

目 录

第一章 Borland C++简介	(1)
1.1 什么是 Turbo C++及 Borland C++	(1)
1.2 Borland C++特色	(1)
1.2.1 Borland C++3.1 新功能	(2)
1.3 Borland C++3.0 与 Microsoft C/C++7.0 比较	(3)
1.4 安装	(6)
1.4.1 3.0 版的安装	(6)
1.4.2 3.1 版的安装	(9)
1.5 与 Borland C++第一次接触	(10)
1.6 Turbo C++主画面选项介绍	(13)
1.7 Borland C++主画面选项介绍	(20)
1.8 内存模式	(33)
第二章 C语言的基本概念	(36)
2.1 C语言基本结构	(36)
2.2 C语言的注解	(37)
2.3 C语言的变量使用	(37)
2.4 变量的基本类型	(39)
2.4.1 整数—int	(40)
2.4.2 字符—char	(40)
2.4.3 浮点数—float	(45)
2.4.4 双精度浮点数—double	(46)
2.4.5 长双精度浮点数—long double	(46)
2.5 变量的声明	(46)
2.6 变量的存贮类型	(47)
2.6.1 自动变量—auto	(47)
2.6.2 外部变量—extern	(47)
2.6.3 静态变量—static	(48)
2.6.4 寄存器变量—Register	(48)
2.7 简单的输出函数—printf()	(49)
2.8 简单的输入函数—scanf()	(51)
2.9 数字运算	(52)
2.10 位运算	(53)
2.10.1 & 运算符	(53)
2.10.2 运算符	(54)
2.10.3 ^ 运算符	(54)
2.10.4 << 运算符	(54)

2.11 预处理程序	(55)
2.11.1 #define	(55)
2.11.2 #include	(56)
2.11.3 #if... #endif	(57)
2.12 命令行参数	(58)
2.13 自定函数的调用	(59)
2.14 变量初值设置及变量类型的转换	(60)
2.15 问题	(60)
第三章 循环与条件控制	(63)
3.1 条件之间关系运算符、逻辑运算符	(64)
3.2 条件语句—if	(65)
3.3 循环语句—for	(68)
3.4 循环语句—while	(70)
3.5 循环语句—do... while	(70)
3.6 多重条件语句—switch	(71)
3.7 break、continue、goto 语句	(74)
3.7.1 goto 语句	(74)
3.8 特殊条件表达式(e1? e2:e3)	(76)
3.9 问题	(77)
第四章 数组、字符串、指针	(78)
4.1 数组	(78)
4.2 字符串	(80)
4.3 指针	(82)
4.3.1 指针与数组	(85)
4.3.2 指针数组	(85)
4.4 问题	(86)
第五章 函 数	(88)
5.1 数学函数	(88)
5.1.1 三角函数	(88)
5.1.2 数值转换数	(90)
5.1.3 数值计算函数	(98)
5.1.4 随机数函数	(108)
5.2 字符函数	(113)
5.3 字符串函数	(114)
5.3.1 strcpy()函数 strcpy()函数	(115)
5.3.2 strcat()函数	(116)
5.3.3 strchr(), strrchr()函数	(116)
5.3.4 strcmp(), strcmpi(), stricmp(), strncmp(), strncmpi(), strnicmp(), Strcoll()函数	(117)

5.3.5	strlen()函数——字符串长度	(119)
5.3.6	strset(),strnset()函数——字符填入字符串	(110)
5.3.7	strlwr(),strupr()—大小写转换	(120)
5.3.8	strtod(),_strtold(),strtol(),strtoul()函数——字符串转换数值	(121)
5.4	时间函数	(123)
5.4.1	读取及设置系统时间	(123)
5.4.2	读取及设置文件时间及日期	(126)
5.4.3	时间,日期的转换	(128)
5.5	问题	(130)
第六章	用户自定数据类型	(133)
6.1	结构—struct	(133)
6.2	结构指针	(138)
6.3	联合—union	(138)
6.4	类型定义—typedef	(140)
6.5	枚举—enum	(140)
6.6	问题	(142)
第七章	文件输入与输出	(144)
7.1	文件的输入——通过缓冲区	(144)
7.2	文件的输出——通过缓冲区	(147)
7.3	文件操作函数	(148)
7.3.1	feof()函数	(148)
7.3.2	ferror()函数	(149)
7.3.3	fputs()函数	(149)
7.3.4	fgets()函数	(149)
7.3.5	fwrite()函数	(150)
7.3.6	fread()函数	(151)
7.3.7	fseek()函数	(152)
7.3.8	rewind()函数	(153)
7.3.9	remove()函数	(154)
7.3.10	rename()函数	(154)
7.3.11	fcloseall()函数	(155)
7.3.12	fgetpos()函数	(155)
7.4	低级输入与输出	(156)
7.4.1	open()函数及 close()函数	(157)
7.4.2	creat()函数	(158)
7.4.3	creatnew()函数	(158)
7.4.4	read()函数	(159)
7.4.5	write()函数	(160)
7.4.6	lseek()函数	(161)

