



21世纪高职高专计算机系列规划教材

国家骨干高职院校精品课程建设成果



C语言程序设计 案例教程

程立倩 主编



北京邮电大学出版社
www.buptpress.com

21 世纪高职高专计算机系列规划教材
国家骨干高职院校精品课程建设成果

C 语言程序设计案例教程

主 编 程立倩

副主编 崔玉礼 孙俊琳 贾丽虹



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书将案例“学生成绩管理系统”贯穿整个教学过程,每个教学模块都有针对性地从案例引入,根据案例需求进行知识讲解,由浅入深地逐步补充和完善程序设计。主要内容包括 C 语言基础、C 程序数据描述与算法、顺序结构程序设计、选择结构程序设计、循环结构程序设计、应用数组程序设计、应用函数程序设计、应用指针程序设计、结构体和其他结构类型、位运算、文件等。书中特别设置了计算机等级考试二级 C 语言考试的重难点讲解与真题解析。

本书适合高职高专学生使用,也可以作为全国计算机等级考试 C 语言考试的参考书,还可供自学者参考。

图书在版编目(CIP)数据

C 语言程序设计案例教程/程立倩主编. —北京:
北京邮电大学出版社,2012.5
ISBN 978-7-5635-3000-7

I. ①C… II. ①程… III. ①C 语言—程序设计—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 075078 号

书 名: C 语言程序设计案例教程
主 编: 程立倩
责任编辑: 韦爱荣
出版发行: 北京邮电大学出版社
社 址: 北京市海淀区西土城路 10 号(邮编:100876)
E-mail: publish@bupt.edu.cn
经 销: 各地新华书店
印 刷: 北京振兴源印务有限公司
开 本: 787 mm×1 092 mm 1/16
印 张: 19.25
字 数: 468 千字
版 次: 2012 年 5 月第 1 版 2012 年 5 月第 1 次印刷

ISBN 978-7-5635-3000-7

定 价: 36.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

21 世纪高职高专计算机系列规划教材

编审委员会

主任：周国烛 北京电子科技职业学院电信工程学院院长

陈桂生 商丘职业技术学院软件学院院长

顾问：礼平 北京水木青青科技有限公司经理、教授

委员：(以姓氏笔画为序)

马慧 贵州师范大学职业技术学院 李俊霞 河南农业职业学院

王欢 黑龙江建筑职业技术学院 吴华 黑龙江农业职业技术学院

王梅 江苏海事职业技术学院 余维 湖南信息职业技术学院

王永乐 许昌职业技术学院 张晓伟 广东工程职业技术学院

王希军 许昌职业技术学院 陈巧莉 陕西国防工程职业技术学院

邓锐 湖南信息职业技术学院 罗汉江 淄博职业学院

冯艳茹 济源职业技术学院 赵德群 新疆工业高等专科学校

向隅 武汉铁路职业技术学院 禹云 娄底职业技术学院

刘君 沈阳农大高等职业技术学院 董义革 北京电子科技职业学院

孙玉明 辽宁石化职业技术学院 程立倩 烟台职业学院

杨洪雪 北京电子科技职业学院 缪勇 扬州环境资源职业技术学院

李斌 商丘职业技术学院 翟鹏翔 辽宁机电职业技术学院

出版说明

高等职业教育以培养生产、建设、管理、服务第一线的高素质技能型专门人才为根本任务,在建设人力资源强国和高等教育强国的伟大进程中发挥着不可替代的作用。

近年来,我国高职高专教育蓬勃发展,积极推进校企合作、工学结合人才培养模式改革,办学水平不断提高,为现代化建设培养了一批高素质技能型专门人才,对高等教育大众化作出了重要贡献。尽管如此,我国高职高专教育的质量、结构、规模还不能很好地适应当前经济社会发展的需要,部分高职高专院校毕业生还不能很好地满足社会工作岗位对相关技术和能力的需求。

要加快高职高专教育改革的步伐、全面提高人才培养质量,就必须对课程体系等问题进行深入探索。教育部在《关于全面提高高等职业教育教学质量的若干意见》中指出,“课程建设与改革是提高教学质量的核心,也是教学改革的重点和难点”,“建立突出职业能力培养的课程标准,规范课程教学的基本要求,提高课程教学质量”,这为高职高专教育课程体系建设指明了方向。在课程体系建设过程中,教材无疑起着至关重要的基础性作用,高质量的教材是培养高素质人才的重要保证。

