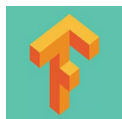


# C# 系统应用之调用SDelete程序粉碎文件及基础原理知识

原创 Eastmount 最后发布于2014-04-19 15:45:40 阅读数 4368 ☆ 收藏

展开



## Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



Eastmount

¥9.90

去订阅

在“个人电脑清除软件”中需要对文件进行彻底粉碎,使其不可恢复.这篇文章主要讲述的是Windows文件系统及调用SDelete程序彻底粉碎文件,希望文章对大家有所帮助.

## 一.Windows文件系统

首先,我回顾汤子丹西电版《计算机操作系统》(此篇部分参考该书)课程中学过的知识,也希望大家去阅读了解,如果不想知道这部分知识的可以跳过.为了文章的完整,我觉得这部分还是必要的.

### 1.文件及文件系统

#### (1).文件系统

为了方便用户使用软件资源,由OS提供的管理文件的软件机构.负责管理在外存上的文件,并把对文件的存取、共享和保护等手段提供给用户,文件系统的管理功能是通过把它所管理的程序和数据组织成一系列文件的方法实现.它是操作系统用于明确磁盘或分区上的文件的方法和数据结构,即在磁盘上的组织文件的方法.

#### (2).文件

它是指具有文件名的若干相关元素的集合,元素通常是记录,而记录又是一组有意义的数据项的集合.基于文件系统的概念将数据组成分为数据项、记录和文件三级.

#### (3).文件控制块

每个文件都有唯一的标识符(FCB,File Control Block,文件控制块),通过该标识符存取文件中的信息.通过文件目录表中文件的名字及物理地址的映射表及FCB可以获取文件中的信息,包括:文件名、文件类型、文件物理位置、文件逻辑结构、存取控制块信息(存取权限)、使用信息(文件创建时间、上次使用时间、当前使用信息)等.

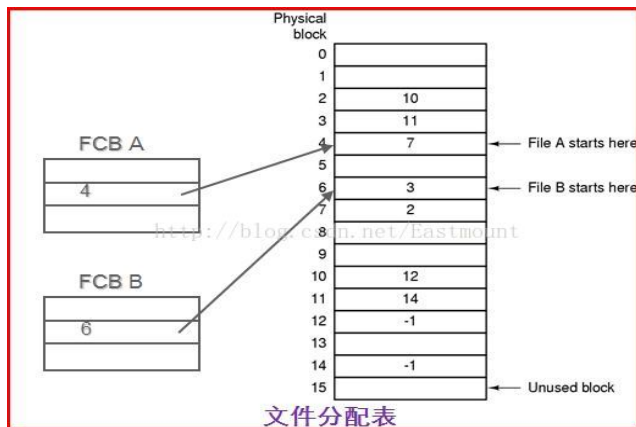
PS:这部分信息只是简单进行了介绍,详细的可看书或自己查阅资料.那么微软的文件系统又是什么呢?

### 2.FAT和NTFS技术

在微软早期的MS-DOS中使用12位的FAT12文件系统,后来使用FAT16.在Windows 95和Windows98操作系统中升级为32位的FAT32,Windows NT、2000、XP操作系统使用新技术文件系统NTFS.而它们所采用的文件分配方式基本都类似于显示链接的方法.

#### (1).显示链接

在外存分配中链接分配方式采取离散的分配方法,有效的消除了外部碎片,提高外存空间利用率.同时,动态分配文件和盘块,十分方便.而显示链接用于链接各物理块的指针,显示地存放在内存的一张链接表中.



其中文件分配表的序号是物理盘块号,从0开始,文件的第一个盘块号都被存储在各相应的FCB中,通过它可查到相应的文件内容,同时在表中存放有链接指针,指向下一个盘块号(链式存储时一个文件可由多个盘块组成).

由于分配给文件的所有盘块号都存储在该表中,故把该表叫文件分配表FAT(File Allocation Table).

