



C语言编程 魔法书

陈轶 著

基于 C11 标准

- 基于 C11 标准，涵盖透彻的语法讲解、高级特性、主流底层编译器支持，配合大量示例与讲解（How-Why），掌握 C 魔法精髓。
- C 语言与汇编语言重度用户与拥趸者撰写，10 余年开发经验结晶，讲究透彻而实用，字字珠玑。



机械工业出版社
China Machine Press

C语言编程 魔法书

基于 C11 标准

陈轶 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

C 语言编程魔法书：基于 C11 标准 / 陈轶著. —北京：机械工业出版社，2017.5

ISBN 978-7-111-56521-5

I. C… II. 陈… III. C 语言—程序设计 IV. TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 072463 号

C 语言编程魔法书：基于 C11 标准

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：高婧雅

责任校对：殷虹

印 刷：北京市荣盛彩色印刷有限公司

版 次：2017 年 6 月第 1 版第 1 次印刷

开 本：186mm×240mm 1/16

印 张：32.75

书 号：ISBN 978-7-111-56521-5

定 价：99.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88379426 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

为什么要写这本书

本人在 2001 年上了大学本科，读计算机科学与技术专业。在第一年的上半学期，对计算机编程还没什么感觉。但是就在考“C 语言程序设计”这门专业课的前一个月，感觉这门课学了那么久几乎什么都不会，可把我急坏了。然后就在这短短一个月的时间里又是看书，又是上机实验，终于考了 70 多分，算是过关了……不过奇怪的是在考试结束后，就发现自己对编程有了感情。到了大二，我们上“数据结构”所使用的教材是基于 C++ 编程语言的，因为之前没学过 C++ 语言，所以只能自学。而在这个过程中，我发现自己对编程更加热爱。在上完大三之后，我在暑假里又把之前的 C 语言重新巩固一番。有了计算机组成、操作系统、汇编语言、数据结构等知识积淀之后再去看 C 语言编程就感觉容易多了。我也是由此喜欢上了 C 编程语言。

10 年之后，发现国内市面上很多 C 语言参考书仍然显得非常陈旧。不仅基于古老的 C89/90 标准，而且还在用 Visual C++ 6.0 这种既收费又老旧的开发环境教学生。对于比较新的 C99 标准的讲解屈指可数，更鲜有针对最新的 C11 标准的书籍。出于对 C 语言的热爱，在此热切希望能把最新标准的 C 语言奉献给各位读者，也想把 C 语言的方方面面讲透并且能讲得通俗易懂，方便读者去思考实践，所以这也是我写这本书的主要原因。当各位阅读完本书之后，会发现 C 语言竟然如此强大！而且在大部分时候，尤其是我们想集中注意力解决某个特定问题的时候，使用 C 语言要比用其他一些基于面向对象的类 C 编程语言（比如 C++、Java 等）要直观得多！

本书之所以叫“C 语言编程魔法书”，是因为像“宝典”、“圣经”之类的词已经被用滥了。再者，C 语言本身就拥有极其强大的魔力，你能用它做几乎所有的东西。而且几乎每一个 C 语言编译器都能内联汇编语言，或者与 C++、Objective-C 直接兼容，而对于像 Java、C#、Python 等许多编程语言也有相应的接口。所以，我认为 C 语言在计算机编程

语言领域中就好比数学在自然科学中的地位和作用，它是很多编程语言的基础，而且很多编程语言的编译器或解释器也都是基于 C 语言来写的。

就在 2015 年 2 月，Khronos 标准组织发布了最具现代化的图形 API——Vulkan，其主机端接口用的 API 是纯 C 语言。此外，像 OpenGL、OpenCL、OpenAL、OpenVG 等开放标准都基于纯 C 语言。此外，最近 10 年来 TIOBE 每月的编程语言排名，C 语言排名始终能进前两名，也能说明它的使用范围之广，而且许多开源项目也多多少少会使用 C 语言来编写。况且学了 C 语言之后，再学习 C++、Java 等面向对象编程语言也会轻松很多。尤其像 C++ 和 Objective-C，没有 C 语言基础是完全不行的。所以个人十分推荐计算机系的大学生将 C 语言作为自己的计算机入门编程语言！