目前,我国高等职业教育教学改革正在深入进行,高职教材建设取得了显著的成效。但从整体上看,教材建设仍不能很好地适应高职高专教育的发展需要,主要表现在:缺乏科学理论的支持,缺乏行业支持,缺少对生产实际的调查研究和深入了解,缺乏对职业岗位所需的专业知识和专项能力的科学分析,出现体系不明、内容交叉或重复、脱离实际、针对性不强等问题;与专业课程相配套的实践性教材严重不足;同类教材建设缺乏统一标准,相关课程的教材内容自成体系,缺乏沟通衔接;版本偏老或内容陈旧,不能及时将新法规、新知识、新技术、新工艺、新装备、新案例反映到教材中来;与劳动部门颁发的职业资格证书或技能鉴定标准缺乏有效衔接。教材的相对落后成为制约高职高专教育发展的瓶颈之一。

在此背景下,为了更好地贯彻《国家中长期教育改革和发展规划纲要(2010—2020年)》相关精神,更好地推进高职高专教育的发展,我们组织了一批具有丰富理论知识和实践经验的专家、一线教师,成立了21世纪高职高专规划教材编审委员会,着力规划出版一批符合高职高专教育特点和需求的优质教材。

依据教育部制定的《高职高专教育基础课程教学基本要求》和《高职高专教育专业人才培养目标及规格》,我们调研了数百所具有代表性的高等职业技术学院和高等专科学校,广泛而深入地了解了高职高专教育的专业和课程设置,系统地研究了课程的体系结构;同时充分汲取各院校在探索培养应用型人才方面取得的成功经验,并在教材出版的各个环节设置专业的审定人员进行严格审查,从而确保了整套教材“突出行业需求,突出职业的核心能力”的特色。



本系列教材除了满足内容充实、完整,结构、体例合理,语言得体、流畅等基本要求外,还力求克服以往高职高专教材的缺陷和不足,在以下方面打造自己的优势和特色:

(1) 本系列教材的定位更加强调“以就业为导向”。紧密依托行业或企业优势,建立产、学、研密切结合的运行机制,是高职高专教育健康发展的关键。我们通过对生产实际的调查研究和深入了解,对职业岗位(群)所需专业知识和专项能力的科学分析,以科学的课程理论为支持,力求使本系列教材定位与就业市场相结合,充分体现出“以就业为导向,以能力为本位,以学生为中心”的风格,从而更具实用性和前瞻性。

(2) 本系列教材打破传统的教材编写模式,力求在编写风格和表达形式方面有所突破,充分体现“项目导向、任务驱动”的教学理念,通过构建具体的工作任务作为学生学习的切入点,这就促使学生能够主动学习,从而达到“教中做、做中学、学中练”的目的,全面提升学生解决问题的实战经验和能力。

(3) 本系列教材编写思路清晰,体系结构安排合理,注重知识体系的有序衔接,力避知识的断层和重复。同时,教材也遵循教育部对高职高专教育提出的“以应用为目的,以必需、够用为度”原则,从实际应用的需要出发,减少枯燥、实用性不强的理论灌输。

(4) 本系列教材的编写及时跟进社会及行业的最新发展动态,将最新、最权威、最具代表性的成果运用于教材当中,从而避免了所讲知识与社会脱节。

为保证教材的总体质量和前瞻性,我们着重加强与示范性高等职业院校的合作,在全国范围遴选了具有丰富教学经验和实践经验、具有较高专业水平的双师型教师参加编写。

为支持“立体化”教学,我们为本系列教材精心策划了精品教学资料包和教学资源网,向教师用户提供教学课件、教学案例、教学参考、教学检测、教学资源推荐、课后习题答案等教学资源,以支持网络化及多媒体等现代化教学方式,有效提升教学质量。

希望各高职院校在使用本系列教材的过程中提出宝贵的意见和建议,我们将认真听取,不断完善。

21 世纪高职高专规划教材编审委员会

C语言是国内外广泛使用的程序设计语言之一,它以功能强大、表达能力强、使用灵活方便、兼具面向硬件编程的低级语言特性及通用性、可移植性等语言特性,成为软件开发的主流语言之一。C语言不仅适合于系统软件设计,也适合于应用程序设计。在操作系统、工具软件、图形图像处理软件、数值计算、人工智能及数据库系统等多个领域都得到了广泛的应用。目前,许多开发工具如微软的 Visual C++ 和 C# 及 Java 等都遵循着标准 C 语言的基本语法,很多嵌入式系统,多数都采用 C 语言来开发。可以说,C语言是程序开发人员必须掌握的程序设计语言,也是国内外高校广泛学习和普遍使用的一种重要的计算机语言。目前,全国计算机等级考试、全国计算机应用技术证书考试、全国计算机职业资格认证等都将 C 语言列入考试范围。学习和使用 C 语言已经成为广大计算机应用人员和学生的迫切需求。

