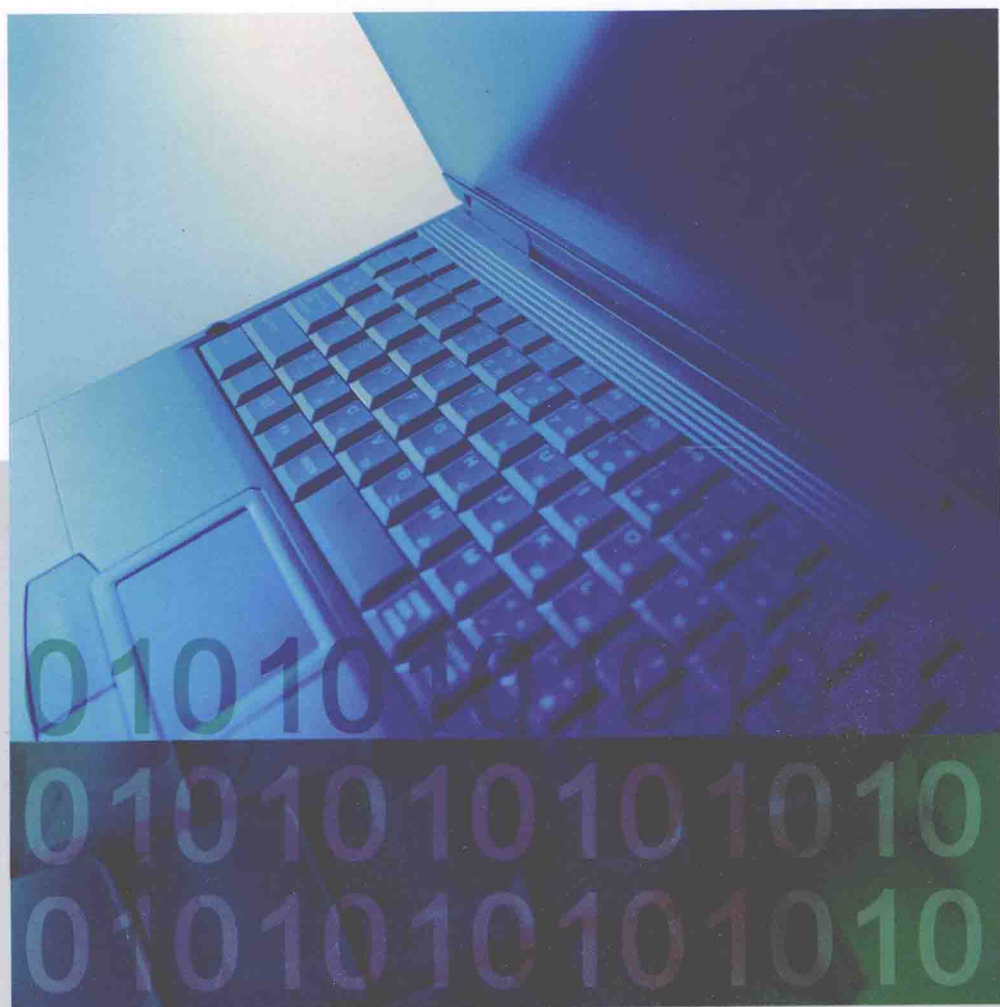


新世纪高等学校计算机系列教材

C语言实验与课程设计

李 开 卢 萍 曹计昌 编著



科学出版社

内容简介

· 新世纪高等学校计算机系列教材 ·

C 语言实验与课程设计

李开 卢萍 曹计昌 编著

第1版 (1991) 第2版 (1994)

1. 本书可作为高等院校计算机专业及相关专业的教材。

2. 本书可作为从事计算机工作的工程技术人员参考。

3. 本书可作为计算机爱好者的参考。

4. 本书可作为高等院校计算机专业及相关专业的教材。

5. 本书可作为从事计算机工作的工程技术人员参考。

6. 本书可作为计算机爱好者的参考。

7. 本书可作为高等院校计算机专业及相关专业的教材。

8. 本书可作为从事计算机工作的工程技术人员参考。

9. 本书可作为计算机爱好者的参考。

10. 本书可作为高等院校计算机专业及相关专业的教材。

11. 本书可作为从事计算机工作的工程技术人员参考。

12. 本书可作为计算机爱好者的参考。

13. 本书可作为高等院校计算机专业及相关专业的教材。

14. 本书可作为从事计算机工作的工程技术人员参考。

15. 本书可作为计算机爱好者的参考。

16. 本书可作为高等院校计算机专业及相关专业的教材。

17. 本书可作为从事计算机工作的工程技术人员参考。

18. 本书可作为计算机爱好者的参考。

19. 本书可作为高等院校计算机专业及相关专业的教材。

科学出版社

北京

内 容 简 介

本书既是与曹计昌、卢萍、李开编著的《C 语言程序设计》配套使用的实践教材,也适合与其他 C 语言类教材配套使用,其内容主要包括实验与课程设计两大部分。实验部分首先较详细地介绍了 Visual C++ 6.0, Turbo C 2.0/3.0 和 Cygwin 开发环境的使用方法,然后给出了精心设计且形式多样的 10 个基础实验,可让学生较好地掌握 C 语言的语法知识、上机调试技巧和代码编写等基本技能;课程设计部分以学生住宿信息管理系统设计为例,详细介绍了其设计与实现方法,同时还对输入/输出库函数、简单 C 编译预处理程序、词法分析程序、模拟器和汇编程序的设计,以及课程设计报告的撰写格式要求等也都作了适当的介绍。此外,书中还给出了三套模拟考试试题及其参考答案供学生自测使用。

