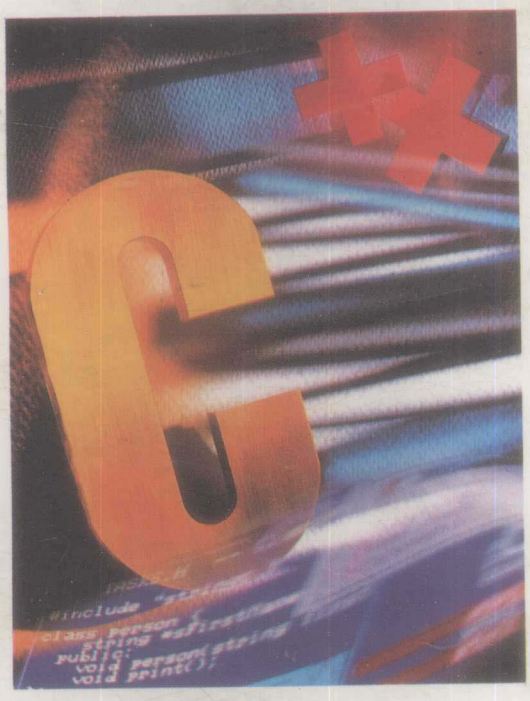
计算机程序设计语言系列丛书

BORLAND C++

程序设计应用与实践

——附盘



王行一　苏倍庆　编著

学苑出版社

计算机程序设计语言系列丛书

Borland C++程序设计应用与实践

王行一
苏倍庆 编著
天 奥 改编
熊可宜 审校

学苑出版社
1993

(京)新登字 151 号

内 容 摘 要

本书介绍目前最被广泛用来开发面向对象软件的语言(C++语言)。它具有面向对象语言的特性，如数据隐藏、函数与操作符重载、继承与动态连接（亦即多元性）等。本书摒弃一般先述说 C 语言再谈 C++的方式，直接将 C++与面向对象的关系呈现在读者眼前，使得即使不懂 C 语言的人，也能立即进入 C++的世界，这个好处使读者能从一开始便以面向对象的思想来思考问题，而不受传统设计程序观念的束缚。

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

版 权 声 明

计算机程序设计语言系列丛书

Borland C++程序设计应用与实践

编　　著：王行一　苏倍庆
改　　编：天　奥
审　　校：熊可宜
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
排　　版：北京天奥科技公司排版中心
印　　刷：双青印刷厂
开　　本：787×1092　1／16
印　　张：26.75　　**字数**：625 千字
印　　数：1—5000 册
版　　次：1993 年 12 月北京第 1 版第 1 次
ISBN 7—5077—0807—1／TP・18
本册定价：59.00 元(含盘)

学苑版图书印、装错误可随时退换

前　言

只要您看计算机杂志，就不难发现这一、二年来，几乎每个月只要看杂志的目录，就会找到一、二篇有关面向对象的文章。的确，面向对象已如一股不可阻挡的巨流，深植入软件的各个层次，由操作系统、编译器，至系统分析、系统设计，甚至于数据库管理，办公室自动化以及工厂自动化，无一不在应用这个观念，面向对象已为计算机开辟了另一个新的世界。

面向对象的观念之所以受到重视，在于它将一个问题或一项工程视为许多小对象的组合。就好比一部机器有许多的零件。零件经过大量制造后，可用于许多的机器上，因而降低成本；而对象被设计后就象零件般成为一项库存零件，有需要时可立即使用，即使规格不符，也只需做些细微的修改即可。与传统程序观念比起来，它不但节省时间，也减少开发与维护的成本。

此外，对象的设计能更贴切地描述问题，使程序更接近事实。对象的独立性使对象之间的依赖度相对减少，对程序员而言，他的可使用资源增加，而降低开发费用，因此，传统程序开发的观念，在某些领域上终将被面向对象的观念取代。

C++是目前最被广泛用来开发面向对象软件的语言。它具有完整面向对象的特性，如数据隐藏、函数与操作符重载、继承与动态连接（亦即多元性）等。本书摒弃一般先述说C语言再谈C++的方式，直接将C++与面向对象的关系呈现在读者眼前，使得即使不懂C语言的人，也能立即进入C++的世界，这个好处使读者能从一开始便以面向对象的观念来思考问题，而不受传统设计程序观念的束缚。

