# Terminal-Oriented Programming

Terminal programming is a 1950s juke box.
Larry Wall

# Motives

- Cope with interview
    - how does sshd handle bash process when net disconnect (ele.me in 2018
- We are dealing with terminal everyday
    - What does docker run -it mean?
    - my iTerm2 messed up and can't print new line!
    - nohup? Cool, but what does it mean?
- We are programming with terminal
    - smc enter
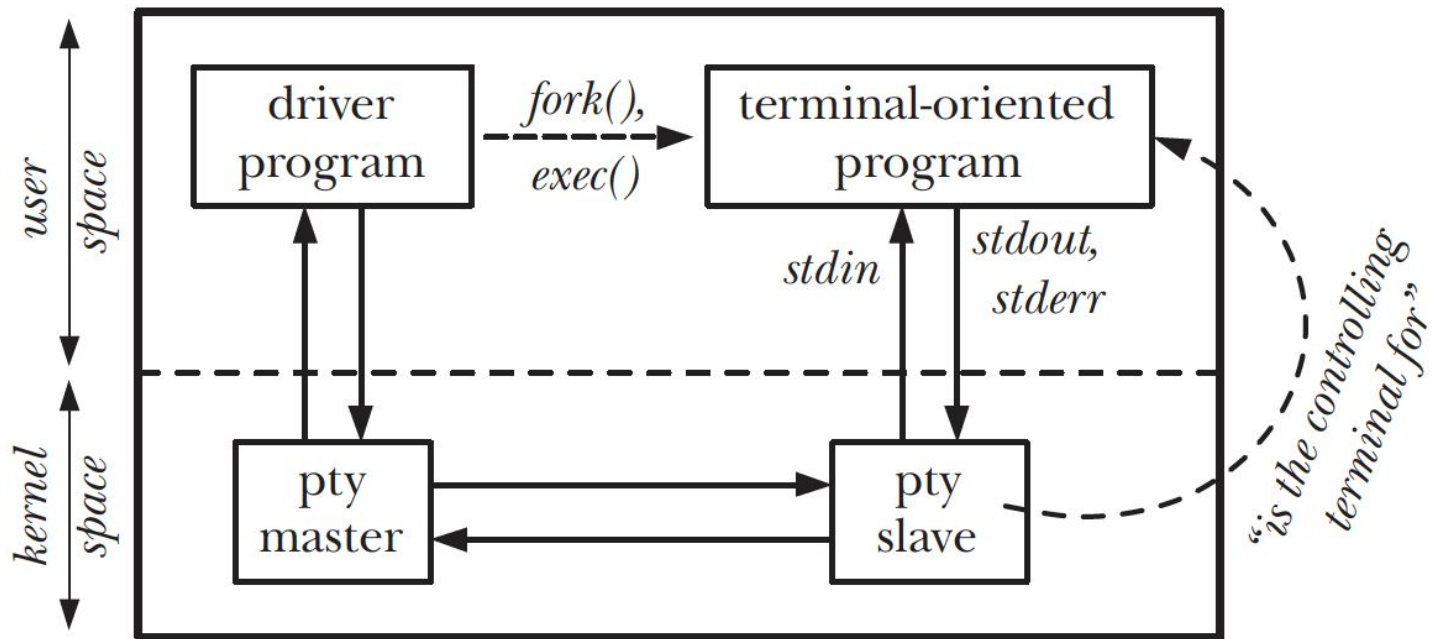    - eru-cli container exec -i
    - web tty