7.4.7	tell()函数	(162)
7.4.8	filelength()	(163)
7.5	问题	(164)
第八章	C++简介一	(165)
8.1	面向对象语言	(166)
8.2	类的设计	(167)
8.3	函数原型(Function Prototype)	(170)
8.4	OOP 的封装性	(173)
8.4.1	对象数据的传递	(175)
8.4.2	友元函数	(178)
8.4.3	类与结构	(179)
8.5	变量声明位置	(180)
8.6	构造函数及析构函数	(181)
8.7	OOP 的继承	(185)
8.7.1	多重继承	(193)
8.8	问题	(197)
第九章	C++简介二	(199)
9.1	数据流—stream	(199)
9.1.1	cout 对象	(199)
9.1.2	cin 对象	(203)
9.1.3	cerr,clog 对象	(205)
9.1.4	数据流函数	(209)
9.2	新方式的文件 I/O	(212)
9.2.1	打开文件	(212)
9.2.2	文件 I/O 函数	(215)
9.3	inline 函数	(219)
9.4	this 指针	(223)
9.5	new 及 delete 运算符	(229)
9.6	函数及运算符的重载	(234)
9.7	C++提供的类	(240)
9.7.1	Stack 类	(245)
9.7.2	Queue 类	(246)
9.7.3	Deque 类	(248)
9.7.4	模板 Stack 类	(249)
第十章	Borland C++程序开发工具	(254)
10.1	使用 BCC	(254)
10.2	TLINK 连结器	(258)
10.3	IMPDEF—模块定义文件产生程序	(260)
10.4	IMPLIB—产生程序库程序	(262)

10.5	TLIB—Turbo Librarian 程序	(263)
10.6	TASM—Turbo Assembler 程序	(265)
10.7	Turbo Profiler—效率分析程序	(270)
10.8	Turbo Debugger—调试器	(275)
10.9	Fconvert—ANSIC 转换	(277)
10.10	其它公共程序	(277)
第十一章	窗口程序开发工具	(278)
11.1	RC—窗口资源编译程序	(278)
11.2	Resource Workshop—窗口资源工作台程序	(280)
11.3	HC—窗口联机帮助编译器	(290)
11.4	WINSIGHT—窗口消息跟踪程序	(293)
附录 A	习题解答	(301)

第一章 Borland C++简介

C 语言可说是近年最受欢迎的计算机语言,因为它适用的范围甚广,从操作系统(O.S.),至系统程序及应用程序无所不包,加上其移植能力(Portable)为所有计算机语言中最高的。而计算机界近来大力推广开放系统(Open System)及 Unix, Unix 又与 C 有相当深厚的关系,因此更将 C 推向另一高峰。

当前在市面上可以找到的 C 编译器甚多,其中以 Borland C 及 Microsoft C 最受欢迎,而这两家也相互较劲,互有千秋,本书所要介绍的就是宝蓝公司于 1992 年出版的 Borland C++ 3.0 及 3.1。该产品承袭宝蓝 Turbo C 的一贯风格,因此如果读者以前使用过 Turbo C 的其他版本,相信一定可以立即进入 C 的奇幻世界,如您尚未使用过 Turbo C,本书将带领您一探胜境。

Borland C++ 3.0 可以在 DOS 的环境下运行,更可以在 Windows 下运行,在 Windows 之下的 Borland C 工作环境更亲切,将使您开发程序更得心应手。

1.1 什么是 Turbo C++ 及 Borland C++

在 Borland 3.0 的产品之中,Turbo C++ 是在 Windows 环境下的软件,在其 IDE (Integrated Development Environment)之下,可以编辑、编译、连接程序并运行设计的程序。而 Borland C++IDE 则是在 DOS 环境下的软件,其操作方式与 Turbo C++ 相类似,该软件还包含 Debug 的功能使用,相当方便,以下两者之比较:

