



新浪微博：C语言图书
腾讯QQ：4006751066

C语言学习路线图



C语言开发入门 及项目实战

快速服务：微博、QQ在线服务

自学视频：72集大型多媒体自学视频

海量资源：模块库、案例库、素材库、题库



明日科技 编著

本书提供了内容丰富的配套资源，可以登录www.tup.com.cn，找到本书后，在该页面的“网络资源”超链接处下载。也可以访问本书的新浪微博，根据提示链接下载。

清华大学出版社

C 语言学习路线图

C 语言开发入门及项目实战

明日科技 编著

（72 集大型多媒体自学视频）

（模块库、案例库、素材库、题库）

（微博、QQ、论坛技术支持）

清华大学出版社

北 京

内 容 简 介

本书全面介绍了使用 C 语言进行编程的各种技术,是学习 C 语言编程技术的完美教程。全书以 Visual C++ 6.0 开发环境为基础,分 3 篇 18 章讲解了 C 语言编程中的各种技术。其中,第 1 篇为入门篇,主要包括初识 C 语言、一个简单的 C 语言程序、算法入门、常用数据类型、运算符与表达式、数据输入/输出函数、选择及分支结构程序设计、循环结构、数组、函数编程基础、指针解析等内容;第 2 篇为提高篇,主要包括常用数据结构、位运算、存储管理、预处理和函数类型、文件的读写操作、图形图像处理等内容;第 3 篇为应用篇,主要利用所学知识开发一个图书管理系统。

本书提供了大量的自学视频、源程序、素材,提供了相关的模块库、案例库、素材库、题库等多种形式辅助学习资料,还提供迅速及时的微博、QQ、论坛等技术支持。

本书内容详尽、实例丰富,非常适合作为零基础学习人员的学习用书和大中专院校师生的学习教材,也适合作为相关培训机构的师生和软件开发人员的参考资料。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言开发入门及项目实战/明日科技编著. —北京:清华大学出版社,2012.2
(C 语言学习路线图)

ISBN 978-7-302-27668-5

I. ①C… II. ①明… III. ①C 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 267918 号

责任编辑:赵洛育 刘利民

版式设计:文森时代

责任校对:王 云

责任印制:

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:

装 订 者:

经 销:全国新华书店

开 本:185mm×260mm 印 张:30 字 数:866 千字

版 次:2012 年 2 月第 1 版 印 次:2012 年 2 月第 1 次印刷

印 数:1~5000

定 价:59.80 元

产品编号:045362-01

前言

Preface

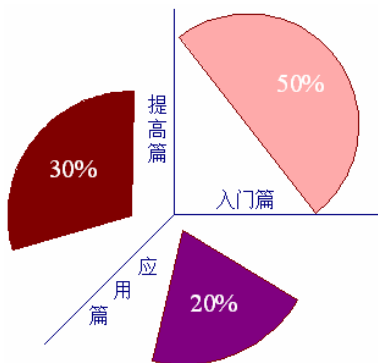


欢迎来到 C 语言的世界，并通过本书学习 C 语言。本书内容反映了 C 语言的最新变化，其编排方式非常适合学习和讲授这门语言。

C 语言是一种面向过程的编程语言，可用于开发各种应用程序及系统软件，近几年，C 语言的发展呈逐年上升趋势，这也说明了 C 语言的简单、现代特点正被更多人认同。

本书内容

本书提供了 C 语言编程从入门到项目开发所必备的知识，并且分篇进行讲解，使读者的学习更有针对性。学习本书内容时，一定要熟练掌握入门篇的所有内容，以便为以后的应用开发打下坚实的基础；然后可以有针对性地学习提高篇的内容，为开发专业软件项目存储必备的知识；最后学习一个项目的完整开发过程，为以后的实际项目开发积累一定的经验。学习本书内容时，各篇内容需要花费的精力比例如下图所示。