# Recipe

- Terminal, the fake one
    - let's make a sshd, the humble one!
- Terminal, the obsolete one
    - let's connect a container, remotely!
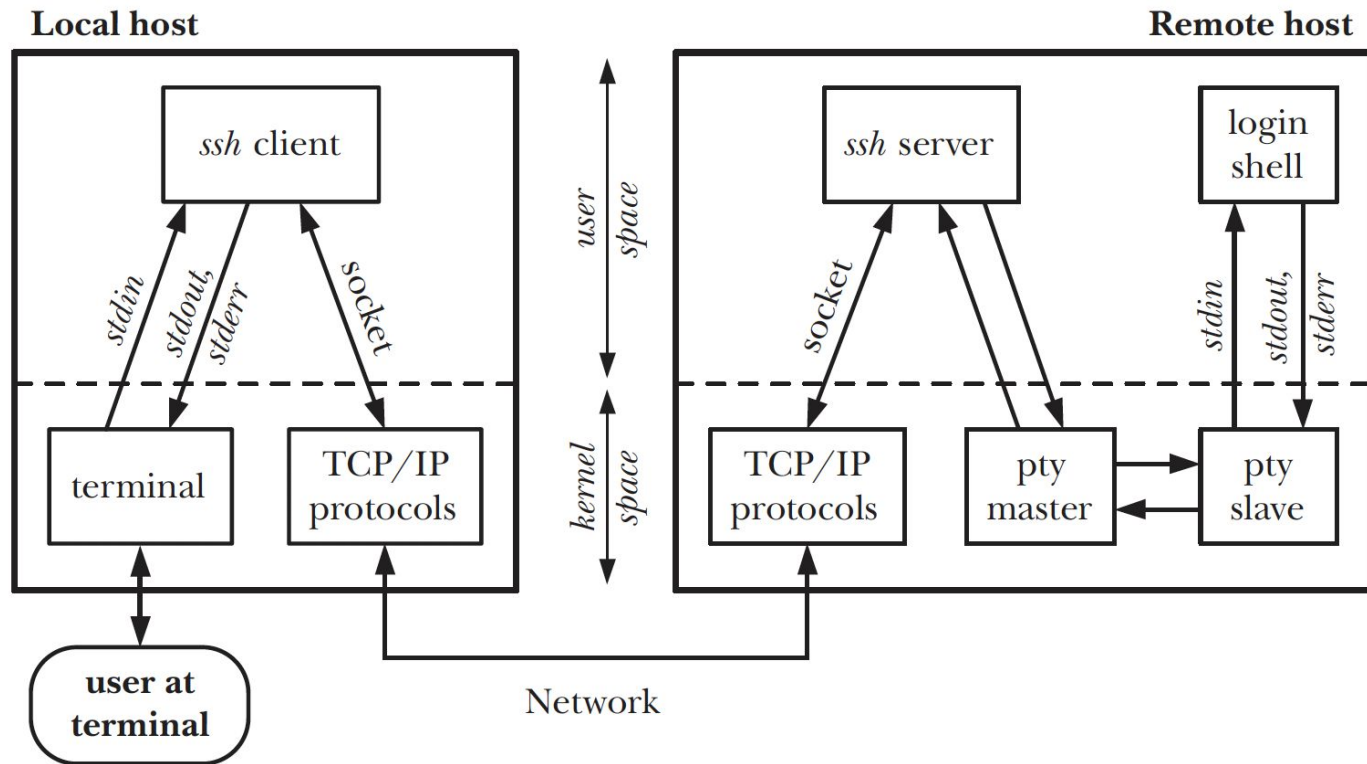- Jobs, not Steven one
    - s..signal?
- Recap

