(d) None of the alternative schedules have conflict graphs equivalent to the original schedule's

C. True / False The original schedule could have been produced by strict two-phase locking.

Initials:

4. [8 points]: Aileen is investigating a two-phase commit set-up that involves multiple clients, a single coordinator, and two servers, A and B. A and B have been involved in multiple transactions so far. Aileen reads the log stored on A and notices that there is a `PREPARE` message for Transaction 6, but not a `COMMIT` message for Transaction 6.

A. What are the possible reasons that A's log would contain a `PREPARE` message for this transaction but no `COMMIT` message? Select all that apply.

(a) The coordinator sent out `COMMIT` messages for Transaction 6, but A hasn't received one yet.

(b) The coordinator sent out `COMMIT` messages for Transaction 6, but A crashed before receiving one and is still in the process of recovering.

(c) The coordinator sent out `COMMIT` messages for Transaction 6, but crashed before sending one to A and is still in the process of recovering.

(d) The coordinator sent out `ABORT` messages for Transaction 6, but A hasn't received one yet.

(e) None of the above.

B. Aileen also notices that A's log contains an `UPDATE` record for Transaction 6, which updates the value of a variable `x` to 20. Assume that the system is meant to achieve single-copy consistency. Is it safe for A to let other parts of the system read the value `x=20`? Write "yes" or "no".

Aileen also notices that A's log contains a `PREPARE` message for Transaction 5—that's 5, not 6—*and* a `COMMIT` message for Transaction 5.

C. Aileen believes that, because A's log contains a `PREPARE` message and a `COMMIT` message for Transaction 5, B's log must also contain at least a `PREPARE` message for Transaction 5 at this stage. Is she correct? Write "yes" or "no".

5. [10 points]: Answer True or False for each of the following questions about MapReduce and GFS

(a) **True / False** If there are m map tasks, using more than m workers in the map phase may still improve performance beyond that achieved with m workers.

(b) **True / False** To achieve locality, map workers always execute on the same machine as the input data that they consume.

(c) **True / False** GFS optimizes for writes that append data to a file, rather than writes to random offsets.

(d) **True / False** From the point of view of clients, GFS provides single-copy consistency.

Initials:

6. [3 points]: Two-phase locking (2PL) produces conflict-serializable schedules. However, as indicated in the Consistency Rationing paper, many systems forgo conflict-serializability (and thus don't need to use two-phase locking). What is (are) the potential benefit(s) of foregoing 2PL? Circle all that apply.

(a) Without 2PL, systems can typically respond faster to clients.

(b) Without 2PL, systems don't have to waste time acquiring locks (both in that transactions don't have to make acquire/release system calls and that they don't have to block while others have the locks).

(c) Without 2PL, systems have more possible schedules (of concurrent transactions) available to them, thus increasing opportunities to run computations in parallel.

(d) None of the above.

7. [6 points]: Consider five machines using Raft. The current log of each machine is below. Log entries are specified as `<term number>.<update ID>`; for example, log entry `1.2` is the update of ID 2 in Term 1. We do not give the specific contents of the updates.

```
Leader's log: 1.1, 1.2, 1.3, 2.1, 2.2, 2.3, 3.1, 3.2, 3.3

   M1's log: 1.1, 1.2, 1.3
   M2's log: 1.1, 1.2, 1.3, 2.1
   M3's log: 1.1, 1.2, 1.3, 2.1, 2.2, 2.3
   M4's log: 1.1, 1.2, 1.3, 2.1, 2.2, 2.3, 3.1, 3.2
```

A. Which of the following log entries are **guaranteed** to be committed at some point? Though there may have been failures—of both machines and the network—in the past, you can assume that there will be no failures from this point on. Select **all** that apply.

(a) `1.3`

(b) `2.1`

(c) `2.3`

(d) `3.2`

(e) None of the above

B. Kevin notices that the leader's log contains entries that aren't in any other log (namely `3.3`). Because of this, he believes that none of the other four machines can be the leader in the next round. Is he correct? Answer "yes" or "no".

8. [6 points]: Consider the code below. This code is similar to one of the examples you saw in Lecture 22, but not exactly the same.

```
void win() { printf("code flow successfully changed\n"); }

int main(int argc, char **argv) {
   int i;
   char buffer[128];
   int j;
   gets(buffer);
   char x;
   char y;
}
```

Joey runs this code on a machine that has no protection against stack-smashing. On his machine, integers are four bytes long and characters are one byte long. Any pointers—e.g., the base and instruction pointers, function pointers, etc.—are also four bytes long. When a function is called, the following happens (in this order):

1. Any arguments to the function are pushed onto the stack
2. The base pointer (`BP`) is pushed onto the stack
3. The instruction pointers (`IP`) is pushed onto the stack
4. Local variables to the function are pushed onto the stack

The list above describes **everything** that happens on the stack. There aren't, e.g., any other pointers pushed onto the stack between `BP` and `IP`. Joey's goal is to overwrite the saved instruction pointer and force the function `win()` to be called, by inputting a string into this program.

A. How long should Joey's string be, in **bytes**?

B. Where should the address of `win()` appear in the string? Select the **best** answer.

(a) As the first four bytes of the string
(b) As the final four bytes of the string
(c) Anywhere; it doesn't matter, so long as the address is there
(d) The address of `win()` doesn't need to appear in the string to get the code to jump to this function; Joey can use a random string, and it will work.

