LING 331
Text Processing for Linguists

Week 2

—

More Unix, and Command-line Text Processing

Notes from Assignment 1

- Use `assignment1, assignment2` folders please!

  (and no spaces in directories or filenames)
- Don't remove the ">>>" or "`YOUR ANSWER HERE`" stuff
- Please remember to fill out the top part of the assignments!
  - (asking about hours is us trying to make sure no one falls behind or is spending their whole life on this)

Notes from Assignment 1

- Is the assignment really ~270 lines?
  - Seems maybe more?

- How does `wc` count lines?
  - Returns number of <newline> characters, aka `\n`
  - Try `-S` and `-N` flags on `less`

Notes from Assignment 1

- Dotfiles (`.bashrc`, `.nanorc`)
  - Saves us from typing e.g. `umask 002`

    every time we open a terminal
  - bash/nano/etc run these files verbatim when they open
  - Generally have to be in your home directory (`~/`)

Notes from Assignment 1

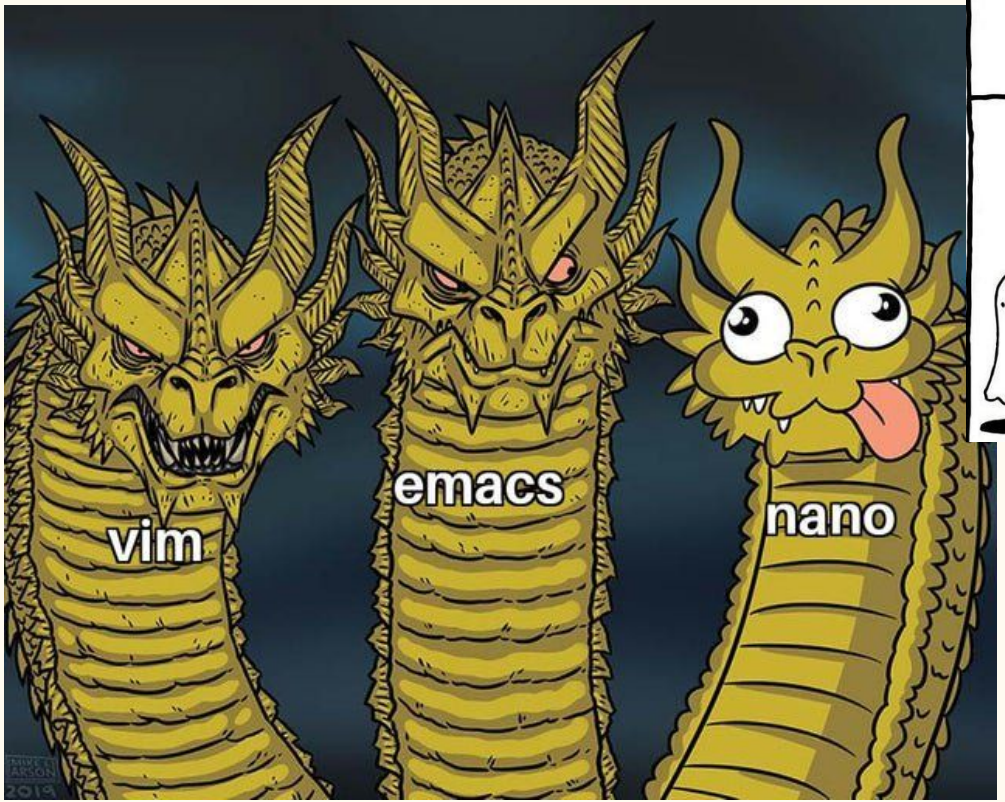
- Remember errors are friends!

- Example from class (with apologies):
  - `-bash: umash: command not found`

Notes from Assignment 1

- Tab completion! Our very good friend.
  - `cd<tab>` vs. `cd <tab>`

Notes from Assignment 1



VIM

Notes from Assignment 1!

- Is there an ‘undo’ on the terminal?

  NO! (Oh no)

- Is stuff that I `rm` gone forever?

  YES! (Oh no)

Notes from Assignment 1!

- Emacs backups:
  - `assignment1.txt` <- actual file
  - `assignment1.txt~` <- backup file
  - `#assignment1.txt#` <- auto-save file

https://emacsredux.com/blog/2013/05/09/keep-backup-and-auto-save-files-out-of-the-way/

Notes from Assignment 1!

- Nano backups:
  - `assignment1.txt` <- actual file
  - `assignment1.txt.save` <- auto-save file

  https://askubuntu.com/questions/601985/what-are-save-files

Notes from Assignment 1!

- Vim backups:
  - `assignment1.txt` <- actual file

  Uh, that's it.

Notes from Assignment 1!

- Other files you may be seeing:
  - `assignment1.txt.1` <- `wget` extra copy

Notes from Assignment 1!

Questions you may have on text editors!

- Are vim/nano/emacs just different ways to write code?
- Do vim/nano/emacs create different kinds of files?
- Why do people make such a big deal about text editors?
- Why do we need them when we can also just write commands in the terminal?

Important Concepts This Week

- Unix operates line-by-line
- You're creating a data pipeline, and ordering matters
- Imagine starting with a big piece of clay (all the lines) and bit by bit carving away and transforming it
- One liners! These techniques are very powerful for getting a quick, high-level picture of what you're working with

Unix Philosophy

- One program does one thing

  Write programs that do one thing and do it well.

  Write programs to work together.

  Write programs to handle text streams, because that is a universal interface. -Doug McIlroy (inventor of pipes)

https://homepage.cs.uri.edu/~thenry/resources/unix_art/philosophychapter.html

Abstraction

(we can trust programs and functions to do their thing)

> We don't need to worry about how something is `sorted`, we just trust that it will be properly done!

Decomposition

(big problems can be solved by breaking them into small steps)

> For each problem, think about how to decompose it into small things we can do

Notes on Assignment 2

- In a way, this is a hands-on guided tour
- We'll see many very useful programs we can chain together to do complicated and powerful things quickly
- Each program is a command, and we can trust it will do its job

abstraction!

- When we want to do something complicated, we think of how to break it down into smaller pieces we have in our toolkit

decomposition!

Notes on Assignment 2

- The assignment file itself is an executable shell script
  (a type of runnable program)

- The `.sh` ending is a suggestion of this
  (your browser may take the suggestion and auto-download)

Notes on Assignment 2 (cont.)

- **Write the command that produces the answer,**

  not the answer itself

- The scripting section (section 7) has been tricky in the past,

  read carefully and slowly there! We'll talk about it more Thurs.