

北京科海培训中心

TURBO C

实用大全

徐金梧 杨德斌 徐科 编

机械工业出版社

北京科海培训中心

Turbo C 实用大全

徐金梧 杨德斌 徐科 编著

机械工业出版社

(京)新登字 054 号

本书全面系统地讲述了 C 语言、Turbo C 库函数、集成开发环境及 Turbo C 的应用。

全书分三部分。第一部分介绍 C 语言基础知识;第二部分按函数功能分类依字母顺序逐一介绍函数的功能调用,参数说明,并针对每一函数附以完整的范例程序,有助于读者理解这些库函数;第三部分介绍用 C 编写各种应用实例,如汉字技术,中断处理技术,图形、图象技术以及工业控制技术。

全书内容深入浅出,通俗易懂,从最简单的例子着手讲解并逐步深入,不仅适用 C 语言初学者,对使用 C 语言编写大型、复杂应用程序的程序员也是一本实用的工具书。

图书在版编目(CIP)数据

Turbo C 实用大全/徐金梧等编著. —北京:机械工业出版社,1996.5

ISBN 7-111-05227-7

I. T... II. 徐... III. C 语言 IV. TP312C

中国版本图书馆 CIP 数据核字(96)第 07282 号

出版人:马九荣 (北京市百万庄南街 1 号 邮政编码 100037)

责任编辑:温莉芳 版式设计:成昊 责任校对:丁丽丽

封面设计:成昊

北京市朝阳区科普印刷厂印刷·新华书店北京发行所发行

1996 年 5 月第 1 版·1999 年 4 月第 4 次印刷

787mm×1092mm 1/16·38.75 印张·967 千字

13 001—15000 册

定价:42.00 元

15638/01

前言

C 语言由于具有灵活、高效、可移植性好等诸多优点,已成为软件开发中最常用的计算机编程语言。目前,微机上运行的最流行的 C 语言版本主要有 Microsoft 公司的 MS C 和 Borland 公司的 Turbo C。虽然这两类 C 语言在功能上非常接近,但 Turbo C 以其快速的编程和简明易学的风格受到广大编程者的欢迎。尤其对初学者来说,从 Turbo C 着手学习 C 语言是一种明智的选择。

本书的目的就是为初学者提供一本全面了解 C 语言基础、库函数和 Turbo C 的集成开发环境的教材,因此本书在内容编排上特别注意了两个字:1. “易”字,通俗易懂,深入浅出,从最简单的例子着手逐步深入,使初学者和有基础的编程者都可以从中受益。2. “全”字,系统、完整地介绍 C 语言的基本概念、基本语法、基本函数以及 Turbo C 所提供的全部库函数、集成开发环境和其他工具;另外,还重点介绍了一些常用的应用实例。因此对于用 C 语言编写大型、复杂的应用程序的编程者来说,也是一本重要的工具书。

全书共分为三部分,第一部分主要介绍 C 语言的基础,共 14 章。

第 1 章介绍 C 语言概述,包括它的发展史、特点、C 语言程序的格式、编译等内容。熟悉 C 语言的读者可跳过这部分内容,直接阅读第 2 章。

第 2 章描述了 C 语言的基本数据类型、运算符和表达式。数据、运算符和表达式构成了 C 语言的基本元素。C 语言的基本数据类型和其他计算机编程语言相似。

但是,由于 C 语言是一种结构化语言,程序中的代码段与数据段是分离、代码享用模块化形式,所以定义了各种数据类型的说明符,这些说明符需要读者熟练掌握,灵活运用,这些说明符能使代码可靠,效率高,但若使用不正确,会造成程序难以查询的错误。除了具备其他计算机编程语言所常用的形式以外,还提供一些 C 语言所独有的运算符,熟练地应用这些运算符是掌握 C 语言的基础。

第 3 章程序控制语句和第 4 章函数这两章内容比较容易理解,但需要反复体会函数的通用值,以及函数原型的说明。函数指针是非常容易引起混淆的,读者可先阅读第 6 章指针部分内容后再仔细琢磨函数指针。

