

C语言

高级编程 及实例剖析

- 深入浅出 C 语言编程高级技术
- 富含精选的实用开发案例
- 汇集丰富的 C 语言编程技巧
- 提供全部源代码和可执行文件

内附光盘



王为青 刘变红 编著

 人民邮电出版社
POSTS & TELECOM PRESS

C 语言高级编程及实例剖析

王为青 刘变红 编著

人 民 邮 电 出 版 社
北 京

图书在版编目 (CIP) 数据

C 语言高级编程及实例剖析 / 王为青, 刘变红编著. —北京: 人民邮电出版社, 2007.3

ISBN 978-7-115-15423-1

I. C... II. ①王...②刘... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 125592 号

内 容 简 介

本书结合实例, 深入浅出地介绍了 C 语言在主要应用领域的编程技术。全书共 9 章, 分别为内存管理、文本屏幕界面设计、文件高级操作、图形图像、中断、通信技术、基本总线接口编程、游戏开发和综合开发实例——信息管理系统。

本书采用技术讲解与实例解析并进的方式展开。每一章中, 先讲述相关技术, 然后通过实例介绍具体的应用。通过研读, 读者可以掌握此方面的技术知识, 并能较快应用到工程开发中去。

本书面向具有一定 C 语言基础的读者, 能够帮助他们快速成长为 C 语言编程高手; 本书也可作为相关领域工程人员的参考用书。

C 语言高级编程及实例剖析

-
- ◆ 编 著 王为青 刘变红
责任编辑 李 岚
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 26.75
字数: 651 千字
印数: 1—5 000 册
 - 2007 年 3 月第 1 版
2007 年 3 月北京第 1 次印刷

ISBN 978-7-115-15423-1/TP

定价: 48.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

前 言

C 语言功能强大, 具有丰富灵活的控制与数据结构、简洁而高效的语句表达、清晰的程序结构、良好的移植性、较小的空间开销等优点, 已被广泛应用于系统软件与应用软件的开发中。

本书以深化程序设计语言 C 语言为目的, 以具体常见的计算机硬件与操作系统为背景, 讲述在微机上编写精巧、美观、友好的应用软件。本书内容与实例并进, 对 C 语言编程做了深入和具体的探讨。涉及的知识面广, 从内存管理, 到文本屏幕界面设计, 再到文件操作、图形图像操作、中断、通信技术、基本总线接口编程和游戏开发应用, 几乎涵盖了 C 语言编程的方方面面。通过对本书的学习, 读者将对 C 语言高级编程有一个深刻的理解和掌握。

本书在内容安排上采用内容与实例并进的方式。每一章中, 先讲述相关内容, 然后通过实例解析讲述内容的具体应用。实例解析注重对实例的分析, 对方法的详细讲解, 以及对知识点的归纳总结。本书对各个实例以需求分析、源代码解析、运行结果和归纳总结为主线进行剖析, 通过实例来讲解知识点, 又通过相应的知识点来分析实例。通过研读, 读者可以很快掌握此方面的知识, 并较快将其应用到工程开发中。在结构安排上, 充分考虑了层次性和内容上的循序渐进。

本书包括 9 章, 主要内容如下。

第 1 章为内存管理, 主要针对优化内存使用、提高内存的使用效率、提高程序运行速度。该章介绍了 PC 的存储器结构和基本的堆管理函数, 包括学生信息数据库实例和巧用内存管理创建通信录实例。通过本章的学习, 读者将对 C 语言的内存管理有一个清晰的了解和认识。

第 2 章为文本屏幕界面设计, 包括文本方式控制、窗口设置和文本屏幕输出函数、清屏和光标操作函数、屏幕移动与存取函数、状态查询函数等内容。该章还详细剖析了创建弹出式菜单实例、文本方式下拉式菜单和将屏幕分为两个窗口进行对话的实例。通过本章的学习, 读者可以掌握如何对文本屏幕界面进行操作。

第 3 章为文件高级操作, 介绍标准文件的基本输入输出操作和一些实用的文件处理方法, 包括文件的加密与解密、文件的分割与合并等。通过本章的学习, 读者将会逐步掌握一些实用的文件处理技巧。

