Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.033 Computer Systems Engineering: Spring 2017

Quiz 2

There are **18 questions** and **14 pages** in this quiz booklet. Answer each question according to the instructions given. You have two hours to answer the questions.

- The questions are organized (roughly) 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.
- Some students will be taking a conflict exam at a later date. **Do not** discuss this quiz with anyone who has not already taken it.
- You may use the back of the pages for scratch work, but **do not** write anything on the back that you want graded. We will not look at the backs of the pages.
- Write your name in the space below. Write your initials at the bottom of each page.

This is an open-book, open-notes, open-laptop quiz, 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.

CIRCLE your recitation section:

10:00	1. Michael/Dustin	7. Karen/Emily	8. Howard/Ben	15. Jing/Paul
11:00	9. Michael/Dustin	10. Karen/Emily	11. Howard/Ben	16. Jing/Paul
12:00	4. Martin/Isabel	12. Mark/Anying		
1:00	2. Martin/Isabel	14. Mark/Anying	5. Melva/Cathie	13. Peter/Xavier
2:00	6. Melva/Cathie	3. Peter/Xavier		

Name:

I Availability via Replication

1. [8 points]: Michael is running some experiments on a system that contains four disks for storage. In each experiment, he writes 10,000 random blocks to the system, and then reads each block back. He monitors the number of reads and writes sent to each disk.[1] He gets the following results:

Experiment 1

Disk	Blocks read	Blocks written
1	3330	3330
2	3340	3340
3	3330	3330
4	0	10000

Experiment 2

Disk	Blocks read	Blocks written
1	2420	3320
2	2560	3340
3	2400	3340
4	2520	10000

Experiment 3

Disk	Blocks read	Blocks written
1	2400	7500
2	2520	7530
3	2580	7560
4	2500	7560

Experiment 4

Disk	Blocks read	Blocks written
1	2430	4980
2	2540	5020
3	2490	5030
4	2540	4070

Michael knows that, during each experiment, his system was using a different disk-redundancy strategy, but isn't sure which experiment corresponds to which strategy.

A. In which experiment is it most likely that Michael's system was using RAID-4? Assume that writes to/reads of parity blocks are reflected in his data. Circle the **best** answer.

Experiment 1 Experiment 2 Experiment 3 Experiment 4

B. In which experiment is it most likely that Michael's system was using RAID-5? Assume that writes to/reads of parity blocks are reflected in his data. Circle the **best** answer.

Experiment 1 Experiment 2 Experiment 3 Experiment 4

C. In which experiment is it most likely that Michael's system was using a GFS-style strategy? Assume that the master existed on a separate machine; Michael's data only reflects the disks actually used to store data. Also assume that each disk corresponds to a unique chunkserver. Circle the **best** answer.

Experiment 1 Experiment 2 Experiment 3 Experiment 4

[1] More accurately, to each disk's controllers. In practice, some disk controllers will optimize multiple small writes into a single larger write. However, we are **not** concerned with any sort of scheduling or optimization done by a disk's controller in this problem.

Moving on from his experiments, Michael begins to design his own system with four storage disks. In his system, when disks fail, they take five minutes to reboot and another 55 minutes to copy a full disk's worth of data from other disks in the system. You can assume that the time it takes to perform any computations—e.g., to calculate parity—is negligible; the bottleneck in his recovery process is the time it takes to copy data.

D. For each of the following failure scenarios, circle **True** if Michael's system will be able to fully recover (and **False** otherwise).

(a) **True / False** Michael is using RAID-4. Disk 1 fails. A little over an hour later, the parity disk fails.

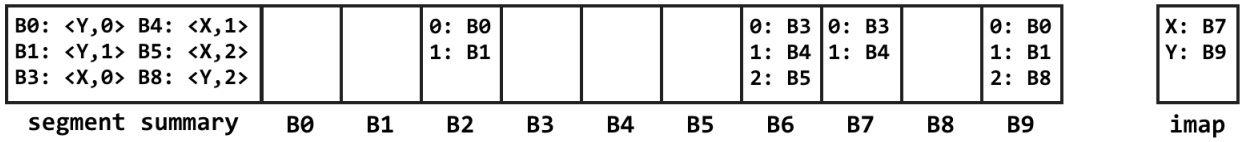
(b) **True / False** Michael is using RAID-5. Disk 1 fails. Thirty minutes later, Disk 2 fails.

