



计算机安全与维护

Windows 系统数据安全与维护



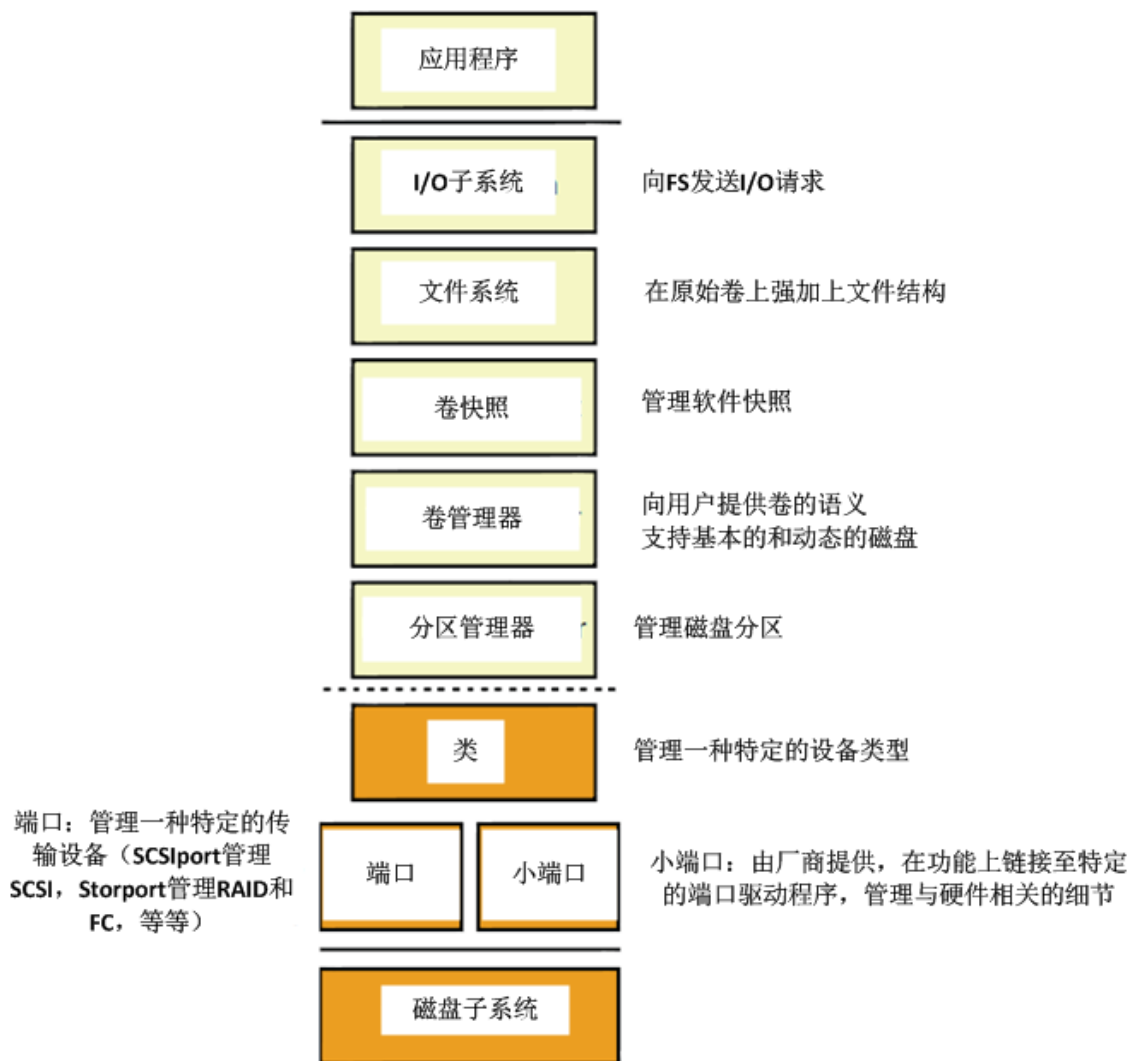
本章内容提要

- 磁盘驱动程序和缓存管理
- 文件系统接口和驱动程序
- 诊断文件系统问题
- NTFS的恢复支持
- 误删除数据的手工恢复
- 数据备份方案
- 加密文件系统（EFS）的安全性



磁盘驱动程序

- 文件系统
相关的存
储栈





磁盘驱动程序

- 磁盘类，端口和小端口驱动程序

- 初始化时，windows I/O管理器启动硬盘的存储驱动程序。Windows中的存储驱动程序符合“类/端口/小端口”结构
- 存储类驱动程序，实现了对于所有的存储设备都共有的功能
- 存储端口驱动程序，实现了对于特定的总线而言共有的功能（SCSI总线或者IDE系统）
- OEM厂商提供小端口驱动程序将windows与特定的实现连接起来



磁盘驱动程序

- 磁盘设备对象

- Windows 磁盘类驱动程序创建代表磁盘和分区的设备对象
- 代表磁盘的设备对象具有形如
\\Device\\HarddiskX\\DRX 的名称（名称中的 X 由表示特定磁盘的编号来代替）
- 磁盘类驱动程序使用 I/O 管理器的
IoReadPartitionTable 函数和
IoReadPartitionTableEx 函数来标示分区，并创建代表这些分区的设备对象



磁盘驱动程序

- 分区管理器

- 分区管理器

- (\windows\system32\Drivers\Partmgr.sys) 负责通知即插即用管理器当前有哪些分区，卷管理器驱动程序可以接收有关分区创建和删除的通知

- 分区管理器在系统引导时，读入所附载的磁盘分区表，并监视与分区表修改相关的I/O请求包。更新内部分区表，并将分区的创建和删除事件通知PnP管理器



磁盘驱动程序

- 卷的管理

- 基本磁盘是依赖于MBR风格或GPT分区方案的磁盘，动态磁盘实现了比基本磁盘更灵活的分区方案
- 动态磁盘是指windows中创建多分区卷（镜像卷，条带卷等）所必须的磁盘格式。动态磁盘使用LDM（逻辑磁盘管理器）分区方案



磁盘驱动程序

- 卷名字空间

- 挂载管理器

- 挂载管理器设备驱动程序 (mountmgr.sys) 为创建的动态和基本磁盘卷, CD-ROM和可移除设备分配驱动器字母。并存储于 HKLM\SYSTEM\MountedDevices 中

- 挂载点

- 挂载点使得可以通过NTFS卷上的目录, 将多个卷链接起来, 使得没有分配到驱动器字母的卷也能够访问

- 卷的挂载



缓存管理

- 缓存管理器是一组内核模式的函数和系统线程，与内存管理器协同工作，为所有的windows文件系统驱动程序提供数据缓存能力
- 缓存管理器的特性
 - 支持所有的文件系统类型
 - 内存管理器控制哪些文件的哪些部分位于物理内存
 - 快速I/O，以虚拟块为基础来缓存数据
 - 支持可恢复的文件系统
 - 允许应用程序打开文件时传递访问方式



缓存管理

- 内存管理器

- 缓存管理器将频繁被访问的数据记录在物理内存中，提高I/O性能
- 缓存管理器访问数据的做法是将文件的仕途映射到系统虚拟地址空间中，使用标准的内存区对象
- 内存管理器把不在物理内存中的数据块换到物理内存中，但内存吃紧时，又把缓存中的数据换出去



缓存管理

- 虚拟块缓存

- 虚拟块缓存跟踪，记录文件位于缓存中的部分。
缓存管理器监视这些文件部分
- 使得智能预读成为可能，预测调用者接下来可能读哪些部分
- 避免到文件系统中请求那些已经在缓存中的数据，
直接返回已被缓存的数据地址，无需调用文件系统



缓存管理

- 对可恢复文件系统的支持

- 一个文件系统将数据写到缓存中时，提供一个逻辑序列号 (LSN)，标示此次缓存更新的记录，缓存管理器跟踪这些编号
- 文件系统往回调用缓存管理器，指示刷新日志文件数据，直至LSN所代表的那一点，再把对应的卷结构更新数据刷新到磁盘上



缓存管理

- 缓存的数据结构

- 系统缓存中的每一个256KB的槽都是通过一个VACB（虚拟地址控制块）来描述的
- 每一个被单独打开的，被缓存的文件都有一个私有的缓存表，其中包含了用于控制预读的信息
- 每一个被缓存的文件都有一个共享的缓存表结构，指向系统缓存中包含有该文件映射视图的那些槽



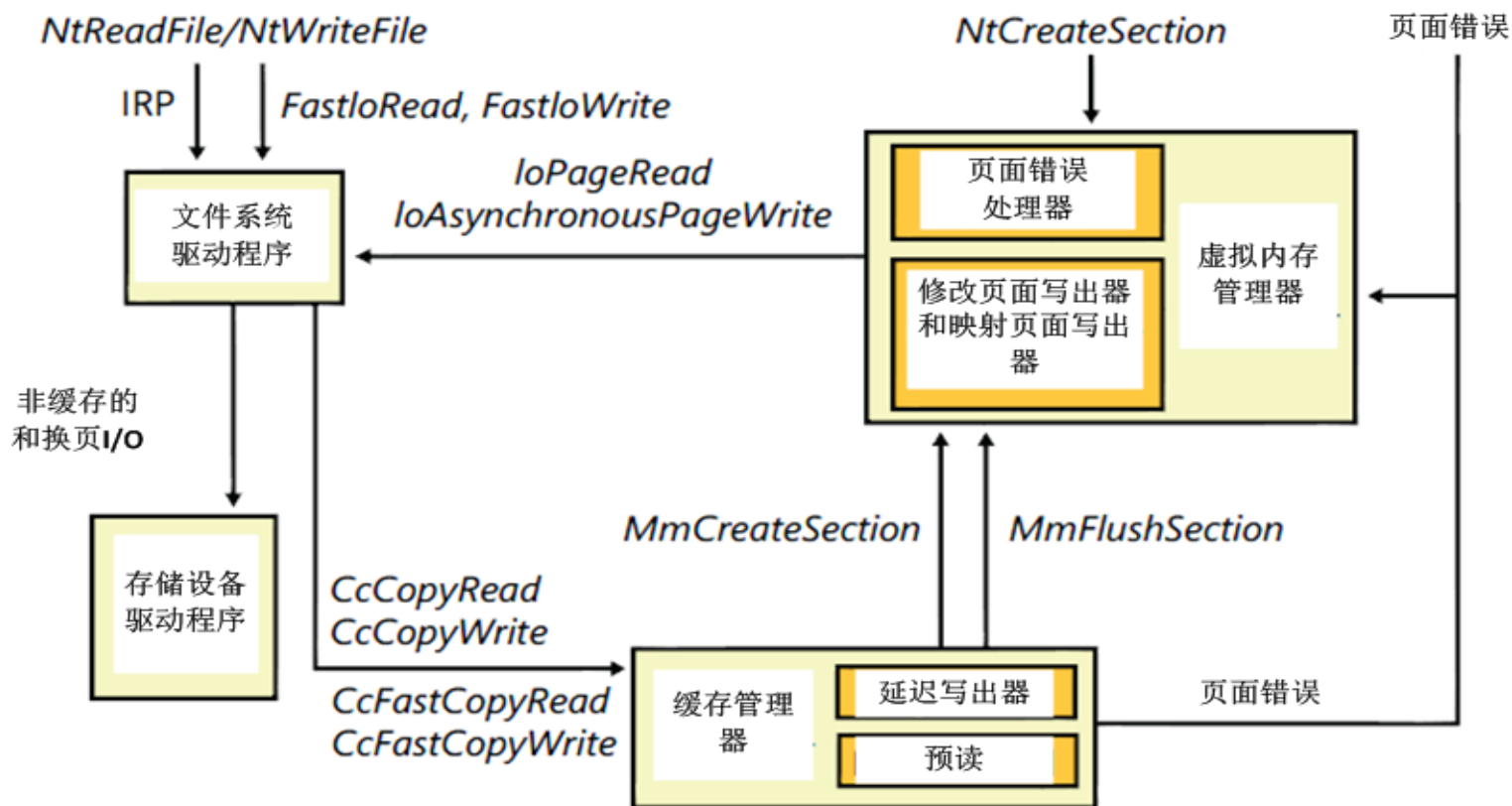
本章内容提要

- 磁盘驱动程序和缓存管理
- 文件系统接口和驱动程序
- 诊断文件系统问题
- NTFS的恢复支持
- 误删除数据的手工恢复
- 数据备份方案
- 加密文件系统（EFS）的安全性



文件系统接口

• 文件系统与缓存管理器和内存管理器的交互





文件系统接口

- 从缓存中拷贝数据

- 系统缓存位于系统空间中，被映射到每个进程的地址空间中
- 系统缓存中的页面无法从用户模式中访问，会导致潜在的安全漏洞
- 用户应用程序若要读写被缓存的文件，必须通过内核模式在“系统空间的缓存区”和“驻留在进程地址空间中的应用程序缓存区”之间拷贝数据



文件系统接口

- 通过映射和锁定接口进行缓存

—如果一个文件系统驱动程序需要读缓存中的文件系统元数据，调用缓存管理器的映射接口来获得目标数据的虚拟地址，缓存管理器寻找所有被请求的页面，并把他们带入内存中，控制权交给文件系统驱动程序

—如果文件系统驱动程序需要修改缓存中的页面，调用缓存管理器的锁定服务，该服务会将这些被修改的页面保持在内存中，直到通知它可以写出，解除锁定



文件系统接口

- 通过直接内存访问接口进行缓存
 - 直接内存访问 (DMA) ，通过DMA函数，可以直接读或者写系统缓存的页面，而无需经由缓存区的介入
 - DMA接口将被缓存的用户数据的物理地址返回给文件系统，这些物理地址可以直接被用于在物理内存和网络设备之间传输数据
 - 内存描述符列表 (MDL) 用于描述对于物理内存的引用



