INSTALLING AN SSH / X-WINDOW ENVIRONMENT ON A WINDOWS PC

Nicholas Fitzkee
Mississippi State University
June 2, 2023

Installing Cygwin

Being able to run graphical programs remotely over the Internet is an extremely powerful ability, and one that is not straightforward using current Windows-based operating systems. While multiple programs have been implemented for secure remote windowing, the most powerful, flexible, and free system is Cygwin. We will use Cygwin for both SSH terminal access as well as sharing X11 window applications.

X11 is a program that allows you to run Windowed programs (like Firefox, PDF viewers, text editors, etc.) on another server. The program itself runs on the remote UNIX or Linux server, but the graphics are displayed on your computer. It's a very powerful feature that makes UNIX a very convenient operating system. For example, an expensive software package (like MatLab) could be installed on the UNIX server, and you could run it in a window remotely just as if you were sitting at the server yourself. To install X11, you should download the following file from the Cygwin site:

https://cygwin.com/setup-x86_64.exe (for 64-bit systems; 32-bit is no longer supported)

This is an installer for Cygwin, although it's much more complicated than a typical installer. This is because Cygwin has many, many optional packages that can be installed to augment the installation. We will only be installing a few of these features, but you can always install more later. Indeed, you should keep the installer handy, because you can re-run it to re-install packages that have been updated. There are two drawbacks to this type of installer: As mentioned, complexity is one. The second is that it will take substantially longer than a typical installer, because it will download files from the Internet.

The figures on the following page show the typical settings I use for installation of Cygwin, based on the description below. You can use them as a guide for settings that you should use. Briefly:

1. To begin, run the installer, and when prompted, select "Install from Internet." Your root directory should be "C:\Cygwin." You can put it on another drive if you like (e.g. "D:\Cygwin"), but don't try to put it under Program Files, etc. You should install it for all users.