本书适合高等院校计算机类本科专业作为实践教材使用,也适合软件开发工程师和广大科技人员自学参考。

注:凡需要本书或其电子原稿备课者,可与唐元瑜老师联系(027-87807752,13907198295)。

图书在版编目(CIP)数据

C 语言实验与课程设计/李开,卢萍,曹计昌编著. —北京:科学出版社,2011
(新世纪高等学校计算机系列教材)

ISBN 978-7-03-030379-0

I. C… II. ①李… ②卢… ③曹… III. C 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2011)第 029529 号

责任编辑:张颖兵 唐 源/责任校对:董艳辉

责任印制:彭 超/封面设计:袁 作

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

安陆市鼎鑫印务有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2011 年 3 月第一版

开本:787×1092 1/16

2011 年 3 月第一次印刷

印张:16 1/2

印数:1—2 500

字数:455 000

定价:29.00 元

(如有印装质量问题,我社负责调换)

前言

本书是曹计昌、卢萍、李开编著的《C 语言程序设计》一书的配套实践教材。由于本书实验题目和课程设计题目在书中都独立给出,因此也适合与其他 C 语言类教材配套使用。

本书主要内容包括 6 章,其中:

第 1 章介绍了 C 语言常用的上机开发环境,包括 Visual C++ 6.0 集成开发环境、传统的 Turbo C 2.0/3.0 开发环境、可以在 Windows 下模拟 Linux 的 Cygwin 环境,以及跨平台 C/C++ 集成开发环境 Code::Blocks。

第 2 章基础实验则给出了熟悉集成开发环境、表达式和标准输入/输出、流程控制、函数与程序结构、编译预处理、标准库及用户自定义库、数组、指针、结构与联合,以及文件方面的 10 个实验。

第 3 章课程设计基础部分介绍了课程设计的任务要求、概要设计和详细设计方法、编程规范、人机交互界面设计、链表操作设计、数据文件 I/O 设计等。

第 4 章课程设计提高部分则介绍了 C 语言输入/输出库函数的程序设计、简单 C 编译预处理程序设计、C 语言程序的词法元素(token)分析提取程序设计、简单处理器的模拟器和汇编程序设计(Simulator and Assembler)。

第 5 章课程设计报告的撰写与评价介绍了课程设计报告的框架结构、程序运行演示、程序编码风格、课程设计报告的评价等。

第 6 章模拟试题部分给出了 3 套模拟试题及其参考答案。

在 C 语言常用的上机开发环境方面,学生可以任选 1 种或 2 种作为学习 C 语言编程实践的开发环境。但关键是要掌握程序的编辑、修改、编译、链接和执行的方法,以及如何调试程序,跟踪程序,通过断点控制程序,观察变量、表达式在程序运行过程中的取值,如何综合运用这些方法定位程序中的错误和排除错误。

在上机实验方面,本书对上机实验的题型进行了较大改进。除了传统的编程设计题之外,增加了源程序改错题,源程序完善、修改、替换题,程序的调试跟踪题等,其目的是为了培养学生在 C 语言程序设计实践方面的实际动手能力。即:培养学生跟踪程序、设置断点、调试程序的能力,排除程序中语法、逻辑方面错误与故障的能力,并养成细致、周密、严谨的编程作风。

实验中的源程序改错题,先给出一些含有典型、常见错误的源程序,让学生通过编译或运行发现错误,并加以改正,从而纠正初学者易犯的常见错误。

实验中的源程序修改替换题,也是先给出源程序,然后要求学生按要求编写修改替换原来程序中的部分代码。如:用 if 语句和 goto 语句替换 while 循环,用 switch 语句替换多个 if 语句,用 while 循环替换 for 循环等。使学生掌握同一种算法的多种不同实现。

实验中的跟踪调试结果填空题,先给出源程序,要求学生填写程序运行过程中的某些中间结果。强迫学生用断点设置、单步跟踪等调试程序,观察并填写相应的中间结果。

实验中的编程设计题,只给出题目,要求学生完成从源程序的编程、编辑、修改,编译产生目标代码,链接产生可执行文件,运行程序得到正确结果的全部环节。同时,要求学生上机前预先编写好程序,上机实验仅从源程序的编辑开始。考虑到学生上机实验的完成进度差别较大,因此每个实验安排 3 个题目,前 2 个必做,后一个供速度快的同学选做。3 个题目的难度依次为易、中、较难。

