



语言 程序设计

张建军 史银龙 刘胜厚 编著



教育部职业教育与成人教育司全国职业教育与成人教育教学用书规划教材
“十一五”全国计算机技能型紧缺人才培养规划教材



语言 程序设计

张建军 史银龙 刘胜厚 编著

海洋出版社

2010年·北京

内 容 简 介

本书是一本关于C语言程序设计基础及应用的教程。C语言是目前流行的通用程序设计语言,是许多计算机专业人员和计算机的爱好者学习程序设计语言的首选。本书针对初学者的需求,从零开始、系统全面地讲解了C语言的专业内容与编程技巧。

本书内容:本书为全国示范校课程建设项目成果。分为两部分,共16章,第一部分为项目案例教程部分,包括第1~5章,主要以汉字显示技术、贪吃蛇游戏设计、黑白棋游戏设计、电话本管理系统设计以及迷宫游戏设计这5个大型项目案例,详细讲解C语言的编程思想及技巧。第二部分为学生资讯部分,主要介绍C语言概述、数据类型、运算符与表达式、程序设计结构、数组、函数、预处理命令、指针、结构体与共用体、文件、图形模式以及系统函数等内容,全面系统地介绍了C语言程序设计的强大功能。

本书特点:1. 改变传统教材的编写模式,以项目为导引,将知识点全面概括在应用实例中。2. 以实际工作中遇到的项目作为案例,最大限度调动读者的兴趣,学以致用。3. 每一条程序代码均给出注释,方便阅读、理解,一目了然。4. 课后配有大量习题,方便检测和巩固学习成果,并做到及时应用。

适用范围:本书适用于高等学校各专业程序设计基础教学,特别适合应用型本科、高职院校的计算机及非计算机相关专业的学生使用,同时也是计算机等级考试备考的一本实用辅导书。

本书说明:本书中项目案例均提供源码,联系方式为jsjwj@163.com。

图书在版编目(CIP)数据

C语言程序设计/张建军,史银龙,刘胜厚编著. —北京:海洋出版社,2010.7
ISBN 978-7-5027-7756-2

I. ①C… II. ①张…②史…③刘… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第119351号

总策划:刘 斌

责任编辑:刘 斌

责任校对:肖新民

责任印制:刘志恒

排版:海洋计算机图书输出中心 晓阳

出版发行:海洋出版社

地 址:北京市海淀区大慧寺路8号(705房间)
100081

经 销:新华书店

技术支持:(010) 62100055

发 行 部:(010) 62174379(传真)(010) 62132549
(010) 62100075(邮购)(010) 62173651

网 址:www.oceanpress.com.cn

承 印:北京画中画印刷有限公司

版 次:2010年7月第1版

2010年7月第1次印刷

开 本:787mm×1092mm 1/16

印 张:17.75

字 数:410千字

印 数:1~4000册

定 价:28.00元

本书如有印、装质量问题可与发行部调换

写在前面的话

当前我国正向现代化、信息化、工业化的国家大步迈进，迫切需要数以千万计的高技能人才和数以亿计的高素质劳动者。社会各行业、工业企业等部门人才短缺、特别是技能型人才严重短缺。近年来，我国的职业教育已日益被经济建设所依赖，技能型人才需求存在巨大缺口，因此培养培训任务迫在眉睫。

温家宝总理在2005年11月7日的全国职业教育工作会议上强调，要大力发展中国特色的职业教育，加快培养高技能人才和高素质劳动者。教育部、劳动与社会保障部、国防科工委、信息产业部、交通部、卫生部也联合颁发了《教育部等六部委关于职业院校制造业和现代服务业技能型紧缺人才培养培训工程的通知》，教育部办公厅和信息产业部办公厅颁发了《关于确定职业院校开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的通知》及《职业院校计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》。这足以说明职业教育在国家人才培养工程中的重要性以及国家对技能型人才需求的紧迫性。

培养培训软件技术专业领域技能型紧缺人才是职业教育的根本使命和当前的紧迫任务，目的就是要刻不容缓地把走进校园的学生培养成适合国家发展和企业需要的有用人才，培养他们成为有一技之长的劳动者和实用型人才，培养的目的主要是面向就业。

根据以上精神和指导方案，中国计算机学会职业教育专业委员会与海洋出版社海洋智慧图书有限公司，特组织北京、河北、内蒙古、大连、长春、唐山、武汉、深圳、肇庆和杭州等地主要职业院校负责人和一线教师，召开教材研讨会，相互交流经验，介绍需求，共同策划和编写了本套《“十一五”全国计算机职业院校精品课程规划教材》。本套教材是面对目前全国职业院校学生的现状和职业需求而编写的、颇具特色的实用培养培训教材。

我们特将这套教材倾心奉献给全国广大的教师和学生，为国家“职业教育与培训创新工程”推波助澜，为满足社会巨大的人才培养需求做出应有的贡献！

整套书的编写宗旨

- 三符合：符合教育部教学大纲、符合市场技术潮流、符合职业院校专业课程需要。
- 技术新、任务明、步骤细致、实用性强，专为技能型紧缺人才量身定制。
- 软件功能与具体范例操作紧密结合，边讲解边动手，学习轻松，上手容易。
- 三适应：适应新的教学理念、适应学生水平现状、适应用人标准要求。