## (2).FAT

文件分配表(File Allocation Table,FAT)文件系统因它使用了文件分配表来描述文件系统内存储单元簇的分配情况和文件内容的前后连接关系而得名.包括FAT12、FAT16和FAT32.

簇(cluster)是一个新的分配单位,是一组连续的扇区,在FAT中作为一个虚拟扇区.其大小一般为 $2n$ ( $n$ 为整数)个盘块.在MS-DOS实际中,簇容量可为仅一个扇区(512B)、两个扇区(1KB)、四个扇区(2KB)、八个扇区(4KB).簇的优点是能适应磁盘容量不断增大.

FAT12的簇标识为12位(二进制数),在FAT12表中最多允许有4096个表项,即最多只能将一个磁盘分区分为4096个簇.如果采用以盘块作为基本分配单位,每个盘块(扇区)大小是512字节,每个磁盘分区容量是2MB( $4096 \times 512B$ ).

这里,FAT16和FAT32就不过多讲述,希望读者自己研究,感觉能力有限,有点讲述不清楚.

FAT文件系统可分为三部分:保留区(DBR)、FAT区和数据区.如下图所示:



a.保留区:存储一个重要的数据结构——系统引导扇区(DBR),其位于文件系统的0号扇区,DBR存储了其所在文件系统有关的重要信息.

b.FAT区:具有两个大小、数据内容相同的FAT表组成,紧跟在保留区后.FAT表用于描述数据区中数据存储单元的分配状态及为文件或目录内容分配的存储单元的前后连接关系.

c.数据区:它被划分为一个个簇,用于存储用户数据.

## (3).NTFS

NTFS(New Technology File System,新技术文件系统)是一个为Windows NT开发的、为特别网络和磁盘配额、文件加密管理安全特性的全新文件系统,并适用于Windows 2000\XP\2003.NTFS也是以簇为单位存储数据文件的,其簇的大小并不依赖磁盘或分区的大小,由于簇尺寸有缩小,故能减少产生磁盘碎片、降低磁盘空间浪费.



图1 NTFS文件系统结构

在NTFS文件系统中,使用卷表示一个逻辑磁盘.它存在一张**主控文件表MFT(Master File Table)**,将一个卷中的所有文件信息、目录信息及可用的未分配空间信息记录在MFT中.NTFS利用B-Tree文件管理方法跟踪文件在磁盘上的位置.现在使用的Win7/Win8/Vista都是NTFS文件系统.如图:



右键“我的电脑”->管理 查看磁盘管理可以发现NTFS的文件系统

## 二.恢复文件简单原理

(该例参考戴士剑的《数据恢复技术》书籍)

在日常生活中我们可能清空回收站或使用Shift+Delete删除文件,但其实它是可以恢复的.

这就好比磁盘文件系统想象成一本书,需要找到某部分内容时,一般通过书最开始部分目录索引查找,当删除一部分内容时,只简单的删除目录索引,而内容可以通过其他方式找到(如一页页查找).同样,OS存取文件时是以簇为单位进行,一簇包括若干实际磁盘空间(扇区),同时以簇链的形式保存文件分配表(FAT),相当于索引.当删除一个文件会将文件在分配表中簇标识记为“空簇”.但数据区会保留着文件内容,因此可以恢复.

下面是通过数据恢复精灵恢复例子:使用Shift+Delete永久删除F盘下两个文件“获取Cookies.txt”和“C#WinForm学习.docx”,使用数据恢复精灵可以实现恢复.(我的是未注册版本,仅成功尝试恢复了txt和word文件).



