



Tmux and Screen



Limitations of the terminal

- How do we multitask?
 - Graphical environments usually have tabs and windows, but what if we're not in one?
 - You can background tasks, but it's not intuitive
- How do we keep a session open when we log out?
- What if we want to share a terminal with someone?
 - This isn't weird if you're doing something like pair programming

Addressing these limitations

- Some applications provide their own facilities for multitasking
 - Vim and Emacs allow for split screen and tabs
 - These have limitations - what if you want to multitask with multiple applications?

- Tmux and Screen allow all kinds of flexibility for multitasking

But first, words!

- Pane: an area in a split-screen style layout
- Window: Something that contains panes and tabs
- Prefix: The key(s) you hit to tell a program you want to tell it to do something (this will make more sense in a minute)

Tmux



Tmux

“A software application that can be used to multiplex several virtual consoles, allowing a user to access multiple separate terminal sessions inside a single terminal window or remote terminal session.” *Wikipedia*

Or in English...

- Tmux lets you have multiple terminals in the same terminal
 - Multiple windows
 - Multiple panes

- You can also control tmux by using command line commands if you don't like using the prefix commands, so you can script tmux inside tmux

Getting Started

- Install tmux from your package manager
- `tmux new -s "my-session"`
- Couple extra things
 - When a capital letter is used in a Tmux command (e.g. Prefix + I), the keys you would hit are Prefix then shift + <letter>
 - Some sites will tell you to do "Prefix + <whatever>". This means hit the prefix, release it, then hit the <whatever>. Confusing since prefix is usually "ctrl+<something>"

We're in Tmux. Now what?

- `Ctrl+b` starts a Tmux command (this is your prefix)
 - You can change this to something else
- Prefix `%` to split your screen vertically (terminals side by side)
- Prefix `"` to split your screen horizontally (terminals top and bottom)
- Prefix `c` to create a new window
- Prefix `w` to switch windows
- Prefix `<arrow>` to switch to a pane in that direction
- Prefix `q <select number>` to switch to the pane with that number

Keeping sessions open

- You can “detach” from a session and close your terminal. Your Tmux and everything running in it will stay open even if you log off
 - Hit Prefix d
- Attach to a session by running ``tmux attach [session name]``
 - List sessions using ``tmux ls``
- Yes, you can use this to leave things running, like django’s test server
 - No, this is not the best way to do that

Sharing a Terminal (non-ideal way)

- Share your account credentials
 - We quickly see why this is the non-ideal way
- Have everyone run ``tmux attach [session]``
- Everyone can see and control the terminal
 - Don't do this if you don't get along
 - Tmux will shrink itself to the smallest screen size

Sharing a Terminal (the better way)

- Create a new session called “shared” with a socket named /tmp/shareds
 - `tmux -S /tmp/shareds new -s shared`
- Change the ownership of the session to a group everyone is in
 - `chgrp [group you're all in] /tmp/shareds`
- Connect! The `-r` is optional and attaches read-only
 - `tmux -S /tmp/shareds attach -t shared [-r]`

Help! I opened Tmux inside Tmux!

- Easy to do if you SSH into a server and open tmux...while in tmux on your machine. It's not the end of the world!
 - If you run tmux on your servers, consider changing your prefix

- Send commands to the inner Tmux by hitting Prefix Prefix
 - Assuming the inner Prefix is the same as the outer
 - Detach from the inner Tmux by hitting Prefix Prefix d
 - Hit prefix as many times as you need if you need to talk to your inner inner inner... tmux

Customization

- Tmux supports extensions (some are really awesome)
- Change your colours
- Change your keystrokes

Setting up Plugins using TPM

- TPM is the Tmux Plugin Manager
 - <https://github.com/tmux-plugins/tpm>
- `git clone https://github.com/tmux-plugins/tpm ~/.tmux/plugins/tpm`
- Edit `~/.tmux.conf`, add the lines below, then restart tmux
 - `set -g @plugin 'tmux-plugins/tpm'`
 - `run-shell ~/.tmux/plugins/tpm/tpm # needs to stay at the very bottom of the file`

Tmux Resurrect

- One of the coolest Tmux plugins (in my opinion)
- Allows you to save your entire tmux session (including some running programs) and reopen it
 - Persist Tmux sessions between computers (saves to a file you can drag over)
 - Persist Tmux sessions between reboots

Setting up Tmux Resurrect (TPM way)

- Edit ~/.tmux.conf, add
 - `set -g @plugin 'tmux-plugins/tmux-resurrect'`
- Restart tmux or hit Prefix I
- Usage:
 - The name of your new session needs to match the saved on
 - Prefix S saves your session
 - Prefix R restores it

Screen



Screen

“a terminal multiplexer, a software application that can be used to multiplex several virtual consoles, allowing a user to access multiple separate login sessions inside a single terminal window, or detach and reattach sessions from a terminal.” *Wikipedia*

In English...

- Does basically the same thing as Tmux. However...
 - Is not client-server like Tmux. This means you can't switch sessions as easily or move windows between sessions. Also means if tmux crashes, everyone loses but if screen crashes, only the person it crashed for loses
 - Tmux is designed to have a better command line interface
 - Tmux is more customizable in a number of ways

Getting Started

- Install screen from your package manager

- `screen -S "my-session"`

We're in Screen. Now what?

- `Ctrl+a` starts a screen command. This is your prefix.
 - This is configurable

- Prefix `S` splits the screen into panes (regions)
- Prefix `tab` Switches between panes
- Prefix `ctrl+a` Switches between current and previous panes
- Prefix `c` Create a new window
- Prefix `"` Show list of windows for switching

Keeping Sessions Open

- Prefix `d` will detach so you can log out or close your terminal. Screen will stay open
- Attach to a session using `screen -x <session name>`
 - List sessions with `screen -s`

Sharing a Terminal (the bad way)

- Log into the same user account
- Have everyone new run ``screen -x [session name]``
- Once again, everyone can see and control the session

Sharing a Terminal (the worse way)

```
sudo chmod u+s /usr/bin/screen
```

```
sudo chmod 755 /var/run/screen
```

```
# Those two commands allows screen root access without sudo
```

```
# Yes, that can be dangerous
```

```
screen -S shared-session # Start the session
```

```
Prefix :multiuser on
```

```
Prefix :acladd user2
```

```
# Those commands give user2 access to the session
```

```
screen -S you/shared-session # Have user2 run this to connect
```

Help! I opened screen inside screen!

- It can happen, no worries!

- Hit Prefix a (sends ctrl+a) to talk to the inner screen
 - Prefix a d detaches the inner screen
 - Prefix a k kills the inner screen

What if I don't like Tmux and Screen?

- You don't need to use them! They are often super useful though
- Screen and Tmux are the two most standard options, but there are others

Terminator

- Graphical alternative
 - While tmux and screen do not require a GUI, Terminator does
- Plugin support
- Key commands are somewhat different (there are multiple key combos to do things, instead of a prefix and a key)

dvtm

- Requires an extra utility (dtach) if you want the ability to detach from sessions
- Similar in functionality
- No plugins

dtach

- NOT a terminal multiplexer like the others listed, but still useful!
- Allows you to “detach” from ANY application you run with it
 - Leaving it running in the background