

Borland
C++
4.0
程序设计

方 旭 张克强 曲文路 等编

北京航空航天大学出版社

Borland C++ 4.0 程序设计

方 旭 张克强·曲文路 等编

北京航空航天大学出版社

(京)新登字 166 号

图书在版编目(CIP)数据

Borland C++ 4.0 程序设计/方旭等编. —北京:北京
航空航天大学出版社,1995.2

ISBN 7-81012-538-9

I. B… I. 方… III. C 语言-程序设计 IV. TP312C

中国版本图书馆 CIP 数据核字(95)第 01259 号

内 容 简 介

本书是针对利用 Borland C++ 4.0 进行 C 和 C++ 程序设计的人编写的,全面系统地介绍了 Borland C++ 4.0。首先介绍 Borland C++ 4.0 集成环境(IDE),说明集成编辑器和调试器的用法;然后介绍 C 和 C++ 语言的基础知识,包括操作符和表达式、说明、控制语句、函数、指针和其它高级数据结构,介绍文件输入/输出;接着介绍了类型兼容、宏、条件编译指令、预编译处理,讨论了面向对象的基本编程方法;再就是说明了处理屏幕文本和图形的技术,介绍了如何充分利用 Borland C++ 4.0 的库函数,描述了与汇编语言接口技术;此后,介绍了用 Borland C++ 4.0 进行 Windows 程序设计的技术;在附录部分给出了 Borland C++ 4.0 实用程序的功能和用法。

- 书 名: Borland C++ 4.0 程序设计
- 编 著 者: 方 旭 张克强 曲文路 等编
- 责任编辑: 曾昭奇
- 出 版 者: 北京航空航天大学出版社
- 地 址: 北京市海淀区学院路 37 号(100083)
- 印 刷 者: 朝阳科普印装
- 发 行 者: 新华书店总店科技发行所
- 经 售: 全国各地新华书店
- 开 本: 787×1092 1/16
- 印 张: 45.25
- 字 数: 1152 千字
- 印 数: 5000 册
- 版 次: 1995 年元月第 1 版
- 印 次: 1995 年元月第 1 次印刷
- 书 号: ISBN 7-81012-538-9/TP·144
- 定 价: 46.00 元

编者前言

从 Turbo C++ for Windows 1.0 开始,集成环境运行于 Windows,是一个真正的 Windows 应用程序。而 Borland C++ 从 4.0 开始,是一个很好用的 Windows 应用程序开发环境,要特别指出的是,在新的 IDE 中可在 Windows 中直接调试程序,工程管理比以前更方便。

Borland C++ 配备了 Turbo Debugger for Windows 用来调试设计的 Windows 应用程序,增强了资源包括图标、光标、位图、对话框、菜单、热键和字体等的设计。特别是在 Borland C++ 4.0 中,资源设计已得到显著增强,用来设计 Windows 应用程序的界面。Borland C++ 在 OWL(ObjectWindows Class Library)中封装了 Windows API 包括函数、数据结构和消息等,可以加快读者用 C++ 开发 Windows 应用程序速度,减轻读者用 C++ 开发 Windows 应用程序的难度。

本书是针对利用 Borland C++ 4.0 进行 C 和 C++ 程序设计的人编写的,全面系统地介绍了 Borland C++ 4.0。首先介绍 Borland C++ 4.0 集成环境(IDE),说明集成编辑器和调试器的用法;然后介绍 C 和 C++ 语言的基础知识,包括操作符和表达式、说明、控制语句、函数、指针和其它高级数据结构,介绍文件输入/输出;接着介绍了类型兼容、宏、条件编译指令、预编译处理,讨论了面向对象的基本编程方法;再就是说明了处理屏幕文本和图形的技术,介绍了如何充分利用 Borland C++ 4.0 的库函数,描述了与汇编语言接口技术;此后,介绍了用 Borland C++ 4.0 进行 Windows 程序设计的技术;在附录部分给出了 Borland C++ 4.0 实用程序的功能和用法。

