

# 第一部分

## Microsoft C / C++基本特征

### 第1章 Microsoft C / C++初步

本章讨论在计算机上安装 Microsoft C / C++，同时说明如何利用 Microsoft C / C++ 的卓越性能。本章首先介绍如何安装 Microsoft C / C++ 以及如何根据特定需要作配置；其次介绍一些有关于程序员工作平台 (PWB) 的最有用的性能；最后，在熟悉了 PWB 之后，将讨论如何编写简单的 C 程序。

#### 1.1 了解 Microsoft C / C++ 磁盘

当拿到 Microsoft C / C++ 时，也许会被其中磁盘的数量吓了一跳。幸运的是，不必担心要考虑这些磁盘中的每个文件。这一艰苦工作已由 SETUP 程序完成了。

开始安装之前，先化几分钟时间为源盘作一份备份。由于 Microsoft 未对磁盘加密，可以用 DOS 下的 DISKCOPY 程序作备份。

##### 1.1.1 安装 Microsoft C / C++

Microsoft C / C++ 发行包中含有 9 张 3 1/2 寸盘或 11 张 5 1/4 寸盘，外加一张单独的 Microsoft 剖视器盘，这些盘片中含有安装一完整的 Microsoft C / C++ 时所需要的任何东西。

为进行安装，必须运行 Microsoft 在“Microsoft C / C++ Setup / Disk1”盘中提供的 SETUP 程序。SETUP 程序自动判定计算机上是否安装有 Microsoft Windows (必须是 Windows 3.0 或 3.1)。如果安装了 Windows，SETUP 程序作为 Windows 的一个应用程序，否则，SETUP 程序作为 DOS 的一个应用程序。

如果希望开发 Windows 应用程序，必须在系统上安装 Windows。当 SETUP 程序成为一个 DOS 应用程序时，它仅安装 DOS 开发工具。

若 Windows 尚未安装在硬盘中，而又希望开发 Windows 应用程序，请在安装 Microsoft C / C++ 7.0 之前安装 Windows。

根据选择，Microsoft C / C++ 要求 10 到 20M 磁盘空间。标准安装 (包括 Windows 开发工具、类库，及样本文件) 需 15-30M 空间。最小系统 (仅含为 DOS 应用开发的基本 C / C++ 编译器，二种内存模式配置，程序员工作平台，CodeView 跟踪程序，联机帮助文件) 大约需要 10M 空间。

### 1.1.2 运行 SETUP

安装 Microsoft C/C++和它的同类产品非常容易。Microsoft 的 SETUP 程序免除了安装大程序包。具体说,所有程序员要做的工作就是告诉 SETUP 程序想要的是哪个选项,然后在提示下装入每张盘。除了插入正确的磁盘外,程序做了所有剩下的工作。

Microsoft 剖析器(Profiler)不在 Microsoft C/C++ SETUP 程序中安装。为安装剖析器,应先安装 Microsoft C/C++程序包。然后插入“Microsoft Source Profile Setup”盘,并键入:

```
n:setup
```

其中 n:是装有 Setup 盘的软盘驱动器。

运行 SETUP 前,确认有足够的硬盘可用空间。标准安装要求 20M 左右的硬盘空间,要花 1 至 2 个小时来完成。

开始安装过程,把标有“Microsoft C/C++ Setup/ Disk1”的磁盘插入一个软盘驱动器,同时输入:

```
n:setup
```

其中 n:是装有 Setup 盘的软盘驱动器。

当 SETUP 开始执行时,自动搜索 Microsoft Windows,如硬盘中已安装,启动 Windows SETUP 过程;如 Windows 不可用,SETUP 作为一个 DOS 应用程序并提供一些不同的安装选项。

### 1.1.3 DOS 模式下 SETUP 的使用

如果使用基于 DOS 的 SETUP 版本,请阅读本节。如使用基于 Windows 的 SETUP 版本,请跳至下节“基于 Windows 的 SETUP 的使用”。

DOS SETUP 显示屏见图 1.1。最简单的安装过程是用箭头键把加亮块移至“Install the MS C/C++ compiler using defaults”并按回车键。

Microsoft C/C++7.0 发行包的缺省安装目录是 \C700。本书的所有例子中,都假设 Microsoft C/C++发行包存储在 \C700 目录中。然而,为了更灵活方便,可把 C/C++ 编译发行包安装在另一目录。

要作常规安装,选择“Custom Installation of MS C/C++ compiler”。该选项显示出当前要安装的项目(见图 1.2)。

修改一个选项,用光标键把加亮块移至想改变的选项。按空格键决定是否选择某一选项。例如,据缺省约定,SETUP 安装示例源代码程序。不想安装 Microsoft 示例程序,把加亮块移至“Sample Source Code”项并按空格键作出选择。使用这一特点可以在安装过程中只选需要的项,也可能会选择不同的“内存模式”配置。内存模式在本章后面的“内存模式”一节中讨论。