2. The "Local Package Directory" can be anything that's convenient. I put mine in "My Documents\Downloads\".

3. You can leave the download settings alone unless you know they need to be changed. When prompted for a download site, though, scroll down until do you find some academic or government mirrors. I typically use mirror.cs.vt.edu (Virginia Tech). You can select multiple mirrors by holding down the CTRL key.
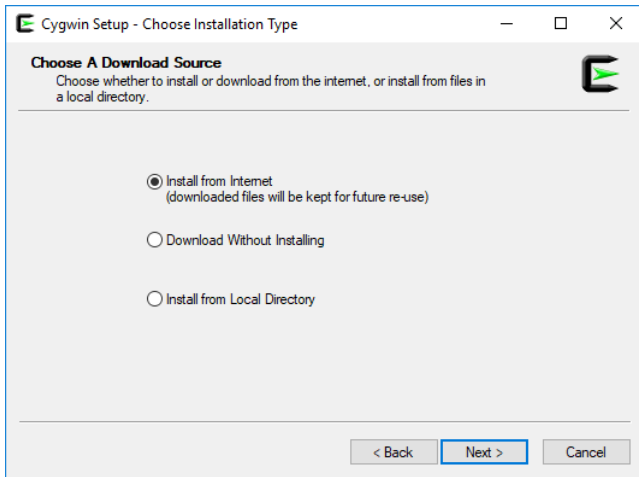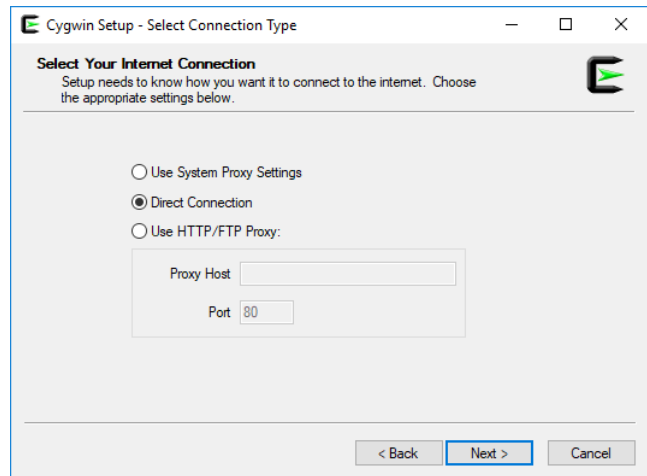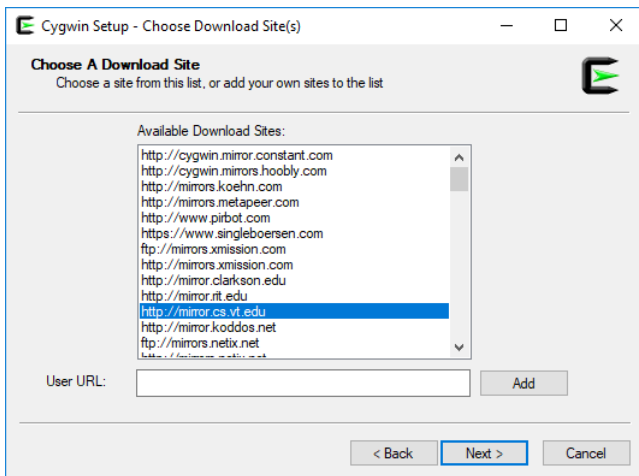
Screen 1 – Installation Options (use defaults)
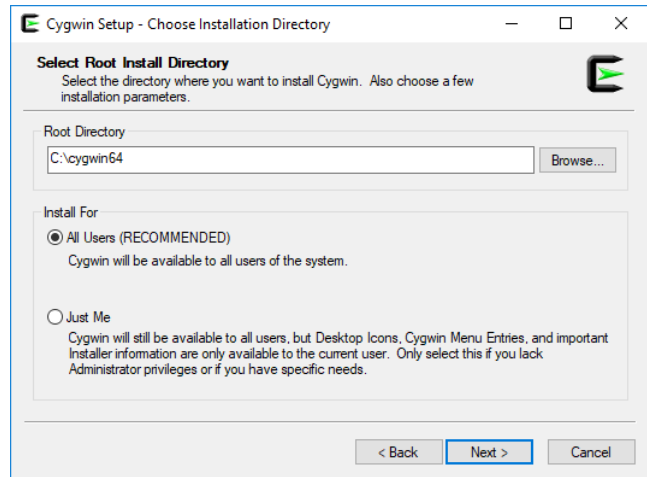
Screen 2 – Installation Directory (use defaults)
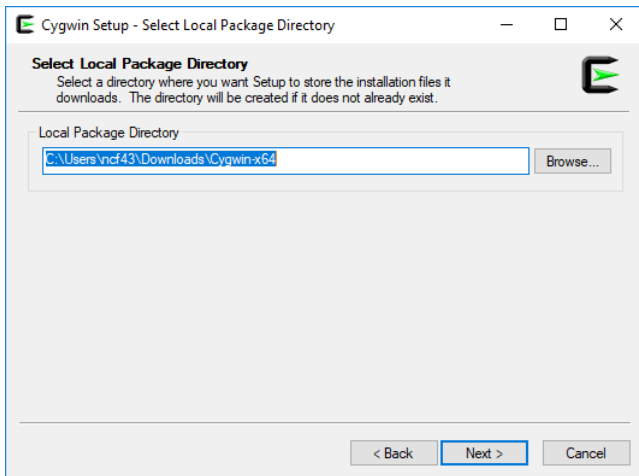
Screen 3 – Local Package Storage (use defaults)
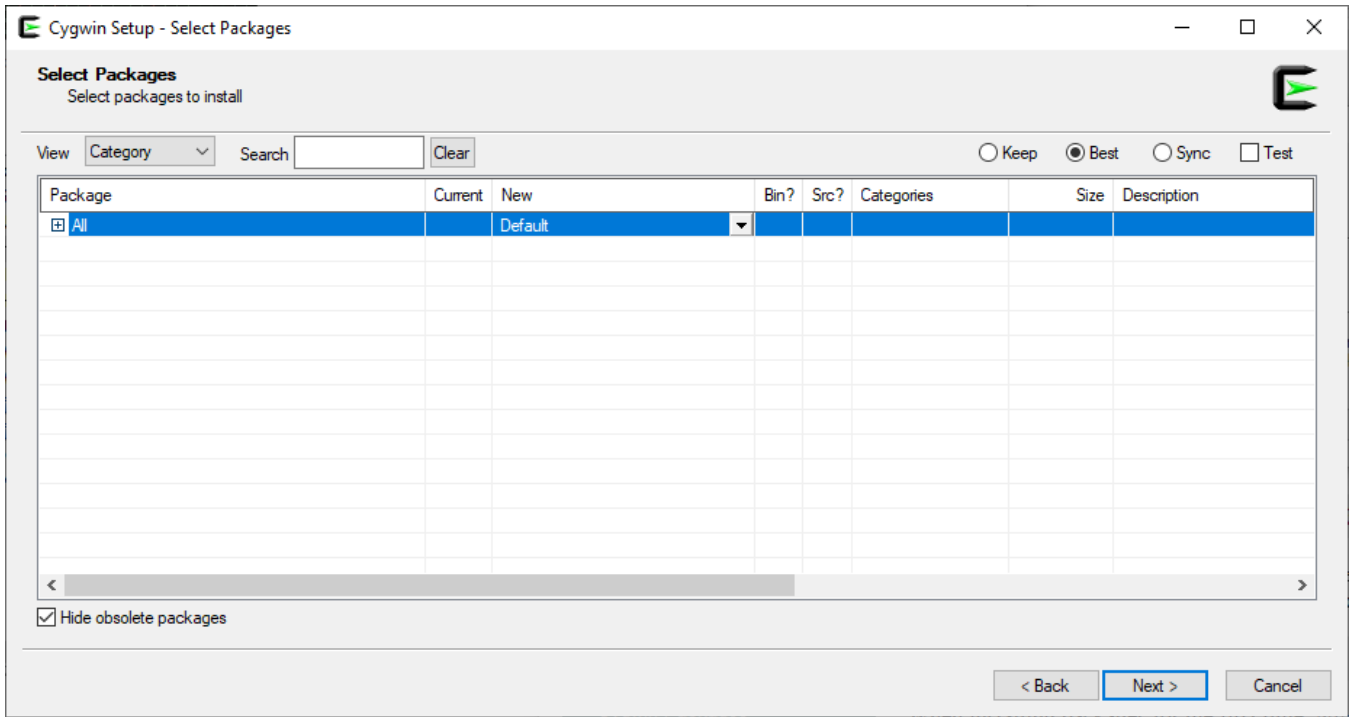
Screen 4 – Internet (use "Direct Connection")

