



万水计算机技术实用大全系列



C/C++/C# 程序员实用大全 —— C/C++/C# 最佳编程指南

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

[美] Kris Jamsa 著

张春晖 刘大庆 李 越 等译

利水文化 审校



ONWORD PRESS
THOMSON LEARNING



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++/C#程序员实用大全

——C/C++/C#最佳编程指南

[美] Kris Jamsa 著

张春晖 刘大庆 李 越 等译

利水文化 审校

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++/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;最后介绍了基于.NET 环境的 C#编程。无论是 C/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++/C#编程的优秀指南。

First published by Delmar, a division of Thomson Learning.

All Rights Reserved. Authorized Simplified Chinese Edition by Thomson Learning and China WaterPower Press. No part of this book may be reproduced in any form without the express written permission of Thomson Learning and China WaterPower Press.

北京市版权局著作权合同登记号:图字 01-2002-1920 号

图书在版编目(CIP)数据

C/C++/C#程序员实用大全 / (美)耶姆塞(Jamsa,K.)著;张春晖译. — 北京:中国水利水电出版社,2002

(万水计算机技术实用大全系列)

书名原文:Jamsa's C/C++/C# Programmer's Bible

ISBN 7-5084-1189-7

I. C… II. ①耶…②张… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 063921 号

书 名	C/C++/C#程序员实用大全——C/C++/C#最佳编程指南
作 者	[美] Kris Jamsa 著
译 者	张春晖 刘大庆 李越 等译
审 校	利水文化
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: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话:(010) 68359286 (万水) 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	北京天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 73.5 印张 1699 千字
版 次	2002 年 10 月第一版 2002 年 10 月北京第一次印刷
印 数	0001—3000 册
定 价	125.00 元(含 1CD)

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

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	94	移位运算	45
60	用指数格式显示浮点数	27	95	位循环运算	46
61	显示浮点数	28	96	条件运算符	46
62	用 PRINTF 显示字符串	28	97	循环控制	47
63	用 PRINTF 显示指针地址	29	98	C 如何表示真 (TRUE) 和假 (FALSE)	47
64	在数值前添加正号和负号	29	99	用 if 判断条件	47
65	用 PRINTF 格式化整数值	29	100	简单语句和复杂语句	47
66	0 填充整数输出	30	101	判断是否相等	48
67	在八进制和十六进制数前显示 前缀	31	102	关系判断	49
68	用 PRINTF 格式化浮点数	31	103	用逻辑与判断两个条件	49
69	格式化指数输出	31	104	用逻辑或来判断两个条件	50
70	左对齐 PRINTF 的输出	32	105	逻辑非运算	50
71	联合使用格式符	32	106	将条件结果赋值给变量	51
72	字符串的换行	33	107	在复合语句中定义变量	51
73	显示 NEAR 和 FAR 字符	33	108	使用缩进来提高程序的可读性	52
74	使用 PRINTF 的转义字符	33	109	使用扩展 CTRL+BREAK 检查	53
75	判断 PRINTF 已显示的字符数目 ..	34	110	判断浮点数	53
76	使用 PRINTF 的返回值	34	111	永远循环下去	53
77	使用 ANSI 设备驱动器	35	112	赋值判断	54
78	用 ANSI 驱动器清除屏幕显示	35	113	IF-IF-ELSE 语句	54
79	用 ANSI 驱动器显示屏幕颜色	35	114	按规定次数执行语句	55
80	用 ANSI 驱动器定位光标	36	115	FOR 语句的有些部分是可选 择的	56
81	在 C 中作基本的数学运算	36	116	在 FOR 语句中减值	57
82	模运算 (取余运算)	37	117	控制 FOR 循环的增值	57
83	运算符的优先级和结合性	38	118	在 FOR 循环中使用字符型和浮 点型数值	58
84	强制操作符运算顺序	38	119	空循环	58
85	C 的自增运算符	39	120	无穷循环	59
86	C 的自减运算符	40	121	在 FOR 循环中使用逗号运算符 ..	59
87	按位或运算	41	122	不要在 FOR 循环中改变控制变 量的值	60
88	按位与运算	41	123	用 WHILE 循环重复执行一条或 多条语句	60
89	按位异或运算	42	124	WHILE 循环的组成部分	61
90	“取反”运算	43			
91	对变量的值进行运算	43			
92	C 的条件运算符	44			
93	C 的长度 (SIZEOF) 运算符	44			