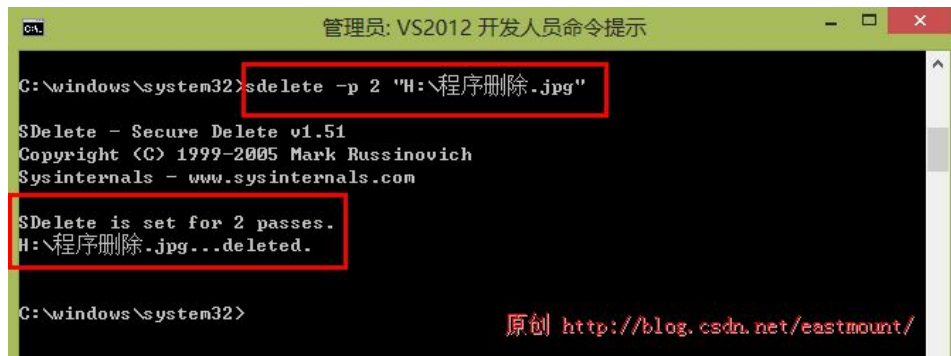
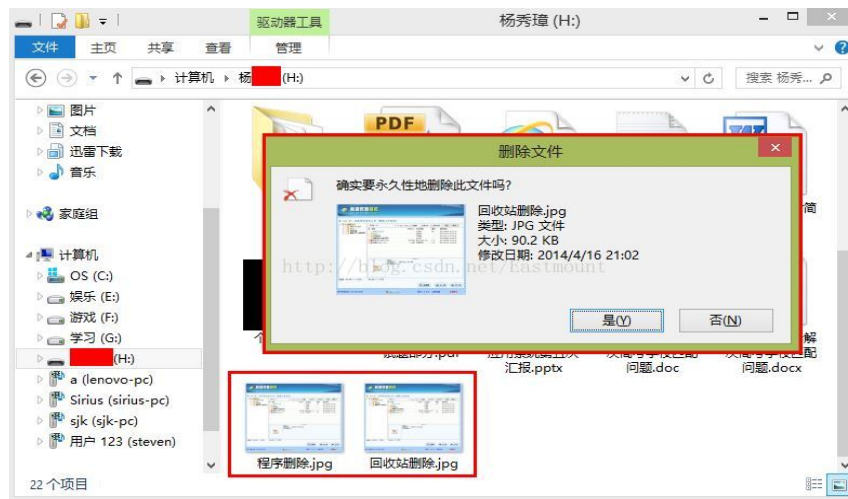
### 三.cmd调用SDelete程序及参数

微软提供了SDelete程序安全删除文件并使磁盘数据无法恢复,使用SDelete安全删除文件也可以安全地擦除存在磁盘的未分配部分中的任意文件数据(包括已删除或加密的文件).SDelete程序实施了美国国防部摧毁标准(Clearing and Sanitizing Standard) DOD 5220.22-M.

