

# Turbo C

## 2.0

### 实用大全

常玉龙  
葛本年  
张再良  
孙和达

编写  
主审

北京航空航天大学出版社

# Turbo C 2.0

# 实用大全

常玉龙  
葛本年      编写  
张再良  
孙和达      主审

北京航空航天大学出版社

(京)新登字 166 号

## 内 容 摘 要

本书共五部分。第一部分: Turbo C 集成环境的功能和用法;组成 Turbo C 屏幕的要素;如何用 Turbo C 进行编辑、编译、调试、连接和运行; Turbo C 编辑器、调试器和多文件管理的工程文件的使用方法;第二、第三部分: Turbo C 语言基础和高级的 Turbo C 程序设计技术;包括指针、结构和联合的常用的各种数据结构;循环、条件等各种语句;屏幕和文件处理;与汇编语言的接口等。第四部分: Turbo C 库函数和全局变量的功能、用法和调用实例。第五部分为总附录: TCC 命令行编辑器;实用程序等。可供计算机工程人员及大专院校师生使用。

- 书 名: Turbo C 2.0 实用大全  
Turbo C 2.0 SHIYONG DAQUAN
- 作 者: 常玉龙 葛本年 张再良 编写 孙和达 主编
- 责任编辑: 许传安
- 出 版: 北京航空航天大学出版社  
(北京市学院路 37 号, 邮编 100083, 发行部电话 2015720)
- 印 刷: 北京朝阳科普印刷厂
- 总 发 行: 北京航空航天大学出版社
- 发 行: 新华书店总店科技发行所
- 销 售: 各地书店
- 开 本: 787×1092 1/16
- 印 张: 63.75 字 数: 1632 千字
- 版 次: 1994 年 9 月第一版, 1994 年 9 月第一次印刷
- 印 数: 5000 册
- 定 价: 60.00 元
- 书 号: ISBN 7-81012-508-7/TP·125

# 目 录

## 绪 论

0.1 两种 Turbo C 环境 .....	1
0.2 设置自己的任务环境 .....	2
0.3 Turbo C 编译器 .....	2
0.4 Turbo C 工程制作实用程序 .....	3
0.5 低级功能的支持 .....	3
0.6 配置 Turbo C .....	4
0.7 存储模式 .....	4

## 第一部分 掌握 Turbo C

### 第一章 Turbo C 集成开发环境

1.1 TC 的使用 .....	7
1.1.1 TC 命令行开关 .....	8
1.1.2 TC 内部命令的使用 .....	9
1.1.3 TC 热键 .....	10
1.1.4 菜单结构 .....	11
1.1.5 主菜单 .....	11
1.1.6 功能键提示行 .....	12
1.1.7 编辑窗口 .....	12
1.1.8 消息窗口 .....	15
1.1.9 监视窗口 .....	15
1.1.10 集成调试器 .....	16
1.2 菜单命令 .....	18
1.2.1 File(文件)菜单 .....	18
1.2.2 Edit(编辑)命令 .....	20
1.2.3 Run(运行)菜单 .....	20
1.2.4 Compile(编译)菜单 .....	22
1.2.5 Project 菜单 .....	23
1.2.6 Options 菜单 .....	25
1.2.7 Debug 菜单 .....	37
1.3 配置和 pick 文件 .....	42
1.3.1 TC 配置文件 .....	42
1.3.2 Pick 表和 Pick 文件 .....	44

### 第二章 使用 Turbo C 编辑器

2.1 编辑器命令 .....	46
-----------------	----

2.2 编辑器激活及文本键入 .....	46
2.3 字符、字和行的删除 .....	47
2.4 移动、拷贝文本和块移动 .....	48
2.5 进一步说明如何移动光标 .....	49
2.6 字符系列的搜索和替换 .....	50
2.7 位置标识的设置和搜索 .....	51
2.8 存储和装入文件 .....	52
2.9 自动缩进 .....	52
2.10 磁盘文件中文本块的移入和移出 .....	52
2.11 对匹配 .....	53
2.12 其它有关命令 .....	53
2.13 命令综述 .....	54
2.14 用文件激活 Turbo C .....	55

### 第三章 Turbo C 调试器

3.1 调试与程序开发 .....	56
3.2 设计示例程序: PLOTEMP.C .....	57
3.3 编写原型程序 .....	58
3.4 使用集成调试工具 .....	60
3.5 跟踪程序的流程 .....	60
3.5.1 跟踪高层的运行 .....	60
3.5.2 跟踪子函数 .....	61
3.6 继续程序的开发 .....	61
3.7 设置断点 .....	64
3.7.1 用 Ctrl-Break 立即中断 .....	65
3.8 计算和修改变量 .....	65
3.8.1 指定显示格式 .....	66
3.8.2 指定值的个数 .....	66
3.8.3 从光标所在位置拷贝 .....	66
3.8.4 查看在别的函数中的变量 .....	66
3.8.5 修改值 .....	67
3.9 通过设置监视项来监视程序运行 .....	70
3.9.1 添加一个监视项 .....	70
3.9.2 观察监视项 .....	70
3.9.3 控制 器窗口 .....	71
3.9.4 编辑和删除监视项 .....	71
3.9.5 寻找一个函数定义 .....	71
3.9.6 查找调用关系 .....	72

3.9.7 多个源文件 .....	72
3.10 预防措施 .....	72
3.11 有系统的软件测试 .....	73
3.11.1 全面测试修改结果 .....	73
3.11.2 仔细观察的部分 .....	73
3.12 完成 PLOTEMP.C .....	74
3.12.1 完成 table_view() .....	74
3.12.2 实现 graph_view() .....	75
3.12.3 save_temps()和 read_temps() .....	77

## 第四章 多文件工程管理

4.1 工程管理程序的使用 .....	79
4.2 出错跟踪 .....	81
4.2.1 终止 MAKE .....	81
4.2.2 多源文件的语法错误 .....	81
4.2.3 保存或删除信息 .....	82
4.3 工程管理程序的功能 .....	82
4.3.1 自身依赖性检查 .....	82
4.4 取代库 .....	83
4.5 工程管理程序的其他功能 .....	83
4.6 生成最终应用程序的集成环境设置 .....	84

