

HOPE

Turbo C2.0

高级系统程序设计技术

北京希望电脑公司

5.1.150

TP311
4072

Turbo C2.0

高级系统程序设计技术

东阳生 王丹峰 张铭

北京希望电脑公司
一九九一年十月

目 录

前 言	(1)
第一章 Turbo C 简介	(2)
建立 TURBO C 环境	(2)
安装 TURBO C	(2)
开发一个程序	(3)
TURBO C 的专门特性	(7)
函数原型	(7)
新的类型修饰符	(8)
结构 / 联合对齐	(10)
预处理命令	(11)
Turbo C 编译程序的其它特性	(13)
库函数	(14)
第二章 Turbo C 高级程序设计	(15)
内存模式和说明	(15)
与汇编语言的接口	(17)
嵌入汇编	(21)
伪变量	(25)
实例: 存取中断的四种方法	(27)
中断函数	(33)
调试	(35)
移植性考虑	(36)
本书中函数的安排	(37)
第三章 键盘和控制台函数	(39)
键盘管理函数	(39)
KbdFlush	(41)
KbdGetShift	(43)
kbdSetShift	(45)
kbdGetC	(46)
控制台输入 / 输出函数	(48)
IocRawMode 和 IocCookMode	(49)

第四章 打印机管理函数	(51)
低级打印机控制函数	(51)
PrtInstalled	(51)
PrtSwsp	(59)
PrtTimeout	(60)
PrtReacdy	(60)
PrtInit	(61)
报表打印函数	(61)
PrtPutC	(62)
PrtPutS	(63)
PrtPosition	(63)
PrtNewPage	(66)
打印机管理函数的说明	(67)
MS-DOS 打印队列管理函数	(68)
PrtQueState	(68)
PrtQueSubmit	(69)
PrtQueCancel	(70)
PrtQueCancelAll	(71)
打印队列函数的使用说明	(71)
第五章 文件管理函数	(73)
FilRenameDir	(73)
FilGetVolid	(81)
FilSetVolid	(83)
FilDelVolid	(85)
FilGetTime	(86)
FilSetTime	(88)
FilOpen 和 FilHandle	(90)
第六章 视频函数	(93)
一般用途的视频函数	(93)
ScrGetMode 和 ScrSetMode	(103)
ScrGetCur 和 ScrSetCur	(105)
ScrGetStyle 和 ScrSetStyle	(106)
ScrPush 和 ScrPop	(108)
ScrReadWindow	(109)
ScrPutWindow	(110)
ScrPutS	(114)
ScrPutBox	(116)

ScrClear	(118)
ScrGetS	(118)
ScrPutAttr	(121)
修改 CGA	(123)
缓冲函数	(128)
BufInit	(130)
BufNextLine	(130)
BufShow	(131)
Bufget	(131)
BufFree	(132)
一个例子	(132)
 第七章 一个交互式屏幕设计程序	(134)
程序的编译和连接	(134)
使用屏幕设计程序	(150)
光标移动	(150)
块命令	(151)
菜单命令	(151)
设计目标	(154)
屏幕设计程序如何工作	(154)
信息显示	(156)
字符和属性选择	(156)
读/写文件	(158)
视频数据传送	(158)
块操作	(159)
一些可能的改进方案	(160)
字符绘制改进	(160)
I/O 文件改进	(160)
 第八章 与设备无关的图形函数	(161)
图形函数	(172)
GraTypes	(173)
GraInit 和 GraQuit	(174)
GraClear	(177)
GraSetCoord 和 GraGetCorrd	(178)
GraPoint 和 GraGetPoint	(181)
GraLine	(183)
GraBox	(185)
GraCircle	(186)

