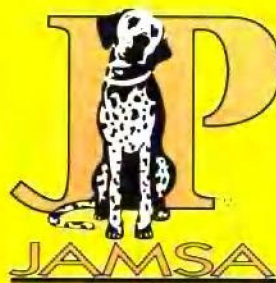




万水计算机技术实用大全系列



C/C++ 程序员实用大全

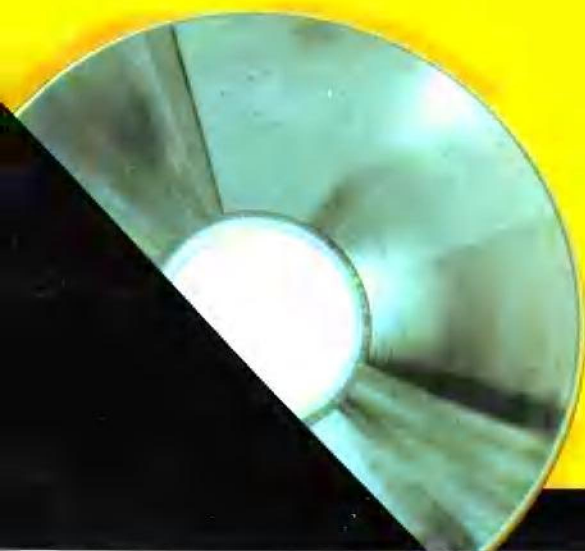
——C/C++ 最佳编程指南

Jamsa's C/C++ Programmer's Bible
The Ultimate Guide to C/C++ Programming

[美] Kris Jamsa, Lars Klander 著

张春晖 刘大庆 等译

八道湾创作室 审校



中国水利水电出版社

www.waterpub.com.cn

万水计算机技术实用大全系列

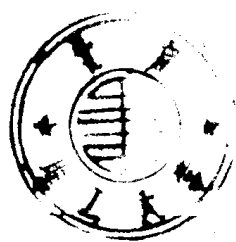
C/C++程序员实用大全

——C/C++最佳编程指南

[美] Kris Jamsa, Lars Klander 著

张春晖 刘大庆 等译

八道湾创作室 审校



964602

有光

中国水利水电出版社

内 容 提 要

本书通过对大量程序实例的分析,深入浅出地讲解了 C/C++语言基本知识、编译方法和程序设计技术,并给出了众多的编程技巧与应用。全书首先介绍了 C 语言的基本知识,对 C 语言编程的实际方法、技巧,以及常见的问题和难点进行了详细阐述和解答。然后逐步过渡到 C++编程与 Windows 编程,用较大的篇幅对它们进行了深入地剖析。无论是 C/C++的初学者还是资深程序员都能从中受益匪浅。

本书是广大程序员和编程自学者学习和掌握 C/C++编程的优秀指南。

“Copyright © 1998 by Jamsa Press, All rights reserved. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any format or by any means, or stored in a database or retrieval system, without the prior written permission of Jamsa Press.”

北京市版权局著作权合同登记号: 图字 01-98-1561 号

书 名	C/C++程序员实用大全——C/C++最佳编程指南
作 者	[美] Kris Jamsa, Lars Klander 著
译 者	张春晖 刘大庆 等译
审 校	八道湾创作室
出版、发行	中国水利水电出版社(北京市三里河路6号 100044) 网址: www.waterpub.com.cn E-mail: sale@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	水利电力出版社印刷厂
规 格	787×1092毫米 16开本 61.75印张 1428千字
版 次	1999年7月第一版 1999年7月北京第一次印刷
印 数	0001—4000册
定 价	100.00元(ICD,含配套书)

凡购买我社图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·侵权必究

译 者 序

在本书的翻译过程中，我们常常为作者对 C/C++ 编程的深入掌握和独到见解而惊讶、赞叹，对我们自己而言，这也是一个学习与提高的过程，同时也为能向国内读者推荐这本好书而高兴。通过认真阅读这本优秀的 C/C++ 编程指南，读者将逐渐成为 C/C++ 的编程大师。全书先后讲述了 C 语言编程、C++ 语言编程、Windows 编程的知识。

C 语言编程

近年来，C 语言无论在国内还是在外国都得到了迅速地推广。C 语言以它的功能丰富、表达能力强、使用灵活、应用面广、目标程序效率高、可移植性好、能对硬件直接进行操作等优点愈来愈赢得人们的青睐。

C 语言功能强、使用灵活，但是普遍认为它比较难学，学习和使用 C 语言需要有一定的硬件和软件基础知识，不熟练的程序员常常出错而不知其所以然。针对这种情况，本书从 C 语言的基础知识开始，然后讲述了 C 语言中的宏与常量、各种变量类型、函数、键盘操作、文件与磁盘操作、各种构造类型的数据结构、DOS 和 BIOS 服务、内存管理、重定向 I/O、编程工具、高级 C 语言编程等知识，使读者很快地掌握 C 语言的核心技术进而全面而深入地掌握 C 语言编程的奥妙之处。

