

Lab_04_1: File System Permissions

Otago Polytechnic, IN616 Operating Systems Concepts,
Semester 1 – 2020

1 Objectives

- Managing file system ownership
- Managing file system permissions

2 Lab Configuration

In the last class we created the following users and groups:

- Username: `frodo`
- Home directory: `/home/frodo`
- Secondary groups: `none`
- Default shell: `/bin/bash`

- Username: `samwise`
- Home directory: `/home/samwise`
- Secondary groups: `sudo`
- Default shell: `/bin/bash`

Q1. How can you determine if these users are on the system? What command would you use?

HINT: There is a file that you can view to determine this information.

Q2. What groups is the user `frodo` associated with?

HINT: There is a file that you can view to determine this information. Or you can log in with the user and use the `groups` command.

Q3. What groups is the user `samwise` associated with?

HINT: There is a file that you can view to determine this information. Or you can log in with the user and use the `groups` command.

Before continuing make sure that the following requirements have been met:

- The user `frodo` exists on the system
- The user `samwise` exists on the system
- Both `frodo` and `samwise` are associated with the group named `fellowship`

We want to make one additional configuration to the user structure we have in place. We want to add the user `frodo` to the `sudo` group.

Q4. Document the command you used to add `frodo` to the `sudo` group?

HINT: You need to modify the user configuration.

3 Managing Permissions

Now that we have users and groups, we can talk about ownership and permissions in relation to the file system: that is, files and directories. An essential command to identify ownership and permission sets is:

```
ls -al
```

Remember, `ls` is used to list the contents of a directory. An easy way to remember the command is **list stuff**. The `-a` argument is to list **all** entries (such as hidden files), and the `-l` argument is the long list format. We need long format to see the file system permissions!

3.1 Exploring Permissions

Log into your Ubuntu Linux using:

```
Username: student
Password: P@ssw0rd
```

Now, change to the `/home` directory. This is the usual directory for storing user home folders.

```
cd /home
```

Make sure you are not in your home directory (`/home/student`, rather you are in the `/home` directory).

Q5. Is `/home` a relative or absolute path?

Now, list the contents of the `/home` directory.

```
ls -lisa
```

You should see output similar to the code snippet below:

```
student@server:/home$ ls -lisa
total 32
  792 4 drwxr-xr-x  8 root    root    4096 Aug  7 07:07 .
    2 4 drwxr-xr-x 23 root    root    4096 Jul 13 08:34 ..
161230 4 drwxr-xr-x  2 frodo   frodo   4096 Aug  2 07:14 frodo
161180 4 drwxr-xr-x  3 manager manager 4096 Aug  2 16:05 manager
161234 4 drwxr-xr-x  2 samwise samwise 4096 Aug  2 07:15 samwise
161249 4 drwxr-xr-x  3 student student 4096 Aug  7 07:08 student
```

Notice how each user is the user owner, and group owner for their respective directories. For example, frodo is the owner of the /home/frodo directory.

Q6. What is the permission set for the manager account home folder (/home/manager)?

Q7. What is the permission set **octal** for the manager account home folder (/home/manager)?

Q8. Based on the permission set, what can you (as the student account) do in the manager account home folder (/home/manager)?

Try to enter the manager account home directory using the cd command:

```
cd manager
```

Q9. Is manager a relative or absolute path (as used in the above command)?

Try to create a file in the manager accounts home directory:

```
touch newFile
```

Q10. Could you create the file? Explain why, or why not?

Try to view a file in the manager accounts home directory. You can list the directory contents using:

```
ls -lisa
```

Then, pick a file at random and view using the cat command.

Q11. Could you view a previously created the file? Explain why, or why not?

4 Creating a Shared Folder for The Fellowship!

At this stage, we should have the users `frodo` and `samwise` configured and added to the `fellowship` group. We are going to create a shared folder for this group so that only fellowship group members can access it. **Make sure you are logged in as `samwise`.**

- Create a new folder `/theFellowshipShared`
- The `mkdir` command will help you here (you might need `sudo`!)
- Change into the root directory `cd /`
- And use `ls -la` to list the directory contents
- Inspect the current directory permissions for `/theFellowshipShared` directory

Q12. What is the permissions for the `theFellowshipShared` directory?