	Turbo C++ IDE	Borland C++IDE
1. 环境	Windows	DOS 或 Windows
2. 功能		比 Turbo C++ 多了 Debug 功能
3. 设计 Windows 程序	可	可
4. 设计 DOS 程序	否	可
5. 运行程序	TCW.EXE	BC.EXE

事实上这两个 IDE 十分类似。作者建议如果开发 Windows 程序,可利用 Turbo C++ IDE 比较方便,因为它本身就是一个 Windows 程序,可以在 Windows 下运行设计出来的 Windows 程序。如果要设计 DOS 程序,则使用 Borland C++IDE。

在 Borland C++ 3.1 版之后 Turbo C++ IDE 已改为 Borland C++ for Windows,运行文件由 TCW.EXE 改为 BCW.EXE。

1.2 Borland C++特色

1. Borland C++ 3.0 及 3.1 可供程序设计人员开发 C++ 及 C 程序。其中的 C++ 是

依照 AT&T2.1 标准设计的,而 C 则是 ANSI C。因此让您开发的程序兼容性高,移植性(Portable)也更强。

2. 全局的优化(Global Optimization):为程序编译出最优的目的码,使执行时间减少,使目的码变小。

3. 缩短编译时间:由于编译器及工具的使用,大大减少了编译时间。

4. 集成开发环境:(即 IDE,Integrated Development Environment),本产品提供一完整的工作平台(Platform),使程序的编写编译、调试、分析、运行相当完整且方便。

5. 窗口程序开发:本产品可以开发出 Microsoft Windows 程序,还包括完整的工具程序:窗口资源编译器(Resource Compiler),在线帮助编译器(Help Compiler)及窗口资源工作程序(Resource Workshop)。

6. DPMI 编译器:DPMI(DOS Protected Mode Interface)使编译器可以在 DOS 或 Windows386 的保护模式下运行,因此开发大型的程序不用担心内存不足。

7. Easywin:或将 DOS 程序直接转成 Windows 程序的功能,不需要改半行指令。

8. WinSight:窗口信息跟踪程序,使我们看到程序与 Windows 之间的联系信息。

9. 在线帮助,在 IDE 环境下,任何错误都可由在线帮助得到说明或者想要任何时间查询函数或其他说明,都可使用此项功能。

10. 虚拟 Run-time 面向对象存储管理功能(VROOMM,Virtual Run-time Object-Oriented Memory Manager)。当程序设计人员选定目的码的覆盖功能后,VROOMM 会处理一切事情,即使程序超过 640K。

11. 丰富的类库,本产品提供常用的数据结构,例如 List、Queue、Qegue、Stack、Array...等(参考本书第九章)。

12. 丰富的函数库:本产品提供 4、5 百个函数,可以说无所不包了。

1.2.1 Borland C++3.1 新功能

1. 3.1 版将 Turbo C++ for Windows 改名为 Borland C++ for Windows。

2. 3.1 版配合 Windows 3.1 版,增加多媒体(Multi-Media)、笔(Pen)和 MCI(Media Control Interface)之控制接口。

3. 新增 Win Spector 工具程序,此工具程序是当 Windows 程序发生 General Protection Faults 时,提供发生此错误的地方,及当时机器的状态。

4. 在 Borland C++ for Windows 提供最优化的选项。

5. 彩色语法高亮显示(Color Syntax Highlighting),在源程序中设置不同的显示的颜色。可在 DOS IDE 内的 Options|Environment|Colors|Edit 及 Windows IDE 内的 Options|Environment|Highlight 内设置。

6. 可利用参数(-3)或 IDC 内的 Options |Compiler|Advanced Coded Generation |80386 设置产生 386 的目的码。

7. 可在程序内控制最大打开文件数,在 _NFILE.H 内 #DEFINE _NFILE_ n 改变 n 的数值,要注意的是 n 必须小于等于 Config.sys 内的 files 数。

