

计算机 C/C++ 语言系列丛书

附范例磁盘

语言双雄

BORLAND C/C++

蔡明志 编著

学苑出版社



计算机 C/C++ 语言系列丛书

语言双雄

Borland C/C++

		蔡明志			编著
知	寒	杨	理	韩	健
夏	明	林	江	李小军	审校

学苑出版社

1994

(京)新登字 151 号

内 容 提 要

本书以大量实例详细讲解了 Borland C/C++ 程序设计的各个方面。主要内容包括:操作环境、数据类型、表达式、语句、文件、输入/输出函数、图形、面向对象的程序设计(OPP)及应用实例。其主要特点是由浅入深,通过实例讲解某一题目,易于读者掌握 Borland C/C++ 的实质,并编写出高质量的程序。

本书不仅适用于高等院校有关专业的师生,也适用于自学者和技术人员。

需要本书的用户,请直接与北京 8721 信箱联系,邮政编码 100080,电话 2562329。

版 权 声 明

本书繁体字中文版原书名为《语言双雄 Borland's C/C++ 》,由松岗电脑图书资料股份有限公司出版,版权归松岗公司所有。本书简体字中文版版权由松岗公司授予北京希望电脑公司和学苑出版社出版、发行。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

计算机 C/C++ 语言系列丛书

语言双雄 Borland C/C++

编 著:蔡明志

改 编:知 寒 杨 理 韩 健

审 校:夏 明 林 江 李小军

责任编辑:徐建军

出版发行:学苑出版社 邮政编码:100032

社 址:北京市西城区成方街 33 号

印 刷:双青印刷厂印刷

开 本:787×1092 1/16

印 张:27.125 字数:628 千字

印 数:1~5000

版 次:1994 年 7 月第 1 版第 1 次

ISBN7-5077-0875-6/TP. 24

本册定价:56.00 元(含盘)

学苑图书厂、装错误可随时退换

序 言

如日中天的C语言以及体现未来程序设计新概念的C++语言,笔者将其命名为“语言双雄”,它们是目前最热门的两种程序语言。而笔者以Borland公司所出版的Borland C++(或Turbo C++)来描述这两种语言,首先是因为它的使用者较多;其次是因为它的用户界面易于操作。

本书先由C语言谈起,一步一步的导引读者进入C语言的领域,笔者尽量将基本概念与程序实例结合加以说明,以使读者可以轻松的了解C语言的精髓,从而进入C++的领域。由于C++是C语言加入面向对象的概念而成的,因此本书最后谈到的C++皆以比较的方式来原因,尤其是第十八章笔者举了四个例子,都分别用C和C++编写并加以分析。当然在有限的篇幅下无法全部表现出C++的各种特性,不过您已可以了解C++的大意了。希望深入研究C++的读者请参阅笔者另一本拙著《Borland C++3.1—软件IC不是梦》它将带给您更多的C++知识。笔者才疏学浅,更期盼您的批评与指正。

蔡明志

目 录

第 0 章 Borland's C 操作环境	(1)
0.1 Borland's C 集成环境	(1)
0.2 编辑程序	(4)
第一章 程序概述	(8)
1.1 程序语言	(8)
1.2 C 程序例子	(9)
1.3 C 程序入门	(12)
1.4 进一步的例子	(14)
1.5 变量说明	(16)
1.6 关键字	(20)
1.7 小结	(21)
第二章 数据类型	(22)
2.1 位、字节和字	(22)
2.2 整数与浮点数	(23)
2.3 int 类型	(24)
2.3.1 八进制与十六进制	(24)
2.3.2 其它整数类型	(26)
2.4 char 类型	(28)
2.4.1 说明 char 变量	(29)
2.4.2 特殊字符	(29)
2.5 float 类型	(31)
2.6 溢出问题	(33)
2.7 常量	(36)
2.8 小结	(37)
第三章 格式化输入输出	(38)
3.1 转换说明	(38)
3.2 格式限定词	(41)
3.2.1 标志(Flags)字段与宽度(Width)	(42)
3.2.2 精度字段	(45)
3.3 函数 scanf()	(48)
3.4 特殊的 * 限定词	(51)
3.5 printf()与 scanf()的返回值	(53)
3.6 转换的意义	(55)