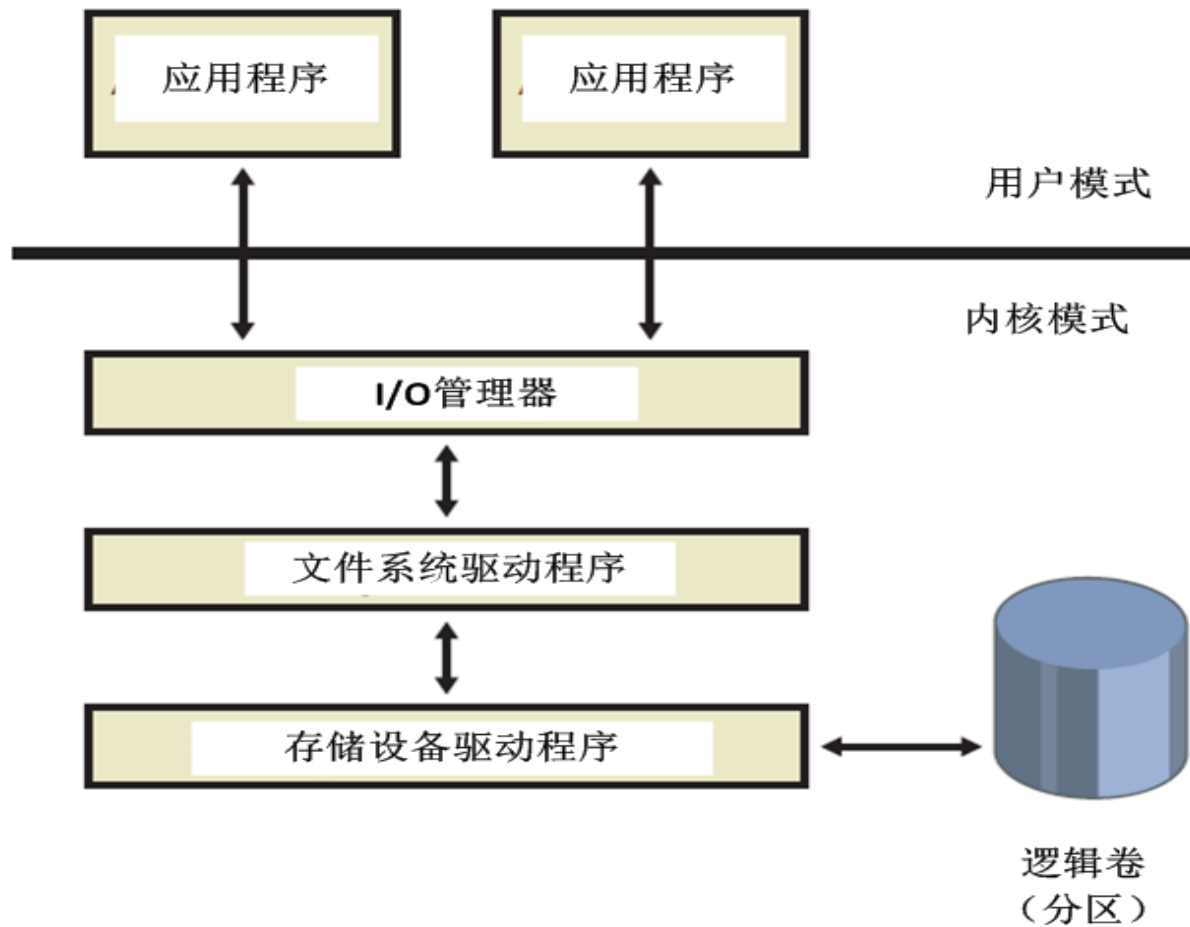
文件系统驱动程序

- 文件系统驱动程序（FSD）管理文件系统格式，虽然FSD运行在内核模式下，但与标准的内核驱动程序有所不同
 - 必须向I/O管理器注册
 - 大量的与内存管理器打交道
 - 依赖于缓存管理器的服务
 - Windows有两种不同类型的文件系统驱动程序
 - 本地FSD管理直接连接到计算机的卷
 - 网络FSD允许用户访问连接至远程计算机的数据卷



文件系统驱动程序

- 本地FSD





文件系统驱动程序

- 本地FSD包括Ntfs.sys,Udfs.sys,Cdfs.sys和Raw FSD（集成于Ntoskrnl.exe中）
- 本地向I/O管理器注册后，I/O管理器就可以在应用程序或系统最初访问卷的时候，调用它来执行卷的识别工作
- Windows支持的每一个文件系统格式第一个扇区被保留为该卷的引导扇区，本地FSD可以识别出该扇区的卷包含了FSD管理的格式，也可以定位到任何的元数据所在位置



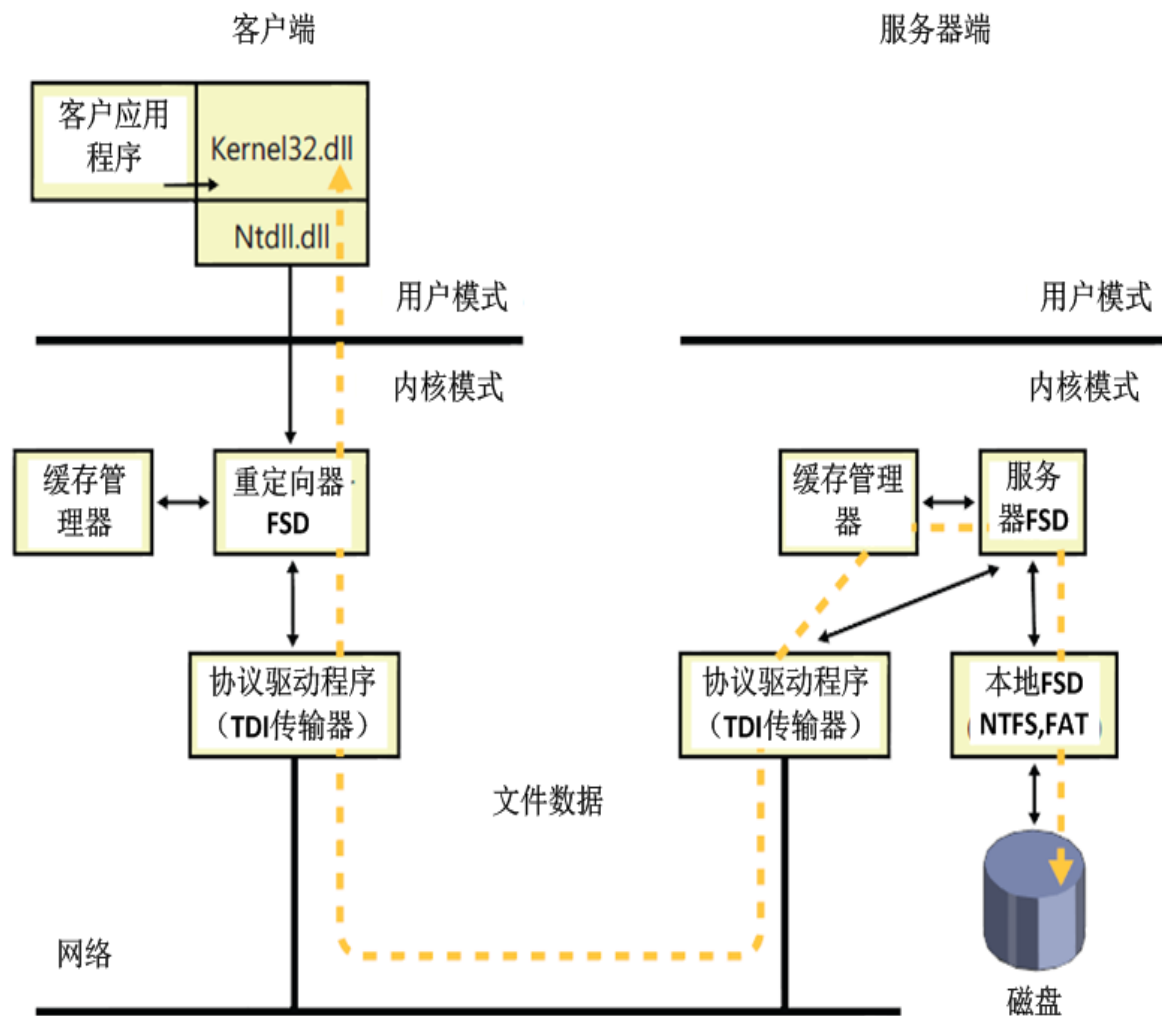
文件系统驱动程序

- 本地FSD识别出卷时，创建一个设备对象，I/O管理器通过卷参数块（VPB）在该卷的设备对象和FSD创建的设备对象之间建立一个连接
- 为了提高性能，本地FSD使用缓存管理器来缓存文件系统的数据
- 本地FSD支持文件系统卸载操作，使得系统可以断开FSD跟卷对象之间的连接当应用程序以原始方式访问一个卷的磁盘内容时，或者与一个卷相关联的介质发生变化，卸载发生



文件系统驱动程序

• 远程FSD





文件系统驱动程序

- 远程FSD由两个部件组成，客户端和服务端
- 客户端FSD接收来自应用程序的I/O请求，翻译成网络系统协议的命令通过网络发送给服务器的部件
- 服务器方的FSD监听来自网络连接的命令，通过向本地FSD发送I/O请求来实现命令，本地FSD负责管理该命令的目标文件或目录所在的卷



文件系统驱动程序

- Windows包含名为“LANMan Redirector”(重定向器)的客户方远程FSD和“LANMan Server”的服务器远程FSD，该重定向器被实现为“端口/小端口驱动程序”组合
- Windows依赖于公共Internet文件系统（CIFS）协议来格式化重定向器和服务器之间交换的消息
- 客户方远程FSD必须实现称为oplocks（机会锁）的分布式缓存一致协议，当一个应用程序访问远程文件时所看到的数据与本地一致



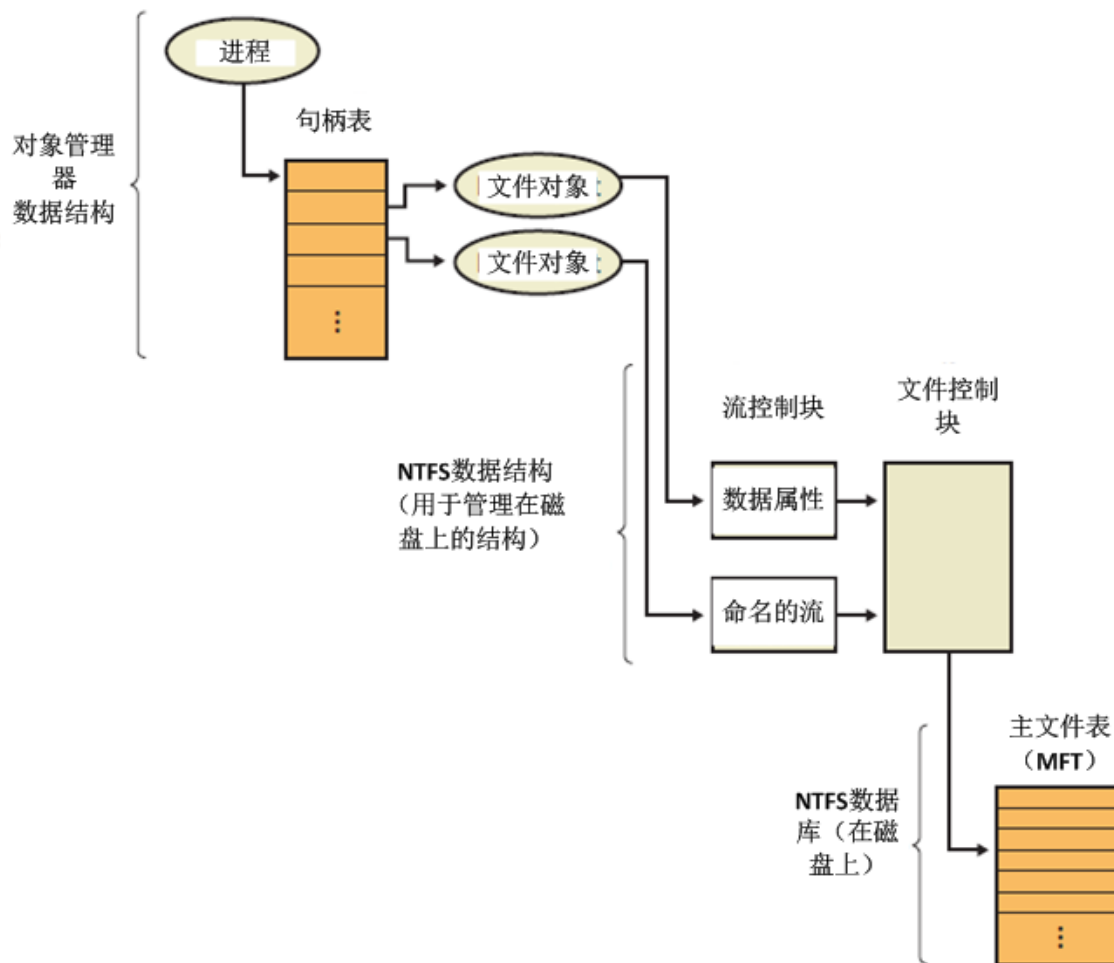
文件系统驱动程序

- 客户想要访问服务器文件时，必须首先请求oplock。指明它可以根据服务器授予的oplock类型来执行相应的缓存
- 三种主要的oplock类型：
 - Level 1 oplock（独占访问）
 - Level 2 oplock（共享）
 - Batch oplock（缓存文件的读写操作，打开关闭文件）