125	使用 DO 重复执行一条或多条语句	62	128	GOTO 语句分支	65
126	C 的 CONTINUE 语句	63	129	判断多个条件	65
127	使用 C 的 BREAK 语句来结束循环	64	130	在 SWITCH 中使用 BREAK	66
第二章	宏与常量	69	131	使用 SWITCH 语句的 DEFAULT CASE	67
132	在程序中定义常量	69	160	宏是没有类型的	80
133	宏与常量扩展	69	第三章	字符串	81
134	给常量和宏命名	70	161	C 字符串的形象化	81
135	使用 _FILE_ 预处理器常量	70	162	编译器是如何表示字符串的	81
136	使用 _LINE_ 预处理器常量	70	163	C 是如何存储字符串的	81
137	改变预处理器的行计数	71	164	'A' 是如何区别于 "A" 的	82
138	生成无条件预处理器错误	72	165	在字符串常量内表示引号	82
139	其他预处理器常量	72	166	判断字符串的长度	83
140	记录预处理器的日期和时间	72	167	使用 STRLEN 函数	83
141	判断是否进行 ANSI C 编译	72	168	将一个字符串的字符复制到另一个字符串中	84
142	判断是 C++ 还是 C	73	169	将一个串的内容追加到另一个串上	85
143	取消宏或常量	73	170	给字符串追加 N 个字符	85
144	比较宏与函数	73	171	把一个字符串转移到另一个字符串	86
145	编译器 PRAGMAS	74	172	不要越过字符串的界限	86
146	预定义值和宏	74	173	判断两个字符串是否相同	87
147	创建用户自己的头文件	74	174	比较字符串时忽略大小写	87
148	使用 #INCLUDE <FILENAME.H> 或 #INCLUDE "FILENAME.H"	74	175	将字符串转换成大写或小写	88
149	判断符号是否被定义	75	176	获取字符串中第一次出现的某个字符	89
150	进行 IF-ELSE 预处理	75	177	返回索引到串的首次出现	89
151	更强大的预处理器条件判断	76	178	搜索字符在字符串中的末次出现	90
152	实现 IF-ELSE 和 ELSE-IF 预处理	76	179	返回指向字符串末次出现的索引	90
153	定义需要多行的宏和常量	77	180	使用 FAR 字符串	91
154	创建自定义宏	77	181	为 FAR 字符串编写字符串函数	91
155	在宏定义中不要放置分号	78	182	计算字符串中字符出现的次数	91
156	创建 MIN 和 MAX 宏	78	183	将字符串的内容反转	92
157	创建 SQUARE 和 CUBE 宏	78	184	将某特定字符赋给整个字符串	92
158	注意宏定义中的空格	79	185	比较两个字符串	93
159	如何使用括号	79			

186	比较两个字符串的前 N 个字符 ...	93	217	函数	107
187	不考虑大小写比较字符串	94	218	在函数中使用变量	108
188	将字符串转换成数字	95	219	把 MAIN 当作函数	108
189	复制字符串的内容	95	220	参数简介	109
190	从给定字符序列中查找字符的 首次出现	96	221	使用多参数	109
191	在字符串中查找子字符串	97	222	老式 C 程序中的参数申明	110
192	计算子字符串出现的次数	97	223	函数返回值	110
193	给子字符串获取索引	98	224	RETURN 语句	111
194	获取子字符串的最右端出现	98	225	函数原型	112
195	不使用 %S 格式标识符显示字 符串	98	226	运行时程序库	113
196	从字符串中删除子字符串	99	227	形参和实参	113
197	用另一个子字符串代替子字 符串	99	228	解决名称冲突	114
198	转换数值的 ASCII 码形式	100	229	返回类型为非 INT 型的函数	114
199	判断字符是否为字母数字	100	230	局部变量	114
200	字符是否为字母	101	231	函数如何使用堆栈	115
201	判断字符是否包含 ASCII 值	101	232	函数的开销	115
202	判断字符是否为控制符	101	233	C 如何存储局部变量	116
203	判断字符是否为数字	101	234	申明全局变量	116
204	判断字符是否为图形字符	102	235	避免使用全局变量	117
205	判断字符是大写还是小写	102	236	解决全局和局部变量的名称 冲突	117
206	判断字符是否可打印	102	237	更好地定义全局变量的有效 范围	118
207	判断字符是否为标点符号	102	238	传值调用	118
208	判断字符是否包含空白 (whitespace) 符	102	239	使用传值调用防止参数值变化 ..	119
209	判断字符是否为十六进制值	103	240	传址调用	120
210	将字符转换成大写形式	103	241	获取地址	120
211	将字符转换成小写形式	104	242	使用变量的地址	121
212	使用 ASCII 字符	104	243	改变参数的值	122
213	将输出格式写进字符串变量	104	244	只改变指定参数	122
214	从字符串中读输入	105	245	使用堆栈进行传址调用	123
215	标志字符串以节省空间	105	246	记住函数变量的值	123
216	初始化字符串	106	247	C 是如何初始化静态变量的	124
第四章	函数	107	248	使用 PASCAL 调用顺序	125
			249	PASCAL 关键字的影响	125
			250	混合编程示例	126

