1 Introduction

While CS17 does not require a laptop, those who do have one may find it very helpful to work from home or the Blue Room rather than walking to the CIT whenever you need to work on an assignment. This guide gives a few different ways of doing that, depending on your operating system. Regardless of the method, there are two steps to successfully working remotely. The first is setting up SSH; the second is using SSH to access your files on the department machines, and there are many different ways to do this second step.

2 Terminology

**Definition:** *Remote Computer* - This is the computer you will be working *from*. This is your laptop or Desktop in your home (or if you’re really ambitious, your phone).

**Definition:** *Host Computer* - This is the computer you will be *connecting to*. In this case, it is some machine in the CIT where all of your files are stored.

**Definition:** *SSH* - SSH stands for Secure Shell. It is a way of securely logging into one computer from another.
3 Setting up SSH

For Mac, Linux, Windows, or even Android, you can set up SSH by following the instructions here. Please follow these instructions very closely: make sure to run commands on the correct computer.

Note: You will need to be in the CIT to set this up.

If you need help during this part of the set-up process, you can always talk to a Sunlab consultant: they sit at the computer closest to the door in the Sunlab. Your friendly TAs have also helped people set things up in the past, and will be happy to help at hours.

4 Using SSH

Phew. Now that SSH is all set up, there are a multitude of different ways to use it. Depending on your operating system, there are a few ways of doing so that are recommended by various CS courses.

4.1 Mac

4.1.1 The Options

Mac has several options for utilizing SSH, each of which has advantages and disadvantages. The more common ones are:

- **FastX** - This is one of the more popular solutions, as it is easy to understand and easy to use. It is the option recommended by the department.

  FastX is an application you get for your Mac. Once you set it up, when you open it, FastX connects to a department machine and gives you the entire graphical environment of a Linux machine. It will look as though your Mac has turned into a department machine! (It hasn’t, don’t worry.)

  Advantages:
  - This is the CS department’s recommended option, so it will be pretty easy to get help if you’re having trouble with it.
  - Easy to set up, easy to use.
  - It’s practically the same experience as working on one of the department machines, except you have the screen for the department machine on your computer.
  - You will only have to get used to one new interface: the department machine’s interface
  - You can work with a graphical environment
  - You can work within the graphical environment in which all of your files are stored.

  Disadvantages:
  - There are some stability problems with FastX, so it does not always work perfectly. You may have to resort to SSH at times.
– You will be working on the department’s graphical system, which you will have to get used to or try to customize.

– Your space is limited by what memory you are allocated by the department. You will sometimes have to run terminal commands to clear your cache, just so you can log in.

– Any applications you like to use on your Mac, you won’t be able to use on the Linux machine, unless you download it yourself.

– You don’t have administrative privileges, and will face the same restrictions as you do on the department machines when attempting to install new software.

– This can be slow, especially over shakier connections, because you are constantly loading an image - it’s like streaming video, though a little more optimized/efficient.

• **Cyberduck** - Cyberduck gives you an easy way to both transfer files between the remote computer and the host computer, and an easy way to edit files remotely using applications on your own computer like Sublime Text or DrRacket.

  Advantages:

  – Relatively easy to set up.
  – Relatively easy to use, there’s a lot you can do but none of it requires knowledge of the Terminal!
  – You can edit files with local applications like DrRacket or Sublime Text.
  – Whenever you save a file, it is uploaded very quickly back to the CS department, and Cyberduck tells you when it’s done.

  Disadvantages:

  – Can’t interact with your files via command line.
  – You will still need to SSH in to run the cs017_handin script.

• **SSH with XQuartz (or X11 on older Macs)** - This requires very little setup, and while it is often clunky it is one of the better options. Using XQuartz and the -Y command when you SSH will allow you to run commands like evince or subl while SSH’d. The XQuartz application will open with the application in question.

  Advantages:

  – Relatively easy to set up.
  – Very easy to use.
  – Not many issues are going to come up.

  Disadvantages

  – This is often incredibly slow. It’s basically a slower version of FastX
  – You can’t use applications on your Mac to edit the files, so you’re limited to applications on the department machines and opening them from the Terminal.
  – This can also be hard on your computer.
• **Fuse and SSHFS** - This requires some extra setup, but has some nice advantages. SSHFS is a way of mounting a filesystem onto your own computer. What does this mean practically? You can have the entire CS department filesystem in a folder on your Desktop (so you can cd /Users/<name>/Desktop/gpfs/main/course/cs0170/...).

Advantages:

- You see and interact with all of your CS files in Mac’s Finder application. This makes it easier and more intuitive to create/delete/rename/email files.
- You can easily open any files in your favorite Mac applications.
- Any changes you make in directory are made in real time on the department machines - no extra work to send your code to the department machines!
- This is pretty easy to use once you have it set up!
- Since the file system is mounted, if you know how to use the Terminal, you can interact with both your own computer and the CS department computers at the same time, making copying files in between very easy. This is the main advantage of Fuse over the other options, so if you’re inexperienced with Terminal/Bash, Cyberduck is a better option.

Disadvantages:

- This requires some knowledge of bash/working in the Terminal, so if you’re new to the Terminal this is not the best choice.
- You will still need to SSH in to run the cs017_handin script. For advanced users, even if you add the cs17 bin (/gpfs/main/course/cs0170/bin) to your PATH, the binaries will not work due to issues with relative versus absolute paths.
- There is a good bit of setup (~15-30 minutes) to making this work effectively.