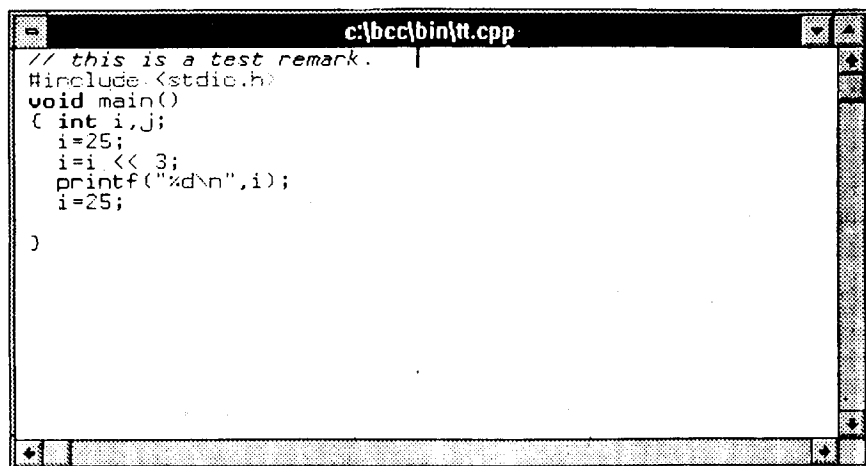


图 1-1

可用以下三种方法达到以上目的。

- 将 `_NFILE.H` 包含在程序内,并修改 `_NFILE_` 之值。
- 在 `LIB` 目录下找到 `FILES.C` 及 `FILES2.C` 编译之

`BCC -C-m<model> files.c files2.c。`

使用时直接将 `files.obj`, `files2.obj` 连接到程序内

`BCC test.cpp files.obj files2.obj。`

注: `model` 为内存模式。

- 如果把全部的打开文件数都改变,则用下列方法先编译 `FILES.C` 及 `FILSE2.C`。

`BCC -C-m<model> files.c files2.c`

再用 `TLIB` 更新 `LIB` 内的模块

`TLIB c<model> +-files.obj +-files.2.obj`

1.3 Borland C++ 3.0 与 Microsoft C/C++ 7.0 比较

Borland C++ 3.0 与 Microsoft C/C++ 7.0 可以说是当前最受使用者欢迎的 C/C++ 编译器。这两家公司相互较劲,你来我往好不热闹。由于两家产品各有特色,因此要看用户需要来决定在采用那一种产品。

首先我们来探讨二者之编译效率。

1. 编译连接时间比较(针对 C 语言, DOS 环境)

	编译时间	运行时间	.EXE 大小
Borland C++ with A/P	76.5 秒	125 秒	135,102 bytes
Turbo C++ for DOS	66.5 秒	123 秒	151,740 bytes
Microsoft C/C++	70.5 秒	176 秒	177,953 bytes

2. 文件大小优化比较(针对 C 语言, DOS 环境)

	文件大小	MAKE 时间	运行时间
Borland C++ with A/P	128,798 bytes	101.2 秒	117 秒
Microsoft C/C++	140,465 bytes	337.6 秒	123 秒

此项 Borland C++ 显然比 Microsoft C/C++ 强, MAKE 的时间短文件较小并且运行时间也较短。

3. 运行速度优化比较(针对 C 语言, DOS 环境)

	运行速度	运行时间	.EXE 大小
Borland C++ with A/P	111 秒	129 秒	132,574 bytes
Turbo C++ for DOS	123 秒	66.2 秒	159,364 bytes
Microsoft C/C++	119 秒	588.8 秒	174,023 bytes

此项 Borland C++ 显然比 Microsoft C/C++ 强。

特别是 MAKE 时间, Microsoft C/C++ 花太多时间去优化, 但结果也不会比 Borland C++ 好。

4. 编译连接时间比较(针对 C++ 语言, DOS 环境)

	编译时间	运行时间	.EXE 大小
Borland C++ with A/P	38.8 秒	69 秒	99,403 bytes
Turbo C++ for DOS	42.3 秒	72 秒	106,289 bytes
Microsoft C/C++	42 秒	50 秒	67.808 bytes

5. 文件大小优化比较(针对 C++ 语言, DOS 环境)

	文件大小	MKE 时间	运行时间
Borland C++ With A/P	97,995bytes	46.2 秒	69 秒
Microsoft C/C++	70,5bytes	125.6 秒	41 秒

Microsoft C/C++ 在文件大小, 运行时间上表现都不错, 只是 MAKE 时间较长。

6. 运行速度优化比较(针对 C++ 语言, DOS 环境)

	运行速度	运行时间	.EXE 大小
Borland C++ With A/P	67 秒	69 秒	98,859 bytes
Turbo C++ for DOS	72 秒	42.5 秒	1107,713 bytes
Microsoft C/C++	37 秒	195.3 秒	73,848 bytes

