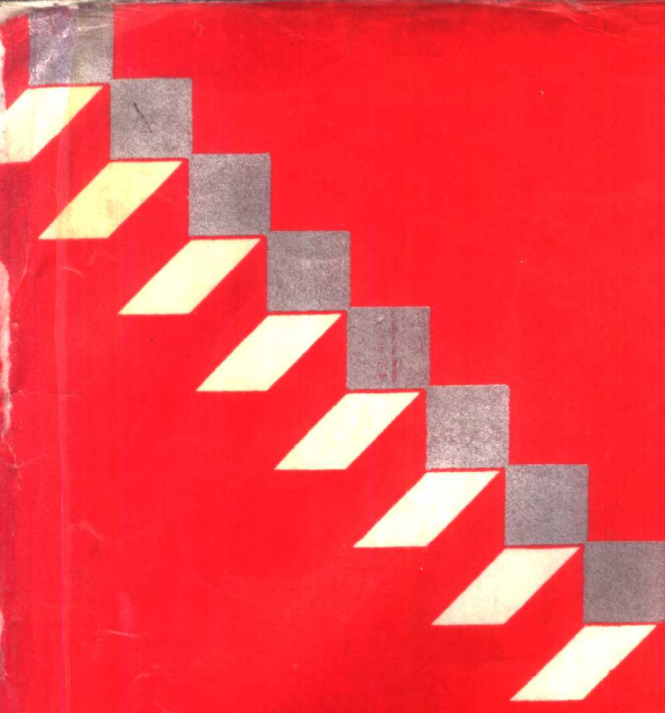




潘金贵 萧 柔 吴卫华等 编著
郑国梁 陈世福 主审

Borland C++ 程序设计教程



南京大学出版社

Borland C++程序设计教程

潘金贵 萧柔 吴卫华等 编著
郑国梁 陈世福 主审

南京大学出版社

1994 · 南京

(苏) 新登字第 011 号

内 容 简 介

本书系统地介绍了最新版本的面向对象程序设计语言 Borland C++ 及其程序设计技术。

全书分为三篇,共三十章五个附录。主要介绍了: Borland C++ 语言元素的规范定义、利用 Borland C++ 的 C++ 特征进行面向对象程序设计的技术、如何使用 C++ 2.0 流库、Borland C++ 高级程序设计技术、与汇编语言的交互、如何编写 Microsoft Windows 应用程序、如何将 Microsoft C 6.0 的程序转换成 Borland C++ 程序、Borland C++ 程序设计环境的使用方法和菜单选项参考、命令行选项的使用、Borland C++ 的类库、Borland C++ 软件包中的实用工具程序的功用并给出了 Borland C++ 的运行和编译时刻的错误及警告信息。

本书叙述通俗易懂,内容循序渐进,示例丰富,便于自学,强调 Borland C++ 面向对象程序设计的特点,注重方法与技巧的训练,有助于读者培养良好的程序设计风格。

本书既可作为高等院校计算机专业的本科生和研究生学习 C 和 C++ 语言的教材或教学参考书,亦可供有关专业的学生以及从事计算机系统软件及应用软件开发工作的工程技术人员作为 Borland C++ 软件包的使用手册。

欲购买与本书配套的例子盘和原版 Borland C++ & AF (V 2.0) 系统盘的读者可与编者联系。

Borland C++ 程序设计教程

潘金贵 萧柔 吴卫华等 编著

郑国梁 陈世福 主审

*

南京大学出版社出版

(南京大学校内)

江苏省新华书店发行 江苏淮安印刷厂印刷

*

开本: 787×1092 毫米 1/16 印张: 31.75 字数: 786 千字

1992 年 5 月第 1 版 1994 年 4 月第 2 次印刷

印数 16001—16000

ISBN 7-303-01660-8 TP·48

定价: 18.50 元

责任编辑: 顾其兵

前 言

一、面向对象的程序设计与 C++

面向对象程序设计方法提出了一种全新的程序设计思想，把数据和对数据的操作封装起来。目前，对面向对象的软件设计技术的研究和应用在我国正蓬勃发展，并越来越流行。

C++是一种面向对象的程序设计语言，它是 C 语言的进一步发展。C++语言是可移植的，可以很容易地将一个用 C++语言编写的程序从一个计算机系统移植到另一个计算机系统上。因此，可以毫不夸张地说，任何应用程序或系统程序均可用 C++语言来开发。