整套书的特色

- 理论精练够用、任务明确具体、技能实操落实，活学活用。

教材编委会

前 言

C 语言是目前流行的通用程序设计语言,是计算机专业人员和编程爱好者学习程序设计语言的首选。它适合作为系统描述语言,既可以用来写系统软件,也可以用来编写应用软件。C 语言的主要有语言简洁、紧凑,源程序简练,运算符灵活、数据结构丰富,语法规则灵活,程序设计自由度大,目标代码质量高,程序执行效率高,可移植性好等特点。

本书作为高职学生学习计算机编程的教材,为全国示范校课程建设项目。全书贯穿两条线索。一条线索为项目案例教学线索,该线索中提供了 5 个项目案例,项目案例来源于实际工程项目,主要为后续的课程——单片机 C 语言编程做好铺垫。项目案例教学的目的是使学生能够运行已有的项目案例程序源码,根据项目改造的需求阅读程序源码,改造项目程序,使之能够满足新项目需求,它主要培养学生阅读程序的能力和改造程序的能力。另一条线索是 C 语言基础知识学习线索,该线索和传统的教材的结构相同,通过循序渐进,由浅入深的方式进行展开。在具体教学实践中,首先按照 C 语言基础知识学习线索进行入门级的教学,重点强调基础知识的教学,完成 C 语言的基本知识体系后,根据学生的学习情况选择两到三个项目开展项目教学。

本书内容翔实,层次分明,结构紧凑,叙述深入浅出,通俗易懂。基础知识部分每章附有大量习题,帮助读者迅速提高编程水平。适合作为高职、高专及各类大专院校的教材,也可作为等级考试和其他从事计算机编程人员的参考书。

本书由张建军主编且统稿。其中张建军编写项目案例一和 C 语言基础知识第 6 章到第 16 章,史银龙编写项目案例二和四,刘胜厚编写项目案例三和五。

由于编写时间仓促、水平有限,书中错误和不妥之处在所难免,恳请读者批评指正,并多多提出宝贵意见。

编 者

目 录

第1部分 项目案例教程

第1章 汉字显示技术	2
1.1 项目需求	2
1.1.1 需求描述	2
1.1.2 程序结构	4
1.2 程序框架设计	6
1.3 项目程序实现	6
1.4 知识点	9
1.5 项目总结	12
1.6 项目改造与应用	12
1.6.1 项目改造	12
1.6.2 项目扩展	13
第2章 贪吃蛇游戏设计	15
2.1 项目需求	15
2.1.1 需求描述	15
2.1.2 功能描述	16
2.2 程序设计	17
2.2.1 程序结构设计	17
2.2.2 数据结构设计	18
2.2.3 界面设计	19
2.2.4 功能设计	21
2.3 程序实现与分析	22
2.3.1 程序编码	22
2.3.2 程序分析	26
2.4 知识点	34
2.4.1 算法与程序	34
2.4.2 流程图	34
2.4.3 结构化程序设计	35
2.4.4 C语言	36
2.5 项目总结	37
2.6 项目改造与应用	37
2.6.1 项目改造	37
2.6.2 项目扩展	39
第3章 黑白棋游戏设计	42

3.1 项目需求	42
3.1.1 需求描述	42
3.1.2 功能描述	43
3.2 程序框架设计	44
3.3 程序实现与分析	46
3.3.1 程序编码	46
3.3.2 程序分析	54
3.4 知识点	55
3.5 项目总结	57
3.6 项目改造与应用	57
第4章 电话本管理系统设计	61
4.1 项目需求	61
4.1.1 需求描述	61
4.1.2 功能描述	62
4.2 程序框架设计	63
4.2.1 程序整体结构	63
4.2.2 数据结构设计	63
4.2.3 界面设计	64
4.2.4 功能设计	65
4.3 程序实现与分析	66
4.3.1 程序编码	66
4.3.2 程序分析	80
4.4 知识点	83
4.4.1 结构体	83
4.4.2 文件操作	83
4.5 项目总结	84
4.6 项目改造及运用	84
第5章 迷宫游戏设计	87
5.1 项目需求	87
5.1.1 需求描述	87
5.1.2 功能描述	88
5.2 程序框架设计	89
5.2.1 程序整体结构	89



5.2.2	数据结构设计	90
5.2.3	界面设计	90
5.2.4	功能设计	91
5.3	程序实现与分析	91
5.3.1	程序编码	91
5.3.2	程序分析	98

5.4	知识点	100
5.4.1	递归	100
5.4.2	C 语言	101
5.5	项目总结	102
5.6	项目改造及运用	102

第 2 部分 学生资讯材料

第 6 章 C 语言概述

108

6.1	C 语言出现的历史背景	108
6.2	C 语言的特点	109
6.3	C 语言程序的构成和格式	109
6.4	在 Turbo C 下运行 C 语言程序	110
6.4.1	开发 C 语言程序的过程	110
6.4.2	Turbo C 程序设计环境及上 机步骤	111
6.5	习题	113

第 7 章 数据类型、运算符与表达式

114

7.1	C 语言的数据类型	114
7.2	常量与变量	115
7.2.1	常量和符号常量	115
7.2.2	变量	115
7.3	整型数据	116
7.3.1	整型常量的表示	116
7.3.2	整型变量	117
7.3.3	整数在内存中的存储形式	118
7.3.4	整型数据的溢出	119
7.4	实型数据	119
7.4.1	实型常量的表示方法	119
7.4.2	实型变量	119
7.5	字符型数据	120
7.5.1	字符型常量	120
7.5.2	字符变量	121
7.5.3	字符数据在内存中的存储形 式和使用方法	121
7.5.4	字符串常量	121
7.6	变量赋初值	122
7.7	各类数值型数据间的混合运算	122

