CS 233 on Your Own Machine

As many of you might have already experienced, the EWS infrastructure can be super unreliable; remotely logging in is especially terrible, particularly when using NX Client. Consequently, it is highly recommended to set up a development environment on your own machine to avoid having to deal with EWS (for the most part at least), and this guide should help you get started.

One caveat to be aware of is that your assignments will still be graded on the EWS machines, so it is your responsibility to ensure they work on those. However, developing locally on your box and just doing final testing on the EWS machines is pretty much guaranteed to be a far less painful experience than developing entirely on EWS.

Error rendering macro 'toc'

Windows

It’s possible to set up most of the tools you need for this course natively on Windows, but I strongly recommend against it. Installing and troubleshooting everything on Windows is far more trouble than it’s worth, and as the philosopher Wilkins once said, “ain’t nobody got time for that”. Instead, get some form of Linux running alongside Windows, which will help you greatly in 225 and 241 as well. CS 225 offers a preconfigured virtual machine (instructions to set up the 233 tools can be found further down), or you can also up your own VM (and I personally prefer VMware Player over Virtualbox). You can also dual boot your distro of choice (my favorite is Linux Mint with Cinnamon).

Arch Linux (CS 2xx VM)

Before doing any of this, make sure you’ve followed the steps in the 2xx VM setup guide. Note that you need to install the Extension Pack separately; double clicking on the downloaded file should do the trick. Also note that double clicking on the downloaded virtual image might not install it, in which case you should open up Virtualbox and use File -> Import Appliance... to get the 2xx VM set up.

NOTE: If you at any point get 404 errors, you need to update your local repos and packages. Run:

```
sudo pacman -Syuy
```

iverilog and gtkwave

The 2xx VM should include iverilog and gtkwave by default. If for some reason it doesn't, run the following command to get them:

```
sudo pacman -S iverilog gtkwave
```

and enter Y when asked if you want to install.

clang

clang should come installed, however, you can use this command to install it.

```
sudo pacman -S clang
```

spim-vasm

For a 64-bit virtual machine, run the following command:
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux64/spim-vasm /usr/local/bin/spim-vasm --username YOUR_NETID

For a 32-bit virtual machine, run the following command:

sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux32/spim-vasm /usr/local/bin/spim-vasm --username YOUR_NETID

Qt 4 libraries

The latest 2xx VM (the one released this semester) should come with these preinstalled. If you have an older version, you need to run the following command:

sudo pacman -S qt4

QtSpim

Make sure you've set up the Qt 4 libraries first.

For a 64-bit virtual machine, run the following command:

sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux64/QtSpim /usr/local/bin/QtSpim --username YOUR_NETID

For a 32-bit virtual machine, run the following command:

sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux32/QtSpim /usr/local/bin/QtSpim --username YOUR_NETID

QtSpimbot

Make sure you've set up the Qt 4 libraries first.

For a 64-bit virtual machine, run the following command:

sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux64/QtSpimbot /usr/local/bin/QtSpimbot --username YOUR_NETID

For a 32-bit virtual machine, run the following command:

sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux32/QtSpimbot /usr/local/bin/QtSpimbot --username YOUR_NETID
Debian-based distros (Debian, Ubuntu, Mint, etc.)

**iverilog and gtkwave**

Run the following command in a terminal and you're good to go:

```
sudo apt-get install iverilog gtkwave
```

**clang**

Just run

```
sudo apt-get install clang
```

**spim-vasm**

For a 64-bit machine, run the following command:

```
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux64/spim-vasm /usr/local/bin/spim-vasm --username YOUR_NETID
```

For a 32-bit machine, run the following command:

```
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux32/spim-vasm /usr/local/bin/spim-vasm --username YOUR_NETID
```

**QtSpim**

For a 64-bit machine, run the following command:

```
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux64/QtSpim /usr/local/bin/QtSpim --username YOUR_NETID
```

For a 32-bit machine, run the following command:

```
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux32/QtSpim /usr/local/bin/QtSpim --username YOUR_NETID
```

**QtSpimbot**

For a 64-bit virtual machine, run the following command:
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux64/QtSpimbot /usr/local/bin/QtSpimbot --username YOUR_NETID

For a 32-bit virtual machine, run the following command:

sudo svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Linux32/QtSpimbot /usr/local/bin/QtSpimbot --username YOUR_NETID

**OS X**

**X11**

Install XQuartz to be able to use X-forwarding over SSH and gtkwave. After installing, log out and log back in for the environment changes to take effect.

**iverilog**

Install Homebrew. Once you've installed brew, run the command

```
brew install icarus-verilog
```

**gtkwave**

Download the app here. Once you've unzipped and run it once, you should be able to view vcd files on the terminal using the `open` command. For example, to view the file `test.vcd`, you would run the following command on the terminal:

```
open test.vcd
```

**clang**

The first time you try to use clang++ (or any other command line development tool), you'll get a prompt asking you to install the command line developer tools. Doing so should get everything up and running.

**spim-vasm**

Try running the following command:

```
svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Mac/spim-vasm /usr/local/bin/spim-vasm --username YOUR_NETID
```

If that fails with a permission error, try prefixing the command with `sudo`. If that still doesn't work, or if you get some other error, post on Piazza and we'll get it figured out.

**Qt 4 libraries**
Make sure you have Homebrew installed, and then do:

```
brew install qt
# or, if you use Sierra (takes a while to compile, block out a good time for this)
brew install cartr/qt4/qt
```

**QtSpim**

Make sure you've set up the Qt 4 libraries first.

Try running the following command:

```
svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Mac/QtSpim /usr/local/bin/QtSpim --username YOUR_NETID
```

If that fails with a permission error, try prefixing the command with `sudo`. If that still doesn't work, or if you get some other error, post on Piazza and we'll get it figured out.

**QtSpimbot**

Make sure you've set up the Qt 4 libraries first.

Try running the following command:

```
svn export --force https://subversion.ews.illinois.edu/svn/fa16-cs233/_shared/Mac/QtSpimbot /usr/local/bin/QtSpimbot --username YOUR_NETID
```

If that fails with a permission error, try prefixing the command with `sudo`. If that still doesn't work, or if you get some other error, post on Piazza and we'll get it figured out.