Intro to your csci linux account

- Dave Wessels, CSCI prof at VIU (David.Wessels@viu.ca)
- Quick intro to accessing your computer science linux accounts using ssh (from linux, mac, or win10)
- Will run through basic process to connect, change your password, explore/play with your account a bit, and logout

Accounts

- You’ll be assigned a linux username and password for your account (this is different than your general VIU account)
- You’ll have to change the password the first time you login
- The username is generally some combination of (part of) your last name, initials, and possibly a digit, e.g. for J. D. Smith the username might be smithjd1
- Instructors can look up your username, but you’ll need the csci tech to reset your password if you ever forget it (tech@csci.viu.ca)
- I’ll be using a fake account name, davestu, for the examples here

Why/when you’ll need it

- Most of your computer science labs, assignments, and projects will be obtained/submitted through your linux account, usually starting in the second week of semester
- You can sign on to your account from machines in our labs (rooms 102, 115 in building 315), or by an ssh connection from your laptop, pc, tablet, or phone
- You don’t need a high end laptop for this, ssh is pretty minimalist, and (as long as you have a network connection) the actual files/software you’re using will be on our server (you’re just using your laptop as a way to connect)

The servers

- If you’re connecting from your own device, you’ll usually connect to otter.csci.viu.ca (otter hereafter), or csci.viu.ca (which in fact also goes to otter)
- If you’re physically in our labs, you’ll be on one of the servers named pup1.csci.viu.ca through pup18 or cub1.csci.viu.ca through cub18, but these are (more or less) mirrors of otter so your files/software will look the same

Valid use

- When you’re given your account credentials, you’ll also be given a user agreement covering what is/is not valid use of your csci account – please do read this carefully, as you’ll be responsible if you misuse your account or allow someone else to

Obtaining ssh

- If you’re connecting from a mac or linux laptop/pc then you already have ssh
- If you’re connecting from a win10 laptop/pc then you can install and run Ubuntu (free from windows store), which has ssh
- If you’re connecting from an older windows device then you can find/install programs like PuTTy, which allow similar connections
- If you’re connecting from a phone/tablet then there are a variety of decent free apps like Termux that support ssh

Using ssh to connect

- First we’ll open a terminal/command window:
  - On linux you’re probably used to this already :)
  - On win10 run ubuntu
  - On mac you’ll find ssh in /applications/utilities/
- Why a terminal/command interface not a GUI?
  - later you’ll be writing software that issues linux commands, you need to be fluent with using them before we get there
  - perk is it uses way less bandwidth than GUI

Connecting (finally!)

- To connect to our server, our ssh command needs to identify the server name and the username we’ll connect as (here using davestu as the username)

  ```
  ssh -l davestu csci.viu.ca
  ```

  Note that -l is a lowercase L, not a 1 or an i

- It will prompt you for your password, type it in and hit enter (it won’t appear to do **anything** while you’re typing, keep going anyway)
- Your first time logging in, it will likely make you change your password, have something secure and memorable in mind

Your starting point

- Whenever you login, you start in your “home” directory, the base location for all your files and directories
- All users have their own home directories someplace on the server, we’ll explore where that is a little later
- When you login, you’ll be given a linux prompt, while it sits and waits for you to type a command, e.g.

```
davestu@otter:~$
```

- When you type a command, if nothing goes wrong it will just give you another prompt, if something goes wrong it’ll give an error message and a prompt (no news is good news!)

What’s in your account

- Where-ever you are in linux, the command to list the files and directories in that location is simply

```
ls
```

- At the moment, that might show nothing (you haven’t created any files or directories yet), but there are actually some hidden system files there as well, to see these use

```
ls -a
```

Let’s create some content

- To create a new file, we can use the **pico** editor, type

  pico somefilename

- Type in some content, and when you’re done use control-O to save (it’ll double-check the name of the file you want to save as, just hit enter), then control-x to exit
- Note that pico gives you a list of commands at the bottom of the screen, the ^ means hold the control key (we’ll talk about better editors another day)
- Now that you’re back at the linux prompt, use **ls** again and it hopefully lists the file you just created

How about a new directory

- You’ll often want to organize your files into directories (like folders if you come from a Windows background), e.g. one for each course, maybe one for your own experiments, etc
- To create a directory use **mkdir** and the name for your new directory, e.g. **mkdir myNewDir**
- Now **ls** should show your new directory too
- You can also remove (**rm**) empty directories or rename (**mv**) directories, e.g.

```
rm somedirectory

mv oldname newname
```

Exploring directories

- The command to change from one directory to another is cd (for change directory), with a variety of options

  cd dirname (looks inside your current directory for dirname and goes there)

  cd .. (goes “up” one level in the directory tree)

  cd (by itself cd takes you back to your home directory)

- To see exactly where you are in the file system use the **pwd** command (print working directory), e.g. in your home directory it may show something like /home/student/davestu

Filenames and extensions

- Most of the files you edit will be plain text files, but you’ll often add a file extension to the name to indicate the purpose of the file (e.g. .cpp for C++ code, .c for C code, .java for Java, .sh for shell scripts, etc)
- Linux doesn’t actually require file extensions, so don’t be surprised when you see files without one

Logging out

- Use the command **logout** to end your session
- The next time you log back in you’ll start in your home directory again, but all the files and directories you created or modified last time will still be there
- Feel free to explore and create files and directories in your own space, and to research other useful linux commands and software
- Your account does have limits on both disk space and cpu utilization, but feel free to experiment ... at least within the bounds of the user agreement :)
- In later sessions we’ll explore more of the tools, commands, and techniques we’ll be using for our courses