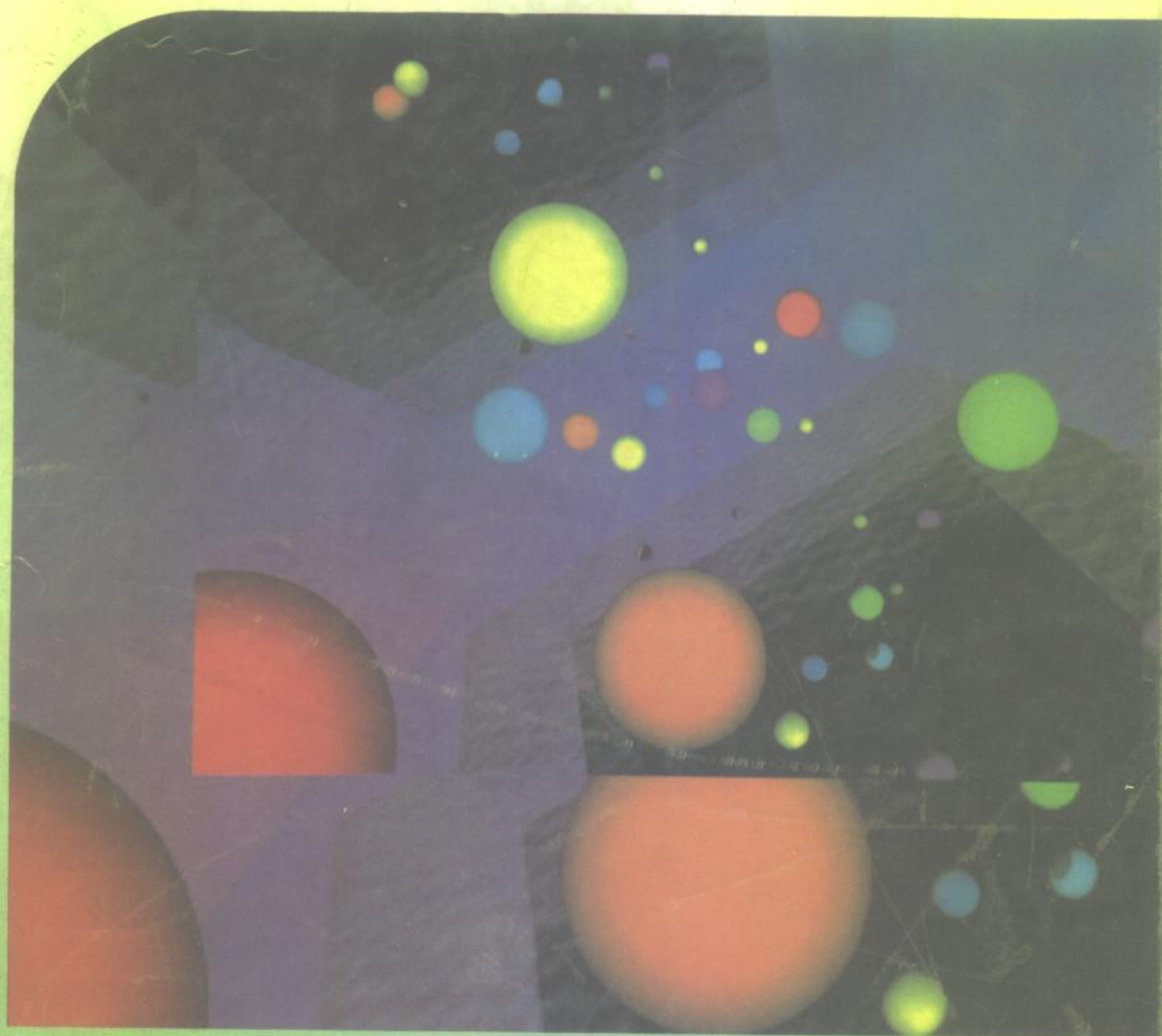


Visual C++ 程序开发指南 (一) ——概念与实例

● 何 亮 朱志强 等 编著



科学出版社

计算机软件应用系列

Visual C++ 程序开发指南(一)