目 录

编者前言

第一部分 集成环境用法

第一章 IDE 概述	2
1.1 安装 Borland C++	2
1.1.1 对硬件和软件的要求	2
1.1.2 安装步骤	2
1.1.3 启动 Borland C++	4
1.1.4 获取帮助信息	5
1.2 配置 IDE	6
1.2.1 改变 SpeedBars	10
1.2.2 设置 IDE preference	12
1.2.3 保存 IDE 的设置	12
1.3 使用编辑器	13
1.3.1 配置 IDE 的编辑器	13
1.3.2 语法制导彩色显示(Syntax Highlighting)	14
1.4 使用消息窗口(Message)工作	15
1.5 浏览代码	16
1.5.1 浏览对象的类	20
1.5.2 过滤器	20
1.5.3 查看所列符号的声明	21
1.5.4 浏览全局符号	21
1.5.5 在浏览器中使用一般通配符	22
1.5.6 浏览源代码中的符号	23
1.6 使用命令行工具	23
1.6.1 DPMI 和命令行工具	23
1.6.2 内存和 MAKESWAP.EXE	23
1.6.3 运行时间管理器和工具	24
1.6.4 控制 RTM 占用的内存	24
1.7 在 IDE 中运行其他程序	25
第二章 IDE 编辑器	28
2.1 编辑器命令	30
2.2 编辑器激活及文本键入	30
2.3 字符、字和行的删除	33
2.4 移动、拷贝文本和块移动	33
2.5 使用剪贴板	33

2.6	进一步说明如何移动光标	35
2.7	字符系列的搜索和替换	35
2.8	位置标识的设置和搜索	38
2.9	存储和装入文件	38
2.10	自动缩进	38
2.11	磁盘文件中文本块的移入和移出	39
2.12	对匹配	39
2.13	其它有关命令	40
2.14	命令综述	41
2.15	编辑器缺省的修改	44
2.16	编辑器的帮助	46
第三章	IDE 工程管理器	49
3.1	什么是工程管理器	49
3.2	建立一个工程	50
3.2.1	建立一个多任务工程	52
3.2.2	转换旧的工程	53
3.2.3	将工程转化为制作文件(makefiles)	53
3.2.4	改变工程视图	53
3.3	建立一个工程	55
3.3.1	建立工程的一部分	56
3.4	编辑工程树	57
3.4.1	使用 TargetExpert 编辑任务属性	57
3.4.2	编辑节点属性	58
3.4.3	增加和删除一个节点	59
3.4.4	增加和删除任务	60
3.4.5	移动节点和任务	62
3.4.6	拷贝节点	62
3.5	使用 Source Pool	63
3.6	设置工程选项	63
3.6.1	Local Override	64
3.6.2	使用 Style Sheet	66
3.6.3	将一个 Style Sheet 与一个节点相连	66
3.6.4	生成一个 Style Sheet	68
3.6.5	编辑 Style Sheet	68
3.6.6	共享 Style Sheet	70
3.6.7	查看工程中的选项	72
3.7	转换器	72
3.7.1	安装一个转换器	74
3.7.2	使用 SpeedMenu 中的 Special 命令	77
3.7.3	安装观察器和工具	77

第四章 编 译	79
4.1 在 IDE 中编译	79
4.1.1 使用 IDE 编译器选项	79
4.2 使用命令行编译器	80
4.2.1 配置文件	80
4.2.2 应答文件	81
4.2.3 选项的优先级规则	81
4.3 编译选项参考	81
4.4 目录选项	93
4.4.1 查找文件算法	94
4.5 编译器宏定义选项(Compiler Defines)	94
4.6 编译器代码生成选项(Compiler Code-generation)	95
4.7 编译器浮点处理选项(Compiler Floating Point)	96
4.8 编译器输出选项(Compiler Compiler Output)	96
4.8.1 编译源文件代码选项(Compiler Source)	97
4.9 编译调试选项(Compiler Debugging)	98
4.10 预编译头文件选项(Compiler Precompiled headers)	99
4.11 16 位编译处理器选项(16-bit Compiler Processor)	99
4.11.1 16 位编译器调用约定选项 (16-bit Compiler Calling Convention)	100
4.12 16 位编译存储模式选项(16-bit Compiler Memory Model)	100
4.13 16 位编译数据段命名选项(16-bit Compiler Segment Names Data)	102
4.14 16 位远程数据段命名选项(Compiler Segment Names Far Data)	103
4.15 16 位编译代码段命名选项(16-bit Compiler Segment Names Code)	103
4.15.1 16 位编译入口 出口代码选项 (16-bit Compiler Entry/Exit Code)	103
4.16 32 位编译处理器选项(32-bit Compiler Processor)	105
4.17 32 位编译调用约定选项(32-bit Compiler Calling Convention)	106
4.18 C++ 选项 成员指针(C++ Options Member Pointer)	106
4.19 C++ 兼容性选项(C++ Options C++ Compatibility)	107
4.20 C++ 虚表选项(C++ Options Virtual Tables)	108
4.21 C++ 模板生成选项(C++ Options Templates)	108
4.22 C++ 例外控制选项(C++ Options Exception handling/RTTI)	109
4.22.1 优化选项	109
4.23 优化声明选项(Optimizations Specific)	109
4.24 优化代码长度选项(Optimization Size)	111
4.25 优化速度选项(Optimization Speed)	112
4.26 消息选项(Messages)	114
4.27 移植性警告选项(Messages Portability)	114
4.28 ANSI 违例警告(Message ANSI Violations)	114