C++语言编程

面向对象的编程是当前的热门课题，C++ 语言作为 C 语言的超集，对面向对象的程序设计提供了强有力的支持，受到了越来越多的程序设计员的青睐。但是面对一大堆 C++ 程序员手册、编程指南等参考资料，学习了 C++ 语言和面向对象程序设计的思想，却不知如何下手编程，无法用类来实现现实的模型。甚至有些程序员仍然用结构化的程序设计思想来编写 C++ 程序，有些程序员设计的类及类构造明显违背了面向对象程序设计的原则，没有发挥出 C++ 潜在的威力。本书正是基于上述的考虑，为培养最优秀的 C++ 程序员而编写的，所讲的编程技巧是作者丰富的经验和体会的结晶。主要内容有：I/O 流和可流类的奥秘；函数模板和类模板的探讨；对象构造和对象析构方面的技术；详细叙述了基于类模板的标准模板类库的使用方法。本书是一本实用手册，它用许多程序和工程实例论述了 C++ 编程的精髓，初学者可以从书中了解到面向对象程序设计的思想；通过套用现成的编程方法，编写出合乎面向对象程序设计原则的程序，少走弯路。对于已有一定 C++ 编程经验的读者，本书会使他们对 C++ 特性及实现有更深入的理解，在编程中更充分地发挥 C++ 的潜在威力。

Windows 编程

Windows 的 32 位操作系统增加了许多新的功能和特性，特别是，32 位的寻址能力完全打破了原来的 640KB 段寻址方式的限制，所以大大提高了应用程序的运行效率和运行速度。本书除了介绍 Windows 的基本编程技巧外，还着重介绍了 32 位操作系统提供的新功能的编程技巧，内存管理、多线程、异步 I/O 等的编程技巧。本书内容丰富，例证充实，每一个技巧都给出了编程人员最关心的问题，并进行了详细的说明和演示，而且所涉及的内容相当广泛，编程人员几乎可以在本书中找到他想利用的任何 Windows 编程主题，通过本书可以使以前感到非常神秘而难以理解的问题，现在看来是那么自然。本书使读者能充分发挥 32 位操作系统的威力。

本书主要由张春晖和刘大庆负责翻译，八道湾创作室审校。参加本书翻译工作的还有：王可欣、刘丽娜、晏飞、王军、赵力、陈京琼、陈昊、沈娟、周兵、杜红英、李越、韩素华、张戈、赵伟、闫智华、晏宝树、王高阳、周春元、窦光瓚、黄阳、吴启华、杜修力、付朝阳、王铁龙、何靖、张晓丹、袁文英、何为、张相文、谢衍球、张春玉、张春霞等。本书翻译过程中得到了孙春亮与闫红梅编辑的大力帮助与支持，在此表示感谢！

由于译者水平有限，错误之处在所难免，恳请广大读者批评指正。

译者

1999 年 5 月

目 录

译者序

第一章 C 语言入门 1

1 编程简介	1
2 创建 ASCII 码源文件	2
3 编译 C 程序	2
4 语法错误	3
5 典型的 C 程序结构	4
6 往程序里添加语句	4
7 在新的一行上显示输出结果	5
8 C 语言区分大小写字母	5
9 逻辑错误 (BUG)	6
10 程序开发过程	6
11 文件类型	7
12 进一步了解连接器	7
13 头文件	8
14 帮助编译器查找头文件	8
15 加速编译	9
16 注释程序	9
17 提高程序的可阅读性	10
18 注意编译器警告信息	10
19 控制编译器警告	11
20 用注释屏蔽语句	11
21 名字的重要性	11
22 分号的作用	12
23 变量	12
24 给变量赋值	13
25 变量类型	13
26 定义同一类型的多个变量	13
27 定义变量时加上注释	13
28 给变量赋初值	14
29 在定义时初始化多个变量	14
30 使用有意义的变量名	14
31 C 关键字	14

32 整型变量	15
33 字符型变量	15
34 浮点型变量	16
35 双精度型变量	16
36 给浮点型变量赋值	17
37 类型标识符	17
38 无符号类型标识符	17
39 LONG (长) 类型标识符	18
40 联合使用无符号和长类型标识符	19
41 使用大数值	19
42 寄存器类型标识符	19
43 短 (SHORT) 类型标识符	20
44 从类型申明中去掉 INT	20
45 有符号类型标识符	20
46 多赋值运算符	21
47 把变量的值赋给另一种类型的变量	21
48 创建用户自己的类型	21
49 赋给十六进制或八进制值	22
50 溢出	22
51 精确度	23
52 赋值为引号或其他字符	24
53 PRINTF 入门	24
54 使用 PRINTF 显示整型数值	25
55 打印八进制或十六进制整数	25
56 用 PRINTF 显示无符号整型数值	26
57 用 PRINTF 显示长整型数值	26
58 用 PRINTF 显示浮点型数值	27
59 用 PRINTF 显示字符型数值	27
60 用指数格式显示浮点数	27
61 显示浮点数	28
62 用 PRINTF 显示字符串	28
63 用 PRINTF 显示指针地址	29