—— 概念与实例

何 亮 朱志强 等 编著

科学出版社

1995

(京)新登字 092 号

内 容 简 介

Visual C++是目前流行的面向对象的 Windows 程序设计环境,它不仅界面友好、工具繁多,而且类库也很丰富。借助 Visual C++ MFC 类库中的类和 Visual C++的交互式工具,很容易就能生成一个可以实际运行的 Windows 或 DOS 应用程序。

本书不仅讨论了 OOP 的基本概念和 Visual C++的基础知识,而且对 Visual C++中的所有全局函数进行了彻底剖析。

本书可作为大专院校计算机专业学生的教材或参考书,也可供普通 Visual C++和 Windows 程序员参考。

计算机软件开发应用系列

Visual C++ 程序开发指南(一)

—— 概念与实例

何 亮 朱志强 等 编著

责任编辑 刘晓融 留 霞

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

化学工业出版社印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1995 年 6 月第 一 版 开本: 787×1092 1/16

1995 年 6 月第一次印刷 印张: 341/4

印数: 1—3 140 字数: 750 000

ISBN 7-03-004351-0/TP·395

定价: 43.90 元

前 言

面向对象程序设计(OOP)是目前流行的程序语言概念,Visual C++是汇集 Microsoft 公司技术精华的主流产品。《Visual C++程序开发指南》是《计算机软件应用系列》丛书中的一套,共四册。这套书从各个方面说明了应如何在 Visual C++环境下以 OOP 的概念设计 DOS 和 Windows 应用程序。

作为一个成熟的软件工程师,面对铺天盖地的 Word,Excel,PowerPoint, FoxPro, Visual Basic 等应用软件,不熟悉 Visual C++是很不明智的,因为借助 Visual C++,软件工程师就可以开发出很多动态链接库,从而为广大用户提供全方位的支持,其商业前景更是无可估量的。

Visual C++是一个面向对象、界面友好的 DOS 和 Windows 程序开发环境。在这个集成环境中开发应用程序时,不一定要手工设计用户界面,而只需选取菜单命令,Visual C++系统就会生成一个可实际运行的 Windows 应用程序框架。此后,程序员就可利用基于 Windows 的 C++编辑器,借助 AppWizard 建立面向对象的应用程序。

除了 AppWizard 外,Visual C++还包含 C++应用程序生成器、基于 Windows 的编辑器、基于 Windows 的面向图形的编程工具、使 C++代码和 Windows 消息及类成员函数相联系的交互式工具、可用来编写 Visual C++文本程序的 QuickWin 库以及预编译的头文件和源文件。当然,类库丰富更是 Visual C++的最大特色。

尽管目前市面上已有一些 Visual C++方面的书籍,但学习 Visual C++的人常常会有下列困惑:

- (1) 对 Visual C++缺乏一致的认识,遇到问题总是缺这少那,缺乏一个值得信赖的良师益友。
- (2) 只了解 OOP 的直观语法,而对于 OOP 有何好处、为什么 OOP 可取代结构化语言并成为未来程序设计风格的主流,都不太明白。
- (3) 无法理解 OOP 对未来程序的管理与维护究竟有何重要性。
- (4) 在 Visual C++环境下设计 OOP 程序时,对 Visual C++提供的资源不太清楚。
- (5) 不知道如何用 Visual C++和 Windows 解决实际问题。

针对这些问题,这套书从各个方面介绍了 Visual C++集成环境、OOP 概念、类库以及基于 DOS 和 Windows 的 C++应用程序开发方法,并提供了丰富的类模板和大量的应用程序实例。

