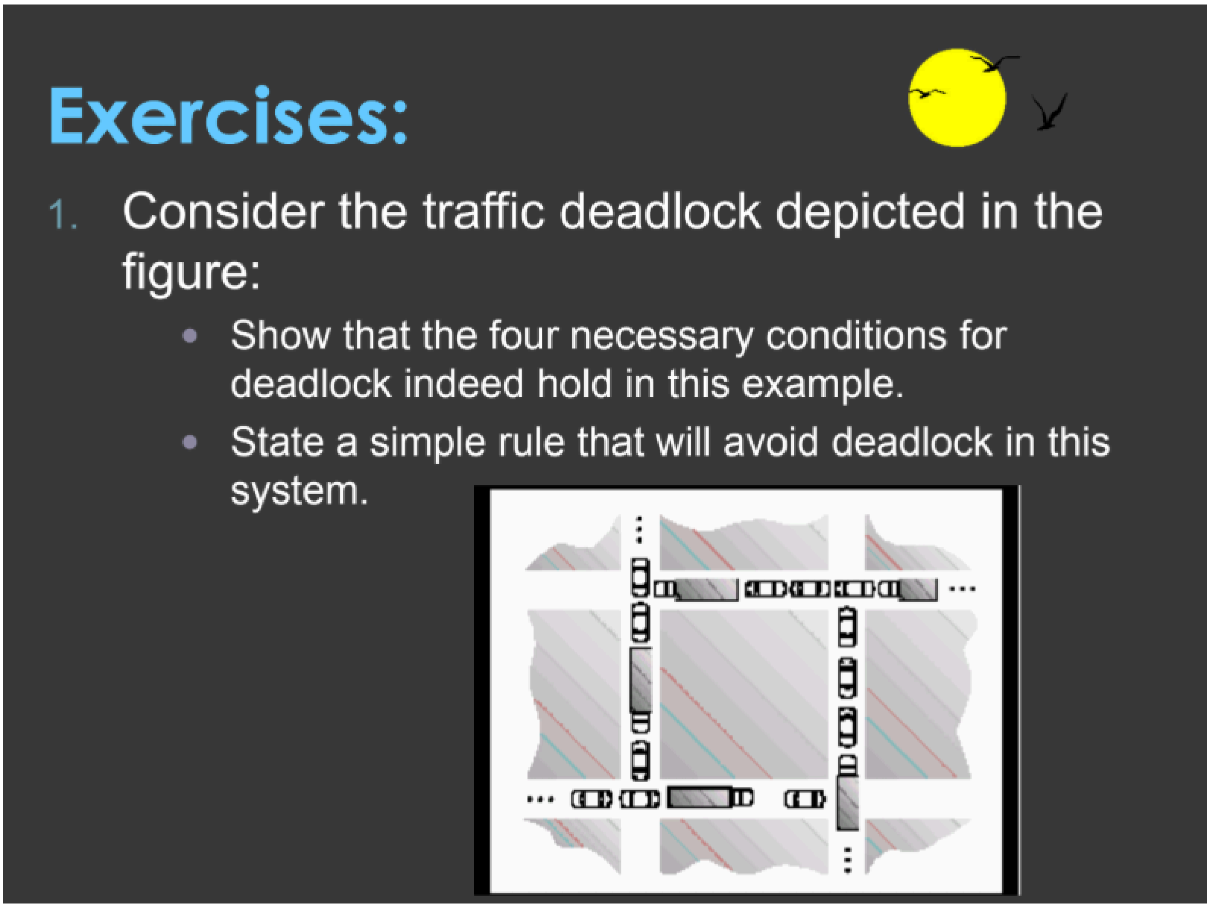
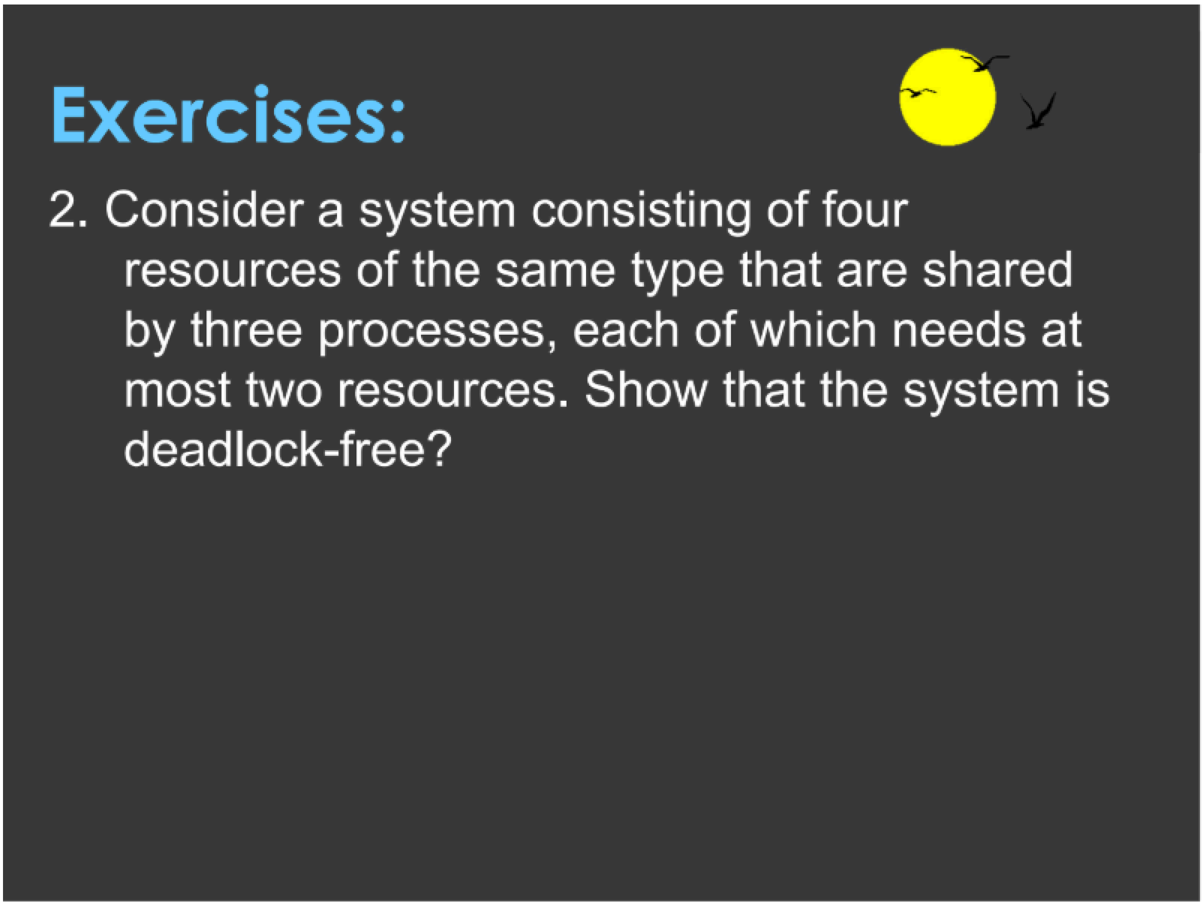