64	在数值前添加正号和负号	29	100	简单语句和复杂语句	47
65	用 PRINTF 格式化整数值	29	101	判断是否相等	48
66	0 填充整数输出	30	102	关系判断	49
67	在八进制和十六进制数前显示前缀	31	103	用逻辑与判断两个条件.....	49
68	用 PRINTF 格式化浮点数	31	104	用逻辑或来判断两个条件.....	50
69	格式化指数输出	31	105	逻辑非运算	50
70	左对齐 PRINTF 的输出	32	106	将条件结果赋值给变量.....	51
71	联合使用格式符	32	107	在复合语句中定义变量.....	51
72	字符串的换行	33	108	使用缩进来提高程序的可读性...	52
73	显示 NEAR 和 FAR 字符	33	109	使用扩展 CTRL+BREAK 检查	53
74	使用 PRINTF 的转义字符	33	110	判断浮点数	53
75	判断 PRINTF 已显示的字符数目	34	111	永远循环下去	53
76	使用 PRINTF 的返回值	34	112	赋值判断	54
77	使用 ANSI 设备驱动器	35	113	IF-IF-ELSE 语句	54
78	用 ANSI 驱动器清除屏幕显示 ...	35	114	按规定次数执行语句	55
79	用 ANSI 驱动器显示屏幕颜色 ...	35	115	FOR 语句的有些部分是可选择的	56
80	用 ANSI 驱动器定位光标	36	116	在 FOR 语句中减值	57
81	在 C 中作基本的数学运算	36	117	控制 FOR 循环的增值	57
82	模运算(取余运算)	37	118	在 FOR 循环中使用字符型和浮 点型数值	58
83	运算符的优先级和结合性	38	119	空循环	58
84	强制操作符运算顺序	38	120	无穷循环	59
85	C 的自增运算符	39	121	在 FOR 循环中使用逗号运算符	59
86	C 的自减运算符	40	122	不要在 FOR 循环中改变控制变 量的值	60
87	按位或运算	41	123	用 WHILE 循环重复执行一条或 多条语句	60
88	按位与运算	41	124	WHILE 循环的组成部分	61
89	按位异或运算	42	125	使用 DO 重复执行一条或多条语 句	62
90	“取反”运算	43	126	C 的 CONTINUE 语句	63
91	对变量的值进行运算	43	127	使用 C 的 BREAK 语句来结束循 环	64
92	C 的条件运算符	44	128	GOTO 语句分支	65
93	C 的长度(SIZEOF)运算符 ...	44	129	判断多个条件	65
94	移位运算	45	130	在 SWITCH 中使用 BREAK ...	66
95	位循环运算	46	131	使用 SWITCH 语句的 DEFAULT	
96	条件运算符	46			
97	循环控制	47			
98	C 如何表示真(TRUE)和假 (FALSE)	47			
99	用 if 判断条件	47			

CASE.....	67	165 在字符串常量内表示引号.....	82
第二章 宏与常量	69	166 判断字符串的长度	83
132 在程序中定义常量	69	167 使用 STRLEN 函数	83
133 宏与常量扩展	69	168 将一个字符串的字符复制到另 一个字符串中	84
134 给常量和宏命名	70	169 将一个串的内容追加到另一个串 上	85
135 使用_FILE_预处理器常量	70	170 给字符串追加 N 个字符	85
136 使用_LINE_预处理器常量	70	171 把一个字符串转移到另一个字 串	86
137 改变预处理器的行计数	71	172 不要越过字符串的界限.....	86
138 生成无条件预处理器错误	72	173 判断两个字符串是否相同.....	87
139 其他预处理器常量	72	174 比较字符串时忽略大小写.....	87
140 记录预处理器的日期和时间 ...	72	175 将字符串转换成大写或小写.....	88
141 判断是否进行 ANSI C 编译.....	72	176 获取字符串中第一次出现的某个 字符	89
142 判断是 C++还是 C	73	177 返回索引到串的首次出现.....	89
143 取消宏或常量	73	178 搜索字符在字符串中的末次出现	90
144 比较宏与函数	73	179 返回指向字符串末次出现的索引	90
145 编译器 PRAGMAS	74	180 使用 FAR 字符串	91
146 预定义值和宏	74	181 为 FAR 字符串编写字符串函数	91
147 创建用户自己的头文件	74	182 计算字符串中字符出现的次数...	91
148 使用#include<FILENAME.H> 或#include"FILENAME.H"	74	183 将字符串的内容反转	92
149 判断符号是否被定义	75	184 将某特定字符赋给整个字符串...	92
150 进行 IF-ELSE 预处理	75	185 比较两个字符串	93
151 更强大的预处理器条件判断 ...	76	186 比较两个字符串的前 N 个字符	93
152 实现 IF-ELSE 和 ELSE-IF 预处 理	76	187 不考虑大小写比较字符串.....	94
153 定义需要多行的宏和常量	77	188 将字符串转换成数字	95
154 创建自定义宏	77	189 复制字符串的内容	95
155 在宏定义中不要放置分号	78	190 从给定字符序列中查找字符的 首次出现	96
156 创建 MIN 和 MAX 宏	78	191 在字符串中查找子字符串.....	97
157 创建 SQUARE 和 CUBE 宏.....	78	192 计算子字符串出现的次数.....	97
158 注意宏定义中的空格	79	193 给子字符串获取索引	98
159 如何使用括号	79	194 获取子字符串的最右端出现.....	98
160 宏是没有类型的	80	195 不使用%S 格式标识符显示字符	
第三章 字符串	81		
161 C 字符串的形象化	81		
162 编译器是如何表示字符串的 ...	81		
163 C 是如何存储字符串的	81		
164 'A'是如何区别于“A”的	82		