本套书的内容基本属于中等难度,适用于初、中级读者。不熟悉 OOP 技术和 C++语言的读者可参阅第一册的第一至十三章,这几章主要介绍 Visual C++集成环境和 OOP 概念。经常从事应用程序开发的读者大都比较关心 Visual C++类库和通用类的构造方法,可参阅第二册和第三册的第七至九章,其中第二册主要介绍通用类的构造方法,并从我们所熟悉的数据结构着手,讨论如何设计和实现自己的通用类,这为进行大程序开发的用户提供了设计开发环境的手段;第三册第七至九章则主要介绍 Visual C++基础类库,讨论如何使用 Visu-

al C++类库。比较熟悉 OOP 概念而对 C++程序设计技术不太精通的读者可参阅第四册, 其中包含大量的程序设计实例。

本书是这套书的第一册, 参与编写的人员有: 何亮、朱志强、覃牧、何国辉、陆明辉、李华清、李杨、魏志国、李蕾、李建、刘心海、刘崇福、李纪鸿、刘石华、赵越、陆静宜、李伯雄、陈楚南、梁友国、陈佳海。全书由吴佳教授和陈忠敏研究员审校。此外, 朱小宝为本书的编排付出了辛勤的劳动。在此对以上同志深表感谢。

限于水平和时间, 书中的错误和不妥之处, 敬请读者批评指正。

目 录

第一章 Microsoft Visual C++集成环境	1
1.1 Microsoft Visual C++简介	1
1.2 Microsoft Visual C++安装指南	5
1.3 如何构造 DOS MFC 库文件	8
1.4 如何使用 Visual Workbench	9
1.4.1 Visual Workbench 简介	9
1.4.2 Visual Workbench 命令行参数简介	10
1.5 Visual Workbench 的帮助系统	10
1.6 使用 Visual Workbench 的菜单系统	12
1.6.1 用鼠标进行菜单选择	13
1.6.2 使用键盘进行菜单选择	14
1.7 如何编辑一个程序	17
1.8 如何编译、链接和运行一个程序	19
1.9 Visual C++的浏览功能	20
1.10 如何设置工程文件选项	21
1.11 Visual C++的调试功能	22
1.12 如何使用 CL	23
1.13 如何用 CL 进行编译和链接	24
1.14 如何使用 CL 环境变量	25
1.15 CL 选项简介	25
1.16 一个 Visual Workbench 实例	31
1.17 小结	31
第二章 Visual C++程序开发工具概述	33
2.1 LINK 链接器	33
2.1.1 如何使用 LINK	34
2.1.2 LINK 的文件名选项	34
2.1.3 覆盖技术	35
2.2 LINK 选项	37
2.3 库文件管理程序 LIB	39
2.3.1 如何使用 LIB	39
2.3.2 LIB 选项及命令的用法	40
2.4 程序维护实用程序 NMAKE	41
2.4.1 建立 MAKE 文件	41
2.4.2 运行 NMAKE	44

2.5	CodeView 调试程序	45
2.5.1	如何使用 CodeView	45
2.5.2	如何启动 CodeView	46
2.5.3	程序调试的方法	46
2.6	小结	52
第三章	ANSI 标准 C 简介	54
3.1	C 程序的结构	54
3.2	ANSI C 转义序列和 trigraph	57
3.3	ANSI C 预处理器伪指令	57
3.3.1	包含文件	58
3.3.2	定义宏	58
3.3.3	条件编译伪指令	60
3.3.4	其他伪指令	60
3.4	变量的声明和定义	61
3.4.1	基本变量类型	61
3.4.2	枚举类型	62
3.5	结构、联合及位字段	62
3.5.1	数组	63
3.5.2	指针	64
3.5.3	类型的定义	64
3.5.4	类型修饰符 const 和 volatile	65
3.6	表达式	65
3.6.1	操作符的优先级	67
3.7	语句	68
3.7.1	break 语句	68
3.7.2	case 语句	69
3.7.3	复合语句或块	69
3.7.4	continue 语句	69
3.7.5	default 标记	69
3.7.6	do 语句	69
3.7.7	表达式语句	70
3.7.8	for 语句	70
3.7.9	goto 语句	70
3.7.10	if 语句	70
3.7.11	if - else 语句	71
3.7.12	null 语句	71
3.7.13	return 语句	71
3.7.14	switch 语句	71
3.7.15	while 语句	72

