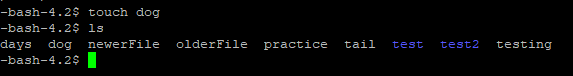
Quiz Week 4

(30 points total)

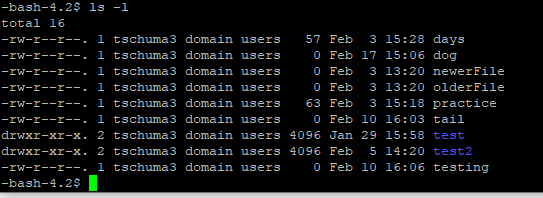
Using what you’ve learned so far in the course, specifically during week 4, answer the following questions.

Part A – Permissions

A-1. (1 point) Use the **touch** utility to create an empty file named **dog** in the working directory. Use **ls** to confirm the file was created. **[Show your command and results]**



A-2. (6 points) Use **ls –l** to display the permissions of the file you just created. Who owns the file? To which group does it belong? Which permissions does the owner of the file have? The group? Others? **[Show your command and results]**

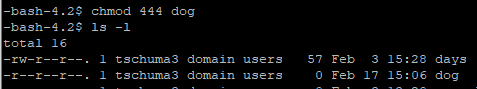
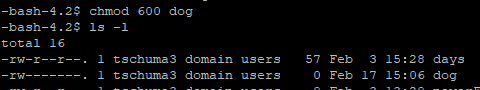


A-3. (7 points) Display the permissions of **/bin/bash**. Who owns the file? To which group does it belong?

Which permissions does the owner of the file have? The group? Others? Which permissions apply to you? **[Show your command and results]**



A-4. (3 points) Only the owner of a file (or a user working with **root** privileges) can change the permissions of a file. Using numeric arguments with the **chmod** command, change the permissions of the file you created in step 1 so that the owner has read and write permissions and the group and others have no permissions. Next change the permissions so the owner, the group, and others have only read permissions. Display the permissions of the file before you start and after each change. **[Show your command and results]**



Part B –Locating Files and Utilizing Command Options

B-1. What are the three major commands utilized when adding additional storage to a Linux system? Note: List them in the order they are executed.

**fdisk** then **mkfs** then **mount**

B-2. List two benefits for each of (a) **yum** verses **RPM** and (b) **yum/RPM** verses using a **tarball**

**Two benefits of YUM are** **that YUM supports package groups and YUM provides library names. Two benefits of YUM/RPM are that YUM/RPM can provide maintenance tasks like removing unneeded packages and YUM/RPM can see where items are installed.**

Part C –Locating Files and Utilizing Command Options

The **find** utility locates files based on simple or complex criteria you specify. For example, find can list all files in **/usr/bin** that were modified more than 900 days ago.

$ **find /usr/bin -mtime +900**

/usr/bin/test\_ppa

/usr/bin/detect\_ppa /usr/bin/lftpget ...

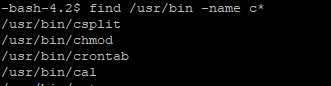
Try specifying **600** or **1200** in place of **900**. Use a minus sign (**–**) in place of the plus sign (**+**) to specify files that were modified more recently than the number of days you specify. To specify an exact number of days, omit the **+** or **–** sign. Send the output through a pipeline to **wc –l** to count the number of files (lines) that find displays.

$ **find /usr/bin -mtime +300 | wc -l**

711

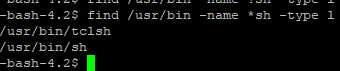
Reference the built-in Linux help pages to explore the functionality of the **find** command and get the correct syntax.

C-1. (2 points) Using the **find** command with the **-name** criterion, search the /**bin** directory for files that beginning with the letter **c**. **[Show your command and results– first 4 lines of the result]**

****

C-2. (2 points) Using the **find** command with the **–size** criterion, search your home directory for files that are smaller than 30 bytes. **[Show your command and results]**

****

C-3. (2 points) Using the **find** command with the **–name** and **–type** criteria, search the **/bin/** directory for files that have the characters **sh** in their names and are symbolic links. **[Show your command and results]** ****