本书第零章说明有关Borland C++集成程序开发环境的操作方式和命令行的编译、连接和运行。若读者已学过任何Borland的产品，则此集成程序开发环境的操作可略过不看。第一章至第四章介绍C++基本语法、函数与基本输入输出。第五章介绍面向对象的观念。而由第六章起则以各别的章节详细介绍面向对象的特性在C++中表现的方式。第十二章以三个实例探讨为本书说明作总结，使读者对面向对象程序设计有个全局而清晰的概念。第十三章则列出C++函数库中较常用的函数和调用示例。各章例题皆在Borland C++上测试。

两年来看了不少高级的著作与文章，为的是吸收别人的精华，使我们能以最完整而易懂的方式将C++与面向对象程序设计的观念呈现在读者眼前，敢说自己很尽力、很认真，却不敢自夸这是一本完美的书，还希望读者多多批评与指正。

目 录

第零章　Borland C++操作与编译、运行

本章内容

0.1　引导 Borland C++

设备需求

在硬件方面，必须具有一部 AT 级或以上的 IBM PC、PS／2 或其他兼容的机器，并配有一个硬盘与一个软盘驱动器及 640K 的存储器。经过安装之后，Borland C++占有 15MB 的空间，因此必须确定硬盘的空间足够，Borland C++同时也支持鼠标(Mouse)的使用。

至于软件方面，当然先要拥有 Borland 公司所提供的 Borland C++的编译程序，它不但提供了 OOP(Object-Oriented Programming)设计环境，也包含了 ANSI C++版的特性。此外，Borland C++必须在 DOS 3.0 或更新的版本中运行。您可以直接使用 Borland C++的集成开发环境来开发程序，而不必另外准备其他的文字处理程序。

安装程序

Borland C++具有 7 张原始磁盘，利用磁盘#1 中的 INSTALL 程序可将整个 Borland C++适当地安装在硬盘中。其操作方式如下：

1. 将磁盘#1 置于 A 驱动器中。
2. 在 A＞下输入“INSTALL”,然后按 Enter。
3. 依屏幕上所提示的信息输入适当的选择，例如路径的设置、选择模块的大小、决定是否安装调试程序等等。
4. 完成上述步骤后，利用Tab键将光标移至“START INSTALLATION”处，按下 Enter 后便开始运行安装程序，此后只须依提示按顺序放入磁盘即可。

对于初学者而言，程序模块的大小设为 SMALL，应该是足够了。在此要特别提醒读者，最好将上述 7 张原始磁盘各作一分拷贝，并妥善保存，以备不时之需。

另外有两个值得一提的文件，一为 README，它提供了 Borland C++ V3.0 的一切最新最正确的信息；另一则为 Helpme!.DOC，存在 DOS 子目录下，它对 Borland C++的使用方式及指令有详细的说明，读者只要在进入 BC 后，选择 File |open 并在命令行输入

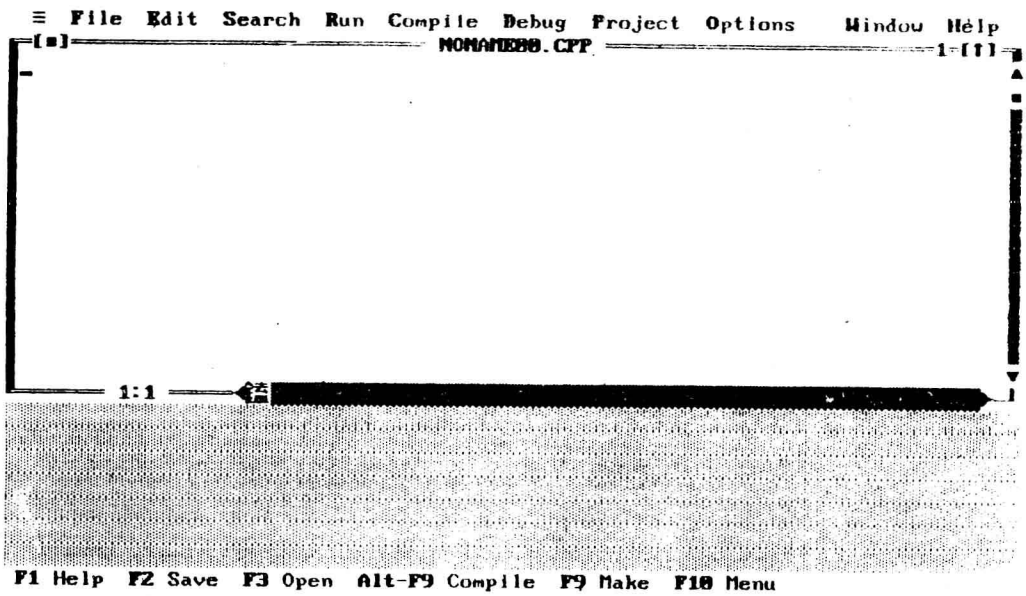
正确的文件全部路径名称（通常为 C:\BC\DOC\Helpme!.DOC）即可观看 Helpme!.DOC 的内容。File|Open 则表示进入 BC 之后选择 File 功能选项下的 Open 子功能。