串	98	230 局部变量	114
196 从字符串中删除子字符串	99	231 函数如何使用堆栈	115
197 用另一个子字符串代替子字符串	99	232 函数的开销	115
198 转换数值的 ASCII 码形式	100	233 C 如何存储局部变量.....	116
199 判断字符是否为字母数字	100	234 申明全局变量	116
200 字符是否为字母	101	235 避免使用全局变量	117
201 判断字符是否包含 ASCII 值 ...	101	236 解决全局和局部变量的名称冲突	117
202 判断字符是否为控制符	101	237 更好地定义全局变量的有效范围	118
203 判断字符是否为数字	101	238 传值调用	118
204 判断字符是否为图形字符	102	239 使用传值调用防止参数值变化...	119
205 判断字符是大写还是小写	102	240 传址调用	120
206 判断字符是否可打印	102	241 获取地址	120
207 判断字符是否为标点符号	102	242 使用变量的地址	121
208 判断字符是否包含空白 (whitespace) 符	102	243 改变参数的值	122
209 判断字符是否为十六进制值 ...	103	244 只改变指定参数	122
210 将字符转换成大写形式	103	245 使用堆栈进行传址调用.....	123
211 将字符转换成小写形式	104	246 记住函数变量的值	123
212 使用 ASCII 字符	104	247 C 是如何初始化静态变量的.....	124
213 将输出格式写进字符串变量 ...	104	248 使用 PASCAL 调用顺序	125
214 从字符串中读输入	105	249 PASCAL 关键字的影响	125
215 标志字符串以节省空间	105	250 混合编程示例	126
216 初始化字符串	106	251 CDECL 关键字	126
第四章 函 数	107	252 递归函数	126
217 函数	107	253 递归阶乘函数	127
218 在函数中使用变量	108	254 另一个递归的例子	128
219 把 MAIN 当作函数	108	255 进一步理解递归	128
220 参数简介	109	256 直接递归与间接递归	129
221 使用多参数	109	257 判断是否要使用递归	129
222 老式 C 程序中的参数申明	110	258 为什么递归函数慢	130
223 函数返回值	110	259 如何消除递归	131
224 RETURN 语句	111	260 将字符串传递给函数	131
225 函数原型	112	261 传递指定的数组元素	131
226 运行时程序库	113	262 形式参数中的 CONST	132
227 形参和实参	113	263 使用 CONST 不会阻止参数值的 修改	132
228 解决名称冲突	114	264 无界字符串的申明	133
229 返回类型为非 INT 型的函数 ...	114		

265	指针的使用与字符串的申明	133	298	快速格式化键盘输入	151
266	C 是如何使用堆栈处理字符串参 数的	134	299	写字符串	151
267	外部变量	134	300	使用直接 I/O 实现更快的字符串 输出	152
268	应用外部变量	134	301	从键盘读入字符串	152
269	外部静态变量	135	302	以更快的速度从键盘输入字符串	153
270	VOLATILE 关键字	136	303	以彩色显示输出	153
271	调用结构和基指针	136	304	清除屏幕显示	154
272	调用汇编语言函数	137	305	删除当前行到行尾的内容	154
273	从汇编语言函数中返回值	138	306	删除屏幕上的当前行	154
274	没有返回值的函数	139	307	定位光标进行屏幕输出	155
275	不使用参数的函数	139	308	判断行与列的位置	155
276	AUTO 关键字	139	309	在屏幕上插入空行	155
277	范围	139	310	将屏幕上的文本拷贝到缓冲区	156
278	范围的分类	140	311	将缓冲区中的文本拷贝到屏幕的 指定位置	157
279	名称空间和标识符	140	312	判断文本模式设置	157
280	标识符的可见性	140	313	控制屏幕颜色	158
281	DURATION	141	314	指定背景色	159
282	支持参数个数可变的函数	141	315	使用 TEXTCOLOR 设置前景色	160
283	支持个数可变的参数	141	316	使用 TEXTBACKGROUND 设 置背景色	160
284	VA_START、VA_ARG 和 VA_END 是如何工作的	142	317	控制文本的明暗度	161
285	创建支持多参数多类型的函数	143	318	决定当前文本模式	161
第五章	键 盘 操 作	145	319	在屏幕上移动文本	162
286	从键盘读入字符	145	320	定义文本窗口	162
287	显示字符输出	145	第六章	数 学	164
288	缓冲输入	146	321	使用整型表达式的绝对值	164
289	将键盘输入赋给字符串	146	322	使用 ARCCOSINE (反余弦)	164
290	联合使用 GETCHAR 和 PUTCHA	146	323	使用 ARCSINE (反正弦)	165
291	记住, GETCHAR 和 PUTCHAR 都是宏	147	324	使用 ARCTANGENT (反正切)	165
292	使用直接 I/O 读入字符	147	325	求复数的绝对值	165
293	不显示字符的直接键盘输入	148	326	对浮点值进位舍入	166
294	知道何时使用 'R' 和 'N'	148	327	使用角的余弦	166
295	直接输出	149	328	使用角的双曲余弦	167
296	将按键放回键盘缓存	149	329	使用角的正弦	167
297	使用 CPRINTF 快速格式化输出	150			

