CS425 Fall 2020 – Homework 3
(a.k.a. “Apollo 111”)


Topics: Snapshots, Multicast, Consensus, Paxos, Leader Election, Mutual Exclusion
(Lectures 13-18)

Instructions:

1. Attempt any 8 out of the 10 problems in this homework (regardless of how many credits you’re taking the course for). If you attempt more, we will grade only the first 8 solutions that appear in your homework (and ignore the rest). Choose wisely!
2. Please hand in solutions that are typed (you may use your favorite word processor. We will not accept handwritten solutions. Figures and equations (if any) may be drawn by hand (and scanned).
3. All students (On-campus and Online/Coursera) – Please submit PDF only!
   Please submit on Gradescope. [https://www.gradescope.com/]
4. Please start each problem on a fresh page, and type your name at the top of each page.
5. Homeworks will be due at the beginning of class on the day of the deadline. No extensions. For DRES students only: once the solutions are posted (typically a few hours after the HW is due), subsequent submissions will get a zero. All non-DRES students must submit by the deadline time+date.
6. Each problem has the same grade value as the others (10 points each).
7. Unless otherwise specified, the only resources you can avail of in your HWs are the provided course materials (slides, textbooks, etc.), and communication with instructor/TA via discussion forum and e-mail.
8. You can discuss lecture concepts and the questions on Piazza and with your friends, but you cannot discuss solutions or ideas on Piazza.
9. For Fall 2020 semester, we are making the following exceptions:
   a. You can discuss up to 4 (out of 8) problems with a group of at most 2 other students who are in this semester’s CS425 class. This is to encourage remote “group work” among students. (We encourage setting up Zoom sessions to discuss.)
   b. At least 4 (out of 8) problems must be on your own.
c. For each solution you write, please say at the top of the solution, either “Discussed with: Hermione Granger (netid: hgranger1), Harry Potter (netid: hpotter1),” OR say “Solved completely on my own”.

d. We encourage that with each subsequent homework, you try to monotonically increase the number of questions you solve “on your own”. If you can solve all HW4 questions on your own (last homework), that will give you confidence!

Prologue: It is the year 2049 A.D. Most of you are in your middle age. Cloud computing, as we know today, does not exist – it’s now called “Solar Computing”. Sure, there are a few quantum computers here and there, but transistor-based computers still rule the roost in 2049. Datacenters are still around, and all the distributed computing concepts you’re learning today in CS425 still apply. The only catch is that datacenters are much smaller (100x) than they were back in the 2010s, but more powerful – this means an entire AWS zone from 2010s can now be stored in one small spaceship!

Anyway, Moon has been colonized by humans. Man is next going to land on Mars. A manned spacecraft New Horizons X is being launched to Mars. Once on board, you meet the astronaut team led by Commander Amelia Brand, Pilot Rheya Cooper, and including you and ten other astronauts. The spacecraft carries its own powerful datacenter. You are the sole “Solar Computing Specialist.” You must ensure that you troubleshoot and solve all problems that arise in the on-board distributed system (solving any 8 out of 10 problems would also suffice to save the mission).

All characters and storylines are fictitious, and purely intended to keep the reading entertaining; these are not intended to be educational. Any resemblance to persons, places, animals, things, or events, living or dead, past, present, or future, is purely coincidental.

Problems:

1. 3...2...1... Liftoff! You’re off to Mars. During liftoff you’re browsing code (what else?). You realize that one of the Earth programmers has written an algorithm for synchronous consensus (the same as that discussed in class), but your datacenter is asynchronous! Your fellow astronauts believe this is not a problem, and you know they are wrong. You know that once you exit the Earth’s atmosphere, cosmic rays will increase in frequency and they can knock out an arbitrary number of machines simultaneously. Point to where the synchronous consensus proof (discussed in class) breaks down when the system is asynchronous. Quick, you’re about to exit the atmosphere!
2. You have detachment from the rocket! Suddenly you see your pet cat (who you had named “Doraemon” when you adopted her) in the corridor of the spaceship—you try to follow her but she disappears. You wonder if it’s your imagination. Anyway, bigger cats to catch… You switch communications on. You see a chart of the multicast communications between your spacecraft New Horizons X (NHX), Earth station, Moon station, and Mars (unmanned). If these stations use the FIFO Ordering algorithm, mark the timestamps at the point of each multicast send and each multicast receipt. Also mark multicast receipts that are buffered, along with the points at which they are delivered to the application.

3. As New Horizons X is passing through the Van Allen belts, the spacecraft’s reactor and engines suddenly shut down. Oops, you realize that you should have used causal ordering in the previous timeline (Question #3). Can you redo it quickly before your spacecraft crashes? Again, mention clearly all timestamps and all buffered messages.