## 第二部分 Turbo C 语言基础

## 第五章 C 语言概貌

5.1 预备知识 .....	89
5.1.1 源文件、目标文件和装载模块 .....	89
5.1.2 程序的逻辑和执行流程 .....	90
5.2 基本数据类型 .....	96
5.2.1 C 语言的基本数据类型 .....	96
5.2.2 何处定义数据对象 .....	98
5.3 编写 C 语言表达式和语句 .....	101
5.3.1 表达式和语句 .....	101
5.4 控制类型转换 .....	105
5.5 显式类型转换的使用 .....	107
5.6 使用 C 的宏 .....	107
5.6.1 定义类似对象的宏 .....	108
5.6.2 定义类似函数的宏 .....	110

## 第六章 操作符和表达式

6.1 什么是操作符 .....	115
6.2 单目操作符 .....	116

6.3 双目操作符 .....	116
6.4 三目操作符 .....	117
6.5 标点符号 .....	117
6.6 操作符语义 .....	119
6.6.1 后缀和前缀操作符 .....	119
6.6.2 单目操作符 .....	121
6.6.3 sizeof 操作符 .....	122
6.6.4 乘法类操作符 .....	122
6.6.5 加法类操作符 .....	123
6.6.6 按位移位操作符 .....	123
6.6.7 关系操作符 .....	124
6.6.8 相等类操作符 .....	125
6.6.9 位运算操作符 .....	125
6.6.10 逻辑运算符 .....	126
6.6.11 条件操作符?: .....	127
6.6.12 赋值操作符 .....	127
6.6.13 逗号操作符 .....	128
6.7 高级运算符的使用实例 .....	128
6.7.1 位运算符 .....	128
6.7.2 移位运算符 .....	130
6.7.3 ?:运算符 .....	134
6.7.4 C 语言的简写 .....	136
6.7.5 逗号运算符 .....	136
6.7.6 运算符优先级表 .....	136
6.8 表达式 .....	136

## 第七章 说明

7.1 有关概念 .....	139
7.1.1 对象 .....	139
7.1.2 左值 .....	139
7.1.3 右值 .....	140
7.1.4 类型与存储类 .....	140
7.1.5 作用域 .....	140
7.1.6 可见性 .....	141
7.1.7 生存期 .....	141
7.1.8 编译单元 .....	142
7.1.9 连接 .....	142
7.2 说明的语法 .....	143
7.2.1 暂时定义 .....	143
7.2.2 可能的说明 .....	144
7.3 类型说明 .....	147
7.3.1 外部说明与定义 .....	147
7.3.2 类型指明符 .....	147

7.3.3 类型分类 .....	147	8.5.3 无穷 for 循环的中断 .....	179
7.3.4 基本类型 .....	148	8.5.4 空循环的用法 .....	179
7.3.5 标准转换 .....	151	8.6 while 循环语句 .....	179
7.3.6 特殊的 char,int 与 enum 间 的转换 .....	152	8.7 do while 循环 .....	181
7.3.7 初始化 .....	152	8.8 循环嵌套 .....	182
7.4 简单说明 .....	154	8.9 循环中断 .....	184
7.5 存储类指明符 .....	155	8.10 continue 语句 .....	186
7.5.1 存储类指明符 auto 的使用 .....	155	8.11 goto 语句 .....	187
7.5.2 存储类指明符 extern 的使用 .....	155		
7.5.3 存储类指明符 register 的使用 .....	155	<b>第九章 函 数</b>	
7.5.4 存储类指明符 static 的使用 .....	155	9.1 函数的初步概念 .....	189
7.5.5 存储类指明符 typedef 的使用 .....	155	9.1.1 说明与定义 .....	189
7.6 修饰符 .....	156	9.1.2 说明与原型 .....	189
7.6.1 const 修饰符 .....	156	9.1.3 定 义 .....	190
7.6.2 中断函数修饰符 .....	157	9.1.4 形参说明 .....	191
7.6.3 volatile 修饰符 .....	157	9.1.5 函数调用与参数转换 .....	191
7.6.4 cdel 与 pascal 修饰符 .....	158	9.2 return 语句 .....	192
7.6.5 指针修饰符 .....	159	9.2.1 从一个函数中返回 .....	192
7.6.6 函数类型修饰符 .....	159	9.2.2 返回值 .....	193
7.7 复杂说明与说明符 .....	159	9.2.3 函数返回非整型值 .....	195
		9.3 有关函数原型的进一步说明 .....	198
<b>第八章 程序控制语句</b>		9.3.1 参数不匹配 .....	198
8.1 程序控制语句的语法 .....	161	9.3.2 使用头文件 .....	199
8.1.1 带标号语句 .....	162	9.3.3 无任何参数的函数原型 .....	199
8.1.2 复合语句 .....	163	9.3.4 有关旧式 C 程序 .....	199
8.1.3 表达式语句 .....	163	9.4 作用域规则 .....	200
8.1.4 选择语句 .....	163	9.4.1 局部变量 .....	200
8.1.5 循环语句 .....	164	9.4.2 形式参数 .....	202
8.1.6 跳转语句 .....	165	9.4.3 全局变量 .....	202
8.2 if 语句 .....	166	9.4.4 有关作用域的最后一个例子 .....	203
8.2.1 else 语句的用法 .....	168	9.5 有关函数的参数和变量的 更详尽说明 .....	204
8.2.2 if—else—if 阶梯的用法 .....	168	9.5.1 赋值调用和赋地址调用 .....	204
8.2.3 条件表达式 .....	169	9.5.2 一个赋地址调用的建立 .....	205
8.2.4 if 语句的嵌套结构 .....	170	9.5.3 数组与函数调用 .....	206
8.3 switch 语句 .....	170	9.6 argc,argv 和 env —— main 中的参数 .....	209
8.3.1 default 语句的用法 .....	172	9.7 从 main() 中返回值 .....	211
8.3.2 break 语句的用法 .....	173	9.8 递 归 .....	212
8.3.3 switch 语句的嵌套结构 .....	174	9.9 参数说明的传统形式和现代 形式的比较 .....	214
8.4 循 环 .....	176	9.10 对一些影响函数的效率和实用 性问题的讨论 .....	214
8.5 for 循环 .....	176	9.10.1 参数和通用函数 .....	214
8.5.1 for 循环的灵活用法 .....	177		
8.5.2 无穷的 for 循环 .....	179		

9.10.2 效率	215
9.11 库函数	215
9.12 改变程序的执行流程	218
9.12.1 使用 exit()和 abort()函数	218
9.12.2 使用 system(),exec()和 spawn()函数	219
9.13 使用可变参数表	220
9.13.1 设计可变参数表	221
9.13.2 使用 va()函数	221