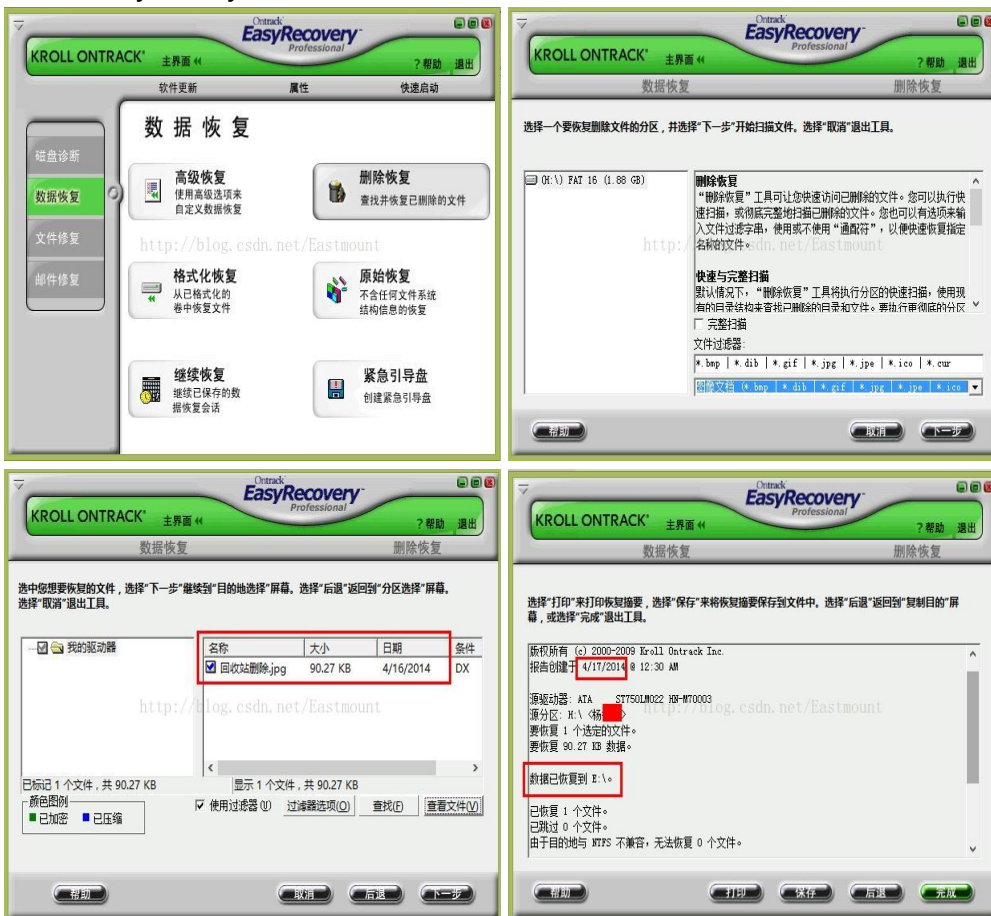
```
//SDelete程序删除一个或多个文件或目录,亦可清理逻辑磁盘上可用空间
sdelete [-p passes] [-s] [-q] <文件或目录> //<file or directory>
sdelete [-p passes] [-z] [驱动器符号] // [drive letter]
// 参数含义
-a Remove Read-Only attribute
-c Clean free space
-p passes 指定覆盖操作的执行遍数(default is 1,默认是1)
-s or -r 对子目录执行递归操作
-q 不显示错误(静默模式)
-z 清理可用空间(good for virtual disk optimization)
```

由于SDelete是一个命令行工具,需要从官方地址下载:放置于%system%目录"C:\Windows\System32\sdelete.exe".<http://technet.microsoft.com/en-us/sysinternals/bb897443.aspx>

我在移动硬盘H中放入两张图片"程序删除.jpg"和"回收站删除",删除过程如下图所示,cmd程序中输入sdelete -p 2 "H:\程序删除.jpg".调用SDelete并执行2次覆盖操作.



下面是使用EasyRecovery软件能实现对回收站删除的图片恢复,而使用SDelete删除不能恢复的图片.最后实现了回收站删除图片的恢复功能.



思考:为啥需要覆盖操作执行多遍呢?我的老师给我说删除6遍时较干净,Why?

## 四.C#实现粉碎文件(未实现)

(部分参考丁士锋的《C#典型模块与项目实战大全》)

我打算通过创建Process,并设置其参数实现删除文件功能,代码如下:

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        // 检查文件是否存在
        if (!File.Exists(textBox1.Text.ToString()))
        {
            MessageBox.Show("输入文件不存在!");
        }
        else if (Convert.ToInt32(textBox2.Text) < 1) // 覆盖次数大于1
        {
            MessageBox.Show("覆盖次数要大于1");
        }
        else
        {
            // 初始化进程 命名空间 using System.Diagnostics
            Process da = new Process();
            // 设置要启动应用程序或文档名
            da.StartInfo.FileName = @"C:\WINDOWS\system32\sdelete.exe";
            // 设置要启动应用程序使用的一组命令行参数
            // 参数: 执行覆盖操作次数 删除文件路径
            da.StartInfo.Arguments = string.Format("-p {0} -q \"{1}\"",
                Convert.ToInt32(textBox2.Text), textBox1.Text.ToString());
            // 设置窗口状态 隐藏窗口显示
            da.StartInfo.WindowStyle = System.Diagnostics.ProcessWindowStyle.Hidden;
            // 直接执行程序 不使用Shell打开
            da.StartInfo.UseShellExecute = false;
            // 输入流 输出流 错误写入流
            da.StartInfo.RedirectStandardInput = true;
            da.StartInfo.RedirectStandardOutput = true;
            da.StartInfo.RedirectStandardError = true;
            // 不创建窗口, 静默模式
            da.StartInfo.CreateNoWindow = true;
            // 开始执行进程
            da.Start();
            // 从输出流中读取程序输出
            string output = da.StandardOutput.ReadToEnd();
            // 等待程序退出
            da.WaitForExit();