Screen 5 – Mirror Selection (select http://mirror.cs.vt.edu/)

After selecting the mirror, the installer will download some files and bring up the package selection window. Next we'll pick that packages we need to install X11 and SSH.

Selecting Packages

The next screen you see will look something like this:

This is the package selection screen. Using this interface, we will select the packages that are needed to run X11 and SSH. The interface is smart, so while there are a lot of options, it will recognize if one package depends on another, and it will adjust your package selection accordingly to make sure all dependencies are met. We will actually take advantage of this, selecting only the bare minimum of packages and allowing the installer to figure out what else is needed.

The packages we need to select are given below (Table 1).

Table 1: Packages Needed for Basic X11 Installation

X11	Net	Others (Optional)
- xlogo - xinit - xorg-scripts - xorg-server - xorg-server-common - All packages starting with "xorg-x11-fonts" - xterm	- inetutils - openssh	- gnuplot-base (Math) - grace (Math) - python2 (Python) - python3 (Python) - R (Science)
Libs	Util	Editors
libmcrypt4	ccrypt	- vim - nano - emacs

To mark a package for installation, you should expand the items under "All," and the expand the subcategories to bring up additional packages. For example, expanding "All" and "Editors" gives me the following screen:
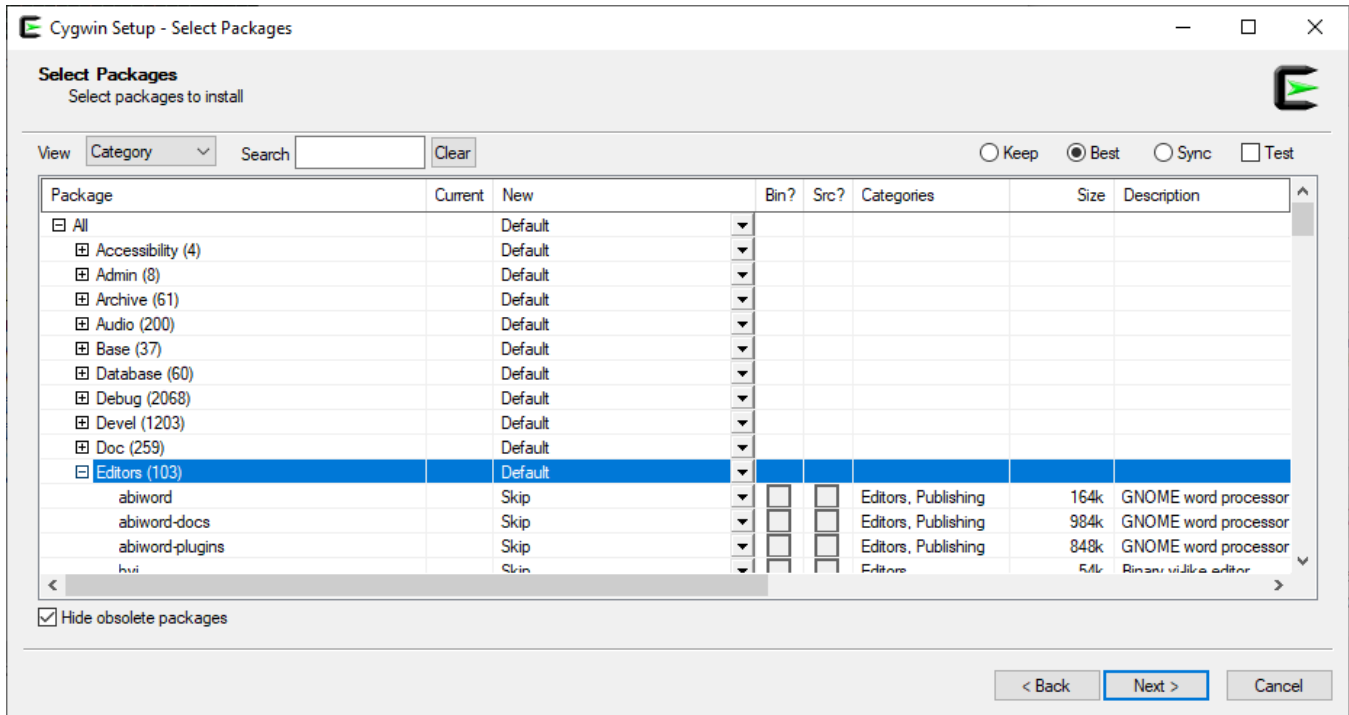

