COMP-206 Assignment 1 Tutorial

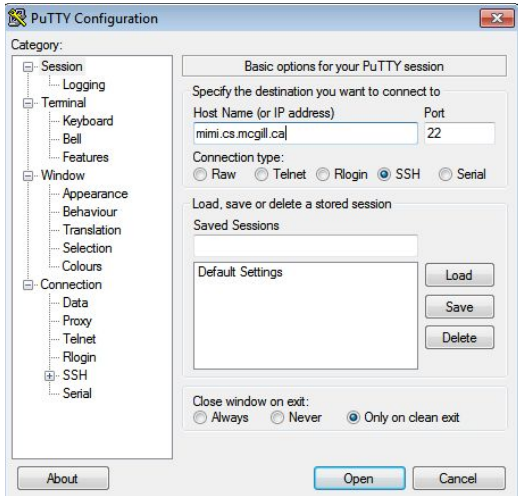
Friday, September 14th, 2018

Important first steps for a smooth COMP-206

- Make a CS account from McGill’s School of Computer Science
    - https://newuser.cs.mcgill.ca/
- All your assignment code should be able to compile and run on these servers
- If you run a Windows computer (laptop or at home), Install the following programs:
    - PuTTY: https://www.putty.org/ - lets you access the Trottier computers using command line
    - WinSCP: https://winscp.net/eng/download.php - program that lets you click and drag files between Trottier and home computer
- Try out the Trottier computers

Login Information

- The Trottier computers can be accessed using various addresses, such as **mimi.cs.mcgill.ca**
- On Windows - use PuTTY or WinSCP
- On Mac or Linux - open terminal and type "ssh username@mimi.cs.mcgill.ca"
- You will be prompted for user/password
- On Trottier computers - simply type your username and password



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): mimi.cs.mcgill.ca

Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

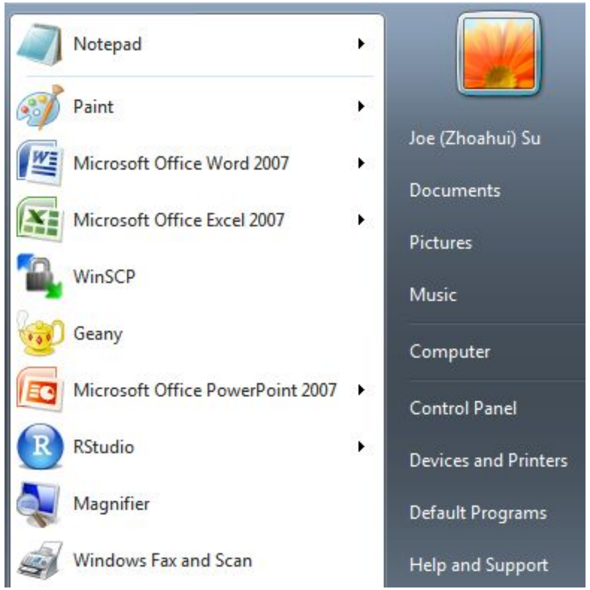
Close window on exit: Always Never [✓ Only on clean exit]

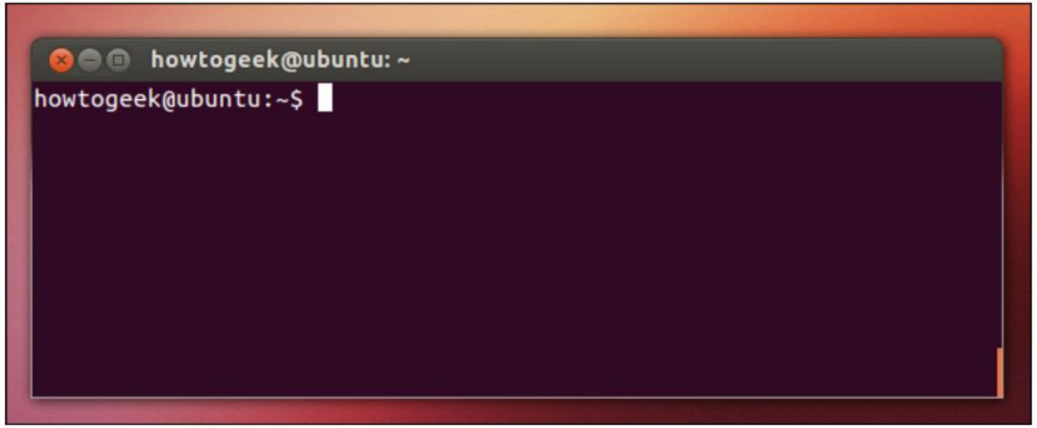
Moving files between McGill and Home

- Mac using your own terminal
  - scp my_file zsu3@mimi.cs.mcgill.ca:/home/2013/zsu3/COMP206
  - scp zsu3@mimi.cs.mcgill.ca:/home/2013/zsu3/COMP206/myfile.txt /users/Joe/Documents
- Windows
  - Download WinSCP https://winscp.net/eng/download.php
  - Click and drag between Mimi and Home