251	CDECL 关键字.....	126	VA_END 是如何工作的.....	142
252	递归函数.....	126	285 创建支持多参数多类型的函数..	143
253	递归阶乘函数.....	127	第五章 键盘操作.....	145
254	另一个递归的例子.....	128	286 从键盘读入字符.....	145
255	进一步理解递归.....	128	287 显示字符输出.....	145
256	直接递归与间接递归.....	129	288 缓冲输入.....	146
257	判断是否要使用递归.....	129	289 将键盘输入赋给字符串.....	146
258	为什么递归函数慢.....	130	290 联合使用 GETCHAR 和	
259	如何消除递归.....	131	PUTCHAR.....	146
260	将字符串传递给函数.....	131	291 记住, GETCHAR 和	
261	传递指定的数组元素.....	131	PUTCHAR 都是宏.....	147
262	形式参数中的 CONST.....	132	292 使用直接 I/O 读入字符.....	147
263	使用 CONST 不会阻止参数值		293 不显示字符的直接键盘输入.....	148
	的修改.....	132	294 知道何时使用 ‘\R’和 ‘\N’.....	148
264	无界字符串的申明.....	133	295 直接输出.....	149
265	指针的使用与字符串的申明.....	133	296 将按键放回键盘缓存.....	149
266	C 是如何使用堆栈处理字符		297 使用 CPRINTF 快速格式化	
	串参数的.....	134	输出.....	150
267	外部变量.....	134	298 快速格式化键盘输入.....	151
268	应用外部变量.....	134	299 写字符串.....	151
269	外部静态变量.....	135	300 使用直接 I/O 实现更快的字符	
270	VOLATILE 关键字.....	136	串输出.....	152
271	调用结构和基指针.....	136	301 从键盘读入字符串.....	152
272	调用汇编语言函数.....	137	302 以更快的速度从键盘输入字	
273	从汇编语言函数中返回值.....	138	符串.....	153
274	没有返回值的函数.....	139	303 以彩色显示输出.....	153
275	不使用参数的函数.....	139	304 清除屏幕显示.....	154
276	AUTO 关键字.....	139	305 删除当前行到行尾的内容.....	154
277	范围.....	139	306 删除屏幕上的当前行.....	154
278	范围的分类.....	140	307 定位光标进行屏幕输出.....	155
279	名称空间和标识符.....	140	308 判断行与列的位置.....	155
280	标识符的可见性.....	140	309 在屏幕上插入空行.....	155
281	DURATION.....	141	310 将屏幕上的文本拷贝到缓冲区..	156
282	支持参数个数可变的函数.....	141	311 将缓冲区中的文本拷贝到屏幕	
283	支持个数可变的参数.....	141	的指定位置.....	157
284	VA_START、VA_ARG 和		312 判断文本模式设置.....	157

