Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.033 Computer Systems Engineering: Spring 2022

Exam 1

There are **14 questions** and **12 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 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: SOLUTIONS

1. [6 points]: Terryn is browsing the web and clicks a link to `http://web.mit.edu/6.033`. Her laptop queries a DNS server `K` to learn an IP address corresponding to `web.mit.edu.`.

A. Which of the following must be true? Select all that apply. Because we are asking what must be true, you must consider all possibilities for K. Namely that it's caching responses

(a) `K` must contact one of the authoritative name servers for `mit.edu.` to resolve the domain name. No - K could have a cached copy

(b) `K` must contact one of the root name servers to resolve the domain name. No - see above

(c) If `K` had answered a query for the IP address corresponding to `web.mit.edu.` at some time in the past, then `K` will correctly respond to Terryn's current query without contacting any other name server. No - cached copy may have expired

(d) If all authoritative name servers for `mit.edu.` have failed, and if `K` has never queried for the IP address of `web.mit.edu.` before, then `K` will be unable to get the IP address for `web.mit.edu.`. No - K could get a cached copy from a different nameserver

[✓ (e)] None of the above must be true.

Terryn, who lives in Cambridge, decides to learn more about this query, and runs `dig web.mit.edu` herself. She receives the following answer section.

```
;; ANSWER SECTION:
web.mit.edu.                1454 IN   CNAME www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net.    60   IN   CNAME e9566.dscb.akamaiedge.net.
e9566.dscb.akamaiedge.net.  20   IN   A 104.102.112.162
```

At the *exact* same moment, Amelia is vacationing in California and runs `dig web.mit.edu`. Assume that all DNS caches in the Internet are up to date; no name server that these queries encounter will see stale data. There are no failures of any sort.

B. Would you expect Amelia's query to also resolve in the IP address `104.102.112.162`?

No.

web.mit.edu is hosted by Akamai, which will send users to the physically closest server with the content. Terryn & Amelia are in different states → different servers ⇒ different IP addresses.

2. [3 points]: Kriti is in charge of running the webserver for Katrina's Amazing Web Suite, which hosts the site `www.kaws.com`. Currently the IP address of the server is `1.2.3.4`, but Kriti wants to upgrade to a larger server that has IP address `5.6.7.8`. Because of DNS, visitors to `www.kaws.com` will see minimal disruption in service as Kriti makes the upgrade. Why? Select the **best** answer.

[✓ (a)] **Indirection:** Because DNS introduces a layer of indirection via naming, Kriti can switch the IP address of the webserver without changing the hostname.

(b) **Hiding:** Users' machines don't need to know the IP address of `www.kaws.com` at all, only the hostname. Users' machines certainly do need the IP address - that's why they use DNS

(c) **Addressing:** Since `www.kaws.com` indicates a physical location, the IP address can change without disrupting the system. Hostnames do not indicate a physical location

Initials: SOLUTIONS

3. [8 points]: Consider the following program running on a Unix system. This program outputs a series of 0's and 1's, as those are the only two possible values for the variable i.

```
int main() {
  int i;

  for (i = 0; i <= 1; i++) {
    fork();
    printf(''%d '', i); // prints the value of i
  }
}
```

A. How many new processes are created by this code? Do not include the original process that runs `main()` in your count.

3.

The original process — call it A — forks a new process, B. Then A & B both fork again, for 3 total.

B. Recall that the Unix scheduler provides no guarantees on the order in which processes run. What are the possible outputs of this program?

This program will print two 0's and four 1's. The question is really about when the 0's could end up; we don't have to go through every possible sequence of processes.

At least one 0 will get printed first; call X the process that prints it. X then forks a process Y. We'll see one of three things.

1. 011011 (X and its 2nd child process finish before Y)

2. 001111 (X prints, Y prints, then the rest finish up in whatever order)

3. 010111 (X prints, then either it or its second child prints 1, then Y prints)

Notice that we can't see something like 011110; there's no way for all of the 1's to print before the 2nd 0.

