

操作系统课程设计

****

**题 目**

学生姓名

学 号

学 院 计算机学院、网络空间安全学院

专 业

指导教师 闫雷鸣

**二Ｏ四零年十二月**

DiskGenius 数据恢复

摘要

本课程设计实验主要围绕U盘数据恢复展开，结合操作系统的基本原理和工具进行深入探讨。实验首先从对U盘进行格式化开始，通过分析不同文件系统（如FAT32、NTFS、EXT4）在磁盘上的表现，探讨了磁盘空间的分配方式、簇链的计算以及文件在磁盘中的存储过程。通过DiskGenius等工具模拟了U盘的数据删除与恢复过程，并设计了一个较为复杂的数据损坏场景进行测试，最终使用相关工具成功恢复了丢失的数据。具体实验步骤包括：

**读取主引导扇区及文件创建过程**：通过对空U盘的主引导扇区进行分析，记录其初始内容，并通过在U盘中创建文件，观察主引导扇区和分区表的变化。文件创建后，分析其簇分配情况，并计算文件在磁盘中的地址。

**文件删除与数据恢复**：通过删除文件并修改分区信息，尝试恢复文件。使用了DiskGenius进行数据恢复，并详细记录了恢复过程。通过破坏磁盘的某些区域，再使用数据恢复工具恢复数据，评估了数据恢复的可行性与完整性。

**可引导盘制作与主引导扇区破坏**：将U盘制作成可引导盘后，读取其主引导扇区并与空盘进行对比。通过故意破坏主引导扇区，观察系统启动异常情况，进一步加深对MBR和GPT分区表以及主引导扇区结构的理解。

**复杂数据破坏场景模拟与恢复**：通过模拟数据破坏场景，如文件系统损坏、分区表丢失等，使用DiskGenius恢复丢失数据。记录了整个恢复过程，分析了恢复成功的条件和步骤，特别是如何保证数据不被覆盖并完整恢复。

实验过程中，所用的工具和技术包括DiskGenius、Win32DiskImager等，帮助完成了数据恢复、破坏与修复等操作。在操作过程中，注意到文件系统（如FAT32和NTFS）的差异，特别是在数据分配、文件存储结构和分区表的更新机制上，进一步理解操作系统在磁盘管理方面的运作原理。在实验过程中，重点分析了主引导扇区的结构和分区表的内容，并通过修改主引导扇区（MBR）破坏系统的引导过程，进一步加深了对磁盘和文件系统管理的理解。通过对文件系统结构、磁盘簇链以及扇区的详细分析，掌握了如何根据文件的起始位置及簇链计算其在磁盘中的物理地址。

关键词：主引导扇区、分区表、数据破坏模拟、DiskGenius、U盘数据恢复

目 录

[题 目 1](#_Toc29755)

[一、 U盘的格式化与文件系统选择 4](#_Toc10614)

[1. 1 U盘的格式化 4](#_Toc12225)

[二、 读取和修改U盘主引导扇区 4](#_Toc22241)

[2.1读取主引导扇区（MBR） 4](#_Toc28589)

[2.2创建文件并观察记录变化 6](#_Toc1588)

[2.3计算文件在磁盘中的地址 8](#_Toc15755)

[2.3.1获取文件的起始簇号 8](#_Toc3504)

[2.3.2计算文件数据的物理地址 8](#_Toc17908)

[2.4删除后尝试恢复文件 9](#_Toc16761)

[三、 可引导盘制作与修改 11](#_Toc18086)

[3.1 将 U 盘做成可引导盘 11](#_Toc25052)

[3.1.1 镜像文件下载 12](#_Toc6709)

[3.1.2 可引导盘制作 12](#_Toc29870)

[3.2读取主引导扇区与空盘进行对比 13](#_Toc13930)

[3.3修改或者破坏主引导扇区并观察结果 15](#_Toc10930)

[3.3.2更改分区表 16](#_Toc23189)

[四、 设计的U盘数据破环场景并恢复数据 17](#_Toc14637)

[4.1创建数据破坏环境 17](#_Toc21841)

[4.1.1格式化U盘： 17](#_Toc4364)

[4.1.2 模拟创建数据环境 17](#_Toc17811)

[4.1.3 模拟破坏数据环境 18](#_Toc26698)

[4.2已破坏数据恢复 20](#_Toc6387)

[4.2.1恢复分区 20](#_Toc28536)

[4.2.2恢复文件内容 20](#_Toc22938)