Microsoft C/C++ 在运行速度优化方面表现不错, 只是 MAKE 时间几乎是 Borland C++ 的 4 倍。

7. 编译连接时间比较(针对 Window 语言, Windows 环境)

	编译时间	.EXE 大小
Borland C++ With A/P	69.9 秒	111,104 bytes
Turbo C++ for DOS	79.2 秒	121,155 bytes
Microsoft C/C++	74.9 秒	90,320 bytes

Windows 程式方面, Borland C++ 及 Microsoft C/C++ 在编辑译时间及执行文件大小各有所长。

8. 文件大小优化比较(针对 Window 语言, Windows 环境)

	文件大小	MAKE 时间
Borland C++ With A/P	107,520 bytes	89.7 秒
Microsoft C/C++	79,952 bytes	91.4 秒

Microsoft C/C++ 可以产生较小的文件。

9. 运行速度优化比较(针对 Window 语言, Windows 环境)

	MAKE 时间	.EXE 大小
Borland C++ With A/P	120.7 秒	110,080 bytes
Turbo C++ for DOS	76.1 秒	122,179 bytes
Microsoft C/C++	96.7 秒	80,784 bytes

Microsoft C/C++ 表现较 Borland C++ 佳。

注:

1. 以上数据来自 PC MAGAZINE。
2. 由于 Windows 程序都是交互式程序, 因此不做运行速度的测试。
3. Borland C++ 在 C++ 方面表现不佳, 原因可能是内存技术上, 每次释放 (Deallocate) 内存时, 会调用 DOS, 要求 DOS 释放内存, 如此可以使编译连接不致于产生内存不足的情况。

4. Borland C++ 3.1 推出后, 有些弱点, 相信已做改善。

综合以上这类比较, 可以得到以下一些结论:

1. Borland C++ 在 C 语言方面表现较佳。
2. Microsoft C/C++ 在 C++ 语言方面表现较佳, 但, MAKE 时间仍在加强。
3. Borland C++ 在编译方面比 Microsoft C/C++ 表现较佳。
4. Borland C++ 在 C 语言方面, 运行速度较快, C++ 语言方面运行速度平平。
5. Microsoft C/C++ 在 C 语言方面, 运行速度平平, 而在 C++ 语言方面则表现不错。

除了以上对性能的比较外, PC MAGAZINE LAB 对这两项产品评分如下:

	Borland C++ with A/P	Microsoft C/C++
小型 DOS 程序	优	好
大型 DOS 程序	优	好
Windows 程序	优	优
C++开发	优	好
移植性(Portability)	平	平

由上表看来,Borland C++应该比 Microsoft C/C++较胜一筹,但相信日后这两个产品还会精益求精更上一层楼,这正是我们大家乐意看到的。

1.4 安装

1.4.1 3.0 版的安装

Borland C++ 安装的硬件设备如下:

1. 386SX 以上主机
2. 软盘驱动器(安装时使用)
3. 硬盘(如安装全部选项则需 29M,安装时需要额外的 5M)
4. 内存 2M 以上

运行时软件设备

Windows 3.0 以上(中英文版本都可,BC.EXE 如直接在 DOS 下运行亦可,但建议宜用 WINDOWS 来运行使用上较方便。)

读者的原版软盘(3.0 版)应该有九张磁盘及九本手册,九本手册分别如下:

1. Borland C++ 使用手册
2. Borland C++ Tools and Utilities Guide
3. Borland C++ Programmer's Guide
4. Borland C++ Library Reference
5. Resource Workshop User's Guide
6. Turbo Debugger User's Guide
7. Turbo Profiler User's Guide
8. Turbo Assembler User's Guide
9. Turbo Assembler Quick Reference

安装步骤

将第一片软盘放入 A 盘

A:>INSTALL

出现第一个 Welcome 画面,按下 **Return** 继续:

Welcome to the Borland C++ installation Program.

:

Press ENTER to continue,ESC to quit.

