

Homework 1: bash

July 9, 2013

Due: Monday, July 15th

1 File Paths (Tuesday)

- a What is the absolute filepath of your home folder on your laptop (or someone else's)?
- b What is the absolute filepath of the home folder of espuser on Athena? Hint: you can use the `realpath` command, or you can use `cd` to navigate around until you figure it out.
- c Give some guesses as to why the path is so much longer on Athena than on a laptop.
- d What file or folder does `~/.` point to (give an absolute file path)?
- e What file or folder does `~/././././././.` point to (give an absolute file path)?
- f Challenge: find a folder where `“.”` and `“..”` point to the same folder.

2 grep (Tuesday)

Grep is a command which looks for the string you give it in a file or in all of the files in a directory. To search for the word “cat” in a file “`animals.txt`”, you can type “`grep cat animals.txt`.” To search for the word “cat” in a folder “`animals`”, you can type “`grep -r cat animals`” into a terminal.

In this exercise, we will investigate how “grep” works. You may have to set up some files and folders to run experiments on, using the commands we talked about in class.

- a Find a text file and see if there are any cats in it by using grep to search for the word “cat”.
- b Give a command to search for your favorite animal in your home directory (`~`).

- c You probably get a bunch of out put that starts with “Binary file ...”. What does that mean?
- d Does **grep** find filenames? For example, if I have a file named “**cat.txt**” which doesn’t actually have the word “cat” in it, will **grep** still find the file?
- e Does **grep** return places where the string appears, but isn’t the whole word? For example, if I search for “cat” in a file with the word “catdog” but no other mention of cats, will I get a result?
- f Can you change this default behaviour? Hint: the next exercise may help you with this.

3 man (Tuesday)

You can find the manual page for a command by typing “**man** ” and then the name of the command into a terminal. For example, “**man grep**” will tell you more about the **grep** command. You can leave a man page by typing “q”

Man pages are notoriously difficult to read. They are written in very technical language and use lots of abbreviations. You’ll probably gradually get more comfortable using man pages. In the mean time, you can use them to figure out what a command does in general (that’s usually right at the top). Some man pages also have an “examples” section near the bottom, which is particularly useful.

- a Give a command that will get you the manual for the **man** program.
- b Take a look at some manual pages for commands you already know. Look at how they’re structured. See what
- c Open the man page for “**grep**”. In the words of the man page, what does the “-r” option do?
- d Use the **grep** man page to figure out what option to use to do a case insensitive search (where capital and lowercase letters are treated the same way).

4 pipes (Wednesday)

Find a single-line command which dose each of the following. Remember to test all of your commands.

- a Writes the phrase “Hello, world!” to a file called “hello.txt.” by redirecting the output of another command (hint: we did this in class).
- b Finds the number of words in the file **/usr/share/dict/words**.
- c Finds a list of words in **/usr/share/dict/words** which have the letter “z”.

- d Finds a list of words in `/usr/share/dict/words` which has both the letter “z” and the letter “x”.

5 Setting up complex commands (Wednesday)

Find a single-line command which finds each of the following:

1. A “long” listing of the files in the `/etc` directory.
2. A “long” listing of all files in the `/etc` directory whose name contains the string “.conf”.
3. A “long” listing of all files in the `/etc` directory sorted by increasing file size.
4. A “long” listing of any five (and only five) files in the `/etc` directory.
5. A “long” listing of the smallest 5 files in the `/etc` directory.
6. A “long” listing of the smallest 5 files in the `/etc` directory whose name contains the string “.conf” sorted by increasing file size.

6 which (Wednesday, optional)

The “**which**” command tells you which executable file runs when you type a

- a Where is the binary executable file for **grep**?
- b What happens if you replace **grep** with the absolute filepath of the executable in a command? Feel free to try this with some other commands.
- c Find a command for which “which” doesn’t return anything, but which still does something (so not a command which doesn’t exist). Why do you suppose this is?

7 Challenge (Wednesday, optional)

- a Where are man pages stored? There are probably several directories on a given computer; you just need to give one here. Hint: read the man page for the **man** command.
- b Lots of people think it’s incredibly sexist that there is a **man** command but no **woman** command. There’s another command, **alias** which will allow you to “nickname” a command– give yourself a personal shortcut which lasts until you close your terminal. Use **alias** to give **woman** the same behaviour as **man**. Hint: read the man page for **alias**.