本书特色

从技术层面上讲，本书介绍了 C 语言的最新标准，即 ISO/IEC 9899:2011。同时，也介绍了主流开源 C 语言编译器 GCC 与 Clang 对标准 C 语言语法的扩充。而且所基于的编译器和开发环境也是比较新的 Visual Studio Community 2017、GCC 5，以及 Clang 3.8 (Apple LLVM 8.0，基于 Xcode 8)。

从适合读者阅读和掌握知识的结构安排上讲，本书分为“预备知识篇”、“基础语法篇”、“高级语法篇”，以及“语法扩展篇”，还有最后的“项目实践篇”。从基础到高级，循序渐进地为读者描述 C 语言编程方法。本书尤其着重 C 语言标准语法上的精确描述，通过许多代码片段给读者介绍各种 C 语言语法知识，并且能反映出 C 语言的灵活性以及在使用上的约束。

本书推崇读者使用合法免费的 C 语言编译器以及集成开发环境，希望读者能有正确的软件版权意识，这样才能更好地为我国软件事业增添光彩，为打造良好的应用市场以及生态环境作出贡献。因此，本书主要选择使用 GCC、Clang 这两个主流开源免费的 C 语言编译器，而集成开发环境 (IDE) 则采用 Visual Studio Community、Eclipse、Xcode 这三个常用的免费开发工具，其中，Visual Studio Community 不是开源的，而 Xcode 则是部分开源的。

本书虽然会讲解整个 C 编程语言，涉及了几乎所有的语法点，但是考虑到本书读者可能是初学 C 语言，且没有多少计算机专业知识，所以本书措辞会尽量通俗，而不过于追求学术化。某些描述可能会不太严谨，但对于本书所用到的 GCC、Clang 这两大主流编译器而言将完全适用。另外，考虑到不少读者从事嵌入式系统开发工作，所以对于 C 语言标准中出现的所谓“由实现定义的”场合会尽量区分情况分别阐明。本书的最终的

目的就是让读者至少能熟练掌握 C 语言编程，能将它灵活地运用于实际工程中。

读者对象

- ☐ 嵌入式系统开发者
- ☐ 移动或桌面客户端应用程序开发者
- ☐ 服务器端应用程序开发者
- ☐ 系统架构师
- ☐ 计算机、电子工程、通信专业的大学生
- ☐ 其他对 C 语言编程感兴趣的人员

如何阅读本书

本书一共分为四大篇。

预备知识篇（第 1 ~ 3 章），简单描述 C 语言的概况、学习 C 语言的预备知识，以及在 Windows、macOS 和 Linux 三大桌面环境下搭建编写 C 环境的方法。

第 1 章 C 魔法概览。主要介绍 C 语言的来历和演化，用它编写代码的编程模式以及我们可以用于实践的主流 C 语言编译器。

第 2 章 学习 C 语言的预备知识。这一章主要为不太熟悉计算机系统的读者提供一些基础的计算机理论知识和相关概念，比如整数与浮点数在计算机中的表示方法、字符编码格式、按位逻辑计算、移位操作等。

第 3 章 C 语言编程的环境搭建。这一章主要介绍了 Windows、macOS 以及 Linux 系统下如何安装并使用主流编译器与集成开发环境。

基础语法篇（第 4 ~ 11 章）讲解 C 语言的基本语法。这是 C 语言程序员必须掌握的。

第 4 章 C 语言中的基本元素。这一章描述了 C 语言中常用字符集以及合法 token 的构成。此外还介绍了标识符、关键字以及标点符号的使用说明。

第 5 章 基本数据类型。这一章介绍了整数类型、字符类型、浮点类型数据的表示，以及它们之间的类型转换。此外还描述了对于这些基本数据类型的算术逻辑操作、投射操作以及通过 sizeof 操作符获取数据类型与对象相应的字节数。