文件系统驱动程序

- 文件系统操作





文件系统驱动程序

- 应用程序和系统通过两种方式访问文件
 - 文件I/O函数直接进行
 - 读写地址空间中代表映射文件内存区的内存间接进行
- 各个组件之间的交互通过以下路径调用
 - 正在执行显式文件I/O的用户或系统线程
 - 内存管理器的修改页面写出器和映射页面写出器
 - 缓存管理器的延迟写出器和预读线程
 - 内存管理器的页面错误处理器



文件系统驱动程序

- 文件系统过滤型驱动程序

- 叠加在文件系统驱动程序之上，修改或者完成文件系统请求，完成文件加密，备份，许可控制，病毒扫描等应用
- Process Monitor工具（文件系统活动监视工具）
- 系统恢复（观察关键的系统文件变化，并做好备份）



文件系统驱动程序

- 输入指令fltmc, 查看加载的文件过滤型驱动程序

```
C:\Windows\system32>fltmc
```

筛选器名称	数字实例	高度	框架
-----	-----	-----	-----
PROCMON23	7	385200	1
qutmdserv			<过时>
360Box	7	382310	0
avgntflt	5	320500	0
luaflv	1	135000	0
FileInfo	7	45000	0



本章内容提要

- 磁盘驱动程序和缓存管理
- 文件系统接口和驱动程序
- 诊断文件系统问题
- NTFS的恢复支持
- 误删除数据的手工恢复
- 数据备份方案
- 加密文件系统（EFS）的安全性



诊断文件系统问题

• Process Monitor 工具

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time	Process Name	PID	Operation	Path	Result	Detail
14:3...	svchost.exe	1148	IRP_MJ_QUERY_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	BUFFER OVERFLOW	Type: QueryAl...
14:3...	svchost.exe	1148	IRP_MJ_QUERY_VOLUME_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Type: QueryIn...
14:3...	svchost.exe	1148	IRP_MJ_QUERY_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	BUFFER OVERFLOW	Type: QueryAl...
14:3...	svchost.exe	1148	IRP_MJ_CLEANUP	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	svchost.exe	1148	IRP_MJ_CLOSE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	svchost.exe	1148	IRP_MJ_CREATE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Desired Acces...
14:3...	svchost.exe	1148	IRP_MJ_QUERY_VOLUME_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Type: QueryIn...
14:3...	svchost.exe	1148	IRP_MJ_QUERY_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	BUFFER OVERFLOW	Type: QueryAl...
14:3...	svchost.exe	1148	IRP_MJ_CLEANUP	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	svchost.exe	1148	IRP_MJ_CLOSE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	svchost.exe	1148	IRP_MJ_FILE_SYSTEM_CONTROL	C:	SUCCESS	Control: FSCT...
14:3...	svchost.exe	1148	IRP_MJ_CREATE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Desired Acces...
14:3...	svchost.exe	1148	IRP_MJ_FILE_SYSTEM_CONTROL	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Control: FSCT...
14:3...	svchost.exe	1148	IRP_MJ_CLEANUP	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	svchost.exe	1148	IRP_MJ_CLOSE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	svchost.exe	1148	IRP_MJ_CREATE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Desired Acces...
14:3...	svchost.exe	1148	FASTIO_QUERY_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Type: QuerySt...
14:3...	svchost.exe	1148	IRP_MJ_READ	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Offset: 0, Le...
14:3...	svchost.exe	1148	FASTIO_QUERY_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Type: QuerySt...
14:3...	svchost.exe	1148	FASTIO_READ	C:\Users\santiago\Desktop\Sysinte...	FAST IO DISALLOWED	Offset: 2,470...
14:3...	svchost.exe	1148	IRP_MJ_READ	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Offset: 2,470...
14:3...	svchost.exe	1148	IRP_MJ_CLEANUP	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	Procmon.exe	17316	FASTIO_NETWORK_QUERY_OPEN	C:\Users\santiago\Desktop\Sysinte...	FAST IO DISALLOWED	
14:3...	Procmon.exe	17316	IRP_MJ_CREATE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Desired Acces...
14:3...	Procmon.exe	17316	FASTIO_QUERY_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Type: QueryBa...
14:3...	Procmon.exe	17316	IRP_MJ_CLEANUP	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	Procmon.exe	17316	IRP_MJ_CLOSE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	Procmon.exe	17316	FASTIO_NETWORK_QUERY_OPEN	C:\Users\santiago\Desktop\Sysinte...	FAST IO DISALLOWED	
14:3...	Procmon.exe	17316	IRP_MJ_CREATE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Desired Acces...
14:3...	Procmon.exe	17316	FASTIO_QUERY_INFORMATION	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	Type: QueryBa...
14:3...	Procmon.exe	17316	IRP_MJ_CLEANUP	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	
14:3...	Procmon.exe	17316	IRP_MJ_CLOSE	C:\Users\santiago\Desktop\Sysinte...	SUCCESS	

Showing 426,927 of 1,628,241 events (26%) Backed by virtual memory



诊断文件系统问题

- Process Monitor工作方式

- 运行时，从可执行映像中提取文件过滤型设备驱动程序，安装到内存
- 通过GUI，监视分配了驱动器字母的本地卷，网络共享体，命名管道，邮件槽上的文件系统活动
- 接收到命令启动对一个卷的监视时，创建一个过滤型设备驱动程序，附载到设备对象



诊断文件系统问题

- 当运行Process Monitor时，在基本模式下启动，此模式下显示最为有用的文件系统活动，忽略了某些特定的文件系统操作
 - 对NTFS元数据文件的访问
 - 发生在system进程中的活动
 - 页面文件的I/O
 - 由Process Monitor进程生成的I/O



- | Time | Process Name | PID | Operation | Path | Result | Detail |
|---------|--------------|-------|---------------------------|--------------------------------------|--------------------|-----------------|
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Windows\System32\dllhost.exe | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Windows\System32\dllhost.exe | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Windows\System32\dllhost.exe | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Windows\System32\dllhost.exe | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Windows\System32\dllhost.exe | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Windows\System32\dllhost.exe | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Users\santiago\Desktop\Sysinte... | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Users\santiago\Desktop\Sysinte... | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Users\santiago\Desktop\Sysinte... | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Users\santiago\Desktop\Sysinte... | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | IRP_MJ_QUERY_INFORMATION | C:\Users\santiago\Desktop\Sysinte... | BUFFER OVERFLOW | ype: QueryAl... |
| 14:3... | svchost.exe | 1148 | FASTIO_READ | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | ffset: 2,470... |
| 14:3... | Procmon.exe | 17316 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | Procmon.exe | 17316 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | Procmon.exe | 17316 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | Procmon.exe | 17316 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | Procmon.exe | 17316 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | nvtray.exe | 1632 | FASTIO_NETWORK_QUERY_OPEN | C:\Program Files\NVIDIA Corporat... | FAST IO DISALLOWED | |
| 14:3... | nvtray.exe | 1632 | IRP_MJ_CREATE | C:\Program Files\NVIDIA Corporat... | NAME NOT FOUND | esired Acces... |
| 14:3... | svchost.exe | 1148 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | svchost.exe | 1148 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | svchost.exe | 1148 | IRP_MJ_CREATE | C:\Users\santiago\Desktop\Sysinte... | NAME NOT FOUND | esired Acces... |
| 14:3... | svchost.exe | 1148 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | svchost.exe | 1148 | IRP_MJ_CREATE | C:\Users\santiago\Desktop\Sysinte... | NAME NOT FOUND | esired Acces... |
| 14:3... | svchost.exe | 1148 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | svchost.exe | 1148 | IRP_MJ_CREATE | C:\Users\santiago\Desktop\Sysinte... | NAME NOT FOUND | esired Acces... |
| 14:3... | svchost.exe | 1148 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | svchost.exe | 1148 | IRP_MJ_CREATE | C:\Users\santiago\Desktop\Sysinte... | NAME NOT FOUND | esired Acces... |
| 14:3... | svchost.exe | 1148 | FASTIO_NETWORK_QUERY_OPEN | C:\Users\santiago\Desktop\Sysinte... | FAST IO DISALLOWED | |
| 14:3... | svchost.exe | 1148 | IRP_MJ_CREATE | C:\Users\santiago\Desktop\Sysinte... | NAME NOT FOUND | esired Acces... |



本章内容提要

- 磁盘驱动程序和缓存管理
- 文件系统接口和驱动程序
- 诊断文件系统问题
- NTFS的恢复支持
- 误删除数据的手工恢复
- 数据备份方案
- 加密文件系统（EFS）的安全性



NTFS的恢复支持

- NTFS的恢复支持可确保，如果发生断电或者系统失败，文件操作不会遗留在未完成的状态，磁盘卷的结构仍然完好无损，无需运行磁盘修复工具
- NTFS使用一种事务处理方案实现可恢复性，确保即使对于非常大的磁盘，磁盘的恢复也会绝对快速，恢复过程仅限于文件系统数据



NTFS的恢复支持

- 文件系统设计的演变

- 谨慎写 (careful write) 文件系统

- 对写操作进行排序
 - 即使系统失败，整个卷仍然处于一致和可用的状态

- 延迟写 (lazy write) 文件系统

- 把文件的修改写到缓存中，再刷新到磁盘
 - 使性能提高，但是风险更高



NTFS的恢复支持

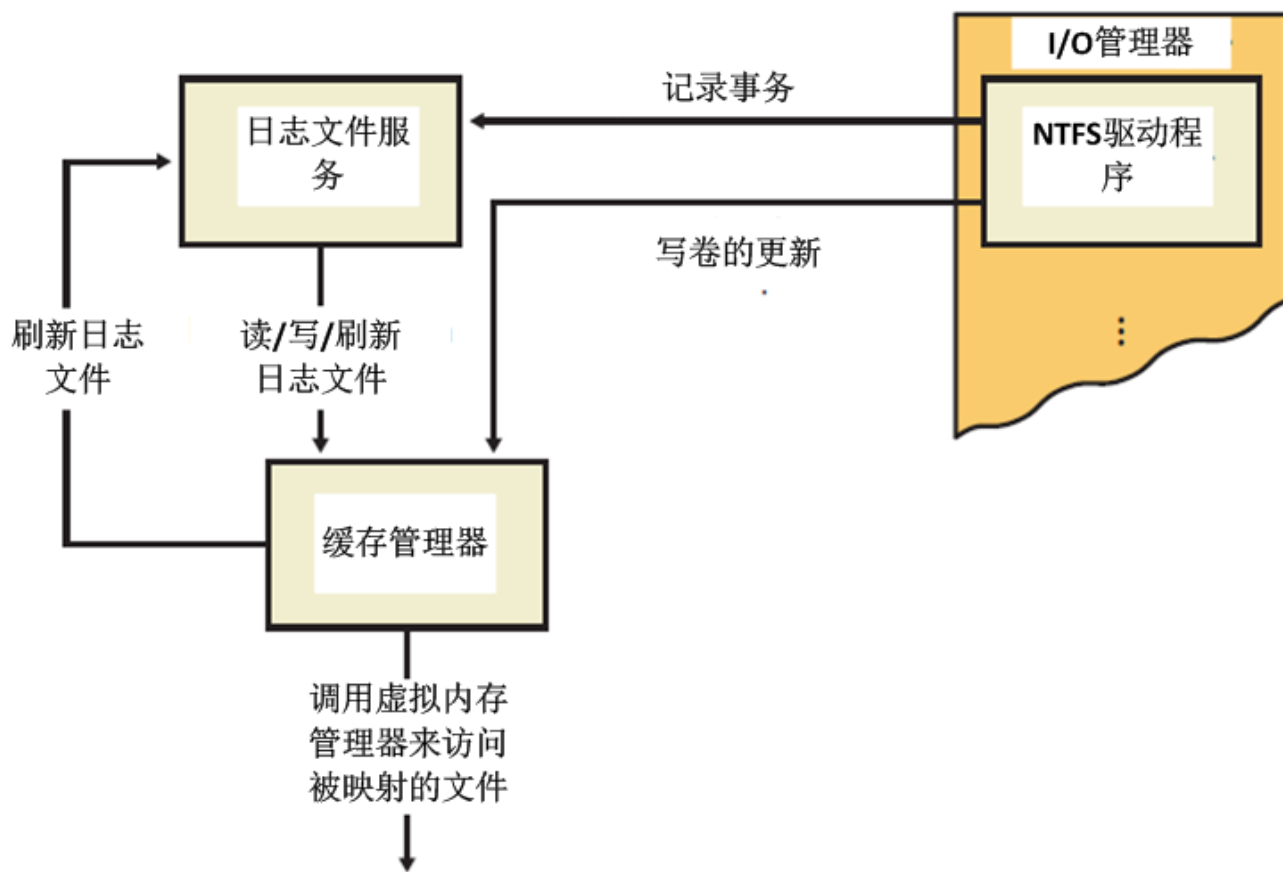
- 可恢复的文件系统

- 超过谨慎写文件系统的保险性，获得延迟写文件系统的性能优势
- 使用最初为事务处理发展起来的日志技术确保卷的一致性
- NTFS可恢复性确保其卷结构不会被破坏
- 在一次直写操作或者缓存刷新以后，用户数据将是一致的，并且立即可以使用



NTFS的恢复支持

- 日志记录





NTFS的恢复支持

- 日志文件服务 (LFS)