Notice how the individual packages (e.g. abiword) have several columns:

- **Current:** This column shows the current version of a package installed on your computer. If no version is installed, it is blank.

- **New:** This column is a dropdown selection that allows you to change the status of a package. If a package is installed, you can "Uninstall" it. If you want to install the package, you can select the version – typically the newest version is best. Finally, you can select "Skip" to make no changes to the package.

- **Bin?:** This column allows you to select whether binary files are installed (as opposed to compiling from source code - "Src?"). It will automatically become checked when you select a version in the "New" column. No other action is needed for this column

- Other columns describe the purpose of the package, give a description, and note the disk space used by each package.

For this tutorial, you will:

1. Go through each package in Table 1,
2. Select in the installer window, and
3. Install the most recent version by clicking the dropdown box in the "New" column.

It may be faster to use the "Search" box to identify individual packages. An example of this is shown below. Here, I searched for "xorg," and I selected installation for the "xorg-scripts," "xorg-server," and "xorg-server-common" packages.
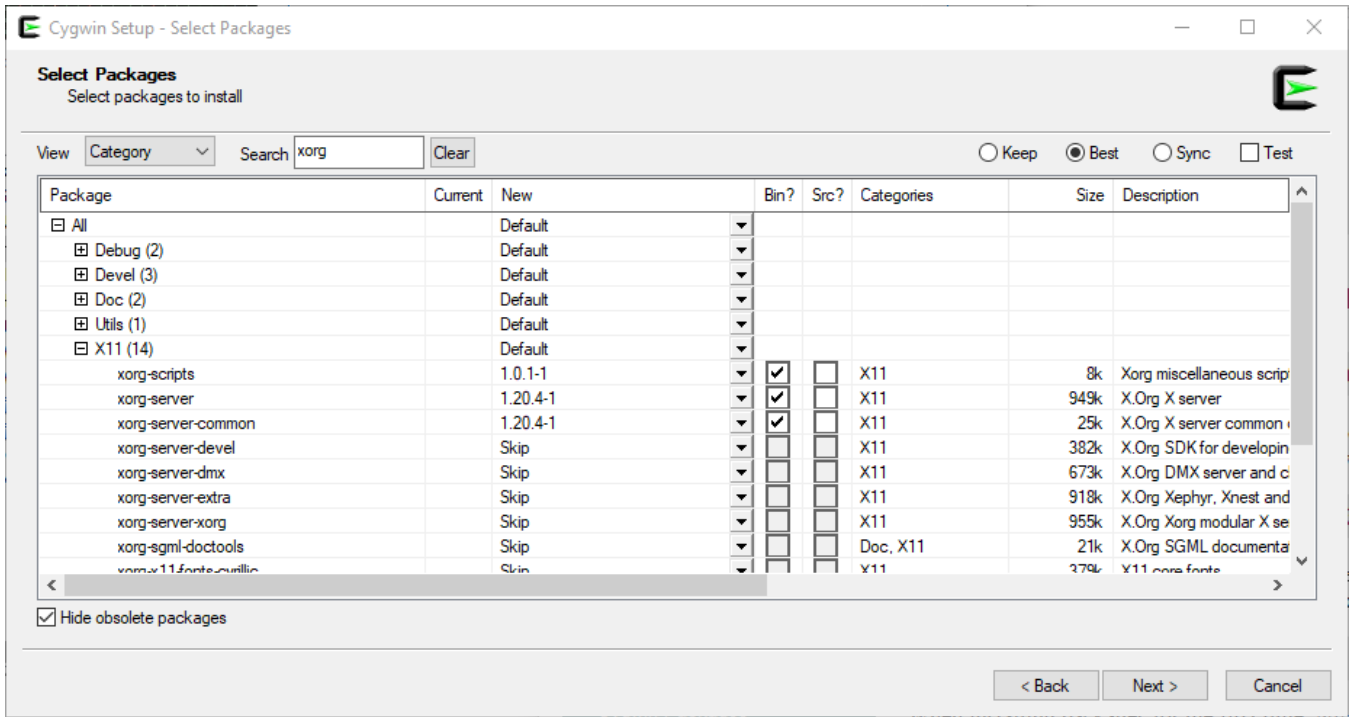

