BSCI238G - Homework 2

Skylar Chan

2023-2-3

Contents

In this homework we will learn some commands that are integral to filesystems and file permissions. For each question, list and run the correct command that will accomplish the desired task. Make sure to log into Grace first. If I say "show the exact text" or similar, you can copy/paste the result or include a screenshot. This homework should be submitted on ELMS.

1 Fix the Bash prompt and the man pager

The default Bash prompt can be a bit unhelpful, because only the name of the current directory is shown. Plus last homework I mentioned that less is more than more. Let's fix these by editing the Bash startup files, which are *sourced* (activated) when running every shell. Editing this allows us to make persistent modifications to the shell. Normally we'd edit `~/.bashrc` and `~/.bash_profile`, however the Grace sysadmins have taken over those files so that we don't accidentally ruin our Bash shell. Instead we'll use `~/.bash_environment` and `~/.bashrc.mine`.

Although it's important to get used to editing in the command line, you can edit files however you like. Let's try some editors that are already installed on Grace.

Nano: Run `cp /cell_root/class/spring2023/bsci/238g/0101/public/nanorc ~/.nanorc` to copy a lightly edited Nano configuration file to your home directory (mouse, smooth scrolling, and syntax highlighting is turned on).

Then run `nano ~/.bashrc.mine` to open that file in nano. Add this line to the bottom of the file, then save and exit (you can use the mouse).

```
PS1='\[\e[1m\]\u@\h: \[\e[m\]\w $? \$ '
```

This defines the prompt, which is just a Bash variable (more on those in the future). This adds the whole path, the username, and the exit code of the last command. We're not going to discuss the prompt much more here, that's an extra credit :)

Emacs: Run `cp /cell_root/class/spring2023/bsci/238g/0101/public/.emacs ~/.emacs` to copy a basic Emacs configuration file to your home directory (mouse mode, smooth scrolling, and familiar copy/cut/paste/undo shortcuts are enabled). Tip: when looking online for Emacs help, include the version number (24) since the version on Grace is quite old. This is a common theme on bioinformatics clusters.

Then run `emacs ~/.bash_environment` and add this line to the bottom of the file. Save and exit the file, like with Ctrl-X Ctrl-C (`C-x C-c`).

```
MANPAGER=less ; export MANPAGER
```

This defines and exports an *enviroment variable*, which is a variable that is defined during your shell session. In this case it tells `man` to use `less` as its pager.

Relogin to Grace to activate your changes.

1. `cat` prints the contents of the given files (it concatenates them). Show the exact output of `cat ~/.bashrc.mine ~/.bash_environment`.

2 Explore the course directory space

Our class is located at `/cell_root/class/spring2023/bsci/238g/0101`. The course's directory structure is listed here.

2. To make it easy to access the course webspace, make a symlink to this directory in your home directory labelled `bsci238g`. Report the command you used to accomplish this. You can check that you made the link correctly: `cd ~/bsci238g` and look at the prompt (or with the `pwd` command).

3 Understand some important `ls` flags

3. Many of the `ls` flags display additional information or format the results differently. Report the flags that achieve the following:

- list all items in a directory
- list file sizes in human readable format
- list items in long format
- list items by size
- list items by modification time
- list items in reverse order

You can mix and match these flags to list items in the order you desire. And as another example of Unix command inconsistency, these flags can be combined with the same leading dash in any order. For example, `ls -ABCD` is the same as `ls -A -B -C -D` or `ls -D -CB -A`.

4 Device files

4. `head` prints the first few lines of a file. What do you get when you run `head /dev/urandom`? If you run it twice, do you get the same result? Why might that be so?

5. `tail` prints the last few lines of a file. What do you get when you run `tail /dev/null`? Why might that be so?

5 File permissions

It's important to remember some common file permission combinations.

6. Describe in words the permissions that these numbers have. For example, 777 means that user, group, and other have read, write, execute permissions.

   - 755
   - 644
   - 400

7. For each of these permissions, report a file on the system that has this permission. Hypothesize why the file might need this permission.

The `chmod` command changes file permissions.

8. Open a Bash session on your local computer, make a directory called `hw2-secret-dir`, and change its permissions to 644. What happens when you `cd` into it? Your answer will depend on how old your Bash is.

6 `mkdir` and `touch`

These commands make directories and empty files.

9. Inside this new directory create a new directory named `hw2-example-dir` and a new file named `hw2-example-file`. Report the sequence of commands that accomplishes this.

10. The `stat` command prints lots of useful information about files. Ensure that `hw2-example-file` is present, then run the following sequence of commands:

```
$ stat hw2-example-file
$ touch hw2-example-file
$ stat hw2-example-file
```

What side-effect does `touch` have besides creating files?

7 File manipulation hazards

The commands `cp`, `mv`, and `rm` copy, move, and delete files. `rm` can be very dangerous if used improperly. The famous `rm -rf`, which deletes any directory recursively, has deleted many home directories, hard drives, and careers. If you *must* use `rm -rf` you should triple check the command you've written before running it.

11. These commands will overwrite files without warning by default. What flag(s) prompt the user to confirm before performing the action?

12. `rm -r` is marginally safer than `rm -rf`. Why?

8 Practice file manipulation

Before going any further, there are some files that you'll need for the next few questions. Run this command to copy the necessary files:

```
cp -r ~/bsci238g/public/hw2 ~/hw2
```

Run `ls -R ~/hw2` to list the contents of hw2, and make sure it matches that of `ls -R ~/bsci238g/public/hw2`. If you need to reset your local copy you can simply run `rm -r ~/hw2` and recopy the contents of `hw2`.

13. `cd` into the `fruits` directory and inspect the contents. Then give a series of commands that recreates the `fruits` directory from scratch.

14. `cd` into the `links` directory and inspect the contents of the `before` directory. Then give a series of commands that creates the `links` directory from scratch. Watch out for links and funny file names. The manual will tell you handle these special file names.

15. Give a series of commands that converts `links/before` to `links/after`. Watch out for links and funny file names.

9 Try the submit command

16. **Extra Credit:** Hooray, you're at the end! For extra credit, use the `submit` command on Grace to submit your answers through the command line.