# Terminal, the fake one
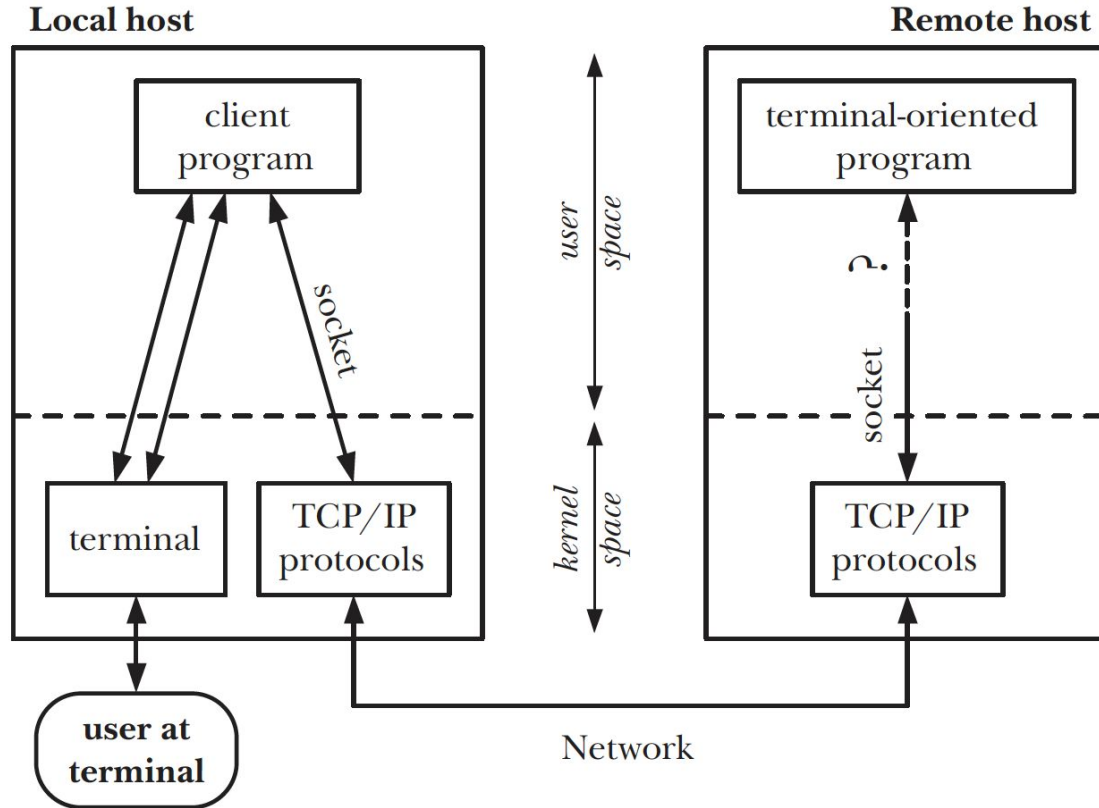
# Terminal, the fake one

# Terminal, the fake one
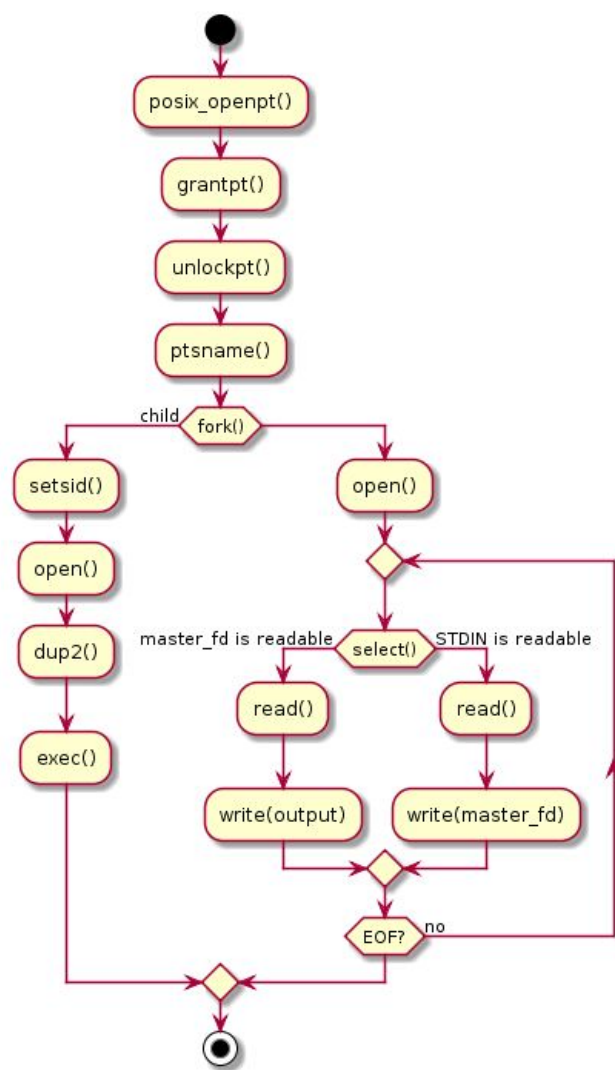
# Terminal, the fake one

syscalls(2) you need remember or not

- int posix_openpt(int flags);
- int grantpt(int mfd);
- int unlockpt(int mfd);
- char *ptsname(int mfd);