3.8 函数	72
3.8.1 函数原型	72
3.8.2 void 类型	73
3.8.3 带有可变个数参数的函数	73
3.9 ANSI C 库	74
3.10 小结	74
第四章 Visual C++ 对标准 C 的扩展	75
4.1 Visual C++ 特有的关键字	75
4.1.1 支持内存模式的关键字	76
4.2 内存模式的定制	77
4.2.1 其他特殊的关键字	79
4.3 全局变量及预处理器宏	80
4.4 预编译指令	83
4.5 基本数据类型的大小和容量	84
4.6 小结	85
第五章 面向对象程序设计基础	86
5.1 什么是面向对象的程序设计	86
5.1.1 面向过程的程序设计	87
5.2 面向对象程序设计技术术语	93
5.2.1 数据抽象	93
5.2.2 继承	94
5.2.3 多态性	94
5.3 C 语言中的面向对象程序设计	95
5.3.1 在 C 语言中定义对象	95
5.3.2 实现几何形状	97
5.3.3 使用图形	103
5.3.4 增加一个新图形对象	104
5.3.5 用 C 语言实现 OOP 的问题	107
5.4 小结	107
第六章 C++ 和面向对象的程序设计	109
6.1 C++ 与面向对象的程序设计	109
6.1.1 C++ 中的数据抽象	110
6.1.2 C++ 类的继承	113
6.1.3 多态和动态链接	114
6.2 用 C++ 设计几何图形	115
6.2.1 图形类	115
6.2.2 添加新的图形类	118
6.2.3 运行时创建对象	120
6.3 小结	120

第七章 C++与ANSI 标准 C 的区别	121
7.1 C++的特性	121
7.1.1 C++函数的新特性	121
7.1.2 C++和C的区别	127
7.2 小结	130
第八章 用于标准 I/O 的 C++类	131
8.1 C++中的 I/O 库	131
8.2 C++中的流 I/O	131
8.2.1 使用 iostream	132
8.2.2 使用操作符	133
8.2.3 使用操作符完成格式化 I/O	134
8.2.4 控制浮点格式	136
8.2.5 重载<<	136
8.2.6 iostream 类的层次	137
8.3 文件 I/O	138
8.3.1 简单文件 I/O	139
8.3.2 文件定位	141
8.4 字符串 I/O	143
8.4.1 向一个串写入	143
8.4.2 从一个串读取	144
8.5 小结	145
第九章 创建类的对象	146
9.1 对象与类	146
9.1.1 用户定义的数据结构	146
9.1.2 对类成员的访问控制	147
9.1.3 public 函数如何返回 private 值	148
9.1.4 成员函数	149
9.2 类的实现	150
9.2.1 头文件描述界面	150
9.2.2 从界面分离操作	152
9.3 类的使用	153
9.3.1 动态创建对象	154
9.3.2 在自由存储区中分配对象数组	155
9.3.3 调用成员函数	156
9.3.4 使用 static 成员变量	156
9.3.5 初始化 static 成员变量	159
9.3.6 使用静态成员函数	160
9.3.7 使用指向类成员的指针	161
9.4 小结	163