C语言程序设计是高职高专院校计算机类各专业的核心课程之一。为适应 IT 行业的迅速发展和课程改革的迫切需要,本书作者与企业专家深入合作,根据工作过程系统化的指导思想,按照软件程序员、数据库管理员、ERP 实施等岗位所需的能力要求,基于软件开发流程和行业实用技术确定课程的职业知识、职业能力和职业素质目标。总结多年程序设计教学与开发的经验与成果,以软件开发工作过程为主线,以案例和项目任务为载体,集程序设计思想与具体实践于一体,兼顾 C 语言程序设计的特点和学生学习的认知规律,强调学习过程是渐进的程序设计过程,编写了这本知识全面、内容适度、技术先进的理论实践一体化教材。

本书精心设计了简单、有代表性、涵盖知识较为全面、贯穿于整个教学过程的“学生成绩管理系统”。该案例涉及本课程的全部知识内容,将项目划分成多个学习性的工作任务,学生可以随着知识的不断深入,逐步补充和完善程序的功能,从而在整个教学过程中,培养学生对应用程序的开发和设计能力。另外,每个模块的知识点讲解尽可能简明扼要,将大部分教学内容用实例的形式进行组织,在案例学习过程中,逐步掌握基本概念、语法规则,正确设计和编写程序。大部分模块后面都附加了技能训练,加强程序设计的理念,培养扎实的代码编写能力。每个模块最后都附加了全国计算机等级考试 C 语言考试的重难点讲解与真题解析,将国家认证的学习融于课程中,达到加强专业实践



能力以及获取认证能力的效果。

本书由烟台职业学院程立倩副教授任主编,崔玉礼、孙俊琳和贾丽虹任副主编。本书编写工作分配如下:模块 1 和模块 10 由崔玉礼编写,模块 2 和模块 11 由贾丽虹编写,模块 3 至模块 8 由程立倩编写,模块 9 和附录由孙俊琳编写。在本书的编写过程中,得到了周慧教授的指导与帮助,在此表示衷心感谢。

由于作者水平有限,加上时间仓促,本书的疏漏和不足之处在所难免,敬请有关专家和广大读者不吝指正。

编 者

模块 1 初识 C 语言	1	2.3.1 整型数据	23
案例引入	1	2.3.2 实型数据	26
1.1 C 语言简介	3	2.3.3 字符型数据	27
1.1.1 计算机语言概述	3	2.4 C 语言的运算符和表达式	29
1.1.2 C 语言概述	4	2.4.1 算术运算符与算术 表达式	30
1.2 C 程序介绍	6	2.4.2 赋值运算符和表达式	31
1.2.1 C 程序的总体结构	6	2.4.3 自增自减运算符和 表达式	33
1.2.2 C 程序的书写规则	8	2.4.4 逗号运算符和逗号 表达式	34
1.3 C 程序的开发过程	8	2.4.5 不同类型数据间的 转换	34
1.3.1 C 程序的实现过程	8	等级考试重难点讲解与真题解析	35
1.3.2 在 Turbo C 下实现 C 程序	9	思考与练习	38
1.3.3 在 Visual C++ 环境下 实现 C 程序	11	模块 3 顺序结构程序设计	40
等级考试重难点讲解与真题解析	14	案例引入	40
思考与练习	16	3.1 程序设计基础	42
模块 2 C 程序数据描述与计算	18	3.1.1 程序	42
案例引入	18	3.1.2 程序设计	42
2.1 C 程序的词法	20	3.1.3 算法	43
2.1.1 字符集	20	3.1.4 结构化程序	44
2.1.2 标识符	20	3.1.5 模块化程序	45
2.2 常量与变量	21	3.2 赋值语句	46
2.2.1 常量	21	3.3 复合语句和空语句	47
2.2.2 变量	22	3.3.1 复合语句	47
2.3 基本数据类型	23		