第 4 章为图形图像, 介绍如何使用 Turbo C 提供的图形函数绘制基本的图形(包括画点、线、矩形和条形图, 画圆、椭圆和扇形图等)、对图形的颜色控制函数、封闭图形的填色函数及有关画图函数、屏幕操作函数、图形方式下的文本输出函数、动画技术、菜单生成技术、图形程序使用环境、直接存储存取和 SVGA 编程, 还介绍使用这些基本图形的实际应用, 图形图像综合应用——用动画演示排序算法。通过本章的学习, 读者将逐步掌握如何使用 C 语言绘制图形。

第 5 章为中断, 介绍中断的基本概念、鼠标中断编程的应用、键盘中断编程的应用和用中断方式制作的菜单等。通过本章的学习, 读者将掌握对鼠标和键盘的中断控制。

第6章为通信技术, 主要介绍 Winsock 编程基础、Cscoket 编程技术、串并口编程, 双机连接方法等, 并通过双机并口通信实例和网络通信编程实例的剖析讲述了通信技术的实际应用。通过本部分的学习, 读者将逐步掌握通信技术。

第7章为基本总线接口编程, 主要介绍 ISA 总线、PCI 总线、PCI 总线配置和采用中断方式的信号采集程序。通过本章的学习, 读者将对总线接口编程有一定了解。

第8章为游戏开发, 介绍 DOS 环境下的 C 语言游戏编程, 是综合实例。魔方游戏实例, 实际上是对直接写屏、中断、多任务、内存技术、动画技术、显示技术和优化算法等各种编程技术, 及内存、声卡、键盘、鼠标等各种硬件原理和硬件编程的综合应用。

第9章是综合开发实例——信息管理系统。本部分将重点向读者介绍如何采用模块化程序设计思想来设计一个综合的 C 程序, 提高读者编写大型程序的能力。

本书主要针对 C 语言程序设计的高级读者, 适合已经熟悉和掌握了 C 语言的基础知识, 想用 C 语言开发更完美的软件, 更为深入地了解 C 语言的一些细节和使用技巧的读者。本书可以使学习过 C 语言语法的初学者快速成长为 C 语言编程高手, 也可以作为工程人员的参考手册。

本书由王为青、刘变红编著, 参与本书编辑和修改的还有徐强、刘艳伟、吴洋、房明浩、严雨、王亮、梅乐夫、刘艳祎等同志。在此, 对以上人员致以诚挚的谢意!

由于本书涉及的范围比较广泛, 作者的经验有限, 书中难免有不足或错误之处, 敬请各位读者批评指正。

编者

2006.10

光盘使用说明

本书附带了一张光盘，具体情况介绍如下。

程序运行环境

硬件环境：要求计算机 CPU 的主频在 500MHz 以上，内存在 128MB 以上。

软件环境：操作系统是 Windows 98/Me/2000/NT/XP，调试环境是 Turbo C for Windows 集成实验与学习环境编译器。另外，部分“串并口通信”的实例需要使用汇编语言来调试、运行。

光盘主要内容

1. /code 目录，包括了全书 9 章的所有源代码、可执行程序、相应的数据文件、使用说明文件等。
2. /tool 目录，包括一个 Turbo C for Windows 集成实验与学习环境编译器的安装文件 TCSetup611.exe。