You will need to do this for all of the packages listed in the table above. *Leave all other packages at their default values. Do not uninstall any of the pre-selected packages.*
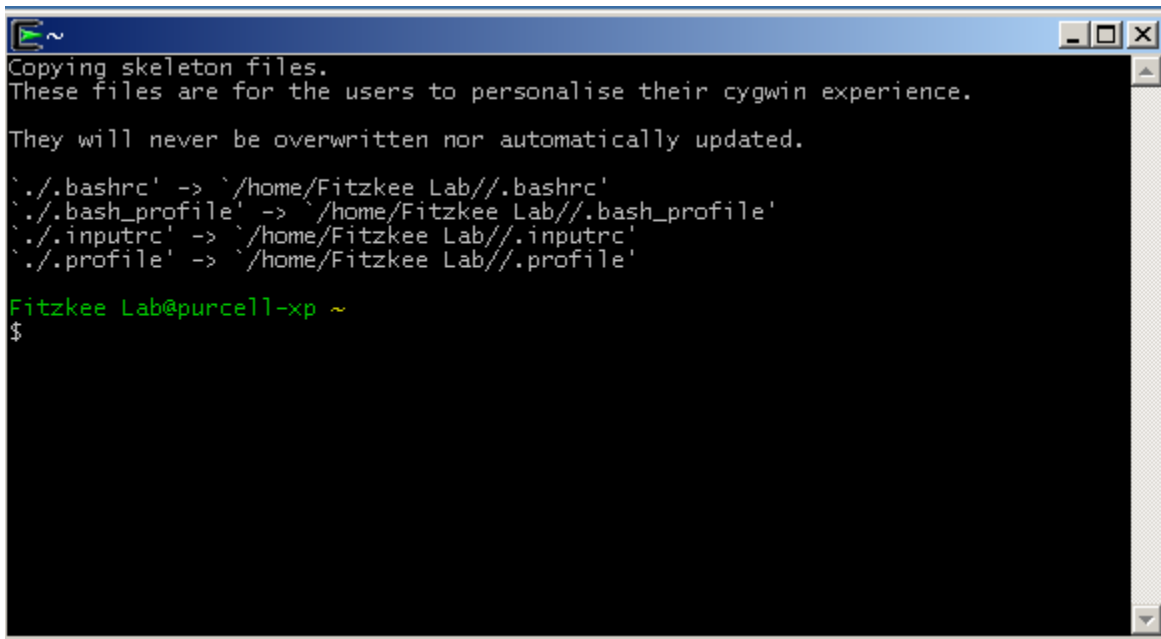
When you have selected the packages in Table 1, click "Next." You will be prompted with additional recommended packages. Just click "Next" again, which will select all recommended packages.

At this point the installer will download the files needed for X11 and SSH. It will take several minutes to download, and then several more minutes to complete the installation.

When the installer is finished, it will ask you to create icons on the desktop and in the Start Menu. Check both of these options, as we will need them.