第 6 章 用户自定义类型。这一章描述了枚举、结构体以及联合体这三种用户自定义类型，并介绍了它们的特性以及各种使用方式。

第 7 章 C 语言的数组和指针。这一章十分关键，也是 C 语言的语法难点。这里详

细介绍了 C 语言中一维数组与多维数组的表示以及如何对它们进行操作，然后介绍了 C 语言中的指针类型，详细阐述了指针类型的使用技巧以及需要注意的事项。

第 8 章 C 语言的控制流语句。这一章介绍了 C 语言的条件语句、选择语句以及循环等控制流语句。

第 9 章 C 语言的函数。这一章介绍了 C 语言中的函数概念，包括 C 语言函数的声明及定义，还有 C 函数的调用。此外还介绍了 C 语言函数标识符作为表达式时的类型。

第 10 章 C 语言的预处理器。这章包含了目前 C11 标准中所支持的所有预处理器特性，包括宏定义、预处理条件、预编译指示符与操作符以及 C 代码的注释。

第 11 章 C 语言的编译上下文。这一章介绍了 C 语言对象与函数的作用域和名字空间。详细介绍了 C 语言中的四大作用域以及在不同作用域中的对象的生命周期。此外还介绍了对象与函数的连接属性，包括外部连接和内部连接。

高级语法篇（第 12 ~ 16 章）讲述 C 语言的一些高级特性。这一部分内容不需要 C 语言程序员必须掌握，但需要对此有个大概了解。

第 12 章 C 语言中的类型限定符。该章介绍了 C11 标准中支持的 `const`、`volatile`、`restrict` 与 `_Atomic` 这四种限定符。详细说明了限定符用于修饰含有指针的对象时，在 * 号的不同位置所起到的不同作用。然后分别介绍这四种限定符的具体含义。

第 13 章 C 语言中的类型系统。这一章把 C 语言语法体系中的整个类型系统再梳理了一遍。这一章介绍了对于一些复杂类型的对象如何去剖析、理解，然后自己如何去声明自己想要的复杂类型的对象和函数。这一章所描述的其实是整个 C 语言语法体系的核心，如果大家能掌握的话，那么基本就算是真正掌握 C 语言了。其实，对于任一强类型的编程语言而言，其系统类型总是扮演着十分重要的角色，我们学习此类语言都需要透彻理解其整个类型系统。

第 14 章 C11 标准中的表达式、左值与求值顺序。该章先介绍了 C11 标准中各类表达式以及它们的计算优先级。然后介绍了“左值”这个概念，并讲解了表达式之间的求值顺序。

第 15 章 函数调用约定与 ABI。该章与 C 语言标准并无太大关系，但却与实际项目开发有关。这一章介绍了主流 C 语言编译器所采用的函数调用约定，然后详细描述了函数调用的过程，包括参数传递和返回值的具体处理。该章对嵌入式系统开发者以及需要将 C 语言与汇编语言进行交互使用的高性能计算开发者而言，将大为有用。

第 16 章 创建动态库与静态库。这一章介绍了用主流 C 语言编译工具构建静态库以及动态库的方法，并介绍如何使用这些库文件。

语法扩展篇（第 17 ~ 19 章）讲述了 GCC 与 Clang 编译器对 C 语言的扩展。

第 17 章 GCC 对 C11 标准的扩展。该章先简单介绍 GNU 语法扩展，然后介绍 GCC 编译器中常用的扩展语法。

第 18 章 Clang 编译器对 C11 标准的扩展。该章介绍了 Clang 编译器对 C11 标准的语法扩展。最后还介绍了 Apple 开源的 Grand Central Dispatch 库的简单使用。

第 19 章 对 C 语言的未来展望。该章主要介绍了 C 语言的设计理念以及当前 C 语言标准委员会的工作组正在为 C 语言新增的内容，还谈到了哪些特性不会被添加到 C 语言中去。