7.8 算术运算符和算术表达式

123

7.8.1	C 运算符简介	123
7.8.2	算术运算符和算术表达式	123
7.9	赋值运算符和赋值表达式	126
7.10	逗号运算符和逗号表达式	128
7.11	位运算	128
7.11.1	位运算符	129
7.11.2	位运算举例	131
7.12	字符数据的输入输出	132
7.12.1	putchar 函数 (字符输出 函数)	132
7.12.2	getchar 函数 (字符输入 函数)	133
7.13	格式输入输出	133
7.13.1	printf 函数 (格式输出函 数)	133
7.13.2	scanf 函数 (格式输入函 数)	137
7.14	习题	140

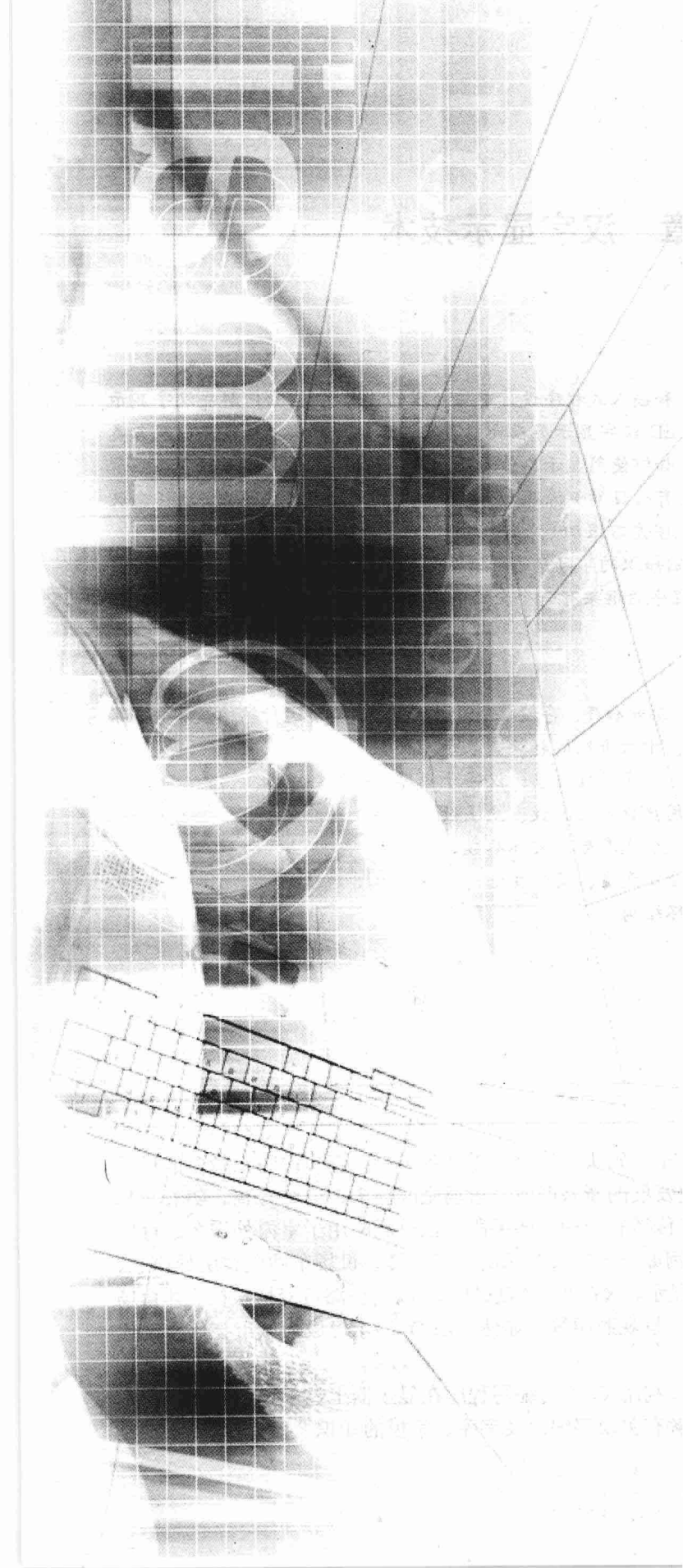
第 8 章 程序结构设计

143

8.1	程序设计的三种基本结构	143
8.1.1	程序的基本结构	143
8.1.2	流程图简介	144
8.2	C 语言的语句概述	145
8.3	选择结构程序设计	146
8.3.1	关系运算符和关系表达式	146
8.3.2	逻辑运算符和逻辑表达式	147
8.3.3	if 语句	148
8.3.4	switch 语句	151
8.4	循环结构程序设计	153