3.3.2 空语句	47
3.4 数据的输入与输出	48
3.4.1 字符的输入与输出	48
3.4.2 输出格式函数	50
3.4.3 输入格式函数	53
3.5 技能训练	55
等级考试重难点讲解与真题解析	57
思考与练习	59

模块 4 选择结构程序设计

案例引入	62
4.1 关系运算符和关系表达式	65
4.1.1 关系运算符	65
4.1.2 关系表达式	65
4.2 逻辑运算符和逻辑表达式	66
4.2.1 逻辑运算符	66
4.2.2 逻辑表达式	66
4.3 if 语句	68
4.3.1 简单 if 语句	68
4.3.2 if...else 语句	69
4.3.3 嵌套 if 语句	71
4.3.4 多分支 if...else...if 语句	72
4.4 条件运算符和条件表达式	73
4.5 switch 语句	74
4.6 技能训练	76
等级考试重难点讲解与真题解析	79
思考与练习	82

模块 5 循环结构程序设计

案例引入	85
5.1 while 语句	87
5.2 do...while 语句	90
5.3 for 语句	92
5.4 循环嵌套	95
5.5 break 语句和 continue 语句	97
5.5.1 break 语句	97

5.5.2 continue 语句	99
5.6 技能训练	100
等级考试重难点讲解与真题解析 ..	103
思考与练习	106

模块 6 应用数组程序设计

案例引入	110
6.1 一维数组	113
6.1.1 一维数组的定义	113
6.1.2 一维数组的初始化	114
6.1.3 一维数组的引用	114
6.2 二维数组	117
6.2.1 二维数组的定义	117
6.2.2 二维数组的初始化	118
6.2.3 二维数组的引用	118
6.3 字符数组和字符串	122
6.3.1 字符数组	122
6.3.2 字符串	123
6.3.3 字符串的输入/输出 ..	123
6.3.4 字符串处理函数	126
6.4 技能训练	131
等级考试重难点讲解与真题解析 ..	134
思考与练习	139

模块 7 应用函数程序设计

案例引入	141
7.1 函数概述	144
7.1.1 库函数	144
7.1.2 自定义函数	145
7.2 函数的定义和返回值	147
7.2.1 函数的定义	147
7.2.2 函数的返回值	148
7.3 函数的调用	148
7.3.1 函数调用的一般 形式	148
7.3.2 函数调用时的语法 要求	150
7.3.3 函数的声明	151

7.3.4 调用函数与被调用函数之间的数据传递	152	8.3.2 指针变量的算术运算(移动)	191
7.3.5 数组作为参数	153	8.3.3 指针比较	192
7.4 函数的嵌套调用和递归调用	155	8.4 指针与一维数组	193
7.4.1 函数的嵌套调用	155	8.4.1 通过指针访问一维数组	193
7.4.2 函数的递归调用	157	8.4.2 函数中一维数组名作为实参	195
7.5 局部变量和全局变量	158	8.5 指针与二维数组	198
7.5.1 程序的存储空间	158	8.5.1 二维数组和数组元素的地址	198
7.5.2 局部变量	158	8.5.2 函数中二维数组作为实参	200
7.5.3 全局变量	161	8.6 指针与字符串	202
7.5.4 全局变量的作用域和生存期	162	8.6.1 通过指针访问字符串常量	202
7.6 内部函数和外部函数	164	8.6.2 通过指针访问字符串数组	202
7.6.1 内部函数	164	8.6.3 字符串指针作为函数参数	203
7.6.2 外部函数	165	8.6.4 使用字符指针和使用字符数组处理字符串的区别	204
7.7 编译预处理	165	8.7 指针与函数	206
7.7.1 宏定义	165	8.7.1 函数间地址传递	206
7.7.2 文件包含	167	8.7.2 函数型指针	207
7.8 技能训练	168	8.7.3 指针型函数	208
等级考试重难点讲解与真题解析	172	8.7.4 指针数组和指向指针的指针	209
思考与练习	178	8.8 技能训练	213
模块 8 应用指针程序设计	184	等级考试重难点讲解与真题解析	214
案例引入	184	思考与练习	219
8.1 地址与指针	187	模块 9 结构体和其他构造类型	221
8.1.1 内存和内存地址	187	案例引入	221
8.1.2 变量、变量名、变量的地址和变量值	187	9.1 结构体变量	224
8.1.3 指针、变量的指针和指针变量	188	9.1.1 结构体类型的定义	224
8.1.4 系统访问变量的两种方式	188	9.1.2 结构体变量的定义	225
8.2 指针变量的定义与引用	188	9.1.3 结构体变量的初始化	226
8.2.1 指针变量的定义	188		
8.2.2 指针变量的引用	189		
8.3 对指针变量的操作	190		
8.3.1 指针变量的赋值	190		