总之,上机实验是进一步深刻了解 C 语言的语法、语义,掌握用 C 语言进行初步程序设计所需要的方法和技能的一个重要环节。优秀的程序员毫无例外地都是在机器上“摸爬滚打”出来的。C 语言创始人 Dennis M. Ritchie 是如此,C++ 的创始人 Bjarne Stroustrup 也是如此。这些语言大师都是在机器上使用当时流行的计算机语言遇到巨大障碍的情况下,才产生了发明新的计算机语言的灵感和动力。Dennis M. Ritchie 直到 20 世纪 90 年代还在他的 386 机器上编程。因此,作为 C 语言的初学者一定要重视实验。

学习 C 语言与学习和使用任何一门其他程序设计语言一样。首先要理解程序设计语言的语法和语义,然后通过上机熟悉该语言集成开发环境,掌握编辑、修改、编译、链接、跟踪、调试程序等方面的技巧。最后,还要熟悉语言编译系统提供的库函数或类库,以便快速实现程序的设计与开发。上机实验就是学习和掌握上面三个环节的重要途径。

“预则立,不预则废”。上机实验前一定要预先编好有关程序,把上机的重点放到程序的编辑、修改、编译、链接、跟踪、调试程序方面来,放到观察程序运行过程中的中间结果和运行完毕后的运行结果方面来。最终使自己的大脑变成一台“计算机”,能够在脑袋中运行自己的程序,判断各种情况下程序的走向等等,从而提高上机的效率和程序设计的准确性。优秀的程序员往往不轻易“出手”,但出手往往令人拍案叫绝!因此,“预”是培养编程者养成一种深思熟虑的好的编程习惯。

课程设计是对 C 语言程序设计课堂理论学习、上机实验实践的一次综合性演练。课程设计的目的是,使学生在已经学习完 C 语言程序设计课程并且初步掌握了一些基本算法的基础上,通过课程设计使学生具备用 C 语言编写小型软件的能力。根据我们近 10 年来的课程设计教学实践,以及近两年在我校计算机学院创新实验班(ACM 班)的教学和课程设计经验,本书将课程设计分为基础部分和提高部分。

课程设计的基础部分,以一般信息管理系统的信息处理为研究开发对象,着重强调用 C 语言编程解决一般性的应用问题,包括:基于菜单的人机交互技术,基于十字交叉链表的动态数据结构及设计技术,以及基于数据文件的文件 I/O 技术。使学生在完成基础部分的课程设计之后,既具有一定的需求分析能力,又具有编写一定难度和规模的小型软件的能力,为今后分析和编写更大规模的软件奠定扎实的基础。

课程设计的提高部分对计算机专业的学生更显重要。如果用 C 语言处理课程设计基础部分的一般应用问题类似于“开车”，则用 C 语言处理课程设计提高部分的程序设计问题就属于“造车”。课程设计提高部分的前 3 个题目可以使学生以极为原始和极为直接的方式感知编写库函数、编写编译软件的重要和艰辛，同时也会使学生感到能够做一些库函数设计和初步编译处理程序设计所获得的成就与骄傲！

而课程设计提高部分的第 4 个题目，即简单处理器的模拟器和汇编程序设计，同样可以使学生以极原始和极朴素的直觉感知计算机的指令系统、体系结构、存储组织、取指令、分析指令、执行指令的过程，以及通过编程实现小汇编程序设计，即根据指令格式将汇编语言程序通过自己编写的小汇编程序汇编成为目标程序，再交给自己编写的仿真处理器执行，最终得到运行结果。这些将为学习汇编语言程序设计、编译原理、计算机组成原理等后续课程，打下极为有利的基础，将使学生带着问题、带着探索的兴趣进入后续课程的广阔天地！

本书第 1、第 2 章由卢萍老师执笔，第 3、第 4、第 5、第 6 章及附录由李开老师执笔，曹计昌老师撰写了前言并对全书进行了统稿和审定。

本书实验部分参考了曹计昌、卢萍、王天江、王涛等老师 2004 年编著的《C 语言程序设计实验指导书》(电子版)和曹计昌、卢萍、李开、张茂元等老师 2008 年编著的《C 语言程序设计实验指导书(第二版)》(电子版)。课程设计基础部分参考了曹计昌、卢萍、李开、李专、周时阳、祝建华、张茂元等老师近 10 年来的课程设计教学资料。课程设计提高部分主要参考了曹计昌老师近两年来在本校计算机学院创新实验班的课程设计教学资料。3 套模拟试题实际上是本校计算机学院本科专业近年来 C 语言程序设计课程的考试试卷，是 C 语言课程组集体智慧的结晶。在此，谨向这些提供无私支持的老师们致以诚挚的谢意！感谢他们默默无闻的奉献！同时，也衷心希望我们的学生将来能够成为编程高手，成为计算机领域的优秀人才！