光盘的使用方法

首先安装 Turbo C for Windows 集成实验与学习环境编译器，然后将光盘中的源代码拷贝到本地机器硬盘上。该环境的具体使用有以下几个步骤。

1. 打开 Turbo C for Windows 集成实验与学习环境共享版界面。如图 1 所示。



图 1 运行环境

2. 打开编译程序源代码。通过菜单“文件”|“打开”打开源代码所在路径。比如打开“第1章 程序源代码”中的“例 1-1 malloc 函数”，如图 2 所示。

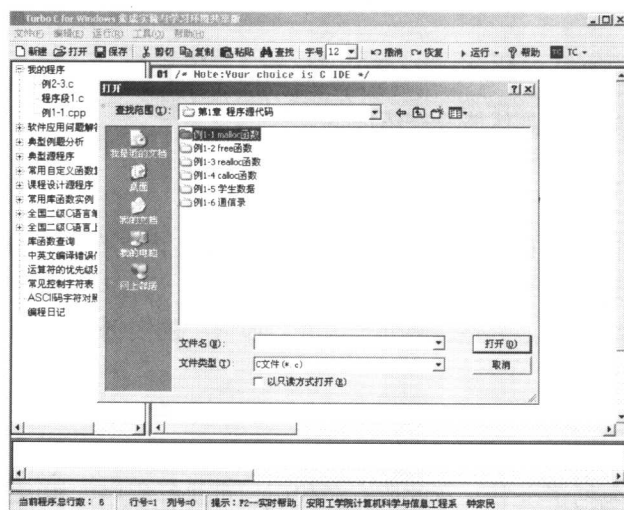


图 2 打开源代码所在文件夹

选择“例 1-1 malloc 函数”文件夹下的“例 1-1.c”，单击“打开”按钮，打开源文件。如图 3 所示。



图 3 打开源程序

3. 运行打开程序。打开源代码后，单击“运行”菜单命令，编译并运行打开的程序。运行结果如图 4 所示。

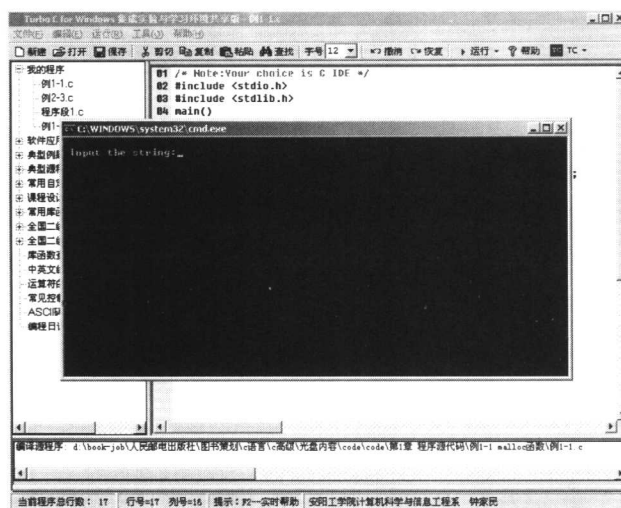


图4 编译并运行

4. 调试程序，运行结束。

目 录

第 1 章 内存管理	1
1.1 预备知识	1
1.1.1 PC 存储器结构	1
1.1.2 CPU 地址线宽度	3
1.1.3 内存管理方式	5
1.1.4 内存编译模式	6
1.1.5 堆概念和结构	9
1.1.6 堆管理函数	10
1.2 高速分配内存技巧	15
1.2.1 类型声明及变量定义	15
1.2.2 函数剖析	16
1.2.3 归纳总结	18
1.3 学生信息数据库实例	18
1.3.1 需求分析	19
1.3.2 源代码解析	19
1.3.3 运行结果	23
1.3.4 归纳总结	23
1.4 巧用内存管理创建通信录	25
1.4.1 需求分析	25
1.4.2 源代码解析	25
1.4.3 程序运行结果	31
1.4.4 归纳总结	32
1.5 小结	32
第 2 章 文本屏幕界面设计	33
2.1 文本方式的控制	33
2.1.1 文本方式控制函数	33
2.1.2 文本方式颜色控制函数	34
2.1.3 字符显示亮度控制函数	36
2.1.4 实例解析	36
2.2 窗口设置和文本输出函数	38
2.2.1 窗口设置函数	38

2.2.2	控制台文本输出函数	38
2.2.3	实例解析	38
2.3	清屏和光标操作函数	40
2.3.1	清屏函数	40
2.3.2	光标操作函数	41
2.3.3	实例解析	41
2.4	屏幕文本移动与存取函数	43
2.4.1	屏幕文本移动函数	43
2.4.2	屏幕文本存取函数	43
2.4.3	实例解析	44
2.5	状态查询函数	46
2.5.1	状态查询函数	46
2.5.2	实例解析	47
2.6	创建弹出式菜单实例	48
2.6.1	需求分析	48
2.6.2	源代码解析	49
2.6.3	运行结果	55
2.6.4	归纳总结	55
2.7	文本方式下创建下拉式菜单	56
2.7.1	需求分析	56
2.7.2	源代码解析	56
2.7.3	运行结果	61
2.7.4	归纳总结	62
2.8	综合应用	62
2.8.1	需求分析	62
2.8.2	源代码解析	63
2.8.3	运行结果	66
2.8.4	归纳总结	66
2.9	小结	67
第 3 章	文件高级操作	68
3.1	文件的基本概念	68
3.2	标准文件的输入输出操作	68
3.2.1	标准文件输入输出	70
3.2.2	标准文件打开函数 <code>fopen()</code>	71
3.2.3	标准文件关闭函数 <code>fclose()</code>	74
3.2.4	标准文件的读写	75
3.2.5	文件的随机读写函数	78
3.2.6	实例解析	82

