1 Introduction

2 Remote Access via Cornell VPN

3 Remote Access via X2Go
   3.1 Installing X2go on Your Laptop/Workstation
   3.2 Configuring X2go to Access ecelinux Servers
   3.3 Logging into ecelinux Servers with X2go
   3.4 Logging out of ecelinux Servers with X2go
   3.5 Troubleshooting Remote Access via X2go

4 Opening a Terminal and Sourcing the Course Setup Script

5 Other Remote Access Options
1. Introduction

All of the programming assignments for this course will be completed by remotely logging into a set of ecelinux servers. The ecelinux servers all run the Red Hat Enterprise Linux 7 operating system, and they all use an identical setup. You do not need to do anything special to create an ecelinux account. You will be using your NetID and Cornell password to login, and an ecelinux account will be automatically created for you when you first login. Any student enrolled in any ECE class should automatically be granted access to the ecelinux servers. Having said this, if you cannot log into the ecelinux servers please reach out to the course staff for assistance.

Later tutorials will discuss how to use the Linux development environment and the Git distributed version control system. In this tutorial, we focus on how to setup remote access to the ecelinux servers using the Cornell VPN and X2go. We will also briefly introduce how to open a terminal through X2go and source the course setup script.

2. Remote Access via Cornell VPN

If you are logging into the ecelinux servers from on campus (i.e., using the Cornell wired or wireless network), then you do not need to enable the Cornell virtual private network (VPN). However, if you are off campus, then you will need to enable the Cornell VPN whenever you want to log into the ecelinux servers. The VPN provides very secure access to all on-campus network resources. More information about the Cornell VPN is available here:

- [https://it.cornell.edu/cuvpn](https://it.cornell.edu/cuvpn)

Simply follow the instructions at the following link to install the Cisco VPN software for the appropriate operating system you use on your laptop/workstation:

- [https://it.cornell.edu/articles/topics/2605/all/822](https://it.cornell.edu/articles/topics/2605/all/822)

Once the Cornell VPN is installed, then connect to the Cornell VPN by following these instructions and using your Cornell NetID and password:

- [https://it.cornell.edu/articles/topics/2605/all/823](https://it.cornell.edu/articles/topics/2605/all/823)

The Cornell VPN uses the Cisco AnyConnect Client, and Figure 1 illustrates running the client.
3. Remote Access via X2Go

We will be using X2go as the primary way to log into the ecelinux servers. X2go is an application that you install on your local laptop or workstation. Then you setup a session, and you can use X2go for log into the ecelinux servers. X2go will provide students with a Linux virtual desktop on the ecelinux server and enable students to open terminals for working at the command line, use text editors for developing code, and run other programs for analyzing and/or evaluating your programming assignments. More information about X2go is available here:

- https://wiki.x2go.org

3.1. Installing X2go on Your Laptop/Workstation

On Mac OS X, you will need to install XQuartz first. Go to this page, download the DMG, and then install XQuartz.

- https://www.xquartz.org

After installing, you should reboot your laptop/workstation and then try starting XQuartz. The very first time you might need to right click on the XQuartz application and explicitly choose open from the pop-up menu. This tells Mac OS X that you really do want to run this application even though you downloaded it from the internet. After doing this once, you can open XQuartz just by double clicking it, and/or X2go will start it automatically if necessary.

The next step for both Windows and Mac OS X is to download and install the X2go client for the appropriate operating system you use on your laptop/workstation. Here are direct links for the Windows and Mac OS X operating systems:

- Windows: http://code.x2go.org/releases/X2GoClient_latest_mswin32-setup.exe
- Mac OS X: http://code.x2go.org/releases/X2GoClient_latest_macosx_10_13.dmg

For more information about the installation process, for versions of the X2go client that can work on older operating systems, or for versions of the X2go client that work on Linux please see the X2go documentation:

- https://wiki.x2go.org/doku.php/download:start

3.2. Configuring X2go to Access ecelinux Servers

Start by opening the X2Go Client. On Mac OS X, the very first time you might need to right click on the X2go application and explicitly choose open from the pop-up menu. This tells Mac OS X that you really do want to run this application even though you downloaded it from the internet. After doing this once, you can open X2go just by double clicking it, and/or X2go will start it automatically if necessary.

If it is your first time opening the client, a window will appear to set up a new session. Configure the options on the Session and Media tabs as shown in Figure 2, making sure you carefully change every circled field. In the Session tab, for Session name enter ecelinux; for Host enter ecelinux.ece.cornell.edu; for Login enter your NetID; and choose XFCE under Session type. In the Media tab, make sure Enable sound support and Client side printing support are not checked. Once you have configured all settings, click OK.
Figure 2: Configuring X2go to Access ecelinux Servers
3.3. **Logging into ecelinux Servers with X2go**

First, make sure you have started the Cornell VPN as discussed in Section 2. You must be have the Cornell VPN running before attempting to use X2go to access the ecelinux servers. Then enter `ecelinux` at the prompt as shown in Figure 3, press enter/return, and enter your NetID password. If you are asked to trust a certificate for the ecelinux servers, you must click yes. If you receive any kind of warnings on Windows related to firewall configuration you should just cancel that warning.