第 1 篇 入门篇（第 1~11 章）：主要包括初识 C 语言、一个简单的 C 语言程序、算法入门、常用数据类型、运算符与表达式、数据输入/输出函数、选择及分支结构程序设计、循环结构、数组、函数编程基础、指针解析等内容。通过本篇的学习，读者应熟练掌握 C 语言编程的基础知识，并能开发一些小型的应用程序。

第 2 篇 提高篇（第 12~17 章）：主要包括常用数据结构、位运算、存储管理、预处理和函数类型、文件的读写操作、图形图像处理等内容。通过本篇的学习，读者可以开发基本的应用程序，并能够掌握开发综合应用程序的方法。

第 3 篇 应用篇（第 18 章）：本篇通过一个完整的图书管理系统，运用软件工程的设计思想，讲解如何进行软件项目的实践开发。书中按照编写项目计划书→系统设计→数据库设计→创建项目→实现项目→运行项目→解决开发常见问题过程进行介绍，带领读者一步一步亲身体验开发项目的全过程。



本书特点

☑ 技术新颖，讲解细致

使用 Visual C++ 6.0 和 Turbo C 2.0 进行讲解，淋漓尽致地展示出了 C 语言的基础知识，使读者能够真正掌握 C 语言初学者应该学习的知识。

☑ 实例丰富，贴近实际

对每个技术点进行详细讲解时，都提供了大量的示例和实例进行演示，这些示例和实例大部分来源于实际开发，读者可以学为所用，快速提升自身开发技能，并有效解决实际开发中遇到的各类问题。

☑ 动手实践，一体学习

每章都提供了大量的上机演练和实战模拟，使读者在学习技术的同时，能够亲自动手实践，体验编程带来的成就感。另外，上机演练和实战模拟的详细讲解及源码与《C 语言经典编程 282 例》中的实例相对应，读者可以根据自身实际情况进行参考。

☑ 注释详尽，视频讲解

为了便于读者更好地学习和使用本书，书中所有的代码都提供了详尽的注释。而且，本书配套资源提供了覆盖全书的语音视频讲解教学录像，读者可以通过视频快速、直观、轻松地学习。

本书配套资源

本书提供了内容丰富的配套资源，包括自学视频、源程序、素材，以及模块库、案例库、题库、素材库等多项辅助内容，读者朋友可以通过如下方式获取。

第 1 种方式：

(1) 登录 www.tup.com.cn，在网页右上角的搜索文本框中输入本书书名（注意区分大小写和留出空格），或者输入本书关键字，或者输入本书 ISBN 号（注意去掉 ISBN 号间隔线“-”），单击“搜索”按钮。

(2) 找到本书后单击超链接，在该书的网页下侧单击“网络资源”超链接，即可下载。

第 2 种方式：

访问本书的新浪微博：C 语言图书，找到配套资源的链接地址进行下载。

读者对象

- ☑ 大中专院校师生
- ☑ 零基础学习人员
- ☑ 初、中级程序开发人员
- ☑ 编程爱好者

- ☑ 面临就业的学生
- ☑ 相关培训机构的老师和学员
- ☑ 准备从事软件开发工作的求职者
- ☑ 立志编程的其他专业人士



读者服务&本书勘误

本书的“专项技能测试”提供了答案，可以从如下所述方式获得。

读者在使用本书过程中遇到的所有问题，均可通过以下方式联系我们。

1. 新浪微博：C 语言图书。

及时发布读者答疑、本书勘误、配套资料更新等内容。

2. 腾讯 QQ：4006751066。

3. 登录网站：www.mingribook.com，在论坛、勘误发布、读者纠错、技术支持、读者之家等栏目中的相关模块中提问、留言或查看。

关于作者

本书由明日科技 C 语言程序开发团队组织编写，主要编写人员有李鑫、王雪、孙秀梅、曹飞飞、朱晓、王小科、赵会东、聂喜婷、宋环雨、李继业、高春艳、赛奎春、杨丽、肖鑫、刘龄龄、张振坤、王国辉、陈丹丹、潘凯华、刘欣、李慧、陈英等。在编写本书的过程中，我们以科学、严谨的态度，力求精益求精，但错误、疏漏之处在所难免，敬请广大读者批评指正。