Graphical vs Command Line

- Almost all of us interact with our computer's operating system using a graphical user interface, but you can also do so using just text commands

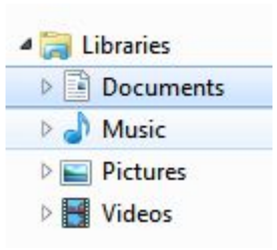


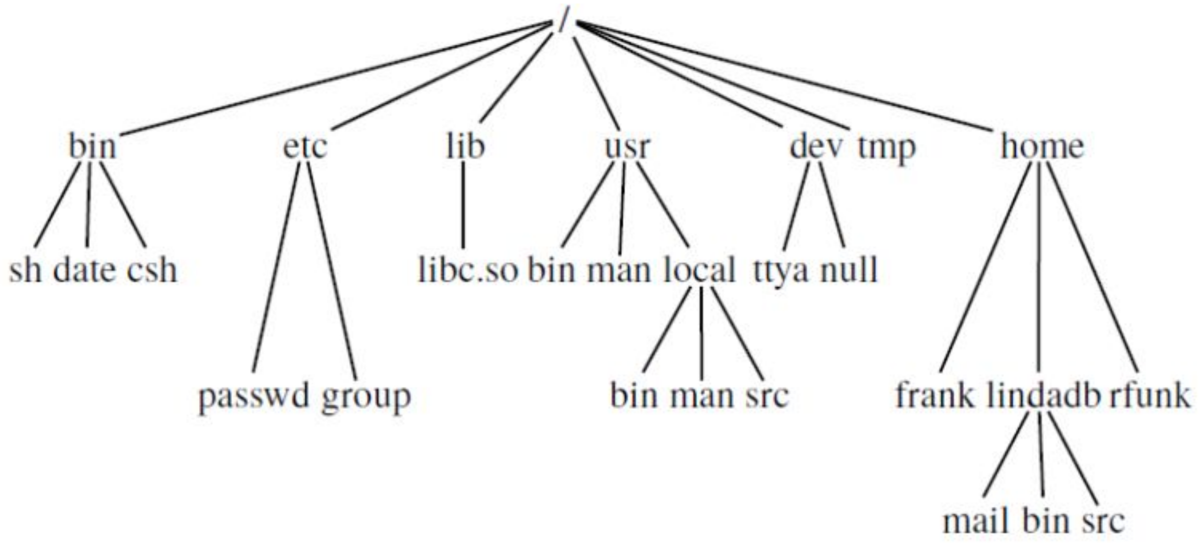


```
howtogeek@ubuntu:~$ 
```

File System

- Just like Windows Explorer or Finder
- Navigate on the command line using the cd command (more on commands later)