Q13. Who is the user owner and group owner for the `theFellowshipShared` directory?

Now we want to configure `/theFellowshipShared` directory to be only accessible to members of the **fellowship** group. Currently, only **frodo** and **samwise** are members of this group, and no-one else.

Assign ownership of the directory to the user **samwise**, and the group **fellowship** by running:

```
chown samwise:fellowship /theFellowshipShared
```

Run `ls -la` to verify the change has occurred (inspect the owner and group columns).

Finally, we are going to assign permissions so that only fellowship group members can access this folder. We want the following configuration:

- User: Full access
- Group: Full access
- Other: No access

Q14. Based on the requirements listed above, what octal permission set are you going to use?

Q15. Document the full command to enforce the permission set on the `/theFellowshipShared` directory?

Excellent, now it is time to check who can access the shared folder. First, try `samwise`:

- Make sure you are in the terminal, logged in as **samwise**
- Check if you can view the contents of the directory: `ls -la /theFellowshipShared`
- Check if you can create a file in the directory: `touch /theFellowshipShared/samwise-testfile`

Now, try `frodo`:

- Make sure you are in the terminal, logged in as **frodo**
- Check if you can view the contents of the directory: `ls -la /theFellowshipShared`.
This time you may notice some interesting output from the `ls -la` command.
- Check if you can create a file in the directory: `touch /theFellowshipShared/frodo-testfile`

Now, try with the `student` account:

- Make sure you are in the terminal, logged in as **student**
- Check if you can view the contents of the directory: `ls -la /theFellowshipShared`.
This time you may notice some interesting output from the `ls -la` command.
- Check if you can create a file in the directory: `touch /theFellowshipShared/student-testfile`

Q14a. Why can you not create a file or view the directory contents in `/theFellowshipShared` directory as the `student` account?

Q14b. If you wanted to give read access to the `/theFellowshipShared` for the other group, what command would you use?

5 Examples

The following question focuses on you identifying and determining a variety of permission sets. You need to identify the correct permissions for 1) owner, 2) groups and 3) others. Write the permission sets in the **r**, **w** and **x** format. For example, `rw-rw-rw-rw` syntax, as seen when you use `ls -la` on a directory or file. Do not use an online tool for this task. You will need to learn permissions sets and octal notation for the assessments in this paper (and will not always have access to an online tool).

Q15. `chmod 111 folder`

Q16. `chmod -R 500 folder`

Q17. `chmod 020 folder`

Q18. `chmod 004 folder`

Q19. `chmod 744 folder`

Q20. `chmod 751 folder`

Q21. `chmod ugo+r folder`

Q22. `chmod -R u=rw,g+r,o=x folder`

6 Advanced Permissions

In the previous section (before the Examples section) we created a shared directory that was named `/theFellowshipShared` and granted access to the users `frodo` and `samwise`. We now want to add in additional shared folders into `/theFellowshipShared`, one for each user. Here is an example of the required folder structure:

```
/
└─ theFellowshipShared
    ├── frodo
    └── samwise
```

The following requirements are specified:

- Create a directory for each user (`frodo` and `samwise`) inside the `/theFellowshipShared` folder
- The directory name should be the same as the user name
- The directory should only be accessible by the specific user
- No other users should be able to access the directory
- Make sure the user and group owners are solely for the specific user

The following hints have been provided:

- It might be easier to log in as each user to create the folders
- Use `chown` to change folder ownership
- Use `chmod` to change folder permission
- You can use primary groups (actually, this is best practice)

Make sure you test your shared folder configuration to ensure it works. Logging into each user account and testing is the best method. Basically, the `frodo` account should be able to access the `frodo` directory. No-one else should have access. Likewise, the `samwise` account should be able to access the `samwise` directory. No-one else should have access.

Make sure to document any commands you used below to help you in later assignments and assessments.

7 Additional Resources

To review and broaden your overview of commands and exercises have a look at the following chapters from the Linux Essentials Course on Netacad:

- Setting up users: Chapter 14
- Managing ownership and permissions: Chapter 15