UNIX Commands & Pair Programming

Objectives

- Able to make use of many common UNIX/Linux commands
- Learn an editor: VIM or Emacs
- Experience pair programming
- Learn Regex

Lab 1 Exercise – UNIX

Step 1: - **Find a partner**
Buddy-up to help get through the answers faster and also experience pair programming in Step 6. You will need to sit next to each other.
If there ends up being an extra person, then you can make one group of 3.

Step 2: - **VM**
Ensure you have the latest and greatest Virtual Machine from the CS department: https://foundation.cs.colorado.edu/ You should have VirtualBox, VM, Dropbox.

Open your VM and open a terminal window so we can play with UNIX!

Step 3: - **Text Editors**
Unix systems feature a lot of different text editors, such as **pico**, **emacs** (prounced "e-macks", and **vi** (pronounced "vee-i", also know as **vim** pronounced sounds like "him"). We do not require a specific text editor in this class, so use whichever program suits your working style best. However, you should experience each one so you are somewhat familiar with them, and you will need to be very familiar with either emacs or vim.
To get you started, you can learn more about these programs by accessing their on-line help and/or tutorials.

Program	Accessing Help	Accessing Tutorial
emacs	Launch **emacs** and then type **Ctrl-h**	Type **emacs** and then type **Ctrl-h t**
vim	Launch **vim** and type **:help**	Type **vimtutor**
pico	Launch **pico** and type **Ctrl-G**	N/A

Of course, you can always get more information on the Web by performing a search in your favorite search engine using, e.g., a phrase like "pico tutorial".

While pico is an editor you can use, you should learn either vim or emacs.

The TA will guide you through a brief introduction to vi as well as emacs.
Then decide…

For the rest of this lab, use either vim or emacs to record your answers!

Step 4: **UNIX commands**

UNIX commands enable you to do some powerful programming in very few characters. You can either enter the commands at the terminal, or put then in a file and run them as a script.
Nearly all Unix commands follow the same basic structure in how they are entered.

```
command -option argument --more-options
```

Using either vim or emacs, create a file with the following information:

What do the following commands do?

Command	Does what?
date	
ls -ltr	
cd ..	
pwd	
who	
whoami	
man man	
env	

How would you do the following? (Work with your buddy! Search the internet!)
Some useful commands you may need: grep, cd, mkdir, rm, find, head, tail, cp, zip, unzip, rmdir, cat, sort, uniq, less, touch

Command	Goal
	Make a directory named 'cs3308' and move into that directory.
	Rename your directory 'cs3308' to 'csci3308'.
	Change to the root directory.
	Make a copy of a file.
	Delete the copy of your file *(Careful!)*
	Make a directory named 'tmp'. Then delete that directory.
	View the contents of a file.
	2nd way to view the contents of a file.

	View just the beginning of a file.
	View just the end of a file.
	List all files that contain the word ‘the’ in the file.
	List full path to all files named ‘books.txt’
	Zip the contents in your directory into a file named ‘dir.zip’
	Unzip your zipped file ‘dir.zip’ into a new directory named ‘tmp’
	Tar the contents in your directory into a file named ‘dir.tar’
	Untar your zipped file ‘dir.tar’ into a new directory named ‘tmp’
	Modify a file’s last modified timestamp to now. This also creates a new file if it doesn’t currently exist.

Step 5: **Fancy UNIX commands**
For the next set of questions, use the following file:

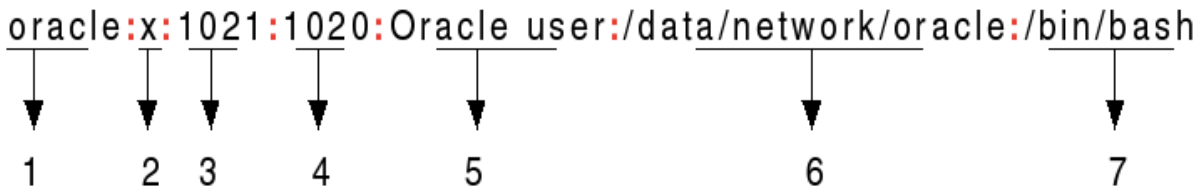
```
root:x:0:0:root:/root:/bin/bash
daemon:x:2:2:daemon:/sbin:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin
harpo:x:12502:1000:Harpo Marx:/home/harpo:/bin/csh
chico:x:12501:1000:Chico Marx:/home/chico:/bin/bash
zeppo:x:12505:1000:Zeppo Marx:/home/zeppo:/bin/zsh
groucho:x:12503:2000:Grouch Marx:/home/groucho:/bin/sh
gummo:x:12504:3000:Gummo Marx:/home/gummo:/usr/local/bin/ksh
```

From: http://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/

Understanding fields in /etc/passwd

The /etc/passwd contains one entry per line for each user (or user account) of the system. All fields are separated by a colon (:) symbol. Total seven fields as follows.

