

Assignment1: Linux File System Permissions

In order to answer the following questions properly,

- 1- log in to “cs1”, and execute all commands in this lab. Record putty/ssh/mobaterm session log for exercise 3 and 4
- 2- Answer all the exercises in the separate answer worksheet (download from elearning)

What to submit to elearning:

One zip file (named <firstname>_<lastname>_<section>_assg1.zip containing:

- Mobaterm/putty transcript of exercise3.
- Mobaterm/putty transcript of exercise4.
- Filled answer worksheet.

Prerequisites

- Go through lecture material
- Go through/study Dr Min’s 4 videos in this week activity.
- Read the “Access Permission” chapter from Sobel’s Safari book. You will have to access the Safari section from the library website, and look for the book (see Syllabus for book name, and review the Video resource under class “Resources/Video Recordings” folder for how to access the Safari ebook online collection. The chapter direct link is :

https://learning.oreilly.com/library/view/a-practical-guide/9780133085129/ch04lev1sec6.html#ch04lev1sec6_ch04lev1sec6

Exercise 1: Conversion between symbolic mode and octal mode

For each nine-character symbolic permission, give the equivalent three-digit octal permission and the three-character symbolic permissions that apply to each of User/Owner, Group, and Other.

[REPLY in WORKSEET]

Row	Symbolic Mode	Octal Mode	Owner	Group	Other
1	rwxrW-r-x				
2	r---WX-W-				
3	-WX-WX--X				
4	r-X-W----				
5	rwxr-X---				

[END of REPLY in WORKSEET]

Exercise 2: Conversion between octal mode and symbolic mode

For each three-digit octal permission in the following table, give the equivalent nine-character symbolic permission and the three symbolic permissions that apply to each of Owner, Group, and Other

[REPLY in WORKSEET]

Row	Octal Mode	Symbolic Mode	Owner	Group	Other
1	000				
2	001				
3	020				
4	220				
5	300				
6	004				
7	050				
8	600				
9	007				
10	715				

[END of REPLY in WORKSEET]

Exercise 3 : Understanding directory permissions

This exercise chooses eight different permissions for a directory and then for each of the eight permissions attempts some basic commands such as `cd`, `touch`, `mkdir`, and `ls`. Different permissions will allow or deny each of these actions. You are to set the permissions, try each of the actions, and record whether or not it worked. Are the results as you would expect?

A) Execute the following three commands to create a testing directory:

```
{cslinux1:~} cd ; rm -rf assg1 ; mkdir -p assg1/top ; cd assg1
```

For each row of the table below, repeat these next two steps (B and C) that will set some permissions on the `top` directory and then try four commands to see if those commands work with those directory permissions:

B) From in the `assg1` directory, change the permission of the `top` directory using the `chmod` command given in the second column (Command line) of the table. Execute this `chmod` command in the `assg1` directory to set the permissions on the `top` subdirectory.

C) Next, for the row and permission value you just set in Step A, try to execute the four commands listed across the top of the table. For each of the four commands, record in the table whether or not the command line executes successfully. Enter in the table

PD (Permission Denied) or OK (for success)

The four commands to try are these (also listed across the top of the table):

1. `cd top` followed immediately by "`cd ..`" only if it worked (gave no error messages)

You need to "`cd ..`" to get back up to the `assg1` directory only if the `cd` command worked.

2. `touch top/file` followed immediately by "`rm top/file`" if it worked

3. `mkdir top/dir` followed immediately by "`rmdir top/dir`" if it worked

4. `ls -l top`

- You will carry out Steps B and C above (one `chmod` and four other commands) for each of the rows of the table below. Do the `chmod`, then try each of the four commands and record PD or OK
- Before you run each `chmod` command in the table below, ensure your current directory is `assg1`.
- If the `cd` into `top` works (no error), follow it immediately with "`cd ..`" to return up to the `assg1` directory again, otherwise you will be in the wrong directory for the next command in the table.
- If the `touch` works (no error), immediately remove the file you just created.
- If the `mkdir` works (no error), immediately remove the directory you just created.

[Capture transcript of mobaterm or putty here]

[REPLY in WORKSEET]

Table of OK and PD for different values of chmod permissions					
Row	Command line	<code>cd top</code>	<code>touch top/file</code>	<code>mkdir top/dir</code>	<code>ls -l top</code>
1	<code>chmod 000 top</code>				
2	<code>chmod 100 top</code>				
3	<code>chmod 200 top</code>				
4	<code>chmod 300 top</code>				
5	<code>chmod 400 top</code>				
6	<code>chmod 500 top</code>				
7	<code>chmod 600 top</code>				
8	<code>chmod 700 top</code>				

[END of REPLY in WORKSEET]

If you see any other error such as "no such file or directory", you are not in the correct `assg1` directory when you are running that command.

Exercise 4: Minimum Permissions for common file operations

In the table below, give in symbolic `rxw` form the minimum permissions an ordinary (non-root) user requires to successfully complete the commands listed in the left column below. "Minimum" means the least amount of `rxw` permissions needed.

How many of `rxw` can you take away and still perform the given command successfully? Test your guess to make sure you are correct, using commands 1-7 above. Do not forget to replace "`srcdir`" and "`targetdir`" with example directories in your home directory, "`srcfile`" and "`oldfile`" with an existing file in your home directory, and "`newfile`" with a name of a new file to create

Recall that some commands require permissions on the directory; some require permissions on the data(file); some require both.

1. {cslinux1:~} cp srcdir/srcfile targetdir/
2. {cslinux1:~} mv srcdir/srcfile targetdir
3. {cslinux1:~} ln srcdir/srcfile targetdir
4. {cslinux1:~} rm srcdir/srcfile
5. {cslinux1:~} cat srcdir/srcfile targetdir
6. {cslinux1:~} date >> dir/oldfile (modify an existing file)
7. {cslinux1:~} date >> dir/newfile (create a new file in an existing directory)

For commands 6 and 7 above, the directory is not a "source" directory since nothing is being read from it, and the file is not a "source" file since it is being written to (it is an output target). Use the first two columns in the table below to record permissions for the directory and the target file for 6 and 7 above.

[Capture transcript of mobaterm or putty here]

[REPLY in WORKSEET

MINIMUM rwx symbolic permissions needed to perform each of the commands			
Command line	On the source directory	On the source file	On the target directory
1. copy a file			
2. move a file			
3. link a file			
4. delete a file			N/A
5. read a file			N/A
6. modify an existing file			N/A
7. create a new file		N/A	N/A

[END of REPLY in WORKSEET]