fork()

1 process makes call
2 processes return.

Copy of parent's state (address space, context, every thing)
Except return value of fork.

> zero ⟹ parent
zero ⟹ child
< zero ⟹ error

No gauvuntee which runs next.

Go over printf here.
Introduce variadics.

Role of wait ⟹ see state diagram

If parent dies 1st or does not wait() for child ⟹ init (pid 1) inherits the zombie.

So fork() creates a new process. New "threads" are made a different way.

To make a process run a different program, use exec()

exec() overlays user address space with a new one. State stored by kernel other than registers are preserved.

E.g. open files.

Go over exec().

Intro to man

RTFM.

Alternatives to fork()

vfork() - lighter weight than fork if all you're going to do is exec. Deprecated on mac. Why? because.

clone - lets you specify what should be copied. Not on mac. why? because.

posix_clone - limited wrapper for fork/exec. not on mac. of course.