## 第十章 指针

10.1 指针的语法规则	226
10.1.1 什么是指针	226
10.1.2 指针说明	227
10.1.3 指针与常量	227
10.1.4 指针算术运算	228
10.1.5 指针转换	229
10.2 指针是地址	229
10.3 指针变量	229
10.4 指针操作符	230
10.5 指针表达式	231
10.5.1 指针赋值	231
10.5.2 指针运算	231
10.5.3 指针比较	233
10.6 指针和数组	233
10.6.1 索引指针	234
10.6.2 指针和字符串	234
10.6.3 如何得到一个数组元素的地址	235
10.6.4 指针数组	236
10.6.5 一个使用数组和指针的实例	237
10.7 指针的指针	241
10.8 指针的初始化	242
10.9 指针的一些问题	243
10.9.1 使用C语言的间接操作符和取地址操作符	244
10.9.2 使用数组和串	246
10.10 使用指向函数的指针	252
10.10.1 指向函数的指针声明和初始化	252
10.10.2 利用指针引用某调用函数	253
10.11 在动态内存中使用指针	256
10.11.1 语言程序和动态内存	257
10.11.2 使用动态存储	258

## 第十一章 数组、结构、位域、联合和枚举

11.1 高级数据类型的语法规则	265
11.1.1 数组	265
11.1.2 结构	265
11.1.3 位域	269
11.1.4 联合	269
11.1.5 枚举	270
11.2 数组	271
11.2.1 一维数组	272
11.2.2 字符串	273
11.2.3 二维数组	278
11.2.4 多维数组	280
11.2.5 数组初始化	280
11.2.6 一个水下搜索游戏	282
11.3 结构	285
11.3.1 访问结构元素	286
11.3.2 结构数组	287
11.3.3 结构赋值	294
11.3.4 将结构传递给函数	294
11.3.5 结构指针	296
11.3.6 结构内部的数组和结构	299
11.4 位域	300
11.5 联合(union)	303
11.6 枚举	306
11.7 使用 sizeof 来确保可移植性	308
11.8 typedef	309

## 第十二章 Turbo C 预处理程序指令

12.1 空指令#	313
12.2 #define 与 #undef 指令	313
12.2.1 简单的 #define 宏	313
12.2.2 #undef 指令	314
12.2.3 -D 与 -U 选择项	315
12.2.4 关键字与保护字	315
12.2.5 带参宏	315
12.3 文件包含指令 #include	317
12.3.1 <头名>形式的头文件搜索	318
12.3.2 “头名”形式的头文件搜索	318
12.4 条件编译	318
12.4.1 #if、#elif、#else 和 #endif 条件指令	318

12.4.2	defined 运算符	319	13.7.9	getw()和 putw()函数的使用	345
12.4.3	#ifdef 和 #ifndef 条件指令	319	13.7.10	fgets()和 fputs()函数	345
12.5	#line 行控制指令	320	13.7.11	fread()和 fwrite()函数	345
12.6	#error 指令	321	13.7.12	fseek()函数和随机访问 I/O	347
12.7	#pragma 指令	321	13.7.13	标准流	349
12.7.1	#pragma argsused	322	13.7.14	fprintf()和 fscan()函数	350
12.7.2	#pragma exit 与 #pragma startup	322	13.7.15	删除文件	352
12.7.3	#pragma inline	322	13.8	非缓冲型 I/O——UNIX 型文件系统	352
12.7.4	#pragma option	323	13.8.1	open(), creat()和 close()函数	353
12.7.5	#pragma saveregs	324	13.8.2	read()和 write()函数	354
12.7.6	#pragma warn	324	13.8.3	unlink()函数	356
12.8	预定义的宏	325	13.8.4	随机访问文件和 lseek()函数	356
<b>第三部分 高级 C 程序设计技巧</b>			13.9	理解 I/O 概念	357
<b>第十三章 文件输入输出</b>			13.9.1	文件与设备	358
13.1	两个预处理指令	329	13.9.2	文件与流	359
13.1.1	#define 指令	329	13.9.3	文本流和二进制流	360
13.1.2	#include 指令	331	13.10	利用标准流进行 I/O	361
13.2	文件与流	331	13.10.1	使用格式化 I/O 函数	361
13.3	流(streams)	331	13.10.2	scanf()函数	365
13.3.1	文本流	331	13.10.3	prints()函数	368
13.3.2	二进制流	332	13.10.4	使用字符 I/O 函数	370
13.3.3	文件	332	13.11	使用文件控制函数	374
13.4	概念和实际	332	13.11.1	开文件、关文件和控制文件	374
13.5	控制台 I/O	333	13.11.2	控制文件缓冲区	377
13.5.1	字符读写	333	13.12	使用直接文件 I/O 函数	378
13.5.2	字符串读写	334	13.12.1	理解直接 I/O 概念	379
13.6	控制台格式化 I/O	334	13.12.2	读写直接文件	380
13.6.1	printf()函数	335	13.13	使用文件定位函数	385
13.6.2	scanf()函数	336	13.13.1	得到当前文件位置	385
13.7	缓冲型 I/O 系统(ANSI 型 I/O 系统)	339	13.13.2	建立一个新文件位置	386
13.7.1	文件指针	339	13.14	处理文件 I/O 错误	388
13.7.2	打开文件	339	13.14.1	查出文件 I/O 错误	388
13.7.3	写字符	341	13.14.2	显示和清除文件 I/O 错误	388
13.7.4	读字符	341	<b>第十四章 屏幕文本和图形程序设计</b>		
13.7.5	feof()的使用	342	14.1	图形系统和要素	390
13.7.6	关闭文件	342	14.1.1	视频模式	390
13.7.7	ferror()和 rewind()函数	342	14.1.2	窗口和视区	390
13.7.8	fopen(),getc(),putc()和 fclose()函数的用法	343	14.1.3	在文本模式下编程	391
			14.1.4	在图形模式下编程	396
			14.2	Turbo C 图形程序设计	405
			14.2.1	基本正文模式函数	406



14.2.2 Turbo C 的图形子系统简介 .....	414
14.3 IBM/PC 的文本方式 .....	422
14.3.1 PC 显示器适配器和屏幕 .....	422
14.3.2 视频缓冲区 I/O .....	423
14.4 控制文本屏幕 .....	424
14.4.1 使用文本方式控制函数 .....	424
14.4.2 使用直接控制台 I/O 以获 得高性能 .....	427
14.5 使用窗口函数 .....	427
14.6 了解 IBM-PC 的图形方式 .....	431
14.6.1 象素点与调色板 .....	431
14.6.2 控制图形屏幕 .....	433
14.7 介绍 BGI 图形库 .....	434
14.7.1 使用画图和填充函数 .....	434
14.7.2 控制屏幕和视口 .....	438
14.8 在图形方式下显示文本 .....	439
14.8.1 BGI 字库 .....	439
14.8.2 使用图形方式下的文本函数 .....	440