313	控制屏幕颜色	158	345	生成随机数	174
314	指定背景色	159	346	将随机值映射到指定范围	175
315	使用 TEXTCOLOR 设置前景色	160	347	给随机数生成器赋初值	175
316	使用 TEXTBACKGROUND 设置背景色	160	348	计算数值的平方根	176
317	控制文本的明暗度	161	349	创建定制的数学错误处理程序 ..	176
318	决定当前文本模式	161	第七章 文件、目录和磁盘	178	
319	在屏幕上移动文本	162	350	判断当前盘驱动器	178
320	定义文本窗口	162	351	选择当前驱动器	178
第六章 数学	164		352	判断可用的盘空间	178
321	使用整型表达式的绝对值	164	353	当心 DBLSPACE	179
322	使用 ARCCOSINE (反余弦) ..	164	354	读入文件分配表 (FAT) 信息 ..	179
323	使用 ARCSINE (反正弦)	165	355	磁盘 ID	180
324	使用 ARCTANGENT (反正切)	165	356	绝对扇区读写操作	181
325	求复数的绝对值	165	357	进行 BIOS 磁盘 I/O	181
326	对浮点值进位舍入	166	358	测试软驱是否准备好	183
327	使用角的余弦	166	359	应用 FOPEN 打开文件	183
328	使用角的双曲余弦	167	360	FILE 结构	184
329	使用角的正弦	167	361	关闭一个打开的文件	185
330	使用角的双曲正弦	167	362	每次读 / 写文件信息的一个字符	185
331	使用角的正切	168	363	文件指针的位置指针	186
332	使用角的双曲正切	168	364	判断当前文件位置	186
333	整数相除	169	365	文件流	187
334	使用指数	169	366	文件翻译	187
335	使用浮点型表达式的绝对值	170	367	CONFIG.SYS 文件的 FILES= 条目	187
336	使用浮点余数	170	368	使用低级和高级文件 I/O	188
337	使用浮点值的尾数和指数	170	369	文件句柄 (FILE HANDLES) ..	188
338	计算 X*2E 的结果	171	370	进程文件表 (PROCESS FILE TABLE)	188
339	计算自然对数	172	371	进程文件表入口	189
340	计算 log ₁₀ x 的值	172	372	系统文件表	190
341	判断最大值与最小值	173	373	显示系统文件表	191
342	把浮点值分解成整数和小数部分	173	374	从流指针中导出文件句柄	192
343	计算 x ⁿ 的结果	173	375	进行格式化文件输出	193
344	计算 10 ^x 的结果	174	376	重命名文件	193

377	删除文件	194	411	获取文件的日期和时间标记	215
378	判断程序如何访问文件	195	412	利用位域获取文件的日期与 时间	217
379	设置文件的访问模式	196	413	设置文件的日期与时间标记	218
380	深入掌握文件属性	196	414	把文件日期和时间设置成当前 日期和时间	219
381	检测文件流错误	198	415	每次读写一个字	219
382	判断文件的长度	199	416	改变文件的长度	220
383	刷新 I/O 流	199	417	控制文件打开操作的读写模式 ..	221
384	一次关闭所有被打开的文件	200	418	将缓冲区赋给文件	222
385	获取文件流的文件句柄	200	419	分配文件缓冲区	223
386	使用 P_TMPDIR 创建临时文 件名	201	420	利用 MKTEMP 创建唯一文 件名	224
387	使用 TMP 或 TEMP 创建临时 文件名	201	421	读写结构	224
388	创建真正的临时文件	202	422	从文件流中读取结构数据	225
389	删除临时文件	202	423	复制文件句柄	226
390	为文件搜索命令路径	202	424	强制文件句柄设置	227
391	为文件搜索环境入口的子目录 ..	203	425	把文件句柄和文件流联系起来 ..	228
392	打开 TEMP 目录中的文件	204	426	文件共享	228
393	最小化文件 I/O 操作	204	427	打开文件进行共享访问	228
394	在目录名中使用反斜杠	205	428	锁定文件内容	229
395	改变当前目录	205	429	获取更精细的文件锁定控制	231
396	创建目录	206	430	使用 DOS 目录	232
397	删除目录	206	431	打开目录	232
398	删除目录树	206	432	读取目录入口	233
399	建立完全路径名	208	433	利用目录服务读 C:\WINDOWS	233
400	分解目录路径	209	434	反绕目录	233
401	建立路径名	210	435	递归读取磁盘文件	233
402	使用低级函数打开和关闭文件 ..	210	436	判断当前文件位置	234
403	创建文件	211	437	打开共享文件流	235
404	进行低级读写操作	212	438	在指定目录中创建唯一文件	236
405	判断文件是否结束	212	439	创建新文件	236
406	应用低级文件例行程序	213	440	利用 DOS 服务访问文件	237
407	为文件句柄翻译指定模式	213	441	强制二进制或文本文件打开	239
408	利用 LSEEK 定位文件指针	214	442	按行读文本	239
409	打开多于 20 个的文件	214	443	按行写文本	239
410	使用 DOS 文件服务	215			

