

CS 3214 lecture #9

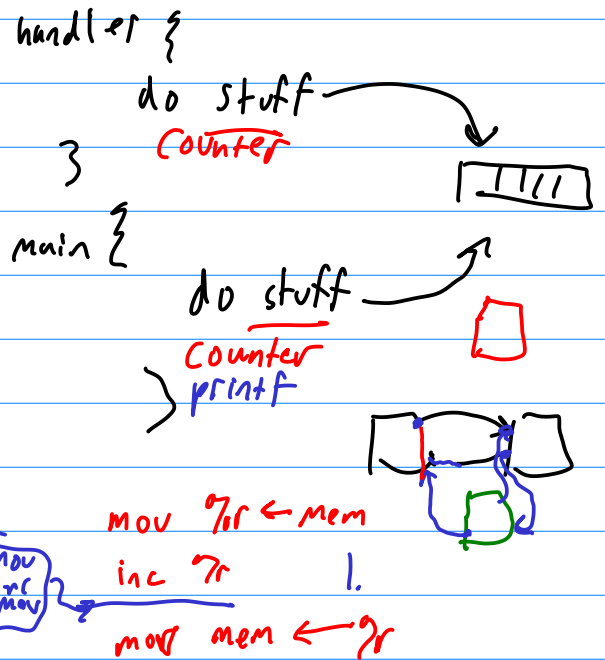
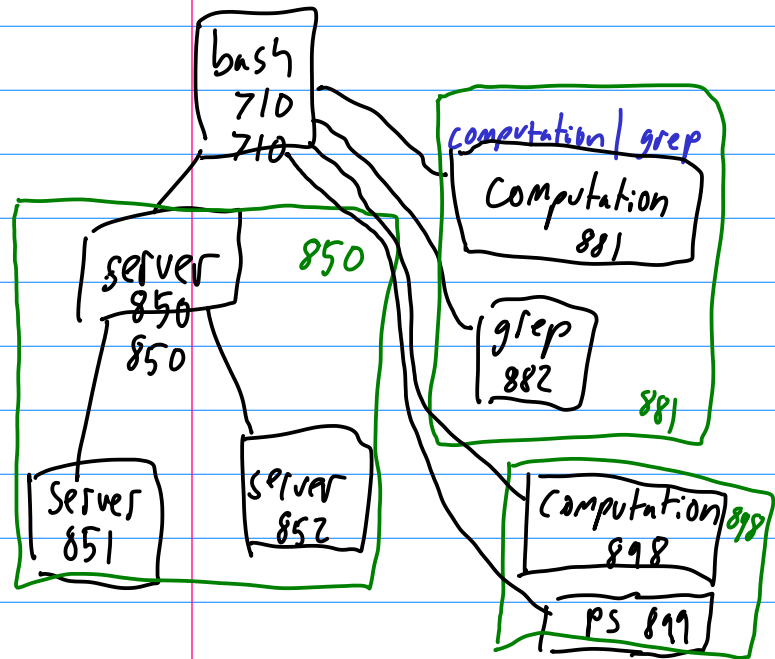
project 1

1. process group wrap up ← jobs fg/bg
2. safe handling of asynchronous signals ← SIGCHLD
3. undefined behavior ← big project in C

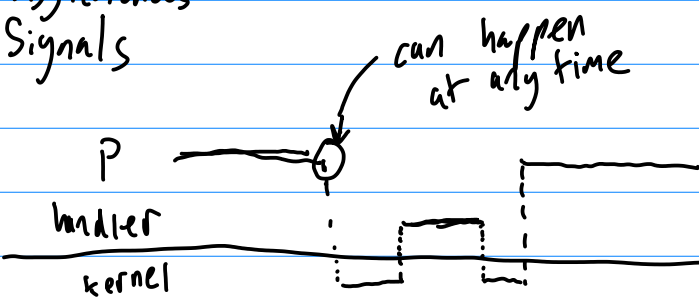
shell recap "job control" { ls | grep | wc }
fg/bg "pipeline"

how to make sure signals are delivered to all

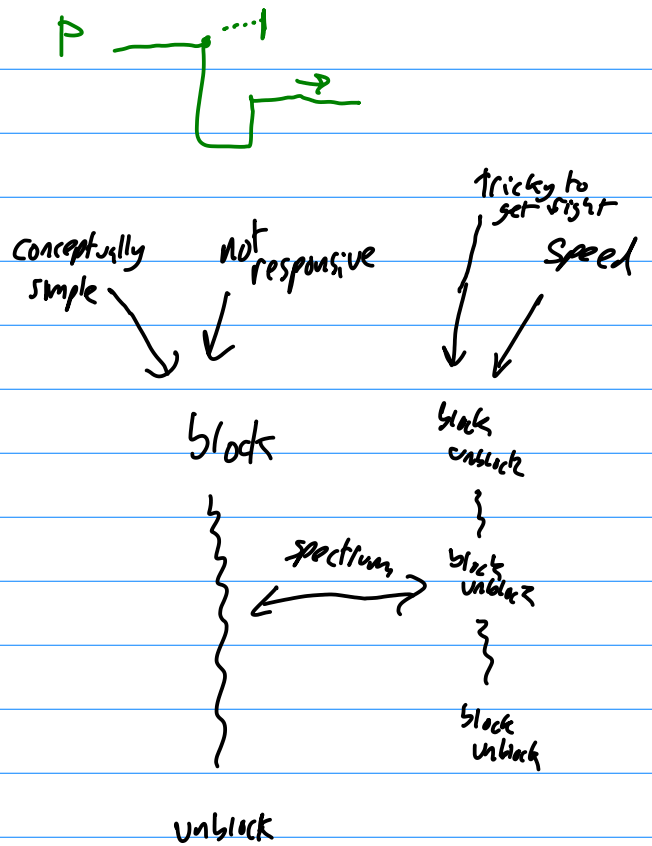
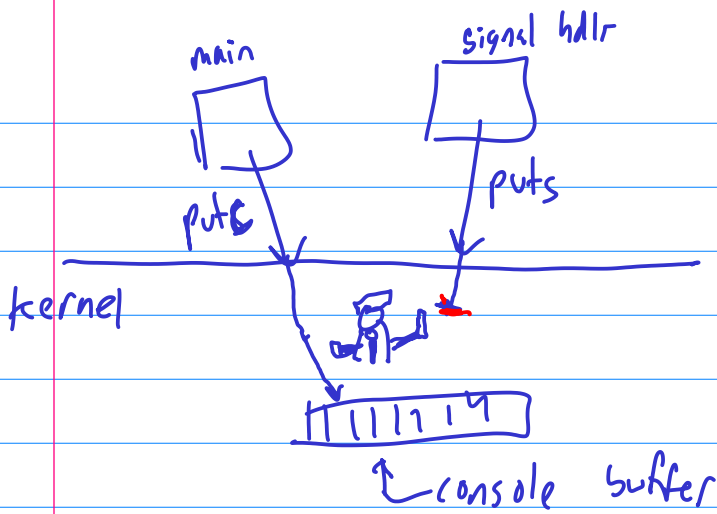
b



Asynchronous Signals



mov %r ← mem
inc %r
mov mem ← %r



How to prevent?

- block signals (mask)
- make signal handlers lean

C and "Undefined behavior" UB

SHOULD
SHOULD NOT } compiler takes advantage of this

Result: surprising & horrible bugs

signed integers overflow

$x+1 > x$ ← compiler: always true!
↑
MAX_INT

$x * 2 / 2$ compiler: always x