第 5 章和第 6 章分别介绍数组和指针。在 C 语言中数组与指针有着非常密切的关系,特别是当数组是以动态分配形式说明时,了解指针与数组之间的内在关系尤其重要。指针是 C 语言最具特色的标志之一,也是检验读者是否真正掌握 C 语言的试金石。正确地运用指针需要在实际中反复细心地体会,建议读者对指针部分的内容应仔细阅读。

第 7 章描述了 C 语言提供的组合数据类型,包括结构、位域联合、枚举和用户定义的类型。这部分内容与第 2 章所介绍的基本数据类型构成了 C 语言程序数据的基础,其他任何复杂形式的数据类型都可由这两类数据类型派生出来。

第 8 章主要介绍 C 语言的输入、输出,与其他计算机编程语言不同,C 语言本身并没有定义输入输出函数,它只定义了“流”和“文件”这两个概念,同时 ANSI 还制订了完整的操作标准。由于输入输出函数与硬件部分有关,所以这部分函数实际上都是由软件公司所编写的

库函数来提供,但是大多数输入输出库函数都是按标准编写的,所以对于不同的文件,C语言的输入输出代码都具有高度的兼容性。

第9章介绍 Turbo C 的各种预处理指令和编译选择。这种指令在编写商用软件时十分有用。设置这种指令是为了适应各种用户的需求和硬件的支持。

第10章介绍 Turbo C 所支持的各种存储模式,正确地选择存储模式将有助于改善程序的运行效率和对内存需求的安排。

第11章描述了 Turbo C 的集成开发环境。Turbo C 2.0 提供了一个十分灵巧、高效的集成开发环境。这章内容需要读者逐步掌握,灵活地运用这个开发环境所提供的工具有利于你在编程与调试过程中提高工作效率。

第12章介绍 Turbo C 与汇编语言的接口,第13章描述了 Turbo C 与软件工程,第14章介绍效率、可移植性及调试,这三章内容都是在高级编程阶段应该注意的问题,虽然 C 语言在效率、可移植性方面都是十分出色的,但若熟练地运用这三章所介绍的内容会使你受益匪浅。

本书的第二部分介绍 Turbo C 所提供的库函数,从第15章到第25章,共11章,按照函数功能进行分类,然后将同类的库函数以字母顺序排列,不仅详细地介绍调用方式、参数的说明,而且给出一个完整的例子,有助于读者理解这些库函数。

本书的第三部分介绍用 C 语言编写各种应用实例,这些例子都是读者在实际工作中可能需要用到的。

第26章介绍用 Turbo C 创造各种图形和汉字显示技术,包括如何在图形模式下使用文本输出函数,产生“图文并茂”的视觉效果,如何在汉字和西文 DOS 下显示汉字,如何编写中文下拉菜单,如何使用鼠标器。这些编程技巧在编写你的界面时经常会遇到。

第27章讨论了图象创作。介绍了如何制作动画,使用位图设计封面,用 16M 真彩色图象显示设计逼真的画面,以及如何调配各种颜色等技术。

第28章介绍调用系统外部硬件资源技术,包括如何测试显示卡和显示器的分辨率,如何调用键盘的功能键,如何利用软件调整和定位软驱的磁头,如何控制打印机和屏幕图形的硬拷贝。

