2014-04-28.sagews

April 28, 2014

Contents

1 Math 480b Sage Course

1.1 The Command Line Terminal

1.2 April 28, 2014

Screencast: `http://youtu.be/AOmUj4c7H4k`

Plan

- Questions
- Homework hw4 collected and grading of hw4 has been assigned. hw5 assigned
- More on the command line terminal
- Git revision control

1.3 The Command Line: part 2

1.4 What makes the terminal so powerful

- Use tab completion (in the actual terminal) to complete file names
- You can use patterns to specify the files or directories that are arguments to commands
- You can redirect the input or output of a command.
- You can combine commands together via pipes (sort of like composing functions).

- You can temporarily pause (control-Z) and restart (fg) commands (and much more).
- There are thousands (!) of additional commands like the above, which all work in a uniform way: cal 2014, grep, du,

Illustrate some of the above points using the following commands:
ls, grep, du, find, man , sage, ipython, gp, git

1.5 Key ideas

- The man command documents every command
- Use patterns to specify filenames
- Use foo [...] — bar [...] to make the output of foo be the input to bar
- Use foo [...] output_file to redirect the output of foo to the given file, and Use foo [...] input_file to make input to foo come from the given file.
- The most important commands are: man, ls, cd, mv, cp
- When you figure out how to do something, write down the command line that does it somewhere. You can paste it in later, or tell other people about it and it is very highly likely to work forever, since UNIX is quite stable (over many decades).

1.6 Git:

"Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency."

Why learn (basics of) git:

- It is the most popular revision control system for open source development.
- It is used for Sage development, so you should know something about git to contribute to Sage. Similar remarks for many other open source projects.
- It could save your ass: its a massively cleaner way to save revisions of a document (with a comment on each revision) than just making lots of random copies all over the place. People who do serious computer-based work and dont use revision control will almost certainly spend several days of their lives recreating documents they stupidly lost. Instead, spend that time learning git.
- It facilities synchronization: Useful when collaborating with other people or when you have a program on several computers.

1.7 A Very Basic Git Tutorial

Do this all in a brand new project

- Create a new git repository: make a directory, then run git init in it.
  mkdir example cd example git init
- Create a file, add it, commit.
  echo "my awesome content" my_file git add my_file git status git commit -a -v fix issues with identity
- Change the file, look at what has changed, commit.
  echo "my super awesome content" my_file git commit -a -v
- Make changes then throw them away
  echo "so awesome" my_file git diff –color git checkout my_file cat my_file
- Checkout early version
  git checkout SHA1 HASH my_file cat my_file get checkout HEAD cat my_file
- Create a github account
  https://github.com
- Create a corresponding project on github
  ...
- Push our repo there.
  ssh-keygen open /.ssh/id_rsa.pub copy this to github git push ...
- Clone it from github to somewhere else
  cd git clone ... example-clone
- Save some changes
  cd example-clone echo "pretty cool" my_file git commit -a -v git push
- Look online and see the commits
- Pull into our original repo
  git pull