COMP 3430

Operating Systems

June 12th, 2019

Goals

By the end of today's lecture, you should be able to:

- Describe the concept of deadlock. (textbook)
- Identify code that may become deadlocked. (textbook, in-class)
- Describe tools/strategies that can be used to avoid deadlock. (textbook, in-class)
- Evaluate strategies for implementing locks. (textbook)
- Describe locks and semaphores. (textbook, in-class)

Assignment/test questions?

Discussion: Locks and scheduling

When we add locks to concurrent code, the scheduler seems to become our enemy.

Why?

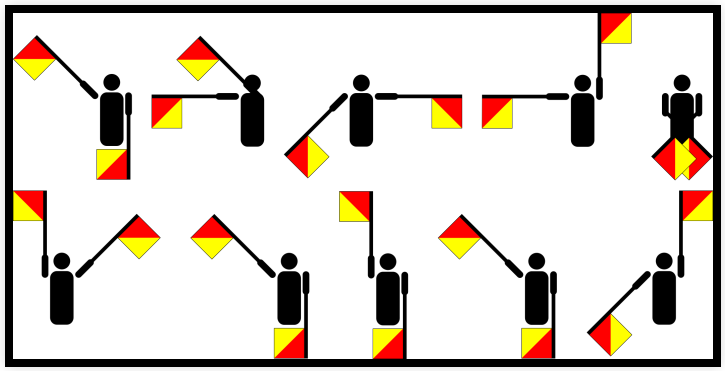
Pixabay License

Why is it called “semaphore”?

How to learn why Dijkstra used the word “semaphore”:

1. ~~Learn Dutch.~~
2. ~~Learn how to raise the dead.~~
3. ~~Read the paper.~~
4. …
5. Generally: semaphore == signaling system.

Semaphores



- A *counter* that can be used for locking .
  - Interface: initialize, decrement/wait, increment/signal (see `man sem_overview`)
  - Implementation: a counter + a queue (see `semaphore.h` and `semaphore.c`)

Deadlock

… and the four conditions

Deadlock can only happen when all four hold:

1. Mutual exclusion
2. Hold-and-wait
3. No preemption
4. Circular wait

Mutual exclusion

Threads claim exclusive control of resources they require (e.g., a thread grabs a lock).

Hold-and-wait

Threads hold resources allocated to them (e.g., locks that they have already acquired) while waiting for additional resources (e.g., locks that they wish to acquire).

No preemption

Resources (e.g., locks) cannot be forcibly removed from threads that are holding them.

Circular wait

There exists a circular chain of threads such that each thread holds one or more resources (e.g., locks) that are being requested by the next thread in the chain.

Deadlock

Deadlock can only happen when all four hold:

1. Mutual exclusion
2. Hold-and-wait
3. No preemption
4. Circular wait

Let's look at some code and decide ⌛:

- *Can* this code deadlock if executed concurrently?
- Pick **one** condition for deadlock. How would you change the code to break that condition?

Goals

You should be able to:

- Describe the concept of deadlock.
- Identify code that may become deadlocked.
- Describe tools/strategies that can be used to avoid deadlock.
- Evaluate strategies for implementing locks.
- Describe locks and semaphores.

Artist: Yes, of course I've seen a horse.