

# CSCI 455 Project 1 Discussion

Your Name

February 25, 2019

1. *Which criterion of the critical section problem is not satisfied by the initial implementation? Explain.*

The “mutual exclusion” criterion was not satisfied, since both cats and dogs ran in the playroom without regard to whether the other was or not. The room was being used freely, not exclusively.

2. *Describe the cats-and-dogs synchronization problem in terms of the events that each pet must wait for.*

If there are any dogs in the room, a cat must wait until the last dog has left, before it can enter. Similarly, if any cat is in the room, a dog must wait for all the cats to leave.

3. *Do you need to add another synchronization construct to the implementation to handle birds? Explain.*

You don’t strictly need to add a new synchronization construct. I did, by adding a condition variable referring to when the birds leave the room, but I later realized that is not the best option. Rather, the bird synchronization could use all the same constructs as the dog, seeing as they are interchangeable in this phase of the simulation. The birds and dogs can keep track of the same condition variables.

4. *Why would using a broadcast instead of signaling pets one at a time result in more useless wakeups than are necessary?*

A broadcast will wake up all animals waiting on a single condition. If any animals are waiting on the same variable, but can’t actually be in the room simultaneously, then some of them will be uselessly woken up. In my case, this was especially for the wolves, since many wolves might be waiting to get in the room, but when they all wake up, only one is allowed in.

5. *Meeting the bounded waiting criterion for the critical-section problem was not required for this assignment, but it is possible that your solution does. If your implementation satisfies it, explain why. Otherwise, describe what additional variables or constructs that would be needed to implement it and when they would be updated.*

My solution does not satisfy bounded waiting, which is why I set NUM\_BIRDS and NUM\_MICE to 1; otherwise, the chances of dogs, birds, and mice leaving the room at the same time are very low. In order to solve it, a sort of timeslice would be required so that every so often the animals in the room have to leave and let others in. I would set the timeslice to about 3 seconds, and every 3 seconds, every animal would have return to their cages, and the animal

that has been waiting the longest will get woken up, while the animals who were removed end up at the back of the line (highest wait\_time number).

6. *The rules regarding the wolves provide an example of a classic synchronization problem (one of the ones mentioned in class). Which problem is it? What do the wolves correspond to and why?*

The wolves correspond to the writers in the Readers-Writers problem. Writers can only write to a file exclusively one at a time, so they don't overwrite each other's work or change what another thread is in the process of reading, while wolves can only run in the room exclusively one at a time so they don't devour any other animal.

7. *This assignment initially used the term "deadlock" to refer to the situation where a pet is never woken up, even after the room empties. Why specifically was this the wrong term to use?*

Deadlock requires circular waiting (e.g. cats waiting for dogs, waiting for birds, waiting cats, etc.), which means that during deadlock, no opportunity arises for waking up a waiting thread. But if an animal is not woken up when it should be, that's just a bug.

8. *Suppose the pet shop wants to give their pets more time to play. They buy the office next door and create a separate shop identical to the first, with the same rules. They want to be sure that both playrooms are being used as much as possible, so if the pets in Shop A are not currently playing, some pets will be brought from Shop B and vice versa. Describe an analogous issue from the textbook's chapter on scheduling and how it would apply to the pet shop's logistics in this scenario.*

This is similar to the issue of scheduling on multiple processors. It takes time to move animals from one location to another, just like it takes significant time to migrate processes between processors, and the processes may not share a cache. So an implementation of this would have to require some form of push and pull migration between shops.

9. *If the two shops were connected instead, so that each pet could choose to wait for Playroom A or Playroom B, would it be possible to reach deadlock? Explain.*

There would still not be a possibility for deadlock in this instance because there's no circumstances in which an animal in a room needs to wait; nothing is stopping the animal from leaving when it wants to. However, deadlock requires the "hold and wait" property, meaning that acquiring one lock prompts the acquiring of another lock. The playrooms, though, can be left whenever.