8.4.1 循环语句概述	153	10.5 内部变量与外部变量	185
8.4.2 while 循环结构	153	10.5.1 内部变量	185
8.4.3 直到型循环 do-while 语句	153	10.5.2 外部变量	185
8.4.4 for 循环语句	154	10.6 内部函数和外部函数	187
8.4.5 循环嵌套	155	10.6.1 内部函数(又称静态函数)	187
8.4.6 break 语句与 continue 语句	156	10.6.2 外部函数	187
8.5 习题	157	10.6.3 多个源程序文件的编译和连接	188
第 9 章 数组	161	10.7 变量的动态存储与静态存储简介	188
9.1 一维数组的定义和引用	161	10.7.1 内部变量的存储方式	188
9.1.1 一维数组的定义	161	10.7.2 外部变量的存储方式	190
9.1.2 数组元素的引用	162	10.8 习题	190
9.1.3 一维数组元素的初始化	162	第 11 章 预处理命令	193
9.1.4 一维数组应用举例	162	11.1 宏定义	193
9.2 二维数组的定义和引用	164	11.1.1 不带参数的宏定义	193
9.2.1 二维数组的定义	164	11.1.2 带参数的宏定义	195
9.2.2 二维数组元素的引用	164	11.2 文件包含	196
9.2.3 二维数组元素的初始化	165	11.3 习题	197
9.2.4 二维数组应用举例	165	第 12 章 指针	199
9.3 字符数组与字符串	166	12.1 指针的概念	199
9.3.1 字符数组的逐个字符操作	166	12.2 变量的指针和指向变量的指针变量	200
9.3.2 字符数组的整体操作	168	12.2.1 指针变量的定义	200
9.3.3 常用的字符串处理函数	169	12.2.2 指针变量的引用	200
9.4 习题	171	12.2.3 指针变量作函数参数	202
第 10 章 函数	176	12.3 数组的指针和指向数组的指针变量	203
10.1 概述	176	12.3.1 用指针指向数组和数组元素	203
10.2 函数的定义与调用	176	12.3.2 通过指针引用数组元素	204
10.2.1 函数的定义	176	12.3.3 数组名与指针变量作函数参数	206
10.2.2 函数的返回值与函数类型	178	12.3.4 用指针处理一维数组应用举例	207
10.2.3 对被调用函数的说明和函数原型	179	12.3.5 指向多维数组的指针和指针变量	208
10.2.4 函数的调用	180	12.4 字符串的指针和指向字符串的指针变量	211
10.2.5 函数的形参与实参	180		
10.3 函数的嵌套调用和递归调用	181		
10.3.1 函数的嵌套调用	181		
10.3.2 函数的递归调用	182		
10.4 数组作为函数参数	183		
10.4.1 数组元素作为函数参数	183		
10.4.2 数组名作为函数的形参和实参	184		

12.4.1 字符串的表示形式	211	14.3.2 文件关闭 (fclose) 函数	241
12.4.2 字符串指针作函数参数	212	14.4 文件的读写	242
12.4.3 字符指针变量与字符数组	213	14.4.1 文件的读写函数	242
12.5 函数的指针和指向函数的指针变 量	214	14.4.2 fread 函数和 fwrite 函数	243
12.6 返回指针值的函数	215	14.4.3 格式化读写函数	244
12.7 指针数组和指向指针的指针	217	14.5 文件的随机读写	245
12.7.1 指针数组的概念	217	14.5.1 文件定位	245
12.7.2 指向指针的指针	218	14.5.2 文件的随机读写	245
12.8 习题	219	14.6 文件检测函数	246
第 13 章 结构体与共用体	223	14.7 习题	246
13.1 结构体类型与结构体变量	223	第 15 章 图形模式	248
13.1.1 结构体类型	223	15.1 图形设计基础	248
13.1.2 定义结构体变量	224	15.1.1 Turbo C 与图形接口	248
13.2 结构体类型变量的引用	226	15.1.2 Turbo C 图形库函数	249
13.3 结构体变量的初始化	226	15.1.3 Turbo C 绘图的基本步骤	250
13.3.1 对外部存储类型的结构体 变量进行初始化	226	15.2 图形模式设置	251
13.3.2 对内部存储类型的结构体 变量进行初始化	226	15.2.1 图形方式初始化	251
13.4 结构体数组	227	15.2.2 关闭图形方式	252
13.4.1 结构体数组的定义	227	15.3 基本图形绘制	253
13.4.2 结构体数组的初始化	228	15.3.1 画点	253
13.5 指向结构体类型数据的指针	228	15.3.2 画直线	253
13.5.1 指向结构体变量的指针	228	15.3.3 画矩形和条形	254
13.5.2 指向结构体数组的指针	229	15.3.4 画椭圆、圆和扇形图形	255
13.5.3 用结构体变量和指向结构 体的指针作函数参数	230	15.4 图形属性设置	256
13.6 共用体	232	15.4.1 屏幕颜色控制	256
13.6.1 共用体的概念	232	15.4.2 设置线条类型	257
13.6.2 共用体变量的引用	233	15.4.3 设置填充类型和填充颜色	258
13.7 枚举类型	233	15.5 图形方式下文本的输出	259
13.8 用 typedef 定义类型	235	15.5.1 文本输出函数	259
13.9 习题	235	15.5.2 定义文本字型	260
第 14 章 文件	239	第 16 章 系统函数	262
14.1 C 语言文件概述	239	16.1 输入输出函数 “stdio.h”	262
14.2 文件类型指针	240	16.2 字符处理函数 “ctype.h”	265
14.3 文件的打开与关闭	240	16.3 字符串处理函数 “string.h”	266
14.3.1 文件的打开 (fopen) 函数	240	16.4 数学函数 “math.h”	267
		16.5 内存管理函数	269
		16.6 其他函数	270
		附录	271
		部分习题参考答案	272



