

存储原理与数据恢复 探究教程

田素诚 魏毅 编著

中国人民公安大学出版社

群众出版社

图书在版编目 (CIP) 数据

存储原理与数据恢复探究教程 / 田素诚, 魏毅编著. —北京: 中国人民公安大学出版社, 2019. 7

ISBN 978 - 7 - 5653 - 3667 - 6

I. ①存… II. ①田… ②魏… III. ①计算机网络—信息存储—高等学校—教材
②数据管理—文件恢复—高等学校—教材 IV. ①TP393②TP309.3

中国版本图书馆 CIP 数据核字 (2019) 第 130008 号

存储原理与数据恢复探究教程

田素诚 魏毅 编著

出版发行: 中国人民公安大学出版社

地 址: 北京市西城区木樨地南里

邮政编码: 100038

经 销: 新华书店

印 刷: 北京市泰锐印刷有限责任公司

版 次: 2019 年 9 月第 1 版

印 次: 2019 年 9 月第 1 次

印 张: 17.25

开 本: 787 毫米 × 1092 毫米 1/16

字 数: 345 千字

书 号: ISBN 978 - 7 - 5653 - 3667 - 6

定 价: 55.00 元

网 址: www.cppsups.com.cn www.porclub.com.cn

电子邮箱: zbs@cppsup.com zbs@cppsu.edu.cn

营销中心电话: 010 - 83903254

读者服务部电话 (门市): 010 - 83903257

警官读者俱乐部电话 (网购、邮购): 010 - 83903253

教材分社电话: 010 - 83903259

本社图书出现印装质量问题, 由本社负责退换
版权所有 侵权必究

内 容 简 介

本书先介绍了存储原理的基础知识，然后简单介绍了存储介质的工作原理，介绍了数据恢复重要概念、过程和原则，为学习存储原理的理论知识做好必要的铺垫。本书的核心内容可以概括为两种分区模式、三种文件系统，即 MBR 分区模式和 GPT 分区模式，FAT32、NTFS 和 exFAT 文件系统，这是存储原理最基础的内容，也是最重要、最核心的内容。通过学习这些理论知识，我们就能理解磁盘和文件系统的工作原理，为完成数据恢复任务奠定理论基础。本书没有把数据恢复和存储原理割裂开，而是将数据恢复作为存储原理的实践应用展开讲解，这既有利于理解存储原理，也有利于建立数据恢复的思路。

本书为公安院校开设的存储原理与数据恢复课程而编写，也可作为普通院校的存储原理与数据恢复相关基础课程的教材。本书语言平实，简明易懂，逻辑关系清晰，突出实践，也可作为相关读者自学参考用书。

前 言

学习存储原理的重要目的是获得数据恢复的理论基础，从这个角度来说，数据恢复是宴席的主菜，存储原理只是开胃菜。然而现实世界并非想象的那么简单。

一、学习存储原理与数据恢复面对的困境

如果是偶然恢复自己的数据文件，使用数据恢复软件就很好，如果这种快餐式的应用不能满足您的需要，那您就求助于专业人员。如果您想成为数据恢复的专业人士，那您就需要掌握更为专业的存储原理知识，这样您就能应对更为复杂的数据恢复任务。既然主要目的是数据恢复，那就少学一点存储原理吧！还真不行，存储原理有自身的知识体系，学太简略，就犹如一锅夹生饭，口感不好，吃下去还会消化不良！了解底层功能的实现方法，可以增加您思考、处理问题的信心和底气，可以帮助您提出另辟蹊径的解决方法。既然如此，那就多学一点存储原理吧！可是存储原理理论性强、内容比较枯燥，面对全是十六进制数字，很容易失去学习兴趣。

二、解决学习存储原理困境的方法

在学习存储原理与数据恢复过程中充满了矛盾，怎样才能学好存储原理理论知识，更好地掌握数据恢复技能呢？

存储原理有自身的知识体系，必须构建足够的、合理的知识框架，才算进入存储原理领域。为了适应将来复杂的技术环境，要尊重存储原理知识体系，不可以任意肢解。

改进存储原理的学习方法才是解决问题的关键。存储原理的知识体系非常独立，在日常生活中很难接触到，很难获得直观的经验。从建构主义学习理论来看，通过自身经验获得知识，这样知识结构才更加稳固，这样的学习才更加有效。

