USCC Programming Assignment 0: Installation (Linux)

CSE 504 Compiler Design
Stony Brook University

Prerequisites

USCC will work natively on Linux, though you must build via the command line. These instructions assume you are using Ubuntu 23.04.3 LTS. But any similar Linux distribution should work. In addition to Linux, you need to install clang, g++, and python which you can do with:

```
$ sudo apt install clang
$ sudo apt install g++
$ sudo apt install python
```

You will also need approximately 5GB of free space. I also strongly recommend you change your default linker to `gold`, because it will link significantly faster than if you use the default `ld` and it requires much less memory, to boot. On Ubuntu 24.04, `ld` is just a symbolic link to `ld.bfd` (or `x86_64-linux-gnu-ld.bfd`), and you want to change that to symbolic link to `ld.gold` (or `x86_64-linux-gnu-ld.gold`) instead. You can do this with the following commands:

```
$ cd /usr/bin
$ sudo rm ld
$ sudo ln -s x86_64-linux-gnu-gold ld
```

Note that if you do this, you will not be able to link kernel modules anymore. However, you can always change the symbolic link to point back to `ld.bfd` if you need to.

Installing and Building LLVM

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: (Yes, it is old, but USCC has been built and tested with LLVM 3.5.)
   ```
   $ wget http://llvm.org/releases/3.5.0/llvm-3.5.0.src.tar.xz
   ```
3. Extract the archive:
   ```
   $ tar zxvf 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 CC=clang CXX=clang++ --disable-optimized
   ```
7. Build LLVM (This may take a while. Adjust parallel build options -j2 based on your vCPU counts):
   ```
   $ make -j2
   ```
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. LLVM is now setup for use with USCC

Building USCC via Command Line

USCC can be built via standard makefiles. Since USCC leverages C++11 features, it must be built with a compiler that supports `-std=c++11` or `-std=gnu++11`. This and other compiler/linker flags can be modified in `uscc/Makefile.variables`.

Here is a series of steps for building USCC using the makefiles and testing it:

1. Make sure you are in the directory that contains the `llvm` directory as well as the two symbolic links to `bin` and `lib`, then download `uscc.tgz` form `Blackboard::Assignments::PA0` and extract it there:
   ```
   $ tar zxvf uscc.tgz
   ```
2. Enter the `uscc` directory:
   ```
   $ cd uscc
   ```
3. Build uscc:
   ```
   $ make -j2
   ```
4. To verify it worked, enter the tests directory:
   ```
   $ cd tests
   ```
5. In the tests directory, run the following:
   ```
   $ ../bin/uscc -i -a test002.usc
   ```
   You should then get the following output:
   ```
   test002.usc:16:1: error: Function implementation missing
   {
   ^
   1 Error(s)
   ```
6. 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 begin working on PA1. In the meantime, please refer to (1) USC language manual (`USCLanguage.pdf`) and (2) USCC complier manual (`USCCompiler.pdf`).