3.7 小结	(57)
第四章 表达式	(59)
4.1 基本运算符	(59)
4.2 赋值运算符: =	(59)
4.2.1 四则运算符	(60)
4.3 sizeof 运算符	(62)
4.4 取模运算符	(63)
4.5 增量与减量运算符	(65)
4.6 优先级	(68)
4.7 表达式的值	(70)
4.8 类型转换	(72)
4.9 小结	(72)
第五章 条件与转移	(74)
5.1 if 语句与关系运算符	(74)
5.2 if...else 语句	(78)
5.3 嵌套 if 语句	(80)
5.4 真值与假值	(83)
5.5 逻辑运算符	(85)
5.6 条件运算符	(88)
5.7 else if 多重选择	(90)
5.8 switch 语句	(92)
5.9 goto 语句	(95)
5.10 小结	(100)
第六章 循环	(101)
6.1 while 循环	(101)
6.2 do...while 循环	(106)
6.3 循环	(109)
6.4 逗号运算符	(113)
6.5 复合赋值运算符	(114)
6.6 嵌套式循环结构	(116)
6.7 break 语句与 continue 语句	(119)
6.7.1 break 语句	(120)
6.7.2 continue 语句	(121)
6.8 小结	(123)
第七章 函数	(124)
7.1 函数的基本概念	(124)
7.2 函数参数	(131)
7.3 具有返回值的函数	(133)

7.4 函数原型	(138)
7.5 递归函数	(140)
7.6 小结	(143)
第八章 数组	(144)
8.1 数组说明与表示法	(144)
8.2 数组的初始化方式	(148)
8.3 二维数组与多维数组	(152)
8.4 应用举例:插入排序法	(158)
8.5 应用举例:二分搜索法	(161)
8.6 小结	(164)
第九章 指针	(165)
9.1 指针的概念	(165)
9.2 指针变量	(167)
9.3 指针的用途	(169)
9.4 数组与指针	(174)
9.5 在函数间传递数组	(179)
9.6 应用举例:选择排序法	(184)
9.7 小结	(186)
第十章 字符串	(187)
10.1 字符串说明与初始化方式	(187)
10.2 字符串的长度	(190)
10.3 字符串拷贝、连结与复制	(193)
10.3.1 字符串拷贝	(193)
10.3.2 字符串连结	(194)
10.3.3 字符串复制	(196)
10.4 字符串的比较	(197)
10.5 命令行参数	(202)
10.6 小结	(205)
第十一章 结构与联合	(206)
11.1 结构的用途	(206)
11.2 结构说明与变量	(209)
11.3 存取结构成员	(212)
11.4 应用举例:链表	(216)
11.5 联合	(225)
11.6 小结	(226)
第十二章 文件	(227)
12.1 标准文件 I/O	(227)

12.2 覆盖方式与添加方式	(231)
12.3 文本文件与二进制文件	(234)
12.4 文件随机存取	(242)
12.5 小结	(246)
第十三章 其它主题	(247)
13.1 位运算	(247)
13.1.1 各种位运算符	(247)
13.1.2 位运算符的用途	(252)
13.2 变量存储类	(254)
13.2.1 存储类	(255)
13.2.2 作用域和生存期	(261)
13.3 枚举类型与位域	(262)
13.3.1 枚举类型	(262)
13.3.2 位域	(263)
13.4 预处理程序	(265)
13.4.1 #define 伪指令	(265)
13.4.2 #include 伪指令	(267)
13.4.3 typedef 设施	(268)
13.5 小结	(269)
第十四章 文本方式处理函数	(270)
14.1 文本窗口	(270)
14.2 光标控制	(273)
14.3 文本显示属性	(276)
14.4 文本块处理	(278)
14.5 存取窗口信息	(283)
14.6 字符编辑函数	(287)
14.7 小结	(287)
第十五章 图形函数	(289)
15.1 进入图形系统	(289)
15.2 基本图形函数	(293)
15.2.1 直线	(294)
15.2.2 矩形	(296)
15.2.3 圆形	(297)
15.2.4 弧	(299)
15.2.5 椭圆	(300)
15.3 线条类型与写入方式	(302)
15.4 视窗	(306)
15.5 填图类型与颜色	(308)

