Exercises-sftp

Description

In this exercise set you will practice transferring files between systems using the secure copy (**scp**) and secure **ftp** (**sftp**) commands.

Although both **scp** and **sftp** can be used to transfer files, each has its own strengths and weaknesses:

sftp [user@]host

This command initiates a connection to the remote system **host**. where host is a domain or IP address. If **sftp** successfully logs in as **user**, it interactively accepts commands from you at the keyboard to tell it how to find the files to transfer and how to transfer them. For the duration of the transfer *you are simultaneously connected to two systems*. The system on which you executed the **sftp** command is referred to as the *local* system. The system indicated by **host** is referred to as the *remote* system. (The **[ ]** around the **user@** indicate that part of the command syntax is optional. If the **user@** portion is omitted, **sftp** will attempt to log you into **host** using your current login.)

Many **sftp** commands *look like* filesystem commands. They are not. They are **sftp** commands with similar names. Do not attempt to use standard filesystem options, or to use complex forms of the commands.

- **cd**, **ls**, **mkdir**, **rmdir**, **pwd** - function similarly to the filesystem commands. They operate on the remote system. Similar commands with the prefix **l** (lower-case L) operate on the local system. For example:
  - **lls** - list the current directory on the local system
  - **lpwd** - print the path of the working directory on the local system.
- The actual transfer commands are **get** and **put**. **get x** transfers the file **x** from the current directory on the remote system to the current directory on the local system. **put x** transfers the file **x** from the current directory on the local system to the current directory on the remote system. ***Don't get these confused.***
- To log off of the remote system and close **sftp** you can use **quit**, **exit**, or **bye**.

scp *source destination*

If you know where the piece of data you want is located and where you want to put it, you can use **scp**. **scp** is a standard command-line program. **scp**

- opens a connection to an indicated system. (You will be prompted for your password.)
- transfers one unit of information between the systems. (actually it can transfer multiple units, but we will keep it simple for now.)
- terminates the connection.

scp has one added advantage. If the **-r** option is used, the unit of information transferred can be a directory (and all of its contents, recursively). **sftp cannot recursively transfer a directory.**

Example:

You are logged onto your home system, and connected to your current directory. You want to transfer the file **dir1/dir2/file1** from your home system and place the copy in the existing directory **~/work** on *hills.ccsf.edu*, where your login is **you1**. The following **sftp** session would accomplish this. (There are explanatory comments to the right of each command.)

bash$ sftp you1@hills.ccsf.edu # initiate the connection, logging in as **you01**.

< you are prompted for **you1**'s password on hills.ccsf.edu here. Then you are connected to the home directory of **you1** on hills.ccsf.edu >

sftp> cd work # change your remote directory to **~/work**

sftp> lcd dir1/dir2 # change your local directory to **dir1/dir2** (relative to where you were)

sftp> put file1 # perform the transfer

sftp > exit # terminate the connection

The **scp** command to transfer this single file is more complex, but is much more compact:

scp dir1/dir2/file1 you1@hills.ccsf.edu:work

In the command above, the source is on the local system, so it does not have a username (**user@**) nor a **host**. The destination is on a remote system, so it requires a **host**. Since the login on the remote system is different than the current login on the local system, **user@** is required. The host specification must be followed by a colon (**:**). Relative paths are interpreted relative to the current directory on the local system (so **./dir1/dir2/file1**) or the home directory on the remote system (so **~/work** on *hills*).

Preparation

ssh to *hills* and connect to the class public directory. In this directory there is a subdirectory **sftpex**. Without connecting to it, list it recursively and **draw its structure**. (***You will have to refer to its structure in the exercises that follow!***) Stay connected to the class public directory.

Find the IP address of any linux system that you want to work on. **ssh** to that system to ensure it is available. Then log off of it. (You are still logged onto *hills*.) We will use the linux system **147.144.23.47**

You will now copy the entire **sftpex** structure to your home directory on linux. Use the following command:

scp –r sftpex ipaddr:

where **ipaddr** is the IP address of the linux system you chose. **scp** will ask you for your linux password. It will then complete the copy.

ssh to the linux system and list (recursively) your new **sftpex** directory. Then exit back to *hills*.

You are now ready to begin the exercise.

Part One (using sftp)

1. Connect to your home directory on *hills*. Create a new directory named **temp** and connect to it. You will use this as your work area for the duration of this exercise. Create a placeholder file **myfile** in the current directory.
2. Initiate an **sftp** connection to your linux system **ipaddr**
3. On the remote system (linux), connect to the **sftpex** directory. Then use **sftp** to show the path to the remote working directory.
4. List the remote directory.
5. Change the remote working directory to **asmt01**. Then display its path and list its contents
6. Display the path to the local working directory (to ensure you are in the **temp** directory). Then transfer the **typescript** file from the remote to the local directory.
7. Transfer the **address** file from the remote to the local directory, naming it **my_addresses**.
8. List the local directory.