最后，感谢您选择本书，希望本书能成为您学习 C 语言编程路上的领航者。

编 者

目 录

Contents



第 1 篇 入门篇

第 1 章 初识 C 语言

(自学视频、源程序:

配套资源\mr\1\) 2

1.1 编程语言的发展 3

1.1.1 机器语言 3

1.1.2 汇编语言 3

1.1.3 高级语言 4

1.2 C 语言的历史 5

1.2.1 C 语言的产生 5

1.2.2 C 程序的特点 6

1.2.3 C 语言的发展趋势 6

1.2.4 C 语言的应用领域 7

1.2.5 如何学好 C 语言 7

1.3 本章小结 8

1.4 技能专项测试 8

1.5 实战模拟 10

第 2 章 一个简单的 C 语言程序

(自学视频、源程序:

配套资源\mr\2\) 11

2.1 一个简单的例子 12

上机演练 15

2.2 认识 C 语言程序 16

2.2.1 认识 C 语言程序的开发

环境 16

上机演练 23

2.2.2 认识标识符 23

2.2.3 认识关键字 25

2.2.4 认识注释 26

2.2.5 认识变量 27

2.3 编译错误信息 27

2.3.1 严重错误信息及分析 27

2.3.2 一般错误信息及分析 27

2.3.3 警告信息及分析 34

2.4 提高程序的可读性 35

2.5 本章小结 36

2.6 技能专项测试 36

2.7 实战模拟 38

第 3 章 算法入门

(自学视频、源程序:

配套资源\mr\3\) 39

3.1 算法 40

3.1.1 什么是算法 40

3.1.2 算法的特性 40

3.1.3 算法的好坏 41

3.2 算法的描述 42

3.2.1 自然语言 42

3.2.2 流程图 42

3.2.3 3 种基本结构 43

3.2.4 N-S 流程图 46

3.2.5 伪代码 50

3.3 本章小结 50

3.4 技能专项测试 51

3.5 实战模拟 52

第 4 章 常用数据类型

(自学视频、源程序:

配套资源\mr\4\) 53

4.1 数据类型 54

4.1.1 什么是数据类型 54

4.1.2 数据类型的分类 54

4.2 常量 55



Note

4.2.1 整型常量.....	55	5.5.1 逻辑运算符.....	99
4.2.2 实型常量.....	57	5.5.2 逻辑表达式.....	99
4.2.3 字符型常量.....	58	5.5.3 优先级与结合性.....	101
4.2.4 符号常量.....	64	①上机演练.....	101
①上机演练.....	65	5.6 位逻辑运算符与表达式.....	102
4.3 变量.....	66	5.6.1 位逻辑运算符.....	102
4.3.1 整型变量.....	66	5.6.2 位逻辑表达式.....	102
4.3.2 实型变量.....	69	5.7 逗号运算符与表达式.....	103
4.3.3 字符型变量.....	72	5.8 条件运算符与表达式.....	104
①上机演练.....	74	5.9 复合赋值运算符.....	105
4.4 变量的存储类别.....	74	5.10 本章小结.....	106
4.4.1 动态存储与静态存储.....	74	5.11 技能专项测试.....	107
4.4.2 auto 变量.....	75	5.12 实战模拟.....	109
4.4.3 static 变量.....	76		
4.4.4 register 变量.....	78	第6章 数据输入/输出函数	
4.4.5 extern 变量.....	79	(自学视频、源程序:	
4.5 混合运算.....	79	配套资源\mr\6\).....	110
4.6 本章小结.....	81	6.1 语句.....	111
4.7 技能专项测试.....	81	6.1.1 表达式语句.....	111
4.8 实战模拟.....	83	6.1.2 函数调用语句.....	111
		6.1.3 空语句.....	112
		6.1.4 控制语句.....	113
		6.1.5 复合语句.....	113
第5章 运算符与表达式		6.2 字符数据输入/输出.....	114
(自学视频、源程序:		6.2.1 字符数据输出.....	114
配套资源\mr\5\).....	84	6.2.2 字符数据输入.....	115
5.1 表达式.....	85	6.3 字符串输入/输出.....	117
5.2 赋值运算符与赋值表达式.....	87	6.3.1 字符串输出函数.....	117
5.2.1 变量赋初值.....	87	6.3.2 字符串输入函数.....	118
5.2.2 自动类型转换.....	88	6.4 格式输出函数.....	119
5.2.3 强制类型转换.....	89	6.5 格式输入函数.....	123
5.3 算术运算符与表达式.....	90	6.6 顺序程序设计应用.....	126
5.3.1 算术运算符.....	91	①上机演练.....	128
5.3.2 算术表达式.....	91	6.7 本章小结.....	128
5.3.3 优先级与结合性.....	92	6.8 技能专项测试.....	129
5.3.4 自增自减运算符.....	94	6.9 实战模拟.....	131
①上机演练.....	96		
5.4 关系运算符与表达式.....	96	第7章 选择及分支结构程序设计	
5.4.1 关系运算符.....	96	(自学视频、源程序:	
5.4.2 关系表达式.....	97	配套资源\mr\7\).....	132
5.4.3 优先级与结合性.....	97	7.1 if 语句.....	133
5.5 逻辑运算符与表达式.....	99		