4.29 C++ 错误(Messages Obsolete C++)	115
4.30 潜在的 C++ 错误(Messages Potential C++ Errors)	115
4.31 无效 C++ 代码(Messages Inefficient C++ Coding)	116
4.32 潜在错误(Messages Potential errors)	116
4.32.1 无效代码(Messages Inefficient coding)	117
4.33 一般错误(Message General)	117
4.34 Make 选项	117
4.35 命令行选项	117
第五章 集成调试器	121
5.1 错误的类型	121
5.1.1 编译时间错误	121
5.1.2 运行时间错误	122
5.1.3 逻辑错误	122
5.2 生成调试信息	122
5.3 指明程序参数	123
5.4 控制程序执行	123
5.4.1 监视程序的输出	124
5.4.2 单步执行程序(Step over code)	124
5.4.3 跟踪执行代码	125
5.4.4 单步跳过一段程序代码	126
5.4.5 停止程序运行	127
5.4.6 重新开始	127
5.5 检查变量的值	127
5.5.1 什么是表达式	127
5.5.2 监视表达式	127
5.5.3 计算和修改表达式	130
5.5.4 检查数据元素	131
5.5.5 检查寄存器的值	132
5.6 使用断点	132
5.6.1 设置断点	133
5.6.2 使用断点工作	133
5.6.3 用户定做断点和执行点	136
5.7 解决一般保护错误	136
5.8 使用 Event Log 窗口	136
5.9 调试动态连接库	137
第六章 连接程序:TLINK	139
6.1 TLINK 基础	139
6.1.1 TLINK.CFG	140
6.1.2 响应文件	140
6.1.3 用 BCC.EXE 使用 TLINK	141

6.1.4 连接库	141
6.2 TLINK 选项	142
6.3 模块定义文件参考	148
6.3.1 CODE 语句	148
6.3.2 DATA 语句	149
6.3.3 DESCRIPTION 语句	149
6.3.4 EXETYPE 语句	149
6.3.5 EXPORT 语句	149
6.3.6 IMPORTS 语句	150
6.3.7 LIBRARY 语句	151
6.3.8 NAME 语句	151
6.3.9 SEGMENTS 语句	151
6.3.10 STACKSIZE 语句	152
6.3.11 STUB 语句	152
6.3.12 缺省的模式定义文件	152

第二部分 C 和 C++ 编程基础

第七章 C 和 C++ 基础	156
7.1 C 的发展史	156
7.1.1 与其他语言的关系	157
7.1.2 C 语言的优势	158
7.1.3 C 的弱点	160
7.1.4 编程的规程	160
7.2 ANSI C 标准	160
7.3 C++ 的进化和面向对象的编程	162
7.4 C++ 的历史	162
7.4.1 使用 C++ 对象来进行流线代码设计	163
7.4.2 对 C 的提高	163
7.4.3 对 C 的主要提高	165
7.5 C 程序的基本元素	166
7.5.1 程序的五个基本成分	167
7.5.2 第一个 C 程序	167
7.5.3 第一个 C++ 程序	168
7.5.4 第二个 C 程序	168
7.5.5 第二个 C++ 程序	172
7.5.6 文件	173
第八章 数 据	176
8.1 标识符	176
8.2 关键字	177
8.3 标准 C 和 C++ 数据类型	178