• **SSH and SCP** This option is only good if you’re skilled with Bash and/or like typing in the same commands over and over, but very often this is what CS17 students default to. We recommend using FastX, Cyberduck, XQuartz, or even Fuse over this option! However, if you don’t want to install any extra applications on your Mac, this is the way to go. With this option, you would do all your work on your Mac, then use the SCP command (already installed) to copy it over to the department machines. While this is a simple in concept method, it is very tedious and requires a lot of repeated typing unless you set up some shortcuts which requires knowledge of Bash.

• **SSH** The go-to choice if you love working with inline text editors like Vim or Nano, but otherwise vastly limits your ability to interact with the department machines (for example, viewing PDFs). For this option, you SSH into the department machine (ssh `username`@ssh.cs.brown.edu), cd into the directory you want to work in, and use vim, nano, emacs, etc. to edit your files. Even if you have a lot of experience with these text editors, keep in mind this option may be aggravating for partners who don’t know them as well when you’re working on group projects.

• **Fugu** Fugu is a somewhat easier to use version of Cyberduck. We don’t recommend it over Cyberduck because it requires you change some security settings on your computer (nothing dangerous). If you’re interested in trying it out, you can find the package [here](#).
So what do I do!? There’s too many options!

Again, the department recommends that you use FastX, as it’s a similar experience to sitting down in the Sunlab and working on one of the machines. If you want to edit files using your favorite Mac applications, or plan on transferring files back and forth between the Sunlab and your device then use Cyberduck. XQuartz is a choice to have as a backup if you’re willing to deal with the slowness. Fuse is good if you’re experienced with Bash. Just scp’ing or ssh’ing is only effective for people with lots of bash experience, but as it is already set up it’s not a bad option once you get used to it.

4.1.2 How to set it up

- FastX: Follow the directions [here].
- Cyberduck: Follow the directions [here]. Once you have Cyberduck installed, you can simply open it, connect, and navigate/edit your CS directory.
- SSH with XQuartz: Follow the instructions at the bottom of this page.
- Fuse: This is one of the more involved setups.
  1. On the remote computer (your computer), you will need to go to this link. Download FUSE for macOS and SSHFS, the links on the right sidebar.
  2. Run the packages, leave any settings as the default.
  3. Go to your Mac terminal and run `mkdir ~/bcs`.

At this point, you’re technically all set but it will be rather inconvenient to use right now since the commands are long.

The commands you will need are:

- `sshfs -o defer_permissions <user-name>@ssh.cs.brown.edu:/ ~/bcs`
  The pieces of this command are: sshfs: This is a version of SSH that, coupled with Fuse, mounts the Host computer onto the remote computer; -o defer_permissions: This will make sure you have the write group permissions as a CS17 student (or as as CS17 TA if you’re reading this!). If, as is possible, you get an error saying defer_permissions is not a valid argument, simply use `sshfs <user-name>@ssh.cs.brown.edu:/ ~/bcs`. The last two arguments are the target computer and the location to mount the department machine, the bcs folder on your computer.
- `umount -f ~/bcs`
  This unmounts the department machine from your computer. The -f "flag" stands for "force," and makes sure the system is unmounted. This will not work if you still have applications with files in the department system still running, or if you have a Terminal window that is still cd’d into a CS directory.

To save time in the future, you can google ‘bash aliases’ to set up shortcuts for these long commands. These would make using Fuse significantly easier, but will still not enable you to use commands like `cs017_handin`.

- SSH and scp: The syntax for SSH is `ssh <user-name>@ssh.cs.brown.edu`. The syntax for scp is:
scp </path/to/local/file> <user-name>@ssh.cs.brown.edu:/path/to/remote/destination

You can download DrRacket and edit files on your local computer, then scp them over. After that, ssh in to submit your assignment.

- **Just SSH**: See above.

### 4.2 Windows

There are a few options for Windows users. The ones that are stable right now are FastX, Winscp, Cyberduck, Putty, and SSH. Brief instructions for each are below.

#### 4.2.1 The options

- **Cyberduck** - Cyberduck gives you an easy way to both transfer files between the remote computer and the host computer, and an easy way to edit files remotely using applications on your own computer like Sublime Text or DrRacket. One disadvantage is that you can’t use commands like `cs017_handin`, so you’ll need to SSH in to do so. See the Mac section for a more in depth explanation of Cyberduck.

- **WinScp** - This works almost exactly the same as Cyberduck. It is recommended by the CS department, but does require a more setup than Cyberduck (not a big difference!).

- **FastX** - This allows you to work on your computer as if it were a department machine, running all commands in the given window as you would if you were in the Sunlab. Easy to set up, easy to use, and well-documented on the department website. See the pros and cons from the Mac section for more information.

- **Putty** - This is the standard for SSH on Windows. This allows you to view graphical applications that are being run on the host machine (so you are limited to Linux programs). However, the interface is clunky compared to the first two options and the setup can be frustrating.

- **SSH** - Once you have SSH set up on your computer from the first step, you can run `ssh <user-name>@ssh.cs.brown.edu` in your Command Prompt to log into the department machines. This gives you limited access to the department machine, and is not recommended; see the Mac section for more details.

#### 4.2.2 How to set it up

- **FastX**:
  1. Follow the directions [here](#).
  2. Install the application and start a new connection to begin! Make sure to select SFTP (SSH) for the correct type of transfer.

- **Cyberduck**:
1. Go to [this link](#) and download Cyberduck.

2. Install the application and start a new connection to begin! Make sure to select SFTP (SSH) for the correct type of transfer.

- WinScp: Follow the instructions [here](#).
- Putty: Follow the instructions [here](#).

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS17 document by filling out the anonymous feedback form: [http://cs.brown.edu/courses/cs017/feedback](http://cs.brown.edu/courses/cs017/feedback)