15.6 长条图	(313)
15.7 图像存储与恢复	(317)
15.8 图形字型	(323)
15.9 小结	(328)
第十六章 由 C 到 C++	(329)
16.1 新注解格式	(329)
16.2 可在程序任何位置说明变量	(329)
16.3 函数的重载(overloading)功能	(331)
16.4 内嵌(inline)函数	(333)
16.5 作用域分辨运算符::	(336)
16.6 引用类型(Reference type)	(337)
16.7 具有缺省值的函数参数	(339)
16.8 new 及 delete 运算符	(342)
16.9 C++ 的流	(345)
16.9.1 标准 I/O	(345)
16.9.2 格式化的输出	(346)
第十七章 OOP 特征	(349)
17.1 类定义	(349)
17.2 重载运算符(overloading operators)	(354)
17.3 类的继承(inheritance)	(355)
17.4 多态性(polymorphism)及虚函数(virtual function)	(357)
17.5 多重继承(multiple inheritance)	(360)
第十八章 应用问题探讨	(362)
18.1 例子一	(362)
18.2 例子二	(372)
18.3 例子三	(380)
18.4 例子四	(394)
附录 A C 语言运算符	(413)
附录 B 表达式、语句、以及程序流程	(417)
附录 C Borland's C 文件	(420)

第零章 Borland's C 操作环境

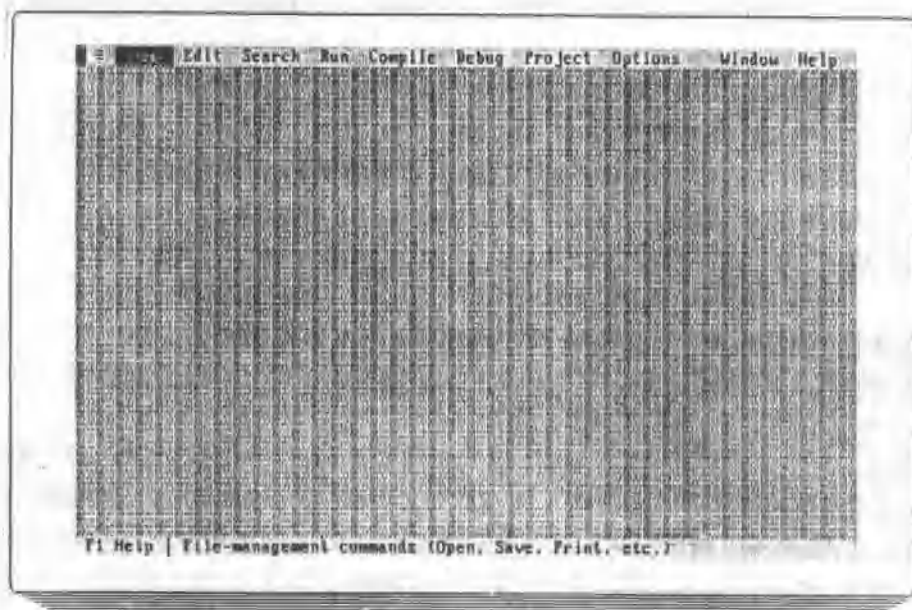
Borland's C 是指在 Borland C++ 或 Turbo C++ compiler 下所编写的 C 程序语言,亦即把 C 当作 Borland C++ 或 Turbo C++ 来执行。Borland's C 不仅包括 C 语言本身,事实上,它是一个集成了编辑程序(editor)、编译程序(compiler)、调试程序(debug)、以及帮助系统(Help)等等部件的软件包(package)。

C 语言是一种编译型的语言,源程序必须先经过编译步骤形成.obj 文件,再与其它.obj 模块连接成为可执行的文件(executable file);这个过程可能需要用到编辑程序、编译程序、以及连接程序等等,对于 C 语言的初学者而言,这些动作显得十分繁琐;幸好 Borland's C 提供了一个集成开发环境(Integrated Development Environment,简称 IDE),综合了上述的各种功能,在这个软件环境下,程序的开发、调试,甚至执行,操作都极其简单。