5
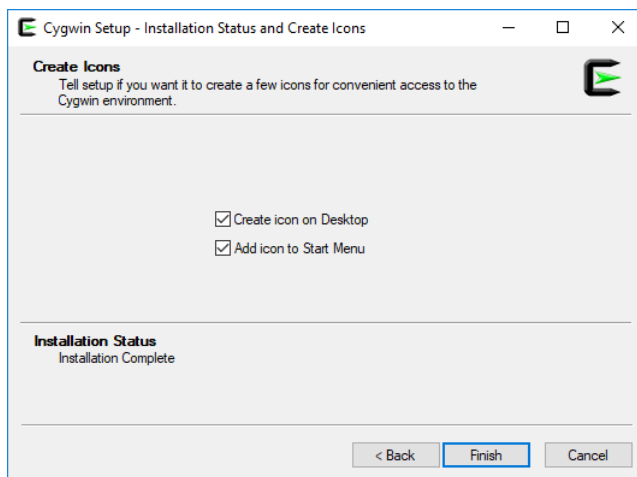
Once the icon appears on your desktop "Cygwin Terminal," double click it to run it. We don't need to do anything here, but we want the program to create the initial directories and files on its "first time" start up (see below). Your screen will look slightly different.
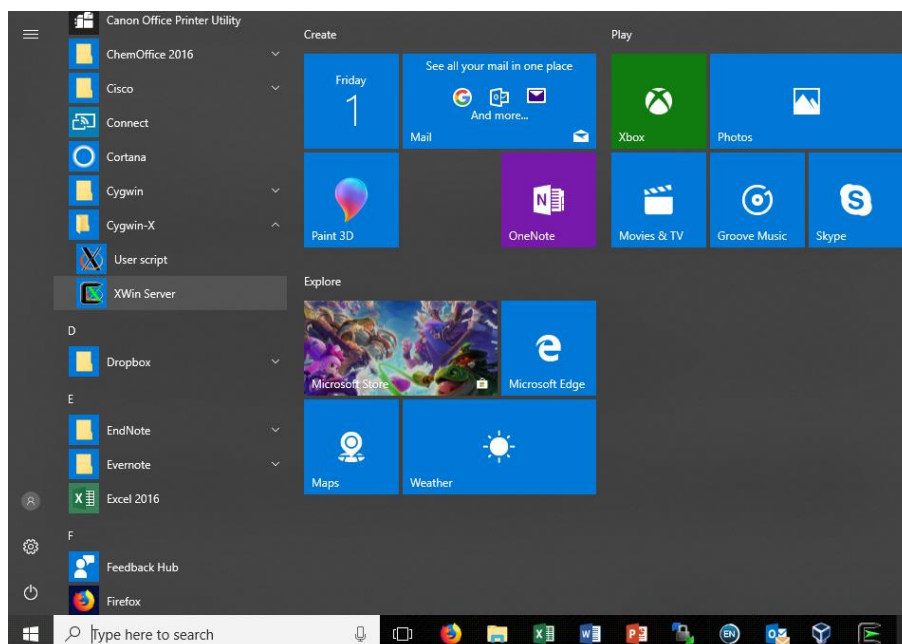

Now that you've done that, you can close the window – nothing else is needed. At this point, you can close the Cygwin terminal. We won't be using it anymore in this tutorial, but you may want to keep the icon around for later.

Starting Cygwin-X

Finally, under your start menu, find "Cygwin-X" and start the program "XWin Server."

If all is well, the program will start and you will see two new icons appear in the system tray. If you are asked whether to block the program by your firewall, choose to "Unblock," otherwise the program won't work properly. The icons look like this:

When you right-click on the right-most icon (the green X), you'll be presented with a series of options. You will want to start an Xterm window:
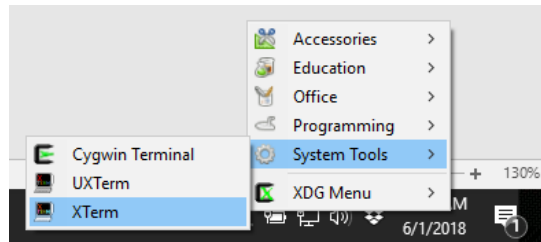
When you start an xterm window, you'll be presented with another command line interface, shown on the next page. This is the interface we will use to run SSH etc. In this example, I have used the "VT Fonts → Huge" option to make my text much bigger and easier to read.
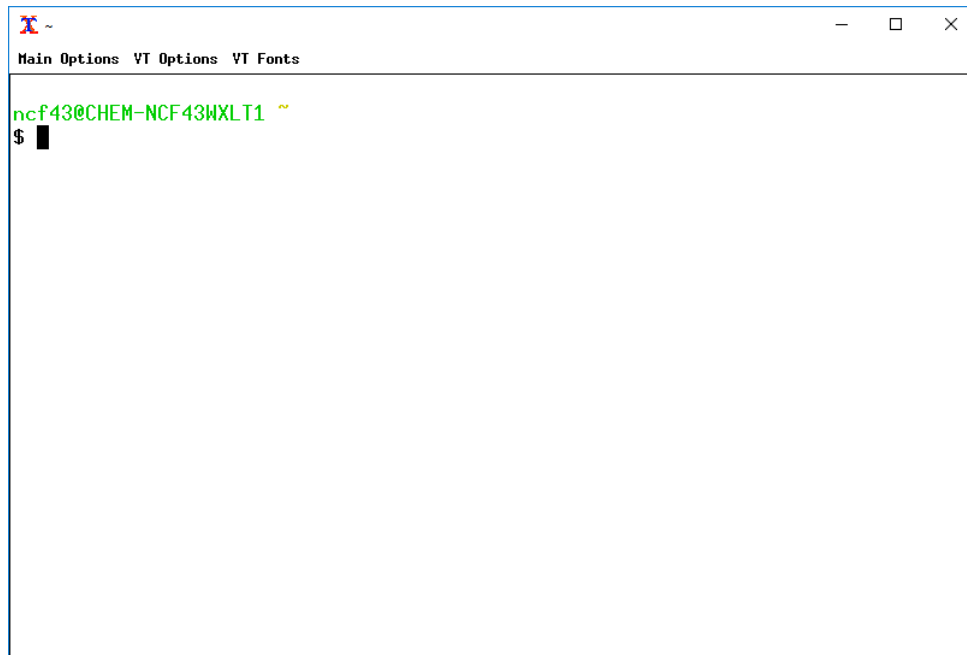
To test the X11 installation, simply type "xlogo" at the command prompt and hit return. If everything is working, you will see a graphic like this appear on your screen:
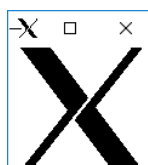
This is a test logo image, and it means that the xlogo client program (running from your local computer) is able to communicate with the X11 *server* program to display graphics on your screen.