直接呈现理论知识往往让人很难接受，本书提供大量案例，使空洞的理论有直观的实际内容作为支撑。本书更强调学习者通过实践观察验证理论知识，书中的绝大多数案例是可以直接实践验证的。通过实践，不仅能提高学习者的实践动手能力，更能获得直观体验，把空洞的理论转换成可见的实验现象，这样就能够调动学习者的主观能动性，让学习者沉浸在底层数据的学习环境中，有更强的学习兴趣，获得更好的学习效果。

正如本书的书名所示，本书亮点、关键点就在“探究”二字，通过探究实验获得经验，经验是知识体系内化的催化剂，是建构知识体系的最好黏合剂，可以帮助您形成稳定的知识体系。本书力图将探究式学习贯穿始终。

三、本书特色

1. 基于探究式学习设计教材内容，能够获得更好的学习效果。
2. 实例丰富，理论与实践相结合。
3. 不追求专业深度，但求易于接受，语言平实，内容简明易懂，循序渐进，非常具有亲和力。

四、学习建议

1. 在实践中学习、在探究中提高。实践是最佳学习方式，不但能提高对理论知识的理解，更能提高实践动手能力。探究式学习，可以帮助

自己沉浸在当前的问题中，在不知不觉中获得知识、构建知识体系。在探究实践中验证自己的猜想，就像找到迷宫出口一样，总是令人兴奋的。

2. 记住重要的知识点。我们身处网络世界，可以随时百度、Google，但只有掌握了足够的知识基础，才能更快速、更深刻地理解新知识点，提升操作能力。总是依靠查阅书籍，依靠百度、Google，永远不能成为真正的高手。记忆琐碎的知识点很艰苦，但所有吃过的苦都会有回报！

3. 相信存储原理的科学性。存储技术方案的设计者都是技术精英，每一个技术方案在当时的环境下，一定是最佳方案。相信存储原理的科学性，消灭抵触情绪，会让您更容易建立知识体系。

五、课程考核方式的建议

如果选择本书作为一门课程的教材，那就必须考虑到教学内容、教学方式与期末考试相适应的问题，既能发挥期末考试的指挥棒作用，也能让采用正确学习方法的学生取得好成绩。

作者建议，采用 180 分钟的开卷考试。存储原理涉及知识点非常多，对知识点准确度的要求非常高，要求学生准确记忆所有知识点确实有点难为学生，对学生而言，更重要的是对存储原理整体上的理解以及对存储原理的正确应用。我们在教学中强调实践应用，在试题中就要体现对实践应用能力的考核，采用开卷能够满足上面的要求。

六、出版预告

作者计划在 2020 年出版与本书配套的《存储原理与数据恢复习题与解析》。该书可以作为教师的辅助教学资料及出具期末试题的参考，也可以作为学习者检验学习成果、磨炼底层数据处理思维的工具书。