第 **1** 部分

项目案例教程

- 第 1 章 汉字显示技术
- 第 2 章 贪吃蛇游戏设计
- 第 3 章 黑白棋游戏设计
- 第 4 章 电话本管理系统设计
- 第 5 章 迷宫游戏设计

第 1 章 汉字显示技术

内容提要

汉字显示技术是在单片机程序设计和嵌入式程序设计中要处理的技术问题,如何解决汉字显示问题是本项目需要考虑的难点。当前 LED 汉字显示屏在大型商场、车站、码头、地铁站以及各类办事窗口等越来越多的场所需要用到,如何使用显示器作为显示屏,利用 C 语言编写程序在显示器上显示汉字的程序呢?本项目利用已有项目案例源码,使用字模工具或汉字子模库,实现任意汉字的现实,并实现汉字的特效显示(汉字左右滚动)。

通过以项目为载体,引入 C 的基础知识的学习,达到满足项目的设计与改造的需要;通过第一个项目的学习,引导读者对 C 语言程序的框架有一个全面的概括性了解。

知识目标

- ☑ 熟练掌握 C 语言程序编辑环境、编辑程序、存储程序、调试程序、查看运行结构、单步跟踪
- ☑ 掌握十进制、二进制、八进制、十六进制的转换与表示
- ☑ 掌握汉字区位码、汉字机内码、汉字子模信息、汉字子模库
- ☑ 理解函数库包含、系统函数、用户自定义函数、主函数
- ☑ 理解函数的定义、函数的调用、形式参数、实际参数
- ☑ 掌握数据类型、变量、常量、全局变量、局部变量
- ☑ 掌握顺序结构、分支结构、循环结构
- ☑ 掌握逻辑运算、位操作
- ☑ 了解数组、指针、文件

1.1 项目需求

1.1.1 需求描述

LED 汉字显示屏在大型商场、车站、码头、地铁站以及各类办事窗口等越来越多的场所需要用到。LED 行业已成为一个快速发展的新兴产业,市场空间巨大,前景广阔。随着信息产业的高速发展,LED 显示作为信息传播的一种重要手段,已广泛应用于室内外需要进行服务内容和服务宗旨宣传的公众场所,例如户内外公共场所广告宣传、机场车站旅客引导信息、公交车辆报站系统、证券与银行信息显示、餐馆报价信息显示、高速公路可变情报板、体育场馆比赛转播、楼宇灯饰、交通信号灯、景观照明等。显然,LED 显示已成为城市亮化、现代化和信息化社会的一个重要标志。

那么如何使用显示器作为显示屏,利用 C 语言编写程序在显示器上显示汉字的程序呢?解决这一编程问题,我们首先必须了解有关汉字编码及字库、字模的知识。

1. 区位码

国家标准的汉字字符集（GB2312—80）在汉字操作系统中是以汉字库的形式提供的。汉字库结构作了统一规定，如图 1-1 所示，即将字库分成 94 个区，每个区有 94 个汉字（以位作区别），每一个汉字在汉字库中有确定的区和位编号（用两个字节），这就是所谓的区位码，区码的第一个字节表示区号，第二个字节表示位号，因而只要知道了区位码，就可知道该汉字在字库中的地址，每个汉字在字库中是以点阵字模形式存储的，一般采用 16×16 点阵形式，所谓 16×16 ，是每一个汉字在纵、横各 16 点的区域内显示的，每个点用一个二进位表示，存 1 的点，当显示时，可以在屏上显示一个亮点，存 0 的点，则在屏上不显示，这样把存某字的 16×16 点阵信息直接用来在显示器上按上述原则显示，则将出现对应的汉字，如一个“大”字的 16×16 点阵字模如图 1-2 所示，当用存储单元存储该字模信息时，将需 32 个字节地址，图 1-2 右边写出了该字模对应的字节值。“大”字区位码为 2083，表示它位于第 20 区第 84 个。

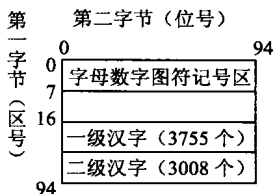


图 1-1 国标（GB2312—80）汉字字符集

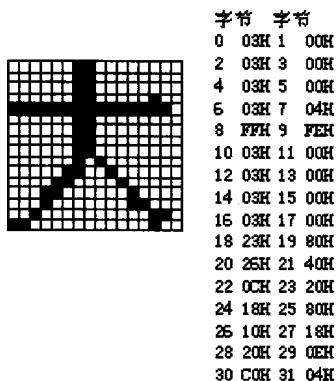


图 1-2 “大”的字模（ 16×16 ）

2. 内码

汉字使用两字节表示，国家制定了统一标准，称为国标码。国标码又称为汉字交换码，在计算机之间交换信息用。它用两个字节来表示，每个字节的最高位均为 0，因此可以表示的汉字数为 $2^{14}=16384$ 个。将汉字区位码的高位字节、低位字节各加十进制数 32（即十六进制数的 20），便得到国标码。为了区别于英文的 ASCII 码，国标码在计算机上使用的时候，规定汉字每个字节第一位设置为 1，以表示该两字节为汉字，称为内码。以“大”字为例子：

国标码 3473H: 00110100 01110011

内码 B4F3H: 10110100 11110011

区位码、国标码、内码之间有一定的转换公式，区位码的区码（16 进制）和位码（16 进制）分别加 20H 可以得到国标码，国标码的高字节和低字节分别加 80H 得到机内码，内码的两个字节分别减去 A0H 可以得到区位码。

3. 字模

汉字字模在字库中存放的位置根据汉字的区位码来确定，内码是汉字在机内的表示。由于区位码和内码存在固定的转换关系，所以当在支持汉字输入的系统中，键盘输入的汉字内码在程序中存在，可以将其转换为区位码，再从字库中找到对应的汉字字模，然后再用有关的位操作和循环语句，对每个字节的每一位进行判断，如同过滤一样，如果某位是 1，则按设置的颜