            // 文件不存在记录输出流信息
            if (File.Exists(textBox1.Text.ToString()))
            {
                richTextBox1.AppendText(output);
            }
        }
    }
    catch (Exception msg) // 异常处理
    {
        MessageBox.Show(msg.Message);
    }
}
```



但是,虽然已经把sdelete.exe放置于“C:\windows\system32”文件夹下,使用cmd运行才能正确删除,但当通过C#代码调用时总是会显示错误“系统找不到指定文件”.这让我万分伤心啊,但同时该方法的不足之处是需要用户放置sdelete程序,因此我需要寻求新的方法实现粉碎文件.



但是代码中所涉及的使用Process的方法还是值得大家学习的,我们还可以通过它实现很多功能,如下:

```
//IE浏览器打开百度
Process.Start("IExplore.exe", "http://www.baidu.com/");
//打开库
Process.Start("explorer.exe");
//打开Excel办公软件
Process.Start("EXCEL.exe");
//打开cmd
Process.Start("cmd.exe");
```

关于Process的详细用法建议大家阅读下面这篇优秀文章:[http://blog.csdn.net/chen\\_zw/article/details/7896264](http://blog.csdn.net/chen_zw/article/details/7896264)

下面是调用cmd.exe程序实现ipconfig查看功能,但是当使用"sdelete -p 2 \"F:\test.txt\""时还是不能运行,我也不知道为什么?不知道怎么访问Sdelete.exe程序,使用管理员权限运行也不行.

```
string command = "";
System.Diagnostics.Process p = new System.Diagnostics.Process();
p.StartInfo.FileName = "cmd.exe";           // 文件名
command = "/c" + "ipconfig";
//command = "/c" + "sdelete -p 3 \"F:\\bbb.txt\"";
p.StartInfo.Arguments = command;           // 参数
p.StartInfo.UseShellExecute = false;        // 是否使用操作系统shell启动
p.StartInfo.RedirectStandardInput = true;   // 接受来自调用程序的输入信息
p.StartInfo.RedirectStandardOutput = true;  // 由调用程序获取输出信息
p.StartInfo.RedirectStandardError = true;   // 重定向标准错误输出
p.StartInfo.CreateNoWindow = true;         // 不显示程序窗口
p.Start();                                 // 启动程序
string output = p.StandardOutput.ReadToEnd();
richTextBox1.AppendText(output);
```

cmd中ipconfig的运行结果如下,但使用sdelete参数就是不行(>.<):



总结:最后希望该篇文章对大家有所帮助,它仅仅是一篇基础性的设计文件系统的文章,如果文章中有错误或不足之处,还请海涵!同时文章主要参考了文中提到的3本书籍,如果有认为文章过于啰嗦或不喜欢的,也请见谅!写这篇文章真心有点难,最难过的是没有实现,如果有知道的请说声.我开始寻找更好的粉碎文件的方法了.

杨友山老师的粉碎文件:<http://blog.csdn.net/yysyangyangyangshan/article/details/18423731>

(By:Eastmount 2014-4-19 下午4点 原创CSDN<http://blog.csdn.net/eastmount/>)

点赞 收藏 分享 ...



Eastmount 博客专家

发布了450 篇原创文章 · 获赞 6227 · 访问量 496万+

他的留言板

关注