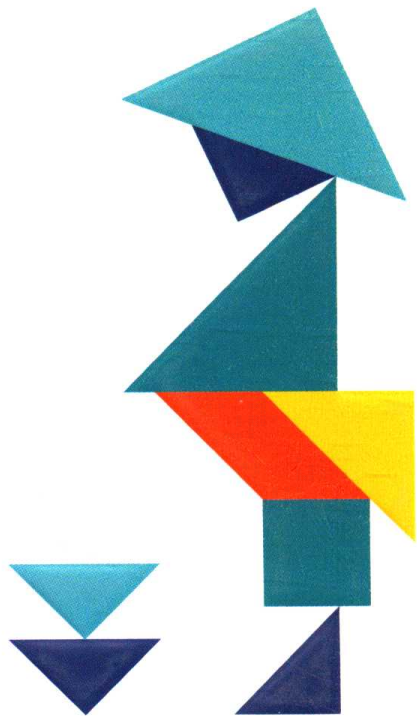


21世纪普通高校计算机
公共课程规划教材

C语言 程序设计



© 孙海洋 编著

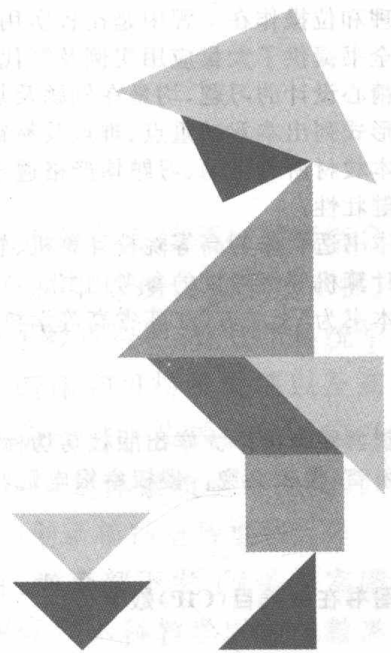


清华大学出版社



21世纪普通高校计算机
公共课程规划教材

C语言 程序设计



◎ 孙海洋 编著

清华大学出版社
北京

内 容 简 介

全书共分为 11 章,前 5 章主要介绍了 C 语言的基本语法、基本数据类型、运算符及表达式和三大流程结构,第 6 章引入批量处理数据的类型——数组。本教材从第 7 章开始对函数、指针、自定义类型等 C 语言的精髓和核心进行了重点介绍。第 10 章输入和输出也是综合性且实用性较强的重点章节,第 11 章预处理和位操作在编程中是比较实用的,例如,解决了头文件重复包含等问题及相关位操作。

全书提供了大量应用实例及源代码,每节均对应复习思考题,便于对所学知识及时巩固提高。每章有大量精心设计的习题,均是在例题及复习思考题基础上的提升,且按章节按知识点划分。每章小结均以表格的形式列出本章的重点、难点及易错点,结构清晰,便于读者学习把握。

本教材所有例题、习题均严格遵守业界较通用的编程规范,设计结构合理,思路清晰,注重程序的可读性和健壮性。

本书适合作为高等院校计算机、软件工程、电子、通信、自动化等专业高年级本科生的教材,同时也可作为计算机等级考试的参考用书。

本书为“十三五”江苏省高等学校重点教材,重点教材编号为 2017-2-021。

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 语言程序设计/孙海洋编著. —北京:清华大学出版社,2018

(21 世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-48391-5

I. ①C… II. ①孙… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 216424 号

责任编辑:贾 斌 薛 阳

封面设计:刘 键

责任校对:时翠兰

责任印制:丛怀宇

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市君旺印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:24.75

字 数:603 千字

版 次:2018 年 7 月第 1 版

印 次:2018 年 7 月第 1 次印刷

印 数:1~1500

定 价:59.00 元

产品编号:072914-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前言

未来的世界是自动化一统天下的时代,智能控制离不开程序,由于C语言是所有高级程序设计语言中最接近硬件且可移植性较好的高效编程语言,其在系统编程和嵌入式开发中具有不可比拟的优势。而且学习C语言是进一步学习其他高级编程语言,如C++和Java的基础。

【本书主要内容】

本书共分为11章,内容包括C语言概述、顺序结构程序设计、运算符与表达式、分支结构、循环结构、数组、函数、指针、自定义类型、输入和输出、预处理和位操作等。

本教材在讲解C语言基础知识时,力求简单、明了,并均有对应实例。而对于C语言的精髓部分,如函数、指针、自定义类型等做了由浅入深的详细讲解,并设计了大量的例题和实例。