Still confused? No dread, let's make a sshd (without authentication)
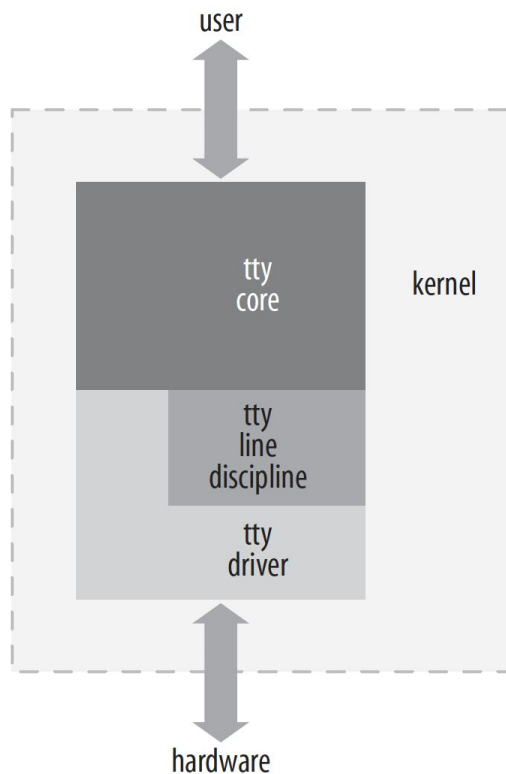
# Terminal, the fake one

SHOWTIME: sshd (

# Terminal, the obsolete one
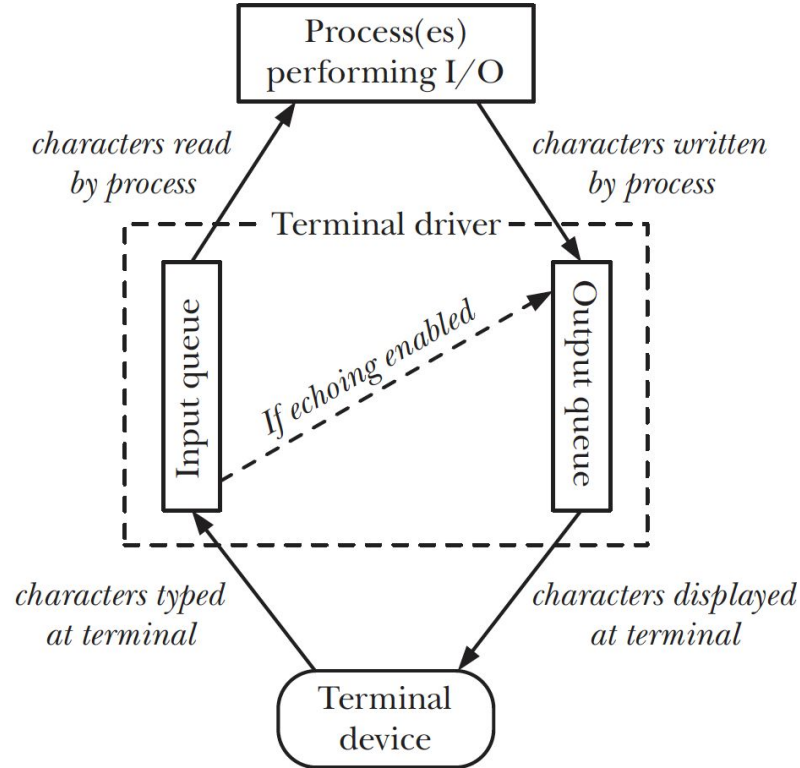


A Televideo ASCII character mode terminal, using a microprocessor, manufactured around 1982

# Terminal, the obsolete one

# Terminal, the obsolete one

# Terminal, the obsolete one

$ stty -a

speed 38400 baud; 62 rows; 118 columns;

lflags: **icanon isig** iexten **echo** echoe echok echoke -echonl echoctl

    -echoprt -altwerase -noflsh -tostop -flusho pendin -nokerninfo

    -extproc

iflags: -istrip **icrnl** -inlcr -igncr ixon -ixoff ixany imaxbel iutf8

    -ignbrk brkint -inpck -ignpar -parmrk

oflags: **opost** onlcr -oxtabs -onocr -onlret

cflags: cread cs8 -parenb -parodd hupcl -clocal -cstopb -crtscts -dsrflow

    -dtrflow -mdmbuf

cchars: discard = ^O; dsusp = ^Y; eof = ^D; eol = <undef>;

    eol2 = <undef>; erase = ^?; intr = ^C; kill = ^U; lnext = ^V;

    min = 1; quit = ^\; reprint = ^R; start = ^Q; status = ^T;

    stop = ^S; susp = ^Z; time = 0; werase = ^W;