GraFill	(187)
对图形模块的增强	(189)
增加图形原语	(189)
对其它图形方式的支持	(189)
第九章 中断处理函数	(190)
中断和 MS-DOS	(199)
定时钟中断函数	(199)
IntClockInstall 和 IntClockRemove	(200)
Clock 程序的一个示例	(201)
BIOS CONTROL-BREAK 函数	(203)
IntBBInstall 和 IntBBRemove	(203)
一个 BIOS Contro__break 应用的例子	(206)
DOS CONTROL__BREAK 函数	(208)
IntBInstall 和 IntDBRemove	(208)
DOS 的 Control__Break 应用的一个示例	(210)
严重错误函数	(211)
IntCEInstall 和 IntCERemove	(212)
严重错误应用示范	(214)
第十章 内存驻留程序	(216)
TSR 函数	(216)
TsrInstall 和 TsrIndos	(221)
一个控制打印机的内存驻留程序	(229)
第十一章 扩展内存接口函数	(235)
扩展内存概述	(235)
扩展内存函数	(237)
EmsInstalled	(243)
EmsVersion	(243)
EmsPageAvail	(244)
EmsAlloc 和 EmsFree	(246)
EmsMap	(248)
EmsSave 和 EmsRestore	(249)
EmsErrorMsg	(251)
从一个 Turbo C 程序中使用扩展内存	(251)
第十二章 实用函数	(255)
字符串函数	(266)

UtyAllBlank	(266)
Utyblank	(267)
UtyDump	(267)
UtyRepeat	(268)
UtySetFar	(268)
UtyRightTrim	(269)
日期和时间函数	(270)
UtyClockCount	(270)
UtyDateString 和 UtyTimeString	(272)
UtyPackDate 和 UtyUnPackDate	(275)
UtyPause	(277)
UtyTime	(277)
数值函数	(278)
UtyRound	(278)
UtyRound5	(279)
其他函数	(279)
UtyBiosPutS	(279)
UtyEnabled	(280)
UtyMachineType	(280)
UtyQuit	(282)
附录 A 按字母顺序函数汇总表	(284)
Buf...: 显示器缓冲区函数	(284)
BufFree	(284)
BufGet	(284)
BufInit	(285)
BufNextLine	(285)
BufShow	(285)
Ems...: 扩展内存函数	(285)
EmsAlloc	(285)
EmsErrorMsg	(285)
EmsFree	(286)
EmsInstalled	(286)
EmsMap	(286)
EmsPagesAvail	(286)
EmsRestore	(286)
EmsSave	(286)
EmsVersion	(287)
Fil...: 文件管理功能	(287)

FilDelVdid	(287)
FilGetTime	(287)
FilGetVolid	(287)
FilHandle	(288)
FilOpen	(288)
FilRenameDir	(288)
FilSetTime	(288)
FilSetVolid	(289)
Gra...: 图形函数	(289)
GraBox	(289)
GraCircle	(289)
GraClear	(290)
GraFicl	(290)
GraGetCoord	(290)
GraGetPoint	(290)
GraInit	(290)
GraLine	(291)
GraPoint	(291)
GraQuit	(291)
GraSetCoord	(291)
GraTypes	(291)
Int...: 中断处理函数	(292)
IntBBinstall	(292)
IntBBRemove	(292)
IntCEInstall	(292)
IntCERemove	(292)
IntClockInstall	(293)
IntClockRemove	(293)
IntDBInstall	(293)
IntDBRemove	(293)
IOC...: 控制台 I / O 控制函数	(293)
IocCookMode	(293)
IocRewMode	(293)
Kbd...: 键盘函数	(294)
KbdGetC	(294)
KbdGetShift	(294)
KbdSetShift	(294)
Prt...: 打印机函数	(294)
PrtInit	(294)