444	应用 FGETS 和 FPUTS.....	240	478	C 是如何存放多维数组的	257
445	强制二进制文件翻译	240	479	按行存放与按列存放	258
446	为什么 TEXTCOPY 不能拷贝 二进制文件	240	480	以数组为成员的结构数组	258
447	判断文件结尾	241	481	联合	259
448	舍弃字符	241	482	使用联合节省内存	260
449	读取格式化的文件数据	241	483	使用 REGS—— 一种典型的 联合	261
450	根据当前位置定位文件指针	242	484	应用 REGS 联合	261
451	获取文件句柄信息	243	485	位字段结构	262
452	重新打开文件流	243	486	形象表示位字段结构	263
第八章	数组、指针和结构	245	487	位字段结构的取值范围	263
453	数组	245	488	在数组中查找指定的值	264
454	申明数组	245	489	对分查找	264
455	形象表示数组	245	490	应用对分查找法	265
456	数组的内存需求	246	491	对数组进行排序	265
457	初始化数组	247	492	冒泡排序法	265
458	访问数组元素	247	493	应用冒泡排序法	266
459	通过循环访问数组元素	248	494	选择排序法	267
460	使用常量定义数组	248	495	应用选择排序法	267
461	把一个数组传送给函数	249	496	SHELL (希尔) 排序法	268
462	把数组看作函数	249	497	应用 SHELL 排序法	269
463	区分字符串数组	250	498	快速排序法	269
464	在堆栈中传送数组	250	499	应用快速排序法	270
465	判断数组能存放多少个元素	250	500	上述排序方法的遗留问题	271
466	为大数组使用 HUGE 内存模式	251	501	对字符串数组排序	271
467	权衡数组与动态存储的利弊	251	502	利用 LFINd 搜索字符串	272
468	多维数组	252	503	利用 LSEARCH 搜索数值	273
469	行与列	252	504	利用 BSEARCH 搜索已排序 数组	273
470	访问二维数组中的元素	253	505	利用 QSORT 对数组排序	274
471	给二维数组元素赋初值	253	506	判断数组元素的个数	275
472	判断多维数组占用的内存	254	507	把指针理解为地址	276
473	通过循环显示二维数组	254	508	判断变量的地址	276
474	遍历三维数组	255	509	C 是如何把数组当成指针的	276
475	初始化多维数组	255	510	对数组应用取地址运算符(&)	277
476	把二维数组传送给函数	256	511	申明指针变量	277
477	把多维数组当作一维数组	257			

512	间接访问指针	278	546	使用嵌套结构	292
513	使用指针值	278	547	包含数组的结构	293
514	指针与函数参数的使用	279	548	创建结构数组	293
515	指针运算	279	第九章	DOS 和 BIOS 服务	295
516	指针的增值与减值	279	549	DOS 系统服务	295
517	联合应用指针引用与增值	280	550	BIOS 服务	295
518	利用指针遍历数组	280	551	寄存器	296
519	使用返回值为指针的函数	281	552	标志寄存器	297
520	创建返回值为指针的函数	281	553	软件中断	298
521	指针数组	281	554	利用 BIOS 访问指针	298
522	形象表示字符串数组	282	555	CONTROL+BREAK 信息	300
523	遍历字符串数组	282	556	可能的 DOS 副作用	300
524	把字符串数组当成指针	283	557	暂时挂起程序	301
525	使用指向一个指向字符串的指 针的指针	283	558	控制声音	301
526	利用指针申明字符串常量	284	559	获取国家专用的信息	302
527	VOID 类型指针	284	560	磁盘传输地址	303
528	创建指向函数的指针	284	561	访问和控制磁盘传输区	303
529	使用指向函数的指针	284	562	BIOS 键盘服务	304
530	使用三级指针	285	563	获取 BIOS 设备列表	306
531	结构	286	564	控制串行口 I/O	307
532	结构是变量申明的模板	286	565	利用 BDOS 访问 DOS 服务	309
533	结构标记是结构的名称	286	566	获取扩展 DOS 错误信息	309
534	用不同的方式申明结构	287	567	判断 BIOS 常规内存数量	310
535	结构成员	287	568	创建远指针 (FAR PRINTER)	311
536	形象表示结构	288	569	把远端地址分解为段地址和偏 移地址	311
537	应用结构	288	570	判断自由核心内存	312
538	把结构传递给函数	289	571	读段寄存器设置	312
539	在函数内部改变结构	289	第十章	内存管理	314
540	(*POINT).MEMBER 间接引用 ..	290	572	内存的类型	314
541	使用 POINTER→MEMBER 格式	290	573	常规内存	314
542	使用无标记结构	291	574	常规内存的分布	314
543	结构定义的范围	291	575	访问常规内存	315
544	初始化结构	292	576	为什么 PC 和 DOS 限制于 1MB ..	315
545	进行结构 I/O	292	577	从段和偏移量中产生地址	316
			578	扩充内存	316