9.1.4 结构体变量的引用	226
9.2 结构体数组与指针	228
9.2.1 结构体数组	228
9.2.2 结构体指针变量	230
9.3 函数之间结构体类型的数据传递	232
9.3.1 结构体变量作实参	233
9.3.2 结构体变量指针作实参	234
9.3.3 结构体函数	234
9.3.4 结构体指针函数	236
9.4 链表	237
9.4.1 链表的概念	237
9.4.2 链表的主要操作	238
9.5 共同体	241
9.5.1 共同体类型的声明	241
9.5.2 共同体变量的定义	241
9.5.3 共同体成员的引用	242
9.6 枚举类型	245
9.6.1 枚举类型的定义	245
9.6.2 枚举变量的定义与赋值	245
9.6.3 关于枚举的说明	245
9.6.4 枚举变量的输入与输出	246
9.7 定义类型别名	247
9.7.1 定义已有类型的别名	247
9.7.2 定义构造类型的别名	247
9.8 技能训练	248
等级考试重难点讲解与真题解析	250
思考与练习	255

模块 10 位运算

案例引入	259
10.1 位运算符	260
10.2 位运算符的运算与应用	261
等级考试重难点讲解与真题解析	263
思考与练习	265

模块 11 文件

案例引入	267
11.1 文件概述	271
11.1.1 文件的概念	271
11.1.2 文件指针	272
11.2 文件的基本操作	272
11.2.1 文件的打开和关闭	273
11.2.2 字符读/写函数	275
11.2.3 字符串读/写函数	276
11.2.4 数据块读/写函数	278
11.2.5 格式读/写函数	280
11.3 文件定位函数	280
11.3.1 rewind 函数	280
11.3.2 fseek 函数	281
11.3.3 ftell 函数	282
等级考试重难点讲解与真题解析	283
思考与练习	287

附录 A 常用字符的 ASCII 码

对照表	290
-----	-----

附录 B C 语言标准库函数

参考文献	294
------	-----

初识 C 语言

案例引入——学生成绩管理系统登录界面设计



任务描述

- (1) 安装 C 编译器 (如 Borland Turbo C、Visual C++、Win-TC、Borland C++、Dev-C++、Turbo C/C++ for Windows 集成实验与学习环境等), 了解所安装的 C 编译器的基本使用方法与特点。
- (2) 使用 C 编译器编辑、编译和执行简单的 C 程序。
- (3) 设计“学生成绩管理系统”的登录界面程序。



任务目标

- (1) 能够掌握 C 程序的构成。
- (2) 能够应用某种 C 集成编译器进行 C 程序的编辑、编译和执行。
- (3) 能够根据系统提示进行操作、检错和纠错, 培养自我学习的能力。
- (4) 能够不断积累程序设计过程中出现的英文, 会认、会读、会写和会用。



源代码展示

```
#include <stdio.h>
#include <string.h>
int login() /* 输入密码登录系统,若三次输入不正确,则退出 */
{
    char secretCode[6];
    int number=3;
    printf("\n\n");
```



```
printf("\t\t\t * * * * * 欢迎使用学生成绩管理系统 * * * * *\n");
printf(" \n\n");
printf("\t\t\t 请输入密码:");
scanf("%s",secretCode);
while(number>0)
{
    if(strcmp(secretCode,"888888")==0)break;
    else
    {
        number--;
        if(number<=0) break;
        printf("\n\t\t\t 您还有 %d 次机会\n",number);
        printf("\n\t\t\t 请输入密码:");
        scanf("%s",secretCode);
    }
}
if(number>0) return 1;
else return 0;
}
mainMenu() /* 系统主菜单,需要以后添加 */
{ }
main()
{
    if(login()==0) return 1;
    do
    {
        mainMenu();
    }while(1);
}
```