色在屏幕的相应位置画点（用 `graphics.h` 中的显示像素点的函数 `putpixel()`），若某位为 0，则不画点，这样就可按预先设置的颜色在相应位置显示出该汉字来。

本项目的功能便是利用字模在屏幕上显示“北京工业职业技术学院”和“信息系”这几个汉字，并且“北京工业职业技术学院”为 16 点阵楷体，“学院”为 32 点阵黑体，“信息系”为 24 点阵宋体。32 点阵和 24 点阵原理与 16 点阵的一样，只不过每一个汉字在纵、横各 32 点或 24 点的区域内显示的。因为不用字库，所以需要自己编写字模信息。

1.1.2 程序结构

汉字显示技术的项目程序可分为 4 部分，即 BGI 初始化部分、字模显示部分、字模信息部分和主函数部分。

1. BGI 初始化

由于图形程序运行并显示图像直接与显示器有关，而如何控制驱动显示器进行显示，Turbo C 并没有向用户提供这种技术，而这也是不必要的，因为它与显示器硬件结构息息相关，编程者并不需要知道这些东西，否则太复杂了。用户的图形程序要能运行并显示，则必须要包含有驱动显示器的这种程序。不同种类的显示器因硬件结构不同，因而驱动程序也不同，这些驱动程序一般由生产厂家提供，对一般标准的显示器，如 CGA、EGA、VGA 等，其驱动程序已经在 Turbo C 系统盘上提供。在用户的图形程序中，进行图形系统初始化时，即执行函数：

```
initgraph(&graphdriver, &graphmode, char *path_for_driver);
```

程序就按照 `path_for_driver` 所指的路径将图形驱动程序装入内存。这样，以后的图形功能才能被支持。若在所指路径下找不到相应显示器的驱动程序，或没有对驱动程序进行装入操作，则运行图形程序时，就会在屏上显示出错信息：

BGI Error: Graphics not initialized(use "initgraph")

若 Turbo C 已安装在硬盘上，则各种显示器的图形驱动程序：EGAVGA.BGI, CGA.BGI, HERC.BGI, IBM 8514.BGI, PC3270.BGI 和 BGI OBJ.EXE 等程序已展开在硬盘上，直接将其拷贝到自己的工作路径下即可，这样在集成环境下，生成的 .EXE 文件，就可以在 DOS 下直接运行了。生成能在 DOS 下直接运行的图形程序的另一种方法是：用 BGI OBJ.EXE 程序将相应显示器的驱动程序（如 CGA.BGI, EGAVGA.BGI 等）转换成 OBJ 文件，即在 DOS 下用命令（对 EGAVGA 显示器）：

```
C:\>BGI OBJ EGAVGA
```

这样就在用户的盘上生成了 EGAVGA.OBJ 文件，然后将该文件连到用户的库盘上 LIB 子目录下的 GRAPHICS.LIB 库中，即用 Turbo C 系统的 TLIB.EXE 文件，将 EGAVGA.OBJ 和 GRAPHICS.LIB 进行连接，可用命令：

```
C:\>TLIB LIB\GRAPHICS.LIB+EGAVGA.OBJ
```

这样就在用户的盘上的 GRAPHICS.LIB 中连入了 EGAVGA 驱动程序。

最后，在用户程序中的 `initgraph()` 函数调用前要使用 `registerbgidriver()` 进行登记，表示相应的驱动程序在用户程序进行连接时已连接，因而执行 `initgraph()` 时，不必再装入相应的 BGI 程序了，即使用（对 EGA, VGA 显示器）：

```
registerbgidriver(EGAVGA_driver)
```

后，执行 `initgraph()` 时，就不用再按该函数的第三个参数 `patch for driver` 去寻找驱动程序了。

对所有的 Turbo C 图形程序, 在 `initgraph()` 函数前, 加上 `registerbgidriver` 函数后 (当然对 `GRAPHICS.LIB` 还要进行如上述的连接操作), 编译连接后生成的 EXE 文件, 就变成了一个可独立运行的执行程序了, 它不需 Turbo C 环境的支持, 可在 DOS 下直接运行。

2. 字模显示部分

了解点阵汉字及汉字库的构成原理后, 显示汉字就变得简单了。以 16×16 点阵字库“大”字为例。“大”字的字模信息如图 1-2 所示。首先扫描第 0 个字节 03H (00000011), 再扫描第 1 个字节 00H (00000000), 根据“0”不打点, “1”打点的规则扫描完第一行。其他行依次类推, 输出结果图 1-3 所示。



图 1-3 扫描完一行

3. 字模信息部分

在汉字的点阵字库中, 每个字节的每个位都代表一个汉字的一个点, 每个汉字都是由一个矩形的点阵组成, 0 代表没有点, 1 代表有点。将 0 和 1 分别用不同颜色画出, 就形成了一个汉字, 对于 16×16 的矩阵来说, 它所需要的位数共是 $16 \times 16 = 256$ 个位, 每个字节为 8 位, 因此, 每个汉字都需要用 $256/8=32$ 个字节来表示。即每两个字节代表一行的 16 个点, 共需要 16 行, 显示汉字时, 只需一次性读取 32 个字节, 并将每两个字节为一行打印出来, 即可形成一个汉字。点阵结构如图 1-4 所示。字模信息部分是用字模数组表示的汉字点阵信息。所需的汉字字模点阵信息可以用“点阵字模工具”生成。