第29章讨论中断技术和串行通信,介绍如何在 DOS 环境下实现多任务操作以及内存驻留技术,如何实现硬件中断和串行通信。

第30章介绍 C 语言用于工业自动控制系统的实例。

第31章介绍用 Turbo C 编写数据库。

附录 A 描述了 ANSI C 与 K&R C 的区别。附录 B 给出了 Turbo C 编译过程中的错误信息,便于读者在编译 C 语言程序时,查找错误原因。

由于时间仓促,编者学识所囿,疏漏之处在所难免,企望各位专家同仁指正。

编著者

1996年4月

目 录

前 言	(1)
-----------	-----

第一部分 C 语言

第 1 章 C 语言概述	(1)
1.1 C 语言的发展史	(1)
1.2 C 语言的特点	(1)
1.3 C 语言程序的格式	(2)
1.4 C 语言程序的编译	(3)
1.4.1 编译程序和解释程序	(3)
1.4.2 分块编译	(4)
1.4.3 库函数和连接	(4)
第 2 章 变量、常数、运算符和表达式	(5)
2.1 标识符名	(5)
2.2 数据类型	(5)
2.3 类型说明符	(6)
2.4 访问说明符	(7)
2.5 变量的说明	(8)
2.6 局部变量	(8)
2.7 形式参数	(9)
2.8 全程变量	(10)
2.9 存储类型说明符	(11)
2.9.1 外部变量(extern)	(11)
2.9.2 静态变量(static variables)	(12)
2.9.3 寄存器变量(register variables)	(15)
2.10 赋值语句	(15)
2.10.1 赋值中的类型转换	(15)
2.10.2 变量初始化	(17)
2.11 常 量	(17)
2.11.1 控制字符常量	(17)
2.12 运算符	(18)
2.12.1 算术运算符	(18)
2.12.2 增 1 和减 1 运算符	(19)
2.12.3 关系运算符和逻辑运算符	(20)
2.12.4 按位运算符	(21)
2.12.5 “?”运算符	(23)

2.12.6	“&”和“*”运算符	(24)
2.12.7	编译状态运算符 sizeof	(25)
2.12.8	逗号运算符	(26)
2.12.9	“.”和“->”运算符	(26)
2.12.10	方括号“[]”和圆括号“()”	(27)
2.12.11	运算符优先次序表	(27)
2.12.12	表达式	(28)
2.12.13	表达式中的类型转换	(28)
2.12.14	强制类型转换	(28)
2.12.15	空格和圆括号	(29)
2.12.16	C 语言的简写	(29)
第 3 章	程序控制语句	(31)
3.1	C 语言中的逻辑变量	(31)
3.2	C 语言的语句	(31)
3.3	if 语句	(31)
3.3.1	if 的嵌套形式	(32)
3.3.2	阶梯式 if-else-if 语句	(33)
3.3.3	? 运算符	(34)
3.4	switch 语句	(35)
3.5	for 语句	(38)
3.5.1	for 循环的变体	(39)
3.5.2	无限循环	(41)
3.5.3	无循环体 for 循环	(42)
3.6	while 语句	(42)
3.7	do-while 语句	(44)
3.8	break 语句	(44)
3.9	exit 语句	(46)
3.10	continue 语句	(46)
3.11	标号和 goto 语句	(47)
第 4 章	函 数	(49)
4.1	返回语句	(49)
4.1.1	从函数返回	(49)
4.1.2	返回值	(50)
4.2	函数作用域规则	(51)
4.3	函数参数	(52)
4.3.1	赋值调用和赋地址调用	(52)
4.3.2	函数调用与指针	(53)
4.3.3	函数调用与数组	(54)
4.4	主函数中的命令行参数——argc 和 argv	(57)
4.5	函数返回非整型值	(59)
4.5.1	返回指针	(60)
4.6	函数原型	(61)

4.6.1 参数说明的现代风格与传统风格	(62)
4.7 函数递归	(62)
4.8 指向函数的指针	(63)
4.9 补充问题	(65)
4.9.1 参数和通用函数	(65)
4.9.2 效 率	(66)
第5章 数 组	(67)
5.1 一维数组	(67)
5.2 传递一维数组给函数	(68)
5.2.1 字符串	(69)
5.3 二维数组	(70)
5.3.1 字符串数组	(73)
5.4 多维数组	(74)
5.5 数组与指针	(75)
5.6 数组空间的分配	(76)
5.7 数组的初始化	(79)
5.7.1 不定长数组的初始化	(79)
5.8 应用举例——井字游戏(Tic-Tac-Toe)	(80)
第6章 指 针	(83)
6.1 指针是地址	(83)
6.2 指针变量	(83)
6.3 指针运算符	(84)
6.4 指针表达式	(84)
6.4.1 指针的赋值	(84)
6.4.2 指针的算术运算	(85)
6.4.3 指针比较	(86)
6.5 Turbo C 动态分配函数	(87)
6.6 指针和数组	(88)
6.6.1 指向字符型数组的指针	(89)
6.6.2 指针数组	(90)
6.7 指向指针的指针	(91)
6.8 指针初始化	(92)
6.9 函数型指针	(93)
6.10 使用指针的一些问题	(95)
第7章 结构、联合、用户定义的类型和枚举	(97)
7.1 结 构	(97)
7.1.1 访问结构元素	(98)
7.1.2 结构数组	(99)
7.1.3 实 例	(99)
7.2 将结构传递给函数	(104)
7.2.1 将结构元素传递给函数	(104)

