Сборка LLVM и Clang

LLVM **и** Clang

- **LLVM** (Low Level Virtual Machine) - открытая клмпиляторная инфраструктура.
- Разработчик: LLVM Developer Group. Написана на C++
- **Clang** является фронтендом для языков программирования C, C++, Objective-C, использующим для оптимизации и кодогенерации фреймворк LLVM.
- Последняя стабильная версия 3.6

Сборка LLVM и Clang Linux

- Установить subversion (Ubuntu: *sudo apt-get isntall subversion*)
- Установить gcc, g++, gcc-multilib, g++-multilib (Ubuntu: *sudo apt-get install gcc g++ gcc-multilib g++-multilib*)
- Выберете директорию куда вы хотите получить исходный код LLVM и выполните следующую команду:

  svn co http://llvm.org/svn/llvm-project/llvm/branches/release_35llvm

- После чего в выбранной вами директории появится папка с исходным кодом LLVM, перейдите в нее, перейдите в папку tools и получите исходный код Clang:

cd llvm

cd tools

svn co http://llvm.org/svn/llvm-project/cfe/branches/release_35 clang

Сборка LLVM и Clang

- Далее необходимо создать директории сборки и установки (build и install соответственно). Рекомендую их создать в директории с исходным кодом llvm:

cd /home/username/llvm

mkdir build install

- Перейдите в директорию build и запустите configure со следующими опциями

../configure --prefix=/home/username/llvm/install \

--enable-optimized --disable-assertions

- Затем запустите сборку командой *make -jN*. Max N=(Количество ядер*2) + 1

На Intel® Core™ i7-4790K CPU @ 4.00GHz сборка занимает порядка 10 минут.

После окончания сборки "скажите" *make install*, дождитесь окончания установки и по завершению в директории */home/username/llvm/install/bin/* будут расположены исполняемые файлы Clang и LLVM

Транслирование .C code в .ll code (LLVM IR)

- Для перевода исходного в представление LLVM необходимо воспользоваться clang`ом, для .c code - clang для .c++ - clang++.
- Команда для транслирования:

/home/username/llvm/install/bin/clang -S -emit-llvm /path/to/c-code.c

В директории запуска создастся файл c-code.ll, содержимым которого будет являться исходная программа в представлении LLVM (LLVM IR).

Пример

- Исходная программа на языке C:

```
#include <stdio.h>


int main() {
	printf("Hello, World!\n");
	int a = 5;
	return 0;
}
```

- "говорим" */home/username/llvm/bin/clang -S -emit-llvm hello.c*

Пример

И получаемhello.ll:

```
; ModuleID = 'hello.c'
target datalayout = "e-m:e-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-unknown-linux-gnu"
@.str = private unnamed_addr constant [15 x i8] c"Hello, World!\0A\00", align 1

; Function Attrs: nounwind uwtable
define i32 @main() #0 {
  %1 = alloca i32, align 4
  %a = alloca i32, align 4
  store i32 0, i32* %1
  %2 = call i32 (i8*, ...)* @printf(i8* getelementptr inbounds ([15 x i8]* @.str, i32 0, i32 0))
  store i32 5, i32* %a, align 4
  ret i32 0
}
declare i32 @printf(i8*, ...) #1
```

•