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 FastX 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.

Also, note that all of our downloads are 64 bit. If you are using a 32-bit system or VM (extremely unlikely), try replacing "64" with 32 in the commands that specify a version, and if we haven't built them, ask. However, if you're really using a 32-bit VM, consider moving to a 64 bit system, as they are typically more performant.

<table>
<thead>
<tr>
<th>1 Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Arch Linux</td>
</tr>
<tr>
<td>2.1 Subversion</td>
</tr>
<tr>
<td>2.2 iverilog and gtkwave</td>
</tr>
<tr>
<td>2.3 clang</td>
</tr>
<tr>
<td>2.4 spim-vasm</td>
</tr>
<tr>
<td>2.5 Qt 4 libraries</td>
</tr>
<tr>
<td>2.6 QtSpim</td>
</tr>
<tr>
<td>2.7 QtSpimbot</td>
</tr>
<tr>
<td>3 Debian-based distros (Debian, Ubuntu, Mint, etc.)</td>
</tr>
<tr>
<td>3.1 Subversion</td>
</tr>
<tr>
<td>3.2 iverilog and gtkwave</td>
</tr>
<tr>
<td>3.3 clang</td>
</tr>
<tr>
<td>3.4 spim-vasm</td>
</tr>
<tr>
<td>3.5 QtSpim</td>
</tr>
<tr>
<td>3.6 QtSpimbot</td>
</tr>
<tr>
<td>4 OS X</td>
</tr>
<tr>
<td>4.1 X11</td>
</tr>
<tr>
<td>4.2 iverilog</td>
</tr>
<tr>
<td>4.3 gtkwave</td>
</tr>
<tr>
<td>4.4 clang</td>
</tr>
<tr>
<td>4.5 spim-vasm</td>
</tr>
<tr>
<td>4.6 Qt 4 libraries</td>
</tr>
<tr>
<td>4.7 QtSpim</td>
</tr>
<tr>
<td>4.8 QtSpimbot</td>
</tr>
</tbody>
</table>

### Windows

With the onset of Windows 10's "Windows Subsystem for Linux", it's possible to do most of what you need in this class on Windows. If you're using something older than Windows 10, then you should be creating a VM with a Linux operating system, and following the relevant guide below for your distribution. If you don't know how to set up a Linux VM, do a search. There are plenty of guides out there.

Install "Bash on Ubuntu on Windows" by following this [MSDN guide](https://docs.microsoft.com/en-us/windows/powershell/viual-studio-code-ubuntu-on-windows). Then, once you've opened the terminal, use `apt-get` to install Qt4:

```
sudo apt-get install libqtgui4
```

In order to run graphical applications, you will need to install an X server. I suggest [VcXsrv](https://vcxsrv.sourceforge.io/), however there are other alternatives. After installing the X server, you will need to tell programs where to look for the display. Run the following command to make sure this happens:

```
echo "export DISPLAY=:0" >> ~/.bashrc
```

Before running a graphical application (like GTKWave or QtSpim), you will have to start VcXsrv.

Now, follow the "Debian-based distros" guide below (as WSL is Ubuntu).

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

```
sudo pacman -Syyu
```

**Subversion**

Run the following command to get them:

```
sudo pacman -S subversion
```

and enter `Y` when asked if you want to install.

**iverilog and gtkwave**

Run the following command to get them:

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

and enter `Y` when asked if you want to install.

**clang**

Run the following command. This will install the latest version of clang (3.9 as of writing), which is likely newer than what is on EWS. Be sure to test that your code compiles on EWS, but if you're using Arch Linux, you probably already know the ramifications of using newer versions.

```
sudo pacman -S clang
```

It's possible to install Clang 3.5 to match EWS by running:

```
sudo pacman -S clang35
```

But this will remove the standard clang and LLVM installation, which may or may not be good depending on your needs.

**spim-vasm**

Run the following command:

```
sudo svn export --force
   https://subversion.ews.illinois.edu/svn/fa17-cs233/_shared/Linux64/spim-vasm
   /usr/local/bin/spim-vasm --username YOUR_NETID
```

**Qt 4 libraries**

Run the following command:
sudo pacman -S qt4

**QtSpim**

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

Run the following command:

```bash
sudo svn export --force
https://subversion.ews.illinois.edu/svn/fa17-cs233/_shared/Linux64/QtSpim
/usr/local/bin/QtSpim --username YOUR_NETID
```

**QtSpimbot**

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

Run the following command:

```bash
sudo svn export --force
https://subversion.ews.illinois.edu/svn/fa17-cs233/_shared/Linux64/QtSpimb
t /usr/local/bin/QtSpimbot --username YOUR_NETID
```

**Debian-based distros (Debian, Ubuntu, Mint, etc.)**

**Subversion**

Run the following command to get them:

```bash
sudo apt-get install subversion
```

and enter `y` when asked if you want to install.

**iverilog and gtkwave**

Run the following command:

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

**clang**

To get the version of Clang we use on EWS (3.5, for now), run:

```bash
sudo apt-get install clang-3.5
```
spim-vasm

Run the following command:

```
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa17-cs233/_shared/Linux64/spim-vasm /usr/local/bin/spim-vasm --username YOUR_NETID
```

QtSpim

Run the following command:

```
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa17-cs233/_shared/Linux64/QtSpim /usr/local/bin/QtSpim --username YOUR_NETID
```

QtSpimbot

Run the following command:

```
sudo svn export --force https://subversion.ews.illinois.edu/svn/fa17-cs233/_shared/Linux64/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
Run the following to get Xcode's command line tools. This should give you clang and other required items. Running clang++ or any other tool should also initiate the setup.

```
xcode-select --install
```

**spim-vasm**

Run the following command:

```
svn export --force
https://subversion.ews.illinois.edu/svn/fa17-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 a good amount of time to compile), and then do:

```
brew tap cartr/qt4
brew tap-pin cartr/qt4
brew install qt@4
```

**QtSpim**

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

Run the following command:

```
svn export --force
https://subversion.ews.illinois.edu/svn/fa17-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.

Run the following command:

```
svn export --force
https://subversion.ews.illinois.edu/svn/fa17-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.