今后本书将以垂直线（|）隔开功能选项的操作步骤，垂直线（|）左边的选项为其右边的选项上一层功能。由左而右依序选择功能选项，即可完成任务。

0.2 启动开发环境——BC

认识 IDE

Borland C++所提供的程序集成开发环境(Integrated Development Environment)具有下拉式菜单、对话框(Dialog Box)、多重窗口(Multiple Windows)，以及联机帮助的功能。这一节先讨论如何在新的 IDE 下开发、编译与连接程序，在下一节中再对 IDE 所提供的调试程序详加说明。



在 DOS 下输入 BC 即可进入 Borland C++的 IDE 环境，所看到的画面如图 0-1。最上方所示为主菜单，空格部分为编辑窗口。读者若同时编辑多个程序，各个程序会显示窗口号码而重叠在另一个窗口上。右上角则会显示窗口号码，编辑窗口的下方有两个数字，说明光标目前的地址。而由这两个数字起，屏幕下方被分割为另一个区域，这里有时会形成一个窗口，同样地在右下角有窗口号码，但这块区域的目的不在编辑，而有其他的用途，稍后会再加以说明。

只要按下 Alt 与窗口号码，便可发现该窗口被双线环绕，我们称其为活动窗口（Ac-

tive Window)，其余窗口则称为非活动窗口(Inactive Window)，多重窗口情形如图0-2。

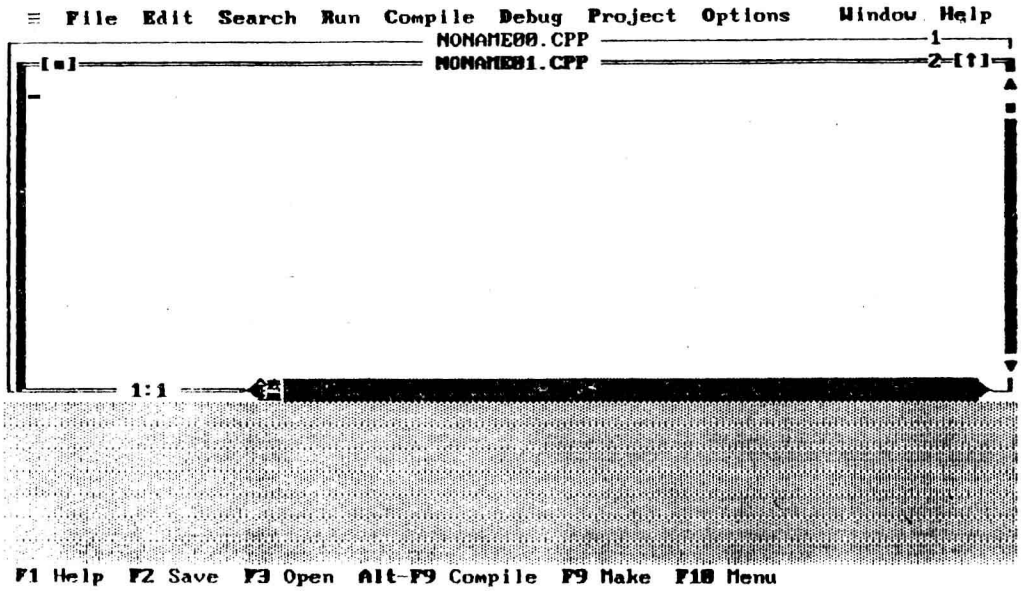


图0-2

主菜单有11项功能，分别为：System、File、Edit、Search、Run、Compile、Debug、Project、Options、Window和Help。其主要作用为：

