How to use eceubuntu (formerly, ecelinux)

The easiest way to connect to the ECE Linux servers is by using a secure shell client (SSH client), many of which are available from the Internet. For example, many people use Putty (just do a Google search for Putty and it will be the first link). Alternatively, if you're on a Mac, you can open a terminal window and give the command `ssh eceubuntu` or `ssh ecelinux4.uwaterloo.ca` (as described below).

On campus connection

If you are connected to any of the UW networks, you just need to open your ssh application and connect to hostname `eceubuntu` using your Nexus/WatIAM username and password.

Off campus connection

If you are off campus, use the same procedure as if you are on campus (indicated above) except for the host name which should be `eceTerm.uwaterloo.ca` or `ecelinux4.uwaterloo.ca`. After entering your password, you'll be logged in (which is the only one of our linux machines that can be accessed from off-campus due to the firewall). Once you're on ecelinux4, use the command `ssh eceubuntu` to connect to our main Linux system. You will again be prompted to enter your UW username and password. Note that nothing will appear while you type your password, to prevent "shoulder surfing".

Using Linux

If you have never used Linux before, here are some useful links:

http://linuxcommand.org/lc3_learning_the_shell.php

https://ryanstutorials.net/linuxtutorial/commandline.php

Editing files

Linux has a number of text editors. Vim is more powerful, Pico is easier to learn.

Tutorial on Vim: https://www.linux.com/learn/vim-101-beginners-guide-vim

Tutorial on Pico: https://www.cs.colostate.edu/helpdocs/pico.html

Compiling and running your C/C++ code

The gcc and g++ commands can be used to compile your code:

```
gcc yourfile.cpp –o yourfile
```

```
g++ yourfile.cpp –o yourfile
```

Use gcc for C, g++ for C++.

Obviously, replace *yourfile* with the name of your source file. The result will be an executable file called *yourfile* which you can run as `./yourfile` (possibly redirecting the input and or/or output). For example:

```
/.yourfile <somedata
```

```
./yourfile >youroutput
```

```
./yourfile <somedata >youroutput
```

The first example has your program read from the text file *somedata*, the second has your program write its output to *youroutput*, the third example does both of those things.

Using Python

On ecelinux, we have both Python 2.7 and Python 3.4 installed. You run them like this:

```
python yourfile.py
```

```
python3 yourfile.py
```

Again, you obviously replace *yourfile* with the name of your python source file. You can also redirect the input and output, just as you did in the C/C++ example above.

Uploading files

If you prefer, you can edit and test your code on your own computer. However, your code **must** run on ecelinux since that is where the TAs will be testing it.

If you're using the SSH Secure Shell on Nexus, there's button in the toolbar called "New File Transfer Window" which will allow you to drag-and-drop files between your computer and ecelinux.

If you're using Putty, you can open up a CMD window and use the `pscp` command. For example:

```
pscp yourfile.cpp youruserid@ecelinux4.uwaterloo.ca:yourfile.cpp
```

Be sure to run your code on ecelinux before submitting it to Learn!