330	使用角的双曲正弦	167	365	文件流	187
331	使用角的正切	168	366	文件翻译	187
332	使用角的双曲正切	168	367	CONFIG.SYS 文件的 FILES=条 目	187
333	整数相除	169	368	使用低级和高级文件 I/O	188
334	使用指数	169	369	文件句柄 (FILE HANDLES) ...	188
335	使用浮点型表达式的绝对值 ...	170	370	进程文件表 (PROCESS FILE TABLE)	188
336	使用浮点余数	170	371	进程文件表入口	189
337	使用浮点值的尾数和指数	170	372	系统文件表	190
338	计算 $X*2E$ 的结果	171	373	显示系统文件表	191
339	计算自然对数	172	374	从流指针中导出文件句柄.....	192
340	计算 $\log_{10}x$ 的值	172	375	进行格式化文件输出	193
341	判断最大值与最小值	173	376	重命名文件	193
342	把浮点值分解成整数和小数部分	173	377	删除文件	194
343	计算 x^n 的结果	173	378	判断程序如何访问文件.....	195
344	计算 10^x 的结果	174	379	设置文件的访问模式	196
345	生成随机数	174	380	深入掌握文件属性	196
346	将随机值映射到指定范围	175	381	检测文件流错误	198
347	给随机数生成器赋初值	175	382	判断文件的长度	199
348	计算数值的平方根	176	383	刷新 I/O 流.....	199
349	创建定制的数学错误处理程序	176	384	一次关闭所有被打开的文件.....	200
第七章	文件、目录和磁盘	178	385	获取文件流的文件句柄.....	200
350	判断当前盘驱动器	178	386	使用 P_TMPDIR 创建临时文件名	201
351	选择当前驱动器	178	387	使用 TMP 或 TEMP 创建临时文 件名	201
352	判断可用的盘空间	178	388	创建真正的临时文件	202
353	当心 DBLSPACE	179	389	删除临时文件	202
354	读入文件分配表 (FAT) 信息	179	390	为文件搜索命令路径	202
355	磁盘 ID	180	391	为文件搜索环境入口的子目录...	203
356	绝对扇区读写操作	181	392	打开 TEMP 目录中的文件	204
357	进行 BIOS 磁盘 I/O	181	393	最小化文件 I/O 操作.....	204
358	测试软驱是否准备好	183	394	在目录名中使用反斜杠.....	205
359	应用 FOPEN 打开文件.....	183	395	改变当前目录	205
360	FILE 结构	184	396	创建目录	206
361	关闭一个打开的文件	185	397	删除目录	206
362	每次读 / 写文件信息的一个字符	185	398	删除目录树	206
363	文件指针的位置指针	186			
364	判断当前文件位置	186			