PrtNewPage	(295)
PrtPosition	(295)
PrtPutC	(295)
PrtPuts	(295)
PrtQueCancel	(295)
PrtQueCancelAll	(295)
PrtQueState	(296)
PrtQueSubmit	(296)
PrtReady	(296)
PrtSwap	(296)
PrtTimeput	(296)
Scr...: 视频函数	(296)
ScrClear	(297)
ScrGetCur	(297)
ScrGetMode	(297)
ScrGetS	(297)
ScrGetStyle	(298)
ScrPop	(298)
ScrPush	(298)
ScrPutAttr	(298)
ScrPutBox	(299)
ScrPutS	(299)
ScrPutWindow	(299)
ScrReedWindow	(299)
ScrSetLur	(300)
ScrSetMode	(300)
Tsr...: 供常驻内存程序使用的函数	(300)
TsrInDos	(300)
TsrInstall	(300)
Uty...: 实用函数	(301)
UtyAllBlank	(301)
UtyBiosPutS	(301)
UtyBlank	(301)
UtyClockCount	(301)
UtyDateString	(302)
UtyDump	(302)
UtyEnabled	(302)
UtyMachineType	(302)
UtyPackDate	(302)

UtyPause	(303)
UtyQuit	(303)
UtyRepeat	(303)
UtyRightTrim	(303)
UtyRound5	(303)
UtySetFar	(303)
UtyTime	(304)
UtyTimeString	(304)
UtyUnpackDate	(304)
 附录 B 建立库文件	 (305)

前 言

本书用于满足 Turbo C 程序员的两种不同需要。首先,对于应用程序设计人员,书中包含了一个功能强大的软件工具包,可以立即加入到 C 语言程序中,极大地扩充了 Turbo C 函数库。其次,对于系统程序设计人员,提供了完整的源代码程序和详细的说明来介绍 Turbo C 的许多高级功能,并且指导如何最优地使用该种语言进行系统级程序设计。

本书中提供的软件工具经过精选,并专门针对 Turbo C 环境编写。该函数库不是从其它系统简单地移植过来的,而是根据 Turbo C 的特点,并使用了该编译器的许多独特功能开发的。这些函数易于使用,弥补了 Turbo C 运行库的缺陷,并为应用程序提供了实用且重要的服务。

书中明确地区别了函数的使用和这些函数的内部工作两个方面。只对 C 运行库的扩充有兴趣的应用程序员可以安全地跳过内部工作部分。没有必要理解函数的实现再来有效地使用,实际上,这些函数都提供了简单的编程接口。

对于那些对 Turbo C 的高级功能感兴趣的程序员,以及那些想用该语言作为系统程序设计工具的人员,第二章中给出了这些功能的概要,例如汇编语言嵌入、伪变量和中断函数,在本书余下部分讨论的许多的例程及实用程序,还有许多很有意义的方法。

“系统程序设计”有多种含义。有时是指用于编写一个操作系统的算法和数据结构。在本书中,是指与操作系统以及其它低层机器资源之间的接口技术。书中包含了一个具有各种函数的集合,但都是为了一个目的:通过使应用程序能直接存取 MS-DOS 机器的许多软硬件资源来增强 Turbo C 的功能。标准的 C 库功能是非常一般的,它的设计是为了系统间移植的方便,其结果无法利用 MS-DOS 环境下的许多重要特性。Turbo C 中增加了一些与 MS-DOS 相关的函数,而本书中的软件工具则使该方面的功能得到极大增强。

本书的安排

如果用户没有用过 Turbo C,应阅读第一章,以便尽快地使系统运行,并熟悉其特性。如果对系统级程序设计有兴趣,应该接着阅读第二章,因为其中包含了对理解本书余下部分中关于例程内部工作有用的信息。其它各章相对独立,可以按任何顺序学习,但第三章中包含有较多的介绍性内容。读者应从各章的开头开始看起,因为许多重要的概念只在第一次出现时介绍。

第一章 Turbo C 简介

本章的主要目的是帮助用户尽可能快地熟悉 Turbo C 编译器，并运行该编译器，而忽略了软件包中各组成部分所包含的复杂性。当用户安装了 Turbo C，并且编译及运行“hello world”程序后，会发觉该系统提示丰富，极易使用，尤其是集成开发环境。

Turbo C 的一些复杂情况归因于该软件包实际上包含两个不同的开发系统：从 MS-DOS 命令行执行的传统编译器和连接程序(tcc.exe 及 tlink.exe)；以及类似于同 Turbo Pascal 的集成开发环境(tc.exe)。在使用手册中这两个系统总是同时说明的。本章每次说明一个系统，但会帮助用户区别这两个系统。

