Week 1
Introduction to Linux

09 January 2019

CS 35L Lab 4

Jeremy Rotman

Reminders

- ➔ Assignment #1 is Due January 12 by 11:55pm
  - ◆ If you do not have CCLE access email your answer files to me
    - ● jrotman@ucla.edu
- ➔ If you are still unenrolled, or wish to switch from one lab to another, please continue to list your name in the notebook being passed around
- ➔ Reminder to join the Piazza
  - ◆ http://piazza.com/ucla/winter2019/cs35l

Outline

- ➔ Useful Shortcuts
- ➔ Man Pages
- ➔ A Couple More Commands
- ➔ Emacs
- ➔ Assignment Hints and Tips

Questions?

Some useful shortcuts

- `ctrl-c`
  - Cancel any running command
  - Useful if you’re running something but realize that it won’t actually work
- `ctrl-z`
  - Sends the current process to the background
  - Useful to leave emacs without actually exiting
  - `fg` returns the last program you pushed to background to the foreground

Some useful shortcuts

- `ctrl-l`
  - Clears your terminal screen
  - Essentially just pushes everything above what you can see
  - Useful to make your screen a bit less of a mess
  - Alternatively use the command clear
- `ctrl-u`
  - Deletes the typed line behind the cursor
  - Useful if you’ve written a large command you no longer want to run

Some useful shortcuts

- `ctrl-a`
  - Moves cursor to the beginning of the line
  - Useful if you forgot to type in the actual command
- `ctrl-e`
  - Moves the cursor to the end of the line
  - Useful if you made an edit but want to now continue typing

man Pages

- man pages are split into sections
  - First line:
    - `<command_name> (`*`number`*`)`
      - The number shows you what section of the man page you are visiting
  - You can find the names of the sections through `man man`
    - 1: User Commands
    - 2: System Calls
    - 3: C Library Functions
    - 4: Devices and Special Files
    - 5: File Formats and Conventions
    - 6: Games et. Al.
    - 7: Miscellanea
    - 8: System Administration tools and Daemons

man Pages

- Standard Headings
  - Name
    - Name of the command followed by a short description of what the command does
  - Synopsis
    - The usage of the command
      - `ls [OPTION]... [FILE]...`
  - Description
    - A more detailed explanation of what the command does
    - Also includes the list of options that can be used with the command

Moving Around in man Pages

- The standard is for the man page to be opened in the format of the command `less`
- Useful hotkeys
  - Move forward one line
    - e, j, <down-arrow>, or <enter>
  - Move backward one line
    - Y, k, or <up-arrow>
  - Move forward one page
    - f, <space>, or <page-down>
  - Move backward one page
    - b, or <page-up>

Moving Around in man Pages

- Useful hotkeys
  - Search for *text*
    - /*text* <enter>
    - n to move to next occurrence
    - N to move to previous occurrence
  - Quit the man page
    - q
  - Go to start of man page
    - g
  - Go to end of man page
    - G

Special Permissions

- setuid (u+s)
  - A process that runs this file is granted access based on the owner of the file
- setgid (g+s)
  - A process that runs this file is granted access based on the group owner of the file
- Sticky bit (o+t)
  - On shared directories, prevents anyone from deleting files from the directory unless they are the owner of the file, owner of the directory, of root

More Commands

- diff
  - Output the differences between two files
  - Has multiple forms of output
    - Standard diff prints out any lines that are modified, inserted, or deleted between the two files
    - Unified output displays lines that are modified, inserted, or deleted along with the context of those lines
      - Requires the -u option
- wget
  - Retrieve content from a web server
  - `wget <url>`

Emacs

- Customizable, extensible, self-documenting, real-time display editor
- Essentially, a text editor that can be used in command line
- For assignment 1, you will have to use Emacs
    - For future assignments, you are free to use whatever command line text editor you prefer
        - e.g. vim, nano, etc.
- There is no need to install Emacs if you are working on the seasnet linux server

Emacs

- To open emacs
  - `emacs` *`filename`*
- Emacs commands begin with “C” or “M”
  - “C” = control
  - “M” = alt (Windows) / option (Mac)
  - Mac users may have to change terminal keyboard preferences to recognize option as “M” (meta)
- To exit emacs
  - `C-x C-c`
- To abort a partially typed or executing command
  - `C-g`

Basic Editing

- Insert text
  - Just type the text (no insert mode like vim)
- Undo
  - `C-x u`
- Save changes
  - `C-x C-s`
- Copy, cut, paste
  - `M-w` (copy), `C-w` (kill), `C-y` (yank)

Directory Edit (dired)

- `C-x d`
  - This will then prompt you to enter a directory
- This creates a buffer containing the list of the directory you enter
- Buffer allows you to navigate file system
- Additionally, allows you to operate on files
  - E.g. remove, rename, encrypt, edit, etc.

Running programs from Emacs

- Emacs can run shell commands
  - `M-! <command>`
  - `M-x shell`
    - Opens an interactive shell buffer
- Emacs as an IDE
  - Emacs can also be used to compile programs
  - `M-x compile`
    - Will prompt you for the command to compile
      - E.g. gcc hello.c -o hello
  - The resulting executable can then be run as a shell command
    - E.g. `M-! ./hello`

Other Emacs hints

- `M-x column-number-mode`
  - This will enable mode to show both row and column number at the bottom of your buffer
    - Displays row and column where the cursor is
- 80-column restriction
  - Some assignments specify the maximum column length of your submission
  - Add “setq-default fill-column 80” to ~/.emacs
  - Then when editing your file use the command
    - `M-x auto-fill-mode`
  - Or, just keep note of the column number and make a newline before you ever reach 80

Assignment #1 Hints

- Another reminder, you will submit 2 files in total
  - ans1.txt holds the answers to the 15 questions in the lab section
  - key1.txt holds the keystrokes from the homework section
    - The keystrokes you will manually have to enter into a text editor, for example keep an extra terminal open with emacs to write down while working
    - Notation for the keystrokes can be found in the assignment spec under Submission in the description of key1.txt

Assignment #1 Hints

- Make sure that your files do not have carriage returns
  - Carriage returns (“\r”) are used in line endings on Windows machines, but not linux
    - Windows: “\r\n”
    - Linux and newer MacOS: “\n”
    - Old MacOS: “\r”
  - If you use a text editor on windows, you **will** have carriage returns
  - To remove carriage returns:
    -
      ```
      sed -i.backup ‘s/\r//g’ <submission-file>
      ```
      - This will make the changes in your file, and save the old version to a file named <submission-file>.backup

Assignment #1 Hints

- Make sure to prepend your path!
- Prepend your path using the command
  - `export PATH=/usr/local/cs/bin:$PATH`
- Alternatively
  - Open .bashrc in your home directory: `emacs ~/.bashrc`
  - Append “`export PATH=/usr/local/cs/bin:$PATH`” to the end of the file
  - Save and exit
  - Repeat the above 3 steps for ~/.bash_profile
  - Log out of linux server and log back in
  - Confirm it worked running `echo $PATH`

Assignment #1 Hints

- Some hints for lab questions
  - Number 4: man readlink
  - Number 5: If you find that the files are the same version, make sure that your path is properly prepended.
  - Number 10: man localedef
  - Numbers 11-15
    - `C-h ?`
      - Opens options for Emacs help
    - `C-h r`
      - Opens the Emacs manual

Questions?