从文件中获取数据进行处理,然后把处理的结果保存到文件中,这可能是多数读者通过学习C语言渴望掌握的技能。故本教材把“输入和输出”作为非常重要的章节,该章节涉及对整本教材的复习巩固提高。该章配以大量的常用实例,提高读者的学习兴趣。

第11章“预处理和位操作”通常是被C语言读者忽略的部分,而该部分在实际的程序开发中占有非常重要的地位,如涉及带参、无参宏定义,以及如何使用预处理命令避免头文件的重复包含等问题。如果能使用位运算代替程序中的乘除等算术运算,将提高运算效率,且能体现掌握C语言的深度。

【本书特色】

(1) 把枯燥、复杂的语法概念简单化、实例化。本教材几乎对涉及的所有知识点均设计实例进行讲解,通俗易懂,便于读者自学。

(2) 例题设计具有代表性,把在实际程序开发过程中经常遇到的错误及不规范的语法,均以例题的形式进行分析总结,分析思路详尽,所有实例均提供了源代码,便于读者使用。

(3) 每一节讲解完均有对应的复习思考题,便于对本节重点知识及时巩固提高。

(4) 分章节分知识点设计的课后习题结构,几乎覆盖了所有重要知识点,且均是在例题及复习思考题基础上的提升,能够让读者由浅入深地加深对知识点的理解,便于及时复习巩固知识点的学习。

(5) 本教材所有例题、习题均严格遵守业界较通用的编程规范,设计结构合理,思路清晰,注重程序的可读性和健壮性。

(6) 每章小结均以表格的形式列出本章的重点、难点及常见易错点,结构清晰,便于读

者复习把握。

【C 语言学习的误区】

(1) 只注重功能实现,不注重编程规范,导致代码的可读性及可维护性较差。

(2) 认为只要编写的代码在编译阶段没有错误,且能运行出正确结果就万事大吉了。编译器不是万能的,尤其是 C 编译器语法检查不够严格,有些潜在的风险难以发现。且由于测试用例的有限性和片面性,一两次的正确运行结果不能保证程序的绝对正确性。

(3) 不注重代码调试。当发现编译错误时,根据编译器的提示很容易发现并修改,出现结果错误时,采用从前往后逐条分析代码的方法排查错误,认为掌握这种方法和具备这种能力已足够了,而不采用科学的代码调试工具进行排查错误。

(4) 仅为了等级考试,死记硬背枯燥的知识点、套题型,造成学习枯燥,所掌握的全是零散的、片面的知识点的堆砌,只见树木不见森林。

【C 语言学习的建议】

(1) 在学习 C 语言的过程中始终要围绕着锻炼编程思维和解决问题的能力,注重编程素养的提高而进行。

(2) 注重读写。多读程序,不仅只读规范的好程序,从而借鉴优秀程序的优点,还要以批判的精神对不规范,可读性、健壮性差的程序提出修改意见。多写程序,在写程序之前一定要有清晰的算法设计思路,选择好的数据结构和程序结构。运行正确后,要思考改进,久而久之,使用 C 语言解决问题的能力会得到进一步提高。

(3) 注重调试能力。程序设计中出现错误在所难免,这里所指的错误不仅是 C 语法错误,还包括逻辑错误。而后者是比较难发现和解决的,要使用调试工具,通过设置断点,进行代码走读,逐步缩小错误源的范围,最终找到并解决。

本书全部章节均由孙海洋编写,在编写过程中得到了黄润生院长、赵志宏院长及其他教师的大力帮助和支持,在此一并向他们表示衷心的感谢!

由于编者水平有限、时间仓促,书中疏漏和不足之处在所难免,恳请专家、读者批评指正。

编著者

2018 年于南京大学金陵学院

目 录

第 1 章 C 语言概述	1
1.1 计算机系统	1
1.2 计算机语言与程序设计语言	2
1.2.1 计算机语言	2
1.2.2 程序设计语言及其发展	2
1.3 C 语言的起源及特点	4
1.3.1 热衷游戏与 UNIX 的起源	4
1.3.2 UNIX 的改进与 C 语言的起源	4
1.3.3 C 语言的特点	5
1.4 C 语言的标准化	6
1.4.1 ANSI C/ISO C 标准	6
1.4.2 C99 标准	6
1.4.3 C11 标准	6
1.5 简单的 C 语言程序举例	7
1.5.1 C 语言程序的结构	7
1.5.2 C 语言的编程风格与规范	9
1.6 C 语言程序设计的一般步骤	9
1.7 算法	11
1.7.1 算法的概述	11
1.7.2 算法的表示	12
小结	14
习题	15
第 2 章 顺序结构程序设计	18
2.1 简单的顺序结构程序	18
2.1.1 标识符、关键字、常量、变量	18
2.1.2 运算符、表达式、语句	21
2.1.3 格式化输入、输出	21
2.2 数据类型	22
2.2.1 整型类型	22