- 日志记录类型

 - 更新记录

 - 每一条记录包含重做信息和撤销信息
 - 创建，删除，扩充，截短，设置文件信息，重命名，改变文件的安全性

 - 检查点记录

 - 周期性的写入检查点记录
 - 用于回滚系统设置信息



NTFS的恢复支持

- 恢复

- 恢复过程依赖NTFS在内存中维护的两张表

- 事务表（记录启动但是未提交的事务）

- 脏页表（记录缓存中页面的未被写到磁盘上的修改信息）

- NTFS扫描日志文件三遍

- 分析扫描

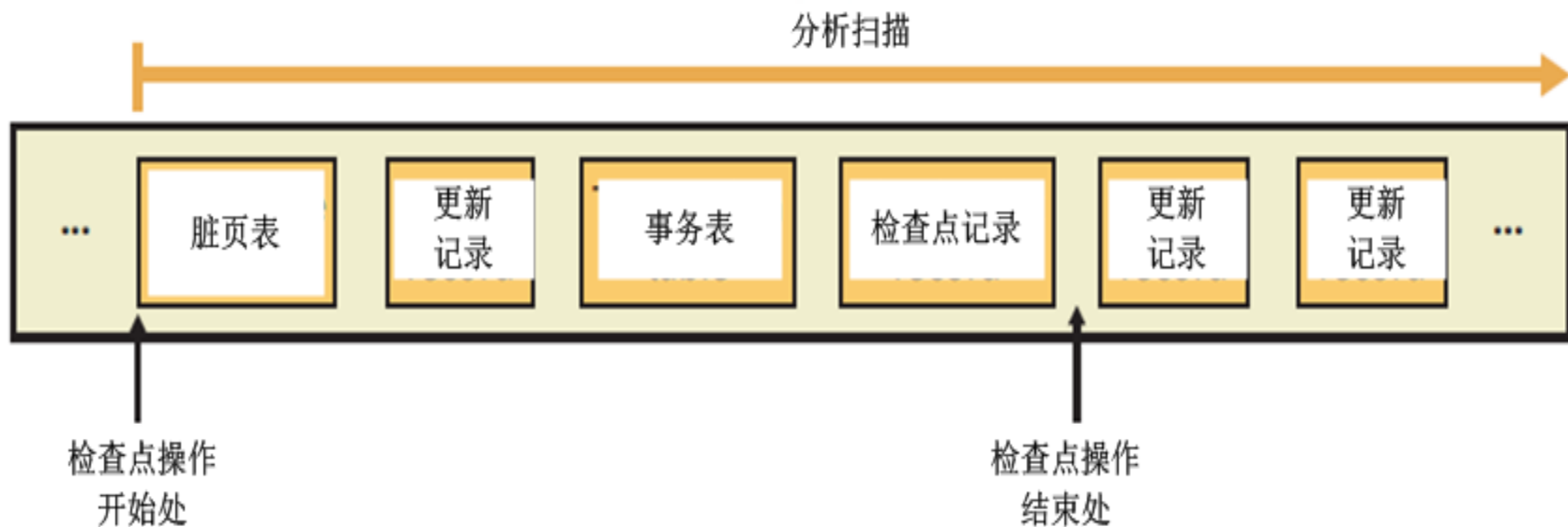
- 重做扫描

- 撤销扫描



NTFS的恢复支持

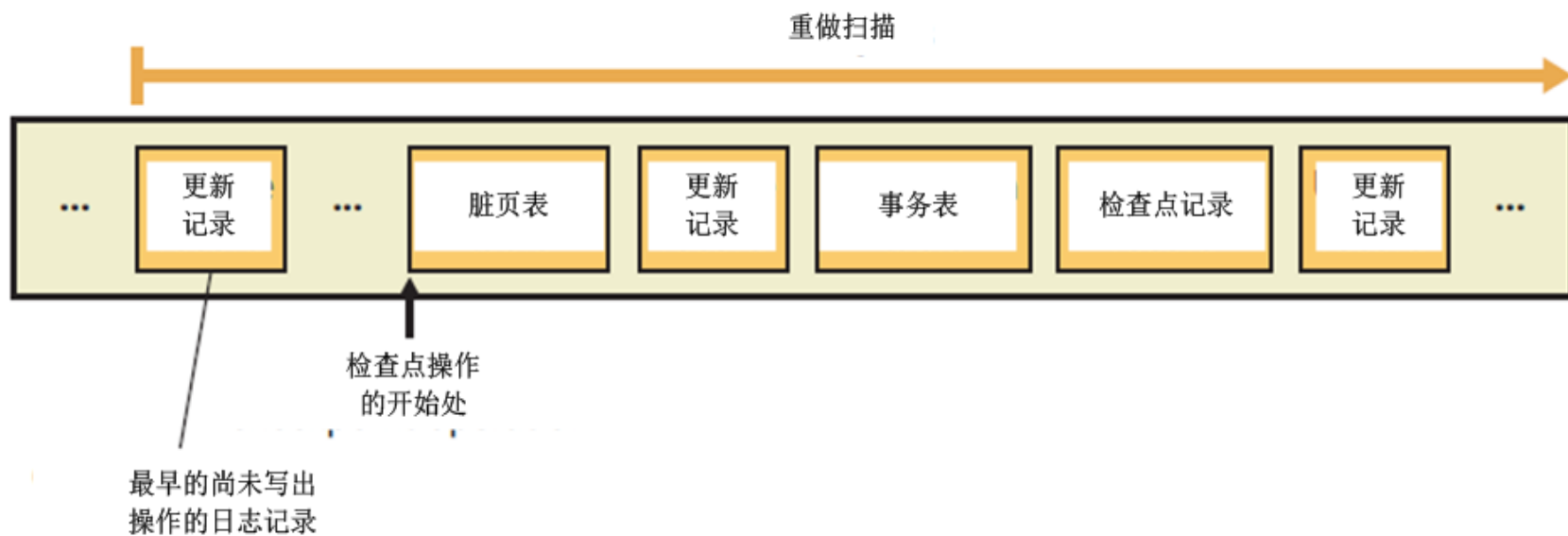
- 分析扫描





NTFS的恢复支持

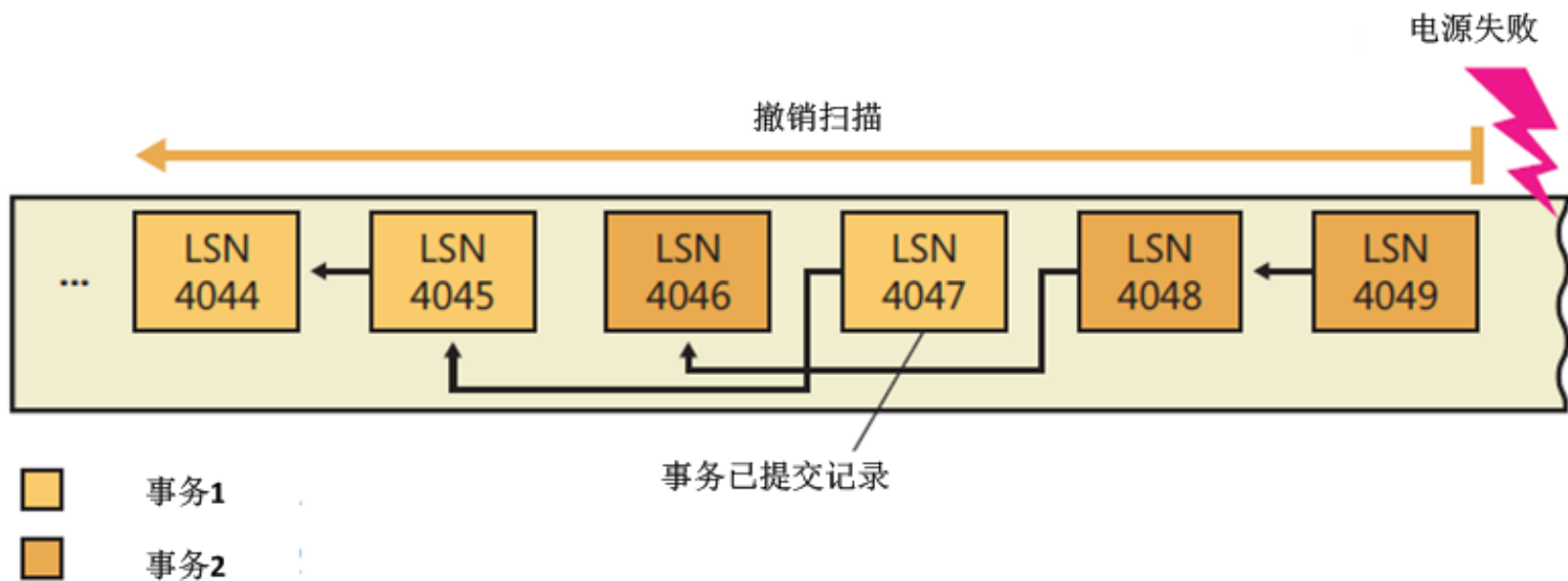
- 重做扫描





NTFS的恢复支持

- 撤销扫描





NTFS的恢复支持

- NTFS的坏簇恢复

- NTFS动态的替换掉包含坏扇区的簇，跟踪记录这一坏簇，以后不会重用

- 容错卷

- 恢复数据

- 替换掉坏扇区

- 非容错卷

- 不能恢复数据

- NTFS执行簇重映射，数据丢失



NTFS的恢复支持

- 自我恢复

- SET_REPAIR_ENABLED

- 开启卷的自我恢复功能

- SET_REPAIR_WARN_ABOUT_DATA_LOSS

- 如果文件不能完全恢复，是否通知用户

- SET_REPAIR_DISABLED_AND_BUGCHECK_ON_CORRUPTION

- 系统崩溃抛出0x24错误



本章内容提要

- 磁盘驱动程序和缓存管理
- 文件系统接口和驱动程序
- 诊断文件系统问题
- NTFS的恢复支持
- 误删除数据的手工恢复
- 数据备份方案
- 加密文件系统（EFS）的安全性



