System Administration for Beginners

Week 3 Laboratory

February 13, 2007

The following exercises are best done with the assistance of a partner. Try exploring the `man` pages and communicating with others around you as you learn the commands and techniques.

1 Playing with Permissions

[**1**] Create an empty files and change its permissions to allow anybody to read and write to it. See if your lab partner can edit the file.

[**2**] Remove the "everyone" (except for the owner and group) permission to write to the file. Can your lab partner still edit the file?

[**3**] Remove the owner's ability to edit the file. Can you edit it?

[**4**] Create a new directory and create some files in that directory. What permissions must the directory have to allow other people to browse that directory? What permissions must the directroy have to allow people to change that directory but not browse its contents?

[**5**] Remove all permissions from the directory. Try copying files to it, moving files to it, and deleting it. What happens?

2 Learning Links

[**6**] Create an empty file and then create a symbolic link to that file. Edit the file using the symbolic link. Does the edit also show up on the original file?

[**7**] Delete the original empty file crated in Problem 6. What happens to the symbolic link? Create a new file with the same name as the file you created in Problem 6. What happens to the symbolic link?

[**8**] Symbolic links can also point to directories. Create a new directory and create a symbolic link to the directory. Try moving files to the symbolic link. List the files in the original directory.

[9] **EXTRA:** Explore hard links. Can you give an example of what it can be used for? What happens to the original file when you delete the hard link?

3 Following up with Filesystem Tools

[10] Create a directory, then create some files in that directory. Using tar, create an archive of that directory. List the contents of the tar archive.

[11] Go to http://httpd.apache.org and figure out how to download a copy of the Apache web server through the command-line. Choose a format that ends in either tar.gz or tar.bz2; the version does not matter. Figure out how to extract the file using either gzip or bzip2 and tar.

[12] Go into the directory that was created when you extracted the Apache file in the previous problem. Figure out how much disk space that directory uses. When finished, you can delete that directory.

4 Managing an Overflow of Output

[13] Come up with some commands that enerate enough output to scroll your terminal screen. Use a pipe to pipe the outpu to fhtese commands to a tool that you can use to scroll the output (more, less).

[14] Rather than piping the output using the commands used in the previous problem to more or less, redirect the output to a file.

[15] Find a command that generates error messages when input. Attempt to redirect the output of that command to a file. Are you successful?

[16] **EXTRA:** Lookup the concepts of `stdin`, `stdout` and `stderr`. How can you redirect each of these outputs to different files?

5 Secure Shell (SSH)

[17] Use SSH to login to the OCF (if you don't yet have an account, use SSH to login to your inst account). Execute some commands to see if SSH works. Logout from SSH when you are done.

[18] Use sftp to access the OCF or your inst account. How do you get your files with sftp?

[19] Use scp to copy files from your inst acount to the OCF (or to another directory in your inst account).

[20] **EXTRA:** SSH to a login on the OCF, run a screen session and try running a program that will generate continuous output. Detach from it, log out, then log in again. Attach the screen. What do you see?

6 Useful Utilities

[21] Use wget to retrieve a very large text file from the web (any large website's homepage will do). Use cat and grep to figure out how many times a certain word appears on the page.

[22] **EXTRA:** Using grep and ls, how would you get the number of files in a directory? **HINT:** you will want to get each file listing in a directory on it's own line.

[23] Figure out how to search the root of a filesystem (and all sub-directories) for the files that have the extension .log. Try the command for a few minutes (ignore any error messages). Since this places a lot of load on the servers, stop the command once you are satisfied with your work using ctrl-c.

[24] tail is a great tool for monitoring log files, especially ones that are changing. Run a command that generates a lot of output and redirect the output to a file. In another terminal, figure out how to use tail to track the file as it changes.

7 Realizing rsync

[25] **EXTRA:** Create a directory, and create some non-empty files in the directory. Use rsync to copy the files over to your OCF account (or to another directory on your inst account if you do not have an OCF account). Verify that the files were succesfully copied using SSH.

[26] **EXTRA:** Modify a couple of the files. Use rsync to update the copy on your OCF account (or the other directory on your inst account). Were the modifications successfully transferred?