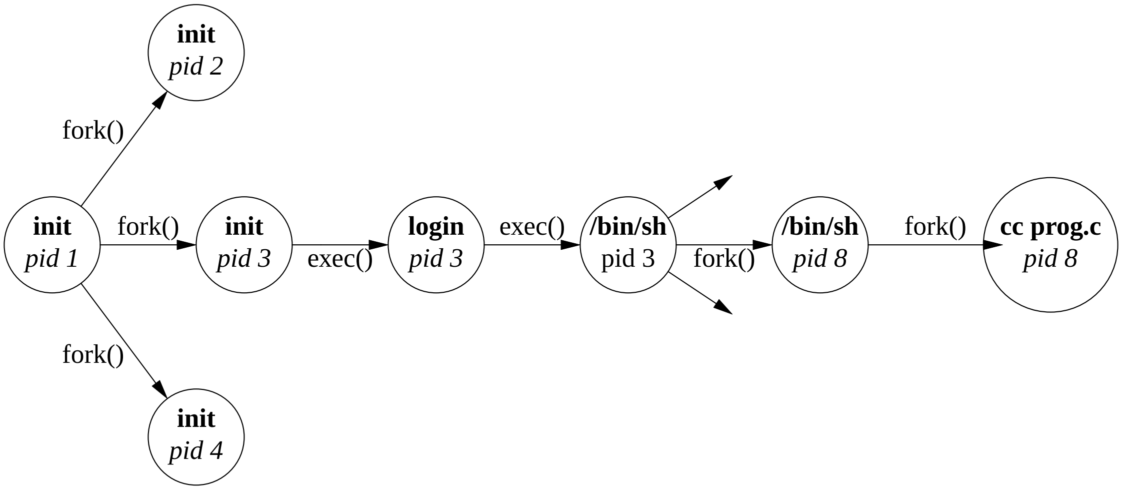
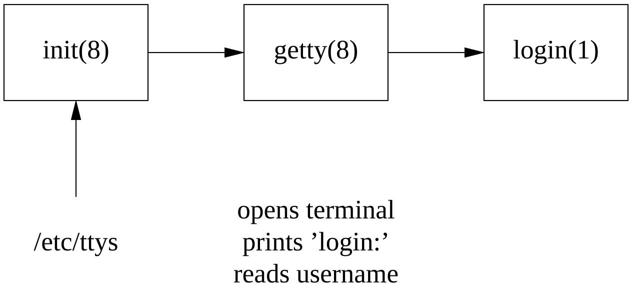
Unix Login Process

Ian Stanley

Init (PID 1) will for a number of times (at least once for each getty to be used. Each forked init then exec's to run login. Once a user enters his/her credentials the login execs to run the user's shell. The shell forks to run each command using a fork-exec combo.



The first few steps of the login process all run with superuser privileges.



login(1) runs the following:

- getpass(3), has, compare to getpwnam(3)
- if successful registers login in system databases

- reads/displays various files eg. motd
- a user may be part of a numnber of supplementary groups, so login initialises group membership: initgroups(3), setgid(2), initialize environment
- chdir(2) to new home directory
- chown(2) of the terminal device
- finally setuid(2) to user’s uid and exec(3) a shell

The kernel explicitly creates init(8).

init(8) creates getty(8) using fork() and exec().

getty(8) execs (but does not fork) login(1).

login(1) execs (but doesn’t fork) $SHELL.

And the $SHELL forks and execs commands.

init(8)	pid 1, ppid 0, euid 0
getty(8)	pid #, ppid 1, euid 0
login(1)	pid #, ppid 1, euid 0
$SHELL(1)	pid #, ppid 1, euid U
ls(1)	pid ##, ppid #, euid U

Where # is a unique PID no <>1 and ## a unique PID <> #