3.3 文件的加密与解密	83
3.3.1 移位加密法	83
3.3.2 伪随机数加密法	84
3.3.3 命令加密法	86
3.3.4 逐字节异或加密法	88
3.4 文件分割程序	91
3.4.1 操作方法	91
3.4.2 源代码解析	91
3.4.3 运行结果	94
3.4.4 归纳总结	94
3.5 文件合并程序	94
3.5.1 操作方法	94
3.5.2 源代码解析	95
3.5.3 运行结果	97
3.5.4 归纳总结	97
3.6 小结	97
第4章 图形图像	98
4.1 图形显示基本概念	98
4.1.1 图形显示的坐标	98
4.1.2 像素	99
4.2 图形函数	99
4.2.1 图形系统的初始化	99
4.2.2 恢复显示方式和清屏函数	102
4.2.3 基本图形函数	102
4.3 颜色控制函数	107
4.3.1 颜色设置函数	108
4.3.2 调色板颜色设置	110
4.4 画线的线型函数	114
4.4.1 设定线型函数	115
4.4.2 得到当前画线信息的函数	117
4.5 填色函数及相关作图函数	118
4.5.1 填色函数	118
4.5.2 用户自定义填充函数	119
4.5.3 得到填充模式和颜色的函数	121
4.5.4 与填充函数有关的作图函数	121
4.5.5 可对任意封闭图形填充的函数	124
4.6 屏幕操作函数	125
4.6.1 屏幕图像存储和显示函数	125

4.6.2 设置显示页函数	127
4.7 图形方式下的文本输出函数	130
4.7.1 文本输出函数	131
4.7.2 文本输出字符串函数	133
4.7.3 定义文本字型函数	135
4.8 动画技术	137
4.8.1 动态开辟图视口的方法	137
4.8.2 利用显示页和编辑页交替变化	138
4.8.3 利用画面存储再重放的方法	139
4.8.4 直接对图像动态存储器进行操作	141
4.9 菜单生成技术	141
4.10 图形程序使用环境	142
4.10.1 BGI 使用	143
4.10.2 图形方式下字符输出的条件	144
4.10.3 BGI 图形驱动	145
4.11 直接存储存取	145
4.11.1 BIOS 中断在显示中的应用	147
4.11.2 VGA 寄存器	149
4.11.3 屏幕图形与 VRAM 地址的关系	151
4.11.4 VRAM 的位面结构	151
4.11.5 将 VRAM 位面信息存入文件	152
4.11.6 将文件图像信息写入 VRAM 位面	153
4.11.7 VGA 标准图形模式 12H 编程	154
4.11.8 VGA 标准图形模式 13H 编程	157
4.12 SVGA 编程	157
4.12.1 SVGA 显卡的检测	158
4.12.2 SVGA 模式信息的获取与模式操作	160
4.12.3 SVGA 的直接存储器显示与内存控制	162
4.13 综合应用实例	163
4.13.1 用户自定义图模填充长方框图像	163
4.13.2 画圆饼图程序	165
4.13.3 画条形图程序	167
4.13.4 画函数曲线	169
4.14 图形图像综合应用——用动画演示排序算法	171
4.14.1 程序介绍	171
4.14.2 源代码解析	172
4.14.3 运行结果	183
4.14.4 归纳总结	184
4.15 小结	184

第 5 章 中断	185
5.1 中断的基本概念	185
5.1.1 BIOS	185
5.1.2 中断和异常	186
5.1.3 BIOS 功能调用	189
5.2 鼠标中断编程的应用实例	191
5.2.1 鼠标中断的基本概念	191
5.2.2 程序功能分析	198
5.2.3 源代码解析	199
5.2.4 归纳总结	202
5.3 键盘中断编程的应用实例	203
5.3.1 键盘中断的基本概念	203
5.3.2 键盘操作函数 bioskey()	207
5.4 鼠标与键盘的综合应用实例	208
5.4.1 需求分析	208
5.4.2 源代码解析	208
5.4.3 运行结果	212
5.4.4 归纳总结	213
5.5 中断应用——菜单制作程序剖析	213
5.5.1 需求分析	213
5.5.2 源代码解析	214
5.5.3 运行结果	227
5.5.4 归纳总结	227
5.6 小结	228
第 6 章 通信技术	229
6.1 概述	229
6.2 Winsock 编程基础	230
6.2.1 Winsock 基本概念	230
6.2.2 Winsock 基本 API	230
6.2.3 关于 Winsock 的异步模式	234
6.3 Csocket 编程技术	238
6.4 串口编程	238
6.4.1 概念	239
6.4.2 串行接口	239
6.4.3 串行通信方式及异步通信协议	240
6.4.4 串口针脚功能	243
6.5 并口编程	244