误删除数据的手工恢复

- 数据手工恢复的两种形式

- 硬恢复

- 硬盘出现物理性损伤，导致普通用户不能取出里面的数据，通过修理硬件的同时保留和恢复里面的数据

- 软恢复

- 硬盘本身没有物理损伤，由于人为或者病毒破坏造成数据丢失，通过软件进行数据恢复

- 本节介绍通过winHex软件进行数据软恢复的方法



误删除数据的手工恢复

- WinHex介绍

- 十六进制编辑软件

- 完善的分区管理功能和文件管理功能

- 自动分析分区链和文件簇链，对硬盘进行不同方式不同程度的备份，甚至克隆整个硬盘

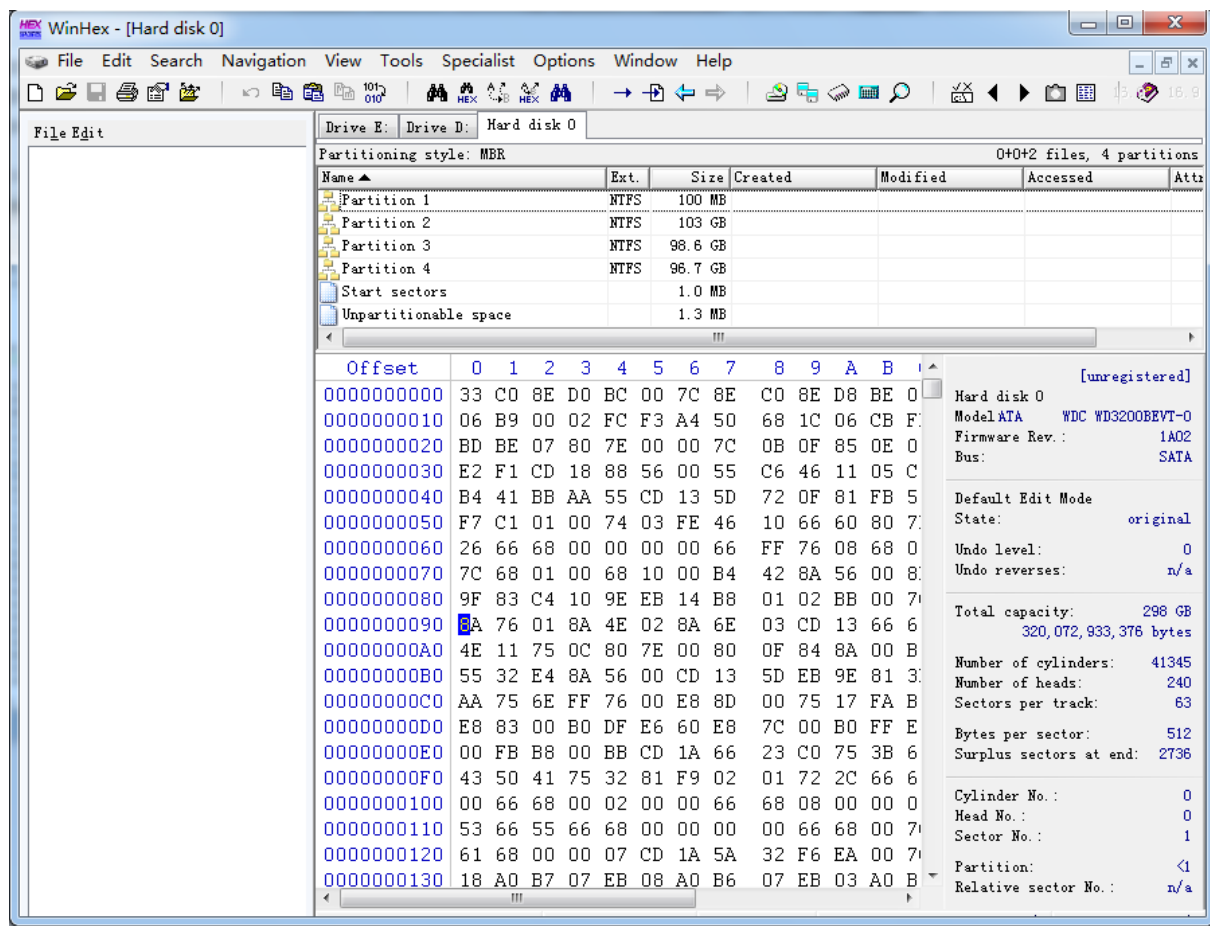
- 编辑任何一种文件类型的二进制内容

- 磁盘编辑器可以编辑物理磁盘或逻辑磁盘的任意扇区



误删除数据的手工恢复

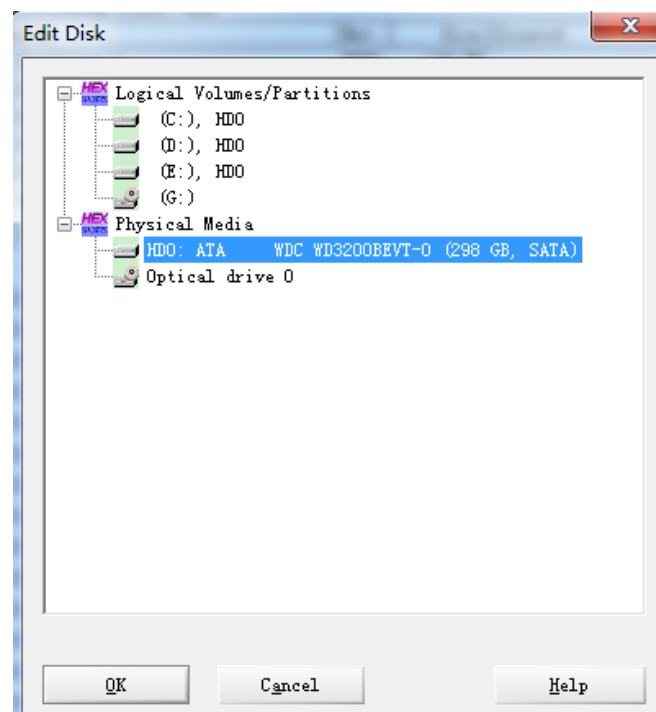
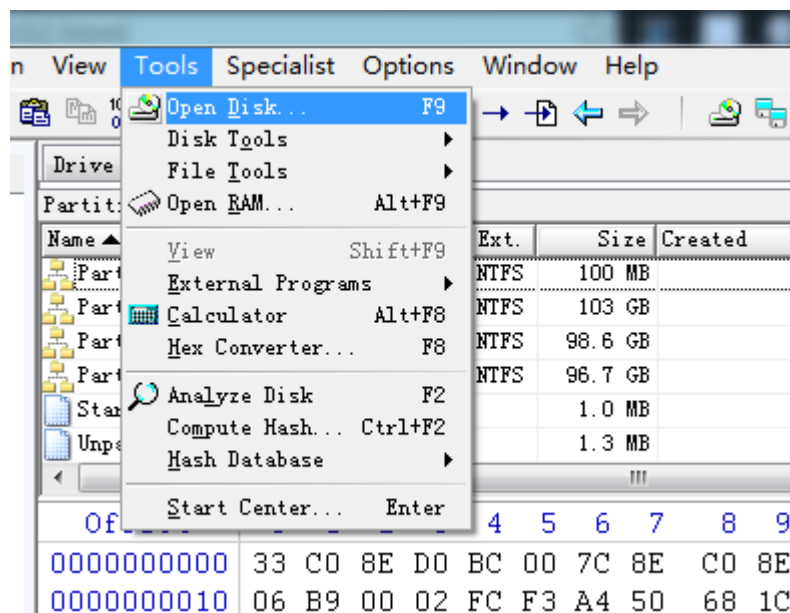
- WinHex主界面





误删除数据的手工恢复

- 可以选择对整个硬盘或者单独的分区进行恢复





误删除数据的手工恢复

• 界面的各部分信息类型

Drive E: Drive D: Hard disk 0 Start sectors

Partitioning style: MBR

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
Partition 1	NTFS	100 MB					2,048
Partition 2	NTFS	103 GB					206,848
Partition 3	NTFS	98.6 GB					215,5...
Partition 4	NTFS	96.7 GB					422,3...
Start sectors		1.0 MB					0
Unpartitionable space		1.3 MB					625,1...

硬盘分区情况

0+0+2 files, 4 partitions

Offset: 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13

0000100924 83 C6 04 66 8B 04 A3 16 00 83 C6 04 1E 07 E8 E8 F7 66 2B F8 1Æ f l E 1Æ èè÷f÷ø

0000100938 0F 84 08 00 F7 26 0B 00 03 D8 EB D9 66 8B 3E 6A 02 1E 07 E8 I -& 0èÜfI>j è

000010094C BF FD 66 A1 6A 02 66 BB 80 00 00 00 66 B9 00 00 00 00 66 8B çyfij f»I f¹ fI

0000100960 D1 E8 81 FB 66 0B C0 0F 84 FF F7 66 8B D8 66 58 66 56 E8 2C Nè úf À l y÷fI0fXfVè,

0000100974 01 66 5E 66 0B C0 0F 84 05 00 66 5B 66 5B C3 66 59 66 5A E2 f^f À l f[f[ÄfYfZä

0000100988 84 66 33 C0 C3 06 1E 66 60 66 50 66 51 66 33 D2 66 0F B6 1E If3ÄÄ f¹ fPfqf30f ¶

000010099C 0D 00 66 F7 F3 66 52 66 57 E8 53 FF 66 5F 66 0B C0 0F 84 B9 f÷ófrfWèSyf_f Ä l¹

00001009B0 F7 66 0F B6 1E 0D 00 66 F7 E3 66 5A 66 03 C2 66 A3 11 00 66 ÷f ¶ f÷äfZf Äfè f

00001009C4 59 66 0F B6 1E 0D 00 66 3B CB 0F 8E 13 00 89 1E 16 00 66 2B Yf ¶ f;È l l f+

00001009D8 CB 66 58 06 03 C3 66 50 66 51 EB 14 90 66 58 66 03 C1 66 50 ÈFXÈ ÄfPfQè fXf ÄfP

00001009E0 00 0E 1C 00 66 00 00 00 00 00 66 51 EB 14 90 66 58 66 03 C1 66 50 ! f¹ SQ SW101%

0000100A00 0C C8 88 C1 E1 84 83 C7 38 87 E8 18 F7 88 3F 87 88 83 3E 32 ÄÄÄÄ Çf è f¹ f¹ K

0000100A14 02 66 59 66 58 66 83 F9 00 0F 8F 70 FF 66 61 90 1F 07 C3 06 fYfXfIù pyfa Ä

0000100A28 1E 66 60 66 F7 26 56 02 66 8B 0E 56 02 E8 55 FF E8 D2 FC 66 f¹f÷&V fI V èUyèöuf

0000100A3C 61 90 1F 07 C3 06 1E 66 60 66 F7 26 72 02 66 8B 1E 36 02 66 a Ä f¹f÷&r fI 6 f

0000100A50 8B 0E 72 02 66 8B 36 2A 02 1E 07 66 8B 3E 46 02 E8 81 FB E8 l r fI6* fI>F è ùè

0000100A64 A7 FC 66 61 90 1F 07 C3 66 50 66 53 66 51 66 8B 1E 4A 02 66 Süfa ÄfPfSfQfI J f

0000100A78 8B C8 66 C1 E8 03 66 83 E1 07 66 03 D8 66 B8 01 00 00 00 66 lÈfÄè fIä f 0f, f

0000100A8C D3 E0 67 84 03 0F 84 04 00 F8 EB 02 90 F9 66 59 66 5B 66 58 ÓägI l øè ùfYf[fX

0000100AA0 C3 67 80 7B 08 01 0F 84 04 00 66 2B C0 C3 67 66 8D 73 10 67 ÄgI{ l f+ÄÄgf s g

0000100AB4 66 8B 56 08 66 3B C2 0F 87 0B 00 67 66 8B 16 66 3B C2 0F 83 fIV f;Ä l gfl f;Ä l

0000100AC8 04 00 66 2B C0 C3 67 03 5E 10 66 2B F6 67 80 3B 00 0F 84 3E f+ÄÄg ^ f+ögI; l>

0000100ADC 00 E8 81 00 66 03 F1 E8 39 00 66 03 CA 66 3B C1 0F 8C 21 00 è f nè9 f Èf;Ä l l

0000100AF0 66 8B D1 66 50 67 66 0F B6 0B 66 8B C1 66 83 E0 0F 66 C1 E9 fIñfPg fI ÄfIä fÄè

Hard disk 0 [unregistered]

Model: ATA WDC WD3200BEVT-0

Firmware Rev.: 1A02

Bus: SATA

Default Edit Mode

State: original

Undo level: 0

Undo reverses: n/a

Total capacity: 298 GB

320,072,933,376 bytes

Number of cylinders: 41345

Number of heads: 240

Sectors per track: 63

Bytes per sector: 512

Surplus sectors at end: 2736

Cylinder No.: 0

Head No.: 32

Sector No.: 37

Partition: 1

Relative sector No.: 4

Mode: hexadecimal

Character set: CP 936

Offsets: hexadecimal

Bytes per page: 24x20=480

Sector 2052 of 625142448

Offset: 1009D8

= 102 Block: n/a Size: n/a



误删除数据的手工恢复

- 通过WinHex查看硬盘的MBR

- MBR 是位于：0 扇区（逻辑扇区），大小为 512 bytes

- 在 MBR 里的后 64 个字节里是磁盘的分区表结构，可定义 4 个分区，每个分区 16 bytes，从 0x1be ~ 0x1fe 共 64 bytes

位置 (hex)	大小 (bytes)	描述
000 - 162	354 bytes	硬盘 MBR 引导记录 (代码区)
162 - 1BD	92 bytes	MBR 数据区域
1BE - 1CD	16 bytes	分区表 1
1CE - 1DD	16 bytes	分区表 2
1DE - 1ED	16 bytes	分区表 3
1EE - 1FD	16 bytes	分区表 4
1FE - 1FF	2 bytes	MBR 标志 (55AA)



误删除数据的手工恢复

• 磁盘分区表结构

位置 (hex)	大小 (bytes)	意义	描述	
1BE	1	分区的启动标志	80 =	可启动分区
			00 =	不可启动区
1BF - 1C1	3	分区的起始扇区	1BF =	heads, 起始 heads (1 个 bytes)
			1C0 =	sector, 低 6 bits 表示起始 sector, 这里只用该节字的低 6 bits 来表示 sector
			1C1 =	cylinder, 1C0 的高 2 btis 加上 1C1 的 8 bits 组成 10 bits 表示起始 cylinder
1C2	1	文件系统	如: 07 表示 ntfs 系统, 详见: 文件系统	
1C3 - 1C5	3	分区的结束扇区	其意义和起始扇区一致	
1C6 - 1C9	4	此分区前扇区数	这 4 bytes 表示此分区前有多少扇区 (实际上等于此分区的起始扇区号), 以 little-endian 排列的。	
1CA - 1CD	4	此分区扇区数	这 4 bytes 用来表示此分区共有多少扇区, 同样是以 little-endian 排列的。	



误删除数据的手工恢复

- 以第一分区表为例

00000001A4	65 72 61 74 69 6E 67 20 73 79 73 74 65 6D 00 00 00 63 7B 9A	erating system c{
00000001B8	02 1C B5 DF 00 00 80 20 21 00 07 A3 13 0D 00 08 00 00 00 20	µß ! £
00000001CC	03 00 00 A3 14 0D 07 EF FF FF 00 28 03 00 00 C0 D5 0C 00 EF	£ iyy (ÅÕ i
00000001E0	FF FF 07 EF FF FF 00 E8 D8 0C 00 38 54 0C 00 EF FF FF 07 EF	yy iyy è0 8T iyy i
00000001F4	FF FF 00 20 2D 19 00 C0 15 0C 55 AA 00 00 00 00 00 00 00 00	yy - Å Ua
0000000208	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
000000021C	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	



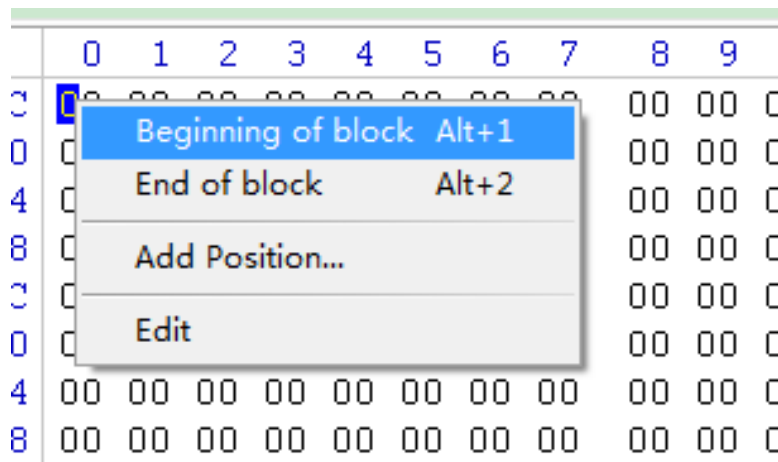
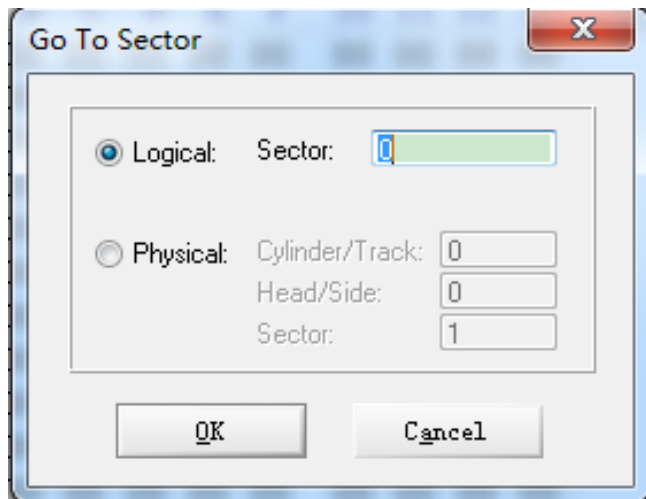
误删除数据的手工恢复