Turbo C 复杂的另一个原因是大量可用的选项。对于原来使用简单开发环境，比如 BASIC 或 Turbo Pascal 的程序员，文件数量、配置参数，以及文档页数可能开始时显得很吓人。但很快就会发现系统的优点并喜欢其灵活性。而且，当水平有了更多的长进后，会更适应于这些特点。对于那些使用过其它 C 编译器的有经验的 C 程序员，Turbo C 是非常好的一个工具，本章述及的内容将帮助用户以基本的、通常使用的选项集来启动系统，更高级的选项将在以后章节中介绍。

本章中的过程假定用户已经能够编写 C 程序。也许用户有一个其它编译器下工作的程序要移植到 Turbo C 中；也许用户是个初学者，准备作一些练习。在任一情况下，即便是最简单的程序也要求一些规定的步骤，这些步骤对于不同的编译器有很大的不同。因此，本章首先总结这些步骤。

本章第二部分主要描述该编译器的专门特性：与传统语言定义的区别以及与其它常用 C 实现的区别。该部分对于已经熟悉标准 C 语言定义，希望快速浏览 Turbo C 专门特性的有经验 C 程序员特别有用。大多数的这些特性在相当低的机器级下操作；这些特性将在此说明，但详细讨论在第二章高级 Turbo C 程序设计中包括。

建立 TURBO C 环境

以下的两个部分说明安装和运行 Turbo C 这两个步骤：首先，在计算机上安装 Turbo C 系统，第二，使用基本步骤工作，包括编译和运行一个程序。为了能尽快地运行 Turbo C，重点在于说明建立一个基本的、通用的配置。系统开始工作以后可以加入另外的工具和选项，这在本书后面都有相应的说明。

安装 TURBO C

将 Turbo C 一号盘插入 A 驱动器，转到 A:提示符下，并键入命令：

Install

以后按照屏幕上的提示换插磁盘即可安装 Turbo C。

开发一个程序

系统安装好以后，就可学习使用 Turbo C 的基本技术。本节介绍编译和运行一个例子 C 程序，来说明该编译器的基本特征及使用过程。虽然 Turbo C 编译程序包含许多的选项和命令，此处总结的是一些最重要，并且是学习本系统的良好起点。程序先使用命令行方式编译，然后使用集成化环境。因为大多数 C 程序包含多于一个的源文件(模块化编译是 C 的重要特点)，例子程序包含两个文件：图 1.1 中所示的 main.c，和图 1.2 中所示的 utility.c。main.c 文件代表一些应用的基本文件，它调用文件 utility.c 中的函数 RepeatStr

```
#include <stdio.h>

char *RepeatStr (int Number, char Ch);

void main ()
{
    printf ("This is the main application.\n");
    printf ("Sign your name on the dotted line ");
    printf (RepeatStr (40, "."));

    /* end main */
}
```

图 1.1 一个 C 的主程序例子

```
char *RepeatStr (int Number, char Ch)
{
    This function returns a pointer to a null-terminated static string
    consisting of 'Number' copies of character 'Ch'. The string is changed with
    subsequent calls to 'RepeatStr'. Maximum value of 'Number' is 255.

    static char Buf [255];
    register int I = 0;

    while (Number-- > 0 && I < 255)
        Buf [I++] = Ch;
    Buf [I] = '\0';
    return Buf;

    /* end RepeatStr */
}
```

图 1.2 一个 C 模块例子

命令行系统

使用传统的命令行方式开发程序过程如下：

1. 建立源文件
 2. 编译并连接程序
- 详细说明如下。

建立源文件

使用最熟悉的正文编译器输入并保存两个源文件 `main.c` 和 `utility.c`。对于开发大型的 C 应用程序，最好使用多窗口、灵活的编译器，比如用 `Brief`，这样的编译器能很方便地在文件之间删除和复制代码。对于双软驱系统，应转到 **B** 提示符下，因为此处包含用户文件，对于硬盘系统，可以转入任一用户希望保存 C 文件的目录下，假定符合前面所述的路径和配置文件要求。