579	使用扩充内存	317	615	小型内存模式	339
580	扩展内存	318	616	中型内存模式	339
581	实模式和保护模式	318	617	压缩内存模式	340
582	访问扩展内存	319	618	大型内存模式	340
583	高端内存区	319	619	巨型内存模式	341
584	堆栈	320	620	判断当前的内存模式	342
585	各种堆栈配置	320	第十一章 日期和时间	343	
586	判断程序的当前堆栈大小	321	621	获取当前日期与时间	343
587	使用_STKLEN 控制堆栈空间	321	622	将日期和时间从秒的形式转 换成 ASCII 码	343
588	给内存区域赋值	321	623	DAYLIGHT SAVINGS ADJUSTMENT	344
589	拷贝内存区域	322	624	延迟若干毫秒	344
590	拷贝内存区域直到某指定字节	322	625	判断程序的耗时	344
591	比较两个无符号字符数组	323	626	比较两个时间	345
592	交换两个相邻字符串字节	323	627	获取数据串	345
593	分配动态内存	324	628	获取时间串	346
594	再谈类型转换	325	629	读 BIOS 计时器	346
595	不再需要时释放内存	325	630	使用当地时间	347
596	利用 CALLOC 函数分配内存	326	631	使用格林威治平时	348
597	堆	326	632	获取 DOS 系统时间	348
598	解决 64Kb 堆限制	327	633	获取系统日期	349
599	从堆栈中分配内存	328	634	设置 DOS 系统时间	350
600	分配巨型数据	329	635	设置 DOS 系统日期	350
601	改变被分配内存区域的大小	330	636	把 DOS 日期转换为 UNIX 格式	351
602	BRK 函数	330	637	利用 TIMEZONE 计算时差	351
603	检测堆	331	638	判断当前时区	352
604	快速堆检测	331	639	利用 TZSET 设置时区区域	352
605	填充自由堆空间	332	640	利用 TZ 环境入口	352
606	检测特定的堆入口	333	641	从用户程序中设置 TZ 环境 入口	353
607	遍历堆入口	334	642	获取时区信息	353
608	访问指定内存单元	334	643	以秒钟的形式设置自 1/1/1970 午夜以来的系统时间	354
609	向内存中置数	335	644	把日期转换成自 1/1/1970 以来	
610	PC 端口	336			
611	访问端口值	336			
612	CMOS	337			
613	内存模式	338			
614	微型内存模式	338			

秒数.....	354	676 环境.....	372
645 判断日期的儒略历日期.....	355	677 把 ENV 当作一个指针.....	373
646 创建格式化日期和时间串.....	355	678 对 MAIN 的参数使用 VOID.....	373
647 PC 时钟类型.....	357	679 使用命令行数字.....	373
第十二章 重定向 I/O 和进程命令行 ...	358	680 出口状态值.....	374
648 等候按键.....	358	681 为出口状态过程使用 RETURN.....	375
649 提醒用户输入密码.....	358	682 判断是否把 MAIN 申明为 VOID.....	375
650 自己编写密码函数.....	359	683 在环境中搜索特定入口.....	376
651 输出重定向.....	360	684 DOS 是如何对待环境的.....	376
652 输入重定向.....	360	685 应用 ENVIRON 全局变量.....	377
653 联合使用 INPUT 和 OUTPUT 重定向.....	360	686 给当前环境添加入口.....	377
654 利用 STDOUT 和 STDIN.....	360	687 给 DOS 环境添加元素.....	378
655 管道(pipe)运算符.....	361	688 退出当前程序.....	379
656 GETCHAR 和 PUTCHAR.....	361	689 定义在程序结束时执行的函数.....	379
657 对重定向输入进行编号.....	362	第十三章 编程工具	381
658 确保信息出现在屏幕上.....	362	690 库.....	381
659 自定义 MORE 命令.....	363	691 重复使用目标代码.....	381
660 显示重定向行的数目.....	364	692 编译 C 和 OBJ 文件时出现的 问题.....	382
661 显示重定向字符的个数.....	364	693 创建库文件.....	382
662 创建定时的 MORE 命令.....	364	694 常用的库操作.....	382
663 防止 I/O 重定向.....	365	695 列出库文件中的子例行程序.....	383
664 应用 STDPRN 文件句柄.....	366	696 利用库减少编译时间.....	383
665 把重定向输出分割到一个文 件中.....	366	697 库管理程序的其他功能.....	383
666 应用 STDAUX 文件句柄.....	367	698 连接器.....	383
667 在重定向输入内寻找子串的 出现.....	367	699 连接器的功能.....	384
668 显示重定向输入的头 N 行.....	368	700 使用连接映像.....	384
669 命令行变元.....	369	701 使用连接器响应文件.....	384
670 显示命令行变元的个数.....	369	702 使用 MAKE 命令简化应用程序 的创建.....	385
671 显示命令行.....	370	703 生成一个简单的 MAKE 文件.....	385
672 使用引号内的命令行变元.....	370	704 通过 MAKE 使用多依赖性文件.....	386
673 从命令行中显示文件内容.....	371	705 说明用户的 MAKE 文件.....	386
674 把 ARGV 当作指针.....	371	706 MAKE 和命令行.....	386
675 C 是如何知道命令行的.....	372	707 在 MAKE 文件中放置多个依	