7.2 if 语句的基本形式	133	8.7.1 goto 语句	180
7.2.1 if 语句形式	133	8.7.2 break 语句	182
7.2.2 if...else 语句形式	136	8.7.3 continue 语句	183
7.2.3 else if 语句形式	139	①上机演练	183
①上机演练	142	8.7.4 区别 3 种跳转语句	184
7.3 if 的嵌套形式	142	8.8 本章小结	184
7.4 switch 语句	145	8.9 技能专项测试	185
7.4.1 switch 语句的基本形式	145	8.10 实战模拟	188
7.4.2 多路开关模式的 switch 语句	149		
7.5 if...else 语句和 switch 语句的 区别	151	第 9 章 数组	
7.6 选择结构程序应用	153	(自学视频、源程序: 配套资源\mr\9\)	189
7.7 本章小结	154	9.1 数组的概念	190
7.8 技能专项测试	155	9.1.1 数组的分类	190
7.9 实战模拟	158	9.1.2 数组的维数	191
		9.2 一维数组	191
第 8 章 循环结构		9.2.1 一维数组的定义	191
(自学视频、源程序: 配套资源\mr\8\)	159	9.2.2 一维数组的引用	193
8.1 循环语句	160	9.2.3 一维数组的初始化	194
8.2 while 型循环	160	9.2.4 一维数组的应用	196
8.2.1 while 语句的一般形式	161	9.2.5 一维数组的注意事项	198
8.2.2 while 语句的死循环	164	9.3 二维数组	198
8.2.3 while 语句的空循环体	165	9.3.1 二维数组的定义	199
8.2.4 while 循环语句的常见 错误	165	9.3.2 二维数组的引用	200
①上机演练	166	9.3.3 二维数组的初始化	200
8.3 do...while 型循环	167	9.3.4 二维数组的应用	202
8.4 for 型循环	169	①上机演练	203
8.4.1 for 循环语句的一般形式	169	9.4 多维数组	204
8.4.2 灵活多变的 for 循环语句	172	9.5 字符数组	204
8.4.3 应用 for 语句中的逗号	174	9.5.1 字符数组的定义	204
①上机演练	175	9.5.2 字符数组的引用	204
8.5 比较 3 种循环	176	9.5.3 字符数组的初始化	205
8.6 运用循环嵌套	176	9.5.4 字符数组的结束标志	206
8.6.1 各种结构的循环嵌套	176	9.5.5 字符数组的输出	207
8.6.2 循环嵌套实例	178	9.5.6 字符数组的输入	208
①上机演练	179	9.5.7 字符数组的应用	209
8.7 转移语句	179	①上机演练	210
		9.6 字符串处理函数	210
		9.6.1 字符串复制	211
		9.6.2 字符串连接	212