Once you have successfully logged into the ecelinux servers, you should see the Linux virtual desktop shown in Figure 4. Note that the Linux virtual desktop is running on the ‘ecelinux’ server, so anything you do in the Linux virtual desktop is actually happening on the server and not on your local laptop/workstation.

![Figure 3: Logging into ecelinux Servers with X2go](image-url)
Figure 4: Virtual Desktop on ecelinux

Figure 5: Log Out Dialog Box on ecelinux
3.4. Logging out of `ecelinux` Servers with X2go

When you are finished working on the `ecelinux` servers you need to explicitly log out. Choose `Applications > Log Out` from the `Applications` menu in the upper-left corner of the Linux virtual desktop. Make sure `Save session for future logins` is not checked. Then click on `Log out` to cleanly log out from the `ecelinux` servers (see Figure 5). Obviously, you should save all of you work frequently and before you log out of the `ecelinux` servers.

3.5. Troubleshooting Remote Access via X2go

The above instructions should work assuming you have a default `.bashrc` file on the `ecelinux` servers. However, if you have taken other courses that use the `ecelinux` servers, then your `.bashrc` may be setup in such a way that it prevents X2go from connecting to the `ecelinux` servers. The key is that X2go does not like any output printed to the console when you log into the `ecelinux` servers. So if anything in your `.bashrc` uses `echo` or `print` statements that can cause connections to hang.

If you have some experience using SSH and the Linux command line, you can potentially SSH into the `ecelinux` servers to fix this issue. On Windows, you can use MobaXterm or Putty. On Mac OS X, you can just use the default terminal program. You need to use `ssh netid@ecelinux.ece.cornell.edu`. Once connected, you can comment out everything in your `.bashrc` file, and try remote access via X2go again. If you are now able to connect, you can try uncommenting parts of your `.bashrc` file to find what setup actions are the culprit.

If you are new to all of this just reach out to the course staff, and we can help you fix this issue and get started with X2go in no time!

4. Opening a Terminal and Sourcing the Course Setup Script

The next tutorial provides a detailed look at using the Linux development environment. However, if you just want to get started, and you already know a bit about working at the Linux command line, then there is an absolutely critical step you must be aware of. **You must source the course setup script!** The course setup script configures everything so you have the right environment to work on the programming assignments.

To see how the course setup script works, first we open a new terminal on the Linux virtual desktop to provide access to the Linux command line. You can open a new terminal by choosing `Applications > Terminal Emulator` from the `Applications` menu in the upper-left corner of the Linux virtual desktop. Figure 6 shows a terminal opened on the Linux virtual desktop.

Once you have opened a terminal, the very first thing you need to do after logging into the `ecelinux` servers is source the course setup script. This will ensure your environment is setup with everything you need for working on the programming assignments. Enter the following command on the command line:

```bash
% source setup-ece2400.sh
```

Note that you do not need to enter `%` character. In a tutorial like this, the `%` simply indicates what you should type at the command line. You should now see `ECE 2400` in your prompt which means your environment is setup for the course. Figure 7 shows what your prompt should look like if you have sourced the course setup script.

It can be tedious to always remember to source the course setup script. You can also use auto setup which will automatically source the course setup for you when you open a terminal. Note that if
Figure 6: Terminal Opened on Linux Virtual Desktop on ecelinux

Figure 7: Terminal After Sourcing Course Setup Script
the environment for ECE 2400 conflicts with the environment required by a different course then you will need to manually source the setup script when you are working on this course. Enter the following command on the command line to use auto setup:

```
% source setup-ece2400.sh --enable-auto-setup
```

Now close the terminal using the X icon in the upper right-hand corner of the terminal window. Reopen a new terminal window. You should see `ECE 2400` in the prompt meaning your environment is automatically setup for the course. If at anytime you need to disable auto setup you can use the following command:

```
% source setup-ece2400.sh --disable-auto-setup
```

Again, if for any reason running the setup script prevents you from using tools for another course, you cannot use the auto setup. You will need to run the setup script manually every time you want to work on a programming assignment for this course.

5. Other Remote Access Options

Note that there are many other ways to access the `ecelinux` servers. Students can login using SSH and MobaXterm from Windows or using the native terminal on Mac OS X. Students can also setup their own text editors such as Sublime or Visual Student Codel to connect remotely to the `ecelinux` servers over SSH. More advanced students are welcome to work directly on their own workstations without logging into an `ecelinux` server. This is not too much work if your are using a UNIX-like system (e.g., Mac OS X, Linux). It might require the student to install the GNU C/C++ compiler and some other packages. While students are free to use whatever method they find most convenient, the course staff cannot offer support for any other method besides X2go. Please remember that all programming assignments must work on the `ecelinux` workstations and servers and also on TravisCI, since that is where the course staff will be doing the assessment.

Acknowledgments

This tutorial was developed for the ECE 2400 Computer Systems Programming courses at Cornell University by Thomas Tapen and Christopher Batten.