赖性	387	740 建立一个链接的列表	401
708 显现的和隐含的 MAKE 法则 ...	387	741 一个简单的链表例子	402
709 使用 MAKE 宏	388	742 理解链表转移	402
710 预定义 MAKE 宏	388	743 创建一个更有用的列表	403
711 用 MAKE 执行条件进程	389	744 增加一个列表入口	403
712 验证一个 MAKE 宏	389	745 插入一个列表入口	404
713 再次包含一个 MAKE 文件	390	746 显示一个存储的目录	404
714 使用 MAKE 的宏修饰符	390	747 从一个列表中删除一个元素	405
715 因错误结束 MAKE 文件	390	748 使用一个双向链表	406
716 关闭命令显示	390	749 创建一个简单的双向链表	407
717 使用文件 BUILTINS.MAK	391	750 理解 NODE->PREVIOUS-> NEXT	407
718 在 MAKE 中执行出口状态进程	391	751 从一个双向链表中移走一个 元素	408
719 同时激活和改变一个宏	391	752 在一个双向链表中插入一个 元素	409
720 为多个依赖文件执行一个 MAKE 命令	391	753 理解子进程	410
第十四章 高级 C 语言编程.....	393	754 派生一个子进程	410
721 判断是否有数学协处理器	393	755 使用其他的 SPAWNLXX 函数...	411
722 理解 CTYPE.H, ISTYPE 宏	393	756 使用 SPAWNVXX 函数	412
723 控制直接的视像	394	757 执行一个子进程	413
724 检查系统和路径错误	394	758 使用其他的 EXECLXX 函数	414
725 显示预定义的错误信息	395	759 使用 EXECVXX 函数	415
726 决定操作系统版本号	395	760 理解覆盖	415
727 理解可移植性	396	761 理解中断	416
728 执行一个非本地的 GOTO	396	762 PC 机的中断	416
729 获得进程 ID (PID)	397	763 使用中断关键字	417
730 激活一个内部的 DOS 命令	397	764 判断一个中断向量	417
731 使用 _PSP 全局变量	398	765 设置一个中断向量	418
732 在变量申明中使用 CONST 修 饰符	399	766 使能与禁止中断	418
733 使用枚举类型	399	767 生成简单的中断处理器	418
734 放置一个枚举类型来使用	399	768 链接一个二次中断	419
735 理解一个枚举值	400	769 生成一个中断	420
736 分配一个特殊的值给枚举类型 ..	400	770 捕获 PC 机时钟	420
737 保存和恢复寄存器	400	771 理解致命错误	421
738 动态列表简介	401	772 C 语言中的致命错误处理	422
739 申明一个链接的列表结构	401		