7.2.2 将整个结构传递给函数	(104)
7.3 结构指针	(105)
7.3.1 结构指针说明	(105)
7.3.2 使用结构指针	(106)
7.4 结构内部的数组和结构	(109)
7.5 位 域	(109)
7.6 联 合	(111)
7.7 枚 举	(113)
7.8 使用 sizeof 来确保可移植性	(114)
7.9 使用 typedef	(115)
第 8 章 输入、输出和磁盘文件	(117)
8.1 流(stream)和文件(file)	(117)
8.1.1 流的概念(stream)	(117)
8.1.2 文件(file)	(118)
8.2 概念和实际	(118)
8.3 控制台 I/O	(119)
8.3.1 getche()和 putchar()函数	(119)
8.3.2 gets()和 puts()函数	(120)
8.4 控制台格式化 I/O	(121)
8.4.1 printf()函数	(121)
8.4.2 scanf()函数	(122)
8.5 缓冲型 I/O 系统	(125)
8.5.1 文件指针	(125)
8.5.2 fopen()函数	(125)
8.5.3 putc()函数	(127)
8.5.4 getc()函数	(127)
8.5.5 fclose()函数	(127)
8.5.6 ferror()和 rewind()函数	(128)
8.5.7 fopen(),getc(),putc()和 fclose()函数的用法	(128)
8.5.8 getw()和 putw()函数	(130)
8.5.9 fgets()和 fputs()函数	(130)
8.5.10 fread()和 fwrite()函数	(130)
8.5.11 fseek()函数和随机访问 I/O	(131)
8.5.12 stdin,stdout 和 stderr	(134)
8.5.13 fprintf()和 fscanf()函数	(134)
8.5.14 删除文件	(136)
8.6 非缓冲型 I/O-UNIX 型文件系统	(136)
8.6.1 open(),creat()和 close()函数	(136)
8.6.2 write()和 read()函数	(138)
8.6.3 unlink()函数	(139)
8.6.4 随机访问文件和 lseek()函数	(139)
8.7 方法选择	(141)

第 9 章 Turbo C 预处理指令和编译选择	(142)
9.1 Turbo C 的预处理指令	(142)
9.2 #define 指令	(142)
9.3 #error 指令	(144)
9.4 #include 指令	(145)
9.5 条件编译指令	(145)
9.5.1 #if, #else, #elif 和 #endif 指令	(145)
9.5.2 #ifdef 和 #ifndef 指令	(147)
9.6 #undef 指令	(148)
9.7 #line 指令	(148)
9.8 #pragma 指令	(149)
9.9 预定义的宏替换名	(149)
第 10 章 Turbo C 存储模式	(152)
10.1 8086 系列微处理器	(152)
10.2 地址的计算	(152)
10.3 16 位与 20 位指针	(154)
10.4 存储模式	(154)
10.4.1 微型模式(Tiny Model)	(154)
10.4.2 小型模式(Small Model)	(155)
10.4.3 中型模式(Medium Model)	(155)
10.4.4 紧凑型模式(Compact Model)	(156)
10.4.5 大型模式(Large Model)	(157)
10.4.6 巨型模式(Huge Model)	(157)
10.4.7 模式的选择	(157)
10.4.8 编译程序的内存模式选择项	(158)
10.5 混合模式编程	(158)
10.5.1 far(远程)	(159)
10.5.2 near(近程)	(159)
10.5.3 huge(特大)	(159)
10.6 Turbo C 的段说明符	(160)
10.7 内存显示和修改程序实例	(160)
10.7.1 display_mem()函数	(160)
10.7.2 change_mem()函数	(160)
10.7.3 完整的内存显示和修改程序	(161)
第 11 章 Turbo C 2.0 集成开发环境	(164)
11.1 怎样使用 TC	(164)
11.1.1 TC 命令行开关	(165)
11.1.2 TC 内部命令的使用	(165)
11.1.3 TC 热键	(166)
11.1.4 菜单结构	(167)
11.2 菜单命令	(169)