# U盘的格式化与文件系统选择

## 1 U盘的格式化

首先，选择一个空U盘，进行格式化并选择不同的文件系统格式，包括NTFS、FAT32和EXT4。由于Windows本身不支持EXT4格式，此处在Windows下使用FAT32。插入U盘，打开我的电脑右键点击U盘，选择“格式化”，如图1所示。

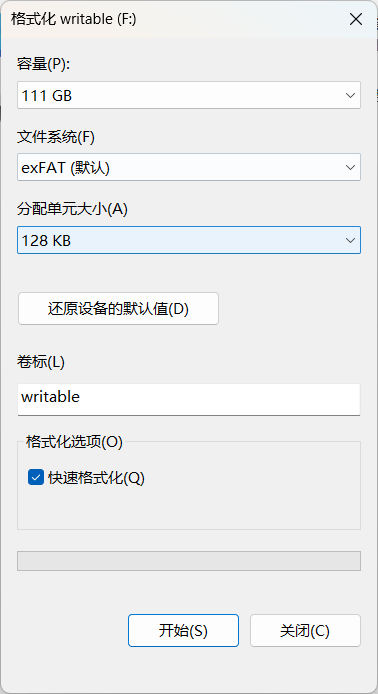
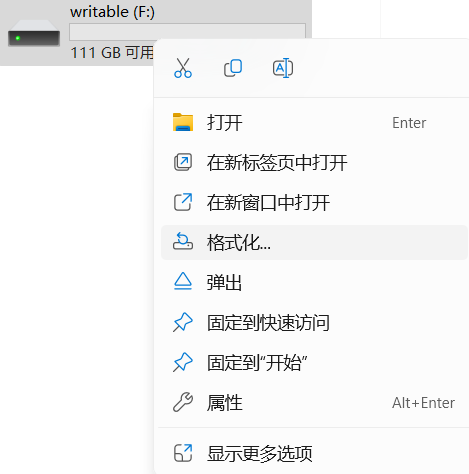


图 1 图 2

如图2所示，此处选择FAT32格式，适合较小的存储设备。U盘常用此格式，兼容性好，但单文件大小限制为4GB，格式化后，U盘变为一个空的存储设备。

# 读取和修改U盘主引导扇区

## 2.1读取主引导扇区（MBR）

主引导扇区（MBR） 位于磁盘的第一个扇区，也就是磁盘的物理偏移量为 0 的地方。MBR 是磁盘的第一个 512 字节（包括 446 字节的启动程序、64 字节的分区表和 2 字节的魔术数）部分。

使用DiskGenius工具读取U盘的主引导扇区。

1.打开DiskGenius，选择U盘所在的磁盘。

2.右键点击U盘，选择“打开磁盘”查看磁盘结构。

3.找到U盘的主引导扇区（MBR），主引导扇区（MBR）位于磁盘的第一个扇区，偏移量为 0x0000，也就是磁盘的第一个 512 字节区域，并查看其中的分区表和引导代码。

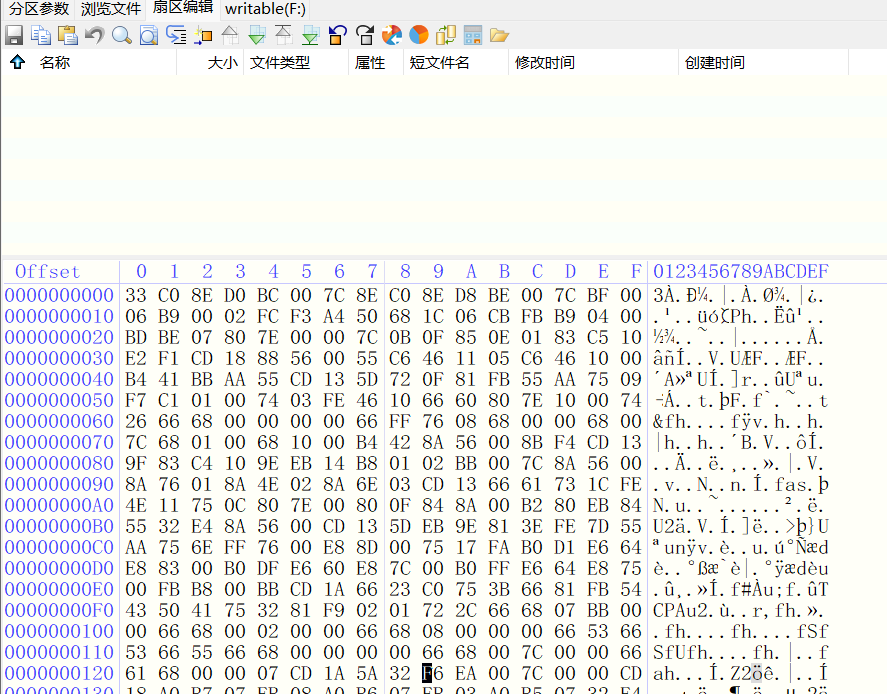


图 3 MBR记录

如图3所示，能够观察到在不同偏移位置上的存储数据。其中Offset 指的是扇区在磁盘上的位置。它是从磁盘的开始位置（即主引导扇区，MBR）到当前扇区的偏移量，以十六进制表示。数据以 16 个字节为一行，并且每行会显示从 0 到 F 的十六进制数据。

