

Efficient Development

Motivations

1. Development speed
2. Focus on what matters
3. Better design decisions

Basic Emacs cmd line shortcuts

tab autocomplete

ctrl + left/right Move by word

ctrl + w Delete word

ctrl + k Cut Line

ctrl + a Move cursor to start

ctrl + r Reverse search

up/down Previous command

cd - Previous directory

~ Home Directory

. Current Directory

Basic Vim Shortcuts

- Suggestion: Use Ctrl-[instead of Esc
- Shortcuts f, F
- `:%s/x/y/gc`

Basic Vi Command Line Shortcuts

- ``set -o vi`` in `.bashrc/.bash_profile`
- Demo in bash

Basic Vi Readline Shortcuts

- readline: library in GNU project to edit + interact w/ command line
- `set editing-mode vi` in ~/.inputrc
- Demo w/ gdb

Why Use Emacs or Vim Keybindings?

- Emacs
 - Most applications default emacs keybindings
 - Robust development environment (git, patching, ...)
- Vim
 - Not having to learn an entire “OS”
 - An editor

TMUX

Split Vertical Pane

Split Horizontal

Scroll within pane

Create Window

Switch Window

Enable Mouse - tmux conf tinyurl.com/213quick

tmux



switching tabs



VS Code tips

fn+f12 goto definition

alt + arrow key go back to where cursor was

ctrl+enter multi window

plugins - version control, remote ssh (DISCLAIMER)

you can change the keybindings to something more comfortable for you

DISCLAIMER: We do not recommend VS code for the shark machines due to known issues with the remote plugins consuming too much disk space.

Bash scripting

basically bunch of terminal commands

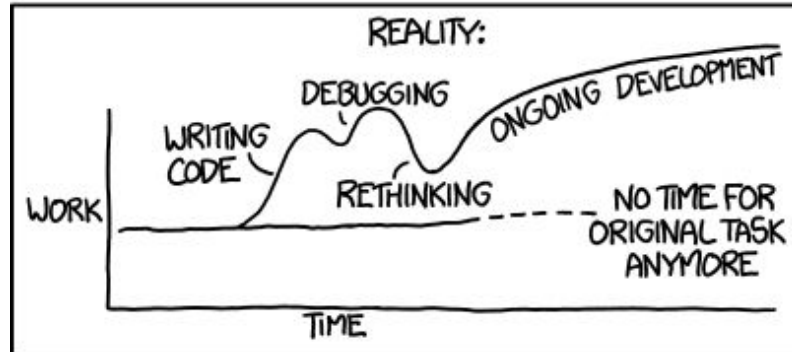
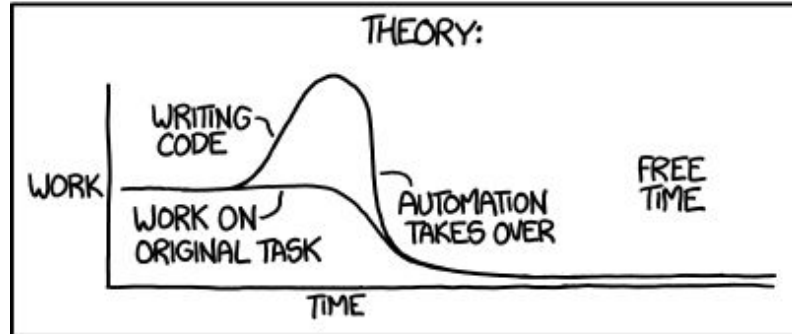
Chaining commands with `&&` , `||`

Setting up alias in `.bashrc` files

```
alias mcm="make clean && make"
```

Automating Testing/Building etc

"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"



Avoid retyping things again / Custom KeyBinds

Windows - autohotkey (autohotkey.com)

Linux - autokey

Mac - Hamperspoon

DISCLAIMER: Be careful when you try out new software. Don't run scripts from the internet without checking their contents.

Quick Screenshots

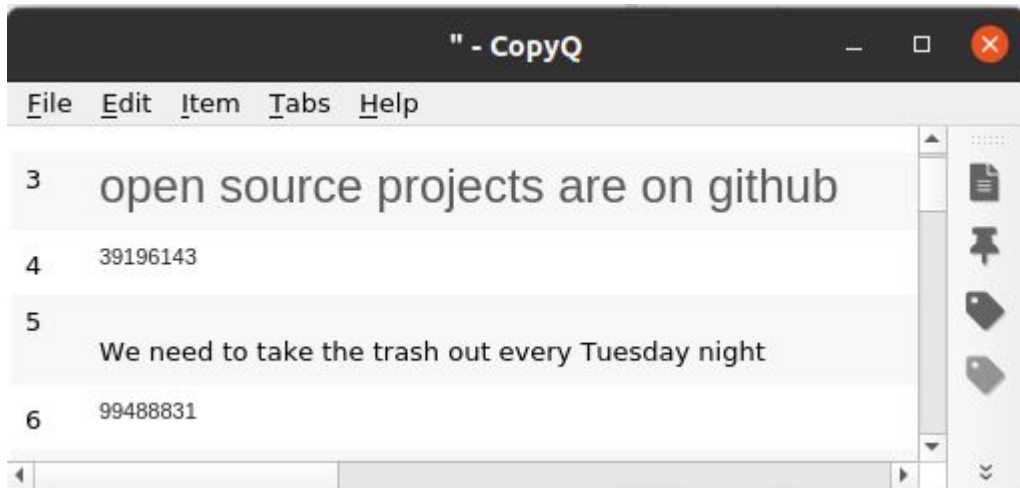
Windows + shift + S

Ctrl + Shift + Print Screen

Historical Clipboards

Windows - inbuilt (Ctrl+windows key + V)

Mac / Linux - install open source projects are on github



DON'T USE THE MAN PAGES

- Jk, you should still use them
- However, your first line of defense should be the `tldr` command
- `tldr` will give you the most used cases for 99% of commands
- So you don't have to dig through a +100,000 character man page

```
dbjasso@Davids-MacBook-Pro-5 ~ % tldr scp
```

scp

Secure copy.

Copy files between hosts using Secure Copy Protocol over SSH.
More information: <<https://man.openbsd.org/scp>>.

- Copy a local file to a remote host:

```
scp path/to/local_file remote_host:path/to/remote_file
```

- Use a specific port when connecting to the remote host:

```
scp -P port path/to/local_file remote_host:path/to/remote_file
```

- Copy a file from a remote host to a local directory:

```
scp remote_host:path/to/remote_file path/to/local_directory
```

- Recursively copy the contents of a directory from a remote host to a local directory:

```
scp -r remote_host:path/to/remote_directory path/to/local_directory
```

- Copy a file between two remote hosts transferring through the local host:

```
scp -3 host1:path/to/remote_file host2:path/to/remote_directory
```

- Use a specific username when connecting to the remote host:

```
scp path/to/local_file remote_username@remote_host:path/to/remote_directory
```

- Use a specific ssh private key for authentication with the remote host:

```
scp -i ~/.ssh/private_key local_file remote_host:/path/remote_file
```

```
dbjasso@Davids-MacBook-Pro-5 ~ % █
```


Keep your eyes open for new shortcuts!