System：取得有关Borland C++的信息。
重现屏幕
访问公用程序
跳至Turbo Assembler
跳至Turbo Debugger

File：创建或打开要编辑的源程序
存储文件
查看目录
改变目录
暂时进入DOS
退出C++

Edit：编辑本文文件
重复运行上次文件
文本数据的删除与搬移
清除窗口

Search：字符串的搜寻与取代
光标在文本中迅速移动

Run：	程序的编译、连接与运行
Compile：	编译程序
	将目标文件连接成执行文件
Debug：	程序调试，可检查变量、表达式值
	设置与清除中断点
Project：	创建计划文件
Options：	设置编译程序选项
	设置连接程序选项
	指定包含文件与程序库的地址
Window：	移动窗口与调整窗口的大小
	开闭窗口
	从一窗口跳至另一窗口
	查看窗口行
Help：	显示帮助说明
	显示帮助说明索引

可以利用 Alt＜功能项目的第一个字母＞两键的组合选取主菜单项目；而后面出现下拉菜单时，再以↑ ↓ ← →上下选取或直接按每个项目前的高量度字符来选取下拉菜单内的项目。

在 IDE 下，所有功能选择或指令下达的方式都可以利用↑ ↓ ← →键，或 Alt 组合键（例如 Alt-X 离开 IDE）或以鼠标在适当指令下击按钮确定。任何情况下若遇困难，可按下 F1，屏幕下即出现有关该指令的帮助说明，读者可由说明中找出解答。

下表归纳在 Borland C++中常用的按键：

按　键	功　能
＜F1＞	提供与文本相关的联机帮助信息
＜F2＞	存储正在编辑的文件
＜F3＞	装入指定的文件进行编辑
＜F4＞	运行程序
＜F5＞	扩大窗口或窗口恢复正常大小，当扩大窗口时该窗口会占满整个画面
＜F6＞	切换窗口（通常用来作 Edit 与 Message 窗口之间的切换）
＜F7＞	运行下一个语句，包含函数内的语句
＜F8＞	与＜F7＞同，但不会进入函数内一行一行地运行
＜F9＞	利用 make 编译和连接计划文件
＜F10＞	回到主菜单
＜Alt＞＜F5＞	作为输出屏幕与 Borland C++的对话屏幕之间的切换
＜Alt＞＜F9＞	编译目前正在编辑的程序
＜Alt＞＜B＞	跳到 Break／Watch 菜单
＜Alt＞＜C＞	跳到 Compile 菜单

(续)

按　键	功　能
<Alt><D>	跳到 Debug 菜单
<Alt><E>	跳到编辑程序
<Alt><F>	跳到 File 菜单
<Alt><O>	跳到 Option 菜单
<Alt><P>	跳到 Project 菜单
<Alt><X>	离开 Borland C++，回到 DOS
<Ctrl><F7>	增加一个要观察的表达式
<Ctrl><F8>	在光标所在处设置或清除中断点
<Ctrl><F9>	开始运行程序
<Ctrl><Break>	中断目前正运行的程序

编辑程序

设计程序的第一个步骤是将文字（程序）利用编辑程序建立文件。程序设计的过程中，主菜单下最常用到的项目有『File』、『Edit』、与『Window』。为了帮助学习的过程，我们将以实例说明如何编辑 C++程序，同时说明『File』、『Edit』与『Window』下一些子功能的意义。