对需要安装的类型选择后,SETUP 程序把程序包拷贝到硬盘并按事先选择进行配

置。SETUP 提示何时需要插入指定磁盘到软盘驱动器中。当所有软件都被拷贝到硬盘，SETUP 创建需要的库去支持不同的内存模式配置。

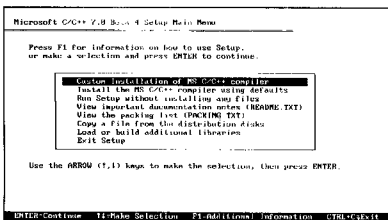


图 1.1 DOS SETUP 主菜单

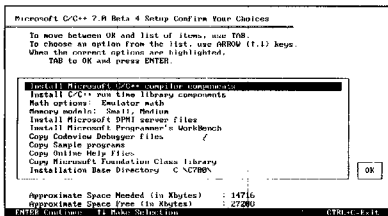


图 1.2 DOS SETUP 用户安装屏幕

#### 1.1.4 基于 Windows 的 SETUP 的使用

开始执行基于 Windows 的 SETUP 程序时，它显示信息“Initializing Setup”。初始化过程需几分钟，所以不要担心这些延迟。初始化期间，SETUP 自动检测运行的 DOS 版本、Windows 的目录及版本等。SETUP 完成初始化后，要求指定安装目录。缺省目录是 \C:\00，但可以修改以便把 Microsoft C/C++7.0 安装在希望的目录之中。

接下来，SETUP 让你选择是缺省安装还是常规安装。缺省安装是最容易执行的，但必须先注意缺省配置不安装 MS-DOS 图形库文件。

## 注意

如果使所写的 DOS 应用程序能调用图形过程或图形库中的公共文本模式图象函数,就必须要选择常规安装。本书中很多例子都要用到图形库,故建议采用常规安装以确保图形库的安装。

要安装图形库,选择常规安装,得到一个 C/C++ 7.0 文件组对话框(见图 1.3)。缺省的常规安装自动包括了 MS-DOS 图形库。

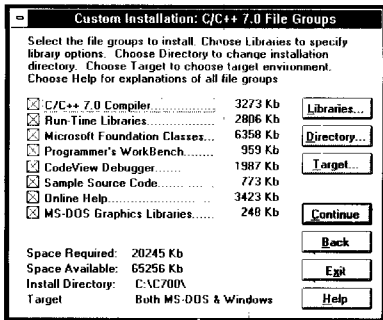


图 1.3 C/C++ 7.0 文件组对话框

此时可以选择性地去安装一些附加支持库,后面用到的一些例子中要求这些库。如果现在跳过这步,以后需要附加库时,可以重新运行 Microsoft C/C++ SETUP 程序并在 SETUP 菜单上选择“Build Libraries”项。利用建库项可以容易地把库加入已存在的 Microsoft C/C++ 中。要放弃附加库加入,按文件组对话框中的“Continue”键。

### 1.1.5 附加库安装

缺省安装和常规安装都只安装了小内存模式和中内存模式(关于内存模式参阅下节)。对本书的大多数程序来说,这些模式是足够应用的。但也有例子要求附加库。如果有足够的磁盘空间并希望编译“使用 Microsoft C/C++ 7.0”中的所有程序,建议尽可能地安装内存模式配置。紧凑模式只在 Windows 一章中使用,如果不想对 Windows 编程,就不需要安装紧凑模式。

要安装所有内存模式,在文件组对话框中选择“Libraries”键,它将显示“C 运行自定义对话框”(见图 1.4)。在“Memory Models”标题下,用鼠标选取每种内存模式,或用箭

头键移到该项并按空格键作出选择。

如果想在 Windows 下编程 (见 19 到 21 章), 还需安装每个 Windows 对象。选择每一个 Windows 选项: “Windows .EXE Files”, “Windows .DLL Files” 和 “QuickWin.EXE Files”。用鼠标选取未选项, 或用 Tab 键从 “Memory Models” 框走到 “Windows Targets” 框, 然后通过箭头键和空格键作出选择。

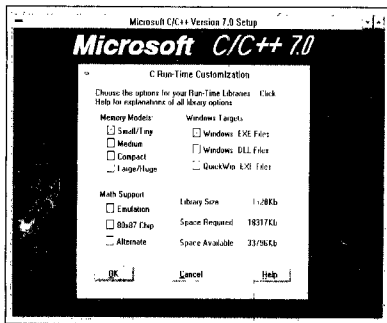


图 1.4 C 运行自定义对话框显示的 MS-DOS 选项框

在安装进行过程中, SETUP 提示何时需另一张盘, 因为 SETUP 是作为一个 Windows 应用程序来运行, 在拷盘过程中 (最多可能费时 90 分钟) 你可以做其他工作。安装过程中, SETUP 将显示 README 图标。如果在这图标上双击鼠标, 就可以在安装过程中阅读 README 文件了。

