USCC
Programming Assignment 0 – Installation (Mac)

By: Sanjay Madhav
University of Southern California

Prerequisites

As the preferred development environment, Mac is the simplest to configure. The only prerequisites are:

- Mac OS X 10.9 (Mavericks) or higher
- Xcode 5 or higher
- ~5 GB free space (when built, LLVM will take about almost 4 GB)

With Xcode 5 installed, several command line tools that are necessary should also be installed.

These installation instructions are all handled via the command line. However, once installed you can edit, build, and debug the code directly in Xcode. Test suites will have to be ran from the command line, however.

Installing and Building LLVM

In case it's unclear, in the instructions below `$` is not something you should type, but it just represents the command prompt.

1. Open up a terminal and cd to the directory you want to work in. You will be making multiple directories and links, so it's not recommended to do this in ~.
2. Download the LLVM 3.5 source:
   ```
   $ curl -O http://llvm.org/releases/3.5.0/llvm-3.5.0.src.tar.xz
   ```
3. Extract the archive:
   ```
   $ tar xJf llvm-3.5.0.src.tar.xz
   ```
4. Rename the directory to `llvm`:
   ```
   $ mv llvm-3.5.0.src llvm
   ```
5. Enter the llvm directory:
   ```
   $ cd llvm
   ```
6. Configure LLVM to build a debug build:
   ```
   $ ./configure --disable-optimized
   ```
7. Build LLVM (this may take 30 minutes to 1 hour depending on the system, but you only have to do it once):
   ```
   $ make
   ```
8. Once you've built LLVM, you need to make a couple of links in the parent directory, so:
   ```
   $ cd ..
   $ ln -s llvm/Debug+Asserts/lib/ lib
   $ ln -s llvm/Debug+Asserts/bin/ bin
   ```
9. Your LLVM is now setup and you can move on to getting the USCC code

Forking and Cloning the Starting Code

1. Open the following URL in your browser: https://bitbucket.org/uscc/uscc-projects
2. Fork the repository *making sure you make your fork private*. Please do not make a public fork as it will allow anyone, including other students, to see your code and potentially copy off of it.
3. Now go back to the terminal and make sure you are in the directory that contains the `llvm` directory as well as the two symbolic links to `bin` and `lib`
4. Clone your fork into a local directory called `uscc`:
   `$ git clone https://username@bitbucket.org/username/forkname.git uscc`
   (You can get the exact URL in the top right corner of your repo's page on Bitbucket. Alternatively, you can checkout using a visual git client such as SourceTree.)

Building in Xcode

1. Browse to your uscc directory in Finder and open `uscc.xcodeproj`
2. Build the project using ⌘ + B.
   (If you really want to, you can also build from the command line inside the `uscc` directory via `make`.)
3. Now go back to the terminal and `cd` to `uscc/tests`
4. In the tests directory, run the following:
   `$ ../bin/uscc -a test002.usc`

   You should then get the following output:
   ```
   test002.usc:16:1: error: Function implementation missing
   {
   ^
   1 Error(s)
   ```
5. Then try running the test suite with the following command:
   `$ python testParse.py`

   You should have 22 out of 23 tests fails.

You are now ready to start working on PA1.

Debugging in Xcode

If you want to debug the project in Xcode, it's fairly straightforward to setup:

1. With the `uscc.xcodeproj` open, go to Product>Scheme>Edit Scheme...
2. Select the "Run" scheme and the "Options" tab
3. Check "Use custom working directory" and set the directory to:
   `$(PROJECT_DIR)/tests`
4. In the arguments tab, add the command line arguments you want to use when debugging, for example:
   `-a test002.usc`
5. Now run with ⌘ + R or by hitting the play button, and it'll run in the debugger.