

CS015 Fall 2018

Working from Home: Using SCP and SFTP to Transfer Files

Once you've set up remote access on your laptop (see Working from Home: Setting up SSH on your Laptop), you can login to a Sunlab Linux machine from the comfort of your own home. If you're unfamiliar with using a terminal to navigate and work within a Unix file system, we have a useful cheat sheet here.

Copying Files with scp

If you're working from home for CS015, you'll need to copy files from your laptop to the CS file system. There's a couple ways to do this remotely, but the simplest is to use the `scp` (secure copy) command. The `scp` command is analogous to the `cp` (copy) command; the difference is that with `scp`, you're copying files between hosts, rather than just from one location in your file system to another. It uses `ssh` to login to the remote host (the Sunlab Linux box) and copy files to or from the local host (your laptop).

The usage of regular old `cp` in a Linux terminal looks something like this:

```
cp ~/source.java ~/course/cs0150
```

This will copy the source `.java` file to your `cs0150` directory.

Now, say the `source.java` file is on the desktop of my MacBook and I want to copy it to my `cs0150` directory on my Brown CS account. On my Mac, the path to (location of) this source file is `/Users/<username>/Desktop/source.java` . On the CS file system, the path to the target directory is still `~/course/cs0150` ; however, because this directory is on the *remote host*, I have to add a prefix that uses my CS login, `jcarber`. I open up a Terminal window on my laptop and type:

```
scp /Users/jcarberry/Desktop/source.txt jcarber@ssh.cs.brown.edu :~/course/cs0150
```

This prefix should look familiar: it's the argument you give to the `ssh` command to login from your laptop, `@ssh.cs.brown.edu` . Note that there's a colon between this prefix and the actual file path (but not a space). You may have to enter your `ssh` password (the same password you use to login to computers in the CIT) for the copying to begin, and you'll be able to watch its progress as each file is processed (in this case, it's just the one).

So now you know how to copy a file from your laptop to the CS file system (copying a file from the CS file system to your laptop is the same—just reverse the order of the two files). A more useful example is to copy a whole directory—or folder—from your laptop to the CS file system. Let's say I have a nicely organized file system on my laptop, with a folder for AndyBot at the location `/Users/jcarberry/Desktop/school/CS/AndyBot` . Before I hand in the project, I need to make sure it runs on a Sunlab computer, so I need to copy the contents of this folder to the location `~/course/cs0150/AndyBot` on my CS file system. To copy a folder and all the files within it, use the `-r` (recursive) flag with the `scp` command:

```
scp -r /Users/jcarberry/Desktop/school/CS/AndyBot jcarber@ssh.cs.brown.edu :~/course/cs0150/
```

You'll notice that I didn't explicitly include the folder name `AndyBot` in the remote host location. Because the `cs0150` directory already exists in my CS file system (it should in yours too!), the `scp` command will put a copy of my `AndyBot` directory within `~/course/cs0150/` . If I already have a `~/course/cs0150/AndyBot` directory, any files within it that are also in the folder on my laptop will be overwritten. Always double-check that your source and target folders are in the correct order before you hit enter! If you switch them, you could accidentally overwrite all the progress you've made on your laptop with a previous version you have on the CS file system!

Once I copy the contents of `AndyBot` from my laptop to the CS file system, I need to test to make sure `AndyBot` runs properly before turning it in. I can either go to the CIT and test it on a computer there, but if I want to do it all from home... In a Terminal window on my laptop, I login to a Sunlab Linux box: `ssh -Y jcarber@ssh.cs.brown.edu` . Once you are logged in with `ssh`, try running your program. If it works without errors, you're ready to hand it in!

To run the `handin` script, type: `cs0150_handin AndyBot` into Terminal and press Enter.

Transferring Files using SFTP

Another way to transfer files from your local computer to your Brown CS file system is to use a service called SFTP , which is right in your terminal. To use SFTP , I need to first open up Terminal on my (local) computer. Then, I type the following command and press return:

```
sftp jcarber@ssh.cs.brown.edu
```

If prompted to do so, I will enter my CS account passphrase. Now, I am logged into both my local computer and a computer in the Sunlab. Most standard commands used to navigate (`cd` , `pwd` , etc.) still work and will navigate around your CS account. Add an `'l'` (lowercase L) at the beginning of commands navigate around your own computer (`l cd` , `lpwd` , etc.).

If I want to move files from my computer to the CS file system, I can use the following command:

```
sftp> put <location of file on your computer> <destination on CIT file system>
```

So, if I wanted to move source.java from my local hard drive to my cs0150 directory in the Sunlab (as in the first scp example), I would type the following into the terminal:

```
sftp> put /Users/jcarberry/Desktop/source.java course/cs0150
```

The “ sftp> ” will already be at the beginning of the line, you don’t need to type it in. If I want to instead move files from the CS file system to my computer, I can use the get command:

```
sftp> get <location of file on CIT file system> <destination on your computer>
```

So, if I wanted to copy source.java from my ssh (let’s imagine it’s in the course/cs0150 directory) to my local Desktop, I would type the following into the terminal:

```
sftp> get course/cs0150/source.java /Users/jcarberry/Desktop/school/CS
```

Also, it is important to note that the paths used to access files or set destinations depend on your local and remote working directories (or the folders you are currently “in”). Instead of writing long and confusing paths, you could use cd and lcd (local change directory) to change the working directories on your local and remote system. This way, you wouldn’t have to write the entire path to access the files and set destinations. For example, to complete the same get command as above, I could change my local working directory to CS (by typing sftp> lcd /Users/jcarberry/Desktop/school/CS) and change my remote working directory to cs0150 (by typing sftp> cd course/cs0150). Then I would type the following into the command line:

```
sftp> get source.java
```

Since I’m already “in” the target directory (CS), I don’t need to set a destination (sftp assumes that you want to copy source.java into your current local working directory).

There is no recursive call (-r) in sftp like there is with the scp command, so to move multiple files, we have to use the * in our put and get commands.

Let’s say I want to copy all of the (.java) files from my Desktop/school/CS/AndyBot local directory to the remote directory course/cs0150/AndyBot . To accomplish this, I would type the following:

```
sftp> put Users/jcarberry/Desktop/school/CS/AndyBot/*.java course/cs0150/AndyBot
```

*.java represents all of the .java files in whatever directory is specified. If I use just * instead of *.java , I can copy all files in a given local directory to a remote directory:

```
sftp> put Users/jcarberry/Desktop/school/CS/AndyBot/* course/cs0150/AndyBot
```

This can submit PDF files for DQs, homeworks, and any other files you might need!

Using the * works the same with the get command as it does for put. So, to copy all files from a remote directory (source) to a local directory (target), I would type:

```
sftp> get path/to/source/* path/to/target
```

Note that using * includes all files in a directory, but does not include files in sub-directories like -r does.

For more about sftp commands, type `sftp> ?` into the command line, or look online!