9.6.3	字符串比较.....	213
9.6.4	字符串大小写转换.....	215
9.6.5	获得字符串长度.....	216
9.7	熟练应用数组.....	217
9.7.1	实现字符匹配.....	217
9.7.2	反转输出字符串.....	218
9.8	本章小结.....	219
9.9	技能专项测试.....	219
9.10	实战模拟.....	221
第 10 章 函数编程基础		
(自学视频、源程序: 配套资源\mr\10\)..... 222		
10.1	函数概述.....	223
10.2	函数的定义.....	225
10.2.1	函数定义的形式.....	225
10.2.2	定义与声明.....	228
上机演练.....		229
10.3	了解返回语句.....	230
10.3.1	从函数返回.....	230
10.3.2	返回值.....	234
10.4	传递函数参数.....	236
10.4.1	形式参数与实际参数.....	236
10.4.2	数组做函数参数.....	238
10.4.3	了解 main()函数的参数.....	242
10.5	调用函数.....	242
10.5.1	函数调用方式.....	243
10.5.2	嵌套调用.....	245
10.5.3	递归调用.....	247
上机演练.....		249

10.6	灵活应用函数.....	249
10.7	本章小结.....	255
10.8	技能专项测试.....	256
10.9	实战模拟.....	258

第 11 章 指针解析

(自学视频、源程序: 配套资源\mr\11\)..... 259	
11.1	指针概述..... 260
11.1.1	地址与指针..... 260
11.1.2	解读变量与指针..... 261
11.1.3	使用指针变量..... 261
11.1.4	指针自加自减运算..... 265
上机演练.....	266
11.2	数组与指针..... 266
11.2.1	一维数组与指针..... 266
11.2.2	二维数组与指针..... 271
11.2.3	字符串与指针..... 273
11.2.4	字符串数组..... 274
上机演练.....	276
11.3	指向指针的指针..... 276
上机演练.....	278
11.4	函数参数的指针变量..... 278
上机演练.....	286
11.5	返回指针值的函数..... 286
11.6	指针数组作 main()函数的 参数..... 288
11.7	本章小结..... 289
11.8	技能专项测试..... 289
11.9	实战模拟..... 292

第 2 篇 提高篇

第 12 章 常用数据结构

(自学视频、源程序: 配套资源\mr\12\)..... 294	
12.1	结构体..... 295
12.1.1	结构体类型的概念..... 295
12.1.2	结构体变量的定义..... 296
12.1.3	结构体变量的引用..... 298

12.1.4	结构体类型的初始化..... 300
12.2	结构体数组..... 301
12.2.1	定义结构体数组..... 301
12.2.2	初始化结构体数组..... 302
上机演练.....	304
12.3	结构体指针..... 304
12.3.1	指向结构体变量的指针..... 304