右边的字符串部分显示的是当前扇区内容的ASCII 字符表示，只显示那些能转化为可打印字符的字节。如果一个字节对应的 ASCII 字符无法打印，则会显示为 .（点），表示该字节不能直接转换为字符。分析16进制下的内容能够得知MBR记录的位置。

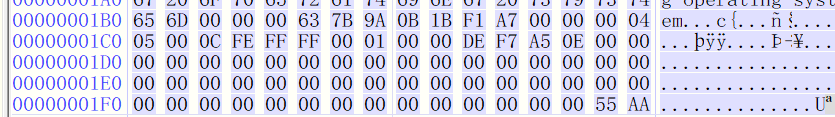


图 4 MBR分区标志

55 AA 是典型的 MBR（主引导记录） 引导扇区的标志，表示扇区的开始。MBR 中通常会包含引导程序和分区表等信息。在这里，55 AA 是扇区标记，指示着扇区的有效性。这通常表示这段数据是引导扇区的一部分，而不是未格式化的数据。在MBR中包含了以下两种结构：

分区表（Partition Table）：

分区表通常会出现在 MBR 中。它包含了磁盘的分区信息，如分区类型、起始和结束位置等。

引导代码（Boot Code）：

负责启动操作系统。在这段数据中能找到一些与引导过程相关的机器代码。

另外，有一部分16进制码能以ASCII码形式翻译成如下信息。

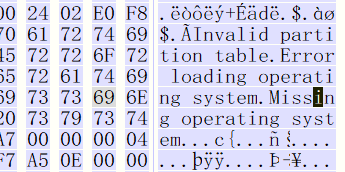


图 5 MBR中能够被翻译的部分

这是以 ASCII 可打印字符的形式显示出来的内容，翻译后为：

"validation partition tableError loading operating systemMissioning operating system"