## 第十五章 存储模式

15.1 80×86 的体系结构 .....	443
15.1.1 段 (Segment)、节 (Paragraph) 以及偏移 地址 (Offset) .....	443
15.1.2 CPU 的地址寄存器 .....	445
15.2 near 指针、far 指针和 huge 指针 .....	446
15.2.1 选择想要的指针大小 .....	446
15.2.2 near、far 和 huge 说明符 .....	448
15.3 六个 Turbo C 存储模式 .....	450
15.3.1 决定使用哪种存储模式 .....	450
15.3.2 以混合模式编程 .....	451
15.4 创建 COM 型的可执行程序文件 .....	453
15.4.1 使用 COM 文件 .....	453

## 第十六章 与汇编语言的接口

16.1 混合语言程序设计 .....	456
16.1.1 参数传递顺序 .....	456
16.2 建立从 Turbo C 对 .ASM 的调用 .....	458
16.2.1 简化的段指令 .....	458
16.2.2 标准段指令 .....	459
16.2.3 定义数据常量和变量 .....	459
16.2.4 定义全局和外部标识符 .....	460
16.3 建立从 .ASM 中对 Turbo C 的调用 .....	461

16.3.1 引用函数 .....	461
16.3.2 引用数据 .....	461
16.4 定义汇编语言过程 .....	462
16.4.1 传递参数 .....	462
16.4.2 处理返回值 .....	463
16.5 寄存器约定 .....	466
16.6 从 .ASM 过程中调用 C 函数 .....	466
16.7 伪变量、嵌入汇编和中断函数 .....	468
16.7.1 伪变量 .....	468
16.7.2 嵌入汇编语言 .....	470
16.7.3 中断函数 .....	475
16.8 使用直接插入 (inline) 汇编语言 .....	477
16.8.1 直接插入式汇编环境 .....	478
16.8.2 使用 asm 关键字 .....	478
16.9 与汇编语言例程的接口 .....	481
16.9.1 在 C 程序里调用汇编例程 .....	481
16.9.2 在汇编例程中调用 C 函数 .....	490
16.10 使用中断功能 .....	497
16.10.1 80×86 的中断结构 .....	497
16.10.2 使用 Borland 的中断接口 .....	498
16.11 使用中断处理程序 .....	501
16.11.1 声明中断处理程序函数 .....	501
16.11.2 实现一个时钟中断处理程序 .....	504

## 第四部分 库函数和全局变量参考

### 第十七章 Turbo C 标准库函数

函数名 函数功能描述 .....	511
abort 异常终止—进程 .....	511
abs 返回整数的绝对值 .....	512
absread 读磁盘的绝对扇区 .....	512
abswrite 写磁盘绝对扇区 .....	513
access 确定文件的存取权限 .....	514
acos 计算反余弦值 .....	515
allocmem 分配 DOS 内存 .....	515
arc 画圆弧 .....	516
asctime 转换日期和时间为其对应的 ASCII 码 .....	517
asin 反正弦函数 .....	518
assert 条件终止函数 .....	519
atan 反正切函数 .....	519
atan2 计算 y/x 的反正切值 .....	520
atexit 定义终止函数 .....	521

atof 将字符串转换成浮点数 .....	521	cprintf 格式化并输出数据至屏幕 .....	559
atoi 把字符串转换成整数 .....	522	cputs 输出一字符串至屏幕 .....	560
atol 把字符串转换成整型 .....	523	_creat 创建一个新文件或重写一个已存在的文件 .....	561
bar 画二维条形图 .....	523	creat 创建一个新文件或重写一个已存在的文件 .....	562
bar3d 画一个三维条形图 .....	525	creatnew 创建新文件 .....	563
bdos DOS 系统调用 .....	526	creattemp 创建一个文件名唯一的文件 .....	564
bdosptr DOS 系统调用 .....	527	cscanf 从控制台执行格式化输入 .....	565
bioscom I/O 通信 .....	528	ctime 把日期和时间转化为对应的字符串 .....	565
biosdisk 调用 BIOS 磁盘驱动程序 .....	530	ctrlbrk 设置 ctrl-break 处理程序 .....	566
biosequip 检查设备 .....	532	delay 暂停 .....	567
bioskey 调用 BIOS 的键盘接口 .....	533	delline 在文本窗口中删去一行 .....	567
biosmemory 返回内存大小 .....	535	detectgraph 检测硬件并确定应使用何种图形驱动程序和图形模式 .....	568
biosprint 调用 BIOS 的打印机 I/O 接口 .....	535	difftime 计算两个时刻之间的时间差 .....	571
biostime 读取或设置 BIOS 时钟 .....	536	disable 屏蔽中断 .....	571
brk 改变数据段内存分配 .....	537	div 将两个整数相除, 返回商和余数 .....	572
bserch 数组的二分法搜索 .....	538	dosexterr 获取扩展错误信息 .....	573
cabs 计算复数的模 .....	539	dostounix 把日期和时间转换成 UNIX 格式 .....	574
calloc 分配内存 .....	540	drawpoly 绘制多边形 .....	574
ceil 舍入 .....	540	dup 复制文件句柄 .....	576
cgets 读字符串 .....	541	dup2 将一个文件句柄(oldhandle)复制到一个已有的文件句柄(newhandle) .....	577
chdir 改变当前目录 .....	542	ecvt 把浮点数转换为字符串 .....	578
_chmod 改变文件的存取权限 .....	543	ellipse 绘制椭圆 .....	579
chmod 改变文件存取权限 .....	545	_emit_ 将文字值直接插入源程序中 .....	580
chsize 修改文件长度 .....	545	enable 开硬件中断 .....	581
circle 画圆 .....	546	eof 检测文件是否结束 .....	583
_clear87 清除浮点状态字 .....	547	_exit 终止程序 .....	587
cleardevice 清图形屏幕 .....	548	exit 终止程序 .....	588
clearerr 复位错误标志 .....	549	exp 计算 e 的 x 次方 .....	588
clearviewport 清除当前图形窗口 .....	550	fabs 返回浮点数的绝对值 .....	589
clock 测定运行时间 .....	551	farcalloc 从远程堆中分配内存 .....	589
_close 关闭文件 .....	552	farcoreleft 返回远程堆中未使用内存的大小 .....	590
close 关闭文件 .....	552	farfree 从远程堆中释放一块已分配内存 .....	591
closegraph 关闭图形系统 .....	553	farmalloc 从远堆中分配内存 .....	592
clreol 清除从当前光标位置到行尾的字符 .....	554	farrealloc 调整远堆中的已分配块 .....	593
clrscr 清除文本窗口, 并把光标放在左上角 .....	555		
_control87 处理浮点控制字 .....	556		
coreleft 返回尚未使用的内存(RAM)大小 .....	557		
cos 计算余弦值 .....	557		
cosh 计算双曲余弦值 .....	558		
country 读取与特定国家有关的格式 .....	558		