399	建立完全路径名	208	434	反绕目录	233
400	分解目录路径	209	435	递归读取磁盘文件	233
401	建立路径名	210	436	判断当前文件位置	234
402	使用低级函数打开和关闭文件	210	437	打开共享文件流	235
403	创建文件	211	438	在指定目录中创建唯一文件	236
404	进行低级读写操作	212	439	创建新文件	236
405	判断文件是否结束	212	440	利用 DOS 服务访问文件	237
406	应用低级文件例程序	213	441	强制二进制或文本文件打开	239
407	为文件句柄翻译指定模式	213	442	按行读文本	239
408	利用 LSEEK 定位文件指针	214	443	按行写文本	239
409	打开多于 20 个的文件	214	444	应用 FGETS 和 FPUTS	240
410	使用 DOS 文件服务	215	445	强制二进制文件翻译	240
411	获取文件的日期和时间标记	215	446	为什么 TEXTCOPY 不能拷贝 二进制文件	240
412	利用位域获取文件的日期与时间	217	447	判断文件结尾	241
413	设置文件的日期与时间标记	218	448	舍弃字符	241
414	把文件日期和时间设置成当前 日期和时间	219	449	读取格式化的文件数据	241
415	每次读写一个字	219	450	根据当前位置定位文件指针	242
416	改变文件的长度	220	451	获取文件句柄信息	243
417	控制文件打开操作的读写模式	221	452	重新打开文件流	243
418	将缓冲区赋给文件	222	第八章 数组、指针和结构	245	
419	分配文件缓冲区	223	453	数组	245
420	利用 MKTEMP 创建唯一文件名	224	454	申明数组	245
421	读写结构	224	455	形象表示数组	245
422	从文件流中读取结构数据	225	456	数组的内存需求	246
423	复制文件句柄	226	457	初始化数组	247
424	强制文件句柄设置	227	458	访问数组元素	247
425	把文件句柄和文件流联系起来	228	459	通过循环访问数组元素	248
426	文件共享	228	460	使用常量定义数组	248
427	打开文件进行共享访问	228	461	把一个数组传送给函数	249
428	锁定文件内容	229	462	把数组看作函数	249
429	获取更精细的文件锁定控制	231	463	区分字符串数组	250
430	使用 DOS 目录	232	464	在堆栈中传送数组	250
431	打开目录	232	465	判断数组能存放多少个元素	250
432	读取目录入口	233	466	为大数组使用 HUGE 内存模式	251
433	利用目录服务读 C: \WINDOWS	233	467	权衡数组与动态存储的利弊	251
			468	多维数组	252

469	行与列	252	505	利用 QSORT 对数组排序	274
470	访问二维数组中的元素	253	506	判断数组元素的个数	275
471	给二维数组元素赋初值	253	507	把指针理解为地址	276
472	判断多维数组占用的内存	254	508	判断变量的地址	276
473	通过循环显示二维数组	254	509	C 是如何把数组当成指针的	276
474	遍历三维数组	255	510	对数组应用取地址运算符(&) ...	277
475	初始化多维数组	255	511	申明指针变量	277
476	把二维数组传送给函数	256	512	间接访问指针	278
477	把多维数组当作一维数组	257	513	使用指针值	278
478	C 是如何存放多维数组的	257	514	指针与函数参数的使用	279
479	按行存放与按列存放	258	515	指针运算	279
480	以数组为成员的结构数组	258	516	指针的增值与减值	279
481	联合	259	517	联合应用指针引用与增值	280
482	使用联合节省内存	260	518	利用指针遍历数组	280
483	使用 REGS —— 一种典型的联合	261	519	使用返回值为指针的函数	281
484	应用 REGS 联合	261	520	创建返回值为指针的函数	281
485	位字段结构	262	521	指针数组	281
486	形象表示位字段结构	263	522	形象表示字符串数组	282
487	位字段结构的取值范围	263	523	遍历字符串数组	282
488	在数组中查找指定的值	264	524	把字符串数组当成指针	283
489	对分查找	264	525	使用指向一个指向字符串的指针的指针	283
490	应用对分查找法	265	526	利用指针申明字符串常量	284
491	对数组进行排序	265	527	VOID 类型指针	284
492	冒泡排序法	265	528	创建指向函数的指针	284
493	应用冒泡排序法	266	529	使用指向函数的指针	284
494	选择排序法	267	530	使用三级指针	285
495	应用选择排序法	267	531	结构	286
496	SHELL (希尔) 排序法	268	532	结构是变量申明的模板	286
497	应用 SHELL 排序法	269	533	结构标记是结构的名称	286
498	快速排序法	269	534	用不同的方式申明结构	287
499	应用快速排序法	270	535	结构成员	287
500	上述排序方法的遗留问题	271	536	形象表示结构	288
501	对字符串数组排序	271	537	应用结构	288
502	利用 LFINF 搜索字符串	272	538	把结构传递给函数	289
503	利用 LSEARCH 搜索数值	273	539	在函数内部改变结构	289
504	利用 BSEARCH 搜索已排序数组	273	540	(*POINT).MEMBER 间接引用 ...	290
			541	使用 POINTER→MEMBER 格式	

