

Lab2

Parts One and Two practice transferring files between systems. You will start each logged onto a linux system. **This means if you are using your laptop, you must ssh to hills, then to a linux system to start.**

This is a long lab exercise. You may need to finish it on your own. Do as much as you can in-class.

If you get stuck, go to the online PDF and see the Answers and Hints at the end of the exercise (parts Three and Four only).

Part One

You should be logged into your linux (not hills!) account and connected to your home directory when you start this exercise. This exercise practices **sftp**. It must be done before Part Two.

1. Create a directory in your home directory named **stest**. Connect to **stest**.
2. Create a simple text file named **sample** that contains a few lines of text. The file should include your name or some other identifying text. (Note: you should not have a file named **sample** in your home directory on either linux or hills. If you do, you will overwrite it now or in a later step.)
3. Change directory back to your home directory on linux.
4. Open an **sftp** connection to **hills.ccsf.edu**.
5. From within **sftp**, list the path of the current directory on hills and its contents. Then list the path of the current directory on linux and its contents.
6. Still in **sftp**, change the current directory on linux to your **stest** directory. List the contents of the current directory on linux.
7. Transfer the **sample** file from linux to hills. Still in **sftp**, list it on hills to be sure it is there. Then exit **sftp**. (Note that this transferred the file **~/stest/sample** on linux to **~/sample** on hills) What directory are you in on linux after you exit **sftp**?
8. Using **ssh**, connect to hills. Cat your **sample** file. Then delete your **sample** file from hills. Then exit back to linux.

Part Two

When you start this exercise, you should be logged onto linux and connected to your home directory. Your file **~/stest/sample** should be intact on linux. Before you begin, write down the IP address of your linux machine. It should be taped to the top or front of the machine, and should be a number like **147.144.23.43**. If you are logged in remotely, use the command **ip addr | grep inet** and look for an address that starts with **147.144.23**. (ignore the **/24** after it))

1. Staying in your home directory, transfer the file **~/stest/sample** from linux to your home directory on hills using **scp**. (If **scp** did not ask you for your hills password, you probably forgot the colon (**:**) after the hills domain name.)
2. **ssh** to hills and confirm that **sample** was transferred. Then delete it. **Remain on hills.**
3. Transfer the **stest** directory and all of its contents from linux to your home (current) directory on hills using **scp**. Remember to use the IP address of your linux system to do the transfer. (It actually does not matter which linux system's IP address you use, so long as you are using your personal linux account.)
4. Still on hills, check that the **stest** directory and its contents were transferred. Then delete **stest** and its contents.
5. Still on hills, transfer the directory **/pub/cs/gboyd/cs160a/wildcards** recursively to your home directory on linux.
6. Exit hills back to linux. Check that the **wildcards** directory was transferred. Then delete it and the

stest directory recursively.

7. log off of linux, but do not shut the system down.

Part Three

This part of the exercise set is challenging. Don't hesitate to refer to the Answers if you get stuck. You should be logged onto one of our linux workstations to do this part. (It is useful if you can log in twice.)

1. Connect to the **Lab2** directory beneath the public work area. List the current directory recursively, showing hidden files and including the mark to distinguish directories. Refer to the listing to do the remaining items. (If you are logged in twice, keep the recursive listing in one window and continue the exercise in the other window.) Use relative paths in all answers. Change directory only when instructed.
2. display the contents of the file named **process**
3. list the directory named **Downloads**. One of the files has spaces in its name. Can you see it?
4. list the file with spaces in its name (only) adding the option **-l** (that's lower-case L). Can you tell what information you are seeing? Can you tell how large the file is?
5. Note that two of the files in the **Downloads** directory are PDF files. If you are **on the console of your linux system**, you can start a PDF reader to read one. Start the PDF reader named **evince**, giving it the path to **file.pdf**. When you are done, quit the PDF reader to get back to the terminal (or just expose the terminal window and type control-C)
6. change directory to the **Downloads** directory. Can you display the contents of the file **process** from where you are? can you list the contents of the hidden directory recursively?
7. change directory to the directory named **homework**. Display the contents of the file **process** again. If you are on a local linux system, start the PDF reader **evince** on **file.pdf** again.
8. there is a file in the **Downloads** directory that does not have a valid extension. Find out what type of file it is by using the command **file**. Do you know what the output means and what type of data the file is?

Part Four

Connect to the **wildcards** directory beneath the public work area on *hills*. Then write **complete commands** using **ls** and wildcards to list the objects **in the current directory only** as follows:

1. the second character in the name is a vowel (a or e or i or o or u). Other characters can be anything.
2. the name begins with a space
3. the name is exactly three letters (upper- or lower- case)
4. the name ends with a character that is not a number
5. the name contains a punctuation character
6. the name contains a space character (blank only)
7. the name begins with a space character (either blank or tab)
8. the name begins or ends with an upper-case character.

Answers and Hints

Part Three

1. `ls -RFA` is the best choice for the listing
2. `cat backup/homework/asmt01/process`
3. `ls -F backup/Downloads` . The file **Use and Care Guide - 12828119.pdf** is the one with spaces in the name
4. referring to a file with spaces in its name is problematic. There are several solutions. The straightforward one is to quote the filename like this:
`ls -F backup/Downloads/'Use and Care Guide - 12828119.pdf'`
another possibility is to use filename completion. Type
`ls -F backup/Downloads/UseTAB` where TAB is the tab character. If the filename is unique (which it is), the shell will fill in the filename. (We will cover tab completion next week.)
the last possibility is to use a wildcard like this
`ls -F backup/Downloads/Use*`
5. `evince backup/Downloads/file.pdf`
6. `cd backup/Downloads`
`cat ../homework/asmt01/process`
`ls -F ../../.config`
7. `cd ../homework`
`evince ../Downloads/file.pdf`
8. `file ../Downloads/lab1a.noext`
The file is a tar archive. Somewhat analogous to a zip file, it is a file that contains copies of other files. Notice that the extension is not required on a linux system.

Part Four

`cd /pub/cs/gboyd/cs160a/wildcards`

1. `ls -dF ?[aeiou]*`
2. `ls -dF \ *`
3. `ls -dF [[:alpha:]][[:alpha:]][[:alpha:]]`
4. `ls -dF *![[:digit:]]`
5. `ls -dF *[:punct:]*`
6. `ls -df '*' '*'`
7. `ls -dF [[:space:]]*`
8. This takes two wildcards - one for 'ends with' and one for 'begins with'. Note that the use of the international 'upper case' class gives some non-ASCII characters.
`ls -dF [[:upper:]]* *[:upper:]`