fclose 关闭一个流 .....	593	fwrite 把参数写入流中 .....	630
fcloseall 关闭打开流 .....	594	gcvt 把浮点数转换为字符串 .....	630
fcvt 将浮点数转换为字符串 .....	595	geninterrupt 产生软中断 .....	631
fdopen 把流与一个文件句柄相联 .....	595	getarccorrd 取得最后一次调用 arc 的 坐标 .....	632
feof 检测流上的文件结束标志 .....	597	getaspectratio 返回当前图形模式的纵 横比 .....	634
ferror 检测流上的错误 .....	597	getbkcolor 返回当前背景颜色 .....	635
fflush 刷新一个流 .....	598	getc 从流中取字符 .....	636
fgetc 从流中读取字符 .....	599	getcbrk 获取 control-break 状态 .....	637
fgetchar 从流中读取字符 .....	600	getch 从键盘无回显地读取一字符 .....	637
fgetpos 取得当前文件指针 .....	600	getchar 从 stdin 流中读取一个字符 .....	638
fgets 从流中读取一字符串 .....	601	getche 从键盘并回显地读取一字符 .....	638
filelength 取文件长度 .....	602	getcolor 返回当前绘图颜色 .....	639
fileno 取得文件句柄 .....	602	getcurdir 读取指定驱动器的当前目录 .....	640
filleshape 画椭圆饼 .....	603	getcwd 读取当前目录 .....	641
fillpoly 画多边形 .....	604	getdate 读取系统日期 .....	642
findfirst 查找第一个匹配文件 .....	605	getdefaultpalette 返回缺省调色板信息 .....	642
findnext 查找下一个匹配文件 .....	607	getdfree 读取磁盘空闲空间 .....	643
floodfill 填充区域 .....	608	getdisk 读取当前磁盘驱动器号 .....	644
floor 下舍入 .....	609	getdrivername 返回指向当前图形驱动程序名字 的指针 .....	645
flushall 刷新所有流 .....	610	getdta 读取磁盘传输地址 .....	646
fmod 计算 x/y 的余数 .....	610	getenv 读取环境变量的当前值 .....	647
fnmerge 建立文件路径 .....	611	getfat 读取指定驱动器的 FAT 信息 .....	647
fnsplit 分解完整的路径名 .....	611	getfatd 读取驱动器 FAT 信息 .....	648
fopen 打开一个流 .....	613	getfillpattern 将用户定义的填充模式拷贝 到内存 .....	649
FP_OFF 获取远地址偏移量 .....	614	getfillsettings 取得当前填充模式和填充颜色的 有关信息 .....	651
_fpreset 重新初始化浮点数学包 .....	615	getftime 读取文件日期和时间 .....	653
fprintf 传送输出到一个流中 .....	617	getgraphmode 返回当前图形模式 .....	654
FP_SEG 获取远地址段值 .....	617	getimage 将指定区域的位图象存入 内存 .....	655
fputc 送一个字符到一个流中 .....	618	getline settings 读取当前线型、模式和 宽度 .....	657
fputchar 送一个字符到标准输出 .....	619	getmaxcolor 返回可选的最大有效颜 色值 .....	659
fputs 送一个字符串到流中 .....	619	getmaxmode 返回当前驱动程序的最大图形模 式号 .....	660
fread 从流中读数据 .....	620	getmaxx 返回屏幕上最大的 x 坐标值 .....	661
free 释放已分配的内存 .....	621	getmaxy 返回屏幕上最大的 y 坐标值 .....	662
freemem 释放先前分配的 DOS 内存 .....	621	getmodename 返回指向含有指定图形	
freopen 把一个新文件同个打开 的流相联 .....	622		
frexp 对双精度数进行科学计数 .....	623		
fscanf 格式化输入 .....	623		
fseek 移动文件指针 .....	624		
fsetpos 定位文件指针 .....	625		
fstat 获取已打开文件的信息 .....	627		
ftell 返回当前文件指针 .....	628		
ftime 把当前时间存入 timeb 结构中 .....	629		