### 1.1.6 选择内存模式

安装实用程序最重要的一个部分是选择想安装的内存模式。Microsoft C/C++7.0 编译器产生适于 Intel 80×86 系列微处理器的程序。由于历史原因, Intel 80×86 处理器被设计成支持 8080 微处理器所用过的内存配置, 8080 是现代 PC 系列的先驱。8080 CPU 只能最大寻址到 64K 内存, 因此, 现代 80×86 系列按 64K 内存段来定义其巨大内存能力。内存模式涉及到如何在应用程序中使用有效的 64K 内存段。

按照惯例, 有 6 种不同内存模式。表 1.1 总结了每种内存模式的特点。对每种内存模式, SETUP 程序为每种内存配置创建库文件。在使用 Microsoft C/C++ 编译时, 可通过选项来选择想要使用的内存模式。对本书中的大多数例程, 可使用编译缺省的小内存模

式来编译程序。

表 1.1 内存模式设定

内存模式	特 点
Tiny(微)	代码、数据、堆栈的可用空间为 64K，使用近指针。
Small(小)	代码可用空间 64K，数据与堆栈可用空间 64K，使用近指针。
Medium(中)	代码可用空间 1M，数据与堆栈可用空间 64K，程序中可用远指针，但数据与堆栈中只能有近指针。
Compact(紧凑)	代码可用空间 64K，数据与堆栈可用空间 1M，程序用近指针，数据用远指针。
Large(大)	代码可用空间 1M，数据与堆栈可用空间 1M，使用远指针。
Huge(巨)	代码及数据中均使用远指针，通常的堆栈 64K 限制被突破。

内存模式的选择影响到编译程序如何把 C 或 C++ 源代码翻译成机器指令。特别是，如果有一个需要很少数据内存的小应用程序，请选择小模式编译，产生最有效的机器指令，这可使程序更小、更快。设置编译内存模式选项，参阅本章后面“语言选项”一节。

另一方面，如果应用程序很大或需要大量的数据内存，必须使用紧凑、大或巨型模式。存取这部分内存需要双字节的机器指令，会使程序略显大一些，更慢一些。

缺省安装提供小模式和中模式。如果硬盘上有空间，可以马上安装所有模式。然而，支持所有内存模式的安装占据很多你也许不想用的额外空间。对此情况，你可以指定想包括进去的特定内存模式。

由于模式间的相似性，Microsoft 综合了一些支持库，以便在某些情况下可以一举两得。这样，如果选择了不同的内存模式支持，而不是缺省配置所提供的，就可以在下面作出选择。

Small / Tiny  
Compact  
Medium  
Large / Huge

#### 使用 PWB 或命令行编译器前请注意

Microsoft C / C++ 7.0 编译器必须在 80386 保护模式下执行。有两种方法可以满足这一需要。

如果有 Windows，可以在 DOS 窗口或作为一个 Windows 弹出的 DOS 应用程序去运行编译程序或程序员工作台(PWB)。在这种操作方式下，DOS 在保护模式下执行。

如果没有 Windows，Microsoft C / C++ 程序包中包含 386MAX 内存管理实用程序。为在 DOS 下(无 Windows)运行 C / C++ 编译程序，必须运行 386MAX 安装程序。386MAX.SYS 设备驱动程序将被安装在 CONFIG.SYS 文件中，代替 WINDOWS 系统的 HIMEM.SYS 驱动程序。一旦安装好，386MAX.SYS 为你提供保护模式的 DOS 环境，而且还提供与 Windows 相兼容的 HIMEM.SYS 接口。

如果试图在不使用 386MAX.SYS 或不在 Windows DOS 窗口中运行 Microsoft C/C++7.0 编译器, 将会收到一条来自编译程序的错误信息, 它要求保护模式接口。

## 1.2 程序员工作台(PWB)

有了程序员工作台 (PWB), 就可以在一个方便的环境中完成所有的编程工作。PWB 提供一个文件编辑器和一些项目维护工具。在 PWB 中, 可以编译、make、执行和调试程序。若需要, PWB 会自动调用 NMAKE 或 CodeView, 使你能直接调用其他开发工具。本节介绍程序员工作台并帮助你马上开始使用它的一些特点。

### 1.2.1 PWB 初歩

开始 Microsoft C/C++7.0 的工作很简单。只需进到 Microsoft C/C++ 目录下的 BIN 目录 (比如: \C700\BIN) 然后在 DOS 提示符下键入 PWB。开始编辑一个已在程序, 可以在命令行选项指定源文件名, 象这样:

```
PWB source C.
```

