

1. Microsoft Visual C++ (简称MSVC)

- 适用语言: C++

- 出现时间: 前身为Microsoft C++, 最早发布于1997年

- 代表人物: 开发者—微软

- 发展过程:

1997年, 微软发布了Visual Studio 97。包含有面向Windows开发使用的Visual C++ 5.0等

1998年, 微软发布了Visual Studio 6.0。所有开发语言Visual Studio的开发环境版本均升至6.0。

2003年, 微软对Visual Studio 2002进行了部分修订, 以Visual Studio 2003的名义发布 (内部版本号为 7.1)。

2005 年, 微软发布了 Visual Studio 2005。.NET 字眼从各种语言的名字中被抹去, 但是这个版本的 Visual Studio 仍然还是面向 .NET 框架的 (版本2.0)。

2007年11月, 微软发布了 Visual Studio 2008。

2010年4月12, 微软发布了 Visual Studio 2010 以及.NET Framework 4.0。

2012年9月12日, 微软在西雅图发布 Visual Studio 2012。

2013年11月13日, 微软发布 Visual Studio 2013。

2014年11月, 微软发布 Visual Studio 2015。

2017年3月8日, 微软发布 Visual Studio 2017。

2019年4月2日, 微软发布Visual Studio2019。

- 与其他类似编译器对比: Visual C++ 6.0出来的时候, 它的编译器对C++标准的兼容性比较低, 只有70%左右。这有其历史原因: Visual C++6.0出生的时候, C++标准还没出来。Visual C++编译器发展到现在对C++标准的兼容性已经很好了, 可以达到98%左右。其为微软的产品, 编译器, 链接器, 运行, 调试等功能于一体的强大开发工具, 特点是功能十分强大, 对于新手来说需要一段时间去摸索。

2. GNU C++ (又名GCC)

- 适用语言: C++

- 出现时间: 1985年由理查德·马修·斯托曼开始

- 代表人物:理查德·马修·斯托曼

- 发展过程:

由理查德·马修·斯托曼在1985年开始的。他首先扩增一个旧有的编译器, 使它能编译C, 这个编译器一开始是以Pastel语言所写的。

在1997年, 一群不满GCC缓慢且封闭的创作环境者, 组织了一个名为EGCS (Experimental/Enhanced GNU Compiler System) 的项目, 此项目汇整了数项实验性的分支进入某个GCC项目的分支中。EGCS比起GCC的建构环境更有活力, 且EGCS最终也在1999年四月成为GCC的官方版本。

最新版本在2019-2-22日更新

- 与其他类似编译器对比: 开源, 支持的语言丰富, 但是有一些兼容问题, 是目前较为成功的编译系统之一

3. llvm编译器

- 适用语言: 用于优化以任意程序语言编写的程序

- 出现时间：2000年
- 代表人物：Chris Lattner
- 发展过程：LLVM计划启动于2000年，最初由University of Illinois at Urbana-Champaign的Chris Lattner主持开展。2006年Chris Lattner加盟Apple Inc.并致力于LLVM在Apple开发体系中的应用。Apple也是LLVM计划的主要资助者。
- 与其他类似编译器对比：Low Level Virtual Machine (LLVM) 是一个开源的编译器架构，它已经被成功应用到多个应用领域。Clang (发音为 /klæŋ/) 是 LLVM 的一个编译器前端，它目前支持 C, C++, Objective-C 以及 Objective-C++ 等编程语言。Clang 对源程序进行词法分析和语义分析，并将分析结果转换为 Abstract Syntax Tree (抽象语法树)，最后使用 LLVM 作为后端代码的生成器。与 GCC 相比，Clang 是一个重新设计的编译器前端，具有一系列优点，例如模块化，代码简单易懂，占用内存小以及容易扩展和重用等。由于 Clang 在设计上的优异性，使得 Clang 非常适合用于设计源代码级别的分析 and 转化工具。Clang 也已经被应用到一些重要的开发领域，如 Static Analysis 是一个基于 Clang 的静态代码分析工具。

4. Watcom C/C++

- 适用语言：C/C++
- 出现时间：上世纪90年代初
- 代表人物：SciTech 软件公司
- 发展过程：（近期）Open Watcom C/C++ 的1.4版于2005年12月发行，采用 Linux x86 为实验目标，支持NT、OS/2等host平台。曾有某被弃置的QNX版本，但其编译所须的库并未开源发行。当前最近的稳定版是1.9版，在2010年6月发行
- 与其他类似编译器对比：在DOS开发环境中，Watcom C/C++ 编译器 以编译后的exe运行高速而著称，且首个支持Intel 80386 “保护模式”的编译器。于90年代中期，大批的雄心技术游戏(例如 Doom、Descent、Duke Nukem 3D 都以 Watcom C 写成)