File System

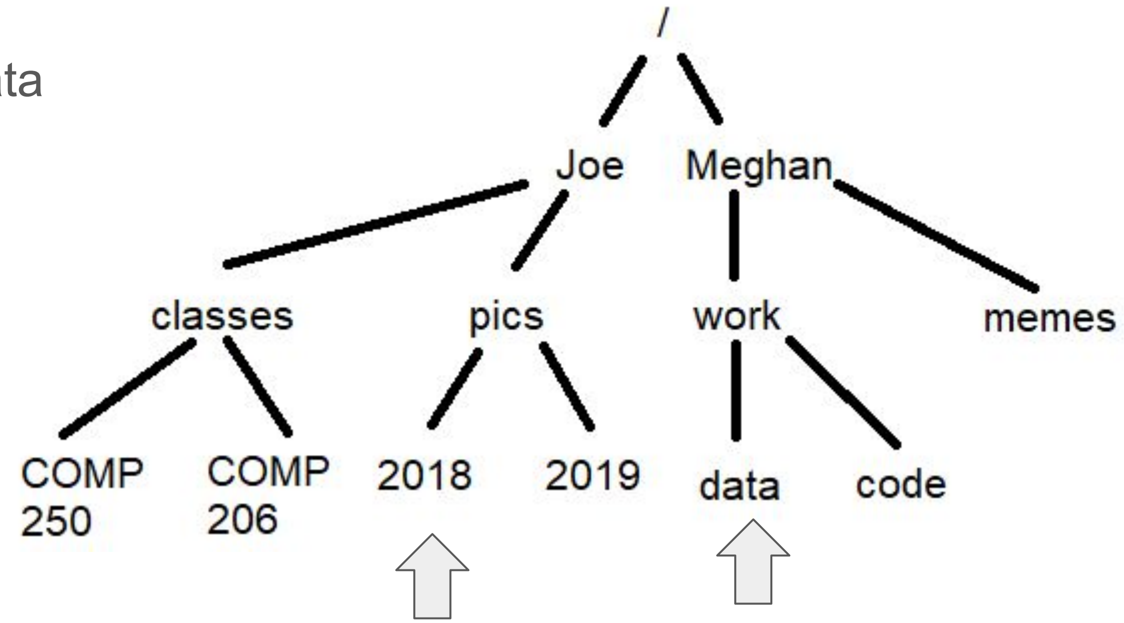
- The command **pwd** prints the current directory
  - If you just logged into mimi, your current directory should be /home/year/username
- Navigate the file system using the command “cd”
- Typing just “cd” without any arguments takes you to **home directory**
- cd directoryname takes you in that directory
- cd directoryname/subdirectory takes you in a directory within a directory
- Layers of directory are separated by the forward slash
- The command **mkdir directoryname** makes a directory

File System

- / (just the forward slash) is the **root directory**
- . (single dot) is **<u>current directory</u>** (optional for things like cd)
    - cd ./pics and cd pics both take you to the directory “pics” if it’s in the current directory
- .. (two dots) is the **parent directory** - this is like pressing ‘back’ in windows explorer

Navigate the File System

- Directories can be referenced from the current directory or the root directory
- If I’m in Joe/pics/2018 and want to go into Meghan/work/data
- cd /Meghan/work/data
- cd ../../../Meghan/work/data
  - Go up three levels



Exercise

- Make a directory in your home directory named COMP206
- Go inside COMP206 (cd) and make five directories one for each assignment
  - COMP206_A1, COMP206_A2, COMP206_A3 ,COMP206_A4, COMP206_A5

Filenames

- Filenames are referenced using directoryname/filename
- If your in the file’s directory, just filename is fine
- Directoryname could be local or rooted
    - /Meghan/work/data/tables.txt
    - ../../../Meghan/work/data/tables.txt (if we’re in the same directory as before)
- DO NOT put spaces in your filenames!!!