第二个画面为安装的选项:

Directories

Install Options

DOS Library Modeles

Install Examples

Windows Options

A. Directories 为 Borland C++ 的主目录其初值为 C:\Borland C

B. Install Options 有

- | | |
|----------|----------------------------|
| 1. CMD | 占 2287K。内容为直接运行软件 |
| 2. IDE | 占 1500K,内容为集成开发环境软件 |
| 3. TD | 占 409K,内容为 Turbo Debugger |
| 4. TASM | 占 690K,内容为 Turbo Assembler |
| 5. TPROF | 占 1122K,内容为 Turbo Profiler |
| 6. CLASS | 占 530K,内容为类库 |
| 7. BGI | 占 215K,内容为绘图接口软件 |
| 8. DOC | 占 331K,内容为相关说明文件 |

详细内容请看本章的其他章节。

C. DOS Library Models:

PC 程序由于内存控制程度不同,因此分为 Tiny,Small,Compact ,Medium,Large,Huge 六种模式,因此安装时可以选定那些程序库(Library)安装。

- | | |
|------------|--------|
| 1. Small | 占 308K |
| 2. Medium | 占 307K |
| 3. Compact | 占 315K |
| 4. Large | 占 320K |
| 5. Huge | 占 316K |

注: Tiny 与 Small 用的程序库相同。

D. Install Examples:

有那些 Boroand C++ 的范例要安装到硬盘内。

- | | |
|----------|--------------------------|
| 1. C/C++ | 占 155K |
| 2. TD | 占 183K(Turbo Debugger) |
| 3. TASM | 占 635K (Turbo Assembler) |
| 4. TPROF | 占 131K(Turbo Profiler) |

E. Windows Options:

窗口选项如下

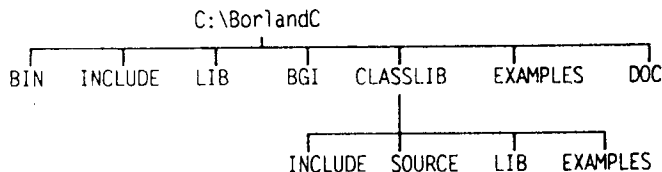
1. WIN 占 6150K
2. WEXM 占 466K (Windows 范例)
3. RW 占 2550K (Resource Workshop, 窗口资源工作软件)
4. RWEXM 占 224K (Resource Workshop 范例)

选好以上选项后,将光标移到

Start Installation

按下 **return**, 便开始安装。此刻依计算机指示将软盘一一插入磁盘驱动器内, 便完成安装动作。

以下是全部安装选项的目录结构:



1. BIN: 存放主程序
2. Include: 为存放表头文件 (heading file)
3. LIB: 存放库数据
4. BGI: 存放绘图接口数据程序
5. Classlib: 存放 Borland C++ 所提供的类库
 - a. include: 存放类库头文件
 - b. source: 存放类库源程序
 - c. lib: 存放类库数据
 - d. examples: 存放类使用范例
6. EXAMPLES: 存放 C++ 程序范例 (以下尚有一些子目录)
7. DOS: 存放 Borland C++ 相关数据文件

安装完成后, 运行 Windows 时, 会发现在主画面, 增加了 Borland C++ 这个选项, 该选项下有以下程序:

1. Borland C++: 传统 C/C++ IDE
2. Turbo Profiler: 程序运行效率分析工具程序
3. Turbo Debugger for Windows: 调试工具程序
4. Turbo C++ for Windows: Windows 下 Turbo C/C++ IDE
5. Resource Workshop: 窗口资源工作程序
6. Win Sight: 窗口信息跟踪程序
7. Import Library: 建立程序库的工具程序
8. Fconvert Utility: C 语言转换成 ANSI C 的工具程序

如读者使用 Windows, 则建议用 Turbo C++, 因为它是一个真正 Windows 程序, 当然如较习惯 Borland C++ 的运作方式即就选用 Borland C++ 这项。

1.4.2 3.1 版的安装

1. 386SX 以上主机
2. 软盘驱动器(安装时使用)
3. 硬盘驱动器(如安装全部选项则 49M, 安装额外的 5M)
4. 内存 2M 以上

运行时软件设备

1. Windows 3.1(中英文版本都可, BC.EXE 如直接在 DOS 下运行亦可, 但建议使用 WINDOWS 来运行, 使用上会较方便。)

读者的原版软盘(3.1 版)应该有十五张磁盘)。

安装步骤

将第一片磁放入 A 盘

A: > INSTALL

出现第一个 Welcome 画面, 按下 **Return** 继续;

```
Welcome to the Borland C++ 3.1 & Application Frameworks
installation program
:
PRESS ENTER to continue. ESC to quit.
```

第二个画面为安装选项:

```
Directories
Install Options
Examples Options
Windows Options
DOS Library Modeles
```

