Getting Started With Unix and vim

A. First, get connected!

Connection Using Windows

- Use an SSH client, such as MS Secure Shell Client or Putty and click for a new connection
- If you are ***off campus***, the host name should be `access1.cs.clemson.edu` or `access2.cs.clemson.edu`
    - Login in using your user name and password
    - Once you are connected to `access1` or `access2` you need to SSH to one of the machines in 110 Lab using the command `ssh <machine name>` for example: `ssh imp6.cs.clemson.edu` and then you'll have to enter your password again
- If you are ***on campus***, you can log directly into one of the named machines bypassing the `access1` or `access2` gateway, so the host name would be `machine_name.cs.clemson.edu`
    - Login in using your user name and password

Connection Using Mac or Linux

- Open a shell (Terminal)
- If you are ***off campus***, type the following: `ssh user_name@access1.cs.clemson.edu` or `ssh user_name@access2.cs.clemson.edu`
    - Once you are connected to `access1` or `access2` you need to SSH to one of the machines in 110 Lab using the command `ssh <machine name>` for example: `ssh imp6.cs.clemson.edu` and then you'll have to enter your password again
- If you are ***on campus***, type: `ssh user_name@machine_name.cs.clemson.edu`

Machine Names

- The machines in the department that you have access to have names – they are named imp, joey, or koala. There are 22 "imp" machines (imp1, imp2, imp3, … imp22). The joeys are joey1, joey2, … joey27. And the koala are koala1, … koala24. So pick any one of those when you are logging into a machine. If the one you pick doesn't work, try another one.

NOTE: Don't forget to exit using the **logout** or **exit** command from your shell

B. Ok, you're connected… Now What?

If you are logging in remotely on your laptops, you have to use SSH (or Terminal on a Mac). Once you log in, you will be using a command-line interface, so you'll need to use the **Unix commands (table on next page)**. The first lab usually has you do a few things using the Unix commands to get you acquainted with the system, setting up your directory where you will likely do your lab work, and possibly creating your first program.

Once you start to become familiar with the Unix commands, you can move on to creating a file. To create and edit files, you would use an editor, such as vi (or vim, which is the new improved vi), pico, emacs, etc. I have always used vi, and the **table on page 3 is a vi cheat-sheet**. I will show you in class what to do with that. If you're brave and adventurous, you can try to use the vi cheat-sheet and see how much you can figure out on your own!

Useful Unix Commands

Command	**Purpose**
`ls`	list all contents in your current directory
`cd`	takes you back to your home directory
`cd ..`	takes you back one directory
`cd dir_path`	takes you to the directory specified by the path provided
`mv src_file dest_file`	renames `src_file` to `dest_file`
`mv src_file dest_path`	moves `src_file` to the folder specified by the `dest_path`
`mv src_file dest_dir/dest_file`	moves `src_file` to the folder specified by `dest_path` and gives it the name `dest_file`
`cp src_file dest_file`	copies `src_file` to `dest_file`
`cp src_file dest_path`	copies `src_file` to the folder specified by `dest_path`
`cp src_file dest_dir/dest_file`	copies `src_file` to the folder specified by `dest_path` and gives it the name `dest_file`
`rm file_name`	deletes the file named `file_name`
`mkdir dir_name`	creates a directory name `dir_name` in your current directory
`rmdir dir_name`	deletes the directory named `dir_name` if it is empty
`rm -rf dir_name`	deletes the directory named `dir_name`
`ps`	shows a listing of processes that are running
`kill -9 [pid]`	kills the process you specify with the pid (process id #) which is shown when you type ps

Useful vim Commands

Command	**Purpose**
`vim file_name`	creates a new file with name `file_name` or opens `file_name` if it already exists
Control Commands	**Purpose**
`i`	puts you in insertion mode where you can start typing your text; to go back to control mode, hit the escape key
`a`	puts you in insertion mode to start typing immediately after the character where you currently were at
`x`	deletes the current character
`r`	replaces current character with the next character that you type
`u`	undo – undoes the last command
`dd`	deletes the line you are currently at (and puts a copy of it in the clipboard)
`/<word_or_letter(s)>`	to look for or move directly to a word or string of letters, type a forward slash and the word or string of letters that you want to go to
`J`	appends the next lower line up to the current line
`<number>dd`	deletes the number of lines specified starting on the line you are currently at (and puts a copy of it in the clipboard)
`yy` or `<shift>y`	copies the line you are currently at and puts it in the clipboard
`p`	pastes what is copied on the next line beneath where you are
`j`	navigates down one line
`<number>j`	navigates down the specified number of lines
`k`	navigates up one line
`<number>k`	navigates up the specified number of lines
`l`	navigates one character to the right (or `<number>l`)
`h`	navigates one character to the left (or `<number>h`)
`<shift>h`	brings you to the top of the screen
`<shift>l`	brings you to the bottom of the screen
`:1`	brings you to the first line in the program
`:<number>`	brings you to that specified numbered line in the program
`<ctrl>f`	advances you an entire page down
`<ctrl>b`	brings you back up an entire page at a time
`<shift>4`	brings you to the end of the current line
`cw`	to change a word to a different word
`%` on top of a parenthesis	to find the matching parenthesis
`<shift> d`	to erase the rest of the line starting from where the cursor is located
`:q`	quit the file without saving
`:q!`	force the quit file without saving
`:w`	save your work to the file but don't quit Vim
`:wq`	save your work to the file and quit Vim
`:w file_name`	save a copy of the current file to another file specified by the `file_name` but don't quit Vim (stay in the current file)