项目实践篇（第 20 ~ 21 章），这里通过两个实际的 C 语言项目来介绍我们如何利用 C 语言来创作出自己的程序。

第 20 章 描述了 UTF-8 编码格式的字符串与 UTF-16 编码格式的字符串进行相互转换的例子。

第 21 章 介绍一个看似简单而功能很丰富的基于控制台的计算器程序。

建议零基础的读者要了解第一篇的预备知识，这对于后面深入学习 C 语言编程很有帮助。另外，这部分读者可以先不用强行看第三篇，尤其是第 15 章。因为第三篇涉及的知识比较深，而第 15 章又会直接引入汇编语言，这对于没有一定计算机专业知识的读者会比较难以理解。如果是有一定计算机专业知识的读者可以略过第一篇，直接阅读第二篇。另外，如果是从事嵌入式系统开发的、或从事系统底层开发的资深程序员，建议仔细阅读第三、第四篇，相信这部分内容会对你的工作很有帮助。

勘误和支持

由于笔者的水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。如果你有更多的宝贵意见，欢迎你访问我的个人博客网站 http://blog.csdn.net/zenny_chen 进行专题讨论，我会尽量在线上为你提供最满意的解答。同时，你也可以通过微博 <http://weibo.com/zenny1chen> 与我联系，或发送电子邮件到 zenny_chen@163.com。期待能够得到你们的真挚反馈，在技术之路上互勉共进。另外，本书最后两章的代码可以在作者的 GitHub 上获取：<https://github.com/zenny-chen>。

致谢

首先感谢我的父母和妻子对我写作此书的大力支持，尤其是我妻子在我忙于工作、

编写此书时帮忙照顾孩子和做饭。然后感谢我公司老板对我写作此书的鼓舞与期待。

这里还要感谢机械工业出版社华章公司的编辑高婧雅，在一年多的时间里给予我的大力支持和帮助。

最后感谢支持我的技术爱好者，感谢你们对我的支持以及对我的信任。

我想和作者聊聊

为了更好地与读者进行联系，笔者这里留了一个 QQ 讨论群。各位如果在阅读此书有任何疑问可以来本群询问，大家可以一起探讨。各位可以扫一扫下方的二维码，进此群的提示语为：“C 语言编程魔法书”，或者查询群号 86540289 申请入群。



陈 轶

目 录

前言

第一篇 预备知识篇

第1章 C魔法概览 /2

1.1 例说编程语言	2
1.2 用C语言编程的基本注意事项	6
1.3 主流C语言编译器介绍	8
1.4 关于GNU规范的语法扩展	10
1.5 用C语言构建一个可执行程序流程	11
1.6 本章小结	12

第2章 学习C语言的预备知识 /14

2.1 计算机体系结构简介	14
2.1.1 寄存器	15
2.1.2 存储器	15
2.1.3 寄存器	16
2.1.4 计算单元	16
2.1.5 程序执行流程	16
2.2 整数在计算机中的表示	17
2.2.1 原码表示法	18
2.2.2 补码表示法	19
2.2.3 八进制数与十六进制数	20

2.3	浮点数在计算机中的表示	22
2.4	地址与字节对齐	24
2.5	字符编码	25
2.6	大端与小端	28
2.7	按位逻辑运算	28
2.8	移位操作	30
2.8.1	算术左移与逻辑左移	30
2.8.2	逻辑右移	30
2.8.3	算术右移	31
2.8.4	循环右移	32
2.9	本章小结	32

第3章 C语言编程的环境搭建 /33

3.1	Windows 操作系统下搭建 C 语言编程环境	33
3.1.1	安装 Visual Studio Community 2017	34
3.1.2	安装 MinGW 编译器	40
3.1.3	安装 LLVM Clang 编译器	43
3.2	macOS 系统下搭建 C 语言编程环境	45
3.3	本章小结	51

第二篇 基础语法篇

第4章 C语言中的基本元素 /54

4.1	C 语言中的字符集	55
4.2	C 语言中的 token	56
4.2.1	C 语言中的标识符	57
4.2.2	C 语言中的关键字	58
4.2.3	C 语言中的常量与字符串字面量	60
4.2.4	C 语言中的标点符号	60
4.3	关于 C 语言中的“对象”	62
4.4	C 语言中的“副作用”	63
4.5	C 语言标准库中的 printf 函数	63