000001BE	80	可启动分区
000001BF	20	起始 header 号
000001C0	21	起始 sector 号
000001C1	00	起始 cylinder 号
000001C2	07	NTFS格式
000001C3	DF	结束 header 号
000001C4	13	结束 sector 号
000001C5	0C	结束 cylinder 号
000001C6	00080000	此分区前的扇区总数
000001CA	00200300	此分区的扇区总数



误删除数据的手工恢复

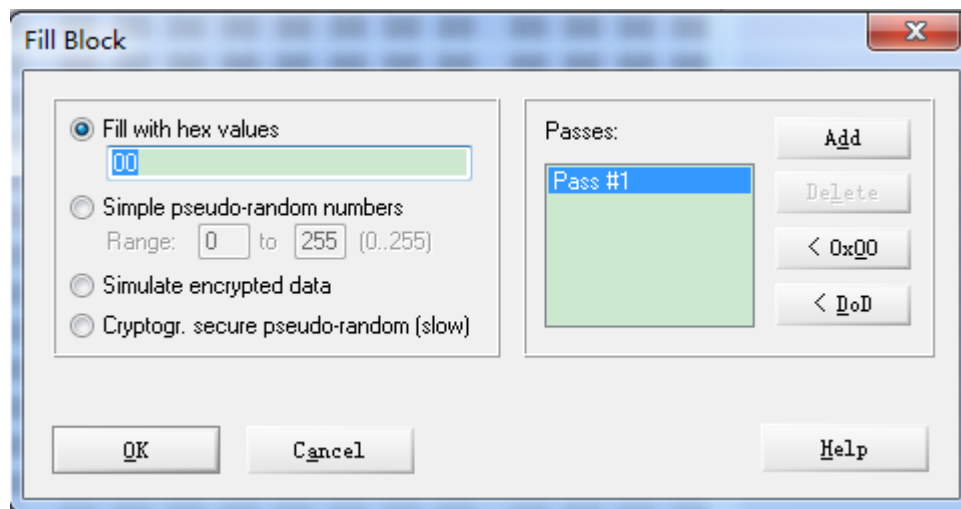
- 通过计算就可以转到对应的扇区进行处理
- 在需要编辑的位置右键开始对选块进行选择





误删除数据的手工恢复

- 使用填充数据或者复制剪贴板的数据来对选定的数据块进行修改





本章内容提要

- 磁盘驱动程序和缓存管理
- 文件系统接口和驱动程序
- 诊断文件系统问题
- NTFS的恢复支持
- 误删除数据的手工恢复
- 数据备份方案
- 加密文件系统（EFS）的安全性



数据备份方案

- 数据备份对于个人和企业用户都是至关重要的，数据本身的脆弱性或者丢失或者损坏会直接的威胁到用户的利益
- 目前威胁数据安全的因素
 - 系统的漏洞
 - 系统的硬件故障
 - 人为的操作失误
 - 供电系统故障
 - 网络的非法访问



数据备份方案

- 正常备份模式

- 优点是可以自动筛选备份文件
- 缺点是效率不高，费时
- 对需要备份的文件在文件属性中标记为存档，当执行备份操作时，对标记过的文件进行备份操作，备份之后自动取消“存档”属性
- 若文件没有被改动过，则在备份时会自动跳过



数据备份方案

- 副本备份模式

- 优点是备份快速
- 缺点是备份的方式是非智能的
- 只是简单的将备份的目标文件复制下来，作为副本添加到备份文件中
- 采用副本备份模式执行备份操作后，目标文件的“存档”属性不受影响



数据备份方案

- 增量备份模式

- 优点是有针对性，速度快

- 缺点是备份的数据份数较多

- 对于需要反复修改的文件比如设计图或者文档等，增量备份是最佳的备份方式

- 对发生变化的文件根据修改的顺序依次进行备份



数据备份方案

- 差异备份模式

- 优点是恢复速度快

- 缺点是占用空间

- 差异备份和增量备份在第一次使用时都需要配合完整的普通备份，针对新建或修改的文件

- 若文件自上次完整备份后曾被更新过，接下来每次做差异备份时，都会被备份，直到下次完整备份

- 差异备份的大小会随时间不断增加



数据备份方案

- 每日备份模式

- 优点是无需干预，自动备份

- 缺点是占用空间

- 每日备份模式省去了手动备份重要文件的操作，添加计划任务就可以

- 每日备份模式的目标是当天创建或修改的文件



数据备份方案

- Windows自带的备份工具
- 在系统的“控制面板”中“备份和还原”选项





数据备份方案

- 备份和还原的主界面，点击“更改设置”进行备份的设置

备份或还原文件

备份

位置:

(D:)



78.83 GB 可用，共 98.63 GB

备份大小: 80.49 MB

[管理空间\(M\)](#)

[立即备份\(B\)](#)

下一次备份:

2013/2/24 19:00

上一次备份:

2013/2/19 16:34

内容:

库中的文件和所选用户的个人文件夹

计划:

每 星期日 的 19:00

[更改设置\(C\)](#)

还原

可以还原在当前位置备份的文件。

[还原我的文件\(R\)](#)

[还原所有用户的文件\(A\)](#)

[选择要从中还原文件的其他备份\(N\)](#)

[恢复系统设置或计算机\(Y\)](#)





数据备份方案

- 选择备份数据的位置，可以选择上传到网络服务器
- 下一步选择需要备份的目标文件

选择要保存备份的位置

建议将备份保存到外部硬盘上。 [备份目标选择指南](#)

保存备份的位置(B):

备份目标	可用空间	总大小
 本地磁盘 (D:)	78.83 GB	98.63 GB
 本地磁盘 (E:)	21.41 GB	96.68 GB

刷新(R)

保存在网络上(V)...

您希望备份哪些内容?

☒ 让 Windows 选择(推荐)

Windows 将备份保存在库、桌面和默认 Windows 文件夹中的数据文件。将定期备份这些项目。

[Windows 如何选择要备份的文件?](#)

☐ 让我选择

可以选择库和文件夹，以及是否在备份中包含系统映像。将定期备份所选项目。

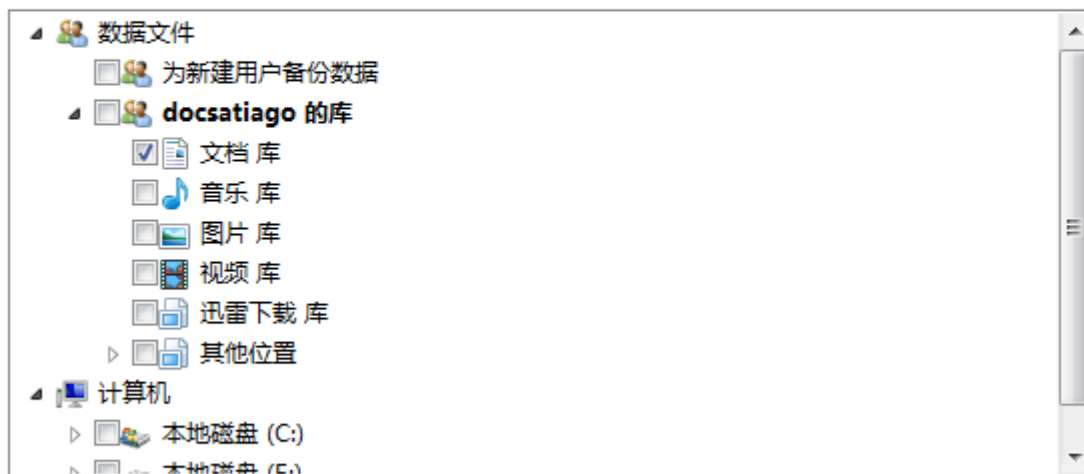


数据备份方案

• 选择备份的数据

您希望备份哪些内容？

选中要包含在备份中的项目对应的复选框。[默认情况下从备份中排除哪些文件？](#)



☐ 包括驱动器 系统保留, (C:), (D:) 的系统映像(S)

选定备份位置不支持创建系统映像。



数据备份方案

- 添加备份计划

您希望多久备份一次？

根据您在下面设置的计划，会将自上次备份后已更改的文件和新创建的文件添加到备份中。

☒ 按计划运行备份(推荐)(S)

频率(H): 每周

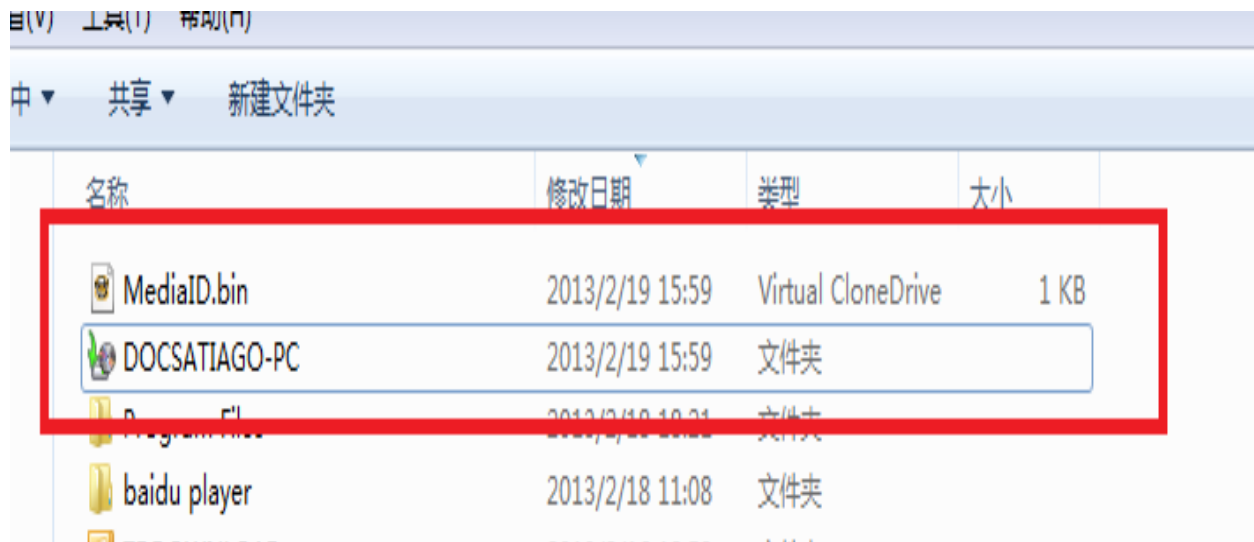
哪一天(W): 星期日

时间(T): 19:00



数据备份方案

- 备份文件保存在指定的磁盘中





本章内容提要

- 磁盘驱动程序
- 缓存管理
- 文件系统接口
- 文件系统驱动程序
- 诊断文件系统问题
- NTFS的恢复支持
- 加密文件系统（EFS）的安全性



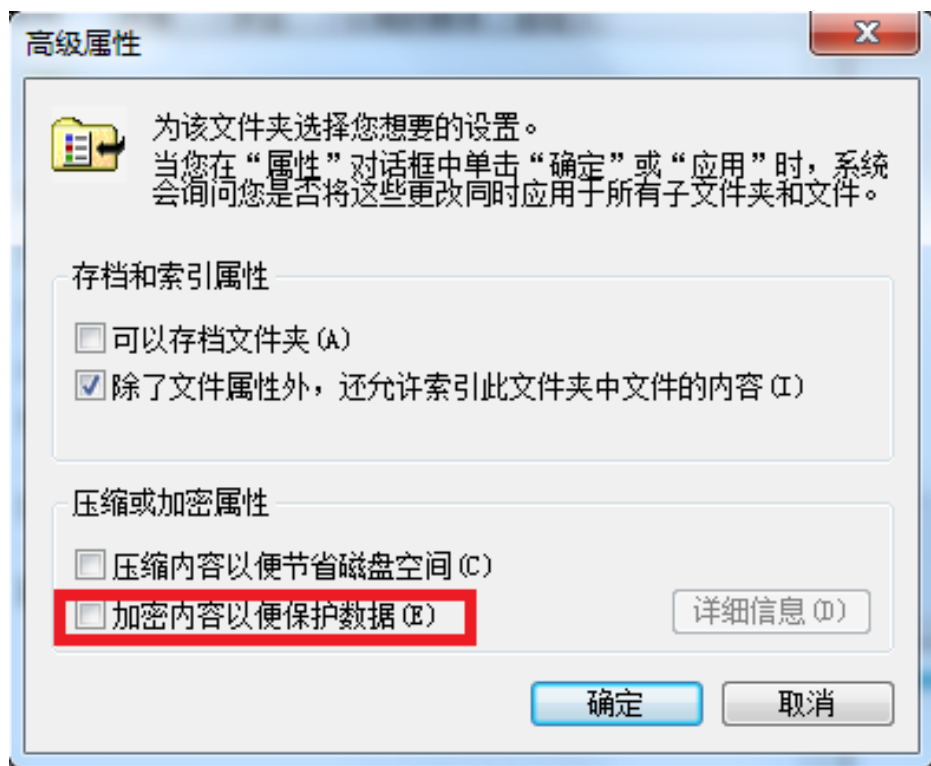
加密文件系统（EFS）的安全性

- EFS的安全性依赖于密码学支持。当一个文件被加密时，EFS为执行此次加密的用户账户分配一对私钥/公钥，以便在加密过程中使用
- 一个文件被加密时，EFS为该文件生成一个随机数，作为文件加密密钥（FEK）
- 加密算法：DES算法的更强变形
- Windows 2000：DESX
- Windows XP及以上：DESX，3DES，AES