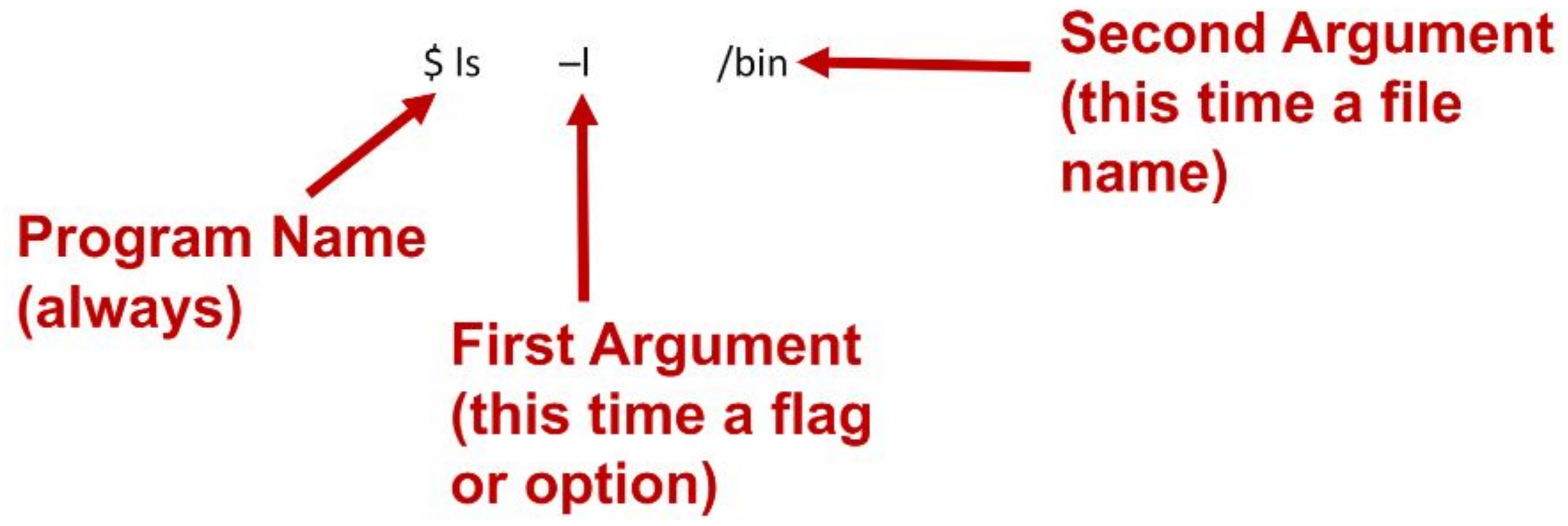
Creating a File

- There are many command-line text editors to use (e.g. nano, vim, emacs)
- Nano is very easy to use but lacks a lot of useful addons
- Simple for short programs/scripts (e.g. assignment 1-2)
- To open a file in nano, type “nano filename” in the command line
- Type some stuff
- Control X, followed by ‘y’, followed by enter to exit (follow the instructions at the bottom of the nano interface)

Vim

- More useful text editor but a harder learning curve (highly recommended on your own time)
- vim filename
- Vim has two interfaces - the “menu” and the “writing” mode, starting in the menu
- press ‘i’ to go from menu to writing mode
- press escape to go from writing mode to menu
- the sequence :wq saves and exits your file from the menu
- ...and many more

Running a Command


$ ls –l /bin
Second Argument (this time a file name)
Program Name (always)
First Argument (this time a flag or option)

You can get the usage of any command by typing **command --help** OR **man command**

ls

- The command ‘ls’ takes some arguments
- Just by itself, ‘ls’ lists all the files and directories in the current directory
- Adding a directory name: lists all the files and directories in that directory
- Adding a file name - lists file (only if it’s there)
- Adding a flag (e.g. -l in the previous example) lists the files with some catch
- You can combine flags (ls -l -a -r OR ls -lar) in any order
- Look up the man or help page to see all the ways ls is used, e.g.:
  - -l lists some relevant information including permissions, last edit time
  - -R lists all contents within directories and subdirectories and so on
  - -t sorts by edit time

Unix wildcard characters

- * any string
- ? any one character
- [abc] any character in the brackets
- [!abc] any character not in the brackets
- Example: ls *.txt lists all files that end in “.txt”
- Example: ls *2018* lists all files with 2018

Moving Files

- Copy: cp old_filename new_filename
    - Copy to current directory without name change: cp /Joe/pics/2018/Vacation1.jpg .
    - The single dot represents current directory
- Move: mv filename new_directory
- Remove: rm filename
    - This will prompt you e.g Are you sure you want to remove?

Viewing files

- You could always open the contents of a file in vim, but sometimes you don’t want to open an entire file (e.g. if they are so large)
- cat filename - displays contents of file to screen
- head filename - displays first 10 lines (default)
- head -5 filename - displays first 5 lines (with flag)
- tail filename - displays last 10 line (default, also works with flags)
- more filename - if you have a large file, it scrolls down
- You can use wildcard characters here too
  - Example: head *.txt lists the first 10 lines of all .txt files

More advanced file viewing

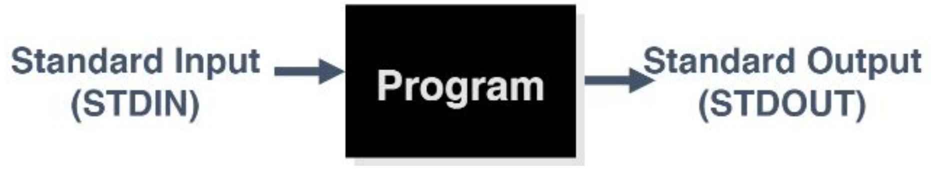
- sort - sorts file
  - Many, many different flags for this command!
- cut - extracts specific columns from a text file.
- grep - searches file
  - grep [query] [filename]
  - grep Joe students.txt
  - Outputs all lines with “Joe” in them - including things like Joel, Joelle, randomstuffJoehello
  - grep uses regular expressions kind of like unix wildcards for complicated queries
- sed and awk
  - More complicated commands to modify contents within files
  - sed - popular for find and replace
  - awk - popular for manipulating the contents of a table
  - Could make assignment 1 a bit easier but you haven’t covered them in class