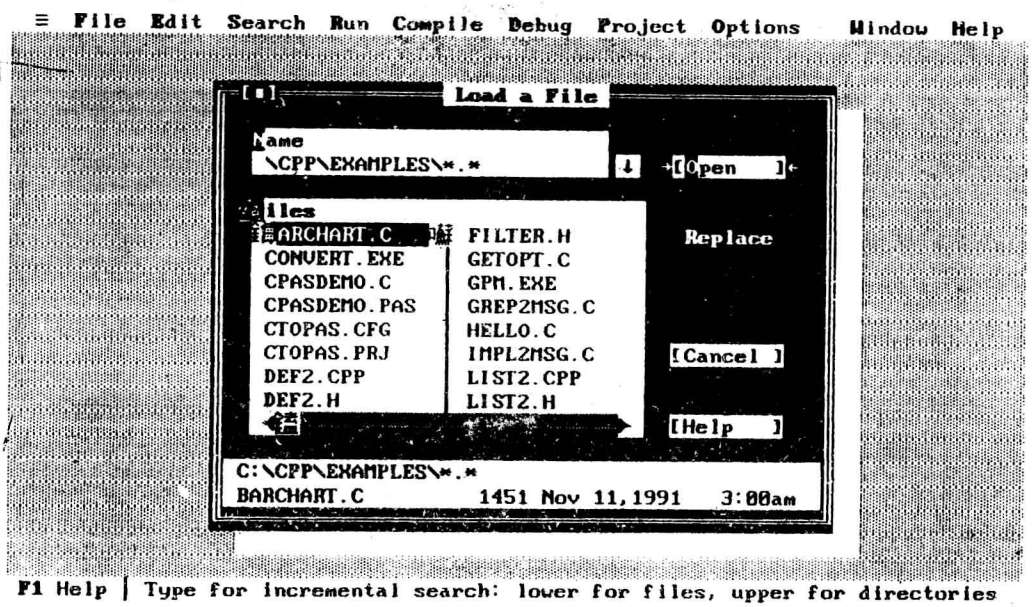


图 0-3

在系统中会发现有一个 EXAMPLES 的子目录，这里存有 Borland C++所提供的一些

程序示例。利用← → 键选择主菜单的『File』以及其子功能『Open』后，屏幕上出现如图 0-3 的画面。这便是前面曾经提到的 Dialog Box(对话框)。Dialog Box 依功能而有各种不同的情形，大致上它包括了输入区、显示区、指令选择、确定运行方式等。各区之间的切换则以 TAB 键来控制。Dialog Box 的提供，可以帮助用户正确地使用各项功能。

图 0-3 中第一个区块的上方有 NAME 这个字，它便是 OPEN 对话框的输入区。如果您目前所看到的内容不是 C:\BC\EXAMPLES*.*,请自行输入并按下 Enter（作者在安装时将目录设为 CPP，故输入 C:\CPP\EXAMPLES*.*），此时，下一个区块 FILE 的内容也会跟着改变，这便是文件显示区，显示刚才所输入的路径的内容。

我们可以在输入区中输入希望编辑的程序如：C:\CPP\EXAMPLES\LIT2.CPP;也可按 TAB 切至文件列示区，再利用↑ ↓键移动光标。本例中，在 LIST2.CPP 按下 Enter，程序内容则会出现在屏幕上，由双线环绕。在右上角出现的 1，表示窗口 1 的意思(视图 0-4)。