4.6 本章小结	65
----------------	----

第5章 基本数据类型 /66

5.1 整数类型	66
5.1.1 int 类型	67
5.1.2 short 类型	68
5.1.3 long 类型	69
5.1.4 long long 类型	71
5.1.5 布尔类型	72
5.1.6 字符类型	73
5.1.7 宽字符以及 Unicode 字符类型	77
5.1.8 size_t 与 ptrdiff_t 类型	81
5.1.9 C 语言中的标准整数类型	82
5.2 浮点类型	84
5.3 数据精度与类型转换	86
5.3.1 整数晋升	87
5.3.2 带符号与无符号整数之间的转换	87
5.3.3 浮点数与浮点数的转换以及浮点数与整数之间的转换	90
5.4 C 语言基本运算操作符	92
5.4.1 加、减、乘、除与求模运算操作符	92
5.4.2 按位逻辑操作符	93
5.4.3 自增、自减操作符	94
5.4.4 关系操作符、相等性操作符与逻辑操作符	95
5.4.5 移位操作符	96
5.4.6 圆括号操作符	97
5.5 sizeof 操作符	98
5.6 投射操作符	99
5.7 本章小结	101

第6章 用户自定义类型 /102

6.1 枚举类型	102
6.2 结构体类型	105
6.2.1 结构体概述	105

6.2.2 用结构体创建对象并访问其成员	107
6.2.3 结构体复合字面量	110
6.3 联合体类型	113
6.4 位域	117
6.4.1 位域的一般特性	118
6.4.2 位域成员的存放与布局	120
6.4.3 匿名位域	122
6.4.4 位域使用示例	124
6.5 字节对齐与字节填充	125
6.5.1 <code>_Alignof</code> 操作符	125
6.5.2 <code>_Alignas</code> 对齐说明符	126
6.5.3 结构体成员的字节对齐与字节填充	129
6.6 复数类型	133
6.7 本章小结	135

第7章 C语言的数组与指针 /136

7.1 一维数组	136
7.2 多维数组	141
7.3 变长数组	145
7.4 一级指针与对象地址	147
7.4.1 地址与指针的基本概念	148
7.4.2 访问指针对象所指对象的内容	149
7.4.3 指针对象的其他操作	150
7.5 多级指针	153
7.6 指向用户自定义类型的指针	156
7.7 指针与数组的关系	159
7.8 指向数组的指针	162
7.9 <code>void</code> 类型、指向 <code>void</code> 类型的指针与空指针	165
7.10 字符数组与字符串字面量	167
7.11 完整与不完整类型	170
7.12 灵活的数组成员	171
7.13 本章小结	173

第8章 C语言的控制流语句 /174

8.1 逗号表达式	174
8.2 条件表达式	176
8.3 if-else 语句	176
8.4 switch-case 语句	179
8.5 while 与 do-while 迭代语句	182
8.6 for 迭代语句	187
8.7 goto 语句	189
8.8 本章小结	192

第9章 C语言的函数 /193

9.1 函数的声明与定义	194
9.2 函数调用与实现	196
9.2.1 函数调用的顺序点	197
9.2.2 函数的栈空间	200
9.2.3 函数的参数传递与返回	201
9.2.4 通过形参修改实参的值	204
9.3 数组类型作为函数形参	205
9.4 带有不定参数类型及个数的函数声明与调用	209
9.5 函数的递归调用	212
9.6 内联函数	218
9.7 函数的返回类型与无返回函数	221
9.8 指向函数的指针	223
9.9 C 语言中的主函数 main	226
9.10 函数与函数调用作为 sizeof 操作符	228
9.11 本章小结	229

