ComS 252 Homework 2: Files and Permissions

Individual assignment

Due September 14, 2021

1 Objectives

In this assignment, you will use file utilities, including the `vim` editor, in a command–line Linux environment. This assignment has rather specific instructions; you will be graded on files being in the correct place, with the correct contents, and having the correct permissions.

2 Download

Download the virtual machine `Hw02.ova`. There are three user accounts on this virtual machine: `alice`, `bob`, and `chuck`. All activities (except submission) should be completed as user `alice`. You may want to login as `bob` or `chuck` to verify that file permissions are set correctly. Passwords on this virtual machine follow the horribly insecure convention of appending "`pw`" to the account name to get the account password (e.g., the password for account "`alice`" is "`alicepw`").

3 Organize files

From `alice`'s home directory, switch to directory `MyLibrary`, used to maintain different releases of a software library. Within this directory, subdirectory `Current` holds the source code for the current (under development) version of the library, while subdirectory `Releases` contains subdirectories that are copies of the `Current` directory corresponding to various releases of the library[1]. Perform the necessary commands (as `alice`) to reorganize this structure as follows (in order):

1. Rename subdirectory `Releases` as `Releases-supported`.
2. Create subdirectory `Releases-unsupported` (it should be a sibling directory of `Releases-supported`); older releases that are no longer supported will go here.
3. Move subdirectories `0.3`, `0.6`, `0.7`, `0.8` from `Releases-supported` to `Releases-unsupported`.
4. Remove all files ending in `.o` appearing anywhere within or below subdirectory `Releases-unsupported`.
5. Create a new subdirectory, `Releases-supported/2.1`, and copy all files and all subdirectories from `Current` to the new subdirectory (as would be done when releasing version 2.1 of the library). For example, file `Current/README.txt` should be copied into `Releases-supported/2.1/README.txt`.

4 Change permissions

Note: the Turnin script will examine these directories and files and complain if files are missing. You are therefore encouraged to take a snapshot of your (shut down) virtual machine before testing your permissions, especially the ability of users to delete files.

[1] This is just an illustration; revision management tools like `subversion` and `git` are better choices in practice.

4.1 Directory `readable`

Adjust the group and permissions of directory `~/other_perms/readable` and the files within, as necessary, so that:

- `alice` has permission (only) to:
  - View the contents of existing files in this directory (using, say, `cat`)
  - Modify the contents of existing files in this directory
  - List files in this directory (using `ls`)
  - Create files in this directory
  - Delete files in this directory
  - Change into this directory
- Users other than `alice` have permission (only) to:
  - View the contents of existing files in this directory (using, say, `cat`)
  - Change into this directory

4.2 Directory `deletable`

Adjust the group and permissions of directory `~/other_perms/deletable` and the files within, as necessary, so that:

- `alice` has permission (only) to:
  - View the contents of existing files in this directory (using, say, `cat`)
  - Modify the contents of existing files in this directory
  - List files in this directory (using `ls`)
  - Create files in this directory
  - Delete files in this directory
  - Change into this directory
- Users in group `albob` have permission (only) to:
  - View the contents of existing files in this directory (using, say, `cat`)
  - List files in this directory (using `ls`)
  - Create files in this directory
  - Delete files in this directory
  - Change into this directory
- All other users have no permissions.

4.3 Web Example

`alice` is the webmaster for a small website[2] whose files are contained under `~/WWW`. The website is expanding, and now a few people will maintain the site. The system administrator created a group, `webadmin`, containing users (currently, `alice` and `chuck`) who should be able to modify the website. Adjust the group and permissions for `~/WWW` and everything contained in it, as necessary, so that:

- `alice`'s permissions are unchanged.

[2]This is just a motivating example. Incidentally, the files are part of the site http://zapatopi.net/treeoctopus/.

- Users in group `webadmin` have permission (only) to:
  - View the contents of existing files in or below `~/WWW`.
  - Modify the contents of existing files in or below `~/WWW`.
  - List items in or below `~/WWW` (using `ls`).
  - Create items in or below `~/WWW`.
  - Delete items in or below `~/WWW`.
  - Change into `~/WWW` or any directory below it.
- All other users have permission (only) to:
  - View the contents of existing files in or below `~/WWW`.
  - List items in or below `~/WWW` (using `ls`).
  - Change into `~/WWW` or any directory below it.

5 Creating and editing text files

There are several useful text editors in Linux. For this assignment, you are forced to use `vim`. For later assignments, you may use `vim` or other editors (such as `nano` or `joe`), but keep in mind that most VMs will not have a GUI. As such, you are encouraged to make yourself a "cheat sheet" for `vim`.

5.1 vim

`vi` is a classic, UNIX text editor. `vim` stands for "vi, improved", and is mostly backward compatible with `vi` but has many improvements. On some systems, `vi` is simply a link to `vim`. The main thing to know about `vim` is that it has several *modes*. You can read about these in `vim` by typing

```
:help vim-modes
```

while `vim` is in *normal mode* (`vim` starts in this mode). From most modes, you can get back to *normal mode* by pressing ESC twice (and usually, *once* is enough). To edit a file with `vim` from the command line, use:

```
vim file-you-want-to-edit
```

Similarly, you can open a file with `vim` in read–only mode with:

```
view file-you-want-to-view
```

Run the `vim` tutorial as `alice`, by editing the file `~/vim/tutor.txt` with `vim` and following the instructions[3]. You will be graded on the modifications you make to this file (follow the directions in the file); as such, you might want to make a backup copy of this file first in case you make mistakes.

6 Submitting your work

Login as `root`, and run `Turnin yourISUusername` to automatically submit your work. Check the `man` page for `Turnin` for more information.

[3] This is a shortened version of the tutorial you get by running `vimtutor`.