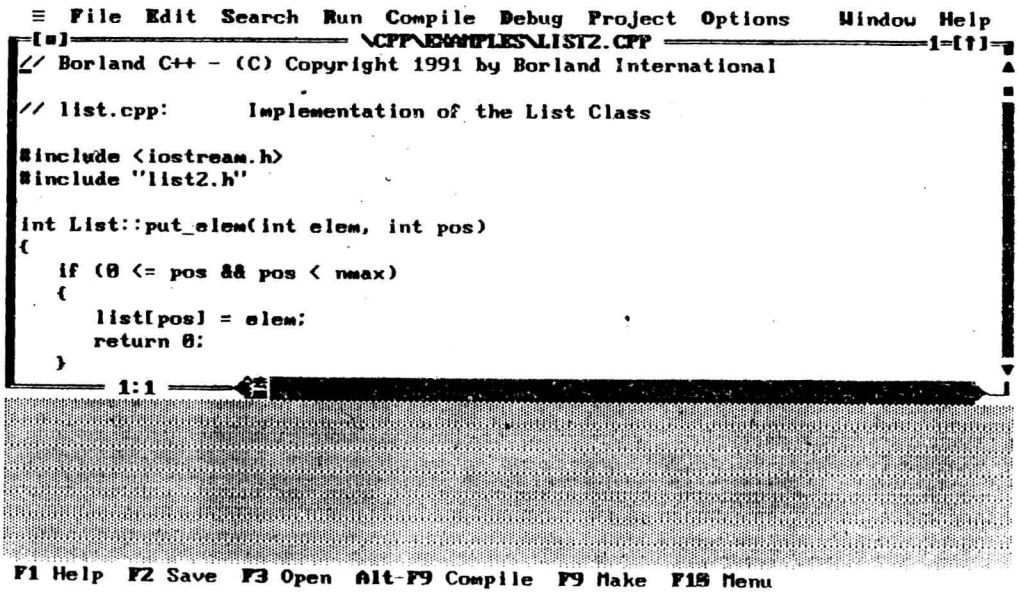


图 0-4

对于已存在系统中的程序作编辑时，利用 OPEN 是一种方式，而另一 种方式则是利用『Window 』功能下『List』子功能。LIST 对话框里显示着曾经出现在窗口内的文件，如图 0-5，我们也可利用↑ ↓键移动光标至我们想使用的文件上，按下 Enter 后便可将其调出至编辑窗口中。

至于编辑系统中完全不存在的新程序，则必须使用『File|New』功能。读者可以现在试试看，这时工作窗口变成 2，编辑区一片空白，而程序名称也被自动设置为 NONAME##.CPP。其中##表示 01 或 02,03...由系统自动给定（视图 0-6）。

在编辑新程序时，其程序码可依需要由各个已写好的程序中 COPY 过来。现在按 ALT 1 回到窗口 1，先将光标移至希望被 COPY 的块开始的地址，按住 SHIFT，配合↑

↓←→、HOME、END、PgDn、PgUp 任何一键，来括出整个块，被包含的范围会呈现反白，此时再利用『Edit|Copy』(或按 CTRL-INS)将块先暂时存入一个称为 Clipboard 的缓冲区。，现在再回到刚才我们使用到的窗口 2（试试看如何回去？），选择『Edit|Paste』或按 Shift INS，剪贴板(Clipboard)里所存的程序码就被 COPY 过来。但注意，在窗口 1 中，那块反白的块仍然留着，利用 CTRL-KH 即可令反白消失。