5. Intel C++

- 适用语言：C++
- 出现时间：未知
- 代表人物：美国 Intel 公司
- 发展过程：该编译器的最新发布是 Intel C++ Compiler 14.0 版本
- 与其他类似编译器对比：Intel 编译支持 IA-32、Intel 64、Itanium 2、Intel Atom 处理器和某些非 Intel 的兼容处理器（例如某些 AMD 处理器）。适用于 IA-32 和 Intel 64 的 Intel C++ 编译器的主要特点是自动向量化器，它能够生成 SSE、SSE2 和 SSE3 的 SIMD 指令及其适用于 Intel 无线 MMX 和 MMX 2 的嵌入式变种。
Intel C++ Compiler 进一步支持 OpenMP 3.0 和适用于对称多处理的自动并行化。借助于 Cluster OpenMP 的附加能力，编译器还可为分布存储多处理根据 OpenMP 指示自动生成消息传递接口调用。

6. TCC (Tiny C compiler)

- 适用语言：C
- 出现时间：2001年
- 代表人物：Fabrice Bellard (开发者)
- 发展过程：TCC 项目起源于 OTCC (Obfuscated TCC, 即代码混淆过的 TCC)，这是 Bellard 为赢得 2001 年国际C语言混乱代码大赛 (IOCCC) 编写的一个程序。其后，Bellard 反混淆并延伸发展了该程序，于是有了 TCC。

在 2012 年 2 月 4 日之前的某个时间，Bellard 在 TCC 的官方网站上宣布他不再参与 TCC 的维护工作。^[11]

自从 Bellard 离开 TCC 项目以来，已经有不少组织和个人分发补丁或创建维护分支，为 TCC 提供构建支持以及问题修复。

- 与其他类似编译器对比：自身体积非常小，编译/链接速度非常快，生成的代码质量一般，可以自举（自己可以编译自己）

7. Dev-C++

- 适用语言：C/C++
- 出现时间：（2005年起不再更新）
- 代表人物：开发公司 Bloodshed
- 发展过程：原开发公司 Bloodshed 在开发完 4.9.9.2 后停止开发。

所以现在由 Orwell 公司继续更新开发，最新版本：5.11

- 与其他类似编译器对比：它是一款免费开源的 C/C++ IDE，内嵌 GCC 编译器（GCC 编译器的 Windows 移植版），是 NOI、NOIP 等比赛的指定工具。Dev C++ 的优点是体积小（只有几十兆）、安装卸载方便、学习成本低，缺点是调试功能弱。

8. Free Pascal

- 适用语言：Pascal
- 出现时间：未知
- 代表人物：由国际组织开发
- 发展过程：由国际组织开发的完全的win32的pascal语言编译器，类似delphi，可编写windows程序。最新的版本为 3.0.4
- 与其他类似编译器对比：自由软件，软件包和运行时库发布在GNU宽通用公共许可证下，允许其他程序静态链接。编译器自身发布在GNU通用公共许可证下。编译器完全用 Pascal 语言写成。

9. Turbo Pascal

- 适用语言：Pascal
- 出现时间：1983年
- 代表人物：Borland公司
- 发展过程：Turbo Pascal语言是编译型程序语言，它提供了一个集成环境的工作系统，集编辑、编译、运行、调试等多功能于一体

Pascal有5个主要的版本，分别是Unextended Pascal、Extended Pascal、Object-Oriented Extensions to Pascal、Borland Pascal和Delphi Object Pascal。其中，Unextended Pascal、Extended Pascal和Object-Oriented Extensions to Pascal是由Pascal标准委员会所创立和维护的，Unextended Pascal类似于瑞士Niklaus Wirth教授和K.Jensen于1974年联名发表的Pascal用户手册和报告，而Extended Pascal则是在其基础上进行了扩展，加入了许多新的特性，它们都属于正式的Pascal标准；Object-Oriented Extensions to Pascal是由Pascal标准委员会发表的一份技术报告，在Extended Pascal的基础上增加了一些用以支持面向对象程序设计的特性，但它属于非正式的标准。Borland Pascal和Delphi Object Pascal是由Borland公司专门为其开发的编译工具设计的Pascal语言，前者是用于DOS的Turbo Pascal系列和Windows 3.x的Turbo Pascal for Windows的传统高级语言，后者是用于Windows的Delphi和Linux的Kylix的面向对象程序设计语言，它们都不是正式的Pascal标准，具有专利性。但由于Turbo Pascal系列和Delphi功能强大并且广为流行，Borland Pascal和Delphi Object Pascal已自成为一种标准，为许多人所熟悉

- 与其他类似编译器对比：Turbo Pascal中增添了集成开发环境（IDE），在这种开发环境中，你可在与WordStar 兼容的文字编辑器中编辑代码，可以运行编译器，查看编译错误并直接跳回到包含错误的行中。在Turbo Pascal之前你不得不退出代码编辑器返回到DOS，然后运行命令行编译器，记下错误行，再打开编辑器跳至错误行，非常烦琐。

10. Java编译器javac

- 适用语言：Java语言
- 出现时间：1995年（Oracle
- 代表人物：SUN公司
- 发展过程：由SUN公司开发，稳定版本为1.7.0_25（2013年7月18日）
- 与其他类似编译器对比：Java语言规范要求每个公共类所在的源代码文件的文件名与其类名相同，除非这个类是一个嵌套类，而class文件所包含的类必须和class文件的文件名相同，包括嵌套类。javac编译之后的class文件名会被自动命名为其中所包含的类的名字。