773	一个更完全的致命错误处理器..	422	806	理解 COUT I/O 流.....	435
774	恢复改变过的中断.....	424	807	使用 COUT 输出值和变量	436
775	生成一个 Ctrl+Break 处理器	424	808	用 COUT 连接不同的数据类型 ..	436
776	在用户的致命错误处理器中 使用 DOS 服务	425	809	显示十六进制和八进制数值	437
777	使用指令集选择改善性能	425	810	重定向 COUT	437
778	直接插入内部函数	425	811	如果钟情 PRINTF, 使用 PRINTF	437
779	使能和禁止内在函数	426	812	输出到 CERR.....	437
780	理解快速函数调用	426	813	用 CIN 得到输入	438
781	_FASTCALL 参数传递的法则..	427	814	CIN 不要使用指针	438
782	理解不变代码.....	427	815	理解 CIN 如何选择数据域	438
783	理解冗余载入禁止	427	816	理解输入输出流如何获得变量 类型	439
784	理解代码紧缩	427	817	使用 CLOG 实现输出	439
785	理解循环紧缩	428	818	CIN、COUT、CERR 和 CLOG 是类的实例	439
786	理解循环引入和强度削减.....	428	819	使用 FLUSH 操纵符快速输出 ...	439
787	消除普通的子表达式	429	820	理解 IOSTREAM.H 头文件包 含的内容	439
788	标准 C 语言转换	429	821	C++需要函数原型	440
789	理解 C 语言的 4 个基本类型	429	822	C++增加的新关键字	440
790	基本类型与派生类型	429	823	C++支持匿名联合	440
791	理解初始化值	430	824	分辨全局范围	441
792	理解连接	430	825	提供缺省参数值	441
793	理解临时申明	431	826	控制 COUT 的输出宽度	441
794	申明和定义	431	827	使用 SETW 设置 COUT 宽度	442
795	理解左值 (LVALUE)	431	828	指定 COUT 的填充字符	442
796	理解右值 (RVALUE)	432	829	左对齐和右对齐 COUT 的输出 ..	443
797	使用段寄存器关键字	432	830	控制 COUT 显示浮点数字的数 目	444
798	谨慎使用远指针	432	831	以小数或科学记数格式显示数 值	444
799	理解正常化的指针	433	832	恢复 COUT 至默认值	445
800	数学协处理器语句	433	833	设置输入输出基数	445
801	理解变量中的 CDECL 和 PASCAL	434	834	在需要的地方定义变量	446
802	防止循环包含	434	835	在函数原型中放置默认参数值 ..	446
第十五章 C++入门.....		435			
803	C++介绍	435			
804	C++源文件的差异	435			
805	从简单的 C++程序开始	435			

836	使用按位运算符及 COUT	446	871	用 SET_NEW_HANDLER 函数设置 NEW 处理器	460
837	理解迟缓（或短路）计算	447	872	判断 C++ 编译	460
838	在 C++ 中使用 CONST 关键字 ..	447	873	理解 C++ 中的结构	461
839	在 C++ 中使用 ENUM 关键字	447	874	结构中定义函数成员	461
840	理解自由空间	447	875	在结构内定义成员函数	462
841	用 NEW 分配内存	448	876	在结构外定义成员函数	462
842	为多个数组分配内存	448	877	给成员函数传递参数	462
843	无自由空间的测试	448	878	同一结构的多个变量	463
844	关于堆空间	449	879	不同结构具有同名函数成员	463
845	使用 FAR 指针和 NEW 运算符 ..	449	880	同名成员不同函数	464
846	释放内存至自由空间	449	第十六章 对象		465
847	理解 C++ 中的引用	450	881	理解对象	465
848	给函数传递引用	450	882	理解面向对象编程	465
849	防止隐藏对象	451	883	理解为什么使用对象	465
850	用三种方法传递参数	451	884	把程序分解成对象	466
851	使用引用的规则	451	885	理解对象和类	467
852	函数可返回引用	452	886	理解 C++ 的类	467
853	使用 INLINE 关键字	452	887	理解封装	468
854	使用 C++ 的 ASM 关键字	453	888	理解多态性	468
855	用 CIN 读字符	454	889	理解继承	468
856	用 COUT 写字符	454	890	类和结构的选择	469
857	简单过滤器程序	454	891	创建简单类模型	469
858	简单的 TEE 命令	454	892	实现简单类的程序	470
859	简单的 FIRST 命令	455	893	定义类构件	471
860	更好的 FIRST 命令	455	894	理解作用域分辨符	471
861	文件结束测试	455	895	在申明中使用或省略类名	471
862	用 ENDL 产生新行	456	896	理解 PUBLIC: 标志	472
863	理解连接规范	456	897	理解信息隐藏	472
864	理解重载	457	898	理解 PRIVATE: 标志	472
865	重载函数	457	899	理解 PROTECTED: 标志	473
866	重载函数的第二个例子	457	900	使用公有和私有数据	473
867	避免不明确的重载	458	901	决定什么隐藏什么公开	474
868	使用 CIN 每次读一行	459	902	公有方法常称为接口函数	474
869	在循环中使用 CIN.GETLINE	459	903	在类外定义类函数	475
870	改变 NEW 运算符的缺省处 理器	459	904	在类的内部和外部定义方法	476