编著者

2011 年 1 月于华中科技大学

目 录

第 1 章 C 语言常用上机开发环境	1
1.1 Visual C++ 6.0 集成开发环境	1
1.1.1 Visual C++ 6.0 的启动	1
1.1.2 VC 的用户界面	2
1.1.3 创建工程及其源文件	5
1.1.4 编译、链接和运行	7
1.1.5 关闭程序工作区	8
1.1.6 联机帮助	8
1.1.7 程序调试	8
1.1.8 运行多文件组成的 C 程序的方法	12
1.1.9 命令行参数处理	13
1.1.10 静态连接库的生成和使用	13
1.2 Turbo C 2.0/3.0 开发环境	15
1.2.1 Turbo C 2.0 的启动	16
1.2.2 设置工作环境目录	17
1.2.3 运行 C 程序的一般步骤	17
1.2.4 程序动态调试方法——单步执行	18
1.2.5 程序动态调试方法——设置断点	20
1.2.6 建立、运行多文件程序的步骤	21
1.2.7 带参数的 main 函数的运行方法	22
1.2.8 使用 TLIB 建立自己的函数库	22
1.3 Cygwin 环境	23
1.3.1 Cygwin 的安装	23
1.3.2 Cygwin 的使用	26
1.3.3 Cygwin 的一些常用命令	26
1.3.4 编译器 gcc 的使用	27
1.3.5 用 GDB 调试程序	29
1.3.6 多源文件的编译方法	36
1.3.7 指定 main 函数的参数	37
1.3.8 gcc 生成静态库和动态库	37
1.4 跨平台 C/C++ 集成开发环境 Code::Blocks	38
1.4.1 Code::Blocks 的安装	38
1.4.2 环境配置	38

第 2 章 基础实验	40
2.1 熟悉 C 语言程序的开发环境	40
(一) 实验目的	40
(二) 实验内容	40
(三) 实验步骤及要求	42
2.2 表达式和标准输入与输出实验	49
(一) 实验目的	49
(二) 实验内容及要求	50
2.3 流程控制实验	51
(一) 实验目的	51
(二) 实验内容及要求	52
2.4 函数与程序结构实验	53
(一) 实验目的	53
(二) 实验内容及要求	53
2.5 编译预处理实验	58
(一) 实验目的	58
(二) 实验内容及要求	59
2.6 标准库及用户自定义库实验	61
(一) 实验目的	61
(二) 实验内容及要求	61
(三) 静态库的生成和使用	62
2.7 数组实验	63
(一) 实验目的	63
(二) 实验内容及要求	63
2.8 指针实验	66
(一) 实验目的	66
(二) 实验内容及要求	66
(三) 指定 main 函数的参数	69
2.9 结构与联合实验	69
(一) 实验目的	69
(二) 实验内容及要求	70
2.10 文件实验	72
(一) 实验目的	72
(二) 实验内容及要求	72
第 3 章 课程设计基础部分	75
3.1 任务要求	75
3.2 概要设计	77
3.2.1 功能结构设计	77
3.2.2 数据结构设计	80
3.3 详细设计方法与编程规范	85

3.3.1 设计方法	85
3.3.2 开发工具选择	86
3.3.3 编程规范	86
3.4 人机交互界面设计	86
3.4.1 控制台窗口和屏幕缓冲区	87
3.4.2 在屏幕上指定位置输出信息	89
3.4.3 弹出窗口的设计	95
3.4.4 键盘和鼠标输入信息的获取	107
3.4.5 输入处理	108
3.4.6 菜单操作与系统功能函数的调用	110
3.5 链表操作设计	119
3.5.1 动态数据结构设计	119
3.5.2 创建链表	120
3.5.3 链表结点的插入、删除和结点数据的修改	124
3.5.4 链表结点的查找	129
3.5.5 链表的遍历和排序	136
3.6 数据文件 I/O 设计	140
3.6.1 数据文件存储结构设计	140
3.6.2 数据加载	140
3.6.3 数据保存	144
3.6.4 数据备份	146
3.6.5 数据恢复	149
第 4 章 课程设计提高部分	153
4.1 C 输入/输出库函数的程序设计	153
4.2 简单 C 编译预处理程序设计	154
4.3 C 程序的词法元素(token)分析提取程序设计	156
4.4 模拟器和汇编程序设计(Simulator and Assembler)	157
第 5 章 课程设计评价	175
5.1 确定题目	175
5.2 程序功能演示的评价	175
5.3 课程设计报告的评价	176
5.4 程序风格的评价	176
5.5 最终成绩的评定	177
第 6 章 模拟试题	178
6.1 模拟试题一	178
6.2 模拟试题二	186
6.3 模拟试题三	195
6.4 模拟试题参考答案	206
参考文献	210