编著者

2019 年 6 月

目录 | CONTENTS

3.7.3	扩展分区的逻辑结构	50
3.7.4	探究案例——扩展分区“链”式结构形成过程	50
3.7.5	再论MBR的磁盘分区表	56
3.8	计算逻辑分区的绝对地址	66
3.8.1	逻辑分区结构体绝对地址的计算公式	67
3.8.2	逻辑分区绝对地址的计算公式	67
3.8.3	应用案例——逻辑分区绝对地址的计算	67
3.9	在WinHex查看EBR	69
3.10	在WinHex查看EBR	69
3.11	在WinHex查看EBR	69
3.12	在WinHex查看EBR	69
3.13	在WinHex查看EBR	69
3.14	在WinHex查看EBR	69
3.15	在WinHex查看EBR	69
3.16	在WinHex查看EBR	69
3.17	在WinHex查看EBR	69
3.18	在WinHex查看EBR	69
3.19	在WinHex查看EBR	69
3.20	在WinHex查看EBR	69
3.21	在WinHex查看EBR	69
3.22	在WinHex查看EBR	69
3.23	在WinHex查看EBR	69
3.24	在WinHex查看EBR	69
3.25	在WinHex查看EBR	69
3.26	在WinHex查看EBR	69
3.27	在WinHex查看EBR	69
3.28	在WinHex查看EBR	69
3.29	在WinHex查看EBR	69
3.30	在WinHex查看EBR	69
3.31	在WinHex查看EBR	69
3.32	在WinHex查看EBR	69
3.33	在WinHex查看EBR	69
3.34	在WinHex查看EBR	69
3.35	在WinHex查看EBR	69
3.36	在WinHex查看EBR	69
3.37	在WinHex查看EBR	69
3.38	在WinHex查看EBR	69
3.39	在WinHex查看EBR	69
3.40	在WinHex查看EBR	69
3.41	在WinHex查看EBR	69
3.42	在WinHex查看EBR	69
3.43	在WinHex查看EBR	69
3.44	在WinHex查看EBR	69
3.45	在WinHex查看EBR	69
3.46	在WinHex查看EBR	69
3.47	在WinHex查看EBR	69
3.48	在WinHex查看EBR	69
3.49	在WinHex查看EBR	69
3.50	在WinHex查看EBR	69
3.51	在WinHex查看EBR	69
3.52	在WinHex查看EBR	69
3.53	在WinHex查看EBR	69
3.54	在WinHex查看EBR	69
3.55	在WinHex查看EBR	69
3.56	在WinHex查看EBR	69
3.57	在WinHex查看EBR	69
3.58	在WinHex查看EBR	69
3.59	在WinHex查看EBR	69
3.60	在WinHex查看EBR	69
3.61	在WinHex查看EBR	69
3.62	在WinHex查看EBR	69
3.63	在WinHex查看EBR	69
3.64	在WinHex查看EBR	69
3.65	在WinHex查看EBR	69
3.66	在WinHex查看EBR	69
3.67	在WinHex查看EBR	69
3.68	在WinHex查看EBR	69
3.69	在WinHex查看EBR	69
3.70	在WinHex查看EBR	69
3.71	在WinHex查看EBR	69
3.72	在WinHex查看EBR	69
3.73	在WinHex查看EBR	69
3.74	在WinHex查看EBR	69
3.75	在WinHex查看EBR	69
3.76	在WinHex查看EBR	69
3.77	在WinHex查看EBR	69
3.78	在WinHex查看EBR	69
3.79	在WinHex查看EBR	69
3.80	在WinHex查看EBR	69
3.81	在WinHex查看EBR	69
3.82	在WinHex查看EBR	69
3.83	在WinHex查看EBR	69
3.84	在WinHex查看EBR	69
3.85	在WinHex查看EBR	69
3.86	在WinHex查看EBR	69
3.87	在WinHex查看EBR	69
3.88	在WinHex查看EBR	69
3.89	在WinHex查看EBR	69
3.90	在WinHex查看EBR	69
3.91	在WinHex查看EBR	69
3.92	在WinHex查看EBR	69
3.93	在WinHex查看EBR	69
3.94	在WinHex查看EBR	69
3.95	在WinHex查看EBR	69
3.96	在WinHex查看EBR	69
3.97	在WinHex查看EBR	69
3.98	在WinHex查看EBR	69
3.99	在WinHex查看EBR	69
3.100	在WinHex查看EBR	69
1	存储原理基础知识	1
1.1	十六进制数制	1
1.1.1	二进制数制	1
1.1.2	十六进制数制	2
1.2	字符编码概览	3
1.2.1	ASCII 码	4
1.2.2	中文编码	5
1.2.3	Unicode 编码和 UTF	6
1.2.4	代码页	8
1.3	Windows 7 虚拟硬盘的使用方法	8
1.3.1	创建虚拟硬盘	9
1.3.2	虚拟硬盘的磁盘初始化	10
1.3.3	虚拟硬盘分区	11
1.3.4	分离虚拟硬盘	13
1.3.5	附加虚拟硬盘	13
1.4	WinHex 基础	14
1.4.1	WinHex 的程序界面	14
1.4.2	WinHex 的工作区	16
1.4.3	WinHex 中的打开操作	19
1.4.4	WinHex 中的跳转操作	21
1.4.5	WinHex 中的搜索功能	22
1.4.6	“程序员”型计算器的使用	24
	习题	25