本书首先为您简要介绍 IDE 的重要功能,以便利用 IDE 来学习 C 语言。

0.1 Borland's C 集成环境

因为本书的 Borland's C 以 Borland C++ 为主,因此在 DOS 提示符号后面键入 bc 命令后,就可以进入 IDE 环境,这时候屏幕上将出现以下的画面:



IDE 的显示包括顶端的菜单(menu),以及底下的一个大窗口(window);当您选择 Window 下的 about 选项时,画面中央会出现 Borland C++ 的版本号码及版权声明,目前的最新版本为 Borland C++ 3.1。

屏幕顶端的菜单包括 File, Edit, Search, Run, ... 等等项目,而主窗口则分为两个子窗口,位于上方大半部的是编辑(Edit)窗口,下方则为信息(Message)窗口。我们所设计的 C 源程序将出现于编辑窗口中,而 Borland's C 系统则把程序开发过程中产生的相关信息放入信息窗口内。

在 IDE 环境下,可利用许多方式来控制菜单的操作。当您进入 IDE 后,菜单将激活(利用左右光标移动键即可选取特定的选项);如果没有被激活,那就需按下 F10 功能键,这时候菜单上的某个选项会呈现高亮度或反白显示。若想查看菜单的内容,可以利用 Enter 或光标下移键,拉出子菜单,这些子菜单内的选项仍然可以利用光标移动键加以选取,或许子菜单底下又有其它菜单。如果要关闭某项菜单而回到主画面,那么直接按 Esc 键就行了。

您可以利用各种光标移动键大致看看各个菜单的内容,只要不按 Enter,大概都不会有什么动作发生。为了不让您产生困扰,我们在此仅介绍几个重要的功能选择,其它选项的功能暂时不会用到。

设计程序之初,先要在头脑中有个概念,然后才着手把这些概念以程序语言表达出来;在 Borland's C 之下,您的程序语言当然要遵循 C 语言的标准。首先是编写源程序文件,就是我们一般所谓的程序(program),C 程序要求源程序文件的文件名一定要以 .C 结尾,假设我们想建立一个叫做 hello.c 的程序,您可以直接在 DOS 提示符号后面键入

```
D:\Borlandc\prog >bc hello.c
```