12.3.2 指向结构体数组的指针	307	13.5 本章小结	346
12.3.3 结构体作函数参数	308	13.6 技能专项测试	346
12.4 包含结构的结构	311	13.7 实战模拟	348
12.5 链表	312	第 14 章 存储管理	
12.5.1 链表概述	312	(自学视频、源程序:	
12.5.2 创建动态链表	313	配套资源\mr\14\)	349
12.5.3 输出链表	315	14.1 内存组织方式	350
12.6 链表相关操作	318	14.1.1 内存组织方式概述	350
12.6.1 链表的插入操作	318	14.1.2 堆与栈	350
12.6.2 链表的删除操作	319	14.2 动态管理	352
12.7 共用体	322	14.2.1 malloc() 函数	353
12.7.1 共用体的概念	323	14.2.2 calloc() 函数	353
12.7.2 共用体变量的引用	323	14.2.3 realloc() 函数	354
12.7.3 共用体变量的初始化	324	14.2.4 free() 函数	355
12.7.4 共用体类型的数据特点	325	上机演练	356
上机演练	325	14.3 内存丢失	356
12.8 枚举类型	326	14.4 本章小结	357
12.9 本章小结	327	14.5 技能专项测试	357
12.10 技能专项测试	327	14.6 实战模拟	359
12.11 实战模拟	329	第 15 章 预处理和函数类型	
第 13 章 位运算		(自学视频、源程序:	
(自学视频、源程序:		配套资源\mr\15\)	360
配套资源\mr\13\)	330	15.1 宏定义	361
13.1 位与字节	331	15.1.1 不带参数的宏定义	361
13.2 位运算操作符	331	15.1.2 带参数的宏定义	362
13.2.1 “与”运算符	332	上机演练	363
13.2.2 “或”运算符	333	15.2 #include 指令	364
13.2.3 “取反”运算符	335	15.3 条件编译	365
13.2.4 “异或”运算符	336	15.3.1 #if 命令、#else 命令和 #elif	
13.2.5 “左移”运算符	338	命令	366
13.2.6 “右移”运算符	339	15.3.2 #ifdef 及 #ifndef 命令	368
13.2.7 位运算复合赋值运算符	340	15.3.3 #undef 命令	369
13.2.8 长度不同的数据进行位		15.3.4 #line 命令	369
运算	341	15.3.5 #pragma 命令	370
上机演练	341	15.4 内部函数和外部函数	371
13.3 循环移位	341	15.4.1 内部函数	371
上机演练	343	15.4.2 外部函数	372
13.4 位段	343	15.5 本章小结	373
13.4.1 位段的概念与定义	343	15.6 技能专项测试	373
13.4.2 位段相关说明	345		



15.7 实战模拟	376	16.4.3 得到文件位置——ftell() 函数	395
第 16 章 文件的读写操作 (自学视频、源程序: 配套资源\mr\16\)	377	16.5 本章小结	397
16.1 文件概述	378	16.6 技能专项测试	397
16.2 文件基本操作	379	16.7 实战模拟	399
16.2.1 文件指针	379	第 17 章 图形图像处理 (自学视频、源程序: 配套资源\mr\17\)	400
16.2.2 文件的打开	380	17.1 字符屏幕	401
16.2.3 文件的关闭	382	17.1.1 定义文本窗口	401
16.3 读写文件	382	17.1.2 颜色设置	401
16.3.1 写字符函数——fputc() 函数	382	17.1.3 文本的输入和输出	402
16.3.2 读字符函数——fgetc() 函数	383	17.1.4 屏幕操作函数	403
16.3.3 写字符串函数——fputs() 函数	384	①上机演练	405
16.3.4 读字符串函数——fgets() 函数	385	17.2 图形显示	406
16.3.5 格式化写入函数——fprintf() 函数	387	17.2.1 图形模式初始化	406
16.3.6 格式化读取函数——fscanf() 函数	388	17.2.2 屏幕颜色设置	408
16.3.7 数据块读写函数——fread() 函数和 fwrite()函数	389	17.2.3 基本图形函数	410
①上机演练	391	17.2.4 封闭图形的填充	417
16.4 定位文件	392	①上机演练	419
16.4.1 随机读写操作——fseek() 函数	392	17.3 图形屏幕	419
16.4.2 返回位置指针——rewind() 函数	394	17.4 图形模式下文本输出	421
		17.4.1 文本输出函数	421
		17.4.2 文本属性设置	422
		①上机演练	424
		17.5 本章小结	424
		17.6 技能专项测试	424
		17.7 实战模拟	426

第 3 篇 应用篇

第 18 章 图书管理系统 (自学视频、源程序: 配套资源\mr\18\)	428	18.2.1 系统目标	429
18.1 概述	429	18.2.2 系统功能结构	429
18.1.1 需求分析	429	18.2.3 系统预览	429
18.1.2 开发工具选择	429	18.2.4 开发及运行环境	430
18.2 系统设计	429	18.3 数据库设计	431
		18.3.1 安装 MySQL 数据库	431
		18.3.2 启动 MySQL 数据库	434