加密文件系统（EFS）的安全性

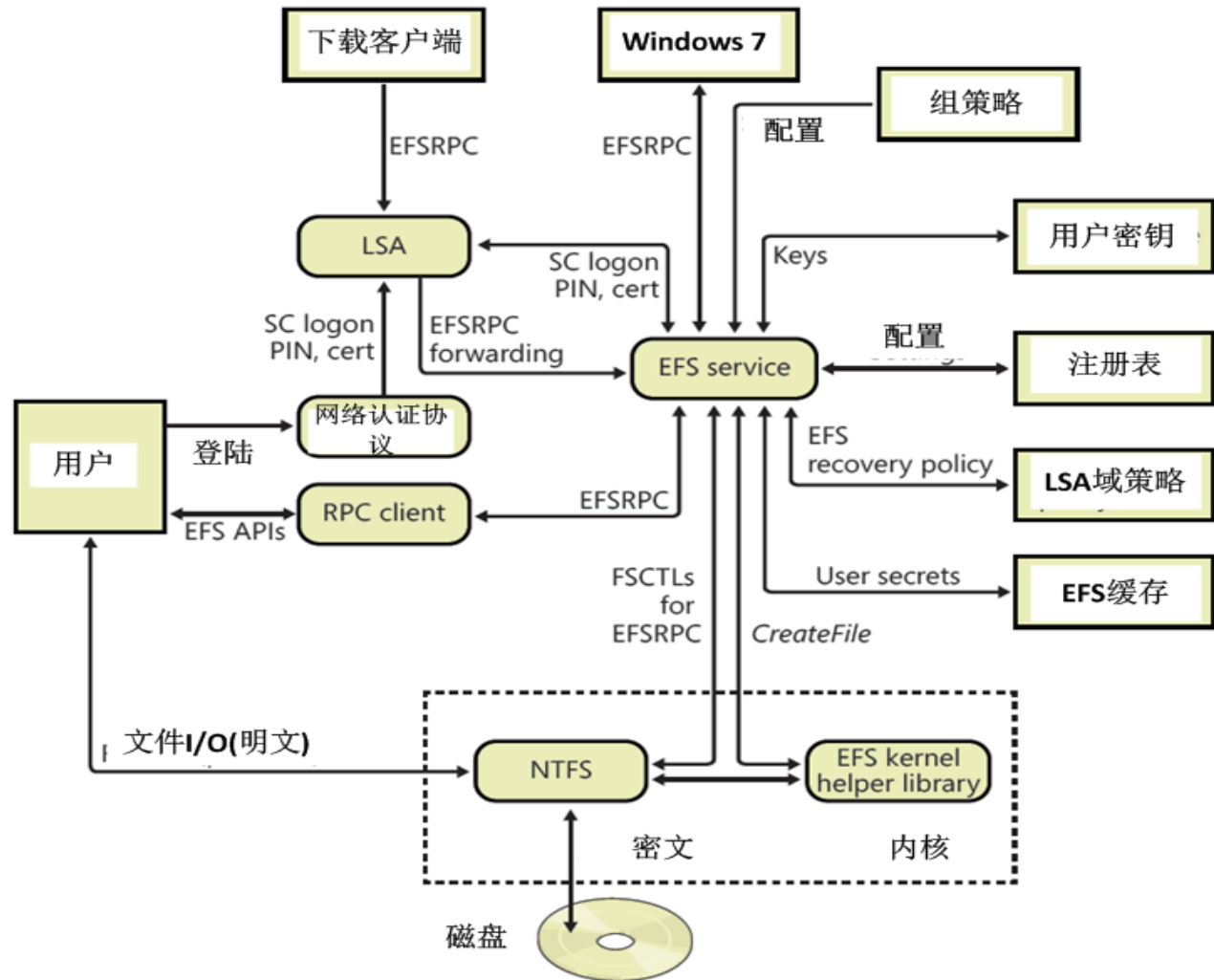
- 使用windows对话框来加密文件





加密文件系统（EFS）的安全性

• EFS的架构





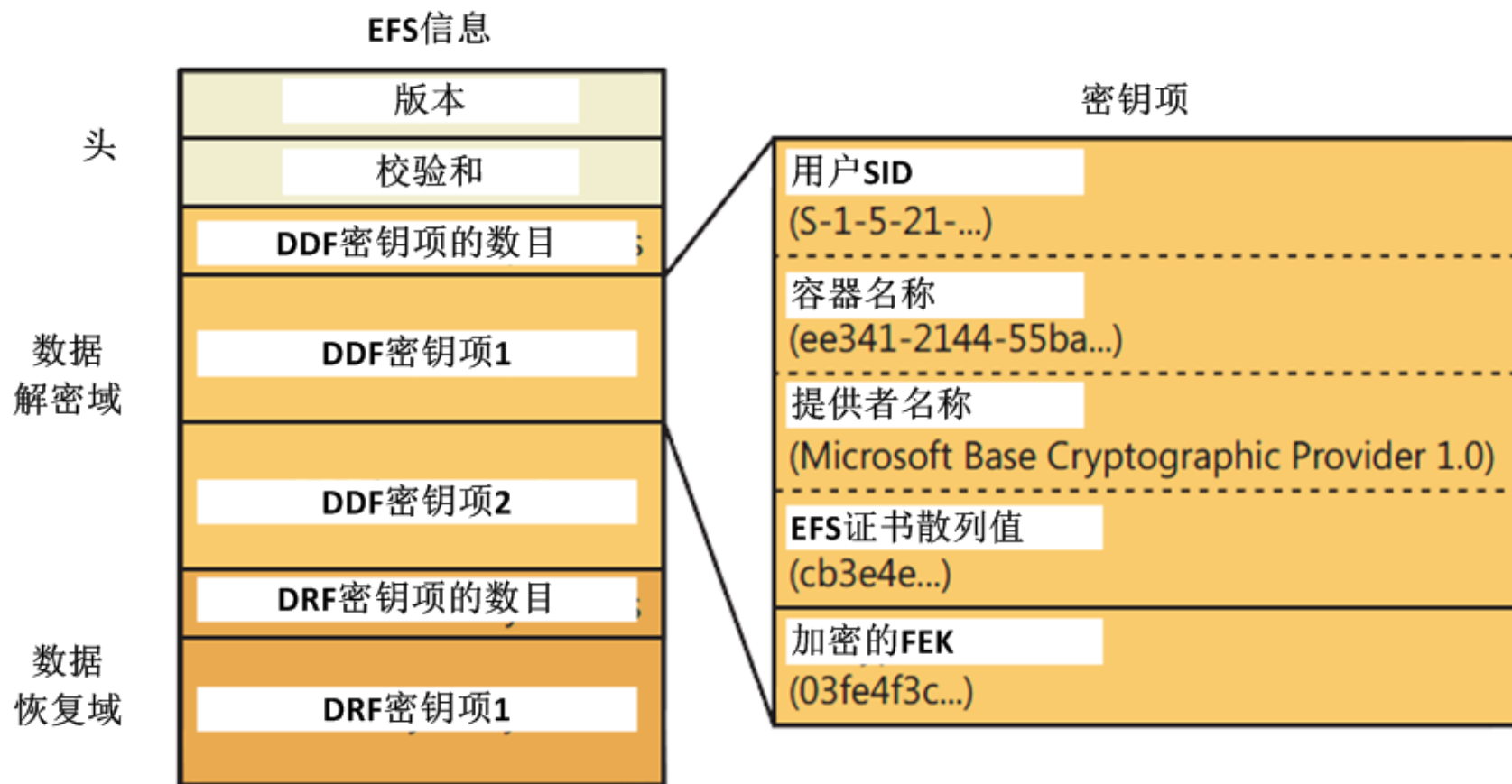
加密文件系统 (EFS) 的安全性

- NTFS驱动程序碰到加密文件时，调用EFS的函数，依赖于Advapi32.dll导出的EncryptFile Windows API函数
- LSASS(本地安全权威子系统)不仅负责管理登陆会话，而且处理与EFS密钥相关的杂务，Lsassrv(LSASS的本地安全权威服务器)组件在监听远过程调用 (RPC) 请求，使用CryptoAPI中的函数来解密此FEK
- CryptoAPI包含了密码服务提供者 (CSP) DLL，使得各种密码服务为应用程序所使用



加密文件系统（EFS）的安全性

• EFS信息格式和密钥项格式





加密文件系统（EFS）的安全性

- 加密过程

- 应用程序请求将数据写到一个加密文件中
- NTFS把数据放在文件系统缓存中
- 缓存管理器延迟把数据写到磁盘
- NTFS请求EFS驱动程序，把将要写到磁盘上的文件内容进行加密
- NTFS将加密的文件写到磁盘上



加密文件系统（EFS）的安全性

- 解密过程

- NTFS识别加密的文件发送请求到EFS驱动程序
- EFS返回DDF（数据解密域）传递到EFS服务器
- EFS服务器返回用户的私钥并解密DDF获得FEK
- EFS服务器传递FEK到EFS驱动程序
- EFS驱动程序利用FEK解密程序需要的文件部分



加密文件系统（EFS）的安全性

- 加密文件的备份

- 备份工具不必具备解密文件数据的能力，在其备份过程中无需解密文件数据
- 备份工具使用EFS API函数
OpenEncryptedFileRaw, ReadEncryptedFileRaw, WriteEncryptedFileRaw和CloseEncryptedFileRaw来访问一个文件的加密内容



加密文件系统（EFS）的安全性

- 复制加密文件

- 当加密文件被复制时，系统并不解密文件再重新加密文件到指定的地址，仅仅拷贝加密的数据和EFS交换数据流到指定的地址
- 复制的地址必须支持加密文件格式，如果不支持，EFS交换数据流将丢失，导致文件只能以非加密的形式被复制



课后实验

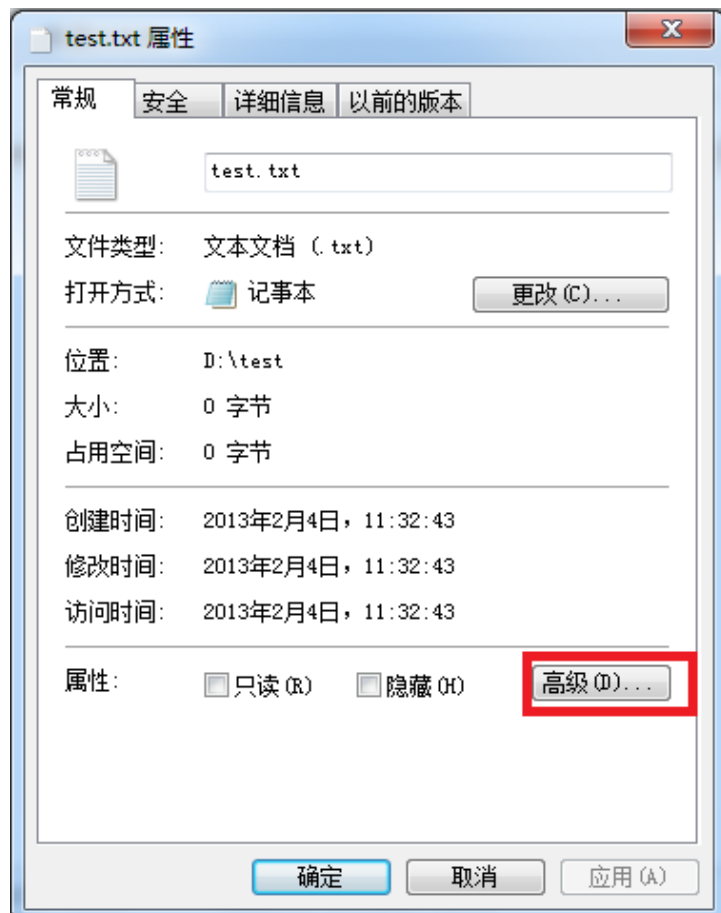
• 实验一 通过界面EFS加密文件





课后实验

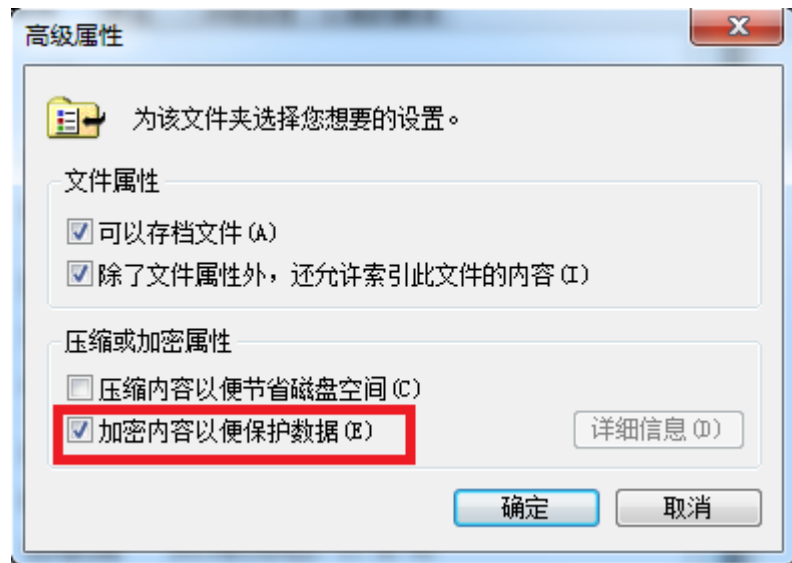
- 点击“高级”按钮





课后实验

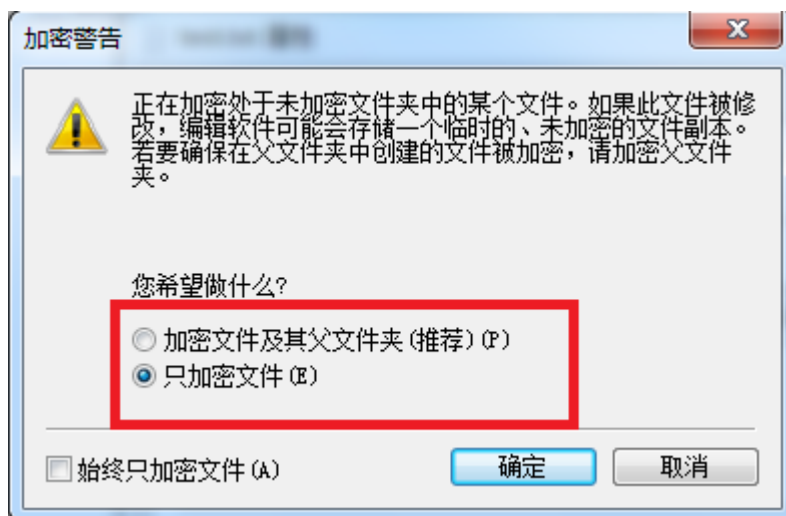
- 在“加密内容以便保护数据”选项前勾选





课后实验


- 点击确定后，根据需求选择加密的方式，点击“确定”