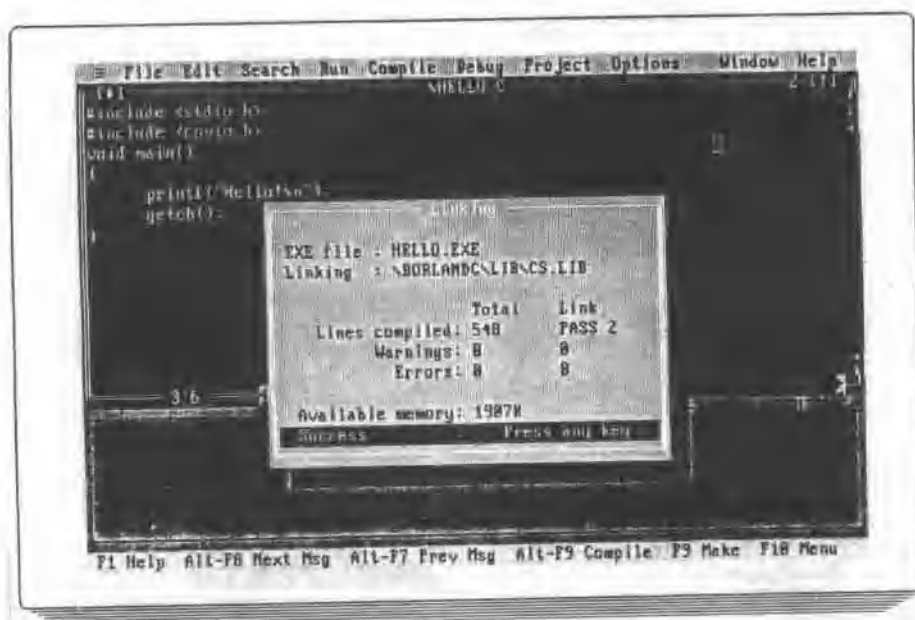
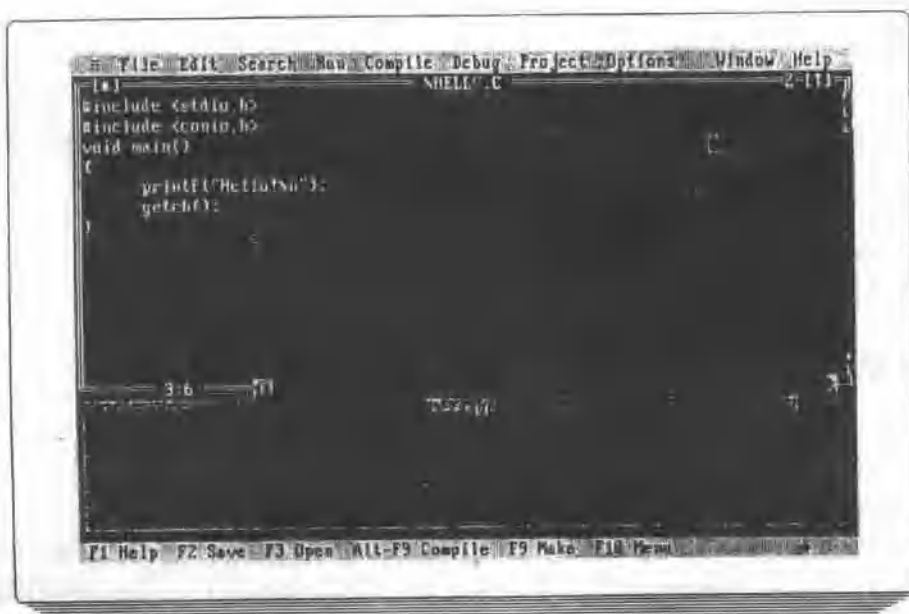
由于文件 hello.c 并不存在,所以 IDE 的编辑窗口内空无一物;您必须利用键盘在编辑窗口内打入源程序的内容(有关编辑命令的相关细节将在下一节说明)。您可以试着输入下列的程序:

```
#include<stdio.h>
#include<conio.h>
void main(){
    printf("Hello! \n");
    getch();
}
```

先别急着知道程序为何写成这样。目前的屏幕应该是如下图所示。

源程序输入之后,别忘了把它存入磁盘,您可以直接按下 F2 功能键,成是在 File 菜单下选取 Save 选项,这样 C 源程序文件就建好了。

程序输入的工作若是没有任何错误,就可以将它编译或连接(compile& link);在 IDE 环境中,编译与连接的动作可由按下 F9 一个键面同时完成。经过这一步骤后,源程序文件 hello.c 会先编译成 hello.obj,然后再与其它文件连接成为 hello.exe;到此为止,hello.exe 就是一个可执行的文件。整个过程的处理动作十分复杂,但对使用看而言,只要轻轻地按下一个功能键就行了。



有了可执行文件后，您还可以直接在 IDE 环境下执行它，按下 Ctrl-F9 或是选取 Run 菜单中的 Run 选项都可以达到这个目的。程序执行的结果将一闪而逝，您可以利用 Alt-F5 切换到输出画面，这时您可以发现以下的输出

Hello!

其实 Alt-F5 不过是把 DOS 画面重新显现出来罢了，随意按下任何一键，就可以再回到

IDE 环境。

若想结束 IDE，只要按下 Alt-X 或是前往 File 菜单下选择 Quit，就能回到 DOS 环境；这时可以检查看看是否存在 hello.c, hello.obj 以及 hello.exe 等文件。在 DOS 提示符号后面，您也可以直接执行 hello 程序，在这里假设本书的所有的例子及其编译后所产生的 .obj 及 .exe 皆存放在 D:\BORLANDC\PROG 目录下：

```
D:\BORLANDC\PROG >hello
```

同样会产生下面的输出：

```
Hello!
```

0.2 编辑程序

本节将介绍 Borland's C 所附的编辑程序，它属于 IDE 环境的一部分，如果您已熟悉 WordStar 字处理程序，将会发现两者之间十分相似；即使从来没有任何经验，Borland's C 的编辑程序也相当容易学习。

