CSCI 3308 Software Development Methods and Tools [Fall 2017]

**Instructor:** Alan Paradise & Grace Muzny

**TA**: Pratima Sherkane, Rashmi Shetty , Prasanna Srinivasachar

Lab1 – Material by Liz Boese and Grace Muzny.

**UNIX Commands & Pair Programming**

**Objectives**

Able to make use of many common UNIX/Linux commands

Learn an editor: VIM/EMACS

Experience pair programming (optional for this lab)

Learn Unix Commands

**Lab 1 Exercise – UNIX**

Step 1: - **Find a partner**

Buddy‐up to help get through the answers faster and also experience pair programming. 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 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 and vi (pronounced as “vee-i” also known as vim. We do not require a specific text editor in this class, so use whichever program suits your working style the best. However, you should experience each once so you are somewhat familiar with them, and you will need to be very familiar with either vim or emacs.

To get you started, you can learn about these programs by accessing their on-line help and/or tutorials.

|  |  |  |
| --- | --- | --- |
| **Program** | **Help** | **Tutorial** |
| **emacs** | Launch **emacs**, type **Ctrl-h** | Type emacs, then **Ctrl-h t** |
| **vim** | Launch **vim**, type **:help** | Type vimtutor |
| **pico** | Launch **pico**, 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”.

The TA will guide you through a brief introduction to vi. However, there is no restriction on using only vim for this course.

**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 them 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

In this section, we will be using UNIX commands to explore the directory lab1/, which can be downloaded from Moodle.

Using either vim or emacs, create a file called lab1.txt. Write all partners’ names at the top of the file, then answer the following questions.

Some useful commands you may need: grep, cd, mkdir, rm, find, head, tail, cp, zip, unzip, rmdir, cat, sort, uniq, less, touch, du.

Feel free to use the man pages, trial and error (but be careful!), and your favorite search engine at your leisure to complete this lab.

1. What do the following commands do?

|  |  |
| --- | --- |
| **Command** | **Does what?** |
| date | Print or set the system date and time |
| ls -ltr | Lists files in current directory reverse ordered by time (oldest first) |
| cd .. | Go up by one directory |
| pwd | Print current directory |
| who | Show who is logged in |
| whoami | Current user |
| man man | Manual for manual |
| env | Run a program in modified environment |
| touch | Create an empty file |

2. 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.





**Fancy UNIX commands**

For the next set of questions, use the file passwd\_demo.txt in lab1 folder. Its contents is reproduced below:

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:Groucho Marx:/home/groucho:/bin/sh

gummo:x:12504:3000:Gummo Marx:/home/gummo:/usr/local/bin/ksh

Source: <http://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/>

**Understanding fields in /etc/passwd**

The /etc/passwd file 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:



(/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 allows 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 exist, 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.

Using the above mentioned file as input, please figure out the commands that do the following tasks:

Some useful commands you may need: grep, sort, chmod

|  |  |
| --- | --- |
| **Command** | **Does what?** |
|  | Sort the file based on the username (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 the word ‘*Marx’* in it. |
|  | Get the number of lines in the file (from a Unix command) |

|  |  |
| --- | --- |
| **Command** | **Does what?** |
|  | Print only the lines in the file with a unique GID |
|  | Use your command for the previous question and now redirect the output to a file named tmp. Use the > operator.  Hint: try echo “hello” > test |
|  | Now write your answer (command) to find the number of lines to a separate file with an extension of .sh, for example, the file name could be  getUniqueGID.sh. Try to run your program with the command filename.sh.  You should get a command not found error. Try to run ./filename.sh. You should get a permission denied error. Change the permissions on the file to allow users to execute the file. Now, what is the command to run 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? |

2. Change directories into lab1/. List the contents of this directory.

3. What is the contents of the hidden text file in lab1/? What command did you use to display it?

4. What command can you use to list the files in the pictures/ directory, sorted according to their size, listing one file per line? Which file is the biggest one and how big is it?

5. Create a new directory called big\_stuff and copy both the largest file from the pictures directory and the largest file from the texts directory into it. How many bytes is this directory and its contents?

6. Use the zip command to create a file called big\_stuff.zip that contains the files currently in big\_stuff. How many bytes is this file?

7. Use the tar command to create a file called big\_stuff.tar that contains the files currently in big\_stuff. How many bytes is this file?

8. Delete the big\_stuff directory and the file big\_stuff.zip. What commands did you use?

9. Use the tar command to extract a file called big\_stuff.tar. What options did you use? What does each one do?

10. What is the difference between little1.zip and little2.zip? Use the unzip command to find out.

11. Use find to find the file called target.txt. What is its path from the lab1/ directory? What’s its full path on your machine?

12. Finally, use the commands head and tail to create a new short story made up of the first line of one of the books in the texts/ directory and the last line of a different book in the texts/ directory. Write down the commands that you used and your new extra short story here.

**Credit:**

To get credit for this lab exercise, show the TA your code and submit your answers for the above tasks on moodle. All partners should submit copies of the same file.