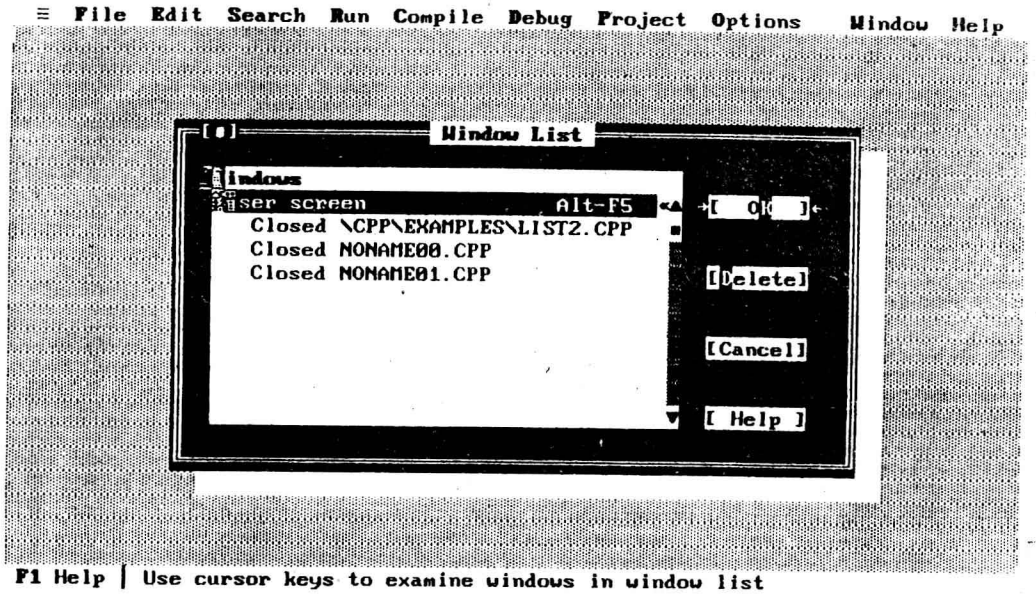


图 0-5

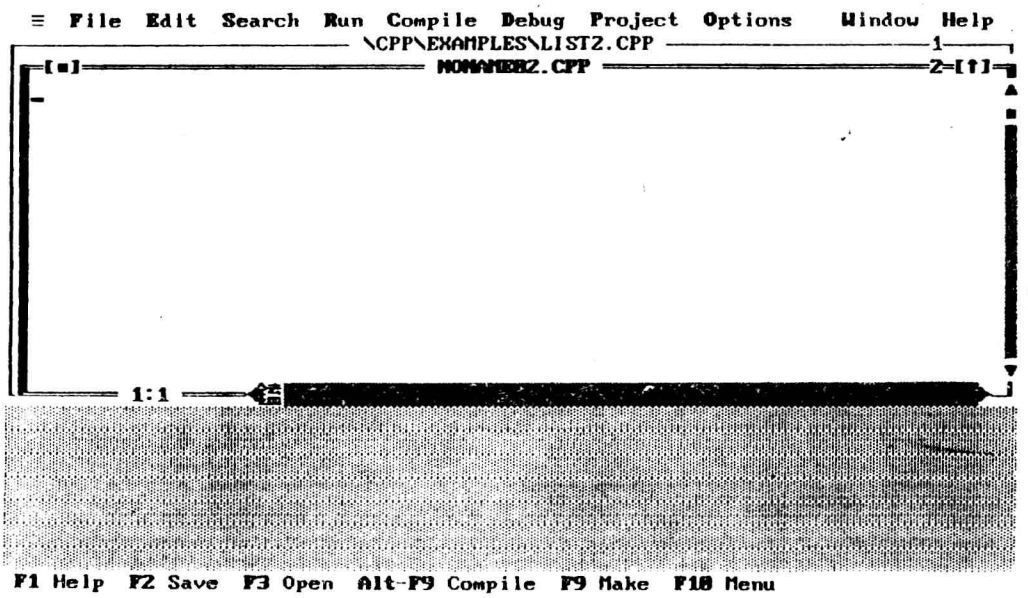


图 0-6

『Edit』功能下还有一个子功能为『Show Clipboard』，它会显示 Clipboard 里的内容，除非使用 CTRL-KH，否则它的内容也以反白出现。同时，Clipboard 的内容是累积的，也就是说，每当我们用到 COPY 功能，新的区块便会加入 Clipboard 中；而每当用到 Paste 时，Clipboard 内所有反白的部分也会被移至编辑程序内。因此，对于不需要的内容，一定得以 CTL-KH 将反白消除，使该部分无法重新被使用（除非重新再设置），或者以 CTL-Y 将其完全删除。编辑后的程序别忘了存储，其方式有三种：