Initials: SOLUTIONS

4. [10 points]: Sabrina's operating system uses 8-bit virtual addresses. Sabrina examines the memory translation for a process p, and sees the MMU translate a **virtual address** v into the physical addresses `10001010` (this address is in binary, not hex).

The entire page table for process p is below. Each page-table entry is simply the physical page number in binary; there are no additional bits such as read/write bits.

100	*
101	
100	*
110	
001	
111	
111	
000	

A. What is the value of the **virtual address** v in **bits**? If there is not enough information to determine the value of some of the bits in v, write a ? for those bits (e.g., if you could not determine the value of any bit in v, you'd write ????????).

Physical addr is 100|01010. Page 100 is mapped from row 0 (000) and row 2 (010). This means $v \in \{000\ 01010, 01001010\}$, i.e., $v = 0?001010$

(Remember offset is preserved

B. What is the page size, in **bytes**, for Sabrina's operating system?

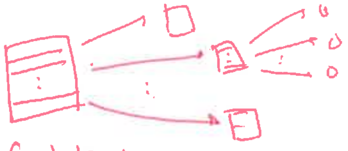
Offset is 5 bits, which means each page holds $2^5 = 32$ bytes of memory.

(Memory is byte-addressable, so the answer is not 32 bits = 4 bytes)

Sabrina moves onto a second operating system, which has no relation to the first. This OS uses multi-level page tables: the first a bits of a virtual address index into the first-level page table, the next b bits index into the second-level page table, and the next c bits index into the third-level page table.

C. For a single process p, what is the **maximum** number of page tables that Sabrina's operating system will generate? Give your answer in terms of a, b, and c.

$1 + 2^a + 2^a \cdot 2^b \quad (= 1 + 2^a + 2^{a+b})$



1. 1 first-level table, a indexes into it, so it has 2^a rows

2. 2^a second-level tables (one per row of the first-level table

3. 2^b third-level tables for each second-level table

Initials: SOLUTIONS

D. Still using her second OS, Sabrina observes a memory access for virtual address v, which triggers **exactly** one exception that was then handled by the kernel. What might have been the cause of this exception? Select all that apply.

(a) The OS needed to allocate a new first-level page table

(b) The OS needed to allocate a new second-level page table

If the OS needed to generate one of these, it'd also need to generate down to the 3rd-level. → > 1 exception.

[✓ (c)] The OS needed to allocate a new third-level page table

[✓ (d)] A process was attempting to write to a read-only piece of memory

(e) None of the above

5. [5 points]: For each of the following questions, select whether the statement applies to hard disk drives (HDDs), solid state drives (SSDs), both, or neither. Circle the correct answer for each part.

A. HDDs / SSDs / [✓ Both] / Neither Batching writes is an effective technique for improving performance.

B. [✓ HDDs] / SSDs / Both / Neither Cost effective for storing large amounts of data across hundreds or thousands of machines. Good for datacenters

C. HDDs / SSDs / Both / [✓ Neither] Provides volatile storage. Storage ≠ memory!

6. [4 points]: In much of our code involving threads—for example, the original `send` code for a bounded buffer and `yield_wait`—we use a paradigm where a thread—call it Thread A—releases a lock ℓ and immediately attempts to acquire it again. Why? Circle the **best** answer.

(a) As a performance enhancement, to increase the speed of Thread A.

(b) As a performance enhancement, to decrease the amount of storage Thread A uses.

(c) To avoid deadlock by letting Thread A make progress on a different part of its computation.

[✓ (d)] To avoid deadlock by letting a second thread acquire the lock ℓ and perform an action that will allow Thread A to progress.

(e) To avoid race conditions between Thread A and a second thread.

Example: In send(), a thread releases ℓ so that another thread can come in and read a message (creating space in the buffer for the send() call to complete).