The next test is to connect remotely to my Linux system using SSH and see if you can run a *remote* client program while displaying graphics on your screen.
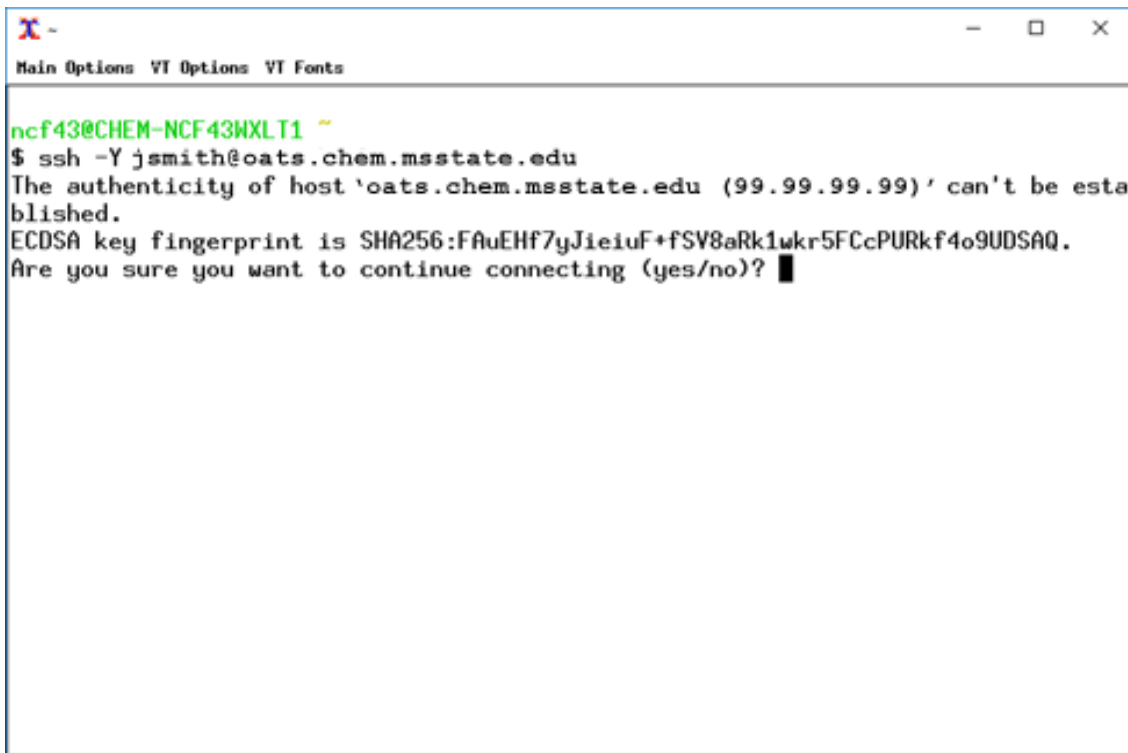
Connecting to Other Computers Using SSH

At the start of this tutorial, you were given a username and password for the Linux system. We will connect to this system using the SSH program that was installed using the Cygwin Installer above. At the Xterm command prompt, type the following command and press enter. Make sure you include the "-Y" flag, as it's required for using X11 graphics via SSH.

```
ssh -Y (username)@(server)
```

In the command above, (username) is your username (e.g. jsmith or js1252), and (server) is the hostname of the Linux server. If my username is jsmith and the server is oats.chem.msstate.edu, I would type:

```
ssh -Y jsmith@oats.msstate.edu
```

The first time you run this command, you will be asked to accept the key for the server. You should type "yes" (see below). You won't be asked about this again provided the server configuration doesn't change. Note that your screen will look slightly different based on the server and username.