4. Just this morning you also saw your pet dog (whom you had named “Einstein” when you adopted him on Earth) roaming inside the spaceship’s corridor. You called out to him by name and he stared at you, but then ran away. You are perplexed, and you talk to the captain of your spaceship, Commander Amelia Brand. She asks you to take some rest. But there are miles to go before you sleep… Suddenly you wake up and see that you are looking at the FLP proof discussed in class. You are puzzled by a couple of things. Can you explain them briefly?
   
   i. You don’t quite understand how, in Lemma 3, Case II, one can have $p' = p$, especially since the set $C$ is obtained by not applying event $e = (p,m)!$ Can you explain this discrepancy?
ii. Why in the proof of Lemma 3, Case II does C \textit{definitely} have a deciding run, i.e., schedule S? What if C never decides? Does the proof not hold then?

iii. It appears that Lemma 3’s proof will hold no matter which event \( e \) is selected for the argument (\( e \) applicable on configuration C). Is this true, or does \( e \) need to have some special characteristics?

5. Now your spaceship is passing by the Dark Side of the Moon. It’s a glorious view! To ensure things are working properly you decide to run the Chandy-Lamport snapshot algorithm on the ongoing communications between your spacecraft, and the manned Earth station, and manned Moon station. But due to a crash at the different stations, the algorithm only outputs the following timeline. In the figure, a, b, c, … are regular application messages. You can use S(a) to denote the send event of a and R(a) to denote its receipt event. Markers shown as dotted lines. Can you help the intern find the snapshot recorded by this run? Don’t forget to include both process states and channel states. For process states, you can use the name of the latest event at that process (For initial state, just say “Initial state”. Note that Markers don’t count as events). Quick, it’s up to you to manually calculate the snapshot!

6. You’re still doing well physically and emotionally in this long trip, mostly because you were trained well at Illinois. You’re about halfway through the trip to Mars. As you’re retiring to your room to sleep, you see both your cat Doraemon and dog Einstein walking together in the spaceship corridor. You call out to them, but they run away again. Before you can chase them, you notice the spacecraft wobbling quite a bit, and you need to fix this. You trace the wobbling...
problem to the on-board storage system, and the fact that there is no leader election algorithm in there! Quick, you need to design one!

The datacenter onboard (with hundreds of machines) uses a ring-based DHT (among the machines) with a Chord-like routing algorithm with each peer maintaining 3 ring successors and 3 ring predecessors. This system needs to elect a leader that has the DHT Id closest to the midpoint in the system, i.e., if the Chord ID has say 32 bits, we want to elect the leader whose ID is closest to \(2^{31} - 1\).

i. Design a leader election protocol that is efficient in that it uses very few Chord messages, i.e., \(O(1)\) Chord messages per participant (A Chord message is a query from a node to a key, with all intermediate hops counting as part of the Chord message). The only messages you can use are the DHT routing messages (Chord_Route(sender_ID, destination_ID)), and Chord DHT routing will route your message.

ii. Argue briefly why your algorithm satisfies safety and liveness when finger tables are all correct and there are no failures during execution (proof not needed).

iii. What is the completion time and number of messages in your leader election protocol (both asymptotic)?

iv. Discuss briefly what might happen if failures occur during the election run, while finger tables stay inconsistent.

7. Space travel has lots of free time, and you’re spending it reading about Mae Jemison, the first Black woman to go to space. Suddenly an inspiration strikes you and you realize that in order to make the spacecraft reach your destination faster, you can code a new algorithm! In a distributed system, a process sends out multicasts sequentially, but is allowed to send multiple multicasts one after another (before the previous multicasts are delivered at itself). Now, write an algorithm to implement FIFO-total (i.e., hybrid) ordering for multicasts in such a distributed system. Argue (informally and briefly) why your algorithm provides the ordering property. There are no failures, and you can assume reliable multicast and unicast are available to you. You may use a sequencer.

8. Your spacecraft needs to perform a slingshot (gravity assist) in order to land on Mars. However, this means going through the dreaded Asteroid belt between Mars and Jupiter! Before the slingshot, you realize the normal leader election algorithm will not work, and that for fault-tolerance you will need multiple leaders. Solve the k-leader election problem (for a given value of k). It has to satisfy the following two conditions:
• Safety: For each non-faulty process $p$, $p$’s elected = of a set of $k$ processes with the lowest ids, OR = NULL.
• Liveness: For all runs of election, the run terminates AND for each non-faulty process $p$, $p$’s elected is not NULL.