(c) **True / False** Michael is using RAID-5. Disk 1 fails. During its recovery process, it fails again.

(d) **True / False** Michael is using GFS. Disk 1 fails. Thirty minutes later, Disk 2 fails.

II Atomicity and Isolation on a Single Machine

2. [6 points]: The following diagram shows the layout of a portion of a log-structured file system along with the relevant imap (which would be cached in RAM). `X` and `Y` are inode numbers. Any imap pieces that would be written to the disk throughout this segment are left out for clarity.



A. Which blocks, `B1`—`B9`, are **not** live? If all blocks are live, write "None".

B. GFS and LFS perform different functions, but both systems rely on several assumptions. For each of the following, decide whether the assumption applies to a single system, both, or neither (in cases where the assumption applies to both systems, circle both GFS and LFS).

(a) **GFS / LFS / Neither** Most workloads are read-only-after-write (i.e., after a file has been fully written, the system sees only reads to the file, not additional writes).

(b) **GFS / LFS / Neither** Contiguous writes to disks are more efficient than random writes.

(c) **GFS / LFS / Neither** Large appends to files are more common than random writes.

(d) **GFS / LFS / Neither** Sequential reads from disk are more common than random reads.

Initials:

3. [8 points]: Consider the following log snippet. The format for `UPDATE` records is `UPDATE var=<old value>; var=<new value>`.

Log Entry	Transaction ID	Record
	...	
228	100	BEGIN
229	100	UPDATE A=0; A=10
230	100	UPDATE B=0; B=20
231	101	BEGIN
232	101	UPDATE C=0; C=15
233	101	UPDATE D=0; D=25
234	101	COMMIT
235	102	BEGIN
236	102	UPDATE C=15; C=30
237	102	UPDATE D=25; D=40
238	102	ABORT
	/** CRASH **/	

Assume that the system operates only using the log, not with additional storage (e.g., cell storage).

A. After the crash, the system recovers, then executes a transaction that writes values into `A`, `B`, `C`, and `D`. Fill in the four blanks below to complete this transaction:

Log Entry	Transaction ID	Record
	/** RECOVER **/	
239	103	BEGIN
240	103	UPDATE A=_____; A=50
241	103	UPDATE B=_____; B=60
242	103	UPDATE C=_____; C=70
243	103	UPDATE D=_____; D=80
244	103	COMMIT

B. The original log snippet only reflects updates (writes) to variables. Suppose that transaction `101` did a read of variable `A` in between its two updates, and found the value of `A` to be 10.

Which of the following is correct? Circle the **best** answer.

(a) This is the expected behavior for a system implementing isolation via strict two-phase locking with read-/write-locks.

(b) This is *not* the expected behavior for a system implementing isolation via strict two-phase locking with read-/write-locks, but could occur in an incorrect implementation where transactions release the write-locks prior to their commit point.

(c) This is *not* the expected behavior for a system implementing isolation via strict two-phase locking with read-/write-locks, but could occur in an incorrect implementation where a write-lock for a variable can be held at the same time as a read-lock for that variable.

(d) Both (b) and (c).

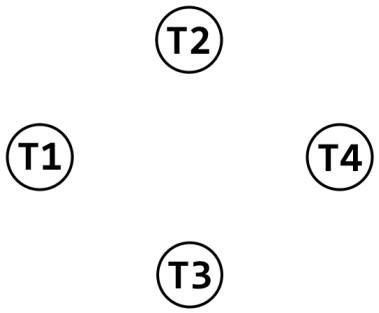
(e) None of the above.

Initials:

4. [8 points]: A transaction-processing system runs the steps of transactions `T1`, `T2`, `T3`, and `T4` in the order shown (time goes downwards; step i runs before step $i + 1$). All transactions commit after step 8 below.

