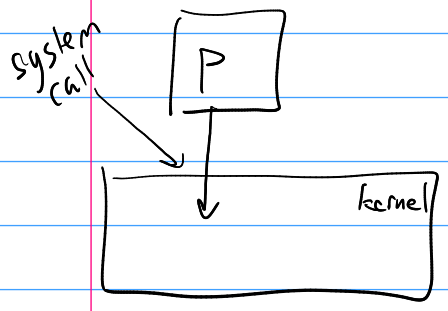
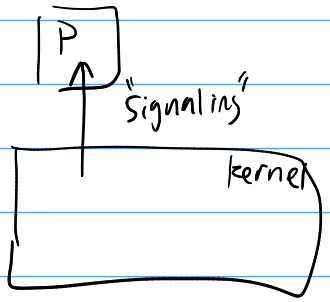
CS 3214 lecture #7 "signals" ← SIGCHLD

SIGTSTP ^Z





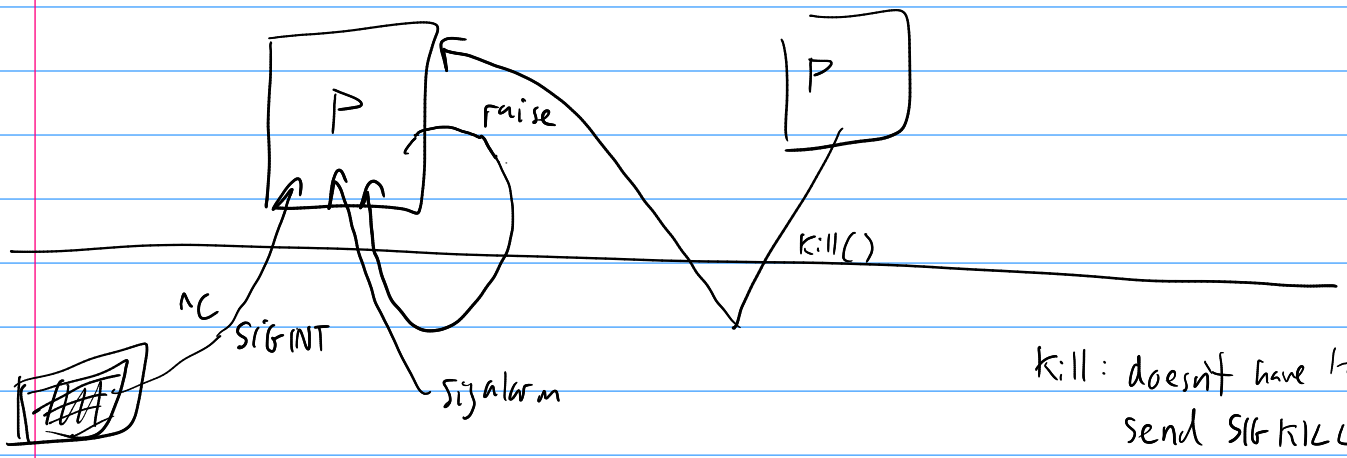
^C
^Z
kill -9 <pid>
SIGKILL
SIGSEGV

2 types: asynchronous (external) something else
Synchronous (internal) because of program

UNIX: unified API
- signal identified w/ int (< 32)
- program can specify what to do on signal
  - terminate
  - ignore
  - "catch" a signal: invoke signal handler
  - stop
  - continue