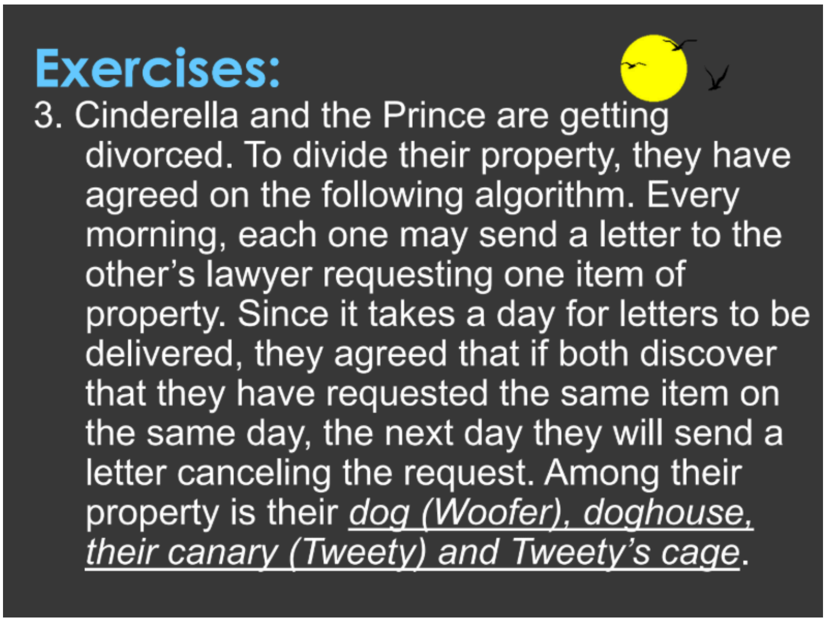
Michael Xavier Canonizado
BSCS 3A
Operating Systems
Deadlock Exercises

