

Practice Problems For Topic 1: Linux Basics

CISC 220, Fall 2016

M. Lamb



This is your first set of practice problems. They will not be collected and marked, but they are still *very important*. You should be working on them as soon as we cover the “Linux Basics” topic in lecture. If you wait until the first assignment is looming (or, even worse, the first quiz) you will not have time to practice and absorb everything you need to. Most of you will have questions as you go along; please feel free to bring these to lectures or office hours.

For those of you who have ever taken piano exercises (or studied another instrument), think of these problems as finger exercises. They are fairly simple and you could fall into thinking of them as trivial or unimportant. But if you don't do your finger exercises you will not build up the muscles you need to play interesting piano pieces -- or in this course, building interesting scripts and programs.

After this assignment I will assume you are comfortable using Linux and the basic elements of the bash shell. If you don't bother to spend time on it and get help right away if you need it, you may find yourself tripping up over this stuff later, at the same time as you're trying to learn more advanced material.

If you've used Linux before, you may be tempted to skip these problems. Please don't! You may be able to get them done very quickly, but you may find there are some details that are new to you. Take the time to read through each problem and practice it if it's not something you're absolutely sure you can always do easily. Better to practice them now than to have to do it later when the term gets busier. Going back to the piano analogy, finger exercises need to be repeated, especially when you've been away from the keyboard for a while.

Warning: At the start of every academic year, our CASLab staff has to set up the computers and make sure that all students in Computing courses (2nd year and up) have working accounts. This takes time and not all of it can be done before classes start. I will be communicating with the staff and making announcements in lectures and on the onQ general forum to let you know when the labs are ready for you to use. You can always try before that, especially if you already have used the CASLab computers, but if things seem broken please be patient and wait until we tell you that things are set up and ready for you before you complain. Thanks!

Choice of Shell: Please do all of these practice problems using the bash shell. There are GUI programs in Linux that have some of the functionality you need, but it's important that you become comfortable with the shell and shell commands for future use and in scripts. And there are other shells that could be useful, but in this course you will be required to use the bash shell for assignments and quiz problems.

While working on these problems, refer to the Linux Basics summary sheet.

Sample Files: For these practice problems, it's helpful to have some sample files to play with. The directory `/cas/course/cisc220` has several files you may use, including the poem files we will use in class as examples. You don't have permission to change anything in that directory, but you may copy some or all of the files into a directory of your own.

Part 1: Account & Lab Access. The first thing you need to do is to make sure that you can get into the CASLab lab rooms containing the Linux/Windows dual boot machines and that your CASLab account is working. And yes, this is important even though this course doesn't have formal lab sessions and even if you have access to Linux elsewhere. Here are the reasons:

1. If your own Linux machine breaks down at a time when you need to be finishing an assignment or studying for a quiz or exam, it's not an excuse. We have Linux machines available for your use on campus where you can finish your work.
2. We may have help sessions in the CASLab rooms at times during the semester and so you need to be able to get into those rooms and use the machines there.
3. Versions of Linux can differ from one another, even two installations of the same version of Xubuntu (because of all the options the administrators can set). The Linux system on the CASLab computers is the standard of correctness for this course. So even if you prefer to do most of your work elsewhere you should be checking it on CASLab.

1A: Make sure your CASLab account is working. If you've taken a Computing course at Queen's recently, you're probably OK, but it's best to make sure early in the term. If you can log into a machine in one of the labs, you're OK. If not, visit the enrollment station in Goodwin 248 to activate your CASLab account. (Or you can do it from home at <http://courses.caslab.queensu.ca/>.) |

After you're done with the activation dialog, it may be up to 10 minutes before you can log into your account. Please don't complain to me or the CASLab staff that your account isn't working unless:

- You have been registered in CISC 220 for at least 2 full days
- You have activated (or re-activated your account)
- You have tried logging 10 minutes afterwards

1B: Get an iButton. If you don't already have an iButton, buy one at the bookstore. Don't count on banging on the lab doors and asking other students to let you in; it's disruptive and there may be times when there's nobody there to help. Follow the directions on the course web page under "CASLab" to enroll your iButton.

1C: Startup. Boot up a machine in Linux mode and log on.

Make sure you're using bash as your default shell. This should be the case unless your account is very old or you have changed the shell yourself. To check, type

```
echo $SHELL
```

and the output should be `/bin/bash`. If not, you can change it yourself by running `caslabUtils.sh very carefully -- set your shell to /bin/bash, not just bash`. Or you can call `bash` as a program from your login; just type **bash** and you'll be switched to bash. Try echoing `$SHELL` again just to make sure.