T1	T2	T3	T4
1. write(x)			
	2. read(x)		
			3. read(y)
	4. write(y)		
		5. read(y)	
			6. read(z)
	7. write(z)		
		8. write(z)	

A. Draw the conflict graph corresponding to this schedule of steps.



B. It turns out that transaction `T3` is incomplete—it also needs to perform a read of x (i.e., `read(x)`). In terms of the semantics of `T3`, though, it does not matter where this read occurs (`T3` does no further operations on the variable x). To create a conflict-serializable schedule, where should this read happen?

(a) The resulting schedule will be conflict-serializable no matter where `read(x)` occurs in `T3`.

(b) The resulting schedule will be conflict-serializable only if the `read(x)` occurs in `T3` prior to `T1`'s `write(x)` (i.e., prior to Step 1).

(c) The resulting schedule will be conflict-serializable only if the `read(x)` occurs in `T3` after `T1`'s `write(x)` (i.e., after Step 1).

(d) It's impossible to create a conflict-serializable schedule that includes `read(x)` in `T3`.

Initials:

5. [6 points]: Two clients, C_1 and C_2, are writing the value of a variable `x` to a distributed storage system. Their sequence of writes is shown below. They are ordered in time.

Client	Action
C_1	`write(x, 1)`
C_2	`write(x, 2)`
C_1	`write(x, 3)`

No other clients write to `x`.

Immediately after all three writes are executed, C_2 reads the variable `x` three times. Below, we've given you some options for the results of those reads (again, ordered in time—each list contains the result of the first read, then the second, then the third). For each option, decide which consistency model(s) it reflects, if any. In each case, circle **all** that apply.

A. `3, 3, 3`
(a) Eventual consistency (d) Strong consistency
(b) Read-my-writes (e) None of the above
(c) Monotonic reads

B. `1, 2, 3`
(a) Eventual consistency (d) Strong consistency
(b) Read-my-writes (e) None of the above
(c) Monotonic reads

C. `2, 2, 3`
(a) Eventual consistency (d) Strong consistency
(b) Read-my-writes (e) None of the above
(c) Monotonic reads

D. `3, 1, 2`
(a) Eventual consistency (d) Strong consistency
(b) Read-my-writes (e) None of the above
(c) Monotonic reads

III Distributed Transactions

6. [7 points]: Anying's replicated state machine (RSM) has four machines: a view server, VS; a primary, P; and two backups B_1 and B_2. In all parts below, assume that VS never fails.

A. True / False Anying's system will remain available even if two of the remaining machines fail at the same time.

B. Anying gives you a list of the views from VS. Unfortunately, view 2 is missing:

```
1: P, B1, B2
2:
3: P, B1, B2
```

Which of the following are possible values for view 2? Circle all that apply.

(a) `2: B1, B2`

(b) `2: P, B1`

(c) `2: P, B2`

(d) `2: P`

(e) None of the above

C. Later on, suppose a portion of the network fails, preventing messages from being sent between VS and P. No other failures of any sort occur. VS deems P to be dead, and updates the view to

`4: B1, B2`.

Clients, however, are not currently aware of this change; nor is P, since it cannot communicate with VS. A client C sends a request to P to commit a transaction. Which of the following best describes the result? You may continue to assume that there are no additional failures in the network. Circle the **best** answer.

(a) The transaction will be committed.

(b) The transaction won't be committed because P won't be able to get a response from VS confirming the transaction.

(c) The transaction won't be committed because P will reject the request from C without forwarding it on to either backup.

(d) The transaction won't be committed because B_1 will reject the request from P.

7. [7 points]: Consider a Raft set-up with N nodes. In Term 1, Node X is elected the leader, and client C_1 sends two updates to X; call them `C1.1` and `C1.2`. At the beginning of Term 2, Node Y is elected the leader. You know that at the beginning of Term 2, both `C1.1` and `C1.2` have been committed.

A. True / False Assuming that there were no failures (i.e., no machines failed and no links went down), `C1.1` and `C1.2` must be reflected in the logs of all N nodes in the system.

