



Learn, Share, Build

Each month, over 50 million developers come to Stack Overflow to learn, share their knowledge, and build their careers.

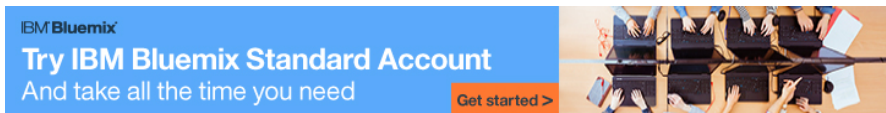
Google

Facebook

OR

Join the world's largest developer community.

How to make clang compile to llvm IR



I want clang to compile my `c/c++` code to `LLVM` bytecode rather than binary executable. How can I achieve that? And if I get the `LLVM` bytecode, how can I take it to further compile it to binary executable.

Basically I want to add some of my own code to the `LLVM` bytecode before compiling to binary executable.

`c` `llvm` `clang`

edited Feb 6 '12 at 9:48



Jav_Rock

16.3k

17

91

157

asked Feb 5 '12 at 11:34



pythonic

6,125

17

66

119

5 Answers

Given some C/C++ file `foo.c` :

```
> clang -S -emit-llvm foo.c
```

Produces `foo.ll` which is an LLVM IR file.

The `-emit-llvm` option can also be passed to the compiler front-end directly, and not the driver by means of `-cc1` :

```
> clang -cc1 foo.c -emit-llvm
```

Produces `foo.ll` with the IR. `-cc1` adds some cool options like `-ast-print` . Check out `-cc1 --help` for more details.

To compile LLVM IR further to assembly, use the `llc` tool:

```
> llc foo.ll
```

Produces `foo.s` with assembly (defaulting to the machine architecture you run it on). `llc` is one of the LLVM tools - [here is its documentation](#).

edited Nov 24 '13 at 18:05

answered Feb 5 '12 at 12:56



Eli Bendersky

139k

56

274

342

3 What does -S do here? – [meawoppl](#) Feb 27 '14 at 16:17

8 @meawoppl: -S like in gcc says emit textual assembly rather than assembled binary – [Eli Bendersky](#) Feb 28 '14 at 16:53

Ahha. I was having a hard time finding anything in the docs about it. It is safe to assume that many flags in clang mirror gcc flag structure? – [meawoppl](#) Feb 28 '14 at 17:35

@EliBendersky Do you know how to compile multiple `.c` and `.h` files into one human readable IR so that I can run the IR using 'lli theirFile'? Thanks – [cache](#) Jul 20 '14 at 18:43