当您进入 IDE 之后，File 菜单将呈现高亮度显示，利用右移键将光标移到 Edit 菜单再按下 Enter 后就可以进入编辑状态；这时候就能看见光标正在编辑窗口中闪动，光标所在的位置即为字符将要出现的地方，键盘的使用方式与普通的打字机相同。在任何时候，都可以直接以 Alt-E 组合进入编辑状态。

下面首先介绍一些最基本的编辑命令：

按键	动作
←	向左移动一字符
→	向右移动一字符
↑	向上移动一字符
↓	向下移动一字符
Home	移到行首
End	移到行尾
PgUp	上翻一页
PgDn	下翻一页
BackSpace	删除光标左边的字符
Del	删除光标所在的字符
Ins	插入状态 On/Off 切换
Enter	换行

正常情况下，当键盘上按下某一字符时，光标所在位置就会出现该字符，至于原来位于光标后面的字符都会向右挪动一格；如果想要覆盖原来的字符，那就该先用 Ins 将插入状态设定

为 Off,再按一次 Ins 就可恢复插入方式。

输入内容后,千万别忘了把数据存起来,前面曾经提过只要按下 F2 就行了,或者是到 File 菜单下选择 Save 命令。File 菜单中还有许多选项,可对文件或目录做处理:

功 能	
Open[F3]	打开一个原有文件
New	建立一个新文件
Save 或 [F2]	以当前文件名存储(将当前文件以原名称存储)
Save all	储存所有打开的文件
Save as. . .	用不同名称存储当前文件
Change Dir	更换到其它目录
Print	用打印机打印
DOS shell	临时回到 DOS 提示符下
Quit	回到 DOS,即结束 Borland's C

其中 DOS shell 是个十分好用的命令,它可以暂时离开 IDE 而回到 DOS,这时候就能执行各种 DOS 命令;若想重回 IDE,并不需要再打 BC,只要键入 exit 就行了:

D:\BORLANDC\BIN >exit

编辑程序不仅提供字符的插入或删除,通常也提供对整个文本块(block)的操作能力。Borland's C 的编辑程序可对任意文本块进行搬移、拷贝、以及删除等处理;也可以把整个文本块从文件中读出来或者写回去。

块操作的第一步是设定文本块的范围:首先将光标移到文本块开始处,按下 Ctrl K+B,然后再把光标移到文本块结束处,按下 Ctrl K+K;这时候,被标记的整个文本块将呈现高亮度显示,然后就可以执行一系列的动作。下表中归纳了文本块操作的各个命令:

按键	动作
CtrlK+B	标记文本块开端
CtrlK+K	标记文本块结尾
CtrlK+H	取消已选取的文本块
CtrlK+C	文本块拷贝
CtrlK+Y	删除文本块
CtrlK+V	文本块移动
CtrlK+W	将文本块写入文件
CtrlK+R	将文件读入

编辑程序应该提供的另一项功能就是特定字符串的搜寻与替换。按下 Ctrl Q+F 即可寻找长达 30 个字符的字符串, 搜寻的动作则从光标目前的位置开始。

按键	动作
CtrlQ+F	查找
CtrlQ+A	查找并替换
CtrlL	重复上一次的查找并替换

若是以 Ctrl Q+A 启动搜寻与替换的功能, 系统会要求您输入想查找以及用来替换的字符串, 每当发现符合的字符串时, 系统还会询问是否真的要替换。

我们可以使用下列的字符组合来限制搜寻的条件:

字符	动作
B	从光标位置往回找
G	搜寻整个文件
U	忽略大小写的不同
W	只找出整个单词(而非单词中的子字符串)
N	不加询问而直接替换

上述字符可以组合, 例如 GU 表示在整个文件内不区分大小写地搜寻。

编辑窗口的上方有个状态行(Status line), 包含许多重要的信息。我们已经知道 Ins 可切换插入模式及覆写模式, 在状态行上将有相应的“Insert”出现或消失。

“Indent”状态则表示自动缩排。如果出现“Indent”时, 当我们按下 Enter 键使光标移到下一行后, 光标将与上一行中第一个字符出现的地方对齐, 如果不是自动缩排, 那么光标将出现在第一列上。大多数情形下, 我们都希望采取缩排模式。