Generally, passwd file entry looks as follows:

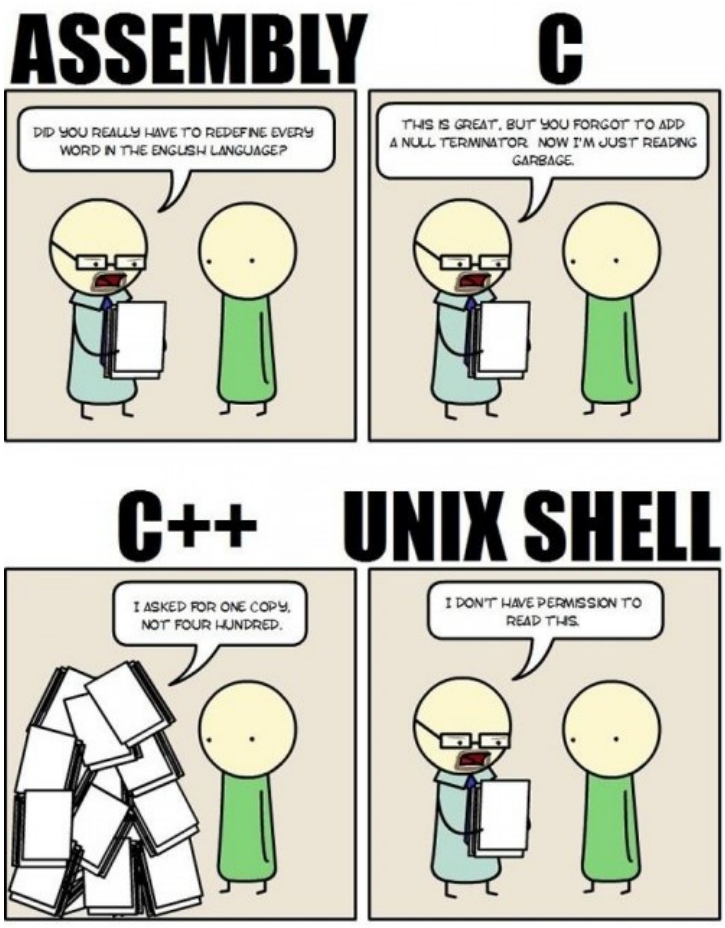


(Fig.01: /etc/passwd file format)

1. **Username**: It is used when user logs in. It should be between 1 and 32 characters in length.
2. **Password**: An x character indicates that encrypted password is stored in /etc/shadow file.
3. **User ID (UID)**: Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
4. **Group ID (GID)**: The primary group ID (stored in /etc/group file)
5. **User ID Info**: The comment field. It allow you to add extra information about the users such as user's full name, phone number etc. This field use by finger command.
6. **Home directory**: The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /
7. **Command/shell**: The absolute path of a command or shell (/bin/bash). Typically, this is a shell. Please note that it does not have to be a shell.

Command	Does what?
	Sort the file based on the userid (first field)
	Sort the file based on the UID. Since it is a number, be sure to specify that it is a number so it sorts the numbers correctly…

	Sort first based on GID, then on UID.
	Show all the lines in the file with 'Marx' in it.
	Get the number of lines in the file (from a UNIX command)
	Use your answer from the previous question and now redirect the output to a file named 'tmp'
	Now store your answer to the previous question in a separate file with an extension of .sh for example **getUniqueGID.sh** Try to run your program. You should get 'command not found' Try to run **./getUniqueGID.sh** You should get 'Permission denied' Change the permissions on the file to allow users to execute the file. Now, which method runs your program? And why doesn't the other one execute? And what is a second way to have changed the permissions? And do you know a third way? ☺



Step 6: **Pair Programming with Regex**

Extreme programming advocates fourteen standard practices; we are going to focus on two of them this semester

- Pair Programming
- Test-Driven Development

For this lab exercise, we will get experience with pair programming.

Pair Programming

- All production code is written by pairs of programmers working together at the same workstation
    - *One member drives the keyboard and writes code and test cases; the second reviews the code, looking for errors and possible improvements, while also thinking strategically about what the team needs to do next.*
    - *The roles will switch between the two frequently.*
- Pair membership changes once per day; so that each programmer works in two pairs each day
    - *Facilitates distribution of knowledge about the state of the code throughout the entire team.*
- Studies indicate that pair programming does not impact efficiency of the team, yet it significantly reduces the defect rate! [Laurie Williams, 2000; Alistair Cockburn, 2001; J. Nosek, 1998]

Step 7:

Find a partner! You should already be working with a partner!

Step 8:

Decide who is driving first. That person has control of the keyboard. After 10 minutes, swap.

Step 9: Working together in a pair programming method, do what the TA tells you to do for a Regex exercise.