

## 3/6 - Memory Problems

Spin me up some Spin.

- Suppose we have several processes operating over shared memory
  - How do we prevent the processes from getting into each other's way?
  - e.g. when both want to write & read from an array at the same time
- How do we control access to our critical section (array)? What protocol?
  - Suppose we don't have a scheduler or third agent - procs need to coordinate themselves
- What properties do we want?
  - No starvation - if a proc wants something, it eventually gets it
  - Mutual exclusion - if one process is working in the critical section, the other process isn't.
- What's a potential solution?
  - Have a flag for each process that means, for some process p, that p requests access to the critical section
    - What's wrong with this?
    - There's a scenario where both processes flags are raised but neither process can do work - both are frozen (deadlock)
    - If there's some evil or dysfunctional scheduler, one thread might raise its flag then get stopped before checking the other flags, then the other process grabs its flag - now both processes hold flags but neither can proceed.
  - Another strategy: polite processes
    - Before entering the critical region, a process labels itself the 'victim' and blocks until it becomes a non-victim (when the other proc defers)
    - Seems to work! Doesn't give deadlock
    - However, we get a livelock when one thread doesn't contend for the critical section - the other thread victimizes itself and waits, but never actual takes control to resume