2.2.2	浮点类型	26
2.2.3	字符类型	31
2.3	输入输出	35
2.3.1	字符型数据的输入和输出	35
2.3.2	数值型数据的输入和输出	38
2.3.3	数值与字符混合输入和输出	39
小结	40
习题	42
第 3 章	运算符与表达式	44
3.1	运算符和表达式中的基本概念	44
3.2	算术运算符及算术表达式	45
3.3	逻辑、关系运算符及其表达式	47
3.4	赋值运算符及赋值表达式	50
3.5	移位运算符及移位表达式	52
3.6	sizeof 运算符及其表达式	53
3.7	逗号运算符及逗号表达式	54
3.8	运算符的优先级与结合性	55
3.9	类型转换	57
3.9.1	自动类型转换	57
3.9.2	强制类型转换	58
小结	59
习题	61
第 4 章	分支结构	63
4.1	if 语句	63
4.2	if-else 语句与条件表达式	68
4.2.1	if-else 语句	68
4.2.2	条件表达式	72
4.3	if 语句嵌套	74
4.4	级联 else-if 多分支语句	79
4.5	switch-case 多分支结构	84
小结	90
习题	93
第 5 章	循环结构	98
5.1	while 循环	98
5.2	do-while 循环	102
5.3	while 和 do-while 的关系	104

5.3.1	while 和 do-while 的等价关系	104
5.3.2	while 和 do-while 的不等关系	105
5.4	for 循环	106
5.5	循环的嵌套结构	109
5.6	执行流程跳转语句	113
5.6.1	goto 语句	113
5.6.2	break 语句	114
5.6.3	continue 语句	117
5.7	综合举例	120
小结		125
习题		126
第 6 章	数组	130
6.1	一维数组	130
6.1.1	一维数组的定义	130
6.1.2	一维数组的引用	132
6.1.3	一维数组的初始化	133
6.2	查找和排序算法	136
6.2.1	顺序查找	136
6.2.2	气泡排序	138
6.2.3	选择排序	143
6.3	二维数组	146
6.3.1	二维数组的定义	146
6.3.2	二维数组的引用	148
6.3.3	二维数组的初始化	150
6.3.4	二维数组的存储	152
6.3.5	二维数组的应用举例	152
6.4	一维字符数组	155
6.4.1	一维字符数组的定义及初始化	155
6.4.2	一维字符数组的引用	157
6.4.3	一维字符数组的应用举例	158
6.5	字符串处理函数	161
6.6	二维字符数组	169
6.6.1	二维字符数组的定义及初始化	170
6.6.2	二维字符数组的引用	170
6.6.3	二维字符数组的应用举例	172
6.7	数组综合举例	175
小结		178
习题		180

第 7 章 函数	184
7.1 函数的定义	184
7.1.1 函数定义格式	184
7.1.2 函数返回值	186
7.2 函数的调用	187
7.2.1 函数调用格式	187
7.2.2 函数调用过程	188
7.2.3 函数原型声明	189
7.2.4 函数调用举例	190
7.3 函数的嵌套调用	193
7.4 传值调用和传址调用	195
7.5 递归函数	200
7.6 变量的作用域和生存期	204
7.6.1 变量的作用域	205
7.6.2 存储区和存储类型	207
小结	211
习题	213
第 8 章 指针	218
8.1 指针的定义与引用	218
8.1.1 内存与地址	218
8.1.2 指针变量的定义	220
8.1.3 指针变量的引用	222
8.2 指针与数组	223
8.2.1 一维数组和指针	223
8.2.2 二维数组和指针	226
8.2.3 数组指针和指针数组	228
8.3 指针与字符串	234
8.3.1 常量字符串与指针	234
8.3.2 变量字符串	236
8.4 指针与函数	238
8.4.1 指针作函数形参——传址调用	238
8.4.2 指针作函数返回类型——指针函数	241
8.4.3 指向函数的指针——函数指针	242
8.5 二级指针	246
8.5.1 二级指针的定义	246
8.5.2 二级指针与二维数组	248
8.5.3 二级指针与指针数组	250