用这条简单命令启动 Microsoft C/C++ 可以工作得很好并能满足你的大部分需要。程序员工作台提供了一个集成开发环境。从本质上说, PWB 就象一个含有编译和运行程序所需各种命令的文本编辑程序, 还含有大量的配置选择用来使 PWB 本身和编译输出规格化。PWB 中还含有一个集成的“源程序浏览器”, 一种功能强大的, 帮助理解和监测程序结构、设置界限和符号参考器、显示层次级别、以及显示有关你的 C 函数调用其他 C 函数的信息的实用程序。

然而, 不是一定要使用 PWB 的。许多程序员喜欢用自己的文本编辑程序, 并且直接调用编译和连接程序。本章后面“PWB 之外的编译与连接”一节将描述命令行编译的使用。

### 1.2.2 PWB 菜单和窗口的使用

有了程序员工作台, 需要的所有编程工具就在一个单一的, 容易使用的环境中。本节介绍如何使用 PWB 菜单和窗口。图 1.5 示出了 PWB 的概貌。

程序员工作台 (PWB) 主要由三部分组成:

- 菜单系统。菜单系统能访问 Microsoft C/C++ 的功能强大的实用程序和功能部件。可以通过菜单项的选择使用它们, 正如你看到的 (图 1.5), 菜单项在屏幕的第一行。菜单项中的每一命令对应一组实用程序或功能部件。
- 窗口系统。窗口系统 (PWB 中最常见和最有用的部分) 提供一组窗口。窗口用于编辑程序、调试程序和观察程序的操作。另外一些窗口用来选择实用程序或显示重要消息。图 1.5 显示了一个编辑窗口, 在该窗口可以创建并修改程序。这个窗口中含有 FIRST.C 文件。
- 状态条。在屏幕的最下面, 状态条显示了最常用命令的捷径键, 告诉 Microsoft

C/C++正在做什么，并提示应选的命令。在图 1.5，状态条显示了一组快捷键。

### 1.2.2.1 菜单系统

菜单系统包含有效管理编程任务所需要的实用程序及功能部件。这些实用程序或功能部件能处理文件，编辑，编译，调试，运行用户自己的支持程序。有三种方式进入菜单系统：两种键盘方式，一种鼠标方式。

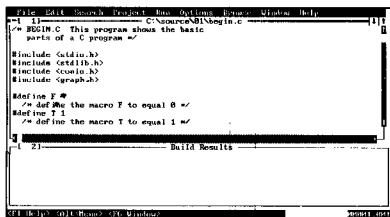


图 1.5 程序员工作平台

移至菜单中某一命令项并选择它。在PWB中按ALT键可从任何地方移至菜单中。当光标移到菜单项时，使用箭头键移至某特定命令处。当命令加亮时，按回车键拉下一个菜单。可以用箭头键去选一个命令或直接键入命令名中的加亮字母。完成命令选择后，按回车键执行。图 1.6 显示了 Search 命令的下拉式菜单。

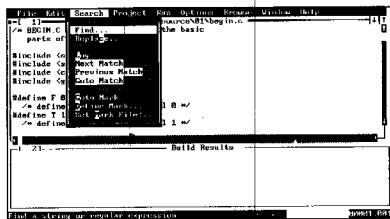


图 1.6 PWB 中的一个下拉式菜单



- 使用 ALT 组合键。要在菜单中选择一命令，按下 ALT 键并按下命令的加亮字母。当下拉菜单出现时，用箭头移到所需命令处，然后按回车键选择。命令的选择会导致三种可能结果：
  1. 命令被执行；
  2. 显示了菜单；
  3. 显示对话框，其中含有一个菜单，提供所选项的其他选择。命令执行之前必须在对话框中提供合适的必要信息。
- 在菜单中用鼠标选取一命令。可以在下拉式菜单中用同样方法选取命令：利用鼠标指向命令并选取。如选择的命令使得弹出一个对话框，也许需要键入一些信息。然而，对于对话框中的许多选择来说可以使用鼠标。本节后面将彻底详细地讨论对话框。

### 1.2.2.2 窗口系统

当使用 Microsoft C/C++ 时，大部分工作是在窗口中完成的。PWB 中有几种窗口可用，它们提供了一些区域用来书写程序，观察程序输出消息，显示状态信息，以及记录正在计算机上发生的事情。大多数窗口有着类似的特征（见图 1.5）。