二、Borland C++

1. 语言特色

美国 Borland 国际公司在 Turbo C 的基础上，于 1990 年推出过面向对象的程序设计语言——Turbo C++。它继承并发展了原来 Turbo C 集成开发环境的优良特性，并引入了面向对象的基本思想和设计方法，曾是国际上最受欢迎的面向对象的程序设计工具。

1990 年 5 月，美国 Microsoft 公司推出了 Windows 3.0，立刻风靡全球，在微机软件世界引起了一场轰动。国际上许多软件公司为适应这场变革，纷纷推出了适用于 Windows 3.0 的软件包，或者对原有的流行软件进行改造，推出了适用于 Windows 的版本。Borland 国际公司为适应这一潮流，于 1991 年 2 月推出了适用于开发 Windows 应用软件的 C++版本——Borland C++ 2.0。

除了与早期版本的 Turbo C、Turbo C++完全兼容外，Borland C++ 2.0 还具有以下新特色：

- 与 AT&T C++和 ANSI C 完全兼容：Borland C++提供了用 C++方法进行编程的全部功能，完整地实现了 AT&T C++ 2.0 版的功能。

- Borland C++也实现了迄今为止最新的 ANSI C 标准。并且完全支持 Kernigham 和 Ritchie 的定义。另外，Borland C++包含有一定的扩展，使得程序员能够用混合语言和混合模型编程来扩充 PC 机的能力。

用户可以设置编译选项，以便在碰到扩充的部分时可以得到提示或警告。也可以设置编译选项，使编译器同等对待 Borland C++中扩充的关键字和常规标识符。

Borland C++中还扩充了 ANSI C 中 `#pragma` 指令所具有的处理非标准的与具体实

现细节无关的功能。

借助 Borland C++, 用户可以调用 ANSI C 及 C++ 的全部功能。可以单独用 ANSI C 编程, 也可以将 C 程序转换成 C++ 程序。

为了帮助用户入门, Borland C++ 中提供了一些 C++ 类库。这些类库使用类来完成各种功能。并提供了对 C++ 2.0 版流的支持。

- 支持 Microsoft Windows 环境下的应用程序开发: 可以用 Borland C++ 编制 Windows 应用程序。Borland C++ 中添加了许多新的功能, 包括资源编译器和资源开发工具。例子盘中提供了一些用 C++ 和 C 编制的 Windows 应用程序的实例, 有助于读者理解这方面的内容。

- 提供了预编译的头文件, 利用它可以缩短程序的编译时间。

- 提供了实模式和保护模式的编译器。Borland C++ 包括四种编译器, 它们分别为实模式和保护模式下的程序员平台和命令行编译器。每类编译器都可以编译 C 和 C++ 程序。在保护方式下运行 C++ 可以取得非常高的效率, 因为它不存在内外存交换的问题。

- 类库中提供了集合、数组等类型。

- 求助系统中添加了 Windows API (应用程序接口)。

- 崭新的 Borland 程序员平台: 它的新一代的用户接口, 其性能大大优于旧式的集成开发环境, 它提供了对计算机上全范围内程序和工具的访问能力。无论是在实模式还是在保护模式下运行, 它都能支持以下功能:

- 鼠标器;

- 多重覆盖窗口;

- 多重文件编辑;

- 内部汇编码;

- 嵌入式汇编器;

- 集成调试器;

- 对大缓冲区的恢复 (undo) 和重复 (redo) 功能, 等等。

- VROOMM 提供面向对象的实时虚拟存储管理: VROOMM 使程序员能够很简单地覆盖代码。程序员只须为覆盖选择代码段, 其他均由 VROOMM 自动处理, 可使用用户的代码保存在 640K 的空间内。

- 新程序员平台的联机访问。

- 联机的快速文本求助, 用户可用拷贝的模拟程序例子试验每一个函数。

- 有着许多独立的库函数, 包括堆检测函数和完整的复数, 以及 BCD 数学函数集, 能快速完成复杂的算术运算和复数运算。

其他特征包括:

- 对 -S 选择的扩展: C 源码作为注释可加到结果汇编代码上。

- 远程对象和大容量数组

- 候补配置文件: 用户可以创建几个配置文件, 并可在需要的时刻使用其中合适的一个。

- 用于命令行编译的响应文件。

- 快速完成复杂的算术运算和复数运算。

2. 硬件和软件环境

Borland C++运行于 IBM PC 系列计算机上, 包括 XT、AT、PS/2 和 IBM 全兼容机。Borland C++要求机器至少配有 3.0 以上版本的 DOS 和至少的 640K 内存。监视器为 80 列即可。最低的要求是一个硬盘和一个软盘驱动器。

Borland C++中包含了利用 8087 协处理器进行浮点运算的例程。如果没有协处理器, 它也可以进行仿真运算。虽然运行 Borland C++时不必有 8087 协处理器, 但 8087 协处理器可以有效地提高程序的运行效率。

Borland C++可支持鼠标器。如果配有鼠标器, 必须与下列要求之一全兼容。

- Microsoft Mouse 6.1 版或更高版本, 或者与此兼容的 mouse 版本。
- Genus Mouse 9.0 版或更高版本。
- IMSI Mouse 6.11 版或更高版本。
- Logitech Mouse 3.4 版或更高版本。
- Mouse Systems'PC Mouse 6.22 版或更高版本。

三、内 容 安 排

为了使我国广大的计算机工作人员能够更好地利用 Borland C++进行程序设计工作, 在南京大学出版社的支持下, 我们根据长期从事 C 语言编程和面向对象程序设计的经验, 参阅了国内外的有关软件、资料, 编译这套 Borland C++程序设计丛书, 第一批包括《Borland C++程序设计教程》、《Borland C++程序开发工具》和《Borland C++实用库函数大全》, 1993 年还将推出第二批。它们全面、系统地介绍了 Borland C++的基础知识和高级程序设计技术, 是一套引导读者进行 C、C++面向对象程序设计和进行 Windows 应用编程的参考书。

本书由两篇组成, 共二十章和一个附录。

第一篇由四部分组成。

第一部分 (第一至第八章) 介绍 Borland C++程序设计基础。

第一章“**OOP 与 C++**”是一个概论, 介绍了程序设计的传统方法和面向对象方法的基本概念, 以及 C 和 C++的特点, 将有助学习和理解 Borland C++语言及其程序设计技术。

第二章到第六章, 完整地介绍了 Borland C++的语言成份, 包括基本语言元素、数据结构、控制结构等各种语言成份和规范定义、使用方法, 以及利用 C++语言进行面向对象程序设计的内容。

第七章“**C++流**”介绍了如何使用 C++ 2.0 流库, 同时对 C++ 1.2 流库也作了介绍。

第八章“**C++程序实例**”提供了基于第四、五章所述概念的实际例子。

第二部分 (第九至第十二章) 讨论了 C++的高级程序设计技术, 其中:

第九章“**存储管理**”讨论了内存模式、混合模式编程及覆盖等有关内容。

第十章“**数学运算**”讨论了浮点数及 BCD 运算。

第十一章“**字符屏幕管理及图形程序设计**”讨论了用 Borland C++对文本和图形进行编

程的设施和方法。

第十二章“支持模块化设计的工程管理器”阐述如何使用工程管理器来管理由多文件组成的工程。

第三部分（第十三章）“与汇编语言的交互”讨论如何从 Borland C++ 程序中调用已编制好的汇编语言程序，还介绍了集成开发环境的内部汇编器。

第四部分（第十四到十六章）是为 Microsoft C 和 Microsoft Windows 程序员准备的。其中：

第十四章“Microsoft C 用户必读”介绍了如何将 Microsoft C 6.0 的程序转换成 Borland C++ 程序。

第十五章“C++ 的 Windows 编程入门”简单介绍了与 Microsoft Windows 有关的几个概念，并给出了用 C++ 编制的一个 Windows 应用程序的例子。

第十六章“Windows 环境下的应用程序开发”讨论了在 Windows 环境下如何组织和创建应用程序。

第二篇为“Borland C++ 程序设计环境”。由四章组成，其中：

第十七章“安装和启动”告诉读者怎样在计算机系统中安装 Borland C++，如何配置 Borland C++ 的颜色、默认值及其他设置。

第十八章“集成开发环境的使用”介绍了集成开发环境的功能，同时交待了如何使用集成开发环境的一些信息和示例，以使读者能方便地使用该软件。该章还讨论了如何启动和退出集成开发环境。