2 存储介质与数据恢复概述	27
2.1 数据概念的界定	27
2.2 主要存储介质简介	27
2.2.1 磁性存储介质	28
2.2.2 电性存储介质	28
2.2.3 光性存储介质	29
2.2.4 三种存储介质比较	29
2.3 硬盘基础知识	30
2.3.1 机械硬盘的发展简史	30
2.3.2 机械硬盘的物理结构	31
2.3.3 机械硬盘的工作原理	33
2.3.4 固态硬盘	34
2.3.5 硬盘接口	35
2.3.6 磁盘的存储结构	38
2.3.7 磁盘的寻址方式	39
2.4 数据恢复概述	40
2.4.1 数据丢失的原因	40
2.4.2 数据恢复的概念	41
2.4.3 机械硬盘的数据恢复方法	41
2.4.4 数据恢复的基本原则	42
习题	43
3 MBR 分区模式	44
3.1 分区的意义	44
3.2 分区与卷的关系	44
3.3 主引导记录 (MBR) 简介	45
3.4 操作系统启动过程及 MBR 的作用	47
3.5 MBR 产生过程及磁盘管理架构的形成	48
3.6 磁盘分区表的结构和作用	51
3.6.1 磁盘分区表表项的结构	51
3.6.2 文件系统类型代码	54
3.6.3 MBR 层次的硬盘结构	56
3.7 扩展分区结构	57
3.7.1 EBR 的结构	57
3.7.2 扩展分区的物理结构	58

3.7.3 扩展分区的逻辑结构	59
3.7.4 探究案例——扩展分区“链”式结构形成过程	60
3.7.5 再论 EBR 的磁盘分区表	66
3.8 计算逻辑分区的绝对地址	66
3.8.1 逻辑分区结构体绝对地址的计算公式	67
3.8.2 逻辑分区绝对地址的计算公式	67
3.8.3 应用案例——逻辑分区绝对地址的计算	67
3.9 在 WinHex 查看 EBR	69
习题	69
4 FAT32 文件系统	71
4.1 文件系统简介	71
4.1.1 文件系统的概念	71
4.1.2 存储空间的单位	71
4.1.3 文件系统的核心要素	72
4.1.4 常见文件系统简述	73
4.1.5 FAT 系列文件系统简介	75
4.2 FAT32 文件系统的结构	76
4.2.1 FAT32 文件系统的整体结构	76
4.2.2 DBR 的功能与结构	77
4.2.3 文件系统参数块 (BPB) 的结构	79
4.3 FAT32 文件系统的目录	83
4.3.1 基本目录项	83
4.3.2 长文件名目录项	88
4.3.3 文件目录项的排列方式及文件改名	92
4.4 文件存储层次结构的表示方法	95
4.5 FAT 的功能	100
4.5.1 FAT 的工作原理	100
4.5.2 FAT 应用分析	103
4.6 删除文件的处理方法和文件恢复原理	108
4.7 FAT32 文件系统恢复	114
4.8 文件的存储策略	119
4.9 探究实验——FAT32 文件系统综合操作	120
习题	124

5	NTFS 文件系统	126
5.1	NTFS 概述	126
5.1.1	NTFS 的发展历程	126
5.1.2	NTFS 的核心设计思想	127
5.1.3	NTFS 的技术特性	128
5.1.4	NTFS 总体布局	129
5.1.5	NTFS 元文件简介	130
5.1.6	案例探究——了解 NTFS 系统结构	134
5.2	NTFS 的 DBR 扇区	138
5.2.1	DBR 扇区的结构	138
5.2.2	文件系统参数块的结构和功能	139
5.3	MFT 主文件表	143
5.3.1	MFT 记录项的结构	144
5.3.2	MFT 记录项的记录头	145
5.3.3	MFT 记录项的属性概述	149
5.3.4	案例探究——了解 MFT 记录项	154
5.3.5	常驻属性的属性头结构	158
5.3.6	非常驻属性的属性头结构	162
5.3.7	非常驻属性的属性体	167
5.3.8	标准信息属性的属性体	172
5.3.9	文件名属性的属性体	176
5.4	文件层次结构实现方法	182
5.4.1	索引的逻辑结构概述	183
5.4.2	索引项的结构	184
5.4.3	索引相关属性解析	189
5.4.4	索引块	193
5.4.5	文件夹的索引结构实例解析	197
5.5	位图属性和 \$ Bitmap 的工作原理	205
5.6	删除文件的处理方法	210
5.7	探究实验——重建 DBR 扇区	211
5.8	探究实验——NTFS 综合操作	213
5.9	重建 MBR	217
5.9.1	知识回顾与重建 MBR 思路	218
5.9.2	重建 MBR 的过程	221
5.9.3	探究实验——重建 MBR	222
	习题	223

