Lab_04_2: Advanced Permissions

Otago Polytechnic, IN616 Operating Systems Concepts,

Semester 1 – 2020

1 Objectives

- Exploring the File System
- Applying default permissions using umask

2 Setting Default Permissions

Login using the **student** account we have been using for the last few classes. In the lecture today we discussed the use of `umask` and how to set default file and directory creation permissions. This is an exceptionally useful feature as it allows files and directories to be created with a default permission set that we enforce.

Q1. What is the default file creation mask (or *umask*) on your system?

HINT: Try running the `umask` command (without any arguments).

Using the answer from your question, we should determine the permissions that are set when creating files and directories. Recall the lecture, where we used the following equation to determine the **effective permission**:

$$EFFECTIVE\ PERMISSION = BASE\ PERMISSION - umask \qquad (1)$$

To determine the effective permission, we first need to know two things:

1. Base permission value
2. umask value

We already know the umask value (from Question 1), but what about the base permission?

Q2. What is the base permission for file creation?

Q3. What is the base permission for directory creation?

Great! Now we have all the values required to determine what the **effective permissions** are going to be on our system.

Q4. Using the answer from Questions 1 and 2 and our knowledge of the file base permission... What is the default creation permissions for **files**? Write your answer in octal and symbolic notation.

Q5. Using the answer from Question 1 and 3 and our knowledge of the directory base permission... What is the default creation permissions for **directories**? Write your answer in octal and symbolic notation.

It would probably be a good idea to test your solution. In your virtual machine, perform the following steps:

- Make sure you are in your home directory: `cd ~`
- Create a file: `touch testfile`
- Then view the file in long list format: `ls -l`
- Compare the permissions for `testfile` to your Question 4 answer
- Create a directory: `mkdir testdir`
- Then view the file in long list format: `ls -l`
- Compare the permissions for `testdir` to your Question 5 answer

If both files and directories have the same permissions in your system, as compared to your answers ... you are correct! Well done.

3 Modifying Default Permissions

After testing the default permissions and default umask, we want to modify the umask value to apply different *effective permissions*.

OUR GOAL: When a user creates a file we want to ensure that:

1. The user (owner) can read and write
2. The group (group) can only read and write
3. Anyone else (other) can not do anything

Q6. What is the symbolic notation for the file permissions we want to set?

HINT: Remember, symbolic notation refers to the **characters** to describe read, write and execute. For example: `rw-r-xr-x`.

Q7. What is the octal notation for the file permissions we want to set?

HINT: Remember, octal notation refers to the **numbers** to describe read, write and execute. For example: `777`.

OK. So we have worked out what we require as the **effective permission set**. We have also previously determined what the **base permission set** is. But how do we use these two values to determine the value we need to use to set the `umask` value? We can use the formula below to help us...

$$umask = BASE\ PERMISSION - EFFECTIVE\ PERMISSION \tag{2}$$

Q8. Set the default file creation mask (`umask`) for your user to allow the owning user to read and write files, but no one else. Document the command.

HINT: Use you answer from the previous two questions as input into the specified umask equation.

Remember, you can check you new file creation mask by creating a new file.

- Try `touch ~/tf1` to create a new file
- Then view the file in long list format: `ls -l ~`
- Check the permissions!

Once done setting the umask value, log out of the **student** account (use the command: `exit` or `logout`). Log in again using the **student** account. Check the `umask` value by executing the following command:

```
umask
```

Q9. Why does logging out reset the `umask` value we set? And, how/where would we save a specific `umask` value to always be set?

4 Setting umask for Users

In the previous exercise we performed some basic umask configuration. We also witnessed that when a umask value is set on the command line only it is removed after logout. This is good and bad. It is good because we can easily set a temporary umask value for one session. But it is bad if we want to always have that umask value set for a specific user.

In this exercise we are going to configure the *effective permissions* for the `frodo` account.

Login using the `frodo` account... and perform the following actions:

- Create a directory in your home directory named `adventures`
- Create a file in your home directory named `theShire`

Q10. What is the permission set for the directory named `adventures`?

Q11. What is the permission set for the file named `theShire`?

Now we want to modify the *effective permissions* for the `frodo` user account so that it meets the following configuration:

1. The user (owner) can read and write both files and directories
2. The group (group) can not do anything
3. Anyone else (other) can not do anything

Q12. What is the umask value we need to set to achieve the above configuration?

Don't run the command yet. We want to add this value to the `.bashrc` file so that it is persistent - that is, it is saved and applied when ever frodo logs in. To achieve this we are going to use a combination of the `echo` command and redirect the output.

```
echo "<answer-to-question-12>" >> .bashrc
```

We should break this command down to understand what is happening...

Q13. Use the help menu or man page to determine what the `echo` command does. What is the purpose of the `echo` command?

HINT: Try running just the first part of the command. For example: `echo "umask 077"`.

Q14. What do the two greater than symbols achieve?

HINT: Try a Google search for more information.

Q15. What is the purpose of the `.bashrc` file?

HINT: Have a quick look at the following resource: `https://www.lifewire.com/bashrc-file-4101947`. We will investigate this file in more depth in the next class.

Make sure you have already run the specified command:

```
    echo "<answer-to-question-12>" >> .bashrc
```

Now, we should check the `.bashrc` file to make sure that the command is listed. Check the contents of the file, our line should appear at the bottom of the file, on the last line (because we appended the line).

```
    cat .bashrc
```

Logout of the `frodo` account, and log back in. Try to create a file and a directory:

```
    touch testUmaskFile
    mkdir testUmaskDirectory
```

You should see the following permissions:

```
frodo@server:~$ ls -la
drwx------ 2 frodo frodo 4096 Aug  9 08:57 testUmaskDirectory
-rw------- 1 frodo frodo    0 Aug  9 08:57 testUmaskFile
```

Success! We have set the effective permissions for the frodo account, and managed to make the settings persistent!

Q16. What is a command to use to list only files in `/home/frodo` that start with `testUmask`?

HINT: You could use the `find` command, or use `ls` and pipe the output to the `grep` command.