8.3.1	字 符	178
8.3.2	三种整数	180
8.3.3	unsigned 修饰符	180
8.3.4	浮点数	181
8.3.5	枚举型	182
8.4	访问修饰符	184
8.4.1	const 修饰符	184
8.4.2	#define 常量	184
8.4.3	Volatile 修饰符	185
8.4.4	const 和 volatile 一起使用	185
8.5	pascal、cdecl、near、far 和 huge 修饰符	185
8.5.1	pascal	186
8.5.2	cdecl	187
8.5.3	near、far 和 huge	187
8.6	数据类型转换	188
8.6.1	显式类型转换	189
8.7	存储类型	189
8.7.1	外层的变量说明	189
8.7.2	内层的变量说明	191
8.7.3	变量作用域	192
8.7.4	外层函数说明	192
8.8	操作符	193
8.8.1	位操作符	193
8.8.2	左移和右移	194
8.8.3	增量与减量	195
8.8.4	算术操作符	195
8.8.5	赋值操作符	196
8.8.6	复合赋值操作符	196
8.8.7	关系和逻辑操作符	198
8.8.8	条件操作符	200
8.8.9	逗号操作符	200
8.9	理解操作符优先顺序	200
8.10	标准 C 和 C++ 库函数	201
第九章	控 制	204
9.1	条件语句	204
9.1.1	if 语句	204
9.1.2	if - else 语句	205
9.1.3	嵌套 if - else	207
9.1.4	if - else - if 语句	208
9.1.5	? 条件语句	209

9.1.6	switch 语句	210
9.1.7	混合使用 if - else - if 和 switch 语句	215
9.2	循环语句	217
9.2.1	for 循环	217
9.2.2	while 循环	221
9.2.3	do - while 循环	222
9.2.4	break 语句	225
9.2.5	使用 CodeView	225
9.2.6	continue 语句	225
9.2.7	使用 Turbo Debugger	226
9.2.8	一起使用 break 和 continue	226
9.2.9	exit()语句	228
9.2.10	atexit()语句	230
第十章	编写和使用函数	232
10.1	函数原型和风格	232
10.1.1	构造原型	232
10.1.2	通过值调用和通过引用调用	234
10.1.3	存储类型	235
10.1.4	作用域	236
10.1.5	递归	236
10.2	函数参数	237
10.2.1	形式参数和实际参数	237
10.2.2	void 类型的参数	237
10.2.3	字符参数	238
10.2.4	整数参数	239
10.2.5	浮点参数	239
10.2.6	双精度参数	240
10.2.7	数组参数	242
10.3	函数类型	243
10.3.1	void 函数类型	243
10.3.2	char 函数类型	244
10.3.3	int 函数类型	245
10.3.4	long 函数类型	245
10.3.5	float 函数类型	246
10.3.6	double 函数类型	247
10.4	main()的参数	248
10.4.1	字符串	248
10.4.2	整数	249
10.4.3	浮点数	250
10.5	重要的 C++ 特性	251

10.5.1	直接插入	251
10.5.2	重载	252
10.5.3	省略号(...)	253
10.6	作用域规则遇到的问题	254
10.6.1	C 程序中一个未定义的符号	254
10.6.2	使用具有文件作用域的变量	255
10.6.3	用局部变量屏蔽全局变量	256
10.6.4	C++ 的作用域问题	256
10.6.5	C++ 作用域限定操作符	257
第十一章	数 组	259
11.1	什么是数组	259
11.2	数组和 C	259
11.3	数组说明	259
11.4	数组初始化	260
11.4.1	缺省初始化	260
11.4.2	显示初始化	261
11.4.3	不定长初始化	261
11.5	访问数组元素	262
11.6	计算数组大小(sizeof())	263
11.7	数组下标越界	265
11.8	字符串输入输出	265
11.9	多维数组	267
11.10	数组作为函数参数	270
11.10.1	将数组传给 C 函数	270
11.10.2	数组传给 C++ 函数	272
11.11	字符串函数和字符数组	277
11.11.1	gets(), puts(), fgets(), fputs() 和 sprintf()	277
11.11.2	strcpy, strcat, strcmp, strlen	278

第三部分 C++ 面向对象编程基础

第十二章	指 针	284
12.1	定义指针变量	284
12.1.1	指针变量说明	284
12.1.2	简单语句使用指针变量	286
12.1.3	指针变量初始化	288
12.1.4	地址操作符的错误使用	289
12.1.5	数组指针	290
12.1.6	指针的指针	291
12.1.7	串的指针	292
12.1.8	指针运算	293