附录	211
附录 1 VC 中常见的编译错误及其警告信息	211
附录 2 Turbo C 中常见的上机错误及警告信息	212
附录 3 实验报告参考样本	216
附录 4 课程设计实例程序中用到的 dorm.h 头文件内容	217
附录 5 编程规范	223
附录 6 控制台函数用法说明	231
附录 7 常用键虚拟键码、扫描码、ASCII 码对照表	244
附录 8 Simulator and Assembler	245
附录 9 课程设计报告样例	253



C 语言常用上机开发环境

用 C 编写的程序,需要经过编译器编译后,才能转化成机器能够执行的二进制代码。当程序中有逻辑错误时,常常需要借助调试器帮助跟踪程序,观察程序中特定变量的值,找到 bug 出在什么位置。在学习中,一定要重视上机实践环节,通过上机实践加深理解有关概念,巩固理论知识,并掌握调试程序的方法与技巧。

C/C++ 的开发环境非常多,对于学习 C/C++ 语言而言,用什么环境并不重要,重要的是学习 C/C++ 语言本身。本章介绍目前常用的 C 语言开发环境,包括 Visual C++ 集成开发环境 IDE(Integrated Development Environment)、Turbo C 集成开发环境、Linux 下的 C 环境和跨平台的 IDE Code::Blocks 开发环境。这些开发环境可以满足不同用户对 C 语言运行环境的要求,读者可根据具体需求选用。通过对这些环境的使用,读者还可以更好地掌握编程技巧。

1.1 Visual C++ 6.0 集成开发环境

VC++ 6.0(简称 VC)是 Microsoft 公司推出的基于 Windows 平台的可视化、面向对象的软件开发环境,它不仅提供了功能强大的 C/C++ 开发环境,而且加入了微软提供的功能强大的 MFC(Microsoft Foundation Class)类库。MFC 中封装了大部分 Windows API 函数和 Windows 控件,它包含的功能涉及到整个 Windows 操作系统。MFC 不仅给用户提供了 Windows 图形环境下应用程序的框架,而且还提供了创建应用程序的组件。这样,开发人员不必从头设计创建和管理一个标准 Windows 应用程序所需的程序,而是从一个比较高的起点编程,故节省了大量的时间。另外,它提供了大量的代码,指导用户编程时实现某些技术和功能。因此,使用 VC 提供的高度可视化的应用程序开发工具和 MFC 类库,可使应用程序开发变得简单。

初学者使用 VC 编写和调试 C 语言程序应在 VC 提供的控制台操作方式下进行。Win32 控制台程序(Win32 Console Application)是一类 Windows 程序,它不使用复杂的图形用户界面,程序与用户交互时通过标准的正文窗口和标准的输入输出流(I/O 流)进行。本节介绍使用 VC 编写和运行 Win32 控制台程序的方法,包括 VC 工作环境和程序的编辑、编译、链接、调试和跟踪,使读者能快速使用 Visual C++ 的集成开发环境进行 C 语言编程。

1.1.1 Visual C++ 6.0 的启动

在已安装好 VC 的 Windows 操作系统中,用鼠标单击任务栏的“开始”按钮,在“程序”菜单中选择“Microsoft Visual Studio 6.0”菜单,执行“Microsoft Visual C++ 6.0”命令,即进入 VC 的 IDE。IDE 是一个将程序编辑器、编译器、调试工具和其他建立应用程序的工具集成在一起的用于开发应用程序的软件体系。Visual C++ 6.0 界面是一个由菜单栏(Menu Bars)、

工具栏(ToolBars)和一系列窗口组成的集成界面,如图 1.1 所示。



图 1.1 Visual C++ 的 IDE 用户界面

1.1.2 VC 的用户界面

1. 菜单栏

菜单以文字和层次化的方式提供命令接口。VC 的菜单栏包含 9 个菜单,即文件(File)、编辑(Edit)、查看(View)、插入(Insert)、工程(Project)、组建(Build)、工具(Tools)、窗口(Window)和帮助(Help)。

“文件”菜单中的命令主要用来对文件和项目进行操作,如“新建”、“打开”、“保存”等。

“编辑”菜单中的命令用来使用户方便、快捷地编辑文件内容,如进行删除、复制等操作,其中的大多数命令功能与 Windows 中标准字处理程序的编辑命令一致。

“查看”菜单中的命令主要用来改变窗口和工具栏的显示方式,打开调试时的各个窗口等。

“插入”菜单中的命令主要用于创建和添加项目及资源。

“工程”菜单中的命令主要用于项目的一些操作,如向项目中添加源文件等。

“工具”菜单中的命令主要用于选择或定制开发环境中的一些实用工具。

“窗口”菜单中的命令主要用于文档窗口的操作,如排列文档窗口、打开或关闭一个文档窗口、重组或切分文档窗口等。

由于很多菜单项一般是用不到的,故下面仅介绍常用的几个菜单项。

1) File 菜单

- 新建(New):创建新的文件、工程、工作区或其他文档。具体的操作见 2.2 节。
- 打开(Open):用于打开一个已存在的文件。注意,这个菜单项不是为打开项目准备的,而只是用来打开单独的文件,因此它的默认打开类型是 .h 和 .cpp 的文件。
- 关闭(Close):关闭当前被打开且处于活动状态的文件。
- 打开工作空间(Open Workspace):打开一个已存在的工作区文件(.dsw)。
- 保存工作空间(Save Workspace):保存当前的工作区状态。这个功能并不常用,因为当 VC 在关闭的时候往往会自动执行这个操作。