No updates occur during Term 2, but a network partition occurs. An election timeout occurs, and Node Z is elected leader in Term 3. Y also remains a leader. Assume that there are no additional failures (including packet losses) in the network.

B. Which of the following is true? Circle the **best** answer.

(a) Node Y is on the smaller side of the partition; Node Z is on the larger side.

(b) Node Y is on the larger side of the partition; Node Z is on the smaller side.

(c) Nodes Y and Z are on opposite sides of the partition, but we can't tell which node is on the larger side.

(d) Nodes Y and Z are not necessarily on opposite sides of the partition.

Assume—regardless of the answer to Part B—that Nodes Y and Z were indeed on opposite sides of the partition. During Term 3, client C_1 sent Y two updates, `C1.3` and `C1.4`; Client C_2 sent Z one update, `C2.1`.

C. Assume that the network partition has healed and that the system has gone through multiple additional terms. There may have been additional failures. Circle **True** or **False** for each of the following.

(a) **True / False** It's possible that none of the updates `C1.3`, `C1.4`, and `C2.1` appear in any log.

(b) **True / False** It's possible that all of the updates `C1.3`, `C1.4`, and `C2.1` appear in every log.

(c) **True / False** It's possible that just one client's updates appear in every log (i.e., either `C1.3` and `C1.4` are in every log, or `C2.1` is in every log).

Initials:

IV Cryptographic Primitives

8. [6 points]: In all below,

- H is a cryptographically secure "slow" hash function. Assume that the output of H is sufficiently large (multiple bytes).
- s refers to a random salt, which is generated per-user. Assume that the salts are sufficiently large (multiple bytes).
- p refers to the user's password.

Isabel is building a system where users authenticate themselves via passwords. She stores each user's username on the server along with two additional pieces of information, x and y.

A. Which of the following values for x and y will allow Isabel to properly authenticate users without opening her system up to attacks from adversaries with read-access to her system (e.g., rainbow-table attacks)? Circle **all** that apply. In all cases below, these are the only two additional pieces of information that Isabel stores per user.

(a) $x = s$ $\quad y = H(p) \mid s$
(b) $x = s$ $\quad y = H(p \mid s)$
(c) $x = s$ $\quad y = H(p \mid H(s))$
(d) $x = H(s)$ $\quad y = H(p \mid s)$
(e) $x = H(s)$ $\quad y = H(p \mid H(s))$

In the current version of Isabel's system, user's passwords are eight bytes long, and the salt is a random sixteen-byte string. Isabel is sick of generating random salts. She decides, instead, to force users to pick passwords that are 24-bytes long, and simply stores $H(p)$ for each user. There are no more salts.

B. Which of the following is true? Circle the **best** answer.

(a) The system is more secure than it was before because now users will be forced to pick longer passwords.
(b) The system is as secure as it was before because it will take an adversary just as long to create a table mapping 24-byte passwords to their hashes as it would to create a table mapping 8-byte passwords + 16-byte salts to their hashes.
(c) The system is less secure than it was before because in practice, the entropy of the stored values will decrease.
(d) None of the above.

9. [8 points]: Pete and Xavier are very excited about the cryptographic primitives they learned about in 6.033, and are ready to set up a secure channel. In their excitement, they've generated quite a few things.

- Three distinct symmetric keys—k, k_1, and k_2—that can be used for encryption and decryption.
- Three random numbers (or "nonces"), n, n_1, and n_2.
- A stream of sequence numbers, $seq_1, seq_2, \ldots$. Assume that this stream is infinite and strictly increasing; sequence numbers do not "wrap".

Pete and Xavier are concerned about providing **confidentiality** against an adversary Eve in the network. Currently, when Pete wants to send a message m_P to Xavier, he transmits $ENC(k, m_P)$; when Xavier wants to send a message m_X to Pete, he transmits $ENC(k, m_X)$. Unfortunately their scheme is insecure against replay and reflection attacks.

Examine each of the schemes below and decide which attacks they're secure against, if any. In all below, H refers to a cryptographically-secure hash function. You can assume that Pete and Xavier have agreed on the scheme ahead of time, and are each aware of the values of all of the variables listed above. You can also assume that Eve does **not** know the values of any of the above variables.