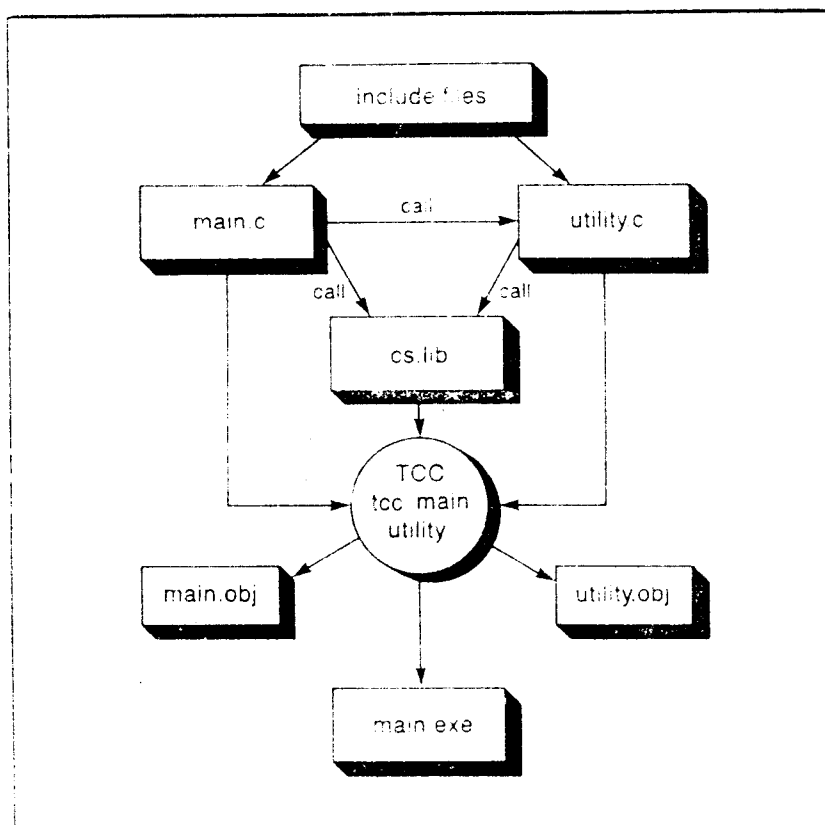


图 1.3 使用命令行系统开发实例程序

编译并连接程序

使用 `tc.exe` 命令行编译程序编译和连接是很简单的。该程序不仅省去了单独的编译和连接步骤，而且自动地结合了适当的起始代码(`c0s.obj`)和库例程(`emu.lib`, `maths.lib` 和 `cs.lib`)。如果使用双软驱系统，前面创建的运行盘应在驱动器 **A** 中，用户应转入 **B** 驱动器。对于硬盘系统，用户应继续转入存放这两个源文件的目录中。对任一种设置，在 DOS 提示符下键入下面的命令：

```
tcc main.c utility.c
```

编译程序将编译这两个 C 程序，然后自动地启动连接程序，传递以正确的命令行。

结果文件是 main.obj, utility.obj 和 main.exe。现在键入:

```
main
```

注意每次运行 tcc 程序时, 它都自动地重编译两个 C 文件, 即使只修改了其中之一。使用命令行系统开发实例程序的过程总结于图 1.3 中。

集成开发环境

本节介绍 Turbo C 集成环境, 给出了开发实例程序的基本步骤, 然后是学习和使用该系统的一些基本要点。

要开发一个实例程序, 有以下步骤:

- 启动集成环境
- 在编译器中建立源文件
- 产生.exe 文件
- 运行程序

启动集成环境

键入 tc 启动 Turbo C 的集成开发环境。

在编译器中建立源文件

启动 Turbo C 后, 按 Alt-E 进入编译器。该编译器所用的命令几乎等同于 Turbo Pascal 和 Sidekick, 类似于 Wordstar。若用户未用过这些编译器, 见附录 A 中 Turbo C 的参数指南, 那里包含了所有的指令, 首先键入图 1.1 中的源程序。输入完成后, 按 Alt-F 进入文件菜单, 然后按 S 保存该文件到盘上。系统询问是否替换默认名 noname.c, 此时键入 main.c 并按回车。

存入文件后, 仍处于文件菜单。此时按 N 清掉编译器准备输入一个新的文件。用同样的方法键入图 1.2 中所列的文件, 以名 utility.c 保存, 然后又清掉编译器。

在程序编译以前还有一个文件需要建立。如果此时调回 main.c 执行编译和连接, Turbo C 无法知道存在另一个文件与该程序相关, 报告以下信息:

```
Undefined symbol '-RepeatStr' in module main.c
```

因此, 用户必须建立一个工程(project)文件来通知 Turbo C 组成该程序的单个模块(工程文件类似于 made 文件。)建立该文件很简单, 只有两行组成:

```
main.c  
utility.c
```

以文件名 main.prj 保存该文件。然后按 Alt-P, 再按 P, 然后键入 main 使 Turbo C 知道该工程文件名, 注意如果整个程序仅由一个文件组成时不需要建立工程文件。

产生.exe 文件

建立了工程文件后, 可在任何时候编译和连接整个系统, 通过简单地按 Alt-C 调出编译菜单, 按 M(或按 F9)产生.exe 文件。此时生成可执行文件 main.exe。

注意使用了工程文件后, 哪个文件当前处于编译器中是无关紧要的, 当要求 Turbo C 根据一个工程文件生成一个程序时, 它自动地存取所有需要的文件, 并输出.obj 和.exe 文

件到磁盘上。(不同于 Turbo Pascal, Turbo C 不编译到内存, 而总是写 .obj 和 .exe 文件到磁盘上。)

Turbo C 的工程机制不仅自动地生成多文件的应用程序, 而且产生最新的 .exe 文件。类似于 make 工具, 它检查文件修改的日期, 不重编译未修改的源代码文件。本书为所有的例子程序包含工程文件, 并对该工具的另外一些特征作了说明。

运行程序

要运行程序, 在系统中按 Alt-R。注意 Alt-R 命令还自动完成将可执行文件变成最新版的编译和连接过程。因此不耐烦的用户可直接按 Alt-R, 而不先用前述命令生成 .exe 文件。