第十九章“集成开发环境菜单系统”提供了集成开发环境中菜单和选项的完整的参考信息。

第二十章“命令行编译器”讨论了如何使用命令行编译器。以及配置文件的用法。

附录“错误信息”，列出并解释了 Borland C++ 的运行和编译时刻可能出现的错误及警告信息，同时指出了可能的出错原因及解决的办法。

参加本书编著工作的人员有潘金贵、萧柔、吴卫华、顾铁成、虞育新、潘东旭、冀惠刚、忻超、李竹华、金子方等，潘金贵和萧柔同志对全书进行了仔细的修改和统编。

本书承蒙南京大学计算机系郑国梁教授和陈世福教授主审，顾铁成同志协助校阅了有关章节，在此谨向他们深表谢意。

由于水平、时间所限，不妥之处在所难免，欢迎广大读者批评指正。

编著者

一九九二年三月于南京

目 录

第一篇 Borland C++程序设计技术

第一部分 Borland C++程序设计基础

第一章 OOP 与 C++	(3)
1.1 什么是 OOP	(3)
1.1.1 面向过程的程序设计	(3)
1.1.2 结构化程序设计	(3)
1.1.3 面向对象的程序设计	(4)
1.2 C++	(5)
1.2.1 C 语言	(5)
1.2.2 C++ 和 Borland C++	(6)
第二章 Borland C++基本语言元素	(8)
2.1 空白符	(8)
2.1.1 用 \ 续行	(9)
2.1.2 注释	(9)
2.2 记号	(10)
2.3 关键字	(10)
2.3.1 标识符	(12)
2.4 常量	(13)
2.4.1 整型常量	(14)
2.4.2 字符常量	(15)
2.4.3 串文字	(18)
2.5 运算符描述	(21)
2.5.1 单目运算符	(22)
2.5.2 双目运算符	(22)
2.5.3 分隔符	(23)
2.6 说明	(26)
2.6.1 对象	(26)
2.6.2 左值	(27)
2.6.3 右值	(27)
2.6.4 类型和存储类	(27)
2.6.5 作用域	(27)

2.6.6	可见性	(28)
2.6.7	生存期	(29)
2.6.8	翻译单元	(30)
2.6.9	连接	(30)
2.6.10	名变换	(31)
2.7	说明语法	(31)
2.7.1	临时定义	(32)
2.7.2	可能的说明	(32)
2.7.3	外部说明和定义	(35)
2.7.4	类型说明符	(37)
2.7.5	类型分类	(37)
2.7.6	基类型	(38)
2.7.7	初始值	(40)
2.7.8	简单说明	(42)
2.7.9	存储类说明符	(42)
2.7.10	修饰符	(44)
2.7.11	复杂说明与说明符	(47)
2.8	指针	(48)
2.8.1	指向对象的指针	(49)
2.8.2	指向函数的指针	(49)
2.8.3	指针说明	(49)
2.8.4	指针与常量	(50)
2.8.5	指针算术运算	(50)
2.8.6	指针转换	(51)
2.8.7	C++引用说明	(51)
第三章	词汇结构语法	(52)
3.1	数组	(52)
3.2	函数	(52)
3.2.1	说明与定义	(52)
3.2.2	说明与原型	(53)
3.2.3	定义	(54)
3.2.4	形参说明	(55)
3.2.5	函数调用与参数转换	(55)
3.3	结构	(56)
3.3.1	无标记结构与 typedef	(56)
3.3.2	结构成员说明	(56)
3.3.3	结构与函数	(57)
3.3.4	存取结构成员	(57)
3.3.5	结构字对齐	(58)

