



Lab 6: File Permissions

In this lab you will learn how to assign permissions to files and directories to provide a measure of security and privacy to your files on a multiuser system.

Preparation

Be sure to make the changes to your home directory asked for in Lab 5. This lab assumes the new names and directory structures.

Everything you need to do this lab can be found in the Lesson 7 materials on the CIS 90 website: <u>http://simms-teach.com/cis90calendar.php</u>. Review carefully all Lesson 7 slides, even those that may not have been covered in class.

Check the forum at: <u>http://oslab.cishawks.net/forum/</u> for any tips and updates related to this lab. The forum is also a good place to ask questions if you get stuck or help others.

If you would like some additional assistance come to the CIS Lab on campus where you can get help from instructors and student lab assistants: <u>http://webhawks.org/~cislab/</u>.

Procedure

Log on to Opus so that you have a command line shell at your service. Be sure you are in your home directory to start this lab. Using the **chgrp**, and **chmod** commands, you will modify the permissions on files and subdirectories in your home directory.

- From your home directory, do a long listing with the **Is -I** command. Who owns these files? To which group do they belong? How can you distinguish file entries from directory entries?
- 2. Do a long listing of the file, */home/rsimms/uhistory*. Who owns it? Can you move the file to your home directory? Why or why not? Can you copy the file to your home directory? Why or why not?
- 3. Now that you have copied the file *uhistory* to your home directory, who owns it? What are the permissions?

- Display the contents of the file *uhistory* on your screen. Now take away read permission using the command: **chmod -r uhistory** Try to display the contents of the file as you did above. Does it work?
- Now give read permission back but take away write permission: chmod 444 uhistory Verify the success of the above command.
- Take away execute (search) permission from the *misc* directory: chmod -x misc Do short and long listings of the *misc* directory using the ls and ls -l commands. Try to display the contents of the fruit file with the command: cat misc/fruit Try to change directories to *misc*.
- Give yourself back execute permission but take away read permission: chmod +x,-r misc Change your current directory to the *misc* directory. Try displaying the contents of the *misc* directory. Display the contents of the *fruit* file.
- 8. Change back to your home directory and set the *misc* directory to full permissions: chmod 777 misc
- 9. Set the permissions of your *poems* directory and its subdirectories so that you have full permissions as owner, but group and others have no write permission. Group and others should still have read and execute permission.
- 10. Set all ordinary files under the *poems* directory to be read only for user, group, and others. We want everyone to read our poetry, but no one should modify it, including ourselves.

See if you can do this using a minimum number of commands. (hint: use filename expansion characters).

- 11. Change the permissions of your *bin* directory so that you have full permission, group has read and execute, and all others have no permissions.
- 12. Set the executable files under bin to have the following permissions: r-xr-x-disallowing others outside the group from executing our commands.
- 13. Change the group id of the following directories: *class/labs*, and *class/exams* to be users: chgrp users class/labs class/exams
- 14. For the *class* directory set the permissions to 710.For the *labs* subdirectory, set permissions to 530.For the *exams* subdirectory, take away all permissions from group and others, leaving full permission for owner.

- 15. Make all ordinary files under *class/labs* and *class/exams* be: read-write for owner no permissions for group and no permissions for others.
- 16. For the *edits* directory, give yourself full permission, but no permission for group or others.For the ordinary files under *edits*, take away read permission from group, leaving everything else as it is.
- 17. Add read permission for everyone to all the files in the *misc* directory.
- 18. Run the **umask** command and note the number displayed.
- 19. Create an empty file called *old* and an empty directory called *olddir*: **touch old; mkdir olddir**
- 20. Run the **umask** command giving it the argument: 000 **umask 000**
- 21. Create an empty file called *new* and an empty directory called *newdir*: **touch new; mkdir newdir**
- 22. Look at the permissions of these four files you've created. Notice how they have changed. Can you figure out what **umask** is for?
- 23. Try setting the umask to 777 and making a *newer* file. To finish, set your umask back to 002 with the command: **umask 002**

Submittal

You have now finished the lab. To submit your work to be counted for this lab, you must run the **submit** command from your home directory.

Be sure to submit before the deadline. Remember late work is not accepted!

Grading Rubric (30 points total)

30 points for successfully completing all steps Less 1 point for each step not completed correctly