6.5.1 概念	244
6.5.2 并行接口	244
6.5.3 并口引脚功能	245
6.6 串并口操作的输入/输出函数	246
6.6.1 输入函数	246
6.6.2 输出函数	246
6.7 双机连接的方法	247
6.8 双机并口通信实例	248
6.8.1 源代码解析	249
6.8.2 归纳总结	280
6.9 网络通信编程——聊天室实例	281
6.9.1 需求分析	281
6.9.2 聊天室服务器端程序分析	282
6.9.3 聊天室客户端程序分析	290
6.10 小结	297
第 7 章 基本总线接口编程	298
7.1 ISA 总线	298
7.1.1 ISA 总线简介	298
7.1.2 ISA 引脚介绍	299
7.2 PCI 总线	302
7.2.1 PCI 总线简介	302
7.2.2 PCI 引脚介绍	303
7.3 中断控制操作	306
7.3.1 软件中断	307
7.3.2 硬件中断	307
7.4 PCI 总线配置	308
7.4.1 PCI 总线配置空间及配置机制	308
7.4.2 用 I/O 命令访问 PCI 总线配置空间	309
7.5 采用中断方式的信号采集程序	311
7.5.1 功能分析	311
7.5.2 源代码解析	311
7.6 小结	316
第 8 章 游戏开发	317
8.1 游戏开发概述	317
8.2 BMP 图像格式	318
8.3 TC 环境下的 256 色显示	324
8.3.1 VGA 的 DAC 色彩寄存器知识	324

8.3.2	置 256 色图形模式	324
8.3.3	访问显存	325
8.3.4	显示卡换页	326
8.3.5	硬件无关屏幕初始化	327
8.4	魔方游戏开发程序剖析	329
8.4.1	功能分析	330
8.4.2	鼠标驱动程序	330
8.4.3	主函数模块	344
8.4.4	初始化图形 to_graphic_mode 模块	347
8.4.5	初始化鼠标 initialize 模块	347
8.4.6	显示程序的作者 word 模块	348
8.4.7	显示游戏标题 title 模块	350
8.4.8	绘制游戏主窗口的 draw_win 模块	352
8.4.9	建立魔方游戏界面 body 模块	353
8.4.10	显示魔方游戏 showcube 模块	362
8.4.11	清除键盘缓冲区 clr_kb_buff 模块	363
8.4.12	返回鼠标按下操作键值 getmousech 模块	363
8.4.13	处理用户对功能热键的操作 handle 模块	364
8.5	小结	377
第 9 章	综合开发实例——信息管理系统	378
9.1	问题定义	378
9.2	算法设计	378
9.2.1	主函数算法	379
9.2.2	各模块算法	379
9.3	流程图设计	381
9.3.1	主函数模块的流程图	381
9.3.2	其他各模块的流程图	382
9.4	编写程序代码	392
9.4.1	基本介绍	392
9.4.2	信息管理系统程序代码	392
9.5	测试与调试	410
9.5.1	基本介绍	410
9.5.2	信息管理系统测试与调试	411
9.6	整理文档	412
9.7	系统维护	412
9.8	归纳总结	412

第1章 内存管理

Turbo C 编译程序对源程序进行编译时, 将函数中命令、语句编译成相应序列的机器指令代码, 放在代码段中; 将已初始化的数据, 如已赋值的全局变量、静态局部变量等, 放在数据段内; 将未初始化的数据放在 BSS 段内; 将临时数据, 如函数调用时传递的参数、局部变量、返回调用时的地址等放在栈段内; 而对一些动态变化的数据, 如在程序执行中建立的一些数据结构, 如链表、动态数组等, 则放在堆结构中。因此, 应对 CPU 存储器结构和编译模式作些了解。本章中, 主要讲解有关内存管理的 C 语言程序设计。