Modify the token ring algorithm described in lecture to create a solution to the $k$-Leader Election problem. You should not have more than $k$ tokens at any time in the ring. Your token(s) may also carry information inside it (and this information can be rewritten by processes). Briefly discuss why your algorithm satisfies the above Safety and Liveness, even when there are failures during the algorithm’s execution (if you need to make assumptions about failure count to prove these properties, state them clearly).

9. Bam! Your New Horizons X spacecraft has just suffered a massive strike from an asteroid! Alarms are going off all around you. And a dog can be heard yelping and a cat can be heard whelping in agony, somewhere inside the spacecraft. You talk to Commander Brand and your colleague on board Pilot Rheya Cooper about this, and they counsel you that the animals are your imagination. Anyway, back to work… You quickly figure out that the alarms are because of the mutual exclusion algorithm implemented in the system – if you can fix it, the spacecraft will return to normal operations.

You see that someone has implemented the Maekawa algorithm, but to “make it more efficient” (a dangerous phrase, if there was one!) they have “optimized” the voting set members similar to how Ricart-Agrawala works. In other words, when a voting set member $pi$ receives a Release message from a process $pj$ (which is exiting the critical section), the voting set member process $pi$ now immediately multicasts a Reply message to all waiting processes in $pi$’s queue, and empties the queue. Give a concrete scenario where this algorithm violates mutual exclusion.

10. Well you fixed one, but now you’ve uncovered another bug! Alarms are still going off all around you. You need to quickly design a file system to back up data. In doing so you encounter a new mutual exclusion problem. Consider a file $F$ that is present in a distributed system of $N$ processes ($N$ large). There are no failures or message losses in the system. The mutual exclusion required on this file has the following safety and liveness conditions (different from those discussed in lecture):

**Safety**: At most 1 process may obtain write access to the file simultaneously. At most $k$ processes may obtain read access to the file simultaneously. If any process has write access to $F$, no other process should be able to read it. If any process has read access to $F$, no other process should be able to write it.

**Liveness**: Requests to access and release the resource eventually succeed.
Answer these three parts:

a. Briefly describe a token ring-based distributed algorithm for the above problem (pseudocode would be a good idea). Your algorithm must not have more than $k$ token messages in the system at any point of time (simultaneously). Hint: Token message can contain writable fields.

b. Argue briefly that your algorithm guarantees all the Safety clauses. (a formal proof is not required, however you are free to write one).

c. Can your algorithm livelock, i.e., violate Liveness? Suggest an idea to address this issue. Argue that your idea reduces the frequency of livelocks (you don’t need to prove Liveness).

d. Given one process that is currently writing, and $k$ processes waiting to read, what is the synchronization delay (i.e., time for all the $k$ processes to start reading, i.e., the last one)? Calculate both best case and worst case. Since this calculation may be hard, you can assume for simplicity that: i) reads take quite long (i.e., tokens are not released by readers until everyone has started reading), ii) tokens cannot be combined into one message, and iii) in one time unit, only one message can be transmitted, anywhere in the system (the latter means that it suffices to calculate the total number of token transfers for the synchronization delay). Also $N \gg k$. If you need to make other simplifying assumptions, be reasonable and specify them clearly. Show all your calculations.

e. (Optional, no points for this part, answer only if you want to) When you set your foot on Mars, as the first human to do so, what will be your first words to the world? (Neil Armstrong had great words, but try to make yours epic!).

--- (HW2 Official End) ---

Epilogue (If you’d like to not spoil surprises, don’t read this epilogue until you’ve read the Prologue and questions above): As you take humankind’s first steps on Mars, you look back at the Horizons X lander spacecraft. You see your dog Einstein and cat Doraemon together peering down at you through the porthole window. You remember they are indeed real, and that you did indeed bring them along with you from Earth! The long trip and cryogenic sleep made you forgetful! You realize that Einstein and Doraemon were just too disoriented by the space travel experience, and that’s why they kept running away from you. It all makes sense now!

The Earth station, from millions of miles away, speaks in your earpiece, “Congratulations! You just completed the first solo human mission to Mars! Woohoo!” You’re happy, but then you stop and ask Earth station, “Solo?! What about the other ten
astronauts? What about Commander Amelia Brand and Pilot Rheya Cooper who were with me?” There is a pause. Earth station responds, “Ten astronauts? Brand and Cooper…? Who…?”

“Just kidding!” says Earth station, following up with, “Brand, Cooper, and other astronauts were lifelike holograms we created to keep you company. On this long journey to Mars, without human company, we knew you’d go mad!... By the way, from their vital signs, we see your pet dog Einstein and cat Doraemon are finally feeling themselves. Say Hi to them for us!"

--- The End ---

(PS: Did you catch all the sci-fi references in this homework?)