课后实验

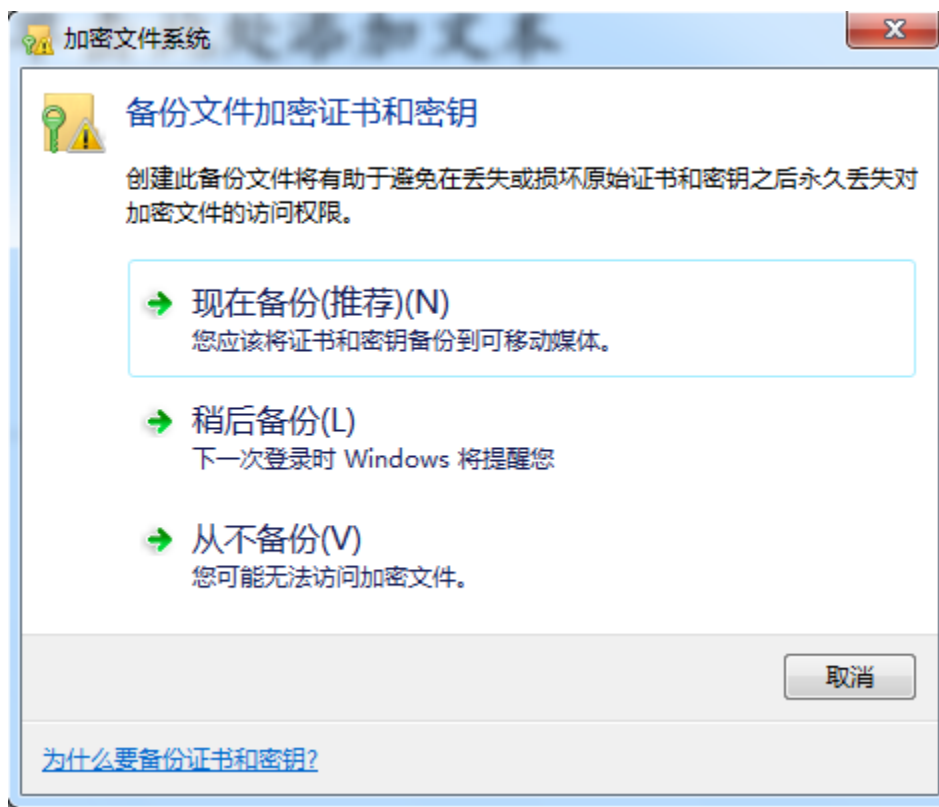
- 加密后的文件颜色发生变化

名称	修改日期	类型	大小
 test.txt	2013/2/4 11:32	文本文档	0 KB



课后实验

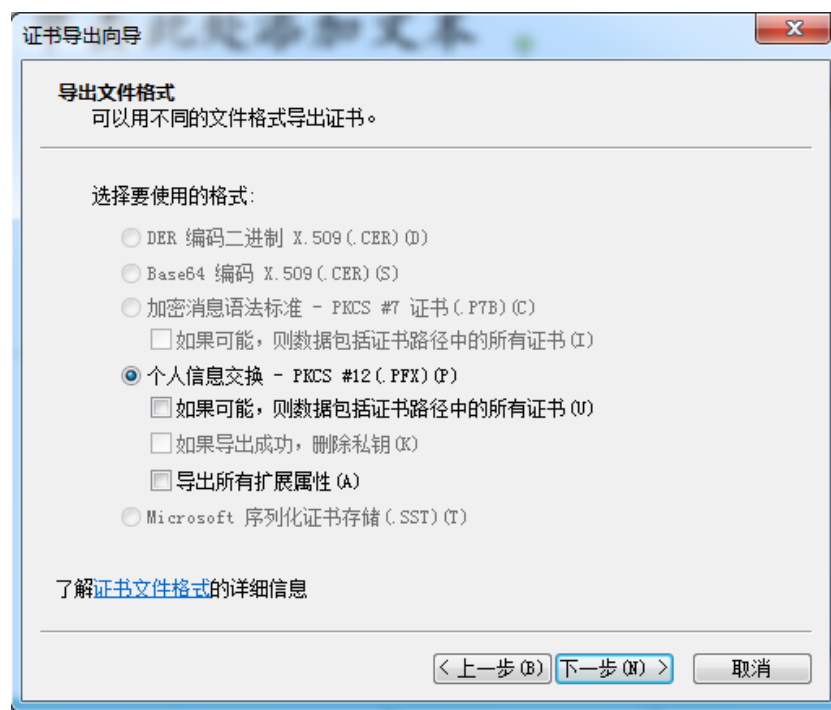
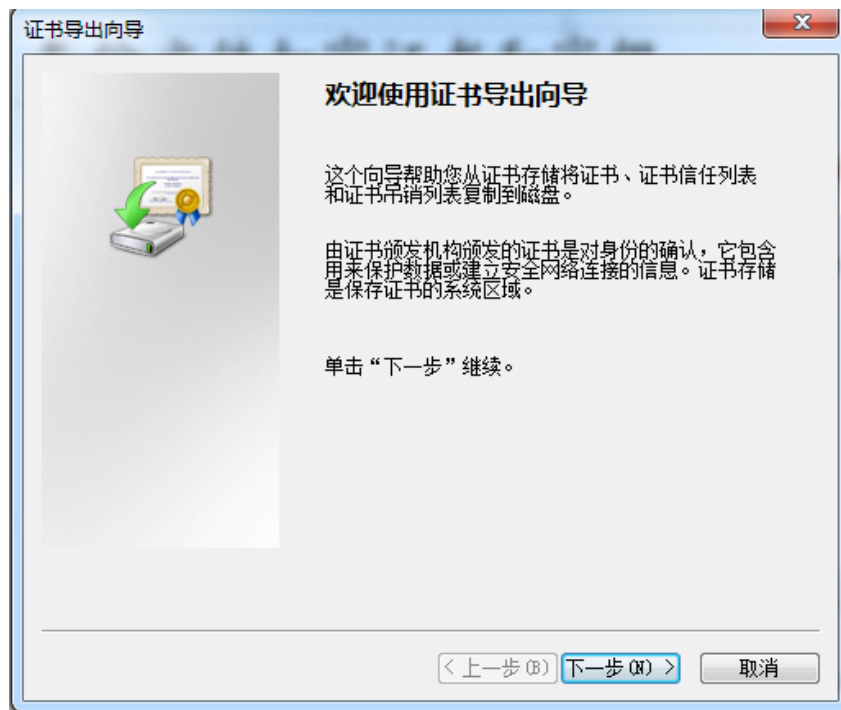
• 实验二 备份文件加密证书和密钥





课后实验

• 证书导出，选择导出的格式





课后实验

- 填写密码和导出证书的文件名

证书导出向导

密码
要保证安全，您必须用密码保护私钥。

输入并确认密码。

密码 (P):

输入并确认密码 (必需) (C):

< 上一步 (B) 下一步 (N) > 取消

证书导出向导

要导出的文件
指定要导出的文件名。

文件名 (F):
 浏览 (R)...

< 上一步 (B) 下一步 (N) > 取消



课后实验

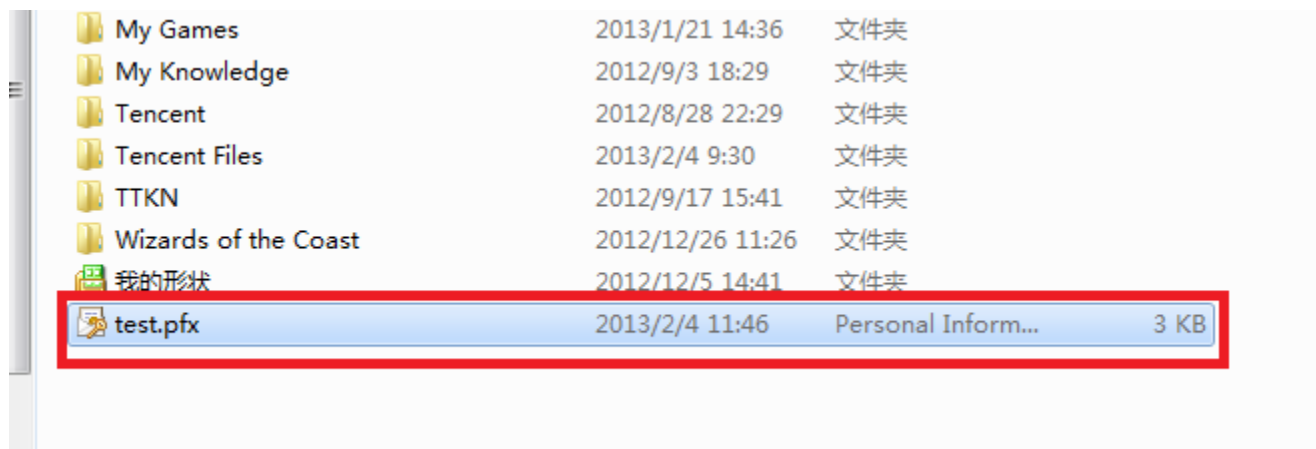
- 导出的证书文件

My Games	2013/1/21 14:36	文件夹	
My Knowledge	2012/9/3 18:29	文件夹	
Tencent	2012/8/28 22:29	文件夹	
Tencent Files	2013/2/4 9:30	文件夹	
TTKN	2012/9/17 15:41	文件夹	
Wizards of the Coast	2012/12/26 11:26	文件夹	
我的形状	2012/12/5 14:41	文件夹	
test.pfx	2013/2/4 11:46	Personal Inform...	3 KB



课后实验

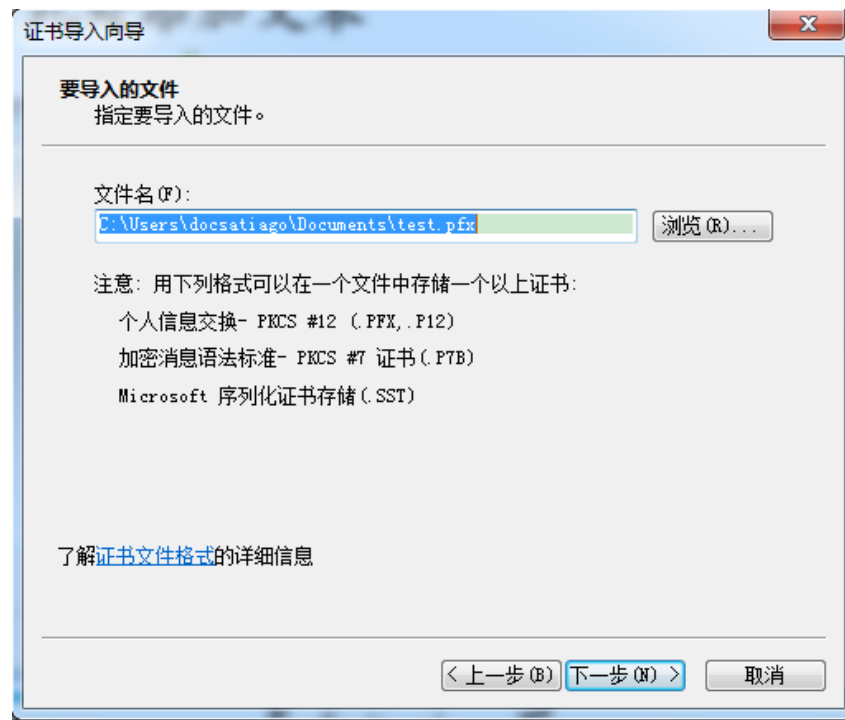
• 实验三 导入备份的加密密钥





课后实验

• 导入选择的证书文件





课后实验

- 输入导出时输入的密码，选择证书导入的位置

证书导入向导

密码
为了保证安全，已用密码保护私钥。

为私钥键入密码。

密码 (P):

●●●●

☐ 启用强私钥保护。如果启用这个选项，每次应用程序使用私钥时，您都会得到提示 (E)。

☐ 标志此密钥为可导出的密钥。这将允许您在稍后备份或传输密钥 (M)。

☒ 包括所有扩展属性 (A)。

了解[保护私钥](#)的更多信息

< 上一步 (B) 下一步 (N) > 取消

证书导入向导

证书存储
证书存储是保存证书的系统区域。

Windows 可以自动选择证书存储，或者您可以为证书指定一个位置。

☒ 根据证书类型，自动选择证书存储 (U)

☐ 将所有的证书放入下列存储 (P)

证书存储:

浏览 (B)...

了解[证书存储](#)的更多信息

< 上一步 (B) 下一步 (N) > 取消



课后实验

- 确定设置无误后，点击“完成”