第10章 C语言预处理器 /230

10.1 宏定义	231
10.1.1 宏的基本使用	232

10.1.2	宏定义中的 # 操作符	234
10.1.3	宏定义中的 ## 操作符	236
10.1.4	宏替换	238
10.1.5	可变参数的宏定义	240
10.2	C 语言中预定义的宏	243
10.2.1	C 语言强制要求的预定义宏	243
10.2.2	环境宏	244
10.2.3	条件特征宏	245
10.2.4	主流编译器及平台预定义的宏	246
10.3	条件预编译	247
10.4	源文件包含预处理指示符	251
10.5	#error 预处理指示符	256
10.6	#line 预处理指示符	256
10.7	#undef 预处理指示符	257
10.8	pragma 预编译指示符与操作符	258
10.9	空指示符与 C 语言中的程序注释	260
10.10	本章小结	262

第11章 C语言程序的编译上下文 /263

11.1	C 语言程序中的作用域和名字空间	263
11.1.1	文件作用域	264
11.1.2	函数作用域	265
11.1.3	函数原型作用域	266
11.1.4	语句块作用域	267
11.1.5	标识符的重定义与作用域的叠交	269
11.1.6	标识符的名字空间	272
11.2	全局对象与函数	275
11.3	静态对象与函数	277
11.4	局部对象	280
11.5	对象的存储与生命周期	282
11.6	_Thread_local 对象	285
11.7	本章小结	286

第三篇 高级语法篇

第12章 C语言中的类型限定符 /290

12.1	const 限定符	290
12.1.1	const 限定符修饰普通对象	291
12.1.2	const 限定符修饰数组元素	293
12.1.3	const 限定符修饰指针类型对象	294
12.1.4	const 限定符修饰函数形参类型为数组的对象	299
12.1.5	类型限定符的本质含义	301
12.2	volatile 限定符	302
12.3	restrict 限定符	304
12.4	_Atomic 限定符	307
12.5	本章小结	314

第13章 C语言的类型系统 /316

13.1	对象类型与函数类型	316
13.2	对声明符的进一步说明	318
13.3	更复杂的声明	322
13.3.1	将某一类型转换为指向该类型的指针	322
13.3.2	判定当前类型属于哪种对象类型	322
13.3.3	复杂复合类型的判断	324
13.4	typedef 类型定义	327
13.4.1	typedef 的一般使用	327
13.4.2	typedef 与类型限定符相结合的使用	330
13.4.3	用 typedef 来定义结构体与联合体的类型	332
13.5	本章小结	335

第14章 C11标准中的表达式、左值与求值顺序 /336

14.1	常量表达式	338
14.2	泛型选择表达式	340
14.3	静态断言	343
14.4	C 语言中的左值	344

14.5 C 语言中表达式的求值顺序	348
14.6 C 语言中的语句	352
14.7 本章小结	355

第15章 函数调用约定与ABI /356

15.1 Windows 操作系统环境下 x86 处理器的函数调用约定	357
15.1.1 Windows 操作系统下 32 位 x86 执行模式的函数调用约定	357
15.1.2 Windows 操作系统下 64 位 x86 执行模式的函数调用约定	363
15.2 Unix/Linux 操作系统环境下 x86 处理器的函数调用约定	368
15.3 ARM 处理器环境下的函数调用约定	370
15.3.1 AArch32 架构环境下的函数调用约定	370
15.3.2 AArch64 架构环境下的函数调用约定	373
15.4 本章小结	375

第16章 创建静态库与动态库 /376

16.1 Windows 系统下创建静态库与动态库	377
16.1.1 Windows 系统下创建并使用静态库	377
16.1.2 Windows 系统上创建并使用动态库	379
16.2 macOS 系统下创建静态库与动态库	382
16.2.1 macOS 系统下创建并使用静态库	383
16.2.2 macOS 系统下创建并使用动态库	386
16.3 Linux 系统下创建并使用静态库与动态库	390
16.3.1 Linux 系统下创建并使用静态库文件	390
16.3.2 Linux 系统下创建并使用动态库	391
16.4 本章小结	393

第四篇 语法扩展篇

第17章 GCC对C11标准的语法扩展 /396

17.1 在表达式中使用复合语句与声明	397
17.2 声明语句块作用域的跳转标签	398