	第一个字节								第二个字节							
	17	16	15	14	13	12	11	10	27	26	25	24	23	22	21	20
0								→								→
1								→								
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																

图 1-4 点阵图

4. 主函数部分

主函数又叫 `main` 函数, `main` 函数是 C 程序的入口, 是每个 C 程序必须有的, 因为每个 C 程序都要从这个函数开始执行。调用其他函数后, 流程返回到 `main` 函数, `main` 函数的名称是系统规定的, 用户可以修改其内容即函数体, 但不能修改其名称和参数, 一个 C 程序必须有一个 `main` 函数, 也只能有一个 `main` 函数。

1.2 程序框架设计

1. 导入头文件:

```
#include "conio.h"
#include "graphics.h"
```

2. 函数的定义和声明:

```
void initgr(void) /* BGI初始化 */
void drawmat(char *mat,int matsize,int x,int y,int color);
/*依次: 字模指针、点阵大小、起始坐标(x,y)、颜色*/
int i,j;
```

3. 主函数

```
main()
{
    initgr(); /* BGI初始化 */

    drawmat(beil6,16,220,125,RED);
    drawmat(jing16,16,250,125,RED);
    ...
    getch(); /* 暂停一下,看看前面绘图代码的运行结果 */
    closegraph(); /* 恢复TEXT屏幕模式 */
}
```

1.3 项目程序实现

```
1      #include "conio.h"
2      #include "graphics.h"
3
4      void initgr(void) /* BGI初始化 */
5      {
6          int gd=DETECT,gm=0; /* 和gd=VGA,gm=VGAHI是同样效果 */
7          registerbgidriver(EGAVGA_driver);/*注册BGI驱动后可以不需要.BGI文件
          的支持运行*/
8          initgraph(&gd,&gm,"");
9      }
10
11     void drawmat(char *mat,int matsize,int x,int y,int color)
12     /*依次: 字模指针、点阵大小、起始坐标(x,y)、颜色*/
13     {int i,j,k,n;
14         n=(matsize-1)/8+1;
15         for(j=0;j<matsize;j++)
16             for(i=0;i<n;i++)
17                 for(k=0;k<8;k++)
18                     if(mat[j*n+i]&(0x80>>k)) /*测试为1的位则显示*/
19                         putpixel(x+i*8+k,y+j,color);
20     }
21     char beil6[]={
22         /* 以下是 '北' 的 16点阵楷体字模, 32 byte */
```

```

23      0x00,0x00,0x00,0x00,0x00,0x00,0x02,0x40,
24      0x02,0x40,0x02,0x40,0x02,0x4C,0x3E,0x50,
25      0x02,0x60,0x02,0x40,0x02,0x40,0x06,0x40,
26      0x3A,0x44,0x22,0x44,0x02,0x7E,0x00,0x00,
27      };
28      char jing16[]={
29      /* 以下是 '京' 的 16点阵楷体字模, 32 byte */
30      0x00,0x00,0x00,0x00,0x01,0x00,0x01,0x80,
31      0x00,0x86,0x7F,0xFF,0x08,0x08,0x0F,0xF8,
32      0x08,0x08,0x0F,0xF8,0x08,0x88,0x0C,0x90,
33      0x10,0x8C,0x20,0x84,0xC3,0x80,0x00,0x00,
34      };
35      char gong16[]={
36      /* 以下是 '工' 的 16点阵楷体字模, 32 byte */
37      0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
38      0x00,0x08,0x3F,0xFC,0x01,0x00,0x01,0x00,
39      0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,
40      0x01,0x0C,0x7F,0xFE,0x00,0x00,0x00,0x00,
41      };
42      char ye16[]={
43      /* 以下是 '业' 的 16点阵楷体字模, 32byte */
44      0x00,0x00,0x00,0x00,0x00,0x00,0x02,0x40,
45      0x02,0x40,0x02,0x40,0x02,0x4C,0x22,0x48,
46      0x12,0x58,0x1A,0x50,0x1A,0x60,0x12,0x40,
47      0x02,0x44,0x7F,0xFE,0x00,0x00,0x00,0x00,
48      };
49      char zhi16[]={
50      /* 以下是 '职' 的 16点阵楷体字模, 32 byte */
51      0x00,0x00,0x00,0x00,0x00,0x00,0x7E,0x00,
52      0x24,0x7C,0x24,0x44,0x3C,0x44,0x24,0x44,
53      0x3C,0x7C,0x24,0x44,0x26,0x28,0x3C,0x74,
54      0x64,0x46,0x04,0x82,0x05,0x02,0x00,0x00,
55      };
56      char ji16[]={
57      /* 以下是 '技' 的 16点阵楷书字模, 32 byte */
58      0x00,0x00,0x00,0x00,0x00,0x20,0x10,0x20,
59      0x10,0x24,0x14,0xFE,0xFE,0x28,0x1F,0xFC,
60      0x11,0x08,0x70,0x98,0x50,0x90,0x10,0x60,
61      0x10,0x70,0x51,0x9F,0x3E,0x04,0x00,0x00,
62      };
63      char shu16[]={
64      /* 以下是 '术' 的 16点阵楷体字模, 32 byte */
65      0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x20,
66      0x01,0x18,0x01,0x1C,0x7F,0xFE,0x03,0x80,
67      0x07,0x40,0x05,0x40,0x09,0x20,0x11,0x30,
68      0x21,0x1E,0x41,0x08,0x01,0x00,0x00,0x00,
69      };
70      char xue32[]={
71      /* 以下是 '学' 的 32点阵黑体字模, 128 byte */
72      0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
73      0x00,0x01,0x81,0x00,0x01,0x83,0x81,0xC0,
74      0x03,0x83,0x83,0xC0,0x01,0xC1,0x83,0x80,
75      0x01,0xC1,0xC7,0x00,0x00,0xC0,0x06,0x00,
76      0x00,0x00,0x0E,0x00,0x1F,0xFF,0xFF,0xF8,