运行结果

程序的运行结果如图 1-1 所示。

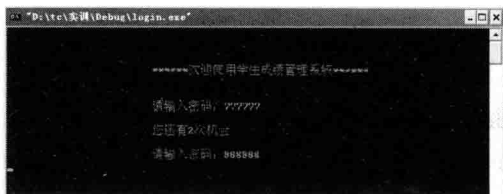


图 1-1 学生成绩管理系统登录界面



程序分析

本程序由主函数 main 和 login、mainMenu 函数构成,login 函数的功能是输出“ * * * * * 欢迎使用学生成绩管理系统 * * * * * ”密码登录界面;mainMenu 函数是学生成绩管理系统的主菜单,由主函数调用 login 函数实现。

此例说明一个 C 程序可以由多个函数组成,并总是从 main 函数开始运行。



知识讲解

C 语言是国际上广泛使用的计算机高级语言,其语法简洁、紧凑,使用方便、灵活,具有丰富的运算符和数据类型,自定义类型也十分灵活。C 程序的模块化主要通过函数实现。由于 C 语言既有高级语言的功能,又有低级语言的一些功能,因此它既可以用来编写系统软件,也可以用来编写应用软件。C 语言在操作系统、工具软件、图形图像处理软件、数值计算、人工智能及数据库系统等多个领域都得到了广泛的应用。目前,许多开发工具,如微软的 Visual C++ 和 C# 及 Java 等都遵循标准 C 语言的基本语法,很多嵌入式系统多数都采用 C 语言来开发。

1.1 C 语言简介

目前,人们使用的程序设计语言有上百种,它们中的大多数被称为计算机的“高级语言”。这些语言都是用接近人们习惯的自然语言和数学语言作为表达形式,便于人们学习和操作。C 语言是近年来非常流行的程序设计语言,是一种面向过程的通用程序设计语言,很多人宁愿放弃已经熟悉的其他语言而改用 C 语言。

1.1.1 计算机语言概述

计算机语言的种类非常多,总的来说可以分成机器语言、汇编语言和高级语言三大类。

1. 机器语言

机器语言是由 0、1 组成的机器指令的集合,是第一代计算机语言。计算机所使用的是由 0 和 1 组成的二进制数,二进制是计算机语言的基础。计算机发明之初,人们只能写出一串串由 0 和 1 组成的指令序列交由计算机执行,这种计算机能够认识的语言,就是机器语言。机器语言难读、难记、难写,容易出错,且不同机型不兼容。

2. 汇编语言

为了减轻使用机器语言编程的繁琐,人们进行了一种有益的改进,用一些简洁的英文字母、符号串来替代一个特定指令的二进制串。例如,用 ADD 代表加法,MOV 代表数据传递等,使程序比较直观,易于阅读和理解,更容易实现纠错及维护,这种程序设计语言就称为汇编语言,即第二代计算机语言。然而计算机是不认识这些符号的,这就需要一个专门的程序



负责将这些符号翻译成二进制的机器语言,这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件,移植性不好,但效率很高。针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精炼且质量高,所以至今仍是一种常用而强有力的软件开发语言。

3. 高级语言

从最初与计算机交流的经历中,人们意识到,应该设计一种语言,这种语言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。经过努力,1954年,第一个完全脱离机器硬件的高级语言——FORTRAN问世了。这么多年来,共有几百种高级语言出现,有重要意义的有几十种,影响较大、使用较普遍的有FORTRAN、ALGOL、COBOL、BASIC、LISP、PL/1、Pascal、C、PROLOG、C++、VC、VB、Java等。

高级语言的发展也经历了从早期语言到结构化程序设计语言,从面向过程到非过程化程序语言的过程。相应地,软件的开发也由最初的个体手工作坊式的封闭式生产,发展为产业化、流水线式的工业化生产。

高级语言的下一个发展目标是面向应用,也就是说,只需要告诉程序你要干什么,程序就能自动生成算法,自动进行处理,这就是非过程化的程序语言。

4. 编译过程

用高级语言所编制的程序不能直接被计算机识别,必须经过转换才能执行,按转换方式可将它们分为两类:解释类和编译类。

(1)解释类:应用程序源代码一边由相应语言的解释器翻译成目标代码,一边执行,因此效率比较低,而且不能生成可独立执行的可执行文件,应用程序不能脱离其解释器,但这种方式比较灵活,可以动态地调整、修改应用程序。

(2)编译类:编译是指在应用源程序执行之前,就将程序源代码翻译成目标代码,因此其目标程序可以脱离其语言环境独立执行,使用比较方便、效率较高。但应用程序一旦需要修改,必须先修改源代码,再重新编译生成新的目标文件才能执行,如果只有目标文件而没有源代码,修改会很不方便。现在大多数的编程语言都是编译型的,如 Visual Basic、Visual C++、Visual Foxpro、Delphi等。