After you type "yes," you'll be able to enter your password (don't worry, it won't display as you type), and eventually you'll have a command line on the *remote* system. You can tell this by the new command line (note the jsmith@oats at the very last line).

We're almost done! Just one more thing to test, and that's running X11 programs remotely.

Remote SSH and X11 Test: Xeyes

Another simple test program that can be used to check X11 is called "xeyes". If everything is configured properly, you will see a set of eyes appear when you type this command on the remote command line. The eyes will follow your mouse around, as shown below:
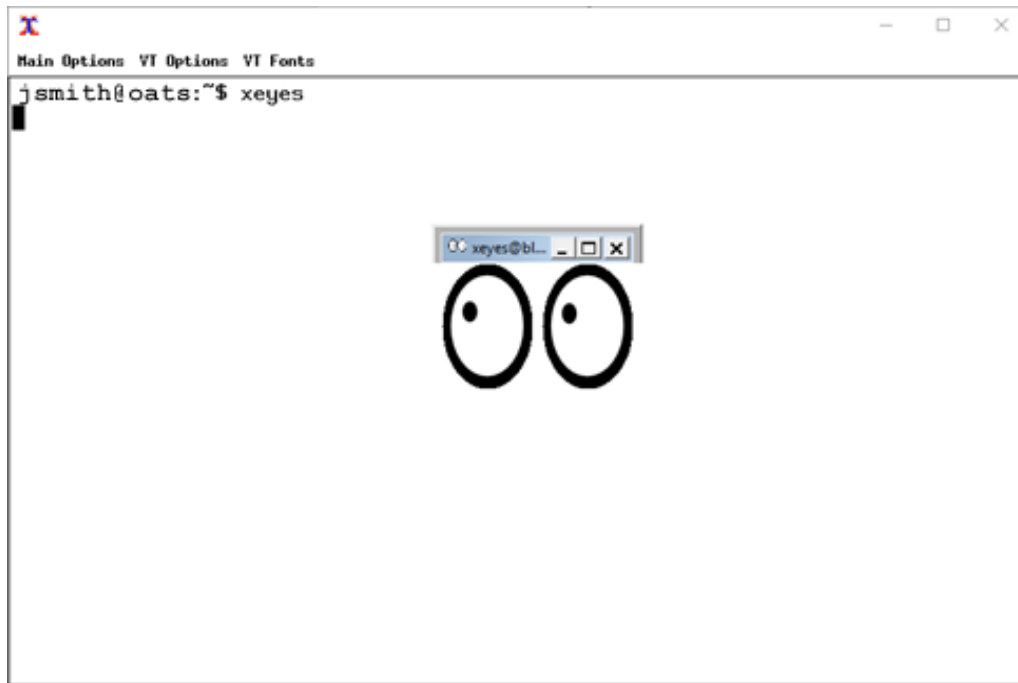
Note that while the program xeyes is running in the foreground, you won't be able to type any commands on the command line in your terminal (try it). There are two ways around this:

- The first is to type Ctrl-Z into the terminal, which suspends xeyes – the eyes will no longer respond to your mouse when the program is suspended, but you will have your command line back. Then, typing "bg" into the command line will put the last process into the "background," meaning that xeyes can run in the background while you type commands into the command line in the foreground.

- The other way is to type an ampersand on the command line when starting xeyes ("xeyes &") which starts the program in the background and immediately returns you to the command prompt. These types of basic multitasking abilities have existed in Linux long before they were stable in Windows and Mac operating systems.

You can close xeyes by either closing the window (clicking "X") or typing Ctrl-C into the command line interface. Remember that the xeyes *client* is running on the remote system (bloch, which is sitting in my lab), but the graphics are being displayed on your system (via the X11 *server* on your PC) over the Internet.

Now that you've got SSH and Cygwin set up on your system, you might also want to download WinSCP. This is a useful (free) program for transferring files between your system and the remote system, and you can download it at http://winscp.net/. The installation is much easier and intuitive.

At this point, you have a working system, and you can enter any commands you like at the UNIX prompt. Using Linux and UNIX takes a lot of practice, and the details are beyond the scope of this simple tutorial. For a good introduction to using the UNIX, check out http://www.ee.surrey.ac.uk/Teaching/Unix/. I also provide a UNIX Cheat Sheet at http://folding.chemistry.msstate.edu/files/win-x11/unix_cheat_sheet.pdf

Good luck!