11.2.1	文件(File)菜单	(170)
11.2.2	编辑(Edit)命令	(171)
11.2.3	运行(Run)菜单	(173)
11.2.4	编译(Compile)菜单	(175)
11.2.5	项目(Project)菜单	(177)
11.2.6	选择项(Options)菜单	(178)
11.2.7	调试(Debug)菜单	(189)
11.2.8	Break/Watch 菜单	(192)
第 12 章	Turbo C 与汇编语言程序的接口	(195)
12.1	调用协定	(196)
12.1.1	Turbo C 的调用协定	(196)
12.2	建立汇编代码函数	(197)
12.3	使用 asm	(203)
12.4	何时用汇编程序编程	(204)
第 13 章	Turbo C 用于软件工程	(206)
13.1	自顶向下的设计	(206)
13.1.1	草拟你的程序	(206)
13.1.2	选择数据结构	(207)
13.2	“防范”功能	(208)
13.3	函数原型	(209)
13.4	编译多个文件	(210)
13.4.1	项目(project)及项目选择项	(210)
13.4.2	你自己去尝试	(211)
13.4.3	指定附加从属关系	(211)
13.5	独立应用 MAKE	(212)
13.6	TLIB——Turbo C 库管理程序	(215)
13.7	GREP 的应用	(217)
第 14 章	效率、可移植性及调试	(218)
14.1	效 率	(218)
14.1.1	增 1 和减 1 运算符	(218)
14.1.2	使用寄存器变量	(220)
14.1.3	指针与数组下标	(222)
14.1.4	函数的使用	(222)
14.2	程序的移植	(225)
14.2.1	使用 #define	(225)
14.2.2	对操作系统的依赖	(226)
14.2.3	数据长度的差别	(226)
14.3	调 试	(227)
14.3.1	运算顺序错误	(227)
14.3.2	指针问题	(228)
14.3.3	重新定义函数	(229)

14.3.4	时断时续的错误	(230)
14.3.5	边界错误	(230)
14.3.6	函数说明的疏漏	(231)
14.3.7	调用参数错误	(232)
14.3.8	栈(stack)一堆(heap)冲突	(232)
14.3.9	使用原型以防出错	(233)
14.4	一般的调试理论	(233)
14.5	程序维护的艺术	(234)
14.5.1	修改错误	(234)
14.5.2	保护源程序	(235)

第二部分 Turbo C 库

第 15 章	连接、库和头部文件	(236)
15.1	连接程序(Linker)	(236)
15.2	C 标准库	(237)
15.2.1	库文件和目标文件的比较	(237)
15.2.2	头部文件(Header File)	(237)
15.3	头部文件中的宏定义	(238)
第 16 章	I/O 函数	(240)
16.1	access()函数	(240)
16.2	cgets()函数	(241)
16.3	_chmod()函数	(241)
16.4	chmod()函数	(242)
16.5	clearerr()函数	(243)
16.6	close()和 _close()函数	(244)
16.7	cprintf()函数	(244)
16.8	creat(), _creat(), creatnew()和 creattemp()函数	(245)
16.9	cscanf()函数	(246)
16.10	dup()和 dup2()函数	(247)
16.11	eof()函数	(248)
16.12	fclose()和 fcloseall()函数	(248)
16.13	fdopen()函数	(249)
16.14	feof()函数	(250)
16.15	ferror()函数	(250)
16.16	fflush()函数	(251)
16.17	fgetc()函数	(251)
16.18	fgetchar()函数	(252)
16.19	fgets()函数	(252)
16.20	filelength 函数()	(253)
16.21	fileno()函数	(253)
16.22	fopen()函数	(254)