Microsoft C/C++ 窗口具有下面基本特征：

- 关闭按钮。关闭框在窗口的左上角，在这儿按鼠标就会关闭窗口。如没有鼠标，可以通过窗口下拉菜单中的关闭命令来关闭一个窗口。
- 题条。题条位于窗口顶部，显示窗口名字。拖动题条就可移动窗口。
- 窗口号。窗口号帮助跟踪已打开的窗口。当窗口重叠时，一些窗口可能隐藏在另一窗口之后。要看当前活动窗口，可以选择窗口菜单选项。头四个菜单，被标为 1 至 4 号，是这个菜单底部的选项。如活动窗口号超过四个，另一选项“**All Windows**”作为窗口菜单的最后一项出现。这一“**All Windows**”选项显示一个含有所有活动窗口表的对话框。从对话框中可以选择一个窗口。采用下面某种方法可以激活一个窗口：
  1. 访问 1 至 9 号窗口，按下 Alt 键再输入数字。
  2. 选择窗口菜单并选取欲激活的窗口。前面四个窗口是菜单的 1 至 4 个选项。超过四个窗口时，下拉菜单底部出现“**All Windows**”选项。选择它就出现所有窗口的对话框。通过对话框可以在所有活动窗口中作出选择。
  3. 在任何菜单中按 F6 键就顺序切换到下一个菜单。
  4. 在窗口的任何地方按鼠标键。

活动窗口出现在所有其他窗口的上面，并在下面及右边显示滚动条。

- 放缩按钮。放缩框在活动窗口的右上角，它允许把窗口放大到其最大尺寸。在放缩框上按鼠标键就激活了“/Zoom”命令。如果没有鼠标，可以在窗口菜单中选择“**Maximize**”或“**Minimize**”。如果窗口已是最大尺寸，在放缩框中按鼠标键则窗口变成其原来尺寸。
- 滚动条。垂直和水平滚动条在窗口的底部和右边。利用滚动条可滚动窗口内容。滚动的两端均为箭头。触发箭头使得屏幕一次移动一行。紧挨着箭头的是阴影框，触

发阴影框使窗口内容换一页。拖动滚动条可快速改变窗口内容。滚动条的位置指出了窗口中与文件中剩下部分相关信息的位置。

尺寸修改按钮。当鼠标光标指向窗口右下角时，通过触发并保持鼠标键可以改变窗口的尺寸大小。这时只要拖动鼠标就可改变窗口大小，窗口大小合适时释放鼠标键。

对话框可能是你不熟悉的一种窗口。对话框由一些菜单命令弹出，能对所选命令作一些选择。任何后面带有省略符 (...)的菜单命令在选择时都将产生一个对话框。图 1.7 显示的是“Search”菜单中“Find”命令的对话框。

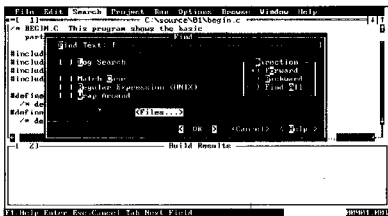


图 1.7 典型的对话框

大多数的对话框包括下列项目：

- 输入框。可以在输入框中输入文本并用基本的编辑命令去处理它们。如需要输入控制码到输入框中，须先键入“P”。如果输入框紧跟一个下箭头符号，则有一序列值与输入框相关，其中含有使用这个域的选择元值，在输入框中上下箭头或在下箭头符中触发鼠标就能访问这个值序列。每按一次箭头键，表中另一个值就循环进入输入框中。
- 列表框。列表框能从列表中选择一项而不必离开对话框。显示的列表长度可变。关于列表框的典型例子是在“FILE”菜单中选择“Open”命令的文件列表的显示。
- 动作按钮。当在对话框中作完选择或输出所需信息后，用适当的动作按钮执行命令，这些动作按钮是一些被尖括号括住的符号，比如<OK>。标准的动作按钮有“OK”、“Cancel”和“Help”（见图 1.7）。激活动作按钮可以用鼠标触发，也可以在活动框中按下 Alt 键再按加亮字母。
- 广播按钮。广播按钮用来从一组相互排斥命令中选取一项（只一项）。按“广播按钮”，可以用鼠标触发或按 Alt 键再按你想要的广播按钮中的亮显字母。也可以用 Tab 键进到一组广播按钮，然后用箭头及空格键去选择特定广播按钮。图 1.7 中对话框右边“Direction”标题下的就是广播按钮。

• 检测框。检测框（由方括号括住的项目）用来打开或关闭选项。如果检测框中含有 X，该选项是打开的。如果检测框中不含 X，该选项被关闭。如几个检测框属于一个项目的不同应用，他们被集中在一起。用 Tab 键进入一组检测框中，用光标键指定一个特定检测框，再按空格键去打开或关闭选项。如果有鼠标，触发检测框决定是打开或关闭。图 1.7 中所示的“Find Text”输入域下的就是检测框。

### 1.3 Microsoft C / C++ 配置

Microsoft C / C++ 的优点之一就是其通用性。有了 Microsoft C / C++，就不会被束缚在一个编程环境之中。Microsoft C / C++ 改变编程环境以适应你的需要。通过选项菜单可以改变 Microsoft C / C++ 环境，也可利用选项菜单使你要对之编程的功能块规格化，选项菜单还可以在运行编辑程序时设置键盘。在后一种情况，能选择重新定义编辑命令键，这样就仿你熟悉的编辑器。