1.利用『File|Save』，或按 F2。

2. 『File|Save as』，它可以保留原来程序的内容，而将修正过的程序以新的名称存入。

3.『File|Save all』，可将所有编辑窗口一一存入。

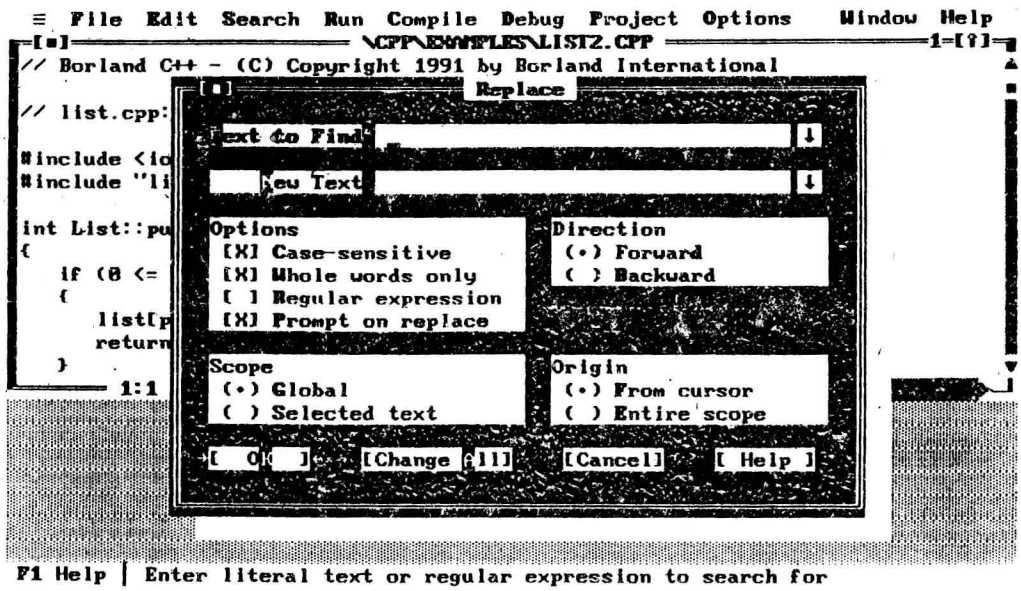


图 0-7

修改程序

程序的修改固然可以在编辑下进行，然而这里所介绍的，乃是快速修改的功能。回到窗口 1，如果我们希望将 LIST2.CPP 中的所有的 i 改为 index，以正确表示这个变量所代表的意义，在编辑环境下，必须一行行的找 i 再将其改为 index，不但浪费时间，也可能忽略了一些应该被修改的地方。

幸好，Borland C++提供查找与替换的功能，在主菜单内选择『Search』再利用『Replace』的 dialog box，如图 0-7，在『Text to Find』的地方输入 i，按 Tab 至 New Text，输入 index，再按 Tab 至 options。在 case sensitive 与 prompt on replace 前各有一个 X，加上 Direction、Scope 和 Origin 中也都设置了缺省值，使这个组合表示不论 i 是某个字(Word)内的一个字母或 i 独立存在，都必须停住由用户回答 Y 或 N 来确定

该 i 是否应改为 index，查找的方向为 Forward，范围为整个程序，并由光标所在处开始，我们可以一直按 Tab 至屏幕下方确定指令运行处，选择 Chang ALL 并按 Enter 开始执行修改。

此时，每一个 i 出现处都会暂停，由 user 判断是否应将那个 i 改为 index。读者也可将 OPTIONS 改为 whole words only，重新运行，以比较其差别。

对于上述对话框中的选项其功能分别述说如下：

『Options』内的『whole words only』表示要寻找的字为独立的字，而非出现在某字中，正规式(regular expression)表示在搜寻时可使用(wildcard)表示要寻找的字符串，其功能类似 UNIX 下的正则表达式用法。

『Prompt on replace』则使在取代之前先出现提示字符串。

在『Scope』对话框内的『Selected text』表示对特定区域内的文本数据进行搜寻。

『Direction』对话框内的『Backword』则设置搜寻方向为由后往前寻找，有别于『Forward』的搜寻方向。

『Origin』对话框内用来设置搜寻起点，『From cursor』表示由光标当前所在处开始寻找；『Entire scope』则表示从头开始整个搜寻。

您也可以利用鼠标来编辑程序。只要将光标移到适当的地方按一下左钮，即可在该处输入新数据。若是处理多个窗口时，只要将鼠标移至要处理的窗口按一下，即可将该窗口变为『活动窗口』。

另外若要处理整块区域的数据，只要将鼠标移至该区块的开始处，按下左钮，再拖移(drag)鼠标至该区块的末尾，该区块的数据即呈现反白。此时可对该区块进行粘贴(paste)、搬移（move)和拷贝(copy)。

下面列出基本的编辑指令：

按　　键	功　　能
<Ctrl><N>	插入一行空行
<INS>或<Ctrl><V>	插入模式和覆盖模式间的切换指令
↑ ↓ ← →	方向键，依其方向使光标自由移动
<PgUp><PgDn>	向上、下滚动屏幕
<HOME>	将光标移至一行的最前头
<End>	将光标移至一行的最后头
<Ctrl><Q><R>	移至文件的开头
<Ctrl><Q><C>	移至文件的最后
<Ctrl><K><B>	标出区块开始的标号
<Ctrl><K><K>	标出区块结尾的标号，被标出的区块数据会以高亮度显示
<Ctrl><G>或<Del>	删掉光标所在字符
<Ctrl><Y>	删除光标所在的那一行
<Ctrl><K><Y>	删除所设置的区块
<Ctrl><Q><F>	寻找字符串。编辑程序会出现提示字符要求输入要寻找的字符串。