1.1 预备知识

内存管理子是操作系统的重要部分。从计算机发展的早期开始, 就存在着对内存的实际需要大于系统所能提供的物理意义上的内存大小的问题, 因此, 如何优化内存管理, 提高内存的使用效率, 尽可能地提高系统运行速度就显得尤为重要。这在 C 语言程序设计中就表现为对内存管理的编程和维护。

在 C 语言程序设计中, 程序员使用库函数 `malloc()` 和 `free()` 来分配和释放内存。再次释放已经释放过的内存块或者释放未被分配的内存块, 都会造成系统的崩溃, 而忘记释放不再使用的内存块则会逐渐耗尽系统资源。因此, 在进行 C 语言程序设计之前, 简单介绍一下 PC 机存储器的结构及基本的堆管理函数。

本节主要内容如下:

- PC 机存储器结构;
- CPU 地址线宽度数;
- 微机常用操作系统的内存管理方式;
- 实地址下的内存编译模式;
- 堆概念和结构;
- 堆管理函数。

1.1.1 PC 存储器结构

存储器由许许多多的基本存储单元排列成矩阵形式, 并加上地址选择及读写控制等逻辑电路构成。当 CPU 要从存储器中读取数据时, 就会选择存储器中某一地址, 并将该地址上存储单元所存储的内容读走。

PC 机的存储器分为主存储器、外存储器和高速缓存 (Cache) 几个部分, 如图 1.1 所示。

1. 主存储器

主存储器是能由 CPU 直接编程访问的存储器，存放需要执行的程序和需要处理的数据。它通常归在主机范畴之内，称为内存。从用户编程的角度看，应注意信息的存取方式，它影响到存储信息的组织，可以把主存储器分为随机存储器（SRAM 和 DRAM）和只读存储器（ROM）。

（1）随机存储器（SRAM 和 DRAM）

随机存储器（Random Access Memory, RAM）是构成内存的主要部分，它可以根据需要随时读出或写入内容，又称随机读写存储器。随机读写存储器又分为静态半导体存储器（Static RAM, SRAM）和动态半导体存储器（Dynamic RAM, DRAM）两种。

静态半导体存储器集成度低，但是不需要刷新，在不加电的情况下信息能够一直保持而不会丢失。静态半导体存储器速度快，电路简单，但是容量小，价格高。

动态半导体存储器集成度高，但是需要刷新，断电后 RAM 中的内容将全部丢失，是非永久性存储介质，动态半导体存储器比静态半导体存储器便宜。

（2）只读存储器（ROM）

只读存储器（Read Only Memory, ROM）在正常工作状态下其内容只能读出而不能写入。只读存储器一般用于存储计算机系统中固定的程序和数据，如引导程序、监控程序等。

只读存储器可以分为 3 种。

- PROM（Programmable ROM）是可编程只读存储器，允许用户一次性写入。
- EPROM（Erasable PROM）是可擦除的可编程只读存储器，可通过紫外光照射来擦除其内容后重新写入，并可以重复多次。
- EEPROM（Electrical EPROM）是电可擦除的可编程只读存储器，可不从电路板上拔下芯片，利用电信号擦除和写入。

2. 外存储器

由于内存容量有限（受地址位数、成本和速度等因素制约），在大多数计算机系统中设置了一些大容量存储器，作为对内存的补充和后援。它们位于传统主机的逻辑范畴之外，常称为外存储器，简称外存。

外存用来存放需要联机保存但暂不使用的程序与数据。通常的外存有软盘存储器、硬盘存储器、磁带存储器、光盘存储器等。

（1）软盘存储器

早期的软盘为 5.25 英寸，低密度盘的容量为 360KB，高密度盘的容量是 1.2MB。目前多采用 3.5 英寸软盘，其容量为 1.44MB。

（2）硬盘存储器

硬盘的存储空间由若干盘面组成，每面有若干柱面（磁道），每道又分为若干扇区，其乘

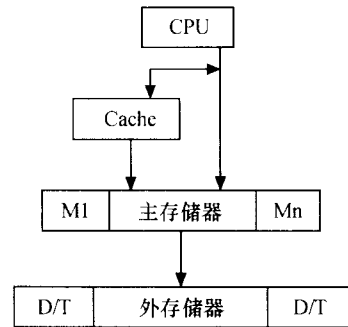


图 1.1 PC 机存储器结构图