8.6	动态内存分配与指针	250
8.6.1	无类型指针和空指针	251
8.6.2	常见动态内存申请和释放函数	252
小结	255
习题	258
第 9 章	自定义类型	264
9.1	结构体类型及其变量	264
9.1.1	结构体类型的引入	264
9.1.2	结构体类型定义	265
9.1.3	结构体类型的变量	266
9.1.4	结构体变量成员的引用	268
9.1.5	结构体变量的存储	271
9.2	结构体数组	274
9.2.1	结构体数组定义与使用	274
9.2.2	结构体数组的应用	276
9.2.3	类型同义词	281
9.3	结构体指针与结构体数组	284
9.4	结构体与函数	286
9.5	单链表	293
9.5.1	数据的存储结构	294
9.5.2	单链表	294
9.5.3	单向循环链表	301
9.6	共用体	304
9.6.1	共用体类型及其变量的定义	304
9.6.2	字节存储机制	305
9.7	枚举类型	307
小结	309
习题	310
第 10 章	输入和输出	315
10.1	文件及其分类	315
10.2	文本文件与二进制文件	317
10.2.1	文本文件与二进制文件	317
10.2.2	C 语言与文件读写	317
10.2.3	缓冲和非缓冲文件系统	318
10.3	文件的打开与关闭	319
10.4	文件的顺序读写	320
10.4.1	按字符输入输出	320

10.4.2	按字符串输入输出	324
10.4.3	按格式化输入输出	326
10.4.4	按二进制方式读写数据块	331
10.5	文件的随机读写	340
	小结	342
	习题	343
第 11 章	预处理和位操作	346
11.1	预处理指令与预处理器	346
11.2	宏定义	348
11.2.1	无参宏定义	348
11.2.2	带参宏定义	349
11.2.3	带参宏调用与函数调用	353
11.3	条件编译	354
11.4	位操作	357
11.4.1	原码、反码、补码	357
11.4.2	位操作符	357
	小结	361
	习题	363
	参考文献	365
	附录	366
附录 A	VC++ 6.0 环境中开发 C 程序的步骤	366
附录 B	ASCII 表	374
附录 C	运算符的优先级和结合性	376
附录 D	ANSI C 常用库函数	377

本章学习目标

- 了解 C 语言的起源及发展历史
- 掌握 C 语言的特点
- 掌握编写简单 C 语言程序的方法
- 掌握在集成开发环境 VC++ 6.0 中开发 C 程序的方法和步骤
- 掌握算法的概念及常见的算法描述形式

本章先向读者简单介绍 C 语言的起源及发展史、特点及 C 语言的标准,再通过简单的 C 语言程序举例介绍 C 语言程序的组成,以及在集成开发环境中开发 C 语言程序的方法和步骤。最后简单介绍算法及其基本概念。

1.1 计算机系统

计算机系统由硬件系统和软件系统两部分组成,一般把没有安装任何软件(包括系统软件、应用软件等)的计算机称为“裸机”,“裸机”不能完成任何操作。

软件又称为程序,是计算机能识别和执行的一系列指令的集合。一般包括系统软件(如操作系统等)和应用软件(如办公软件、QQ、WeChat 聊天软件等)。

显然仅是计算机硬件无法完成各功能操作,而如果在计算机硬件上仅安装了单纯的系统软件,只能说此时具备了一个最基本的计算机系统,即搭好了硬件和系统软件平台。但是,该计算机系统依然无法实现聊天、收发邮件、整理 Word 文档等任何具体操作。只有在该平台(硬件+系统软件)上增加各种“角色”(应用软件),才能实现各种操作,发挥出其巨大价值。

由此可见,若要想使用计算机完成各种功能操作,不仅需要安装系统软件,还必须安装各对应的功能软件。例如,只有安装了 QQ 聊天应用软件后,才可以实现 QQ 用户之间的聊天通信功能;只有安装了 Microsoft Office 等相关办公类的应用软件后,才能使用 Word、Excel 等工具辅助日常管理工作;只有安装了绘图应用软件后,才可以实现各种绘图操作功能;只有安装了 VC++ 6.0 等 C 语言集成开发应用软件后,才可以实现编写、运行、调试 C 语言程序的相关操作功能。

综上所述,要使计算机发挥作用,必须在硬件上安装相应的系统软件,即操作系统。然后在操作系统上安装各种应用软件,来实现各种功能操作。

【复习思考题】

1. 简述计算机系统的组成。
2. 仅在计算机硬件上安装了操作系统后,就可以实现聊天、收发邮件等功能吗?
3. 列举常见的系统软件和应用软件。
4. 买来一台“裸机”,如何让其具备编辑 Word 文档、收发邮件、开发 C 语言程序的各种功能?