- 关闭工作空间(Close Workspace):关闭当前的工作区。
- 保存(Save):保存当前处于被编辑状态的源文件。
- 另存为(Save As):将当前文件另存为一个新文件。
- 保存全部(Save All):保存目前所打开的所有文件,即保存目前对该项目所做的一切修改。

2) Project 菜单

- 设置活动工程(Set Active Project):用于激活另一个已打开的工程。
- 增加到工程(Add to Project):该菜单有一个级联菜单,其中:
 - New:用来向工程中添加新文件。
 - File:用来向工程中插入已存在的文件。
- 设置(Setting):设置工程编译和调试的选项。

3) Build 菜单

- 编译(Compile XXX):编译当前的C或C++源文件生成其对应的.obj文件。
- 组建(Build):首先编译所有没编译过或已被修改过的源文件,链接它们的.obj文件和其他的文件生成最终的.exe文件。
 - 全部重建(Rebuild All):同样用来构建.exe。它和Build的区别是:Build只编译过时的.obj文件,而它不考虑目前这些的.obj是否为最新版本,故而重新编译所有源文件,然后链接产生.exe文件。一般很少使用这个功能。
- 执行(Execute XXX):运行程序。
- 开始调试(Start Debug):该菜单带有一个级联菜单,其中:
 - Go:开始或继续调试程序。
 - Step Into:单步运行调试。
 - Run Cursor:运行程序到光标所在行。

2. 工具栏

除快捷、简明的菜单栏外,VC还向用户提供了一种图形化的操作界面,即工具栏。它是由许多与菜单命令相关的按钮组成的,它与菜单栏的功能是重合的,但它具有更直观、更快捷的特点。熟练掌握工具栏的使用会使工作效率大大提高。启动VC时,在菜单栏下方有三个工具栏被自动打开:

(1) 标准(Standard)工具栏:它包括绝大多数标准工具,如打开、保存、剪切、复制、粘贴文件,以及可能用到的其他各种命令(把鼠标移到图标上会显示命令名)。

(2) 小型连编(Build MiniBar)工具栏:它提供开发和测试应用程序时最可能用到的编译、链接和运行命令。

(3) 向导(WizardBar)工具栏:它能执行许多类向导动作,而不用打开类向导。

此外,VC还有许多其他工具栏,这些工具栏都可以根据需要被打开或关闭。例如,选择“工具(Tools)/定制(Customize)”命令,此时可打开一个“定制(Customize)”对话框。在该对话框中点击“工具栏(ToolBars)”选项卡,如图1.2所示。此时,若要显示某工具栏则选择复选按钮,否则取消复选。

打开或关闭某个工具栏的另一个方法是,右击工具栏的空白处,在弹出的快捷菜单中复选其按钮,否则取消复选,如图1.3所示。

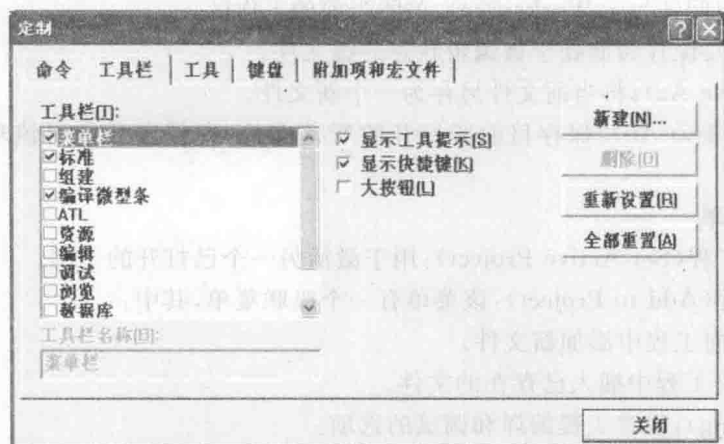


图 1.2 定制(Customize)对话框界面



图 1.3 右击工具栏的空白处打开/关闭工具栏

3. 项目工作区

项目工作区(Workspace)用于生成一个最终程序的各个源代码文件和其他辅助文件的集合,该集合被称为一个“项目(Project)”或“工程”。项目中所有的源文件都采用文件夹的方式进行管理,它将项目名作为文件夹名。在此文件夹下包含有源程序代码文件(.cpp,.h)、项目文件(.dsp)以及项目工作区文件(.dsw)等。