3.3.6	结构名空间	(59)
3.3.7	不完整说明	(59)
3.3.8	位域	(59)
3.4	联合	(60)
3.4.1	联合说明	(61)
3.5	枚举类型	(61)
3.6	表达式	(63)
3.6.1	表达式与 C++	(67)
3.6.2	求值次序	(67)
3.6.3	出错与溢出	(67)
3.7	操作符语义	(67)
3.7.1	后缀和前缀操作符	(68)
3.7.2	增量或减量操作符	(69)
3.7.3	单目操作符	(69)
3.7.4	sizeof 操作符	(71)
3.7.5	乘法类操作符	(71)
3.7.6	加法类操作符	(71)
3.7.7	移位操作符	(72)
3.7.8	关系操作符	(73)
3.7.9	相等类操作符	(74)
3.7.10	按位与操作符 &	(75)
3.7.11	按位异或操作符 ^	(75)
3.7.12	按位同或操作符 	(75)
3.7.13	逻辑与操作符 &&	(75)
3.7.14	逻辑或操作符 	(76)
3.7.15	条件操作符 ?:	(76)
3.7.16	赋值操作符	(76)
3.7.17	逗号操作符	(77)
3.7.18	C++ 操作符	(78)
3.8	语句	(78)
3.8.1	分程序	(79)
3.8.2	标号语句	(80)
3.8.3	表达式语句	(80)
3.8.4	选择语句	(80)
3.8.5	循环语句	(81)
3.8.6	跳转语句	(83)
第四章	C++ 特性及示例	(85)
4.1	封装性	(85)
4.2	继承性	(87)

4.3	多态性	(88)
4.4	重载	(88)
4.5	用类来模拟现实世界	(89)
4.5.1	建立类的一个图形例子	(89)
4.5.2	说明对象	(90)
4.5.3	成员函数	(90)
4.5.4	调用一个成员函数	(91)
4.5.5	构造函数和析构函数	(91)
4.5.6	代码和数据相结合	(93)
4.5.7	成员访问控制: 私有、公有和保护	(94)
4.5.8	默认为私有的类	(94)
4.5.9	运行 C++ 程序	(95)
4.6	继承	(97)
4.6.1	对 Point 类的再认识	(97)
4.6.2	把类装入模块中	(100)
4.6.3	扩充类	(102)
4.6.4	多重继承	(105)
4.7	虚函数	(109)
4.7.1	虚函数的作用	(110)
4.7.2	定义虚函数	(111)
4.7.3	开发一个完整的图形模块	(112)
4.7.4	使用普通成员函数或虚成员函数	(119)
4.8	动态对象	(119)
4.8.1	析构函数和 delete	(120)
4.8.2	一个动态对象分配的例子	(121)
4.9	C++ 中更多的灵活性	(125)
4.9.1	在类定义之外的嵌入式函数	(126)
4.9.2	带有默认参数的函数	(126)
4.9.3	有关重载函数的进一步讨论	(127)
4.9.4	重载运算符以提供新的含义	(129)
4.9.5	友元函数	(132)
4.9.6	C++ 流库	(133)
4.10	用户定义数据类型的 I/O	(139)
4.11	下一步	(140)
4.12	小结	(140)
第五章	C++ 的进一步讨论	(141)
5.1	引用	(141)
5.1.1	简单引用	(141)
5.1.2	引用参数	(141)

5.2	作用域存取操作符	(143)
5.3	new 与 delete 操作符	(143)
5.3.1	错误处理	(144)
5.3.2	关于数组的 new 操作符	(144)
5.3.3	::operator new	(144)
5.3.4	用 new 操作符进行初始化	(144)
5.4	类	(145)
5.4.1	类名	(145)
5.4.2	类类型	(145)
5.4.3	类名作用域	(145)
5.4.4	类对象	(146)
5.4.5	类成员表	(146)
5.4.6	成员函数和友元函数	(146)
5.4.7	关键字 this	(146)
5.4.8	嵌入式函数	(147)
5.4.9	静态成员	(147)
5.4.10	成员作用域	(148)
5.4.11	基类与派生类存取	(150)
5.5	虚基类	(151)
5.6	类的友元	(152)
5.7	构造函数与析构函数	(153)
5.8	构造函数	(154)
5.8.1	默认构造函数	(155)
5.8.2	拷贝构造函数	(155)
5.8.3	构造函数的重载	(155)
5.8.4	构造函数的调用次序	(156)
5.8.5	类的初始化	(157)
5.9	析构函数	(160)
5.9.1	何时调用析构函数	(160)
5.9.2	atexit、#pragma exit 与析构函数	(160)
5.9.3	exit 与析构函数	(160)
5.9.4	abort 与析构函数	(160)
5.9.5	虚析构函数	(161)
5.10	重载操作符	(162)
5.11	操作符函数	(163)
5.11.1	重载操作符与继承	(163)
5.11.2	重载 new 和 delete	(163)
5.11.3	重载单目操作符	(164)
5.11.4	重载双目操作符	(165)

