COMP 3430

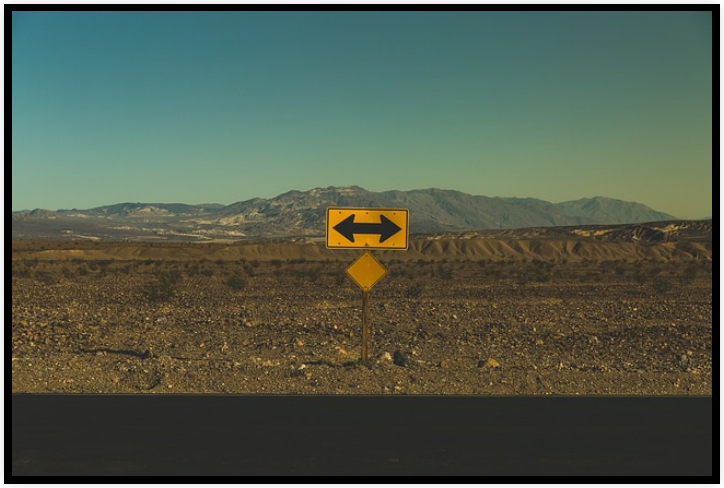
Operating Systems

May 27th, 2019

Goals

By the end of today's lecture (and readings), you should be able to:

- Describe problems that come up with concurrent code
- Select appropriate strategies for dealing with concurrency
- Write a program that employs concurrency
- Describe different strategies for message passing (IPC)

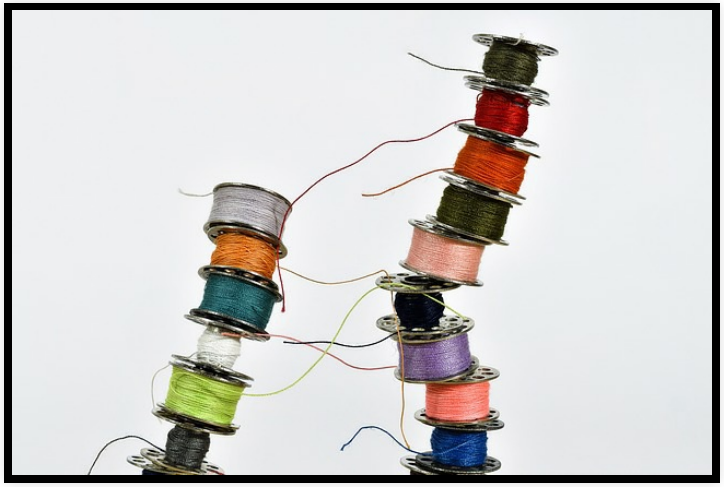


Pools

1. Let's draw pictures to help remind ourselves of what a threadpool is.
2. Let's *keep* looking at some code.

Summarizing Pools

- Threads are *cheap* – but **not** free.
- Thread*pooling* is a design pattern to help reduce:
  - Long-term costs of thread creation.
  - Cost for creating many, short-lived task threads.



Pixabay License

Problems with threads

- Threads are *great*!
  - … even if they aren't *free*.
- Threads can be *tricky*
  - Let's take a look at some code that *has* problems and we'll try to **identify** the problems.
    1. `thread_clobber.c`
    2. `thread_clobber2.c`
    3. `thread_clobber3.c`

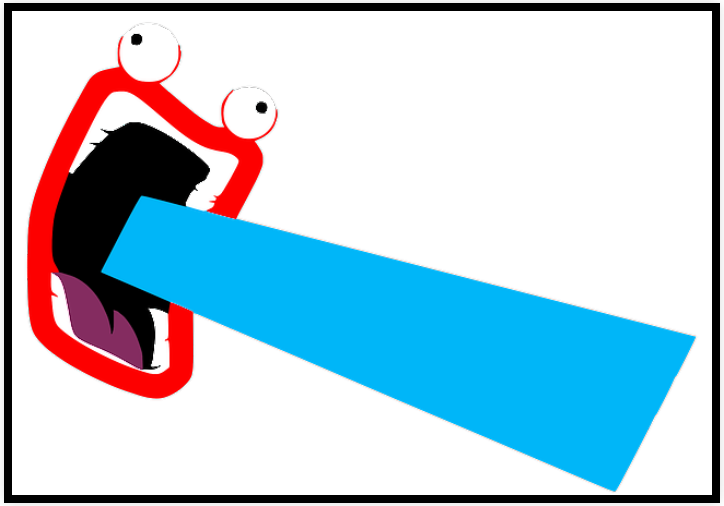
Thread-world problems. (Pixabay License)

Threads, processes, and concurrency

- Processes are course-grained concurrency primitives provided by an OS.
  - Many unrelated things working at the same time.
- Threads are fine-grained concurrency primitives provided by an OS.

