

# USCC

## Programming Assignment 0 – Installation (Linux)

CS 502 Compiling and Programming Systems  
Purdue University

### Prerequisites

USCC will work natively on Linux, though you must build via the command line. These instructions assume you are using Ubuntu 14.04. But any similar Linux distribution will work. In addition to Linux, you need to install clang and g++, which you can do with:

```
$ sudo apt-get install clang
$ sudo apt-get install g++
```

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 14.04, `ld` is just a symbolic link to `ld.bfd`, and you want to change that to symbolic link to `ld.gold` instead. You can do this with the following commands:

```
$ cd /usr/bin
$ sudo rm ld
$ sudo ln -s ld.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:  

```
$ wget 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 CC=clang CXX=clang++ --disable-optimized
```
7. Build LLVM (this may take over an hour depending on your machine, but you only have to do this 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
```

---

Copyright © 2014, Sanjay Madhav. All rights reserved.

Thanks to Dr. Changhee Jung for talking through some ideas, being a second set of eyes, as well as creating the Makefiles. This document may be modified and distributed for nonprofit educational purposes, provided that this copyright notice is preserved.

9. LLVM is now setup for use with USCC

## Building USCC via Command Line

USCC can be built via standard makefiles contributed by Dr. Changhee Jung from Virginia Tech. These makefiles will work not only on Mac OS X, but on other UNIX flavors such as Linux and Solaris. The easiest way to build with the makefiles is as follows:

```
cd uscc (move to the uscc package directory above)
make (Build everything)
```

The makefiles also support the following build rules:

```
make clean (delete all objects, libraries, and the executable)
make depend (generate dependencies for uscc source files)
```

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 `cs502-uscc.tgz` from Scholar's resources page and extract it there:  

```
$ tar xvpz cs502-uscc.tgz -C ./
```
2. Enter the `uscc` directory:  

```
$ cd uscc
```
3. Build `uscc`:  

```
$ make
```
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.