5.11.5	重载赋值操作符=	(165)
5.11.6	重载函数调用操作符()	(165)
5.11.7	重载下标操作符[]	(165)
5.11.8	重载类成员存取操作符->	(166)
5.12	虚函数	(166)
5.13	抽象类	(167)
5.14	C++作用域	(168)
5.14.1	类作用域	(168)
5.14.2	隐藏	(168)
5.14.3	C++作用域规则小结	(169)
第六章	预处理器	(170)
6.1	预处理指令	(170)
6.2	空指令	(172)
6.3	#define 与#undef 指令	(172)
6.3.1	简单的#define 宏	(172)
6.3.2	#undef 指令	(173)
6.3.3	-D 与-U 选择项	(174)
6.3.4	关键字与保留字	(174)
6.3.5	带参数的宏	(174)
6.4	文件包含指令#include	(176)
6.4.1	用<头文件名>搜索头文件	(177)
6.4.2	用"头文件名"搜索头文件	(177)
6.5	条件编译	(177)
6.5.1	#if, #elif, #else 和#endif 条件指令	(177)
6.5.2	#ifdef 和#ifndef 条件指令	(178)
6.6	#line 行控制指令	(179)
6.7	#error 指令	(180)
6.8	#pragma 指令	(181)
6.9	预定义宏	(184)
第七章	C++流	(187)
7.1	何谓流	(187)
7.1.1	iostream 库	(187)
7.1.2	输出	(189)
7.1.3	输入	(193)
7.1.4	简单文件 I/O	(194)
7.1.5	串流处理	(195)
7.2	流类参考	(197)
	filebuf <fstream.h>	(197)
	fstream <fstream.h>	(198)

fstreambase	<fstream.h>	(198)
ifstream	<fstream.h>	(199)
ios	<iostream.h>	(199)
iostream	<iostream.h>	(201)
iostream__withassign	<iostream.h>	(201)
istream	<iostream.h>	(201)
istream__withassign	<iostream.h>	(202)
istrstream	<strstream.h>	(202)
ofstream	<fstream.h>	(203)
ostream	<ostream.h>	(203)
ostream__withassign	<iostream.h>	(204)
ostrstream	<strstream.h>	(204)
streambuf	<iostream.h>	(204)
strstreambase	<strstream.h>	(205)
strstreambuf	<strstream.h>	(206)
strstream	<strstream.h>	(206)
第八章 C++程序实例		(207)
8.1 从C向C++过渡		(207)
8.1.1 程序1		(207)
8.1.2 程序2		(208)
8.1.3 程序3		(209)
8.1.4 程序4		(209)
8.2 对象支持		(211)
8.2.1 程序5		(211)
8.2.2 程序6		(213)
8.2.3 程序7		(216)
8.2.4 程序8		(218)
8.2.5 程序9		(220)
8.3 小结		(222)

第二部分 Borland C++高级程序设计技术

第九章 存储管理	(224)
9.1 内存溢出错误	(224)
9.1.1 8086寄存器	(224)
9.1.2 内存段	(226)
9.1.3 指针	(227)
9.1.4 六种内存模式	(229)
9.2 混合模式程序设计: 寻址修饰符	(233)

9.2.1	段指针	(233)
9.2.2	说明 far 对象	(234)
9.2.3	说明 near 或 far 函数	(234)
9.2.4	说明 near、far 或 huge 指针	(235)
9.2.5	使用库文件	(236)
9.2.6	连接混合模式	(236)
9.3	覆盖 (VROOMM)	(237)
9.3.1	覆盖的工作过程	(237)
9.3.2	要求	(239)
9.3.3	使用覆盖	(239)
9.3.4	覆盖程序	(240)
9.3.5	调度	(242)
第十章	数学运算	(244)
10.1	浮点选项	(244)
10.1.1	仿真 80x87 芯片	(244)
10.1.2	使用 80x87 代码	(244)
10.1.3	无浮点代码	(244)
10.1.4	快速浮点选项	(245)
10.1.5	87 环境变量	(245)
10.1.6	寄存器和 80x87	(245)
10.1.7	消除浮点异常	(246)
10.2	复数运算	(246)
10.2.1	BCD 运算	(247)
第十一章	视频函数	(249)
11.1	视频模式概述	(249)
11.2	窗口和视区概述	(249)
11.2.1	什么是窗口	(250)
11.2.2	什么是视区	(250)
11.2.3	坐标	(250)
11.3	文本模式下的程序设计	(250)
11.3.1	控制台 I/O 函数	(250)
11.3.2	文本窗口	(252)
11.3.3	text_modes 类型	(253)
11.3.4	文本颜色	(254)
11.3.5	高性能输出	(255)
11.4	图形模式下的程序设计	(255)
11.4.1	图形库函数	(255)
第十二章	支持模块化设计的工程管理器	(266)
12.1	使用工程管理器	(266)