```

```

77     0x1F, 0xFF, 0xFF, 0xF8, 0x18, 0x00, 0x00, 0x18,
78     0x18, 0x00, 0x00, 0x18, 0x18, 0x00, 0x00, 0x18,
79     0x19, 0xFF, 0xFF, 0x18, 0x01, 0xFF, 0xFF, 0x00,
80     0x01, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x3E, 0x00,
81     0x00, 0x00, 0xF0, 0x00, 0x00, 0x01, 0xC0, 0x00,
82     0x00, 0x01, 0xC0, 0x00, 0x3F, 0xFF, 0xFF, 0xFC,
83     0x3F, 0xFF, 0xFF, 0xFC, 0x00, 0x01, 0xC0, 0x00,
84     0x00, 0x01, 0xC0, 0x00, 0x00, 0x00, 0x01, 0xC0, 0x00,
85     0x00, 0x01, 0xC0, 0x00, 0x00, 0x00, 0x01, 0xC0, 0x00,
86     0x00, 0x0F, 0xC0, 0x00, 0x00, 0x0F, 0xC0, 0x00,
87     0x00, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
88     };
89     char yuan32[]={
90     /* 以下是 '院' 的 32点阵黑体字模, 128 byte */
91     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
92     0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x18, 0x00,
93     0x00, 0x00, 0x1C, 0x00, 0x1F, 0xE0, 0x0C, 0x00,
94     0x1F, 0xEF, 0xFF, 0xFC, 0x18, 0xE7, 0xFF, 0xF8,
95     0x18, 0xC6, 0x00, 0x18, 0x18, 0xC6, 0x00, 0x18,
96     0x18, 0xC6, 0x00, 0x18, 0x19, 0x80, 0x00, 0x00,
97     0x19, 0x81, 0xFF, 0xE0, 0x19, 0x81, 0xFF, 0xE0,
98     0x19, 0x80, 0x00, 0x00, 0x19, 0xC0, 0x00, 0x00,
99     0x18, 0xE0, 0x00, 0x00, 0x18, 0x67, 0xFF, 0xF8,
100    0x18, 0x77, 0xFF, 0xF8, 0x18, 0x70, 0x63, 0x00,
101    0x18, 0x70, 0x63, 0x00, 0x18, 0x70, 0x63, 0x00,
102    0x19, 0xF0, 0xE3, 0x00, 0x19, 0xE0, 0xC3, 0x00,
103    0x19, 0x80, 0xC3, 0x00, 0x18, 0x01, 0xC3, 0x08,
104    0x18, 0x03, 0x83, 0x0E, 0x18, 0x0F, 0x03, 0x0C,
105    0x18, 0x3E, 0x03, 0xFC, 0x18, 0x1C, 0x03, 0xF8,
106    0x18, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
107    };
108    char xin24[]={
109    /* 以下是 '信' 的 24点阵宋体字模, 72 byte */
110    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x06,
111    0x00, 0x03, 0x03, 0x00, 0x02, 0x03, 0x04, 0x07,
112    0xFF, 0xFE, 0x04, 0x00, 0x00, 0x0C, 0x00, 0x00,
113    0x0C, 0x00, 0x10, 0x1C, 0x3F, 0xE0, 0x14, 0x00,
114    0x00, 0x24, 0x00, 0x10, 0x44, 0x7F, 0xF8, 0x04,
115    0x00, 0x00, 0x04, 0x00, 0x00, 0x04, 0x20, 0x30,
116    0x04, 0x3F, 0xD0, 0x04, 0x20, 0x10, 0x04, 0x20,
117    0x10, 0x04, 0x20, 0x10, 0x04, 0x3F, 0xF0, 0x04,
118    0x20, 0x10, 0x04, 0x20, 0x10, 0x00, 0x00, 0x00,
119    };
120    char xi24[]={
121    /* 以下是 '息' 的 24点阵宋体字模, 72 byte */
122    0x00, 0x00, 0x00, 0x00, 0x20, 0x00, 0x00, 0x30,
123    0x00, 0x00, 0x20, 0x00, 0x03, 0xFF, 0xE0, 0x02,
124    0x00, 0x40, 0x02, 0x00, 0x40, 0x02, 0x00, 0x40,
125    0x03, 0xFF, 0xC0, 0x02, 0x00, 0x40, 0x02, 0x00,
126    0x40, 0x03, 0xFF, 0xC0, 0x02, 0x00, 0x40, 0x02,
127    0x00, 0x40, 0x03, 0xFF, 0xE0, 0x02, 0x10, 0x60,
128    0x01, 0x88, 0x20, 0x09, 0x8C, 0x18, 0x09, 0x84,
129    0x0C, 0x09, 0x80, 0x8C, 0x11, 0x80, 0x44, 0x31,
130    0x80, 0xE0, 0x20, 0xFF, 0xC0, 0x00, 0x00, 0x00,

```