模式名字符串的指针 .....	663	区大小 .....	700
getmoderange 获取图形驱动程序的模式 范围 .....	664	initgraph 初始化图形系统 .....	702
getpalette 返回当前调色板的有关信息 .....	665	inport inp 从端口中读入一个字 .....	705
getpalettesize 返回调色板的颜色数目 .....	667	inportb 从端口中读入一个字节 .....	706
getpass 读入口令 .....	668	insline 在文本窗口插入一空行 .....	706
getpid 读取进程号 .....	669	installuserdriver 安装设备驱动程序到 BGI 设备 驱动程序表中 .....	707
getpixel 读取得像素的颜色 .....	669	installuserfont 安装未嵌入 BGI 系统的字体 文件(.CHR) .....	709
getpsp 读取程序段前缀 .....	671	int86 调用 8086 软中断 .....	710
gets 从标准输入流 stdin 中读取一字 符串 .....	671	int86x 通用 8086 软中断接口 .....	711
gettext 拷贝文本屏幕上的文本拷贝到 内存中 .....	672	intdos 通用 DOS 中断接口 .....	712
gettextinfo 读取文本模式的显示信息 .....	673	intdosx 通用 DOS 中断接口 .....	713
gettextsettings 返回当前图形字体的有关 信息 .....	674	intr 改变软中断接口 .....	714
gettime 读取系统时间 .....	676	ioctl I/O 设备控制 .....	715
getvect 读取中断向量 .....	677	isalnum 字符分类宏 .....	716
getverify 取得 DOS 的当前校验状态 .....	677	isalpha 字符分类宏 .....	717
getviewsettings 返回有关当前视区的 信息 .....	678	isascii 字符分类宏 .....	717
getw 从输入流中读取一整数 .....	679	isatty 检查设备类型 .....	718
getx 返回当前图形方式下位置的 x 坐 标值 .....	681	isctrl 字符分类宏 .....	718
gety 返回当前位置的 y 坐标值 .....	682	isdigit 字符分类宏 .....	719
gmtime 把日期和时间转换为格林威治标准时间 (GMT) .....	683	isgraph 字符分类宏 .....	720
gotoxy 在文本窗口中定位文本光标 .....	684	islower 字符分类宏 .....	720
graphdefaults 复位图形设置 .....	684	isprint 字符分类宏 .....	721
grapherrormsg 返回一个指向错误信息串的 指针 .....	685	ispunct 字符分类宏 .....	721
_graphfreemem 可修改的图形内存释放 函数 .....	686	isspace 字符分类宏 .....	722
_graphgetmem 可修改的图形内存分配 函数 .....	688	isupper 字符分类宏 .....	722
graphresult 返回最后一次失败图形操作的错误 码 .....	689	isxdigit 字符分类宏 .....	723
Harderr 建立一个错误处理程序 .....	691	itoa 把整数转换为字符串 .....	723
hardresume 硬件错误处理函数 .....	694	kbhit 检查当前按下的键 .....	724
hardretn 硬件错误处理函数 .....	697	keep 驻留并退出 .....	724
highvideo 选择高亮度字符 .....	699	labs 给出长型绝对值 .....	726
hypot 计算直角三角形的斜边长 .....	700	ldexp 计算 x 乘以 2 的 exp 次方 .....	726
imagesize 返回保存位图象所需的缓冲 区大小 .....	700	ldiv 两个长整型数相除, 返回商和余数 .....	727
		lfind 线性搜索 .....	728
		line 在指定两点间画一直线 .....	729
		linereel 从当前位置(CP)到与 CP 有一相对距离 的点画一直线 .....	730
		lineto 从当前位置到(x,y)画一直线 .....	731
		localtime 把日期和时间转变为结构 类型 .....	732
		lock 设置文件共享锁 .....	733
		log 计算 x 的自然对数 .....	734

log10 计算 $\log(X)$ .....	735	outtextxy 在指定位置显示一字符串 .....	762
longjmp 执行非局部跳转 .....	735	parsfnm 分析文件名 .....	763
lowvideo 选择低亮度字符 .....	736	peek 返回由 segment:offset 指定的 内存中的字 .....	764
_lrotl 将无符号长整型数向 左循环移位 .....	737	peekb 返回由 segment:offset 指定的 内存中的字节 .....	765
_lrotr 将无符号长整型数向 右循环移位 .....	738	perror 打印系统错误信息 .....	766
lsearch 线性搜索 .....	738	pieslice 绘制并填充扇形 .....	767
lseek 移动文件指针 .....	740	poke 在由 segment:offset 指定的内存中 存储一个字 .....	768
ltoa 把一个长整型数转换为字符串 .....	741	pokeb 在由 segment:offset 指定的内存中 存储一个字节 .....	768
malloc 分配内存 .....	741	poly 根据参数产生一个多项式 .....	769
matherr 用户可修改的数学错误处理 程序 .....	742	pow 计算 $x$ 的 $y$ 次方 .....	770
max 返回两数中较大的数 .....	744	pow10 指数函数 10 的 $p$ 次方 .....	770
memcpy 拷贝一个 $n$ 字节长的字符串 .....	744	printf 写格式化输出到 stdout .....	771
memchr 字符串中搜索字符 .....	745	putc 输出一个字符到流中 .....	777
memcmp 比较两个字符串 .....	745	putch 向屏幕输出字符 .....	777
memcpy 拷贝字符串 .....	746	putchar 在 stdout 上输出字符 .....	778
memicmp 比较两个字符数组中的 $n$ 个字节,忽 略大小写 .....	747	putenv 将字符串放入当前环境中 .....	779
memmove 拷贝块中的 $n$ 字符 .....	747	putimage 输出一个位图象到图形屏 幕上 .....	780
memset 将一个内存块的 $n$ 个字节都设 置为 $c$ .....	748	putpixel 写像素点 .....	782
min 返回两个值中较小的一个 .....	748	puts 输出一字符串到标准输出(stdout) ..	783
mkdir 创建目录 .....	749	puttext 从内存区拷贝文本到屏幕 .....	784
MK_FP 设置一个远指针 .....	750	putw 输出一整数到流中 .....	784
mktemp 建立一个唯一的文件名 .....	751	qsort 用快速排序算法进行排序 .....	786
modf 把双精度数转化为科学计数法 .....	751	raise 向正在执行的进程发送一个软 中断信号 .....	787
movedata 拷贝数据 .....	752	rand 产生随机数 .....	787
moverel 从当前位置(CP)移动一相对 距离 .....	752	randbrd 随机块读 .....	788
movetext 将屏幕上的文本从一个矩形区域拷贝 到另一个矩形区域 .....	754	randbwr 随机块写 .....	789
moveto 从当前坐标位置(CP)移到 ( $x,y$ ) .....	754	random 随机数发生器 .....	791
movmem 移动一长为 length 字节的串 .....	755	randomize 初始化随机数发生器 .....	791
normvideo 选择正常亮度字符 .....	756	_read 读文件 .....	792
nosound 关闭 PC 机扬声器 .....	756	read 读文件 .....	793
_open 打开一个文件进行读或写 .....	757	real 返回复数的实部 .....	795
open 打开一个文件进行读或写 .....	758	realloc 重新分配内存 .....	795
outport outp 输出一个字到端口中 .....	760	rectangle 画一个矩形 .....	796
outportb 输出一个字节到端口 .....	760	registerbgidriver 注册已加载或连接进来的图形 驱动程序 .....	797
outtext 显示一个字符串 .....	761	registerbgifont 注册已连接进来的矢量 字体代码 .....	798