8. [6 points]: Suppose Eraser observed the following code on multiple threads during program execution:

```
int v;

acquire(a);
acquire(b);
v = v + 1;
release(b);
release(a);

acquire(b);
acquire(a);
v = v + 1;
release(a);
release(b);
```

A. What is the lock set for the variable v after executing this code? Assume Eraser's initial lock set algorithm (e.g., no enhancements for variable initialization, read-write locks, etc.)

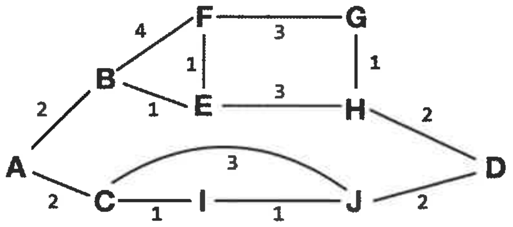
$\{a, b\}$

B. Will Eraser issue a warning for this code?

No. $\{a, b\} \neq \emptyset$

C. Is this code correct? By "correct" we mean that this code is free of race conditions and will always run to completion (e.g., it will not deadlock).

No. Because the locks aren't acquired in a consistent order, we can see thread 1 holding a and needing b, and thread 2 holding b and needing a.



9. [10 points]: Consider the network above, with link costs given next to each link.

A. What is the min-cost path from A to D?

A - C - I - J - D

This network uses link-state routing. A generates its first advertisement, which is exactly 100 bytes. This advertisement is then flooded through the network.

B. How many bytes of traffic are sent to flood A's first advertisement? Write down any assumptions you make about the flooding process as part of your answer.

$100 \times 2L = 100 \times 2 \times 13 = 2600$ Bytes

↑ # of links

C. Assume that the network is stable and every node has calculated its shortest paths to every other node. **Which links in the network could fail** without changing *any* of the shortest paths in the network? If there are no such links, write "None".

C - J ↓ use C-I-J , B - F ↓ use B-E-F

Notice that H-D is not correct. It is not enough to look only at shortest paths out of A. H-D is not used in any of A's paths, but it is the shortest path from H to D.

A wants to send TCP traffic to D. A knows that there are at least two distinct paths to D: one through B and one through C. A overrides the link-state routing protocol and uses both paths, sending packets with even sequence numbers to B and odd packets to C. A does **not** change anything about TCP, nor do any other nodes in the network change their approach to routing.

D. Which of the following **best** describes the problem with this approach? Select only one answer.

(a) Packets may arrive out of order at D causing the receiving application to receive its data out of order. No - TCP will fix this

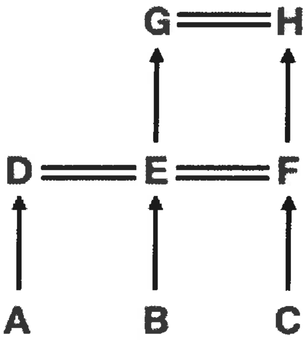
B's best path is via E

(b) Routing loops. B will route packets back through A, because its best path to D is via A.

(c) ACKs will not travel on the correct reverse path for all packets. Doesn't really matter

[✓ (d)] TCP will struggle to correctly estimate the window size, and A may end up transmitting more slowly than if it had just used a single path. This is the same problem we see with multipath routing in datacenter networks

Initials: SOLUTIONS



10. [12 points]: Consider the above graph of autonomous systems (ASes) using BGP. The direction of arrows indicate the direction in which money flows (A → D means A pays D for transit); traffic can flow in both directions. Peers are denoted by double lines. In this graph, ASes A and D are able to send traffic to each other, and to E and B, but not to any other ASes because of the rules of BGP (namely that E will only advertise to D about itself and its customers).

A. Suppose D is willing to buy transit from an AS X in order to increase its connectivity. What possible value(s) of X will allow D to send traffic to every other AS? If there are no such values, write "None". Assume that D is allowed to buy transit from *any* AS in the graph except E and A.