A. Directories 为 Borland C++ 的主目录其初值为 C:\Borland C

B. Install Options 有

- | | |
|----------|-----------------------------|
| 1. CMD | 占 2287K, 内容为直接运行软件 |
| 2. IDE | 占 1500K, 内容为集成开发环境软件 |
| 3. TD | 占 2409K, 内容为 Turbo Debugger |
| 4. TASM | 占 720K, 内容为 Turbo Assembler |
| 5. PROF | 占 1122K, 内容为 Turbo Profiler |
| 6. CLASS | 占 1012K, 内容为类库 |
| 7. BGI | 占 225K, 内容为绘图接口软件 |
| 8. DOC | 占 860K, 内容为相关说明文件 |
| 9. TV | 占 1500K 内容为 Turbo Vision |

详细内容请看本章的其他章节。

C. DOS Library Models:

PC 程序由于内存控制不同,因此分为 Tiny, Small, Compact, Medium, Large, Huge 六种,因此安装时可以选定那些程序库(Library)安装,每种模式约 300K 左右。

D. Examples Options:

有那些 Borland C++ 的范例要安装到硬盘内。

1. C/C++ DOS 占 235K
2. TD 占 240K (Turbo Debugger)
3. TASM 占 635K (Turbo Assembler)
4. TPROF 占 210K (Turbo Profiler)
5. RW 占 300 (Resource Workshop)
6. OWL 占 2871K (Object Windows files)
7. C/C++ (Windows) 占 1250K
8. TV 占 466K (Turbo Vision)

E. Windows Options:

窗口选项如下

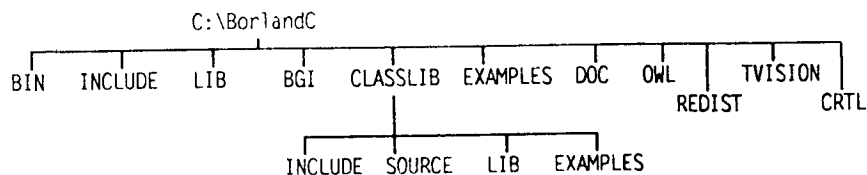
1. Windows Cap. 占 1.5M
2. Windows3.1 占 2.1M
3. RW 占 2.5M
4. OWL 占 5.5M

选好以上选项后,将光标移到

Start Installation

按下 `return`, 便开始安装。此刻依计算机指示将软盘一一插入磁盘内, 便完成安装动作。

以下是全部安装选项的目录结构:



1.5 与 Borland C++ 第一次接触

相信大家已迫不及待的想试一试 Borland C++。我们就利用 Borland C++ 来写一个小程序, 试编译连接、运行。以下是这个“Hello.cpp”程序:

```

#include <stdio.h>
void main()
{
    printf("Hello, word!");
}

```

我们可以用三种方法来达到运行此程序的目的

1. 利用字处理程序(如 PE2)编写程序,再运行

C:\Borlandc\BIN>BCC Hello

BCC 将编译连接 Hello.cpp 程序等执行 BCC 结束后,就产生 Hello.exe 了。各位可执行 Hello.exe 看看,这个程序将在屏幕上显示“Hello, World!”。

2. 利用 Borland C++

C:\Borlandc\BIN>BC

执行 BC 后你会看到以下画面:

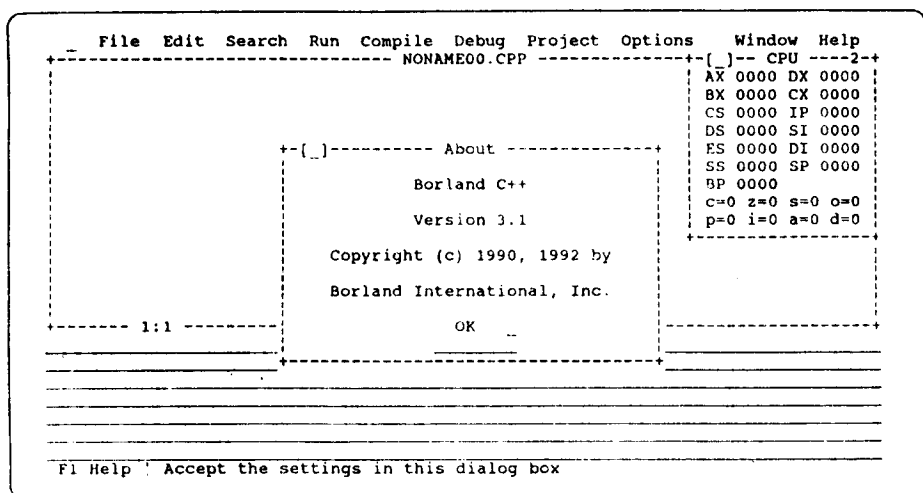


图 1-2

选定“File”后,再选定“New”则出现另一窗口,我们开始写 Hello.cpp 程序,编写完毕,则再选“File”的“Save as”输入文件名 Hello(其扩展文件名 cpp、Turbo C++ 会自动补上)。