- remove 删除一个文件 ..... 800
- rename 文件改名 ..... 800
- restorecrtmode 恢复屏幕为调用 initgraph  
前的设置 ..... 801
- rewind 将文件指针重定位于流的开  
始处 ..... 802
- rmdir 删除目录 ..... 803
- \_rotl 将一个无符号整数(unsigned)左  
循环移位 ..... 804
- \_rotr 将一个无符号整数向右  
循环移位 ..... 805
- sbrk 改变数据段地址 ..... 806
- scanf 格式化输入 ..... 806
- searchpath 按 DOS 路径查找一个文件 ..... 813
- sector 画并填充椭圆扇区 ..... 814
- segread 读段寄存器值 ..... 815
- setactivepage 设置图形输出活动页 ..... 816
- setallpalette 改变所有的调色板颜色 ..... 817
- setaspectratio 设置图形纵横比 ..... 819
- setbkcolor 用调色板设置当前背景颜色 ... 821
- setblock 修改已分配的内存的大小 ..... 822
- setbuf 把缓冲区与流相联 ..... 823
- setcbreak 设置 control-break ..... 824
- setcolor 设置当前要画的线的颜色 ..... 825
- setdate 设置 DOS 日期 ..... 827
- setdisk 设置当前驱动器 ..... 827
- setdta 设置磁盘传输地址 ..... 828
- setfillpattern 选择自定义的填充模式 ..... 830
- setfillstyle 设置填充模式和颜色 ..... 831
- setftime 取得文件日期和时间 ..... 833
- setgraphbufsize 改变内部图形缓冲区的  
大小 ..... 834
- setgraphmode 将系统设置成图形模式并  
清屏 ..... 835
- setjmp 非局部跳转(在 MS-WINDOWS 中不能  
使用本功能) ..... 837
- setlinestyle 设置当前画线宽度和类型 ..... 838
- setmem 设置内存 ..... 840
- setmode 设置打开文件方式 ..... 841
- setpalette 改变调色板的颜色 ..... 841
- setrgbpalette 定义 IBM 8514 图形卡的  
颜色 ..... 843
- settextjustify 为图形函数设置文本的对齐  
方式 ..... 845
- settextstyle 为图形输出设置当前的文本  
属性 ..... 847
- settime 设置系统时间 ..... 849
- setusercharsize 修改矢量字体字母的宽度和  
高度 ..... 850
- setvbuf 使缓冲区与流相联 ..... 851
- setvect 设置中断矢量入口 ..... 852
- setverify 设置 DOS 中的校验标志状态 ..... 853
- setviewport 为图形输出设置当前视口 ..... 854
- setvisualpage 设置可见的图形页号 ..... 855
- setwritemode 设置图形方式下画线的输出  
模式 ..... 856
- signal 设置某一信号的对应动作 ..... 858
- sin 计算正弦值 ..... 861
- sinh 计算双曲正弦值 ..... 861
- sleep 执行挂起一段时间 ..... 862
- sopen 打开一共享文件 ..... 862
- sound 按指定频率打开 PC 扬声器 ..... 864
- spawnl, spawnl, spawnlp, spawnlpe, spawnv,  
spawnve, spawnvp, spawnvpe 创建并运行子  
进程 ..... 865
- sprintf 送格式输出到字符串 ..... 867
- sqrt 计算参数平方根的绝对值 ..... 868
- srand 初始化随机数发生器 ..... 869
- sscanf 从某串中扫描格式化输入 ..... 869
- stat 读取文件信息 ..... 871
- \_status87 取浮点状态 ..... 872
- stime 设置系统日期和时间 ..... 873
- strcpy 拷贝字符串 ..... 874
- strcat 串连接 ..... 874
- strchr 搜索串中某个给定字符的第一次  
出现 ..... 875
- strcmp 串比较 ..... 876
- strcmpi 忽略大小写的串比较 ..... 876
- strcpy 串拷贝 ..... 877
- strcspn 搜索串中不包含给定字符集之子集  
的第一个段 ..... 878
- strdup 复制串 ..... 878
- \_strerror 建立用户定义的错误信息 ..... 879
- strerror 返回指向错误信息字符串的  
指针 ..... 879
- stricmp 忽略大小写的串比较 ..... 880

strlen	计算字符串的长度	881
strlwr	转换字符串中的大写字母为小写字母	881
strncat	把字符串的一部分附加到另一个串之后	882
strncmp	把串的一部分与另一个串的一部分进行比较	882
strncmpi	忽略大小写的串部分比较	883
strnset	将串中指定数目字节设置为字符	884
strpbrk	搜索给定集合中任一字符在串中的首次出现	884
strrchr	搜索给定字符在串中的最后一次出现	885
strrev	颠倒串中各字符的顺序	885
strset	设置串中所有字符为给定字符	886
strspn	搜索给定字符集的子集在串中第一次出现的段	886
strstr	搜索给定子串在某串中的出现位置	887
strtod	把串转换为双精度数值	887
strtok	搜索串中的某单词,该单词由第二个串中指定的符号进行分隔	888
strtol	转换串为长整型数	889
strtoul	将字符串转换为给定基数的无符号长整型值	890
swab	交换字节	891
system	执行 DOS 命令	891
tan	计算正切值	892
tanh	计算参数 x 的双曲正切值	892
tell	取文件指针的当前位置	893
textattr	设置文本属性	894
textbackground	选择文本的背景颜色	895
textcolor	选择文本模式的前景颜色	896
textheight	返回以像素为单位的字符串高度	897
textmode	将屏幕设置成文本模式	899
textwidth	返回以像素为单位的字符串宽度	900
time	取时间	901
tmpfile	以二进制方式打开临时文件	901
tmpnam	创建唯一的文件名	902
toascii	转换字符为 ASCII 格式	903

_tolower	转换字母为小写	903
tolower	转换字符为小写	904
_toupper	转换字母为大写	904
toupper	转换字符为大写	905
tirg	三角函数	906
tzset	设置全局变量 daylight、timezone 和 tzname 的值	906
ultoa	转换无符号长整型值为字符串	907
ungetc	把一个字符回退到输入流中	908
ungetch	把一个字符回退到键盘缓冲区	909
unixtodos	把 UNIX 格式的日期和时间转换成 DOS 格式	909
unlink	删除文件	910
unlock	解除文件共享锁	911
va_arg, va_end, va_start	实现可变参数表	912
vfprintf	送格式化输出到一流中	913
vfscanf	从流中搜索和格式化输入	914
vprintf	送格式化输出到 stdout	916
vscanf	从 stdin 中搜索和格式化输入	916
vsprintf	送格式化输出到串中	917
vsscanf	从流中搜索和格式化输入	918
wherex	给出窗口内光标水平位置	919
wherey	给出窗口内光标垂直位置	920
window	创建活动文本模式窗口	920
_write	写文件	921
write	写文件	922

## 第十八章 全局变量

_8087	协处理器芯片标志	924
_argc	保存命令行的参数个数	924
_argv	命令行参数指针数组	924
_ctype	字符属性信息数组	924
daylight	指示是否进行夏令时间调整	925
directvideo	视频输出控制的标志	925
environ	存取 DOS 环境变量	925
errno, _doserrno, sys_errlist, sys_nerr	使 perror 能打印错误信息	925
_fmode	设置缺省文件传送模式	927
_heaplen	保存近堆的长度	928
_openfd	存取模式数组	928
_osmajor, _osminor, _version	包含 DOS 版本的主号和次号	929