Part 2: Bash Command Practice. Go through the Linux Basics summary sheet on the web site and try out each command and all of the options shown for each command. Practice creating, deleting, copying and moving files and directories until you feel comfortable with all the commands. You don't need me to spell out practice exercises to do; just try out each command until you feel like you can use it without a lot of thought.

As an example, the summary sheet tells you that the `mkdir` command creates a new directory. So try creating a new directory as a sub-directory of your home directory. Practice other directory-related commands by moving in and out of the directory, creating new files in the directory, and moving or copying existing files into the directory. Finally, delete the directory when you're done.

It's important to practice each command; don't just read it and think "yeah, I get it". You will remember the command much more easily if you've actually used them, and practicing will show up any misunderstandings.

Part 3: File Transfers. Make sure you've got a good way to transfer files to and from the Linux systems. Try at least one of the following:

- move or copy files back and forth between your Linux folders and a flash drive
- use an ftp program to transfer files between CASLab and your own computer (most easily done from your own computer)
- use a web-based mail application to send mail to yourself with files attached
- use a browser to move files to and from a cloud location such as Dropbox or Google Drive

Even if you plan to do all of your course work in the labs, it's very important to know how to do one of the above in order to make backups of critical files. **Reminder: it is your responsibility to keep regular backups of all of your course work, especially marked assignments. I will not give extensions if you lose your work because you neglected to do this.**

Part 4: Text Editors. Please make sure you're comfortable with at least one Linux text editor. I recommend taking at least half an hour to learn the basics of vim or emacs. The time you take to do this will pay off over the course of the semester. But if you're really struggling with Linux right now and would rather start with an easier editor, I suggest nano.

You only need to know how to use one editor and you don't need to become an expert in all the commands in your chosen editor. The emacs and vim editors have lots and lots of features and it's not necessary to learn them all right away. The summary sheet for this topic tells you how to get to the tutorial for each of these editors. You do *not* have to go through the whole tutorial for either editor! Just get far enough that you know how to do at least the following:

- create new files, open existing files, and save modified files
- move around in a file, by character, line or page
- add characters to a file
- delete characters from a file

Practice until you are comfortable doing these things without having to refer to notes or tutorials. The following weeks will go much more smoothly if you aren't struggling with the mechanics of your text editor. Once you're comfortable with the basics, you might wonder how to do more advanced things, such as finding and replacing strings. Then you can go back to the tutorial or search the web for answers.

Once you're used to working with a small file in an editor, try a large file as well to make sure you know how to move through it. If you've got a large text file handy, you can experiment with that, or you can make a copy of `/cas/course/cisc220/pride.txt`.

Part 5: File Permissions. Use `ls -l` to look at the protections on the files in your current directory. Use the `chmod` command to add and remove permissions from some of them and do another `ls -l` to see the results. Try removing your own read permission to a file and then try to display it with `cat`. Try removing write permission from a different file and then try to edit it.

Create a subdirectory of your home directory if you don't already have one. Experiment with changing its protections. See what happens if you remove your own read, write or execute permissions on that subdirectory. Can you still use `ls` to see the contents of your directory? Can you use `cat` to display a file in the directory? Can you create a new file in the directory? Can you delete a file from the directory?

Use `umask` to find out what your current protection mask is. Create a new file and predict what its permissions will be. List it in long mode to see if you are right.

I highly recommend using a `umask` that doesn't allow anyone but you to see the files you create. It's safer to let all your new files start out as private, and then change the ones you want to share. If your `umask` allows others to see your new files, please fix it. A change you make with `umask` will be in effect until the end of your current Linux session. Soon we'll talk about how to put commands like this into your initialization files so that they will be executed every time you log in.

You absolutely must make sure that you use a secure directory for your marked lab assignments. Leaving your solutions easily visible to be copied by other students is a violation of Queen's Academic Integrity. If you work in a group, share code via e-mail instead.

Part 6: Try Linux at Home (optional). If you want to work on CISC 220 outside of the lab locations, try that out now and get the details ironed out before you have more work to do. Some options are:

- Using `putty` or `ssh` to connect to one of the CASLab computers
- install Linux on a computer of your own
- create a virtual machine running Linux on your own computer using Virtual Box or Vagrant

The course web page will have information about how to do these things. Remember that your marked assignments have to work on CASLab's version of Linux and that quiz and exam answers have to be appropriate for that version as well. So get a copy of `putty` or be prepared to visit a lab location to check your work against the CASLab standard.