12.1.9	指针运算和数组	295
12.1.10	操作符++和--的有关事项	296
12.1.11	比较指针	296
12.1.12	可移植性	297
12.1.13	指针的大小	297
12.2	函数的指针	299
12.3	动态存储	301
12.3.1	使用空指针	303
12.4	指针和数组	306
12.4.1	串(字符型的数组)	306
12.4.2	指针数组	307
12.4.3	更多的有关指针的指针的内容	309
12.4.4	串指针的数组	314
12.5	C++引用类型	316
12.5.1	函数返回地址	316
12.5.2	使用 Turbo Debugger	317
12.5.3	何时使用引用类型	317
第十三章	C 语言中的 I/O	319
13.1	流函数	321
13.1.1	打开流	321
13.1.2	输入和输出的重定向	322
13.1.3	改变流缓冲区	323
13.1.4	关闭流	324
13.2	C 中的低级输入和输出	324
13.3	字符输入和输出	325
13.3.1	使用 getc()、putc()、fgetc()和 fputc()	325
13.3.2	使用 getchar()、putchar()、fgetchar()和 fputchar()	326
13.3.3	使用 getch()和 putch()	326
13.4	串输入和输出	327
13.4.1	使用 gets()、puts()、fgetx()和 fputs()	327
13.5	整数输入和输出	328
13.5.1	使用 getw()和 putw()	328
13.6	格式化输出	330
13.6.1	使用 printf()和 fprintf()	330
13.7	使用 fseek()、ftell()和 rewind()	334
13.7.1	使用 Turbo Debugger	337
13.8	格式化输入	338
13.8.1	使用 scanf()、fscanf()和 sscanf()	339
第十四章	C++的输入、输出介绍	341
14.1	使用 C++精简 I/O	341

14.1.1	cin、cout 和 cerr	341
14.1.2	抽取(>>)和插入(<<)操作	341
14.2	从 stream.h 到 iostream.h	347
14.2.1	操作符和成员函数	348
第十五章	结构、联合和其它数据结构	361
15.1	C 和 C++ 结构	361
15.1.1	C 和 C++ 结构:语法和规则	361
15.1.2	C++ 结构:语法和规则的扩展	363
15.1.3	访问结构成员	363
15.1.4	构造一个简单的结构	364
15.1.5	将结构传递给函数	365
15.1.6	构造一个结构数组	366
15.1.7	使用结构指针	369
15.1.8	传递结构数组给函数指针	371
15.1.9	C++ 中结构的使用	373
15.1.10	对结构的附加操作	376
15.2	联合	377
15.2.1	联合:语法和规则	377
15.2.2	构造一个简单的联合	378
15.3	其它数据类型	379
15.3.1	使用 typedef	380
15.3.2	使用枚举类型	381
第十六章	高级 C 和 C++ 编程专题	384
16.1	类型兼容	384
16.1.1	ANSI C 对于类型兼容的定义	384
16.1.2	什么是同一类型	384
16.1.3	枚举类型	385
16.1.4	数组类型	385
16.1.5	函数类型	386
16.1.6	结构和联合类型	386
16.1.7	指针类型	386
16.1.8	多源文件的相容性	386
16.2	宏	386
16.2.1	定义宏	387
16.2.2	宏和参数	387
16.2.3	宏扩展的问题	388
16.2.4	生成和使用自己的宏	389
16.2.5	编译器预置的宏	390
16.3	高级预处理语句	390
16.3.1	#ifdef 和 #endif	390

16.3.2	#undef	391
16.3.3	#ifndef	391
16.3.4	#if	391
16.3.5	#else 指令	392
16.3.6	#elif 指令	392
16.3.7	#line 指令	392
16.3.8	#error 指令	393
16.3.9	#pragma 指令	393
16.4	条件编译	393
16.5	高级预处理操作	394
16.5.1	#stringize operator(串化操作)	394
16.5.2	连接操作(##)	394
16.5.3	字符操作(#@)	395
16.6	头文件的正确使用	395
16.7	有效地使用头文件	396
16.8	预编译的头文件	397
16.8.1	产生预编译的头文件	397
16.8.2	在PWB(程序员工作台)上使用预编译头文件	397
16.9	limits.h 和 float.h	397
16.10	错误处理:perror()	399
16.11	存储模式	399
16.11.1	Tiny(微)	400
16.11.2	small(小)	400
16.11.3	medium(中)	400
16.11.4	Compact(紧缩)	400
16.11.5	Large(大)	400
16.11.6	Huge(巨)	400
16.12	动态存储分配:链表	401
16.12.1	使用链表时需要考虑的事项	402
16.12.2	一个简单的链表	402
第十七章	面向对象的编程介绍	406
17.1	并非是新内容	406
17.2	传统结构化设计	406
17.3	面向对象的程序设计	407
17.3.1	C++和面向对象设计	407
17.4	面向对象的术语	408
17.4.1	封装	408
17.4.2	类的层次性	409
17.5	认识C++类	410
17.5.1	一个作为私有类的结构	410