.....	290	577 从段和偏移量中产生地址.....	315
542 使用无标记结构	291	578 扩充内存	315
543 结构定义的范围	291	579 使用扩充内存	316
544 初始化结构	292	580 扩展内存	317
545 进行结构 I/O.....	292	581 实模式和保护模式	318
546 使用嵌套结构	292	582 访问扩展内存	318
547 包含数组的结构	293	583 高端内存区	318
548 创建结构数组	293	584 堆栈	319
第九章 DOS 和 BIOS 服务	295	585 各种堆栈配置	319
549 DOS 系统服务	295	586 判断程序的当前堆栈大小.....	320
550 BIOS 服务.....	295	587 使用_STKLEN 控制堆栈空间 ..	320
551 寄存器	296	588 给内存区域赋值	321
552 标志寄存器	297	589 拷贝内存区域	321
553 软件中断	298	590 拷贝内存区域直到某指定字节...	321
554 利用 BIOS 访问指针	298	591 比较两个无符号字符数组.....	322
555 CONTROL+BREAK 信息	300	592 交换两个相邻字符串字节.....	323
556 可能的 DOS 副作用	300	593 分配动态内存	323
557 暂时挂起程序	301	594 再谈类型转换	324
558 控制声音	301	595 不再需要时释放内存	324
559 获取国家专用的信息	302	596 利用 CALLOC 函数分配内存 ..	325
560 磁盘传输地址	303	597 堆	326
561 访问和控制磁盘传输区	303	598 解决 64Kb 堆限制	327
562 BIOS 键盘服务.....	304	599 从堆栈中分配内存	327
563 获取 BIOS 设备列表	306	600 分配巨型数据	328
564 控制串行口 I/O.....	307	601 改变被分配内存区域的大小.....	329
565 利用 BDOS 访问 DOS 服务.....	309	602 BRK 函数	330
566 获取扩展 DOS 错误信息	309	603 检测堆	330
567 判断 BIOS 常规内存数量	310	604 快速堆检测	330
568 创建远指针 (FAR PRINTER) ..	311	605 填充自由堆空间	331
569 把远端地址分解为段地址和偏移 地址	311	606 检测特定的堆入口	332
570 判断自由核心内存	312	607 遍历堆入口	333
571 读段寄存器设置	312	608 访问指定内存单元	334
572 内存的类型	313	609 向内存中置数	334
573 常规内存	313	610 PC 端口	335
574 常规内存的分布	313	第十章 内存管理	336
575 访问常规内存	314	611 访问端口值	336
576 为什么 PC 和 DOS 限制于 1MB	315	612 CMOS	336
		613 内存模式	337

614	微型内存模式	338	647	PC 时钟类型	356
615	小型内存模式	338	第十二章 重定向 I/O 和进程命令行		357
616	中型内存模式	338	648	等候按键	357
617	压缩内存模式	339	649	提醒用户输入密码	357
618	大型内存模式	340	650	自己编写密码函数	358
619	巨型内存模式	341	651	输出重定向	359
620	判断当前的内存模式	341	652	输入重定向	359
第十一章 日期和时间		342	653	联合使用 INPUT 和 OUTPUT 重定向	359
621	获取当前日期与时间	342	654	利用 STDOUT 和 STDIN	359
622	将日期和时间从秒的形式转换成 ASCII 码	342	655	管道(pipe)运算符	360
623	DAYLIGHT SAVINGS ADJUST- MENT	343	656	GETCHAR 和 PUTCHAR	360
624	延迟若干毫秒	343	657	对重定向输入进行编号	361
625	判断程序的耗时	343	658	确保信息出现在屏幕上	361
626	比较两个时间	344	659	自定义 MORE 命令	362
627	获取数据串	344	660	显示重定向行的数目	363
628	获取时间串	345	661	显示重定向字符的个数	363
629	读 BIOS 计时器	345	662	创建定时的 MORE 命令	363
630	使用当地时间	346	663	防止 I/O 重定向	364
631	使用格林威治平时	347	664	应用 STDPRN 文件句柄	365
632	获取 DOS 系统时间	347	665	把重定向输出分割到一个文件中	365
633	获取系统日期	348	666	应用 STDAUX 文件句柄	366
634	设置 DOS 系统时间	349	667	在重定向输入内寻找子串的出现	366
635	设置 DOS 系统日期	349	668	显示重定向输入的头 N 行	367
636	把 DOS 日期转换为 UNIX 格式	350	669	命令行变元	368
637	利用 TIMEZONE 计算时差	350	670	显示命令行变元的个数	368
638	判断当前时区	351	671	显示命令行	369
639	利用 TZSET 设置时区区域	351	672	使用引号内的命令行变元	369
640	利用 TZ 环境入口	351	673	从命令行中显示文件内容	370
641	从用户程序中设置 TZ 环境入口	352	674	把 ARGV 当作指针	370
642	获取时区信息	352	675	C 是如何知道命令行的	371
643	以秒钟的形式设置自 1/1/1970 午夜以来的系统时间	353	676	环境	371
644	把日期转换成自 1/1/1970 以来 的秒数	353	677	把 ENV 当作一个指针	372
645	判断日期的儒略历日期	354	678	对 MAIN 的参数使用 VOID	372
646	创建格式化日期和时间串	354	679	使用命令行数字	372
			680	出口状态值	373

