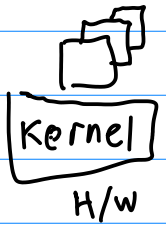
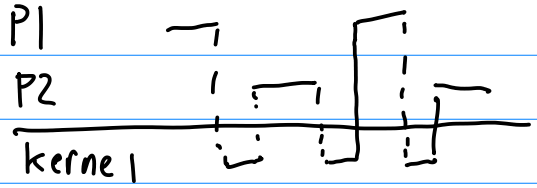
CS 3214 lecture #5 "fork + friends"

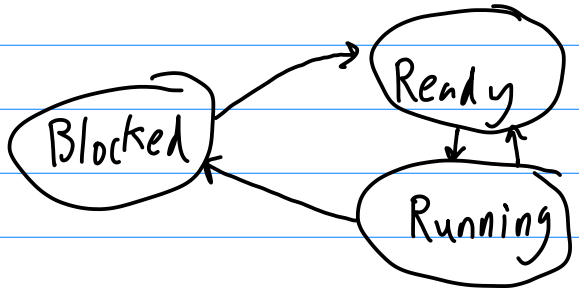
Ex 1 is out

Recap: Process : OS abstraction for running program



time-sharing
multi-programming





Programmers don't control scheduling!

→ start

```
int main() {
}
```

* NULL

Process Management

OS provides API

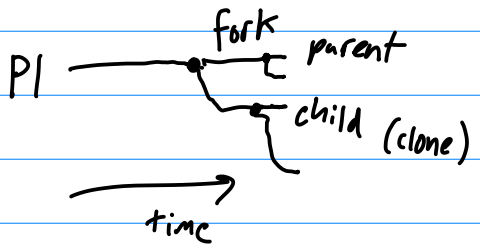
- creation (process + environment)
- termination
  - exit (normal) ↙ exit
  - crash / kill / signals (abnormal)
  - → OS cleans it up (memory / fds, etc.)
- interaction
  - waiting
  - w/ user stopping C-z / job control-
- scheduling priority (e.g., nice)
- learning about resource

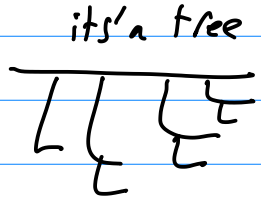
Creating new processes

1. Create process (binary, cmd line args, environment? ↙)
   "spawn"
   CreateProcessA ( )
   POSIX: spawn ( )

2. UNIX idea: separate "new process" from "new programs"

fork()



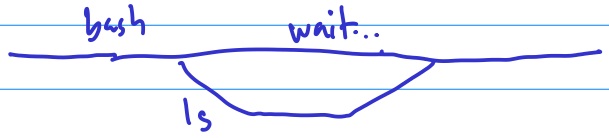


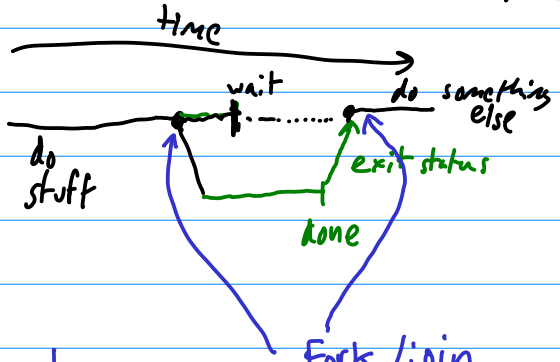
parent can have >1 child

child starts as a clone
has its own destiny

Child can have only 1 parent

wait ← parents are supposed to wait for children





don't assume anything about order!