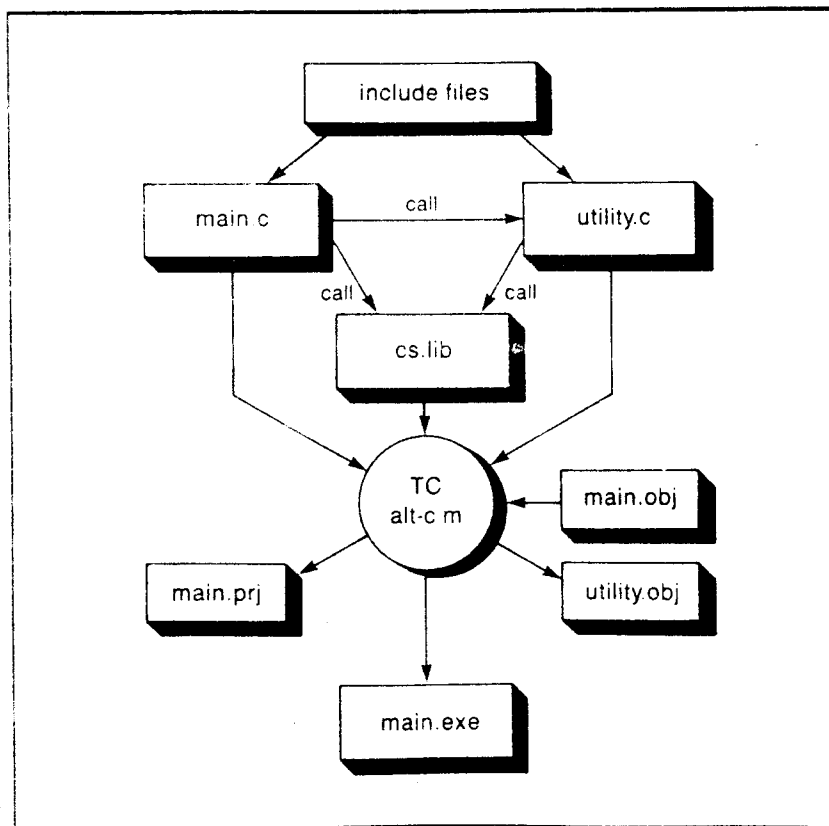


图 1.4 使用集成环境开发实例程序

学习集成环境的要点

经过了前面一节的基本步骤以后, 学习集成环境其它特征的最好方法是经常使用帮助功能(按 F1 启动)来熟悉窗口和菜单。通常, 在线帮助提供的内容与手册上的内容相同, 更方便之处是通过选择相应的关键词来查询所需内容。要记住的一个重要命令是 Alt-F1, 它使你获得当前屏幕的上一屏幕。使用 F1 和 Alt-F1 用户可以查看各页的内

容。

Turbo C 集成化环境的另一个方便特征是可通过热键执行命令。这些键或是功能键，或是 Alt 键组合。与当前上下文相关的功能热键表位于屏幕的底行，若按住 Alt 几秒钟，则会出现相关的 Alt 键组合。并不是所有的有用键都出现于快速参考行中。

集成系统中其它的可用选项在本书高级技术部分讨论。图 1.4 中总结了在集成环境中产生可执行文件的步骤。

同时使用两种环境

虽然命令行编译器和集成环境是独立的系统，但有时在开发一个程序的不同阶段使用相应的一个是很有优点的。

例如，在多窗口的编辑器中建立一个大型应用是很方便的，允许在文件之间删除和复制，同时浏览几个文件。文件编辑完成并存盘后，可用命令行系统编译(使用 Turbo C 提供的 make 程序)。如果存在许多编译错误和 / 或需要调整一些运行特征，可以用集成环境方便地消除错误及优化运行特征，此时集成环境的优点是快速修改 / 编译 / 运行循环，以及在源代码文件中自动定位编译错误的调试功能。(用户可能已注意到当键入源程序出错时系统的报错功能。)

TURBO C 的专门特性

1978 年 Kerighan 和 Ritchie 编写的《C 语言程序设计》说明了 C 语言的最小标准。几乎所有的 C 编译程序，包括 Turbo C 完全支持该分类标准。但 Turbo C 跟大多数最新的 C 编译程序一样，给该基本的语言定义增加了许多的新功能。大多数的这些功能与 ANSI 开发的联合标准一致。另外的增加功能不是 ANSI 标准的一部分，或为 Turbo C 独有，或出现于另外的最新编译程序中。

这些特性包括新的数据类型，符号元素，预处理命令，以及一组标准库函数。多数的新编译程序，例如 Microsoft C 4.0 和 5.0，实现了一些 ANSI 特性。Turbo C 则包含了其中的许多。

本节不想说明所有的 ANSI 标准特性，因为太多，无法在此列出，而且其中许多是相当普通且一般书中都有述及。这里只集中讨论较不普遍的 ANSI 特性以及非 ANSI 标准的特性，它们在 Turbo C 中出现。目的是为已经使用其它编译器工作过的 C 程序员快速浏览这些特性，或帮助未曾注意到这些内容的初学者。非 ANSI 标准的特性将被指出。

函数原型

Turbo C 完全以 ANSI 标准实现函数原型。注意在 Turbo C 中使用函数原型是可选的，用户也可以将原型方法与传统方法结合使用。但要充分利用这种新格式，应使用下面两条规则来说明和定义函数：

1. 所有的函数先说明后调用，不但说明返回类型，而且说明每个参数的类型，例如：

```
int prtposition(char * string, int Row, int col);
```