Initials:

9. [4 points]: Sadhana is experimenting with the toy example in Listing 1 of the Meltdown paper. She uses a system that has 4096-byte pages, just like in the paper. However, in her experiments, she uses the number 1024 where the example uses 4096. Sadhana's code is below. Each element in `probe_array` is one byte.

```
1 raise_exception();
2 // the line below is never reached
3 access(probe_array[data * 1024])
```

Sadhana runs the example on her system, and observes that the access time for Page 63 is around 100 milliseconds; for every other page, the access time is around 400 milliseconds. Assume that pages are numbered starting at 0. Given Sadhana's observations, what is the value of `data`? If there is more than one possible value of `data`, list them all.

10. [9 points]: Katrina (an attacker) hopes to build a data structure to aid in stealing users' passwords. This data structure should allow Katrina to look up any of the k most common passwords, along with a salt, and retrieve the appropriate (salted) hash of the password. She'll store it as a table; each row of this table will contain a password p, a salt s, and the salted hash of the password.

Each of the k passwords is b bytes long. Each salt is a random string of length ℓ bytes (the salt can be any bitstring that is ℓ bytes long, not just ASCII characters). The hash function in use outputs a d-byte string. Katrina is able to compute the hash of an arbitrary-length string in t time.

For each part below, give your answers in terms of k, b, ℓ, d, and t (you may not need to use all of those variables in each part).

A. How much time will it take Katrina to generate this table? You can assume that the time it takes to store the data in the table is negligible; the dominant computation here is calculating the hash.

B. Roughly how much storage (in bytes) will the table consume?

C. In practice, it is infeasible for Katrina to build this table and use it to attack an authentication system S. Why? Circle the **best** answer.

(a) Katrina won't know which hash function S is using.
(b) Katrina won't know what the k most common passwords are.
(c) Even if Katrina gains access to S and can read its password table, she won't know which salt corresponds to which user.
(d) It will take too much time for Katrina to build this table.

Initials:

11. [7 points]: Alice and Bob are communicating via what they hope is a secure channel. They've exchanged a symmetric key, k, which they use to encrypt, decrypt, and MAC.[1] You can assume that only Alice and Bob know k.

To send a message m to Bob, Alice does the following:

1 Computes $c = encrypt(k, m)$

2 Computes $h = MAC(k, c)$

3 Sends $c \mid h \mid seq$ to Bob, where seq is a sequence number. Alice's first message to Bob has sequence number 1, and she increments the sequence number for each message (i.e., her second message has sequence number 2, the message after that has sequence number 3, etc.). You can assume that the sequence number space is infinite (i.e., sequence numbers will not "wrap").

When Bob receives a message from Alice, he recomputes $MAC(k, c)$ and makes sure that the result equals h. He also checks that the sequence number he receives is the next one he expects (if Bob just received a message with sequence number n, he expects a message with sequence number $n + 1$ next; if Bob hasn't received a message from Alice yet, he expects a message with sequence number 1). If it is not, he ignores the message.

Communication between Alice and Bob is always one-way: Alice sends data to Bob, but Bob never sends data to Alice. There are **no** network failures between them, no packets are ever dropped, and the route between Alice and Bob never changes.

Consider an on-path attacker, Eve. Eve happens to know that Bob runs a bank, and each of the messages that Alice is sending to him initiate a bank transfer that takes money out of Alice's account. Eve would like to drain Alice's bank account. She is **not** interested in launching DDoS attacks on Alice or Bob, stealing their passwords, etc.

A. As an on-path attacker, what could Eve do to achieve her goal? Be precise.

B. What could Alice and/or Bob do to prevent Eve from launching this attack? Assume that Alice still wants to be able to communicate with Bob using the network (i.e., "stop sending messages" is not a correct answer).

[1] In reality, we would use different keys for encryption and MAC'ing, but for simplicity we're going to use the same key for both, just like we did in Lecture 23.

Initials:

12. [6 points]: Answer the following questions about DNSSEC.

A. Grace owns the domain `grace.com` and wants to defend it from botnet attacks. She believes that enabling DNSSEC will help. Is she correct? Circle the **best** answer.

(a) Grace is correct. DNSSEC would help defend against a botnet because DNSSEC encrypts DNS requests.

(b) Grace is correct. DNSSEC would help defend against a botnet because domains would be authenticated.

(c) Grace is correct. DNSSEC would help defend against a botnet because DNSSEC responses tend to be smaller than regular DNS responses (and thus a botnet utilizing DNS traffic would have less traffic to work with).

(d) Grace is incorrect. Enabling DNSSEC on her server will not help her defend against botnets.

B. What is the root zone's role in DNSSEC? Select the best answer.

(a) It distributes public keys, similar to a certificate authority.

(b) It encrypts public keys for distribution (but need not be in the zone that distributes the encrypted keys).

(c) It signs public keys with its own secret key.

13. [8 points]: Joey is sending data to a server S using Tor. His data travels through a series of three proxies: P_1, P_2, and P_3, in that order.

However, unlike in lecture, Joey has **not** set up a secure channel with S. Which of the following attacks is Joey vulnerable to? Circle True or False for each of the following.

(a) **True / False** If the attacker observes both the link between Joey and P_1, and the link between P_3 and S, they can mount a correlation attack on Joey's traffic, and infer that Joey is communicating with S.

(b) **True / False** If the attacker gains control of P_2, and can read all data stored on P_2, as well as observe all packets passing through P_2, they can infer that Joey is communicating with S.

(c) **True / False** If an attacker observes **only** the link between P_3 and S, they can read the data that Joey has sent to S.

(d) **True / False** If an attacker observes **only** the link between Joey and P_1, they can read the data that Joey is sending to S.

Initials: