

Terminal Customization

Aditya & David

Attendance Link

<https://tinyurl.com/terminalextration>

Choosing a terminal emulator

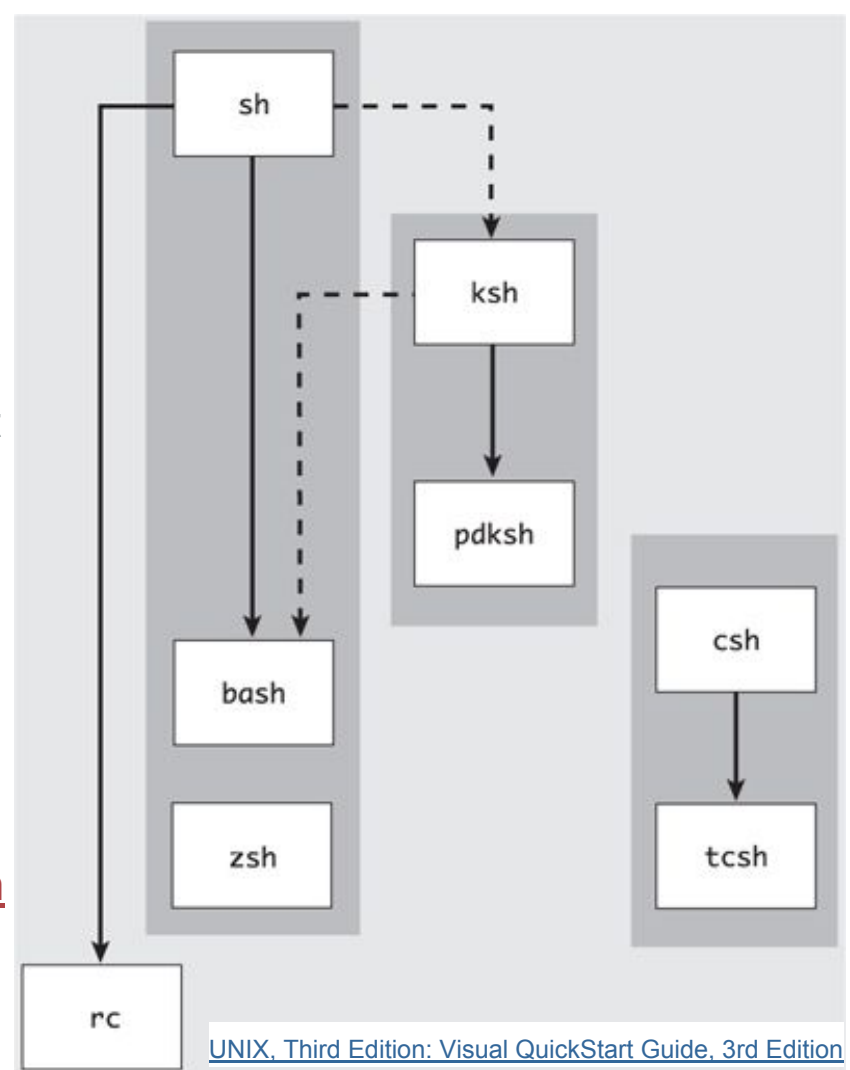
Terminal emulators

- Linux:
 - Default options GNOME Terminal, KDE Konsole, xterm
 - Low-latency options are alacritTY and ST
- MacOS:
 - Really no reason to use anything other than iTerm2
- Windows:
 - Bash on Ubuntu on Windows terminal
 - MobaXTerm
- Link to latency analysis by Dan Luu:
 - <https://danluu.com/term-latency/>

Choosing a shell

Traditional Shells

- **Always** write scripts for either bash or sh.
 - If you write scripts for bash, use a shebang because some Unixes default to sh otherwise
- For interactive use, the most relevant are **bash** and **zsh**
- Most of these shells are POSIX-compliant, with the exception of csh and tcsh
 - Avoid these at all costs
 - <http://harmful.cat-v.org/software/csh>



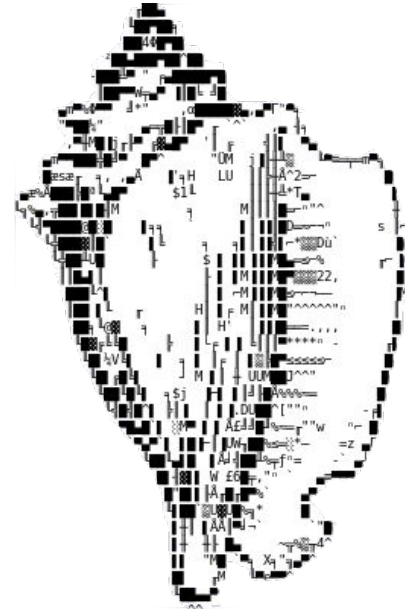
Oh my ZSH

- Starter pack for ZSH
 - Includes basically everything you could ever want
- This is the Spacemacs of terminal configuration
- <https://ohmyz.sh/>



Nontraditional shells

- Scheme shell (scsh)
 - <https://scsh.net/>
- The Friendly Interactive Shell (fish)
 - <https://fishshell.com/>
 - This is the most well-supported nontraditional shell
- XONSH
 - <https://xon.sh>
 - Python-based shell
 - My personal favorite <3
- None of these are POSIX-compliant, but all of them are awesome



Choosing a terminal multiplexer

Terminal multiplexers

- These are programs that let you split and manipulate “windows” inside of your terminal
- The two main contenders are **tmux** and **screen**
 - You should probably always use **tmux**
- **Byobu** is the starterpack for terminal multiplexers
 - Compatible with both **tmux** and **screen**
- Unfortunately, tmux and screen don’t share the same editing language as Vim for manipulating windows, so I usually just work inside of Spacemacs
 - But a lot of people swear by terminal multiplexers, so definitely check them out!

Configuring Bash

Bash dotfiles

- Bash reads from a complicated set of files, depending on whether a bash session is a **login** and/or **interactive** shell

```
/bin/bash
    The bash executable
/etc/profile
    The systemwide initialization file, executed for login shells
~/.bash_profile
    The personal initialization file, executed for login shells
~/.bashrc
    The individual per-interactive-shell startup file
~/.bash_logout
    The individual login shell cleanup file, executed when a login shell exits
~/.inputrc
    Individual readline initialization file
```

Using vi-mode in Bash

- In the ~/.inputrc file:

```
set editing-mode vi
```

```
set keymap vi-command
```

Configuring the prompt

- Bash has two main prompts: PS1 and PS2
 - PS1 is the main prompt, at the start of each command
 - PS2 is the continuation prompt, if you press enter before finishing a command
- To set the PS1 prompt, add something like the following to your bashrc:
 - `export PS1="\u@\h \W]\\$ "`
 - There is an editing language for customizing your prompt
 - Or you can use a nifty site like this to create the string for you:
 - <https://www.kirsle.net/wizards/ps1.html>