通常，可以不修改任何配置而用 PWB 去书写，编译并执行程序。如果这样，请直接跳到“编制第一个 C 程序”这一节。在使用某些 PWB 特性时再返回到本节来。

在程序员工作台上可以对编辑环境作许多修改。环境修改功能在菜单系统的“Options”命令下面，“Options”菜单分成五部分：PWB 环境、项目选项、指定语言选项（C 或 C++），CodeView 测试程序的连接与访问，以及对象浏览程序。图 1.8 示出了“Options”菜单。

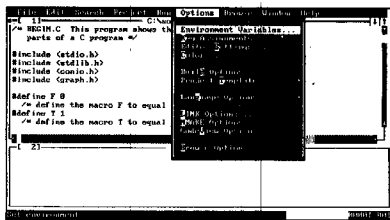


图 1.8 Microsoft C / C++ 的“Options”菜单

“Options”菜单的第一部分用于整体环境控制。使用“Environment Variable...”选项可以在查看或编辑一组 DOS 环境变量。后面二个选项用来控制 PWB 自身的外部显示及操作。

#### 1.3.1 键设置

要重新定义编辑器所用的命令键，请选择“Key Assignments...”。功能缺省设置包括

许多 DOS 和 Windows 应用程序使用的与标准公共用户访问 (CUA) 可兼容性的设置。另外, 编辑器支持一度被广泛作为标准的 WordStar 字处理软件的键盘设置。这意味着常常可用两种命令键去完成同一功能。例如, 转换插入和修改输入状态既可以按“Ins”键, 也可以按“Ctrl-V”。

要重新定义键, 从图 1.9 中所示的“Key Assignments”对话框中的“Macro / Function List”列表框中选取所希望的命令函数。例如, 要修改插入 / 修改状态转换键, 可把光标移至列表框中的插入模式项上。在标题“Current Keys”下, 可看见“Ctrl-V”和“Ins”项。要定义完成这一功能的新组合键, 在“Unassigned Keys”列表框中选取一个未设置键。当在该列表框中滚动查找时, 当前选择出现在“New Key”输入域之中。确定好所希望键后, 选择 <Assign> 设置命令。

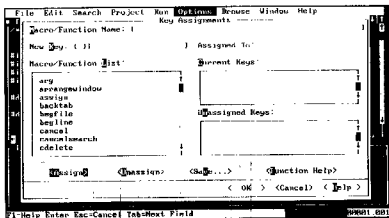


图 1.9 键设置对话框

对所有想定义的功能只要重复这一过程, 即可要去掉一个相关功能的键, 在“Current Keys”序列表中选择当前键并选<Unassign>命令。

完成功能键重定义后, 选择<Save...>。这时产生一个序列框, 其中存有所作的所有功能键修改。如果想去掉某些修改, 把序列表中的光标放在指定项上, 同时选择<Delete>。修改完毕, 选<OK>存储功能键定义。此后, 功能键定义就可在编辑器中使用。

### 1.3.2 编辑器设定

利用编辑器设定 (Editor Setting) 对话框可以改变大量编辑器操作的缺省值。例如, 编辑器通常不作字包装, 然而可以通过使用“Editor Settings”对话框 (见图 1.10) 来改变编辑器的字包装选项。

要改变一个指定选项, 用“Switch List”列表框滚动找到该选项。注意编辑器设定出现在三个不同的分离组之中。图 1.10 只显示了有布尔参数的选项。要看其他设定, 在“Switch Type”广播按钮中选择“Numeric”或“Text”。

当在想要改变的项目上完成定位之后，就可在“Switch”输入域编辑参数值，然后选<Set Switch>动作按钮。完成修改之后，请选择<Save...>项。

如果不能肯定含义隐藏的文字标题功能是否要改变，可选择<Switch Help>。它显示关于当前“Switch List”中加亮项的详细解释信息。

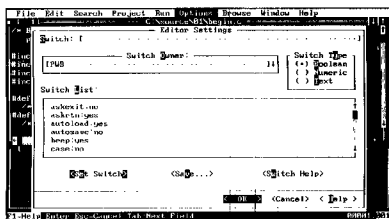


图 1.10 编辑器设定对话框

### 1.3.3 颜色

PWB 允许在 PWB 中创建自己的颜色模式。进入颜色选择对话框，在“Options”菜单下选择“Colors...”项即可。图 1.11 示出“Colors”对话框。