9. Change the working remote directory to the **sftpex** directory and verify where you are.
10. Create a new directory named **xxx** in the local directory. Then connect to it. Check your work by displaying the path to the local working directory.
11. Copy the file named **scp** (yes, the file is named **scp**) from the remote system to the local working directory.
12. List the local working directory.
13. Transfer the file you made (**myfile**) to the remote working directory (where is **myfile**?)
14. Exit **sftp** and check the contents of your **temp** directory recursively.
15. Delete your **temp** directory and all its contents.

Part Two (using scp)

1. Connect to your home directory on *hills*. Create a new directory named **temp** and connect to it. You will use this as your work area for the duration of this exercise. Create a placeholder file **myfile** in the current directory.
2. Using **scp**, copy the single file **myfile** from your current directory on hills to a linux machine, placing it in your home directory. (Note: if you execute an **scp** command and it is silent, without giving you an error message or asking you for a password, you probably forgot the colon after the host name of the remote system. In this case, **scp** becomes the local copy command **cp**.)
3. Change directory to your home directory, which should be the parent directory of **temp**. Using **scp**, copy the single file named **scp** from your linux system to your directory **temp**. List the **temp** directory to ensure it was successful.
4. Change directory to the **temp** directory. Using a single **scp** command, transfer the entire directory **sftpex** from your linux system to the current directory. List it recursively to ensure it was successful.
5. Again using a single **scp** command, transfer the entire directory **notes** from beneath your newly-copied **sftpex** directory to your home directory on linux.
6. **ssh** to linux and examine your home directory to ensure the file **myfile** and the directory **notes** were transferred. Then delete **myfile**, and the **notes** and **sftpex** directories.
7. Exit back to hills and delete the **temp** directory and its contents.

Answers

Preparation

```
bash$ scp -r sftpex 147.144.23.47:
Password:
typescript                                    100%   47     0.1KB/s   00:00
address                                       100%   44     0.0KB/s   00:00
sftp                                          100%   16KB  15.6KB/s   00:00
scp                                           100% 7320     7.2KB/s   00:00
permissions                                   100%   20     0.0KB/s   00:00
FSIntro                                       100%   16     0.0KB/s   00:00
bash$
```

Part One

1.
```
-bash$ mkdir temp
-bash$ cd temp
-bash$ touch myfile
```

2. **-bash$ sftp 147.144.23.47**
   Connecting to 147.144.23.47...
   Password:
   sftp>
3. **sftp> cd sftpex**
   sftp> pwd
   Remote working directory: /home/gboyd/sftpex
4. **sftp> ls**
   asmt01 notes
5. **sftp> cd asmt01**
   sftp> pwd
   Remote working directory: /home/gboyd/sftpex/asmt01
   sftp> ls
   address   typescript
6. **sftp> lpwd**
   Local working directory: /users/gboyd/temp
   sftp> get typescript
   Fetching /home/gboyd/sftpex/asmt01/typescript to typescript
   /home/gboyd/sftpex/asmt01/types 100%   47   0.1KB/s   0.1KB/s   00:00
   Max throughput: 0.1KB/s
7. **sftp> get address my_addresses**
   Fetching /home/gboyd/sftpex/asmt01/address to my_addresses
   /home/gboyd/sftpex/asmt01/addre 100%   44   0.0KB/s   0.0KB/s   00:00
   Max throughput: 0.1KB/s
8. **sftp> lls**
   my_addresses myfile   typescript
9. **sftp> cd ..**
   sftp> pwd
   Remote working directory: /home/gboyd/sftpex
10. **sftp> lmkdir xxx**
    sftp> lcd xxx
    sftp> lpwd
    Local working directory: /users/gboyd/temp/xxx
11. **sftp> get notes/remote/scp**
    Fetching /home/gboyd/sftpex/notes/remote/scp to scp
    /home/gboyd/sftpex/notes/remote 100% 7320   7.2KB/s   7.2KB/s   00:00
    Max throughput: 7.2KB/s
12. **sftp> lls**
    scp
13. **sftp> put ../myfile**

```
   Uploading ../myfile to /home/gboyd/sftpex/myfile
   ../myfile                     100%    0     0.0KB/s   0.0KB/s   00:00
   Max throughput:   0.0KB/s
14. sftp> exit
   -bash$
15. -bash$ pwd
   /users/gboyd/temp
   -bash$ ls -R
   my_addresses  myfile        typescript    xxx

   ./xxx:
   scp
16. -bash$ cd -..
   -bash$ rm -r temp
```

Part Two

The output of the scp commands has been omitted for brevity.

```
1. -bash$ mkdir temp
   -bash$ cd temp
   -bash$ touch myfile
2. -bash$ scp myfile 147.144.23.47:
3. -bash$ cd
   -bash$ scp 147.144.23.47:sftpex/notes/remote/scp temp
   -bash$ ls temp
   myfile  scp
4. -bash$ cd temp
   -bash$ scp -r 147.144.23.47:sftpex .
   -bash$ ls -R sftpex
   asmt01  myfile  notes
   sftpex/asmt01:
   address     typescript
   sftpex/notes:
   FSIntro      permissions  remote
   sftpex/notes/remote:
   scp   sftp
5. -bash$ scp -r sftpex/notes 147.144.23.47:
```