

C/C++ 中相关概念梳理

作者: [ChenforCode](#)

原文链接: <https://ld246.com/article/1593082662437>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1.gcc

GCC (GNU Compiler Collection, GNU编译器套装), 是一套由 GNU 开发的编程语言编译器。它一套以 GPL 及 LGPL 许可证所发行的自由软件, 也是 GNU计划的关键部分, 亦是自由的类Unix及苹果电脑 Mac OS X 操作系统的标准编译器。GCC 原名为 GNU C 语言编译器, 因为它原本只能处理 C 言。GCC 很快地扩展, 变得可处理 C++。之后也变得可处理 Fortran、Pascal、Objective-C、Java, 以及 Ada与其他语言。

2.llvm

LLVM 是 Low Level Virtual Machine 的简称, 这个库提供了与编译器相关的支持, 能够进行程序语的编译期优化、链接优化、在线编译优化、代码生成。简而言之, 可以作为多种语言编译器的后台来用。如果这样还比较抽象的话, 介绍下 Clang 就知道了: Clang 是一个 C++ 编写、基于 LLVM、发于 LLVM BSD 许可证下的 C/C++/Objective C/Objective C++ 编译器, 其目标 (之一) 就是超越 CC。一点小历史: Apple对Objective-C语言 (甚至后来对C语言) 新增很多特性, 但GCC开发者并不买Apple的帐——不给实现, 因此索性后来两者分成两条分支分别开发, 这也造成Apple的编译器版远落后于GCC的官方版本。另一方面, GCC的代码耦合度太高, 不好独立, 而且越是后期的版本, 代质量越差, 但Apple想做的很多功能 (比如更好的IDE支持) 需要模块化的方式来调用GCC, 但GCC直不给做。甚至最近, 《GCC运行环境豁免条款 (英文版)》从根本上限制了LLVM-GCC的开发。以, 这种不和让Apple一直在寻找一个高效的、模块化的、协议更放松的开源替代品, 于是Apple请了编译器高材生Chris Lattner。。。。。

3.clang

Apple吸收Chris Lattner的目的要比改进GCC代码优化宏大得多——GCC系统庞大而笨重, 而Apple量使用的Objective-C在GCC中优先级很低。此外GCC作为一个纯粹的编译系统, 与IDE配合得很差加之许可证方面的要求, Apple无法使用LLVM 继续改进GCC的代码质量。于是, Apple决定从零开写 C、C++、Objective-C语言的前端 Clang, 完全替代掉GCC。

正像名字所写的那样, Clang只支持C, C++和Objective-C三种C家族语言。2007年开始开发, C编译器最早完成, 而由于Objective-C相对简单, 只是C语言的一个简单扩展, 很多情况下甚至可以等价地写为C语言对Objective-C运行库的函数调用, 因此在2009年时, 已经完全可以用于生产环境。C++支持也热火朝天地进行着。

4.make和cmake

一个C语言程序的开发和运行流程是: 1.编写一个源文件, 即.c文件。2.用编译器编译代码生成目标文即.o文件。3.用链接器链接目标代码生成可执行文件即.exe文件。但是如果源文件太多, 那么一个一的去编译就太过麻烦, 因此出现了一个make工具, 他可以以类似于批处理的方式进行批处理编译, 是需要编写一个规则文件, 即makefile文件。有了这个文件和这个工具, 就可以只使用一条命令完成文件的编译。

对于一个大工程, 编写makefile实在是件复杂的事, 于是人们又想, 为什么不设计一个工具, 读入所源文件之后, 自动生成makefile呢, 于是就出现了cmake工具, 它能够输出各种各样的makefile或者project文件,从而帮助程序员减轻负担。但是随之而来也就是编写cmakelist文件, 它是cmake所依据的则。