Communicating among *processes*

- Communicating among threads is *straightforward*
  - Think about it: What **enables** communication among threads *within* a process?
- We want to be able to communicate among *processes*.
  - Why *couldn't* we communicate among processes?



Communicating among processes

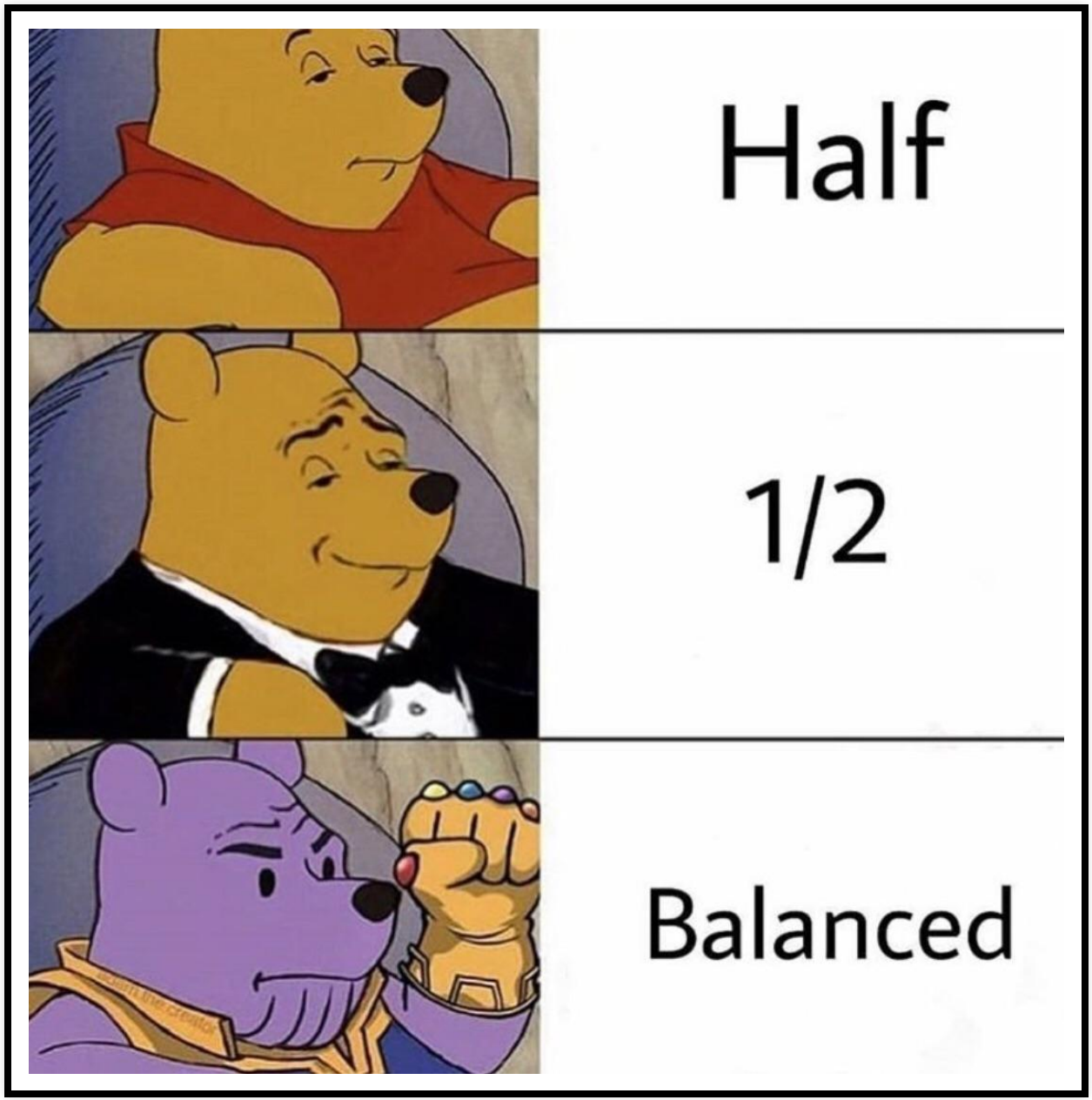
- … let's back up for a second:
  - Discussion: Why do we even *want* to communicate among processes?
  - For thought: What does this have to do with the OS?

Methods for communication

We're going to look at 3 approaches for **I**nter **P**rocess **C**ommunication (IPC):

1. Signals.
2. Files (… but *differently*)
3. Shared memory.

- Something to think about for next time: How do these three approaches (based only on their names) resemble communication with *threads*?


Half
1/2
Balanced