

CSCI 315 Lab 10 Exercise

22 April 2010

Objectives: In this lab you will carry out a few small experiments using the UNIX `ln` and `chmod` commands.

Background: Read the man pages for `ln` and `chmod` UNIX shell commands. The general form you will find for these commands is:

```
ln -s filename linkname
```

and

```
chmod val filename
```

For `chmod` the value `val` means a three-digit number or the funny '+' and '-' operands.

Problems: This week's lab has three short parts.

- **Directory Protection:** In this part you will investigate the effect of various protection patterns on the behaviour of directory access. Create a directory `Temp` in your current directory. Also create a small (one line) file `text` in the `Temp` directory. Use the `chmod` instruction to set permissions on the new text file to 600. You will carry out the following sequence of activities four times, first with protection on `Temp` set to 700, second time set to 500, third time set to 600, and the fourth time set to 100.

Here is the sequence of activities:

1. Check to see that there is a copy of the file `text` in the directory `Temp`. If not, set the permissions on `Temp` to 700 and copy the file `text` into `Temp`.
2. Change the permissions on `Temp` to one of the specified values and execute the following commands:

```
ls -l Temp
more Temp/text
rm Temp/text
ls -l Temp
```

3. Explain why the result of the command execution make sense in the context of the permissions set.

For each of the runs hand in the output from the execution of the designated commands and your explanation.

- **Links:** In this problem you will investigate setting links to files and the properties of the resulting links. First, change the permissions on the directory `Temp` back to 700.

Now change to a different directory (any of your directories will do as long as you have 700 access to the directory). **Carry out the following sequence of activities twice, once when making a symbolic (soft) link, the second time making a hard link.**

1. Copy `text` to the directory `Temp` if a copy of that file is not currently in the directory.
2. Set a link (soft or hard) back to the file `Temp/text`. Name the link `A`.
3. Execute the following commands, where `<path>` indicates the path to the directory `Temp`:

```
ls -l A
more A
rm <path>/Temp/text
ls -l A
more A
```

4. Explain why the output is as it is.

Hand in your outputs and explanations, along with the answer to the following question: What do these results imply about the nature of hard vs. soft links?

- In your directory, create a directory `test`. Carry out the following steps and **hand in answers to the associated questions**:

1. Move into the `test` directory. Issue the following command to create a hard link to the parent directory:

```
ln ../ newlink
```

What happens? Why do you think Unix does this?

2. Still in `test`, issue the following command to create a soft link to the parent directory:

```
ln -s ../ newlink
```

Now move back to the parent directory of `test`. Issue the command

```
ls -l test
```

What do you see?

Now issue the command

```
ls -R test
```

This command recursively lists the contents of a directory and its subdirectories. **We might expect it to go into an infinite loop. Does it? Explain what happens and why the system might work that way.**

3. Now issue the command

```
cd test/newlink/test/newlink
```

followed by the command

```
pwd
```

which gives the path of your current working directory. **What do you see? Why do you think Unix behaves this way? Describe a case in which listing a path with a symbolic link in this way makes sense.**

Handin: Hand in the requested output and answers to the questions, along with any output that you might want to support your answers to the third exercise by next Monday.