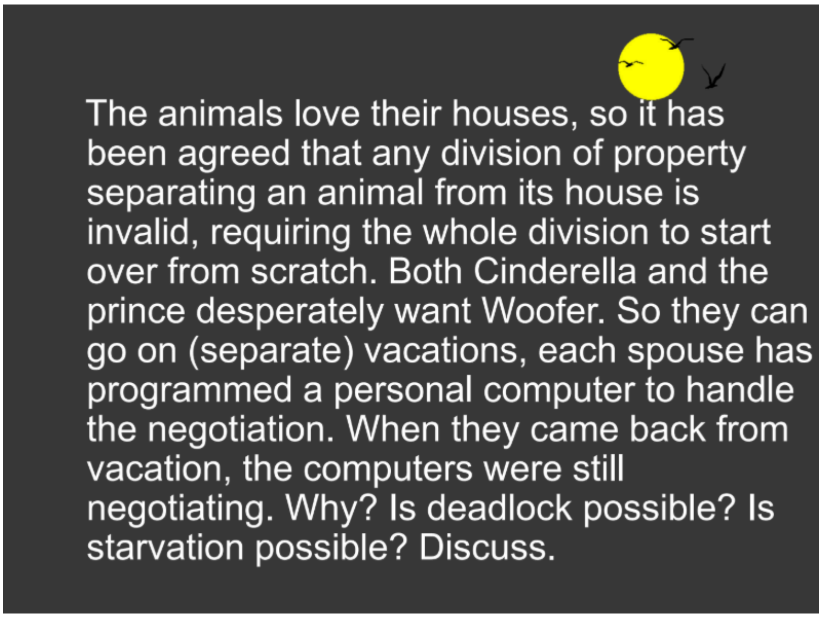


1) Mutual Exclusion
    - Each car needs access to the road in front of it. If two cars cannot occupy the same road at the same time, the road cant be shared hence mutual exclusion.
2) Hold and Wait
    - A lane is waiting for the cars in the other lane it merges to, to move.
3) No Preemption
    - A car cannot give up its position on the road. Since it cant be moved, a deadlock cannot be resolved.
4) Circular Wait
    - Similar to hold and wait, but a cycle. The north lane is waiting for the cars to move from the west lane, which is also waiting for cars to move from the south lane, which is also waiting for cars to move from the east lane, which is waiting for the north lane to move. This creates a circular wait.



Sine there are 4 identical resources and 3 process that each needs at most 2 of those resources. The system is deadlock free as there will always be an extra resource that a process can utilize, finish its task, and free its resource.