6 exFAT 文件系统	226
6.1 exFAT 文件系统简介	226
6.1.1 exFAT 文件系统的优点	226
6.1.2 exFAT 文件系统在移动存储设备中的应用	227
6.2 exFAT 文件系统的系统结构	227
6.3 DBR 及保留扇区	230
6.4 exFAT 文件系统的目录	234
6.4.1 exFAT 文件系统的初始目录项	234
6.4.2 exFAT 文件系统的文件目录项	236
6.4.3 文件夹目录项及文件存储层次结构表示方法	239
6.5 exFAT 文件系统的存储空间管理	241
6.6 删除文件的处理方法	243
6.7 探究实验——exFAT 文件系统综合操作	245
习题	248
7 GPT 分区模式	250
7.1 计算机启动管理平台	250
7.2 GPT 分区模式概述	251
7.3 GPT 分区模式的结构	253
7.4 GPT 头的结构与各项参数的含义	254
7.5 GPT 分区表表项的结构	256
习题	259

1.1.1 二进制数制

在计算机内部,所有信息都是以二进制数形式进行存储、处理和传输的,包括程序指令、数值、文本、图片、音频、视频等。

二进制数制只有0和1两个数字符号,进位基数是2,逢2进1。同样一个数字,符号在不同位置,代表的大小不同,也就是说不同数位的“权值”不同。二进制数整数部分的权值从低位到高位分别 2^0 、 2^1 、 2^2 、 2^3 、 2^4 ……即1、2、4、8、16……小数部分的权值从高位到低位分别 2^{-1} 、 2^{-2} 、 2^{-3} ……即0.5、0.25、0.125……

人类普遍使用的是十进制数,将十进制数转换成二进制数的标准计算方法是:整数部分是“除2取余法”,小数部分是“乘2取整法”。

对于一个不是太大的十进制整数,还可以使用“权值分解法”将十进制整数转换成二进制数。权值分解法是将十进制数分解成尽可能大的、差两倍的权值,然后在转

1 存储原理基础知识

我们身处信息技术高度发达的现代社会，信息技术已经完全融入我们的生活，并扮演着越来越重要的角色。我们每天都要与各种存储设备和数据文件打交道，出现设备故障、误操作在所难免，数据恢复有实实在在的社会需要。要想完成数据恢复任务，掌握基本的存储原理是必要的基础。

为了更好地学习和理解存储原理，我们需要储备必要的基础知识，本章第1节介绍本课程主要的处理对象——十六进制数；第2节介绍字符编码整体情况，为在分析电子数据时选择适当的编码类型奠定基础；第3节介绍虚拟硬盘的使用方法，虚拟磁盘是我们主要的实验对象；第4节介绍 WinHex 的使用方法，该软件是功能强大的底层数据分析、编辑软件，是本书使用的主要软件工具。

1.1

十六进制数制

在信息技术基础相关课程中，有二进制数制、十六进制数制的介绍。由于学习存储原理需要直接与十六进制数打交道，所以读者必须十分熟悉十六进制数，本节将巩固相关知识。

1.1.1 二进制数制

在计算机内部，所有信息都是以二进制数形式进行存储、处理和传输的，包括程序指令、数值、文本、图片、音频、视频等。

二进制数制只有0和1两个数字符号，进位基数是2，逢2进1。同样一个数字，符号在不同位置，代表的大小不同，也就是说不同数位的“权值”不同。二进制数整数部分的权值从低位到高位分别 2^0 、 2^1 、 2^2 、 2^3 、 2^4 ……即1、2、4、8、16……小数部分的权值从高位到低位分别 2^{-1} 、 2^{-2} 、 2^{-3} ……即0.5、0.25、0.125……

人类普遍使用的是十进制数，将十进制数转换成二进制数的标准计算方法是：整数部分是“除2取余法”，小数部分是“乘2取整法”。

对于一个不是太大的十进制整数，还可以使用“权值分解法”将十进制整数转换成二进制数。权值分解法是先十进制数分解成尽可能大的二进制数的权值，然后在转

换成二进制数时，有权值的数位取值为1，没有权值的数位取值为0。例如： $53 = 32 + 16 + 4 + 1 = (110101)_2$ 。

将二进制数转换成十进制数，是将二进制数按权值展开，然后求和。例如： $(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 8 + 2 = 10$ 。