# Terminal, the obsolete one
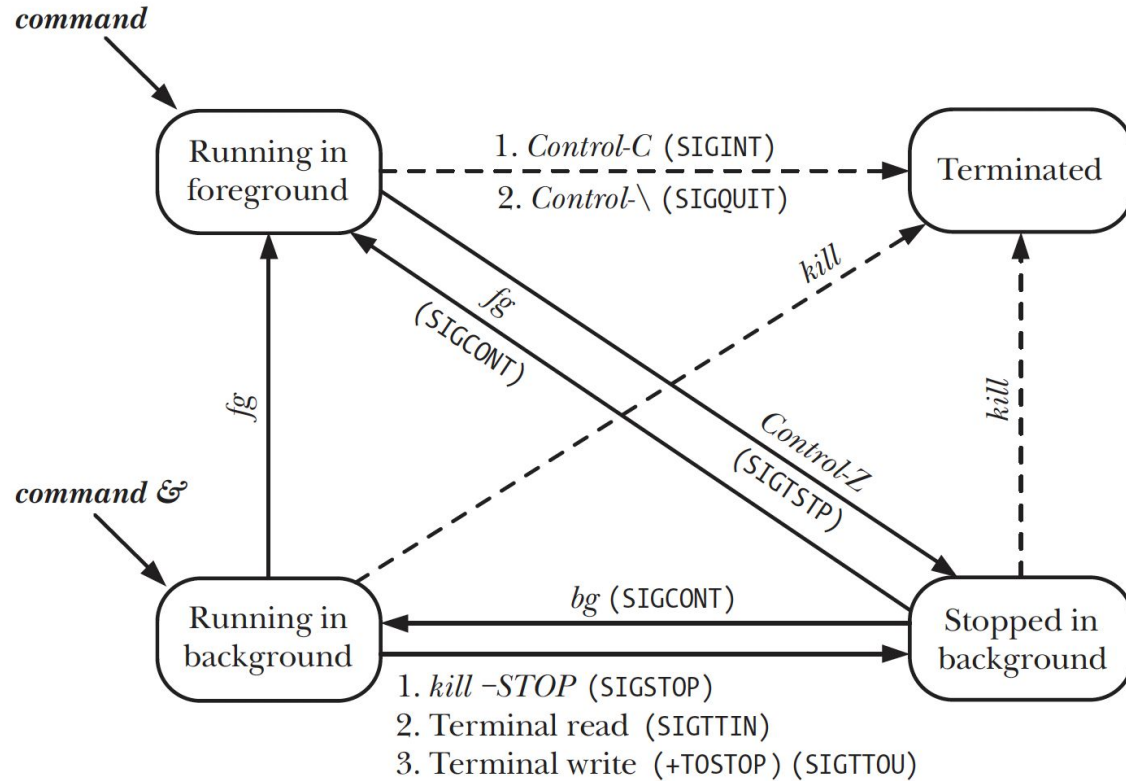
how to connect remote shell properly

- stty -echo       // off echo
- stty -icanon    // non-canonical mode
- stty -isig        // disable signals
- stty -inlcr       // disable handling CRNL
- // adjust window size

# Terminal, the obsolete one

SHOWTIME: <u>docker exec -it</u> from scratch

# Jobs, not Steven one



command

Running in foreground

1. *Control-C* (SIGINT)
2. *Control-\* (SIGQUIT)

Terminated

*kill*

*fg* (SIGCONT)

*fg*

*kill*

command &

*Control-Z* (SIGTSTP)

Running in background

*bg* (SIGCONT)

Stopped in background

1. *kill −STOP* (SIGSTOP)
2. Terminal read (SIGTTIN)
3. Terminal write (+TOSTOP) (SIGTTOU)

# Jobs, not Steven one

TERMINAL SIGNALS

- SIGHUP (1)
- SIGTSTP
- SIGTTOU
- SIGTTIN
- WINCH

# Jobs, not Steven one

SIGHUP

- When a "disconnect" is detected by the terminal driver
- When a terminal window is closed on a workstation

  then

- If the controlling process is a shell: shell sends a SIGHUP to each of the jobs that it has created.
- Else: kernel sends a SIGHUP to members of foreground process group of the terminal

# Jobs, not Steven one

SHOWTIME: use job control … but don't abuse

# Recap

- Cope with interview
    - how does sshd handle bash process when net disconnect (ele.me in 2018
- We are dealing with terminal everyday
    - What does docker run -it mean?
    - my iTerm2 messed up and can't print new line!
    - nohup? Cool, but what does it mean?
- We are programming with terminal
    - smc enter
    - eru-cli container exec -i

wubba lubba dub dub