17.5.2	C++类的语法和规则	414
17.5.3	一个简单的C++类	415
第十八章	C++类	419
18.1	附加的类特性	419
18.1.1	一个简单的类	419
18.1.2	嵌套类	420
18.1.3	构造函数和析构函数	423
18.1.4	重载类的成员函数	429
18.1.5	使用友元函数来存取私有类变量	432
18.1.6	使用 this 指针	435
18.2	使用操作符重载	435
18.2.1	操作符重载和函数调用	436
18.2.2	重载的语法	436
18.3	派生类	438
18.3.1	派生类的语法	439
18.3.2	产生派生类	439
第十九章	C++的 I/O	444
19.1	枚举类型	444
19.2	引用变量	444
19.3	缺省参数	446
19.4	memset()	447
19.5	格式化的输出	448
19.6	C/C++ I/O 选择	452
19.7	iostream 类列表	452
19.7.1	输入流类	455
19.7.2	输出流类	456
19.7.3	缓冲流类	458
19.7.4	串流类	459
19.8	二进制文件	461
19.9	使用 extern "C" 来结合 C 和 C++ 代码	463
19.10	编写自己的操纵符	465
19.10.1	无参数的操纵符	465
19.10.2	带有一个参数的操作器	466
19.10.3	多参数的操纵符	467
第二十章	基于面向对象环境的工作	470
20.1	C++中一个面向的对象的栈	470
20.2	C++中面向对象的连接表	472
20.2.1	父类的产生	473
20.2.2	派生子类	473
20.2.3	使用友元类	475

20.2.4 测试一个完整的程序.....	477
20.2.5 连接表输出.....	485
20.3 面向对象 C++ 的其它内容.....	487

第四部分 DOS 图形、系统访问库及混合语言接口

第二十一章 屏幕文本和图形程序设计.....	490
21.1 图形系统和要素.....	490
21.1.1 视频模式.....	490
21.1.2 窗口和视区.....	490
21.1.3 在文本模式下编程.....	491
21.1.4 在图形模式下编程.....	495
21.2 Borland C++ 图形程序设计.....	505
21.2.1 基本正文模式函数.....	505
21.2.2 Borland C++ 的图形子系统简介.....	514
21.3 IBM/PC 的文本方式.....	521
21.3.1 PC 显示器适配器和屏幕.....	522
21.3.2 视频缓冲区 I/O.....	522
21.4 控制文本屏幕.....	523
21.4.1 使用文本方式控制函数.....	523
21.4.2 使用直接控制台 I/O 以获得高性能.....	526
21.5 使用窗口函数.....	526
21.6 了解 IBM-PC 的图形方式.....	530
21.6.1 像素点与调色板.....	530
21.6.2 控制图形屏幕.....	532
21.7 介绍 BGI 图形库.....	532
21.7.1 使用画图和填充函数.....	532
21.7.2 控制屏幕和视口.....	537
21.8 在图形方式下显示文本.....	538
21.8.1 BGI 字库.....	538
21.8.2 使用图形方式下的文本函数.....	538
第二十二章 更具威力的编程:选择重要的 C 和 C++ 库.....	541
22.1 Turbo C 和 C++ 头文件.....	541
22.2 标准库函数(stdlib.h).....	542
22.2.1 实现数据转换.....	543
22.2.2 实现查找和排序.....	544
22.2.3 混合操作.....	546
22.3 字符函数 ctype.h).....	548
22.3.1 查测是否是字母/数字、字母、ASCII 值.....	549
22.3.2 检测是否是控制、空白和标点字符.....	550
22.3.3 转换为 ASCII 码、小写字母、大写字母.....	552