A. To transmit m_P, Pete sends $ENC(k, m_P|H(n))$; to transmit m_X, Xavier sends $ENC(k, m_X|H(n))$

(a) **True / False** This scheme is secure against replay attacks.
(b) **True / False** This scheme is secure against reflection attacks.

B. To transmit m_P, Pete sends $ENC(k, m_P|H(n_1))$; to transmit m_X, Xavier sends $ENC(k, m_X|H(n_2))$

(a) **True / False** This scheme is secure against replay attacks.
(b) **True / False** This scheme is secure against reflection attacks.

C. To transmit m_P, Pete sends $ENC(k_1, m_P|H(n))$; to transmit m_X, Xavier sends $ENC(k_2, m_X|H(n))$

(a) **True / False** This scheme is secure against replay attacks.
(b) **True / False** This scheme is secure against reflection attacks.

D. To transmit m_P, Pete sends $ENC(k_1, m_P|seq_i)$; to transmit m_X, Xavier sends $ENC(k_2, m_X|seq_i)$ (seq_i is the next sequence number in the stream; both Pete and Xavier will ignore any earlier sequence numbers).

(a) **True / False** This scheme is secure against replay attacks.
(b) **True / False** This scheme is secure against reflection attacks.

Regardless of their security against reply and reflection attacks, all of the schemes above do provide confidentiality for Pete and Xavier's messages.

E. True / False As a result, if Eve tampers with any of Pete and Xavier's messages, they will be able to detect the tampering.

Initials:

10. [4 points]: Cathie is sending messages to Dustin using cryptographic signatures. For each message m, Cathie signs the message using her secret key, SK_C. When Dustin receives m, he gets Cathie's public key, PK_C, from a certificate authority, and uses it to verify that the signature is correct.

Unfortunately for Cathie and Dustin, an adversary Eve manages to take over the certificate authority. Which of the following is true? Circle **all** that apply.

(a) Eve now knows SK_C and so can forge messages to Dustin.

(b) Eve does not know SK_C, but can change the value of SK_C, meaning that Dustin will no longer be able to verify messages sent by Cathie.

(c) Eve can map Cathie's name to a different public-key, meaning that Dustin will no longer be able to verify messages sent by Cathie.

(d) Eve can map Cathie's name to a different public-key and impersonate Cathie.

(e) None of the above.

V "Secure" Systems

11. [4 points]: An adversary, Eve, is mounting a classic stack-smashing attack. To mount this attack, her code overwrites the return address of a function and causes the function to return to the wrong place.

To combat Eve, Karen writes a compiler that generates code that maintains two different stacks. The first is the usual kind but without the function return address. The second stack, stored in a different part of memory, is used only for the function return address.

Which attacks—as described in the paper "Beyond Stack Smashing"—does Karen's compiler protect against? Circle **all** that apply.

(a) The classic stack smashing exploit

(b) The arc injection exploit

(c) The function-pointer-clobbering subterfuge exploit

(d) The exception-handler hijacking exploit

(e) The heap-smashing exploit

(f) None of the above

12. [4 points]: Answer **True** or **False** for each of the following statements about DNSSEC.

(a) **True / False** DNSSEC improves availability (with respect to DNS).

(b) **True / False** DNSSEC provides confidentiality.

Initials:

13. [4 points]: Answer **True** or **False** for each of the following statements about Ross Anderson's paper "Why Cryptosystems Fail".

A. True / False As described in the paper, on an ATM card, if one stores the encrypted version of the customer's PIN on the magnetic stripe using a shared secret key, the account is easily broken into. In contrast, if one uses public key encryption, the account is much more secure.

B. True / False Anderson suggests that system designers must consider the operation of the equipment as part of the security of the system.

C. True / False The "dual control" concept, where two individuals must collaborate to perform a function, is inconvenient and is sometimes bypassed by people wanting to save time.

D. True / False Anderson suggests that if we had a set of "perfectly secure" components, a system composed of these components would also be perfectly secure.

14. [4 points]: The advent of botnets has changed the landscape of security threats. How? Circle **all** that apply.