若是出现“Tab”状态, 表示当我们按下 Tab 键时, 光标会向右移到 8 的倍数列上, 如果没有“Tab”显示, 则不会有这种效果。

“Fill”状态则表示每行开头的空白必须用制表符存放, 通常 Fill 是打开的。

最后的“Unindent”则作用于回退键的动作。假设光标位于行首, 此时处于自动缩排模式而且 Unindent 是打开的, 那么当按下回退键时, 将会删除足够的空白而使本行左移一个制表(符)的位置; 而当 Unindent 不起作用时, 回退键只能删除一个空白或字符。

上述各种状态可由下列搜键切换:

按键	动作
Insert Ins	插入方式
Indent Ctrl O+I	自动缩排
Tab Ctrl O+T	自动制表

Fill	Ctrl O+F	最佳充填
Unindent	Ctrl O+U	取消缩排

除了上述命令外, Borland's C 编辑程序还提供许多常用的功能:

按键	状态
CtrlW	上滚一行
CtrlZ	下滚一行
Ctrl←	向左移动一个单词
Ctrl→	向右移动一个单词
CtrlT	删除一个单词
CtrlY	删除整行
CtrlN	插入一行
CtrlK+P	打印文件(块)
CtrlQ+L	取消前次动作

对于这么多的命令与按键组合, 您不必刻意去强记, 只要常常使用它们, 不知不觉就会深深植入脑中, 再想忘也忘不掉了。

第一章 程序概述

从现在开始,我们将要正式进入 C 语言的世界。也许您曾经使用过 BASIC 或 Pascal 之类的程序设计语言,这些经验对您学习 C 语言有着极大的帮助;假使 C 语言是您接触的第一种程序语言,那也不用担心,在没有任何包袱或成见之下,经由本书浅显的介绍,C 语言必将成为您最喜爱的工具。

1.1 程序语言

程序是个什么玩意儿?C 程序看起来又像什么样子呢?简单来说,程序(program)是一组能让计算机做出有意义动作的命令;而程序语言(programming language)则能让我们(指人类)更容易与计算机沟通。

经过几十年来信息科学的发展,数百种的程序语言纷纷设计完成,它们各有不同的目标与优点;譬如为人熟知的 BASIC 即适合于计算机的初学者;FORTRAN 对于工程或数值上的运算特别有效率;Pascal 则多用于教学;其它如 Lisp,COBOL,Assembly... 等等,都拥有其一席之地。

至于本书所探讨的 C 语言,则是一种强有力而且专业化的程序设计工具,如今已受到广大业余和专业设计者的欢迎。市面上发行的多数大型软件均由 C 语言写成,而日益重要的 UNIX 操作系统本身几乎全部是由 C 语言构成的。

C 语言是 1972 年由贝尔实验室的 Dennis Ritchie 设计的,其最主要的目标在于为专业设计人员提供一种良好的工具,从 C 语言的实践来看,它的确具有许多特色:

1. 控制结构:

C 语言的程序流程控制结构实现了计算机科学理论及实际应用上的要求;诸如结构化程序设计(Structural Programming)、自上而下设计法则(Top-down design)、以及模块化(Modular)等等,这些原则在 C 语言看来是极其自然的。

2. 效率良好:

C 是种十分简洁的语言,它能充分利用现代计算机的各方面功能,譬如近似汇编语言(assembly)的控制指令,以及灵活访问存储器(memory)的相关信息的能力等。C 语言的简洁也使得执行结果更有效。

3. 可移植性:

所谓可移植性就是指在某种系统上开发的程序仅需少量的修改,甚至原封不动就能在另外一种系统上执行。可移植性愈高将可使程序开发的成本愈低,C 语言在这个方面一向居于领先地位,由于设计得当,使 C 程序极易在各系统间移植。

也许上述优点尚且无法打动您,但是从愈来愈多使用者以 C 语言做为“母语”的趋势看来,您必定不甘心自绝于潮流之外。

当然,优缺点常是一件事情的两面,例如 C 语言的简洁常常导致程序不易理解,或是造成