第十章 定义对象的操作	164
10.1 参数和返回值	164
10.1.1 理解指针和引用	164
10.1.2 值传递与引用传递的比较	165
10.1.3 返回一个引用	166
10.1.4 使用引用的准则	167
10.2 对象的创建和释放	167
10.2.1 类 String 的构造函数和析构函数	168
10.2.2 缺省构造函数	169
10.2.3 拷贝构造函数	170
10.2.4 提供一个拷贝构造函数的时机	172
10.2.5 成员初始化表	173
10.2.6 利用构造函数和析构函数的副作用	175
10.3 定义函数和操作符	176
10.3.1 指针 this	177
10.3.2 操作符作为函数	177
10.3.3 为类 String 定义 operator+	180
10.3.4 验证串是否相等	181
10.3.5 访问并改变串中的某个字符	181
10.3.6 定义类型转换操作符	182
10.3.7 为类 String 定义赋值操作符	182
10.3.8 为什么 operator= 返回一个引用	183
10.3.9 装载输入和输出操作符	183
10.3.10 装载操作符 new 和 delete	184
10.3.11 使用 friend 类	186
10.3.12 把文件当作数组来使用	186
10.4 小结	189
第十一章 在 C++ 中使用继承	190
11.1 派生类	190
11.1.1 继承和“is a”关系	190
11.1.2 继承和类扩展	190
11.1.3 派生类的语法	191
11.1.4 访问基类	192
11.1.5 使用继承来建造子串类	193
11.1.6 派生类的其他情况	200
11.1.7 多重继承	201
11.1.8 iostream 与多重继承	202
11.1.9 虚拟基类	202
11.2 使用继承	204

11.2.1 链表	205
11.2.2 single_link 类	206
11.2.3 双向链表	213
11.2.4 String 对象的队列	217
11.3 小结	219
第十二章 虚拟函数和多态性	221
12.1 动态链接	221
12.1.1 静态链接	221
12.1.2 通过指针调用函数	222
12.2 虚拟函数	223
12.2.1 纯虚拟函数	223
12.2.2 虚拟函数的具体实现	224
12.2.3 通过虚拟函数的动态链接	224
12.2.4 使用多态	226
12.2.5 隐含的类的多态用法	227
12.3 小结	229
第十三章 C++ 的高级技术	231
13.1 函数和类模板	231
13.1.1 整型和浮点型的栈	232
13.1.2 类模板	232
13.1.3 函数模板	233
13.1.4 成员函数模板	233
13.1.5 模板的优越性	234
13.2 异常处理	234
13.2.1 异常处理的优点	234
13.2.2 setjmp 和 longjmp 的问题	235
13.2.3 C++ 建议的异常处理原理	236
13.3 小结	238
第十四章 文件和目录管理函数	239
14.1 MS-DOS 文件系统	239
14.1.1 路径名	239
14.1.2 作为文件的设备	240
14.1.3 文件属性	240
14.1.4 文件句柄	241
14.2 基本文件和目录管理任务	241
14.2.1 改变驱动器和目录	242
14.2.2 改变文件属性	244
14.3 函数参考	245
14.3.1 _access	245

14.3.2	_chdir	246
14.3.3	_chdirve	247
14.3.4	_chmod	247
14.3.5	_chsize	248
14.3.6	_filelength	248
14.3.7	_fstat	249
14.3.8	_fullpath	250
14.3.9	_getcwd	250
14.3.10	_getdcwd	251
14.3.11	_getdrive	251
14.3.12	_isatty	252
14.3.13	_locking	252
14.3.14	_makepath	253
14.3.15	_mkdir	254
14.3.16	_mktemp	254
14.3.17	remove	255
14.3.18	rename	255
14.3.19	_rmdir	256
14.3.20	_searchenv	256
14.3.21	_setmode	257
14.3.22	_splitpath	257
14.3.23	_stat	258
14.3.24	_umask	258
14.3.25	_unlink	259
第十五章	流 I/O 函数	260
15.1	流	260
15.1.1	I/O 缓冲	260
15.1.2	FILE 类型	261
15.1.3	I/O 的格式化和非格式化	261
15.1.4	文件的当前位置	261
15.1.5	预定义流	262
15.1.6	字符串输入/输出	262
15.2	基本流 I/O 任务	262
15.2.1	给文件添加行号	264
15.2.2	简单的通信录	266
15.2.3	显示错误信息的简单方法	272
15.3	函数参考	273
15.3.1	clearerr	273
15.3.2	fclose	273

