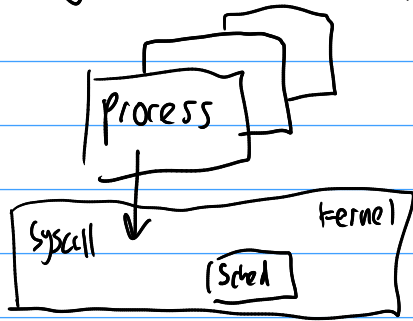


CS 3214 lecture #4 process states + control

My office hours Mon/Wed 1-2 Gilbert 4207



user

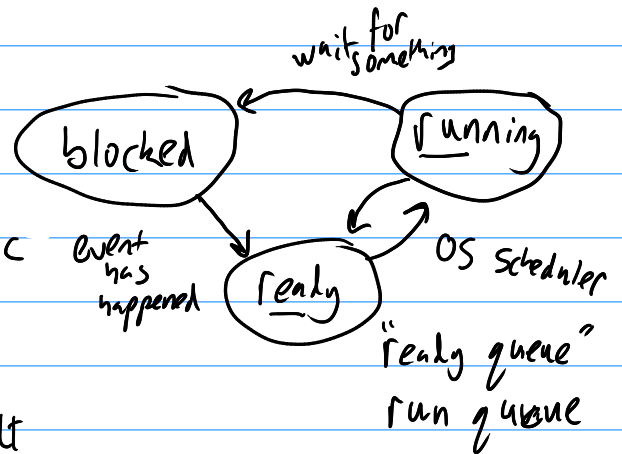
kernel

privilege : dual mode
multiple process
context switch

Next: { process states
creating new processes { fork() }
fd's, pipes, etc

Are all processes ready to run?

No: waiting on I/O



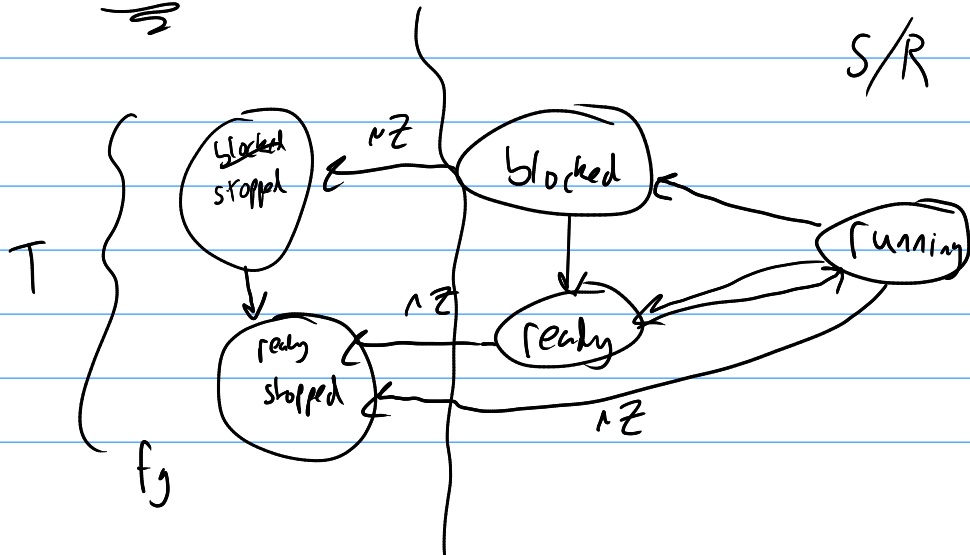
n CPUs n Ready processes \checkmark all run n sync

n CPUs 0 ready idle loop hlt

n CPUs $\geq n$ ready processes { sched pick n to run }

load avg

n CPUs $m \gg n$ ready



Final thoughts on process states

transitions out of programmer's control

- user actions, I/O, IPC, synch.
- OS sched
- block/unblock happen a lot

don't assume anything about order

Creating new ^{program} ~~process~~

- → binary to run
- → command line args.
- → environment to run in

"spawn" new program

createProcessA
posix_spawn

UNIX: idea: separate "process" creation from "new program"

fork() new process, not a new program
call once, return twice