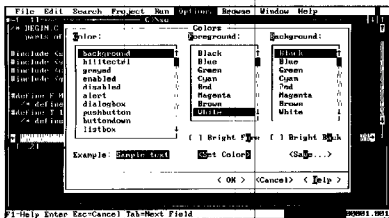


图 1.11 颜色对话框

要改变用户界面中某指定项的颜色，必须先在标有“Color:”的列表框中找出该项，在此列表框中滚动查找时，对应的当前颜色在右边的“Foreground”和“Background”前、后

景颜色框中显示出来。缺省的颜色模式采用相对简单的颜色，最常见的是白底黑字、黑底白字或兰底白字。

改变某一用户界面元素的颜色，要选取希望的新前景、背景颜色。对话框右下角的名称为“Example”的例子区域中显示颜色组合情况的。如对所选颜色满意，请选 <Set color> 项。完成修改后请选择 <Save...> 项。

下一个问题是用户界面元素的正确选取。在“Color:”列表框中的某些名字是比较明显的，但很多却不是。毫无疑问，对于 PWB 的编写人来说，名字的意义当然是明确的。但对用户来说却不然。很可能必须反复进行颜色修改。作一次改变，再返回 PWB 去看是否修改的是正确的用户界面元素。重复这一过程直至定位到所希望的界面分量并成功地改变了它的颜色。

### 1.3.4 建立选项

“Build”建立选项有一个简单的功能：通过一些简单选择，使编译和连接之后的文件可用 CodeView 测试，或选择一种可释放可执行的构成结构。“Release”选项中的对话框供你选择 组合适当的编译、连接选项，使生成的可执行文件小、快，并且不包含任何用于调试的代码或检测功能。一旦程序完成并且检测过，可使用“Release”选项设定去编译程序的最终版本。

如果要使用 Codeview，则记住选择使用“Debug”选项。仅需在“Link”选项中的对话框中选择其中的两个广播按钮之一就可到达该级选择。

### 1.3.5 设置项目模板

项目模板指定运行时支持类型及可执行文件的类型。通过对话，可以生成不同种类的可执行文件，象 DOS 的 .EXE 文件，Windows 的 .EXE 文件，或 P-code 可执行文件。P-code 程序是 Microsoft C 编译器第 7.0 版的新产品。它以一定的执行速度为代价，提供一种被编译程序的紧凑存储模式。详见第 10 章“使用 p-code”一节中的“使用 Microsoft C/C++ 高级功能”。

可在运行支持列表框中选取所需的语言，然后在项目模板序列表框中选择你所想要的输出文件类型。

### 1.3.6 语言选项

根据所用的编程语言是 C 还是 C++，语言选项略有不同，选择了一种语言，就显示相应的编译选项。图 1.12 显示了 C 编译器选项的对话框。

“Memory Model”域用于选择合适的内存模式：小、中、紧凑、大、巨或常规内存模式。“Processor”域指定执行程序在何种处理器上运行。选择 8086 可使程序在任何类型的 80×86 处理器上运行。然而，如果已知道执行程序将在 80286 上运行，选择 80286 使编译器产生只有 80286 或更好的 CPU 上才有的机器指令，使得代码更紧凑，执行速度更快。

用 C 编译器选项对话框中的“Warning Level”项，选择警告级。只有在源程序中出现问题，但又构不成错误时，才出现警告。警告级从 0 级（关闭所有警告信息的产生）到 4

级（产生所有可能的警告信息）。级别 1 是缺省设置，只显示隐含的严重警告信息。级别 2 加进了对许多不太严重状态的警告。级别 3 对较不严重的情形也作警告。最后，级别 4 提供尽可能多的警告信息，包括对使用 Microsoft C/C++ 语言中非移植特征的使用也提出警告。

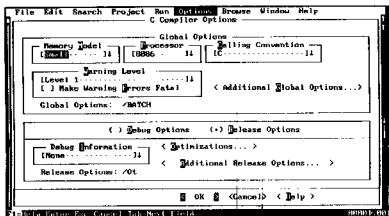


图 1.12 C 编译器选项对话框

<Additional Global Options...> 按钮作一些附加的的编译器设置。从这个对话框（不显示），可以对保护模式下窗口应用程序兼容性进行选择，还有 ANSI C 对 Microsoft C 语言的扩展，以及其他特征。

<Optimizations...> 动作按钮显示 组广播按钮和检测框，见图 1.13。

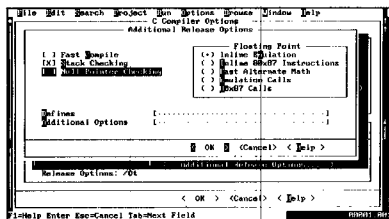


图 1.13 附加的释放选项对话框

该对话框控制编译器优化的级别和类型。

<Additional Release Options...> 按钮提供了对浮点控制特征的访问。在这个对话框

中，可以请求生成与 80×87 兼容的代码。

### 1.3.7 连接选项

可使用“LINK”连接选项，选择一组与连接步骤相关的项。这里用户可指定附加库，递增连接（它比普通连接快），以及 CodeView 兼容性。假定程序文件在打开文件调试选项时被编译（见“Build Options”），连接器把调试码从目标模块结合到最终生成的可执行文件中。如希望生成一个把各种公共变量地址显示出来的映象文件，可以从“LINK”选项中选择<Additional Release Options...>，在此子对话框中选择“Map File”选项。