16.23	fprintf()函数	(255)
16.24	fputc()函数	(256)
16.25	fputchar()函数	(257)
16.26	fputs()函数	(257)
16.27	fread()函数	(258)
16.28	freopen()函数	(259)
16.29	fscanf()函数	(259)
16.30	fseek()函数	(260)
16.31	fstat()函数	(261)
16.32	ftell()函数	(262)
16.33	fwrite()函数	(263)
16.34	getc()函数	(263)
16.35	getch()和 getche()函数	(264)
16.36	getchar()函数	(264)
16.37	getpass()函数	(265)
16.38	gets()函数	(265)
16.39	getw()函数	(266)
16.40	ioctl()函数	(267)
16.41	isatty()函数	(268)
16.42	kbhit()函数	(268)
16.43	lock()函数	(269)
16.44	lseek()函数	(270)
16.45	open()和 _open()函数	(271)
16.46	perror()函数	(273)
16.47	printf()函数	(273)
16.48	putc()函数	(275)
16.49	putch()函数	(275)
16.50	putchar()函数	(276)
16.51	puts()函数	(276)
16.52	putw()函数	(277)
16.53	read()和 _read()函数	(278)
16.54	remove()函数	(278)
16.55	rename()函数	(279)
16.56	rewind()函数	(279)
16.57	scanf()函数	(280)
16.58	setbuf()函数	(283)
16.59	setftime()函数	(283)
16.60	setmode()函数	(284)
16.61	setvbuf()函数	(285)
16.62	sopen()函数	(286)
16.63	sprintf()函数	(287)
16.64	sscanf()函数	(287)
16.65	stat()函数	(288)

16.66	tell()函数	(288)
16.67	tmpfile()函数	(289)
16.68	tmpnam()函数	(289)
16.69	ungetc()函数	(290)
16.70	ungetch()函数	(291)
16.71	vprintf(),vfprintf()和 vsprintf()函数	(291)
16.72	vscanf()和 vscanf()函数	(292)
16.73	write()和 _write()函数	(293)

第 17 章 字符串、内存和字符函数 (294)

17.1	isalnum()函数	(296)
17.2	isalpha()函数	(296)
17.3	isascii()函数	(297)
17.4	isctrl()函数	(297)
17.5	isdigit()函数	(298)
17.6	isgraph()函数	(298)
17.7	islower()函数	(299)
17.8	isprint()函数	(299)
17.9	ispunct()函数	(300)
17.10	isspace()函数	(300)
17.11	isupper()函数	(301)
17.12	isxdigit()函数	(301)
17.13	memcpy()函数	(302)
17.14	memchr()函数	(303)
17.15	memcmp()和 memicmp()函数	(303)
17.16	memcpy()函数	(304)
17.17	memmove()函数	(304)
17.18	memset()函数	(305)
17.19	movedata()函数	(305)
17.20	movmem()函数	(306)
17.21	setmem()函数	(306)
17.22	strcpy()函数	(307)
17.23	strcat()函数	(307)
17.24	strchr()函数	(308)
17.25	strcmp(),stricmp()和 strcmpi()函数	(308)
17.26	strcpy()函数	(309)
17.27	strcspn()函数	(310)
17.28	strdup()函数	(310)
17.29	_strerror()函数	(311)
17.30	strerror()函数	(311)
17.31	strlen()函数	(312)
17.32	strlwr()函数	(312)
17.33	strncat()函数	(313)

17.34	strncmp()和 strnicmp()函数	(313)
17.35	strncpy()函数	(314)
17.36	strnset()函数	(315)
17.37	strpbrk()函数	(315)
17.38	strrchr()函数	(316)
17.39	strrev()函数	(316)
17.40	strset()函数	(317)
17.41	strspn()函数	(317)
17.42	strstr()函数	(318)
17.43	strtok()函数	(319)
17.44	strupr()函数	(319)
17.45	toascii()函数	(320)
17.46	_tolower()和 tolower()函数	(320)
17.47	_toupper()和 toupper()函数	(320)
第 18 章	数学函数	(322)
18.1	abs()函数	(322)
18.2	acos()函数	(322)
18.3	asin()函数	(323)
18.4	atan()函数	(323)
18.5	atan2()函数	(324)
18.6	cabs()函数	(324)
18.7	ceil()函数	(325)
18.8	cos()函数	(325)
18.9	cosh()函数	(326)
18.10	exp()函数	(326)
18.11	fabs()函数	(327)
18.12	floor()函数	(327)
18.13	fmod()函数	(328)
18.14	frexp()函数	(328)
18.15	hypot()函数	(329)
18.16	labs()函数	(329)
18.17	ldexp()函数	(330)
18.18	log()函数	(330)
18.19	log10()函数	(331)
18.20	matherr()函数	(331)
18.21	modf()函数	(332)
18.22	poly()函数	(333)
18.23	pow()函数	(333)
18.24	pow10()函数	(334)
18.25	sin()函数	(334)
18.26	sinh()函数	(335)
18.27	sqrt()函数	(335)