1.2 计算机语言与程序设计语言

1.2.1 计算机语言

计算机硬件仅能直接识别由 0、1 组成的二进制序列形式的指令,即通过该二进制指令可以与计算机硬件进行“交流”,故通常把计算机硬件能够直接识别的二进制指令集合形象地称为“计算机语言”或“机器语言”。

计算机所能识别和执行的全部指令的集合称为该机器的指令系统,不同型号的计算机的指令系统不同,故机器语言是面向具体机器的语言,不具有可移植性,不便于推广。

计算机硬件直接识别的每条指令均是由 0、1 串组成的,对应机器语言的一条语句,代表一个具体的操作。指令的一般格式为:操作码+地址码。例如在某型号的机器上,指令 1011011000000000 表示执行一次加法操作,而指令 1011010100000000 则表示执行一次减法操作。它们的高 8 位表示操作码,而低 8 位表示地址码,即对某个地址空间做相应操作。而在其他型号的机器上,可能无法识别上述加减操作对应的二进制指令。

综上所述,计算机硬件仅能直接识别机器语言这唯一的“母语”,且机器语言是面向具体机器的,不具有可移植性。

【复习思考题】

1. 什么是计算机语言?
2. 若查询并掌握了某机器上对应的二进制序列形式的加法指令,是否可以使用该指令在其他类型机器上执行加法操作?为什么?

1.2.2 程序设计语言及其发展

1. 程序设计语言

如前所示,人们通过使用各种软件或称程序来控制计算机硬件完成各种功能操作。而编写这些软件的过程称为编程或程序设计,把程序设计过程中使用的一系列符号及相关规则的集合称为程序设计语言,而把编写程序的人通常称为“程序员”或“编程者”或“程序设计者”。

2. 低级语言阶段

编程者可直接使用机器语言进行编程来实现某种功能。由于机器语言直接被硬件识别和执行,故执行效率极高;但由于其对应 0、1 二进制序列指令烦琐难记、可读性差,且不可移植,故直接采用机器语言进行编程的效率极低,且对编程者的专业化训练程度要求非常高。因此,采用机器语言作为程序设计语言,不便于计算机的普及和发展。

后来设计出了使用一些特定符号来代替机器语言中二进制指令的程序设计语言,称为汇编语言(Assembly Language)或符号语言。在汇编语言中,使用助记符(Mnemonic)代替机器语言中的操作码,例如,使用 ADD 表示加法操作、SUB 表示减法操作、MOVE 表示传送指令等;使用地址符号(Symbol)或标号(Label)代替地址码,故汇编语言也称为符号语言。

使用汇编语言编写的程序,计算机硬件并不能直接识别和执行,需要由一种中间程序将汇编语言编写的程序翻译成该机器语言对应的 0、1 指令后才能被识别和执行,这种起翻译作用的中间程序称为汇编编译器或汇编程序,把汇编语言翻译成机器语言的过程称为编译或汇编。

通常,把除机器语言外的其他编程语言“翻译成”另一种语言(通常为低级语言)的程序或软件,称为编译器。例如,把汇编语言转换成机器语言的程序就是一种汇编编译器。这种转换的过程称为编译。

和机器语言一样,汇编语言也依赖于具体的机器类型,具有不可移植性。一般把机器语言和汇编语言称为低级语言。

3. 高级语言阶段

把接近于人类自然语言和数学语言且不依赖于具体硬件的编程语言,称为高级语言。

通常,把除了机器语言和汇编语言外的其他编程语言,均称为高级语言。例如,要实现两个数相加的操作功能,用 C 语言可表述为 $c=a+b$;。该语句使用了数学中的加运算符+,非常接近人们的数学思维。仅通过该语句就容易看出编程者的意图。再比如,实现“如果某学生成绩大于等于 90 分,则打印输出优秀”的功能,使用 C 语言可以表示为:

```
if(score>=90)
    printf("该同学成绩等级:优秀!\n");
```