} job ctrl

Usual default behavior: terminate
fail-stop



kill: doesn't have to send SIGKILL

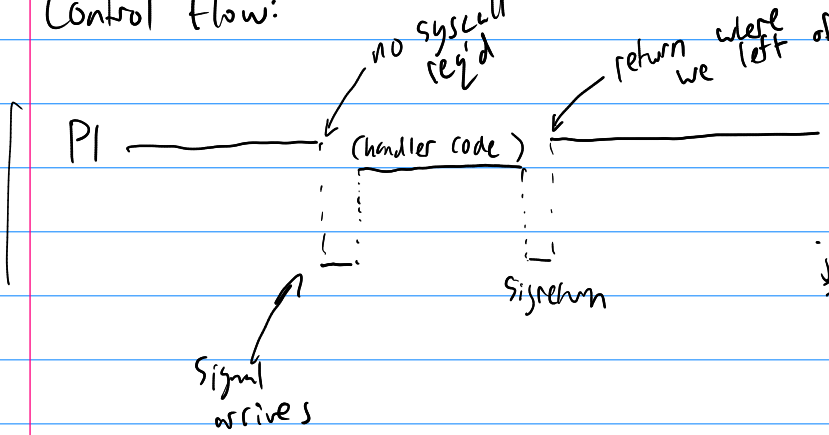
kill: program vs builtin → function impl. in the shell

"normal"

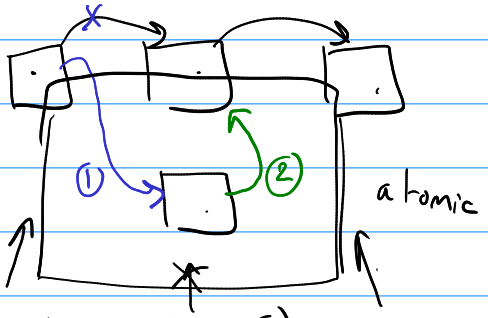
If a signal arrives, is it handled immediately? delivered/

bitmap specifies which signals are pending / kernel delivers later

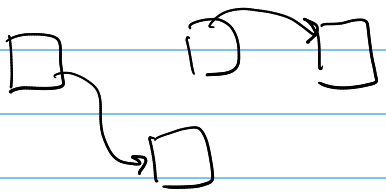
Control flow:

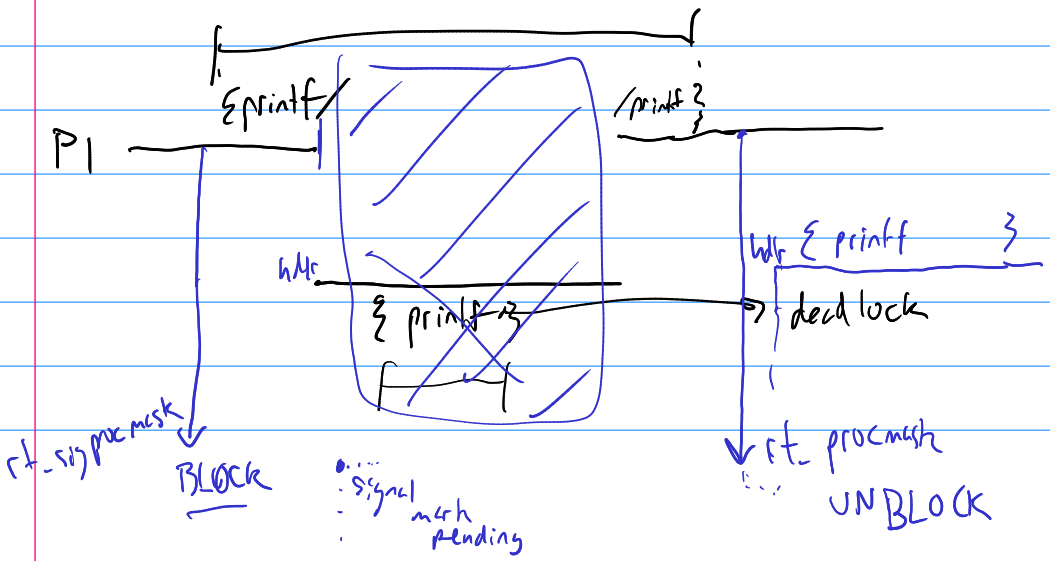


jobs



SIGCHLD between ① & ②





tradeoff: fine grained = good low latency for signal / hard to program / get right

Coarse grained: worse latency / easier to program