18.3.3 创建数据库.....	435	18.5.6 删除图书表记录.....	444
18.3.4 数据表结构.....	436	18.6 文件引用	445
18.4 C 语言开发数据库程序的 流程	436	18.7 变量和函数定义	446
18.5 C 语言操作 MySQL 数据库....	438	18.8 主要功能模块设计	446
18.5.1 MySQL 常用数据库操作 函数	438	18.8.1 显示主菜单信息.....	446
18.5.2 连接 MySQL 数据.....	440	18.8.2 显示所有图书信息.....	448
18.5.3 查询图书表记录.....	440	18.8.3 添加图书信息	450
18.5.4 插入图书表记录.....	442	18.8.4 修改图书信息	454
18.5.5 修改图书表记录.....	443	18.8.5 删除图书信息	459
		18.8.6 查询图书信息	463
		18.9 本章小结	464



入门篇

- ▶▶ 第 1 章 初识 C 语言
- ▶▶ 第 2 章 一个简单的 C 语言程序
- ▶▶ 第 3 章 算法入门
- ▶▶ 第 4 章 常用数据类型
- ▶▶ 第 5 章 运算符与表达式
- ▶▶ 第 6 章 数据输入/输出函数
- ▶▶ 第 7 章 选择及分支结构程序设计
- ▶▶ 第 8 章 循环结构
- ▶▶ 第 9 章 数组
- ▶▶ 第 10 章 函数编程基础
- ▶▶ 第 11 章 指针解析

第1章

初识 C 语言

( 自学视频、源程序：配套资源\mr\1\)

本章不介绍 C 语言的具体技术，只介绍和 C 语言相关的一些内容。计算机语言有机器语言、汇编语言和高级语言 3 种，C 语言属于高级语言。C 语言不仅在过去非常流行，现在在编程语言的排行榜中也是稳居前三，发展前景非常可观。正是因为 C 语言有使用方便、功能强大、应用广泛、移植性好等优点，所以 C 语言不仅可以编写应用软件，而且非常适合编写系统软件。

当读者了解到 C 语言的历史和特性，就会对 C 语言产生浓厚的兴趣，真正从心里喜欢上这门语言。本章致力于让读者对 C 语言有个初步的认识，能够对 C 语言有所了解。

学习摘要：

- » 了解编程语言的发展
- » 了解 C 语言的产生
- » 了解 C 语言的特点
- » 了解 C 语言的发展趋势
- » 了解 C 语言的应用领域
- » 了解如何学好 C 语言



1.1 编程语言的发展

计算机程序设计语言，通常简称为编程语言，是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应采取的行动。计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程，如图 1.1 所示。

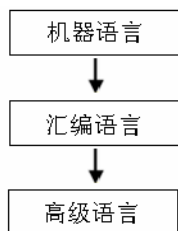


图 1.1 计算机语言的发展历程

1.1.1 机器语言

机器语言是直接用二进制代码指令表达的计算机语言，指令是用 0 和 1 组成的一串代码，它们有一定的位数，并分成若干段，各段的编码表示不同的含义。例如，某台计算机字长为 16 位，即由 16 个二进制数组成一条指令或其他信息。16 个 0 和 1 可组成各种排列组合，通过线路变成电信号，让计算机执行各种不同的操作。如某种计算机的指令为 1011011000000000，它表示让计算机进行一次加法操作；而指令 1011010100000000 则表示进行一次减法操作。它们的前 8 位表示操作码，而后 8 位表示地址码。从上面两条指令可以看出，它们只是在操作码中从左边第 0 位算起的第 6 和第 7 位不同。这种机型可包含 $256 (2^8)$ 个不同的指令。如图 1.2 所示为一种存放机器语言指令的 CPU 芯片。