B, C, F, G, H

Buying transit from any of these ASes opens up the righthand side of the graph.

B. Suppose D is willing to peer with an AS Y in order to increase its connectivity. What possible value(s) of Y will allow D to send traffic to every other AS? If there are no such values, write "None". Assume that D is allowed to peer with *any* AS in the graph except E and A.

G, H

If D peers with B, C, or F, it won't gain access to their providers

C. Suppose we use RON, and place a RON node in every single AS. Will RON allow E (not D) to send traffic to any destinations that it could not send traffic to before? If so, list the destination(s) below; if not, write "None"

None.

E can already reach all destinations

D. Will RON allow E to utilize any *paths* that it could not use before? If so, list the path(s) below; if not, write "None"

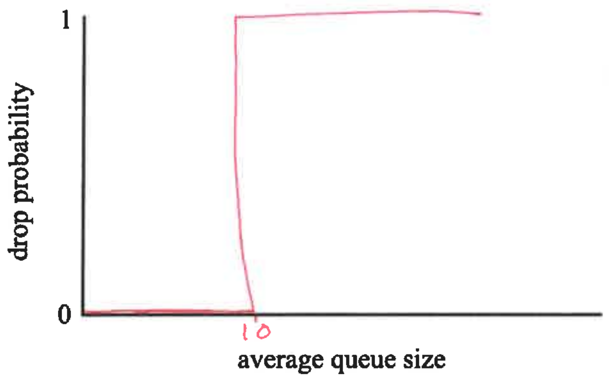
Yes. E-F-H is the most obvious new path. It would also technically gain paths E-F-H-G and E-G-H-F-C, even though E would be unlikely to select those paths. (We gave full credit for any subset of these paths.)

Initials: SOLUTIONS

11. [9 points]: Suzanna runs a datacenter where all traffic runs over DCTCP. Each switch in the datacenter is capable of holding 500 packets in its queue, but because she's utilizing DCTCP, the **average** number of packets in each queue is quite low: just twenty packets.

Amir runs a completing datacenter that has the exact same physical infrastructure as Suzanna—including switches that are capable of holding 500 packets. All of the traffic in Amir's datacenter runs over TCP, and his switches implement RED with the following parameters: $q_{min} = q_{max} = 10; p_{max} = 1$.

A. On the axis below, plot the average queue size vs. the drop probability in Amir's datacenter. Your plot should be clearly labeled.



Notice that Amir's datacenter is actually using DropTail with a queue size of 10

B. Assuming the same workload in each datacenter, what would you expect to be true about in Amir's datacenter compared to Suzanna's? Select all that apply.

[✓ (a)] Packets in Amir's datacenter will spend less time waiting in queues.

(b) There will be better tolerance for large bursts of data in Amir's datacenter

Amir's datacenter is terrible for this!

(c) Flows are less likely to synchronize in Amir's datacenter

No, because he's using DropTail

(d) None of the above

You gain access to some packet traces from one of these datacenter networks. In the packet traces, you see a TCP flow with a window size of 12. In the next round-trip-time, the window size for this flow drops to 10.

C. Which datacenter was this trace from? Circle the **best** answer.

(a) Amir's datacenter

[✓ (b)] Suzanna's datacenter

(c) There's not enough information to tell

We never see a slight decrease like this with TCP, only DCTCP

Initials: SOLUTIONS

12. [9 points]: Consider a switch with two queues for incoming traffic: one for Zoom traffic and one for email. All traffic through this switch is either Zoom or email; there is no other type of traffic. All packets are 1000 bytes and the switch's goal is to split the outgoing bandwidth equally between Zoom and email over both short and long time scales. The switch is using deficit round robin (DRR) to schedule packets.

The questions below reference short-term and long-term fairness. "Short-term" fairness means fairness measured over a few seconds or minutes. "Long-term" fairness means fairness measured over a few hours or days.

A. Assume that there is enough traffic such that, at the start of each round, each queue has at least one packet in it. What should each queue's quantum be to maximize both short-term and long-term fairness and minimize latency?