Exercise

- Download Assignment 1 if you haven't already done so, and follow the setup commands
- Try: wget https://github.com/dmeger/COMP206_Fall2018_Lectures_Public/raw/master/Assignments/Assignment1/COMP206Fall2018_Assign1_Provided.zip
    - wget LINK downloads content.
- Follow the instructions listed on the assignment
- Use some of the file viewing commands to look at the starter files
    - Take a look at all the Question 1 datasets with head *.dat
- Note: Last year we had a problem where accessing Prof. Meger's github may not have worked depending on where you accessed it from. Let a TA or himself know if it were to happen again.

Input / Output

- By default, programs output to standard out (STDOUT). This is your screen.
- The redirection > writes the output to a file (>> appends it)
  - ls > filenames.txt
  - head -1 filename.txt > header.txt
- The pipe | takes the output of a command and feeds is as input for the next command
  - ls | sort | head -1 > firstline.txt
  - Same as:
  - ls > file1, sort file1 > file2, head -1 > firstline.txt



More redirection

- ` backticks (the thing beside the 1 on your keyboard) - nested execution
- cat [filename 1, filename 2, filename 3…] concatenates every file listed
- Example: cat `ls *.txt`

cat `ls *.txt`

- cat a.txt b.txt c.txt

ls *.txt | cat

- cat [a file whose contents is that of ls *.txt]

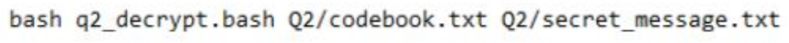
Scripts

- List of commands to be executed one after the other
  - Download everyone’s assignment from mycourses
  - Create a folder for each student
  - Unzip their assignment into the folder
  - bash A1Q1marker.bash ./*/Q1.bash > results.txt
  - and so on
- For example, the assignment 1 setup could be done using a script
- You could technically do everything in a 1 line command, but that could get messy
- Begin scripts with #!/bin/bash
- Run scripts with the command bash filename.bash

Script variables

- my_var=hello
- echo $my_var
- Several variables are set with default
  - Example: $PWD is your current directory
- $0 is the filename
- $1, $2, $3…. are arguments passed at runtime (like java argv)

$0 $1 $2



```
bash q2_decrypt.bash Q2/codebook.txt Q2/secret_message.txt
```

Script conditionals/loops

- If, for, while, just like how you've seen them in Java
- Check lecture 4 slides for full implementation
- You can convert a loop into a one-liner using semicolon
    - for i in code*.txt; do modify.bash $i; done
- **bash is space sensitive - important for comparisons**
    - \`expr 3 + 4\` = 7 - this is true
    - \`expr 3+4\` = 7 - this is false. Bash treats 3+4 as a string that's "3+4"

Assignment 1 Question 1

- Find the right command or series of commands to do the given task without looking inside the contents of the file!
- Don’t worry about the length or complexity of your commands
- You can save information in variables or temp text files for this assignment

Assignment 1 Question 2

- Implement a cipher
- The tr command does ciphering
- Read the man page for tr
- tr OLD_ALPHABET NEW_ALPHABET
- What are the arguments to tr and how do I get them?
- How do I pass my message to tr?

Assignment 1 Question 3

- Sort a list of image filenames and run a program with those names as input
- The commands ls and find can list files and sort by particular options

FAQ From Friday, September 14th, 2018

- If you installed linux on your home computer you may not have all the programs installed, which you will have to do yourself
- Pressing up-arrow takes you to previous commands, e.g. if you made a mistake and don't want to type the whole thing again
- Keep an eye peeled on specific assignment submission methods. Prof. Meger may ask you to upload to mycourses the old fashioned way, or use terminal to upload to his github (or both) and may differ between different assignments
- I made a mistake while describing the cut command - cut by default cuts columns that are separated by tabs. To switch to columns separated by spaces, you need the flag -d ‘ ‘ (single quotes with a space in between). This may or may not be a good way to solve some of the assignment questions ;)
- When you want to go to mimi from another terminal (such as mac) you have to type ssh username@mimi.cs.mcgill.ca because otherwise ssh defaults to using your previous username.