VC 的项目组成是十分复杂的,但是 VC 对项目有良好的自动维护机制。VC 通过左边的项目工作区窗口来对项目进行各种管理,它是确定开发项目中各部分位置的关键。“工作区”这个概念正如它的名字,就是用于描述用户当前工作状态的。比如打开了哪些文件、当前输入位置在什么地方等这一类信息被保存在工作区文件(.dsw)中,它在 VC 关闭的时候会自动保存。当用户下一次继续工作的时候,只需要打开这个工作区文件,就可以回到上一次关闭前的状态,而不需要再费心思回忆上一次到底改哪一行代码了。

工作区与项目并不一定是一一对应的,一个工作区中也可以包含两个或多个项目。但一般情况下,人们还是喜欢每次只在一个项目中进行工作,因此一个工作区中往往只有一个项目,或者说,每个项目往往都有与之对应的工作区。

当 VC 中有活动的项目时,项目工作区窗口的底部会出现一组选项卡,分别是 Class View

(类视图)、Resource View(资源视图)和 File View(文件视图)。每个视图都是以树形图的形式显示的,可以展开文件夹和其中的项查看其内容。如果某一项不可以再展开,那么它是可编辑的,双击这一项便可打开相应的文档编辑器进行编辑。对类和源程序文件来说,是打开文本编辑器;对于对话框来说是打开对话框编辑器等。每个视图还支持右键快捷菜单。

4. 编辑区

编辑区(Edit Area)是 VC 的一个最重要的组成部分,大部分工作都要在该区进行。当编辑 C/C++ 源程序代码时,它显示源程序代码;在设计对话框时,窗口绘制器也在此显示。另外,当设计应用程序中使用的图标时,编辑区将显示图标绘制器。

5. 输出窗口

输出(Output)窗口是显示编译程序的进展说明、警告及出错信息的地方,在调试程序时,它是 VC 调试程序显示所有变量当前值的地方。当关闭输出窗口后,它会在 VC 需要显示有关信息时自动打开。

1.1.3 创建工程及其源文件

1. 新建工程

在 VC 中要编制程序应该首先创建一个工程,因为只有这样,VC 才能选择合适的编译、链接选项(如果不先建立工程,而是一开始就写 .c 文件,那么编译的时候系统会询问是否建立一个缺省的项目工作区)。下面说明 VC 中工程的创建步骤及方法。

(1) 选择“文件(File)/新建(New)”命令,弹出“新建”对话框,在对话框中点击“工程(Project)”选项,再在左侧列表框中选“Win32 Console Application”工程类型,然后在右侧的“工程名称(Name)”文本框中输入工程名称,如“prj1”,在“位置(Location)”文本框中输入工程路径,如“d:\MyProject\prj1”,如图 1.4 所示。按“确定(OK)”按钮继续。



图 1.4 创建控制台工程

(2) 在打开的窗口中有 4 个选项,选择“一个空工程(An empty project)”选项,再单击窗口中的“完成(Finish)”按钮继续。

(3) 此时出现“新建工程信息”对话框,该对话框提示用户创建了一个空的控制台应用程序,并且没有任何文件被添加到新工程中。按“确定(OK)”按钮,则创建好了一个名为“prj1”的空白工程。创建结果如图 1.5 所示。

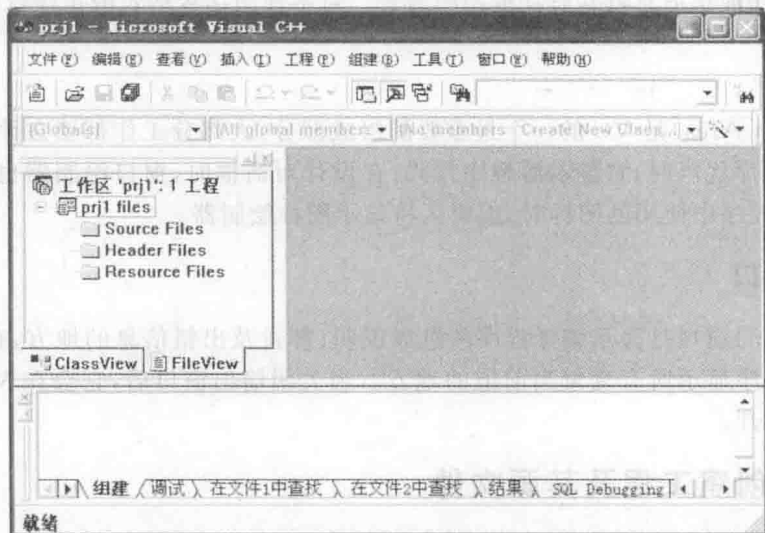


图 1.5 新建工程的工作区

在图 1.5 的工作区上可以看到两个视图:ClassView 和 FileView。对于仅编写 C 语言程序来说,ClassView 视图暂时不用,FileView 视图页面显示了当前项目中的所有文件。