1.1.2 C 语言概述

1. C 语言的发展

在 C 语言诞生以前,系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件,其可读性和可移植性都很差;但一般的高级语言又难以实现对计算机硬件的直接操作,于是人们盼望有一种兼有汇编语言和高级语言特性的新语言——C 语言。

C 语言是在 B 语言的基础上发展起来的,它的根源可以追溯到 1960 年出现的 ALGOL 60,ALGOL 60 是一种面向问题的高级语言,但离硬件比较远,不宜用来编写系统软件。1963 年英国剑桥大学推出了 CPL 语言,它在 ALGO 60 的基础上接近硬件一些,但规模较大,难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 做了简化,推出了 BCPL 语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 为基础,设计出了简单而且很接近硬件的 B 语言,并用 B 语言写了第一个 UNIX 操作系统,在 PDP-7 上实现。1971 年在 PDP-11/20 上实现了 B 语



言,并编写了UNIX操作系统。但B语言过于简单,功能有限。1972年至1973年之间,贝尔实验室的D. M. Ritchie在B语言基础上设计出了C语言。C语言既保持了BCPL和B语言的优点,又克服了它们的缺点。1973年,Ken Thompson和D. M. Ritchie合作把UNIX的90%以上用C语言改写,即UNIX第五版。

虽然对C语言进行了多次改进,但主要还是在贝尔实验室内部使用。直到1975年UNIX第六版公布后,C语言的突出优点才引起人们的注意。1977年出现了不依赖于机器的C语言编译文本《可移植C语言编译程序》,使C语言移植到其他机器时所需做的工作大大简化,这也推动了UNIX操作系统迅速在各种机器上的实现。可以说,C语言与UNIX是一对孪生兄弟,在发展中相辅相成。1978年以后,C语言已先后移植到大、中、小、微型机上,已独立于UNIX和PDP了。

1983年,美国国家标准化协会(ANSI)根据C语言问世以来各种版本对C语言的发展和扩充,制定了新的标准,称为ANSI C。1987年ANSI又公布了C语言新标准——87 ANSI C。1990年,国际标准化组织ISO接受了87 ANSI C为C语言的国际标准。目前流行的C编译系统都是以ANSI C为基础的。

目前流行的C语言的编译器有:Microsoft C/C++、Borland C/C++、Visual C++ 6.0、Win-TC、Turbo C/C++ for Windows集成实验与学习环境等,各种版本基本部分是相同的,但略有差异,因此应了解所用计算机系统配置的C编译系统的特点和规定。

2. C语言的特点

C语言之所以能存在和发展,并具有生命力,在于它有不同于其他语言的特点。C语言的主要特点如下。

(1)简洁、紧凑。C语言一共只有32个关键字,9种控制语句。

(2)运算符丰富。C语言共有44种运算符。它把括号、赋值、强制类型转换等都作为运算符处理,从而使C语言的运算类型极其丰富,表达式多样化。

(3)具有丰富的数据类型。C语言具有整型、实型、字符型、数组类型、指针类型、结构体类型、共同体类型等,能方便地构造更加复杂的数据结构(如链表、树、栈等)。

(4)C语言是一种结构化的程序设计语言。C语言具有结构化的控制语句(如if、switch、for、while、do...while)。用函数作为程序的模块单位,便于实现程序的模块化。

(5)语法限制不严格,程序设计灵活。例如,C语言不检查数组下标越界,C语言不限制对各种数据转化(编译系统可能对不合适的转化进行警告,但不限制),不限制指针的使用,程序正确性由程序员保证。灵活和安全是矛盾的,对语法限制的不严格可能也是C语言的一个缺点,黑客可能使用越界的数组攻击用户的计算机系统。

(6)能进行位操作,可以直接对部分硬件进行操作。例如,C语言可以直接操作计算机硬件,如寄存器,各种外设I/O端口等;C语言的指针可以直接访问内存物理地址;C语言类似汇编语言的位操作可以方便地检查系统硬件的状态。

(7)可移植性好。用C语言编写的程序基本上不需要修改或只需要少量修改就可以移植到其他的计算机系统或操作系统中。

(8)C语言编译后生成的目标代码小,质量高,程序的执行效率高。有资料显示只比汇编语言代码效率低10%~20%。