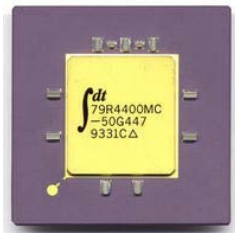


图 1.2 存放机器语言指令的 CPU 芯片

由于机器语言是 CPU 直接使用的语言，与人类平日使用的语言差异太大，因此被称为“低级语言”。机器语言如下：

01	00110100	10100001	10001000	11011001
02	11000101	10100100	11010101	11101010
03	11001001	10101100	10010001	10100001
04				

机器语言具有灵活、直接执行和速度快等特点，缺点是程序长、难记、难理解、不易查错。不同型号的计算机其机器语言是不相通的，按照一种计算机的机器指令编制的程序，不能在另一种计算机上执行。

1.1.2 汇编语言

机器语言虽然执行速度快，但难记、不易理解。为了提高编程效率，人们开始对机器语言进行改进，用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串。例如，在上面提到的机器码指令 1011011000000000 在汇编语言中用 ADD 代替，代表加法；机器码指令 1011010100000000 在汇编语言中用 SUB 代替，代表减法。这种用助记符描述的指令系统的语言，就是汇编语言（Assembly Language）。汇编语言相比机器语言，更容易理解，例如，要计算 $c=3+5$ ，用几条汇编命令即可实现：



Note

标号	指令	说明
START	GET 3;	把 3 送进累加器 ACC 中
	ADD 5;	把累加器 ACC+5 送进累加器 ACC 中
	PUT C;	把累加器 ACC 送进 C 中
END	STOP;	停机

汇编语言是一种功能很强的程序设计语言，也是利用计算机所有硬件特性并能直接控制硬件的语言。汇编语言属于第二代计算机语言。

使用汇编语言编写程序或阅读已经编写好的程序比起机器语言要简单和方便多了。人们使用这种助记符编写程序后，要使计算机能够接受，还必须把编好的程序逐条翻译成二进制编码的机器语言，这个过程需要借助“汇编程序”来完成。汇编程序的功能就是把由汇编语言编写的程序翻译成机器语言程序，使计算机能够执行该程序。这个翻译过程称为汇编。

汇编语言比起机器语言在很多方面都有很大的优越性，如编写容易、修改方便、阅读简单、程序清楚等，但在计算机语言系统中，汇编语言仍然被列入“低级语言”的范畴，它仍然是属于面向机器的语言，也就是说，不同的计算机可以有不同的指令集。

汇编语言的应用如下：

- ☒ 70%以上的系统软件是用汇编语言编写的。
- ☒ 某些快速处理、位处理、访问硬件设备等高效程序是用汇编语言编写的。
- ☒ 某些高级绘图程序、视频游戏程序是用汇编语言编写的。
- ☒ 80%的病毒来自汇编语言。

虽然汇编语言相比机器语言更进了一步，但汇编语言与机器语言很像，都是直接面向机器的，仍然是符号形式的机器语言，执行效率和理解上都很差。

1.1.3 高级语言

无论是机器语言，还是汇编语言，编程过程都是非常枯燥和费时的。这时人们意识到，如果采用接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用，这样编程就容易多了。经过努力，第一个现代意义上的编程语言 Fortran 于 1954 年出现，因其不再过度地依赖某种特定的机器或环境，所以这种语言被叫做高级语言。到目前为止，共有几百种高级语言出现，有重要意义的有几十种。

面向电子计算机的高级语言的出现是计算机发展史上的里程碑式事件。目前的高级语言分为面向过程和面向对象两种。

☒ 面向过程的高级语言

面向过程的高级语言，又称算法语言，是围绕实现功能进行的，用一个函数实现一个功能。如 Fortran、Basic、Pascal、C 等。

优点：灵活，易于理解，容易查错。

缺点：维护性较差，难以扩充、修改。

☒ 面向对象的高级语言

面向对象的高级语言是非面向过程的语言，所有数据属于不同的对象，把数据和操作代码封装在一个对象中。如 C++、Java、PHP、C#、Delphi 等。

优点：灵活，易于理解、维护、修改和扩充。

缺点：掌握难度较大。

使用高级语言编写出的可供人阅读的程序叫做源程序（Source Program），也可以称为源