1000 Bytes.

To minimize latency and maximize fairness, we want to alternate: one Zoom packet, one email, one Zoom, etc. A smaller quantum means we waste time accumulating small amounts of credit (so latency increases). A larger quantum means short-term fairness is worse

Now assume that, starting at time t, the switch sees **only Zoom traffic**. The quantums are set as described in Part A. After ten minutes the switch starts to receive email traffic again. Though there may be bursts of traffic from one application or the other, once again, at the start of each round, each queue will have at least one packet available to send.

B. What is the drawback of DRR in this scenario? Select the **best** option.

See below

(a) The switch will send **only** email traffic for approximately the next ten minutes.

[✓ (b)] The switch may see **significant** short-term unfairness over the next 10-20 minutes.

(c) The switch may see **significant** long-term unfairness over the next day.

(d) The switch may see significant short-term unfairness (over the next 10-20 minutes) **and** significant long-term unfairness (over the next day).

C. How could you fix the drawback described above? Select the **best** option. You need only consider this specific scenario; don't worry about how this fix would affect other scenarios (more types of traffic, different traffic patterns, etc.).

(a) Lower the value of both quantums

(b) Increase the value of both quantums

(c) Lower the value of the email queue's quantum

(d) Increase the value of the email queue's quantum

[✓ (e)] Prevent queues from accumulating credit when they're empty

The problem is that the email queue has built up a ton of credit. Once email traffic returns, the email queue will get emptied each round until credit gets back to zero (eg, a burst of 10 email packets will all go out)

Initials: SOLUTIONS

13. [4 points]: Which of the following best describes how Akamai uses DNS to direct a user who is trying to visit `web.mit.edu` to the appropriate Edge Server?

(a) The client issues a DNS request that gets routed through the DNS hierarchy to one of MIT's authoritative nameservers, which then forwards the request to an Akamai name server, which responds to the client with the IP address of the appropriate edge server.

(b) The client issues a DNS request that gets routed through the DNS hierarchy to one of MIT's authoritative nameservers, which then forwards the request to the appropriate Akamai edge server.

[✓ (c)] The client issues a DNS request that gets routed through the DNS hierarchy to one of Akamai's name servers, which responds to the client with the IP address of the appropriate edge server.

(d) Akamai does not use DNS as part of its system.

In fact you can see this happen in Question 1 of this exam!

14. [6 points]: As part of Katrina's Amazing Web Suite, Katrina has designed an application-layer protocol called KAWS. KAWS traffic always utilizes TCP as the underlying transport protocol.

As packets generated by KAWS travel across the Internet, they're encapsulated by lower-level protocols. Felipe observes one of these packets as it traverses an Ethernet link, and is able to see the original packet data as well as four headers, one for each layer. Packets, then, look like this, where each element (A-E) is either a packet header or the original packet data:

A	B	C	D	E
Link	Net	Trans	App	Data

A. Which element (A-E) corresponds to the TCP header?

C

B. Which element (A-E) corresponds to the Ethernet header?

A

In hopes of improving performance for her amazing web suite, Katrina decides to utilize a different addressing scheme. Instead of IP addresses, she develops KP addresses, which have an entirely different format. Instead of using IP as the network-layer protocol, she uses KP.

C. What else would Katrina need to do in order to successfully send these packets across the Internet? Assume that Katrina's packets will travel over a fixed path (and the ACKs will travel the same path in reverse) and that path will never change. Select the **best** answer.

Network layer = "Narrow waist" of the Internet. We need the whole Internet on board.

(a) Make sure that the destination endpoint was also using KP, but make no changes to the switches on the path.

(b) Make sure that the destination endpoint and some of the switches along the path were using KP.

[✓ (c)] Make sure that the destination endpoint and all of the switches along the path were using KP.

(d) Nothing; KP will work fine without any changes.

Initials: SOLUTIONS