(a) Botnets can be rented, so non-technically-saavy adversaries can utilize their attack power.

(b) Botnets are often comprised of particularly powerful machines, each capable of generating more traffic than the average user's machine.

(c) Botnets can be hard to take down. C&C machines are difficult to locate, and the bots themselves can be geographically distributed.

(d) None of the above.

15. [6 points]: Consider a botnet such as Torpig, which uses domain flux to make the C&C server difficult to locate. Circle **True** or **False** for each of the following.

(a) **True / False** Domain flux is necessary to allow Torpig's C&C server to change its IP mappings on the fly.

(b) **True / False** Domain flux makes Torpig more resilient to takeovers of the C&C server.

(c) **True / False** Because the authors of the Torpig paper have discovered that they can takeover a botnet by pre-registering future domains, most future bots using domain-flux will be able to be compromised in this way.

Initials:

16. [4 points]: Bitcoin's proof-of-work prevents Sybil Attacks by making it computationally infeasible for an adversary to do which of the following? Circle **all** that apply.

(a) Create multiple identities.

(b) Validate a block.

(c) Validate multiple blocks in a row (say, more than five blocks).

(d) None of the above.

17. [6 points]: To look up the address associated with the domain name `mit.id`, Blockstack does the following:

1. Uses the virtualchain to retrieve the public-key and the hash of the zonefile associated with `mit.id`.
2. Uses the zone database to retrieve the zonefile associated with the hash of the zonefile returned in Step 1.
3. Makes a request to the URI contained in the zonefile for the address associated with `mit.id`.

Each step in this process is verified in one of the following ways:

- Using **hash functions**: the user can verify that the hash of the relevant piece of data matches the expected hash for that data.
- Using **digital signatures**: the user can verify that the relevant piece of data was signed by the appropriate public key.
- Using **the blockchain**: the user can verify that the relevant piece of data exists in the blockchain.

In each part below, decide how the user verifies a particular step in this process. Circle the **best** choice for each answer.

A. What aspect of the system is used to verify that the lookup in the virtualchain (Step 1) returns the correct result?

(a) Hash functions
(b) Digital signatures
(c) The blockchain

B. What aspect of the system is used to verify that the lookup in the zone database (Step 2) returns the correct result?

(a) Hash functions
(b) Digital signatures
(c) The blockchain

C. What aspect of the system is used to verify that the response to the request in Step 3 is the correct value associated with `mit.id`?

(a) Hash functions
(b) Digital signatures
(c) The blockchain

Initials:

VI Wrap-Up

This question is not worth any points.

18. [0 points]: Which 6.033 paper did you like the most? If you have multiple favorite papers, feel free to circle more than one option.

(a) "The UNIX Time-Sharing System"
(b) "Eraser: A Dynamic Data Race Detector for Multithreaded Programs"
(c) "MapReduce: Simplified Data Processing on Large Clusters"
(d) "The Design Philosophy of the DARPA Internet Protocols"
(e) "Resilient Overlay Networks"
(f) "Bufferbloat: Dark Buffers in the Internet" (by Gettys)
(g) "Comments on Bufferbloat" (by Allman)
(h) "Data Center TCP (DCTCP)"
(i) "The Akamai Network: A Platform for High-Performance Internet Applications"
(j) "The Google File System" (GFS)
(k) "Log-structured File Systems" (LFS)
(l) "Concurrency Control and Recovery"
(m) "Replicated Data Consistency Explained Through Baseball"
(n) "In Search of an Understandable Consensus Algorithm" (Raft)
(o) "Beyond Stack Smashing: Recent Advances in Exploiting Buffer Overruns"
(p) "Why Cryptosystems Fail"
(q) "Security Vulnerabilities in DNS and DNSSEC"
(r) "Your Botnet is My Botnet: Analysis of a Botnet Takeover" (about Torpig)
(s) "Blockstack: A Global Naming and Storage System Secured by Blockchains"

End of Quiz 2

Please double check that you wrote your name on the front of the quiz,
circled your recitation section, and initialed each page.

Initials: