Lab 5: **The Linux Labs – Standard Linux Commands**

CSI4103 – Web Application Software Design

Faculty of Engineering – University of Ottawa

Objective:

Understand and practice how to use linux command line instructions. This is useful when working on server backends through a terminal window (e.g. using telnet or an SSH connection). This part of the lab continues on with a review of basic linux commands that are needed when manipulating permissions on files and directories.

Instructions:

- Read the online help for each command listed below.
  - **chmod**- change a file's mode
  - **umask** - display & modify default permissions
  - **chown** - change ownership
  - **Any other command that you require additional information about**.

- Additional useful commands
  - **cd** - to go back to your home directory
  - **pwd** - to verify that you are in your home directory
  - **whoami** - use this command when you don't know who you are

Viewing permissions with the `ls` command

The permissions control who is allowed to access a file or directory. There are three sets of permissions, one set for the file's owner, one set for the group, and a third set for all other users. This is called the file's mode.

When performing a long directory listing, as in ls –l, the file's mode appears in the first field (column) of the directory listing.

- The first character identifies the file type:
  - (-) for regular file
  - (d) for directory
- The following nine characters refer to the access permissions. The permissions you will encounter are:
  - (r), for read
  - (w) for write
  - (x) for execute

The first three characters are the permissions associated with the user (owner). These are the permissions we will explore for now.

For example:

```
$ ls

-rw-rw-r--     1  arnold   arnold   1      Nov 12 14:14      eraser

-rw-rw-r--     1  arnold   arnold   1      Nov 12 14:14      conan

drwxrwxr-x     2  arnold   arnold   1024   Nov 12 14:14      scripts
```

In this example **conan** and **eraser** are identified as regular files; **scripts** is identified as a directory. The owner of all three objects is **arnold** (the user). The permissions of scripts for the owner (arnold) are read (r), write (w) and execute (x). Don’t worry about the rest for now.

Changing permissions with the chmod command

To change permissions you use the `/bin/chmod` command. `chmod` requires two arguments: the permissions to be changed and the object for which the access permissions is set. For example:

- `chmod u+rwx eraser`

  Gives the owner (u) all three access permissions (read, write, and execute) for test.

For more information use the online help: `man chmod`.

Testing permissions

While logged in as a user, use **mkdir** to create the following directory: /home/user/top.
To complete exercise #1, follow the steps listed below.

1.) Change the permission of the **top** directory using the `chmod` command. The exact command is given in the first column of the table.

2.) Execute the commands listed in the first row (starting with the third column) for that permission level. For each command line document whether the command line executes successfully or not. Use **PD** for Permission Denied; **OK** for success. Do NOT fill in the boxes that contain **---**. When a command did not execute successfully, write a brief statement explaining why it didn't.
The commands are:
- **ls -l top**
- **mkdir top/sub**
- **rmdir top/sub**
- cd top
- cd .. (execute this ONLY if your current directory is top!)

3.) Follow the above procedure for each row of the table (row 1 to 8).

Note #1: If a command completes successfully, but it gives you a "Permission Denied" message, mark it as "Partial".
Note #2: Permissions are checked after *a command verifies that the object exists. In other words, "File does not exist" is **NOT** an indication of "Permission Denied".*

Exercise #1: Directory permissions

Row #	Command line to modify permissions	ls -l top	mkdir top/sub	rmdir top/sub	cd top	cd ..	Your comments
1	chmod u+r-w+x top						
2	chmod u-r+wx top						
3	chmod u+rw-x top						
4	chmod u-rw+x top						
5	chmod u-r+w-x top						
6	chmod u+r-wx top						
7	chmod u-rwx top						
8	chmod u+rwx top						

Summary

The read permission on a directory allows you to

____

The write permission on a directory allows you to

____

The execute permission on a directory allows you to

____

The read and execute permission on a directory allows you to

____

The write and execute permission on a directory allows you to

____

Default permissions

Exercise #1: Viewing a user's default permissions

Login as user.

- Type umask and record the output of the command: _______________
  - Based on the bitmask, what are the default permissions for directories and files in octal mode, based on your bitmask:
    directory: ___________& file: ___________
- Verify it by creating a new file with the touch command.
  - Record the default permissions set on the file in symbolic mode: ________________
  - Record the default permissions set on the file in octal mode: ________________
- Verify it by creating a new directory with the ..., well you know which command to use.
  - Record the default permissions set on the directory in symbolic mode: ________________
  - Record the default permissions set on the directory in octal mode: ________________

Exercise #2: Viewing root's default permissions

Login as root.

- Type umask and record the output of the command: _______________
  - Based on the bitmask, what are the default permissions for directories and files in octal mode, based on your bitmask:
    directory: ___________& file: ___________
- Verify it by creating a new file.
  - Record the default permissions set on the file in symbolic mode: ________________
  - Record the default permissions set on the file in octal mode: ________________
- Verify it by creating a new directory.
  - Record the default permissions set on the directory in symbolic mode: ________________
  - Record the default permissions set on the directory in octal mode: ________________

Exercise #3: Changing default permissions

Log in as user.

- Set the umask to 077 (paranoia setting)
- Type umask and record the output of the command: _______________
  - Based on the bitmask, what are the default permissions for directories and files in octal mode, based on your bitmask:
    directory: ___________& file: ___________
- Verify it by creating a new file.
  - Record the default permissions set on the file in symbolic mode: ________________
  - Record the default permissions set on the file in octal mode: ________________
- Verify it by creating a new directory.
  - Record the default permissions set on the directory in symbolic mode: ________________
  - Record the default permissions set on the directory in octal mode: ________________

Exercise: Ownership

Creating new users

For this lab we will require one user account. To create the two user accounts follow these steps (you have to be root):

- **useradd pinky** - to create a user, whose logon name will be pinky (or use your name)
- **useradd brain** - to create a user, whose logon name will be brain (or use another name)
- **passwd pinky** - type in a password when prompted. If you do not type the username after the passwd command, you are changing the root password!!
- **passwd brain** - type in a password when prompted.

exit - to logout as root.

Part #1

- Login as root and create a directory called public under the root directory: **/public** (NOT /root/public).
  - o Who is the owner of the /public directory? ____________
  - o Which group is the owner of the /public directory? ____________
- Give full access permissions for everybody.
  - o Record the command you use: ____________________

Part #2

- Login as brain and create a file named **plan** in the /public directory using the cat command.

  Hint: Type cat > plan *and add text at the blinking cursor. Press* ctrl+d *when you are done.*

  - o Who is the owner of that file? _______________
  - o Which group is the owner of that file? ____________
- Make sure that others have no access permissions (use the command chmod o-rwx /public/plan). Verify with ls -l that you achieved the desired result.

Part #3

- Login as pinky and modify the file using the following command: cat >> /public/plan.
  - o Record the message: _________________________________

Part #4

- Login as root and change the ownership of **plan** to pinky using the following command: chown pinky.pinky /public/plan

  Verify that pinky is the owner of plan.

- Login as pinky and modify the /public/plan. Can you do it? ________
- Login as brain and modify the /public/plan. Can you do it? ________
- While you are logged in as brain delete the file. Can you do it (eventually)?

Note: If you use vi instead of the cat command you may override the read-only option by appending the save command with an exclamation mark (:wq!). However, this will also change the ownership of the file to the person who modified it. This only applies if you have write permissions to the parent directory.

Review Exercise

Exercise #1: Directory permissions

Circle the minimum permissions required *by a regular user* to successfully complete the actions listed below, based on the information collected in Table #1 of the Lab document.

- To produce a directory listing of a directory (ls –l /test1/test2/), the user requires for that directory: R W X
- To change into a directory, the user requires for that directory: R W X
- To add a file or subdirectory into a directory, the user requires for that directory: R W X

Exercise #2: File management

Circle the minimum permissions required *by a regular user* to successfully complete the actions listed below. The information in Table #2 is not sufficient to answer the following questions.

- To copy a file the user requires
  - for the source directory: R W X
  - for the target directory: R W X
  - for the source file: R W X

- To move a file the user requires
  - for the source directory: R W X
  - for the target directory: R W X
  - for the source file: R W X

- To delete a file the user requires
  - for the directory: R W X
  - for the object file: R W X

Exercise #3: File permissions

Circle the minimum permissions required *by a regular user* to successfully complete the actions listed below.

- To display the contents of a file (cat myfile)? R W X
- To modify the contents of a file (vi myfile)? R W X