18.28	tan()函数	(336)
18.29	tanh()函数	(336)
第 19 章 时间转换和操作函数		(337)
19.1	asctime()函数	(337)
19.2	clock()函数	(338)
19.3	ctime()函数	(339)
19.4	difftime()函数	(339)
19.5	dostounix()函数	(340)
19.6	getdate()函数	(340)
19.7	gettime()函数	(341)
19.8	gmtime()函数	(341)
19.9	localtime()函数	(342)
19.10	setdate()函数	(342)
19.11	settime()函数	(343)
19.12	stime()函数	(344)
19.13	time()函数	(344)
19.14	tzset()函数	(345)
19.15	unixtodos()函数	(345)
第 20 章 接口函数		(346)
20.1	absread()和 abswrite()函数	(346)
20.2	bdos()和 bdosptr()函数	(347)
20.3	bioscom()函数	(348)
20.4	biosdisk()函数	(350)
20.5	biosequip()函数	(351)
20.6	bioskey()函数	(352)
20.7	biosmemory()函数	(354)
20.8	biosprint()函数	(354)
20.9	biostime()函数	(355)
20.10	country()函数	(356)
20.11	ctrlbrk()函数	(357)
20.12	delay()函数	(357)
20.13	disable()函数	(358)
20.14	dosexterr()函数	(358)
20.15	enable()函数	(359)
20.16	FP_OFF()和 FP_SEG()函数	(359)
20.17	geninterrupt()函数	(359)
20.18	getcbrk()函数	(360)
20.19	getdfree()函数	(360)
20.20	getdta()函数	(361)
20.21	getfat()和 getfatd()函数	(361)
20.22	getpsp()函数	(362)
20.23	getvect()函数	(363)

20.24	getverify()函数	(363)
20.25	harderr(),hardresume()和 hardretn()函数	(364)
20.26	inport()和 inportb()函数	(366)
20.27	intdos()和 intdosx()函数	(366)
20.28	int86()和 int86x()函数	(367)
20.29	interrupt()函数	(368)
20.30	intr()函数	(368)
20.31	keep()函数	(368)
20.32	MK_FP()宏	(369)
20.33	outp()函数	(369)
20.34	outport()和 outportb()函数	(369)
20.35	parsfnm()函数	(370)
20.36	peek(),peekb(),poke()和 pokeb()函数	(371)
20.37	randbrd()和 randbwr()函数	(372)
20.38	segread()函数	(372)
20.39	setcbkr()函数	(373)
20.40	setdta()函数	(374)
20.41	setvect()函数	(374)
20.42	setverify()函数	(374)
20.43	sleep()函数	(375)
第 21 章	动态地址分配	(376)
21.1	allocmem()函数	(376)
21.2	brk()函数	(377)
21.3	calloc()函数	(378)
21.4	coreleft()函数	(378)
21.5	farcalloc()函数	(379)
21.6	farcoreleft()函数	(379)
21.7	farfree()函数	(379)
21.8	farmalloc()函数	(380)
21.9	farrealloc()函数	(380)
21.10	free()函数	(380)
21.11	freemem()函数	(381)
21.12	malloc()函数	(381)
21.13	realloc()函数	(382)
21.14	sbrk()函数	(383)
21.15	setblock()函数	(383)
第 22 章	目录函数	(385)
22.1	chdir()函数	(385)
22.2	findfirst()和 findnext()函数	(386)
22.3	fnmerge()和 fnsplit()函数	(387)
22.4	getcurdir()函数	(388)
22.5	getcwd()函数	(389)