681	为出口状态过程使用 RETURN	374	714	使用 MAKE 的宏修饰符	389
682	判断是否把 MAIN 申明为 VOID		715	因错误结束 MAKE 文件	389
		374	716	关闭命令显示	389
683	在环境中搜索特定入口	375	717	使用文件 BUILTINS.MAK	390
684	DOS 是如何对待环境的	375	718	在 MAKE 中执行出口状态进程	390
685	应用 ENVIRON 全局变量	376	719	同时激活和改变一个宏	390
686	给当前环境添加入口	376	720	为多个依赖文件执行一个 MAKE	
687	给 DOS 环境添加元素	377		命令	390
688	退出当前程序	378			
689	定义在程序结束时执行的函数	378			
第十三章 编程工具		380	第十四章 高级 C 语言编程		392
690	库	380	721	判断是否有数学协处理器	392
691	重复使用目标代码	380	722	理解 CTYPE.H, ICTYPE 宏	392
692	编译 C 和 OBJ 文件时出现的问		723	控制直接的视像	393
	题	381	724	检查系统和路径错误	393
693	创建库文件	381	725	显示预定义的错误信息	394
694	常用的库操作	381	726	决定操作系统版本号	394
695	列出库文件中的子例行程序	382	727	理解可移植性	395
696	利用库减少编译时间	382	728	执行一个非本地的 GOTO	395
697	库管理程序的其他功能	382	729	获得进程 ID (PID)	396
698	连接器	382	730	激活一个内部的 DOS 命令	396
699	连接器的功能	383	731	使用 _PSP 全局变量	397
700	使用连接映像	383	732	在变量申明中使用 CONST 修饰	
701	使用连接器响应文件	383		符	398
702	使用 MAKE 命令简化应用程序		733	使用枚举类型	398
	的创建	384	734	放置一个枚举类型来使用	398
703	生成一个简单的 MAKE 文件	384	735	理解一个枚举值	399
704	通过 MAKE 使用多依赖性文件	385	736	分配一个特殊的值给枚举类型	399
705	说明用户的 MAKE 文件	385	737	保存和恢复寄存器	399
706	MAKE 和命令行	385	738	动态列表简介	400
707	在 MAKE 文件中放置多个依赖性		739	申明一个链接的列表结构	400
		386	740	建立一个链接的列表	400
708	显现的和隐含的 MAKE 法则	386	741	一个简单的链表例子	401
709	使用 MAKE 宏	387	742	理解链表转移	401
710	预定义 MAKE 宏	387	743	创建一个更有用的列表	402
711	用 MAKE 执行条件进程	388	744	增加一个列表入口	402
712	验证一个 MAKE 宏	388	745	插入一个列表入口	403
713	再次包含一个 MAKE 文件	389	746	显示一个存储的目录	403
			747	从一个列表中删除一个元素	404
			748	使用一个双向链表	405

749	创建一个简单的双向链表	406	783	理解冗余载入禁止	426
750	理解 NODE->PREVIOUS->NEXT	406	784	理解代码紧缩	426
751	从一个双向链表中移走一个元素	407	785	理解循环紧缩	427
752	在一个双向链表中插入一个元素	408	786	理解循环引入和强度削减	427
753	理解子进程	409	787	消除普通的子表达式	428
754	派生一个子进程	409	788	标准 C 语言转换	428
755	使用其他的 SPAWNLXX 函数	410	789	理解 C 语言的 4 个基本类型	428
756	使用 SPAWNVXX 函数	411	790	基本类型与派生类型	428
757	执行一个子进程	412	791	理解初始化值	429
758	使用其他的 EXECLXX 函数	413	792	理解连接	429
759	使用 EXECVXX 函数	414	793	理解临时申明	430
760	理解覆盖	414	794	申明和定义	430
761	理解中断	415	795	理解左值 (LVALUE)	430
762	PC 机的中断	415	796	理解右值 (RVALUE)	431
763	使用中断关键字	416	797	使用段寄存器关键字	431
764	判断一个中断向量	416	798	谨慎使用远指针	431
765	设置一个中断向量	417	799	理解正常化的指针	432
766	使能与禁止中断	417	800	数学协处理器语句	432
767	生成简单的中断处理器	417	801	理解变量中的 CDECL 和 PASCAL	433
768	链接一个二次中断	418	802	防止循环包含	433
769	生成一个中断	419	第十五章 C++ 入门	434	
770	捕获 PC 机时钟	419	803	C++ 介绍	434
771	理解致命错误	420	804	C++ 源文件的差异	434
772	C 语言中的致命错误处理	421	805	从简单的 C++ 程序开始	434
773	一个更完全的致命错误处理器	421	806	理解 COUT I/O 流	434
774	恢复改变过的中断	423	807	使用 COUT 输出值和变量	435
775	生成一个 Ctrl+Break 处理器	423	808	用 COUT 连接不同的数据类型	435
776	在用户的致命错误处理器中使 用 DOS 服务	424	809	显示十六进制和八进制数值	436
777	使用指令集选择改善性能	424	810	重定向 COUT	436
778	直接插入内部函数	424	811	如果钟情 PRINTF, 使用 PRINTF	436
779	使能和禁止内在函数	425	812	输出到 CERR	436
780	理解快速函数调用	425	813	用 CIN 得到输入	437
781	_FASTCALL 参数传递的法则	426	814	CIN 不要使用指针	437
782	理解不变代码	426	815	理解 CIN 如何选择数据域	437
			816	理解输入输出流如何获得变量类 型	438