15.3.3	_fcloseall	274
15.3.4	_fdopen	274
15.3.5	feof	275
15.3.6	ferror	276
15.3.7	fflush	276
15.3.8	fgetc	277
15.3.9	_fgetchar	277
15.3.10	fgetpos	278
15.3.11	fgets	278
15.3.12	_fileno	279
15.3.13	_flushall	280
15.3.14	fopen	280
15.3.15	fprintf	281
15.3.16	fputc	281
15.3.17	_fputchar	282
15.3.18	fputs	282
15.3.19	fread	283
15.3.20	freopen	283
15.3.21	fscanf	284
15.3.22	fseek	285
15.3.23	fsetpos	285
15.3.24	_fsopen	286
15.3.25	ftell	287
15.3.26	fwrite	287
15.3.27	getc	288
15.3.28	getchar	288
15.3.29	gets	289
15.3.30	_getw	289
15.3.31	perror	290
15.3.32	printf	290
15.3.33	putc	292
15.3.34	putchar	293
15.3.35	puts	293
15.3.36	_putw	294
15.3.37	rewind	294
15.3.38	_rmtmp	294
15.3.39	scanf	295
15.3.40	setbuf	297
15.3.41	setvbuf	297

15.3.42	_snprintf	298
15.3.43	sprintf	298
15.3.44	sscanf	299
15.3.45	_tempnam	300
15.3.46	tmpfile	300
15.3.47	tmpnam	301
15.3.48	ungetc	301
15.3.49	vfprintf	302
15.3.50	vprintf	303
15.3.51	_vsnprintf	303
15.3.52	vsprintf	304
第十六章	低级 I/O 函数.....	305
16.1	低级 I/O 函数基础.....	305
16.1.1	文本和二进制模式.....	305
16.1.2	通过操作系统完成缓冲	305
16.1.3	文件句柄	306
16.1.4	最大文件句柄数	306
16.2	基本的低级 I/O 任务.....	306
16.2.1	拷贝文件	307
16.2.2	用 _dup 和 _dup2 重定向 stdout	309
16.3	函数参考.....	311
16.3.1	_close	311
16.3.2	_commit	311
16.3.3	_creat	312
16.3.4	_dup	312
16.3.5	_dup2	313
16.3.6	_eof	313
16.3.7	_lseek	314
16.3.8	_open	315
16.3.9	_read	316
16.3.10	_sopen	316
16.3.11	_tell	317
16.3.12	_write	318
第十七章	控制台和端口 I/O 函数.....	319
17.1	控制台和端口 I/O 基础.....	319
17.1.1	控制台和端口 I/O 的任务	319
17.1.2	用 _inp 和 _outp 产生声音	320
17.2	函数参考.....	322
17.2.1	_cgets	322

17.2.2	_cprintf	322
17.2.3	_cputs	323
17.2.4	_cscanf	323
17.2.5	_getch	324
17.2.6	_getche	324
17.2.7	_inp	325
17.2.8	_inpw	325
17.2.9	_kbhit	326
17.2.10	_outp	326
17.2.11	_outpw	327
17.2.12	_putch	327
17.2.13	_ungetch	328
第十八章	字符和数据转换函数	329
18.1	字符和数据转换基础	329
18.1.1	字符分类	329
18.1.2	转换数据	330
18.2	字符和数据转换任务	330
18.2.1	将文本字符串转换为小写	332
18.2.2	一个简单的计算器	333
18.3	函数参考	334
18.3.1	atof	334
18.3.2	atoi, atol	334
18.3.3	_atold	335
18.3.4	_ecvt, _fcvt	335
18.3.5	_gcvt	336
18.3.6	isalnum	337
18.3.7	isalpha	337
18.3.8	__isascii	337
18.3.9	isctrl, isdigit, isgraph, islower, isprint, ispunct, isspace, isupper, isxdigit	339
18.3.10	__iscsym, __iscsymf	339
18.3.11	_itoa, _ltoa	339
18.3.12	strtod	340
18.3.13	strtoul	340
18.3.14	_strtold	341
18.3.15	strtoul	342
18.3.16	__toascii	342
18.3.17	_tolower, tolower	343
18.3.18	_toupper, toupper	343
18.3.19	_ultoa	344