由于二进制数数位长，不好识别和记忆，不便书写表达，因此引入了十六进制数。

1.1.2 十六进制数制

学习存储原理面对的直接对象是十六进制数，使用的主要软件工具 WinHex 也是一个十六进制编辑器。为了顺利学习存储原理，掌握基于存储原理的数据恢复操作，必须熟悉十六进制数。具体来说，至少掌握三方面的知识或技能：能够熟练地将较小的十六进制数（3 位以内）转换成十进制数；熟悉几个常用的十六进制数与十进制数的对应关系，如 $FFH = 255$ 、 $200H = 512$ （一个扇区的大小）、 $400HB = 1024B = 1KB$ ；掌握十六进制数的加减运算。

说明

由于十进制数和十六进制数都用到了 0~9 这十个数字符号，为了避免混淆，在本书中，十六进制数后面必须加上后缀 H，如 81H 是十六进制数 81，不是十进制数 81。另外，在 WinHex 中，地址也是采用十六进制数形式表示，十六进制数作为地址或偏移地址时，前面加前导符 0x，如 0x81 是十六进制数地址或偏移地址。

十六进制数制共有 0~9、A、B、C、D、E、F 十六个数字符号，进位基数是 16，逢 16 进 1。

十六进制数的优势是数位较短，可以非常简便地与二进制数进行转换。

一位十六进制数和四位二进制数一一对应。一位十六进制数与二进制数、十进制数的具体对应关系如表 1-1 所示。

表 1-1 一位十六进制数与二进制数、十进制数的对应关系

十六进制数	二进制数	十进制数	十六进制数	二进制数	十进制数
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	A	1010	10
3	0011	3	B	1011	11
4	0100	4	C	1100	12

(续表)

十六进制数	二进制数	十进制数	十六进制数	二进制数	十进制数
5	0101	5	D	1101	13
6	0110	6	E	1110	14
7	0111	7	F	1111	15

十六进制数与二进制数之间的相互转换非常容易。从十六进制数转换成二进制数的方法是一位拆四位，把一位十六进制数直接写成对应的四位二进制数即可。从二进制数转换成十六进制数时，要先把二进制数从小数点位置开始，向前向后每四个位分成一段，不足四位时用0补齐，然后将四位二进制数写成对应的十六进制数即可。

将十六进制数转换成十进制数，是将十六进制数按权值展开，然后求和。十六进制数整数部分的权值从低位到高位分别是 16^0 、 16^1 、 16^2 ……小数部分的权值从高位到低位分别是 16^{-1} 、 16^{-2} ……如9AB.CH = $9 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 12 \times 16^{-1} = 2304 + 160 + 11 + 0.75 = 2475.75$ 。

十进制数转换十六进制数的计算方法：整数部分采用除16取余法，余数要写成十六进制数，先得到的余数是低位，后得到的余数是高位；小数部分采用乘16取整法，每次得到的整数部分要写成十六进制数，先得到的整数是高位，后得到的整数是低位。

另外，十六进制数与十进制数也可以以二进制数为中间数进行转换。

十六进制数的加法和十进制数的加法类似，都从小数点位置对齐，从最后一位开始，从后往前加，当和大于等于16时产生进位。例如： $78H + 89H = 101H$ 。十六进制数的减法也是一样道理，先按小数点位置对齐，然后从最后一位开始，用被减数减去减数，如果不够减就从上一位借位，借1当16。例如： $123H - ABH = 78H$ 。

1.2

字符编码概览

要想用计算机处理字符，必须给字符编码，即根据一定规则给每一个字符设置一个唯一的编码，在计算机内部，这个编码就代表这个字符，有了编码，字符就可以在计算机及网络中存储、处理和传输。

在人类的发展过程中，不同地区创建了不同的文字符号，为了给这些文字符号编码，不同的国家、不同的国际组织在信息技术发展过程中不断创造出来各种各样的编码规则。这些编码规则各有各的优点，各有各的适用范围。在网络产生和普及后，为了相互交流，一些国际组织还构建了国际通用编码。在国际通用编码中，逐

步将不同国家、不同文字体系的字符融合到一个统一的编码方案中，使国际交流变得越来越简便。

字符编码本身并不算是特别复杂的技术，只要按一定次序给字符分配一个编码即可。字符编码问题的复杂性在于不同编码体系的兼容以及同一字符编码体系的前后版本之间的兼容问题和不同应用领域字符编码的兼容问题等。字符编码在更新过程中存在大量不同步问题，遗留下很多历史问题。