再选“Run”的“Run”开始进行编译及连接的动作,完成以上动作,便会开始执行 Hello.exe。

以上动作整理如下:

- ①File|New
- ②程序编写
- ③File|Save as→输入 Hello
- ④Run|Run

3. 利用 Turbo C++

进入 Windows(在此已假设读者了解 Windows 的操作方式,如读者未使用过 Windows 则建议利用 Borland C++(即 BC.EXE)。或到书店买本 Windows 的书籍,相信会喜欢 Windows 的)。将鼠标指到 Borland C++ 按下鼠标左按键,出现 Borland C++ 主画面。Borland C++ 3.0 主画面:

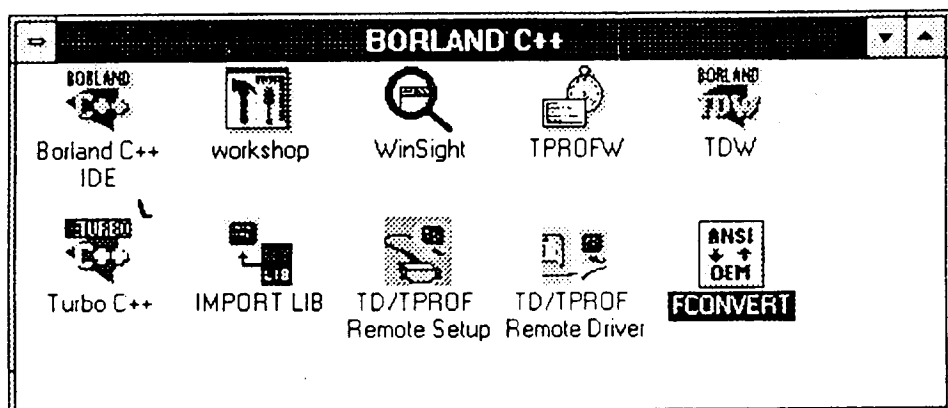


图 1-3

Borland C++ 3.1 主画面

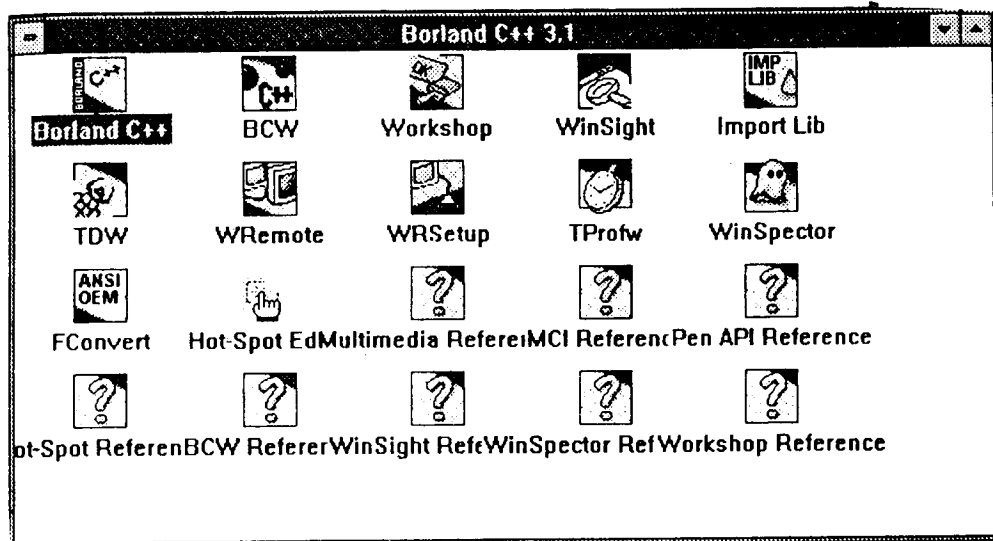


图 1-4

将鼠标指到 TURBO C++处按二下左按键,运行 Turbo C++,以下是其画面。