了解有关概念:工程建立好后,在窗口中显示一些信息,对照自己建立的工程了解下列相关信息的含义。

(1) 工程:在图 1.5 中可以看到,VC 创建了一个称为 prj1 的工程。图中左边树状结构中的“prj1 files”结点代表了该工程。

(2) 文件和逻辑文件夹:在该工程下面有三个预定义的逻辑文件夹,即 Source Files, Header Files 和 Resource Files。这三个文件夹下面都没有文件,因为创建的是空工程。就编写单一源文件的 C 程序而言,只需要使用 Source Files 一个文件夹就够了。

这三个文件夹之所以被称为逻辑文件夹,是因为它们只是在工程的配置文件中定义的,在磁盘上并没有物理地存在。

(3) 工作区:在创建 prj1 工程的同时,也创建了一个名为 prj1 的工作空间,并且该工作空间只包含一个工程,如结点“工作区‘prj1’:1 工程”所示。

当创建新工程的时候,一个同名的工作区同时被创建,该工作区只包含一个项目,就是新创建的这个项目。

查看物理文件夹:打开 Windows 资源管理器,选择 d:\MyProject\prj1 目录,可以看到如图 1.6 所示的文件和文件夹。其中:

- .dsw:工作区描述文件。当要打开一个工程时,常用“文件(File)/打开工作区(Open Workspace)”命令打开这个项目文件夹中的.dsw 工作区文件。当 VC 在关闭的时候会自动保存.dsw 文件。

- .dsp:项目配置文件。它描述了项目中文件间的相互关系,包括逻辑文件夹在内的关于该项目的配置都保存在此文件中,其内容由 VC 自动维护,不需要也不应当被程序员修改。

.ncb:VC 内部使用的一个临时文件。

Debug 文件夹:编译连接时的输出文件(如. obj 和. exe 等)将被保存在该文件夹下。

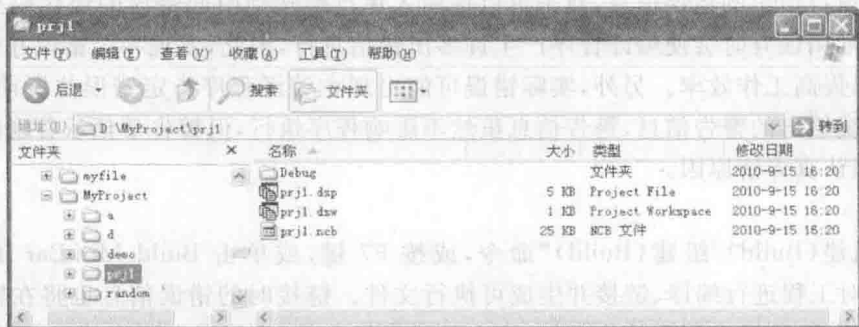


图 1.6 工程的物理文件夹

2. 编辑源文件

选择“文件(File)/新建(New)”命令,或直接按对应的快捷键“Ctrl+N”,此时可打开一个如图 1.7 所示的“新建”对话框,在对话框中点击“文件”选项,再在左侧列表中选择“C++ Source File”项,然后在右侧的“文件名”文本框中输入源文件名,如“test1.c”,选中“添加到工程”复选框。注意:文件的扩展名一定要用“.c”,以确保系统将输入的源程序文件作为 C 文件保存。否则,系统默认为 C++ 源程序文件(后缀为. cpp)。

按“确定(OK)”按钮,打开源文件编辑窗口,在其中输入源程序,然后选择“文件(File)/保存(Save)”命令或按“Ctrl+S”组合键存盘。

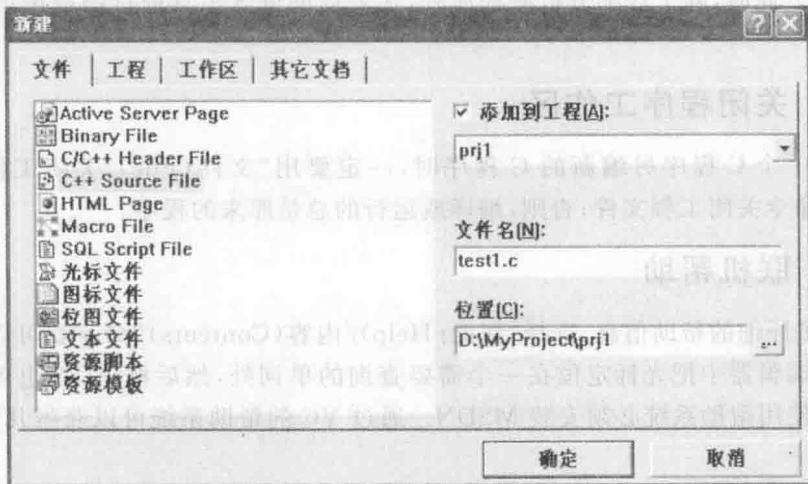


图 1.7 “新建”对话框

1.1.4 编译、链接和运行

1) 编译

选择“组建(Build)/编译(Compile)”命令,或按快捷键“Ctrl+F7”,或单击 Build MiniBar 工具栏上的 Compile 图标,该命令用来编译当前工作区的 C 或 C++ 文件。编译时的错误