本节对中文字符编码进行一个简要的概述，帮助读者了解一些重要的编码规则，对字符编码形成一个初步的、整体的认识，为分析存储介质中的内容奠定良好的基础。

1.2.1 ASCII 码

ASCII 码 (American Standard Code for Information Interchange, 美国标准信息交换代码) 是一种最常用的西文编码。ASCII 码一共有 128 个编码，使用了一个字节的后 7 位，即存储 ASCII 码的这个字节的最高一位取值总是 0。ASCII 码的前 32 个编码是控制符，即没实际符号与之对应，起到控制作用。ASCII 码的主体内容是 10 个阿拉伯数字、26 个大写英文字母和 26 个小写英文字母。ASCII 码中还包含一些常用的标点和符号。

(1) ASCII 码的使用情况

在 FAT32 文件系统，以英文、数字命名的文件/文件夹是以 ASCII 码形式存储的。在 NTFS 文件系统中，文件名是以 Unicode 编码存储的，在 ASCII 码前面填充 00H，扩展成两个字节就是对应的 Unicode 编码，所以掌握 ASCII 码也能够识别用英文、数字命名的文件/文件夹。在中文版 Windows 操作系统中，通常是以代码页的形式确定文档内容的编码形式，中文操作系统最常用的代码页是 936 (ANSI/OEM - 简体中文 GBK)，代码页 936 与 ASCII 兼容，英文字母和数字等符号也是以 ASCII 码形式存储的。

(2) 常用 ASCII 码

为了更快的理解 WinHex 显示的内容，如文件扩展名、英文文件名等，应该熟悉英文字母、数字符号和几个常用符号的十六进制编码。在表 1-2 中列出了常用字符及其对应的 ASCII 码 (表格中，偶数行 ASCII 码编码是十六进制数)，即大写字母 A~Z 的编码是 41H~5AH；小写字母 a~z 的编码是 61H~7AH；数字符号 0~9 的编码是 30H~39H。使用表 1-2 以外的 ASCII 码字符时，可以到网上查找。

表 1-2 常用 ASCII 码对照表

A	B	C	D	E	F	G	H	I	J	K	L	M
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
4E	4F	50	51	52	53	54	55	56	57	58	59	5A
a	b	c	d	e	f	g	h	i	j	k	l	m
61	62	63	64	65	66	67	68	69	6A	6B	6C	6D
n	o	p	q	r	s	t	u	v	w	x	y	z
6E	6F	70	71	72	73	74	75	76	77	78	79	7A
0	1	8	9			换行	回车	空格	\$.	~
30	31	38	39			0A	0D	20	24	2E	7E

(3) ASCII 码应用案例

下面通过一个简单案例，熟悉一下 ASCII 码在存储空间中实际存储形式。创建一个文本文件，输入三行内容，第一行内容是“abcd”，第二行内容是“ABCD”，第三行内容是“1234”。这个文本文档在磁盘是如何存储的？用 WinHex（参见“1.4 WinHex 基础”）打开这个文本文档，这个文本文档内容的存储形式如图 1-1 所示。第一、第三、第五个矩形框中的内容对应就是“abcd”、“ABCD”、“1234”。

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
61	62	63	64	0D	0A	41	42	43	44	0D	0A	31	32	33	34
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

图 1-1 文本文档的实际存储方式

在文本文档中如何创建一个新段落？通过图 1-1 可以清楚地看出，在当前位置插入“0D0A”两个控制符即可，0DH 代表回车，即插入点移至行首，0AH 代表换行，即把插入点移至下一行。

1.2.2 中文编码

中文编码也经历了从简单到复杂的发展过程，目前有 GB2312、GBK、GB18030 三个主要编码方案。

GB2312（GB 是国标的拼音缩写）编码方案全称是《信息交换用汉字编码字符集》，于 1981 年发布，采用双字节编码，收录汉字 6763 个，非汉字图形字符 682 个。整个字符集分成 94 个区，每区有 94 个位。每个区位上只有一个字符，因此可用所在的区和位来对汉字进行编码，称为区位码。创建区位码这种编码形式，为制造中文打字机带来便利，早期的中文打字机采用了区位码的布局模式。在计算机内部中文并不是以区位码的形式存储，中文的编码必须与 ASCII 码兼容，特别是要与 ASCII 码中的控制符兼容，因此，区位码必须转换成机内码，才能在计算机中处理。区位码转换成机内码的方法是：先把区位码换算成十六进制，然后加上 2020H