### 1.3.8 NMAKE 选项

NMAKE 是 Microsoft 的项目管理实用程序。NMAKE 通过判定哪个源文件发生了改变以及哪个必须重编译来管理大项目。当某源文件改变时，NMAKE 自动调用编译器（汇编文件调用 Microsoft 汇编程序进行编译）。缺省情况下，如 NMAKE 在运行时遇到错误，它停止执行，如项目中含有多个源文件，NMAKE 遇到一个错误就停止编译并且不再编译其他余下的文件。如果希望 NMAKE 跳过去处理下一个文件，可以在 NMAKE 选项对话框中选择“出错时继续建立不相关目标”。

如果你的计算机有扩展内存(EMS)或扩充内存(XMS)，可以在编译和连接过程中使用这部分内存。缺省情况下，只要可用，NMAKE 自动使用可用的 EMS 或 XMS。

## 1.4 在 PWB 内编译并执行程序

为编译单个源文件的程序，在“Project”菜单中选择“Compile File”。这时编译源文件，产生一个.OBJ 模块。要编译并连接程序，则要选择“Build”功能（参见第五章“Microsoft C/C++程序”的构造、编译、连接和检测，以便学习如何建立一个多个文件组成的程序）。

运行已编译和连接的程序，在“RUN”菜单中选择“Execute:”，程序就可装入运行。一旦程序执行结束，控制返回 PWB。这样，可以把编辑、编译和运行工作都放在 PWB 中完成。

在 PWB 中还可登录到 CodeView 调试器，第九章介绍了它的使用。

## 1.5 在 PWB 外编译和连接

也可以在 PWB 环境之外编辑、编译和连接程序。利用正文编辑器键入并编辑源程序，再在 DOS 命令行状态下通过键入命令：

```
cl source.c
```

来直接运行编译器。

因为编译器既可编译又可连接程序，因此被命名为 cl，最后生成可执行文件 Source.exe。此后，就可以象运行任何其他 DOS 应用程序一样去执行该程序：即只需在命令行输入程序名。



使用 Microsoft C/C++ 工具的命令行界面时,可能希望获得联机帮助,这只要在 DOS 提示符后输入命令“QH”即可。它运行 Microsoft 快速求助实用程序,提供综合联机求助功能。

如果程序中有多个源文件,可以在命令行上加上其他文件,如

```
cl source.c source1.c source2.c
```

cl 编译每个源文件并调用连接器把三个目标模块连接成一个可执行程序文件。如果要访问非标准库,或以前编译好的目标模块,也可在命令行中指定,如下所示:

```
cl source.c source1.obj graphics.lib
```

为帮助管理那些分成多个文件的大项目,可使用 NMAKE 程序。NMAKE 的使用程序在第 5 章中描述。

控制编译操作的大量选项可由命令行“开关”指定。每个开关都打开或关闭某一编译特性。开关是在斜杠字符后跟一个或多个字母,比如 /AM 或 /qc。开关可出现在命令行中的任何地方(但 /f、/qc 或 /oq 选项必须出现在命令行开始处)。习惯上,编译开关通常放在欲编译的文件名之前,例如:

```
cl /AM begin.c
```

表 1.2 中列出了一些最常用的开关。除了 /HELP 外所有开关编译器都区分大小写。关于其他的开关,可通过快速求助实用程序或通过查阅 Microsoft 参考手册来了解其功能。

要产生执行速度最快的代码,用选择开关:

```
/Oxaz /Dbz /Gr
```

来编译。

要产生最短代码,可使用 Microsoft C/C++ 的 p-code 或(紧凑代码)表示。采用这种形式,编译器生成指定的压缩执行代码,同时 Microsoft 连接器把一个 9K 的 p-code 编译程序加到代码文件中。由于这个原因,p-code 对于短程序优越性不大。另外,由于 p-code 程序比一般编译输出程序运行速度慢,故可在对速度要求不严格的地方使用 p-code 程序。要编译 p-code 程序,可把开关 /Oq 放在命令行的起始处。

下面是一些 cl 编译命令行开关。注意这些命令是区分大小写字母的。例如: /c 必须写成小写字母 c。

表 1.2 常用 cl 命令行开关

开关	目的
/AT	选择微型内存模式。
/AS	选择小型内存模式。
/AM	选择中型内存模式。
/AC	选择紧凑型内存模式。
/AL	选择大型内存模式。