在英语中 if 为“如果”的意思,printf 也较容易看出是实现“打印”的功能,该 C 语言程序就像是人类对某实际问题思维或算法思想的一种“直接翻译”。

同汇编语言一样,所有高级语言都不能直接被计算机硬件识别和执行,都必须通过编译器,最终转化为机器语言对应的二进制指令形式。

高级语言执行效率不如低级语言高,但其编程效率明显高于低级语言,且便于掌握和推广。

高级语言与低级语言相比,具有如下优点。

- (1) 由于高级语言比较接近人类自然语言和数学语言,更容易学习和掌握。
- (2) 高级语言摆脱了低级语言对硬件的严重依赖性,具有较好的可移植性。
- (3) 高级语言采用结构化程序设计,编写代码的可读性及可维护性较高。
- (4) 高级语言提供了较丰富的运算符和表达式,便于编程者较灵活地使用该语言表达自己的设计思想。程序开发周期短、开发效率较高。

4. 常见的高级语言

自从 1954 年美国 IBM 公司的 John Backus 开发并发布了世界上第一个高级程序设计语言 FORTRAN 之后,各种高级程序设计语言接踵而至,至今六十多年来,世界范围内已公布的高级程序设计语言就有数百上千种之多。

比较优秀且影响力较大的高级程序设计语言有数十种,常见的如 C、C++、Java、C#、VB、

Delphi、Pascal、Python、Perl、Ruby 等。

【复习思考题】

1. 计算机硬件能直接识别 C 语言等高级语言吗?
2. 什么是计算机语言? 什么是程序设计语言?
3. 什么是编译器?
4. 什么是高级语言? 与低级语言相比, 高级语言的优点是什么?
5. 列举常见的高级程序设计语言的名称。

1.3 C 语言的起源及特点

1.3.1 热衷游戏与 UNIX 的起源

1964 年, 由美国通用电气公司和麻省理工学院发起了一个合资项目, 该项目旨在开发一套能运行在 GE-645 等大型主机之上的多用户、多任务的分时操作系统, 简称 MULTICS。1965 年, 贝尔实验室派出开发人员 Ken Thompson 等也加入了该项目, 虽然项目期间发布了一些版本的 MULTICS 产品, 但由于运行性能较差, 1969 年该项目以失败告终。

由于 Ken Thompson 酷爱游戏及游戏编程, 他在项目 MULTICS 期间编写了一款名叫“星际旅行”(Star travel) 的游戏, 并运行在该 MULTICS 系统上, 但运行速度非常慢, 且耗费昂贵。

1969 年 MULTICS 项目宣告失败后, Ken Thompson 没有放弃其游戏的梦想, 他在贝尔实验室的库房中, 找到一台闲置的 PDP-7 裸机, 但由于缺少操作系统, 无法在该 PDP-7 上运行其游戏。在 Dennis Ritchie 的帮助下, 他使用汇编语言为该 PDP-7 编写了一个操作系统雏形, 并把其游戏成功运行在了该操作系统之上。该操作系统体现出了很多优势, 受 MULTICS 项目开发经验的启发, Dennis Ritchie 和 Ken Thompson 在该游戏操作系统雏形的基础上, 进一步完善和开发新功能, 最终于 1970 年开发出了一款新的多用户、多任务操作系统, 称为 UNIX 操作系统。

综上所述, 1969—1970 年, 美国贝尔实验室的 Ken Thompson 和 Dennis Ritchie 等使用汇编语言编写了第一个版本的 UNIX 操作系统。

【复习思考题】

1. 简述 MULTICS 项目的发起者和参与者及该项目的目标。
2. 裸机上能运行游戏吗? 如果不能, 指出其缺少的条件。
3. UNIX 操作系统最初是由哪种程序设计语言编写的?

1.3.2 UNIX 的改进与 C 语言的起源

由于 UNIX 操作系统良好的性能, 在其发布初期, 就得到迅速的推广和应用。1973 年, Ken Thompson 和 Dennis Ritchie 在做系统内核移植开发时, 感觉使用汇编语言很难实现。后来决定使用一种称为 BCPL (Basic Combined Programming Language) 的语言进行开发, 在开发过程中, 他们在 BCPL 的基础上做了进一步的改进, 推出了 B 语言 (取 BCPL 第一个字母)。后来发现使用 B 语言开发的 UNIX 内核, 还是无法达到他们的预期要求, 于