

Why the Terminal?

BISC195 – Lecture 2
Kevin Bonham, PhD
2021-06-11

Introducing Clara Kim

- tk2@wellesley.edu (she's in Zulip chat too @**Clara Kim**)
- Summer session tutor – ask her to help!
 - OK to share assignment code with her
 - She doesn't report to me

Outline

- Reasons to use the terminal
 - Terminal gotchas
 - Handy terminal tricks
 - Customizing your terminal
-
- Lab 2 – Scavenger Hunt!

Outline

- Reasons to use the terminal
- Terminal gotchas
- Handy terminal tricks
- Customizing your terminal
- Lab 2 – Scavenger Hunt!

Reasons to Use the Terminal

- Many bioinformatics tools are only usable via “Command Line Interfaces” (CLIs)
- Many biological file types are enormous and will break your GUI application
- Most “big data” or “-omics” type of analysis must be done on High-performance clusters (HPCs), most of which run headless linux
- Concepts needed for effective terminal use (eg paths) are useful / necessary in all kinds of programming
- It’s fun! (no, really)
- Once you’re over the learning curve, many common tasks are easier

Outline

- Reasons to use the terminal
- **Terminal gotchas**
- Handy terminal tricks
- Customizing your terminal
- Lab 2 – Scavenger Hunt!

Terminal gotcha #1 – no mouse

Solutions

- Use arrow keys
 - `alt + ←` and `alt + →` (Mac) or `ctrl + ←` and `ctrl + →` (Linux/Windows) can move whole words at a time
- Some useful shortcuts:
 - `ctrl + a` : jump to beginning of line
 - `ctrl + e` : jump to end of line
 - `ctrl + u` : delete from cursor to beginning of line
 - `ctrl + k` : delete from cursor to end of line

Terminal gotcha #2 – less visual feedback

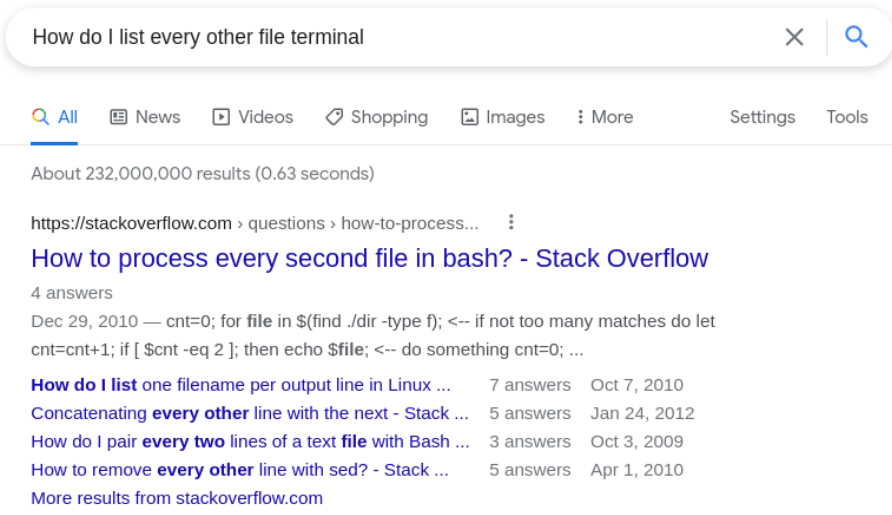
Solutions

- Use `ls`, `pwd`, `git status`, etc. frequently to keep track of where you are
- Use `TAB` for completions (especially for paths!)
- Use terminal customizations (more on that later)

Terminal gotcha #3 – no menus

Solutions

- Search “How do I list every other file in terminal”
 - Eg “how do I list every other file in terminal”
 - “How do I list every other file in terminal”
- Use explanations of the gotcha (and the answers you find by search)
- Practice!




al | shell | bash]”
sh”
,
ally helpful with

Outline

- Reasons to use the terminal
- Terminal gotchas
- **Handy terminal tricks**
- Customizing your terminal
- Lab 2 – Scavenger Hunt!

Handy terminal tricks

- TAB completion – when typing paths, TAB should be every 2nd or 3rd keystroke
- TAB completion (it's really that important)
- Use `↑` and `↓` to cycle through previous commands
 - Also `ctrl-s` and `ctrl-r` to search
 - Also `$ grep <pattern> ~/.bash_history` (or `~/.zsh_history`)
- Most of these also work for julia REPL!
 - Some subtle differences



Break!
(back at --:--)

Outline

- Reasons to use the terminal
- Terminal gotchas
- Handy terminal tricks
- Customizing your terminal
- Lab 2 – Scavenger Hunt!

Picking your shell

- “What is a shell?”
- Default is often `bash` (Bourne again shell), sometimes `zsh`, another popular shell is `fish` (friendly interactive shell)
- `chsh` (change shell) is used to change your default shell
 - Use `-l` flag to list those available
- Different shells have subtly different behaviors around completion, fwd/rev search, etc
 - Some syntax difference too, but not for basic stuff

Picking your terminal app

- There are many options for terminals that run shells
- Defaults Terminal.app (Mac), Powershell (Windows), gnome-terminal (Ubuntu)
- iTerm2 (mac) – lots of customizability
- Windows Terminal (Windows) – integration with powershell and WSL
- Alacritty (mac / linux / windows)
- There's also a terminal built into VS Code!

Picking your prompt

- Your prompt is determined by a variable called `PS1`
 - See what the current value is with `echo $PS1` (it probably won't make much sense)
 - Change the value with `export PS1='$ '` - the next prompt should just be "\$ "
- Use bashrcgenerator.com (or other web tool) to design your own prompt
 - Add colors!
- More advanced: checkout OhMyZSH, starship.rs

Make changes “permanent”

- Each shell type has a file that's run at the start of every terminal, eg `~/.bashrc`, `~/.zshrc`, `~/.bash_profile`
- Open correct file (can use VS code for this), add your changes here, then reload terminal
- Some other suggestions:
 - `alias la="ls -A"`, and `alias ll="ls -l"`
 - `alias mv="mv -iv"` (prompt before overwrite, verbose)
 - `alias rm="rm -i"` (prompt before removing, use -f to skip)
 - `alias gitlog="git log --graph --abbrev-commit --pretty=oneline"`

Better `ls`, `find`, `cat`, `grep`, etc

- `ls delux`: <https://github.com/Peltoche/lsd>
- `fd-find` <https://github.com/sharkdp/fd>
- `cat with wings` <https://github.com/sharkdp/bat>
- `ripgrep` <https://github.com/BurntSushi/ripgrep>

Outline

- Reasons to use the terminal
- Terminal gotchas
- Handy terminal tricks
- Customizing your terminal
- Lab 2 – Scavenger Hunt!