_ psp	包含当前程序的程序段前缀 (PSP)的段地址 .....	929
_ stklen	保存堆栈的大小 .....	929
timezone	包含当地时间与格林威治时间(GMT) 之间的差值(以秒为单位) .....	930
tzname	时区名指针数组 .....	930
_ version	DOS 版本号 .....	930
_ wscroll	指示控制台 I/O 函数是否滚屏 .....	930

## 第五部分 附录

### 附录 A TCC 命令行编译器

A.1	使用命令行编译器 .....	933
A.1.1	使用选择项 .....	933
A.1.2	语法和文件名 .....	935
A.1.3	应答文件 .....	936
A.1.4	配置文件 .....	936
A.2	编译器选项 .....	937
A.2.1	存储模式 .....	937
A.2.2	宏定义 .....	938
A.2.3	代码生成选项 .....	938
A.2.4	优化选择项 .....	939
A.2.5	源代码选项 .....	940
A.2.6	出错报告选择项 .....	941
A.2.7	段命名控制 .....	942
A.2.8	编译控制选择项 .....	943
A.4	环境选项 .....	943
A.4.1	查找包含文件和库文件 .....	943
A.4.2	文件搜索算法 .....	944
A.4.3	一个实例 .....	944

### 附录 B 实用程序

B.1	MAKE 实用程序 .....	946
B.1.1	MAKE 的工作过程 .....	946
B.1.2	启动 MAKES .....	947
B.1.3	MAKE 的一种简单运用 .....	948
B.1.4	制作 makefile 文件 .....	950
B.1.5	makefile 文件的组成 .....	950
B.1.6	命令表 .....	950
B.1.7	显式规则 .....	953

B.1.8	隐式规则 .....	955
B.1.9	宏 .....	956
B.1.10	指令 .....	959
B.1.11	MAKE 出错信息 .....	963
B.2	TLIB:库管理程序 .....	966
B.2.1	为什么使用目标模块库 .....	967
B.2.2	TLIB 命令行 .....	967
B.2.3	操作列表 .....	968
B.2.4	使用响应文件 .....	969
B.2.5	建立扩展字典:/E 选项 .....	969
B.2.6	设置页大小:/P 选项 .....	969
B.2.7	高级操作:/C 选项 .....	969
B.2.8	例子 .....	970
B.3	连接程序 TLINK .....	970
B.3.1	调用 TLINK .....	970
B.3.2	使用响应文件 .....	972
B.3.3	和 Turbo C 模块一起使用 TLINK .....	973
B.3.4	与 TCC 一起使用 TLINK .....	974
B.3.5	连接选项 .....	975
B.3.6	TLINK 的限制 .....	977
B.3.7	出错信息 .....	977
B.4	THELP 帮助 .....	980
B.4.1	装入和调用 THELP .....	980
B.4.2	THELP 选项 .....	981
B.5	GREP 查找程序 .....	984
B.5.1	命令行形式 .....	984
B.5.2	GREP 的选项 .....	985
B.5.3	正常的优先次序 .....	986
B.5.4	搜寻字符串 .....	986
B.5.5	正则表达式的操作符 .....	987
B.5.6	文件说明 .....	987
B.5.7	GREP 使用示例 .....	987
B.6	其它实用程序 .....	990
B.6.1	BGIOBJ:图形驱动程序 和字体的转换程序 .....	991
B.6.2	CPP:预处理实用程序 .....	995
B.6.3	OBJXREF:目标模块的 交叉引用列表实用程序 .....	997
B.6.4	PRJCFG .....	1004
B.6.5	TOUCH .....	1004



## 绪 论

1986 年 12 月,美国 Wizard Software System 公司推出了一个 C 语言编译器,称为 Wizard C。Wizard C 是一个颇受欢迎的 C 语言编译器,一般评论者称之为完美的 C 语言编译器。Wizard C 的优点是:快速的编译速度;有效率的代码优化;与 X3J11 委员会建议的 ANSI 标准兼容;提供开发中断服务子程序的许多 C 语言扩展功能等。这些扩展功能包括一个特别的中断函数类型、行内汇编语言,及允许 C 语言对微型机寄存器做访问动作的虚拟内存变量(Virtual memory variable)。

1987 年,Borland 公司推出 Turbo C。它是 Turbo Pascal 编译器的新伙伴。Borland 宣称 Turbo C 可用每分钟 7000 行的速度对程序进行编译。

在同一个月,Wizard 刊出最后一次的杂志广告,表示现在已到了 Turbo C 的时代了。

事实上,Borland 从 Wizard 那里学到 Turbo C 设计方法。当 Borland 公司想在 Turbo Pascal 及 Turbo Basic 之外开发出一个大众化的快速 C 语言编译器时,会详细考虑整个软件界的情况。与其重新开发出一个全新的编译器,Borland 公司做了一个明智的选择,买一个最好的 C 语言编译器,然后再改良这个编译器。

Borland 公司不再介绍一个令人厌烦的 C 语言编译器,决意为 C 语言的开发环境重定出一个新的模式。

### 0.1 两种 Turbo C 环境

Turbo C 是两个 C 编译器的集成:一个是通常 UNIX 格式的命令行(command line)编译器、连接器等组合,另一个是 Turbo C 集成开发环境(Integrated Development Environment 缩写为 IDE)。

在命令行编译环境中包括一个 MAKE 实用程序、一个 TCC 编译器以及目标文件管理程序 TLIB 等实用程序。命令行语言编译器与其他个人计算机上的 C 编译器类似,只是速度更快。所有有关命令行环境的任何信息都可以在章节和附录中找到。

集成开发环境是一个称为 TC 的程序。TC 集成了一个程序员的编辑器、一个联机面向工程的 MAKE 实用程序、一个程序运行实用程序,还包括一个源代码级的调试器。

这个操作环境是 Turbo C 的表示方式,功能最强大的地方是由于编辑器、编译器、连接器和调试器间的集成。在操作环境下,程序员可以编辑一个程序,进行编译工作,然后再与其他原始模块及函数库连接在一起以运行这个程序,并可利用其内部的调试器对程序的错误进行定位并帮助分析,富有成效地纠正程序的错误。这个特性显示出新一代 C 语言编译器