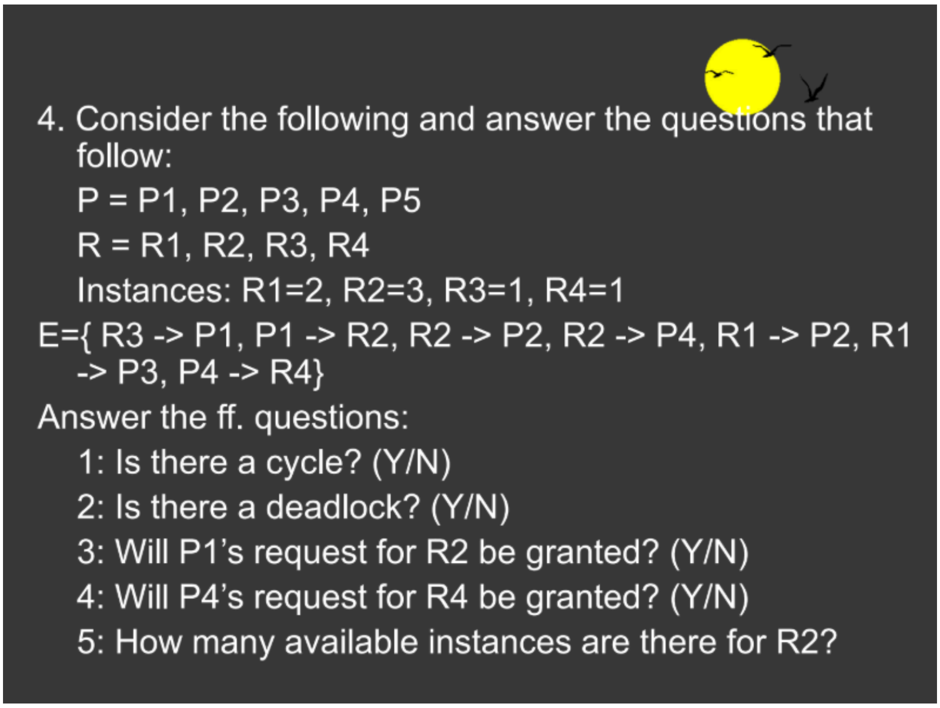


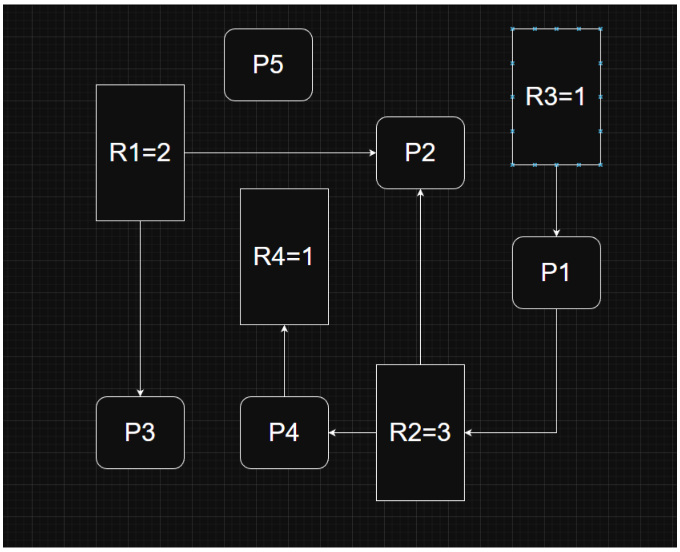


No, it's not a deadlock, it's a livelock since not all parties are blocked but are just going on indefinitely.

- Since both parties want and prioritize to get Woofer, a livelock will happen.
- Both parties want Woofer, so each morning their program sends a request for Woofer.
- If both parties request at the same time on the same day for Woofer, they will send a cancellation on the next day.
- The next day, they will still request for Woofer as they again prioritize getting woofer, resulting in the same loop.
- Another thing that could cause a loop is when 1 party requests for Woofer and the other requests for the dog house, causing the division to restart.

There is also no starvation as starvation means that 1 side keeps on losing and the other side keeps on winning. In this case no one is winning, both are losing.





1. Is there a cycle?
   - No
2. Is there a deadlock?
   - No
3. Will P1’s request for R2 be granted?
   - Yes it will be granted. There is still an available instance for P1
4. Will P4’s request for R4 be granted?
   - Yes it will be granted, though R4 only has 1 instance, it is not being used by any other process so P4’s request will be granted.
5. How many available instances are there for R2?
   - Considering P1’s request for R2 will be granted, there will be no available instances left to be allocated.

5. Eight (8) processes P_0 through P_7 with 4 resource types: *A* (10 instances), *B* (5), *C* (11) and D (6).

	Allocation	*Max*
	A B C D	*A B C D*
P_0	0 1 2 1	7 4 9 2
P_1	2 1 0 0	3 2 2 0
P_2	0 0 2 0	9 0 7 2
P_3	2 1 0 2	2 2 4 3
P_4	0 0 2 0	4 3 6 3
P_5	2 0 2 0	2 1 3 2
P_6	2 1 1 1	5 2 2 1
P_7	1 0 2 0	7 4 8 5

Question: Is the system safe? Justify your answer.