

导航

博客园
首 页
联 系
订 阅
管 理

< 2019年4月 >						
日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

公告

昵称: owlman
园龄: 4年5个月
粉丝: 3
关注: 0
+加关注

搜索

随笔分类(24)

C++(3)
Git(3)
Linux(3)
MacOSX(1)
PHP
Python
Web前端
出版作品(5)
读书体会(4)
工作/学习(2)
技术评论(1)
一些问题(2)

随笔档案(20)

2017年5月 (3)
2016年6月 (1)
2016年5月 (2)
2016年4月 (2)
2016年3月 (1)
2015年3月 (1)
2015年2月 (2)
2014年11月 (8)

积分与排名

积分 - 8689
排名 - 54515

最新评论

1. Re:Makefile简易教程
谢谢老板, 有了初步认识
--luyi07
2. Re:Makefile简易教程
感谢博主, 让我对makefile
的用法有了初步的了解。
--Andrés
3. Re:Makefile简易教程
写的很详细, 按流程跑下来
基本知道makefile的简单写
法了。
--风雨中的前行者
4. Re:clang编译器简介
学习
--NewSea

评论排行榜

1. Makefile简易教程(3)
2. clang编译器简介(1)

clang编译器简介

本文部分内容引用:

[中文维基百科](#)。

[结构化编译器前端--clang介绍](#)。

什么是clang编译器?

clang是LLVM编译器工具集的一个用于编译C、C++、Objective-C的前端。LLVM项目的目标是提供一个GNU编译器套装(gcc)的替代品, 由苹果公司的赞助开发, 其源代码授权采用的是类BSD的伊利诺伊大学厄巴纳-香槟分校开源码许可。

clang编译器的优势与劣势

相比于gcc, clang具有如下优点:

- 编译速度更快: 在某些平台上, clang的编译速度要明显快于gcc。
- 占用内存更小: clang生成的AST所占用的内存通常是gcc的五分之一左右。
- 模块化的设计: clang采用基于库的模块化设计, 更易于IDE的集成及其他用途的重用。
- 诊断信息可读性强: 在编译过程中, clang会创建并保留大量详细的元数据(metadata), 这将更有利于调试和错误报告。
- 设计更清晰简单, 容易理解, 易于扩展加强。与代码基础较为古老的gcc相比, 学习曲线会显得更为平缓。

当前 Clang 还处在不断完善过程中, 相比于gcc, clang在以下方面还需要加强:

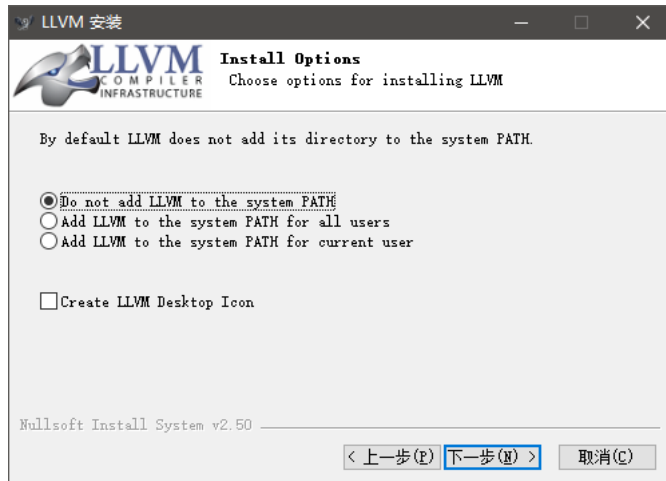
- 需要支持更多语言: gcc除了支持 C/C++/Objective-C, 还支持Fortran/Pascal/Java/Ada/Go等其他语言。clang 目前基本上只支持C/C++/Objective-C/Objective-C++这四种语言。
- 需要加强对C++的支持: clang对C++的支持依然落后于gcc, clang 还需要加强对C++ 提供全方位支持。
- 需要支持更多平台: 由于gcc流行的时间比较长, 已经被广泛使用, 对各种平台的支持也很完备。clang目前支持的平台有 Linux/Windows/Mac OS。

安装LLVM + clang

二进制安装

- 纯Windows环境:

可以从<http://llvm.org/releases/>下载相应的安装包进行安装。目前最新版本为3.8。



- cygwin下的apt-cyg:

```
apt-cyg install clang
```

- Linux下的yum:

```
sudo yum install clang
```

- Linux下的apt-get (有时候需要自己配置一下相关的软件源):

```
sudo apt-get install clang-3.4 clang-3.4-doc libclang-common-3.4-dev libclang-3.4-dev  
libclang1-3.4 libclang1-3.4-dbg libllvm-3.4-ocaml-dev libllvm3.4 libllvm3.4-dbg lldb-3.4 llvm-  
3.4 llvm-3.4-dev llvm-3.4-doc llvm-3.4-examples llvm-3.4-runtime clang-modernize-3.4 clang-format-  
3.4 python-clang-3.4 lldb-3.4-dev
```

源码安装

- 第一步，下载llvm代码：

```
git clone git@github.com:llvm-mirror/llvm.git
```

- 第二步，进入llvm/tools目录并下载clang代码

```
cd llvm/tools  
git clone git@github.com:llvm-mirror/clang.git
```

- 第三步，进入llvm/projects目录并下载compiler-rt代码：

```
cd ../projects  
git clone git@github.com:llvm-mirror/compiler-rt.git
```

- 第四步，在llvm所在目录新建与llvm同一级的目录build，并在其中构建llvm和clang：

```
cd ../..  
mkdir build  
cd build  
../llvm/configure --enable-optimized --enable-assertions  
make  
make install
```

使用clang编译C程序

clang的用法与gcc基本相同，我们可以写一个脚本来验证一下编译器是否已经安装完成：

```
import os  
import sys  
import shutil  
  
if not len(sys.argv) in range(2, 3):  
    print("Usage: hello_c.py <compiler>")  
    exit(1)  
  
code = "#include <stdio.h>\n int main(void) { printf(\"hello world!\\n\"); return 0;} "  
  
if(not os.path.exists("example")):  
    os.mkdir("example")  
  
file = open(r"example/hello.c", 'w')  
file.writelines(code)  
file.close()  
  
cmd = sys.argv[1] + r" example/hello.c -o example/test.exe"  
os.system(cmd)  
os.system(r"example/test.exe")  
  
if(os.path.exists("example")):  
    shutil.rmtree("example")
```

然后，我们只需要在shell中输入 `python hello_c.py clang` 即可，如果看到输出一行“hello world”说明编译器已经可以正常工作。

```
lingjie@lingjie-PC: ~/bin
→ bin git:(master) x python hello_c.py clang
hello world!
→ bin git:(master) x python hello_c.py gcc
hello world!
→ bin git:(master) x |
```

分类: [C++](#), [Linux](#)

好文要顶

关注我

收藏该文

owlman
关注 - 0
粉丝 - 3
[+加关注](#)

0

0

« 上一篇: [Makefile简易教程](#)
» 下一篇: [《大教堂与集市》的启示 — 软件工程的另一种选择](#)

posted on 2016-06-07 13:50 owlman 阅读(10159) 评论(1) 编辑 收藏

评论

#1楼

学习

支持(0) 反对(0)

2017-09-30 08:50 | NewSea

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】超50万C++/C#源码：大型实时仿真组态图形源码
- 【推荐】专业便捷的企业级代码托管服务 - Gitee 码云
- 【推荐】云服务器、云数据库特惠，服务更稳，速度更快，价格更优
- 【活动】2019第四届全球人工智能大会解码“智能+时代”