第十九章 串和缓冲区处理函数	345
19.1 C 中的串和缓冲区	345
19.1.1 声明串和缓冲区	345
19.1.2 字典顺序	345
19.1.3 多字节和宽位字符串	345
19.1.4 远程缓冲区和串	346
19.2 基本的串和缓冲区处理任务	346
19.2.1 拷贝视频内存	349
19.2.2 分析文本行	350
19.3 函数参考	351
19.3.1 fmblen, _fmblen	351
19.3.2 mbstowcs, _fmbstowcs	352
19.3.3 mbtowc, _fmbtowc	353
19.3.4 _memccpy, _fmemccpy	353
19.3.5 memchr, _fmemchr	354
19.3.6 memcmp, _fmemcmp	354
19.3.7 memcpy, _fmemcpy	355
19.3.8 _memicmp, _fmemicmp	356
19.3.9 memmove, _fmemmove	356
19.3.10 memset, _fmemset	357
19.3.11 _movedata	358
19.3.12 strcat, _fstrcat	358
19.3.13 strchr, _fstrchr	359
19.3.14 strcmp, _fstrcmp	359
19.3.15 _strcmpi	360
19.3.16 strcoll	361
19.3.17 strcpy, _fstrcpy	361
19.3.18 strcspn, _fstrcspn	362
19.3.19 _strdup, _fstrdup, _nstrdup	362
19.3.20 _strerror	363
19.3.21 strerror	364
19.3.22 _stricmp, _fstricmp	364
19.3.23 strlen, _fstrlen	365
19.3.24 _strlwr, _fstrlwr	365
19.3.25 strncat, _fstrncat	366
19.3.26 strncmp, _fstrncmp	366
19.3.27 strncpy, _fstrncpy	367
19.3.28 _strnicmp, _fstrnicmp	368
19.3.29 _strnset, _fstrnset	368

19.3.30	strpbrk, _fstrpbrk	369
19.3.31	strrchr, _fstrchr	370
19.3.32	_strrev, _fstrrev	370
19.3.33	_strset, _fstrset	371
19.3.34	strspn, _fstrspn	371
19.3.35	strstr, _fstrstr	372
19.3.36	strtok, _fstrtok	372
19.3.37	_strupr, _fstrupr	373
19.3.38	strxfrm	374
19.3.39	_swab	374
19.3.40	wcstombs, _fwcstombs	375
19.3.41	wctomb, _fwctomb	375
第二十章	查找和排序函数	377
20.1	查找和排序的基本任务	377
20.1.1	查找环境变量	377
20.2	函数参考	379
20.2.1	bsearch	379
20.2.2	_lfind, _lsearch	380
20.2.3	qsort	381
第二十一章	日期和时间管理函数	382
21.1	时间格式	382
21.1.1	当地时间、GMT 和 UCT	382
21.1.2	时间转换	382
21.2	基本的日期和时间管理任务	383
21.2.1	获取并打印当前的日期和时间	384
21.2.2	打印某个月的日历	384
21.3	函数参考	388
21.3.1	asctime	388
21.3.2	clock	389
21.3.3	ctime	390
21.3.4	difftime	390
21.3.5	_ftime	391
21.3.6	gmtime, localtime	391
21.3.7	mktime	392
21.3.8	_strdate	393
21.3.9	strftime	393
21.3.10	_strtime	394
21.3.11	time	395
21.3.12	_tzset	395