12.2 错误跟踪	(269)
12.2.1 中止制作	(269)
12.2.2 多个源程序文件中的语法错误	(269)
12.2.3 保存或删除消息	(270)
12.3 工程管理器的功能	(270)
12.3.1 自动依赖性检查	(270)
12.4 使用其他文件译码器	(271)
12.5 重载库	(272)
12.6 工程管理器的其他特性	(273)
12.6.1 查看工程中的文件	(274)
12.6.2 工程记事	(275)

第三部分 与汇编语言的交互

第十三章 与汇编语言的交互	(276)
13.1 混合语言程序设计	(276)
13.1.1 参数传递序列	(276)
13.2 从 Borland C++调用.ASM 程序	(279)
13.2.1 建立从 Borland C++对.ASM 程序的调用	(279)
13.2.2 书写一个汇编语言例程	(281)
13.2.3 用 C++连接汇编程序模块	(286)
13.3 从汇编程序调用 C 和 C++例程	(287)
13.3.1 准备工作	(287)
13.3.2 一个调用汇编程序模块的例子	(288)
13.4 伪变量、嵌入式汇编和中断函数	(290)
13.4.1 伪变量	(290)
13.4.2 嵌入式汇编语言	(292)
13.4.3 中断函数	(298)
13.4.4 使用低层支撑的练习	(298)

第四部分 Windows 应用编程技术

第十四章 Microsoft C 用户必读	(301)
14.1 环境与工具	(301)
14.1.1 IDE 与 Windows	(302)
14.1.2 .h 和.LIB 文件的路径	(302)
14.1.3 MAKE	(303)
14.1.4 命令行编译器	(306)
14.1.5 命令行选项和库的兼容性	(311)

14.1.6	连接器	(311)
14.2	源级兼容性	(313)
14.2.1	__MSC	(313)
14.2.2	头文件	(313)
14.2.3	内存模式	(313)
14.2.4	关键字	(314)
14.2.5	寄存器约定	(314)
14.2.6	浮点返回值	(314)
14.2.7	值返回结构	(315)
第十五章	用 C++ 进行 Windows 编程入门	(316)
15.1	面向对象的思考	(316)
15.2	Windows 词汇	(316)
15.3	Window 环境下的 C 程序设计	(318)
15.4	WHELLO 应用程序	(323)
15.4.1	伪指令和初始化	(323)
15.4.2	定义 Main 类	(324)
15.4.3	定义 Windows 类	(325)
15.4.4	定义基类	(325)
15.4.5	定义派生类	(326)
15.5	指向 Windows 的指针	(328)
15.6	回头调用函数	(329)
15.7	WinMain 各种版本比较	(330)
第十六章	Windows 环境下的应用程序开发	(331)
16.1	基本处理过程	(331)
16.2	在集成开发环境中编译和连接	(332)
16.2.1	资源文件	(332)
16.2.2	模块定义文件	(332)
16.2.3	编译和连接例子程序	(332)
16.3	从命令行编译和连接	(334)
16.3.1	从命令行进行编译	(334)
16.3.2	从命令行进行连接	(334)
16.3.3	使用制作文件	(335)
16.4	入口和出口	(336)
16.4.1	__export 关键字	(338)
16.4.2	入口、出口与移出小结	(338)
16.5	存储模式	(339)
16.6	Windows 程序的连接	(339)
16.6.1	在集成开发环境中连接	(339)
16.6.2	用 TLINK 来连接	(339)