这里似乎包含了一些文本信息。可以看到有一部分是 错误信息，表示在该区域保存了一些操作系统或分区信息。

## 2.2创建文件并观察记录变化

在U盘中创建一个新文件empty.txt如下图所示：



图 6 空文件创建

查看当前的分区信息如下：

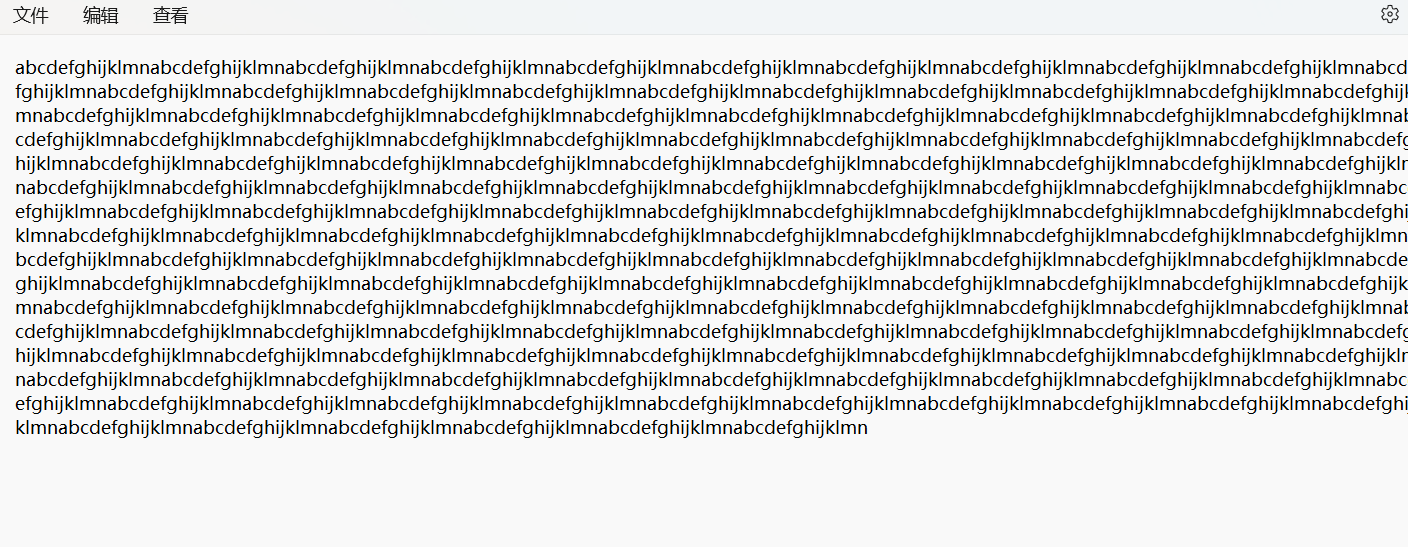


图 7 分区情况

由上图得知在 FAT32 文件系统中，数据是以 簇（Cluster） 为单位存储的，每个簇由若干个扇区组成（通常为1到64个扇区，具体取决于文件系统的配置）。图中每个簇是 64 KB，即一个簇包含 128个扇区（512字节/扇区）。

此处已用簇数变为1 个簇，说明现在U盘上已经创建了一个小文件（文件没有内容，但FAT32仍为其分配一个完整的簇）。新文件会占用一个簇，并在FAT表中标记为已使用，簇号会记录在FAT表中。

再次新建一个文件，观察后续的变化。新建一个文件名为file.txt，并在其中输入内容。



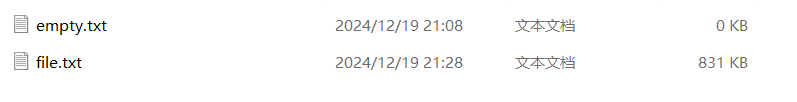


图 8 新建有内容文件

这里连续复制多次使文件大小达到100kb以上，观察变化。

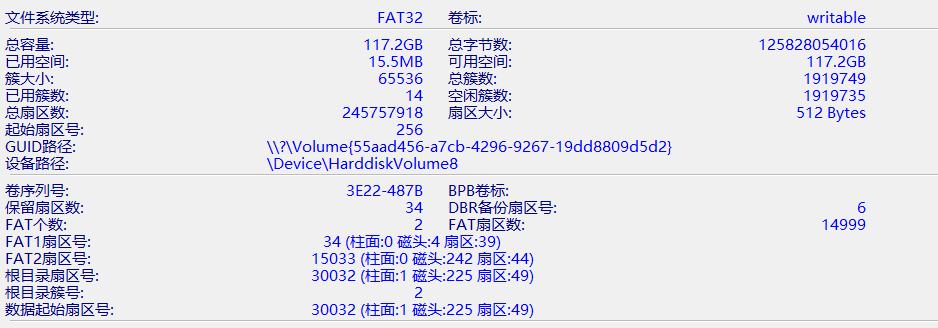


图 9 再次新建文件后的分区信息

能够发现已用簇数从1变为了14。这是由于新建文件的大小超过了一个簇的大小65536 (换算后为64KB）因此其占据了不止一个簇。此处文件大小为831KB ，大约占用了 13 个簇，文件的起始簇号在 FAT 表 中记录，并且通过簇链来链接所有的簇。

## 2.3计算文件在磁盘中的地址

在 FAT32 文件系统中，通常涉及以下几个步骤：

1.获取文件的起始簇号。

2.计算文件数据的物理地址，也就是文件数据在磁盘上的实际扇区地址。

3.追踪簇链，直到文件占用的所有簇都被遍历完。

### 2.3.1获取文件的起始簇号

在 FAT32 文件系统中，文件的起始簇号保存在文件的 目录表项 中。通过 DiskGenius 或其他工具，查看文件的 起始簇号。



图 10 起始簇号及扇区查看

此处能够看到在改文件系统中，起始簇号为2。起始扇区为30032。

### 2.3.2计算文件数据的物理地址

FAT32 文件系统中的每个簇占据一定数量的扇区，通常是 1 个簇 = 1 个或多个扇区（通常每个簇的大小为 16KB、32KB 或更大，具体大小取决于磁盘格式）。每个簇有一个对应的物理地址，它指向磁盘上的一块实际区域。

根据之前截图可知：簇的大小是 65536 字节（即 64KB），每个扇区的大小是 512 字节。这样每个簇包含 128 个扇区（65536 / 512 = 128）。

公式：

簇的起始扇区地址 = 起始扇区号 + (簇号 - 2) \* (簇大小 / 扇区大小)

簇号为 2，文件的实际物理扇区地址就是 256，因此它占据了磁盘的第 256 个扇区开始，接着每个簇会占据 128 个扇区。此处起始扇区为起始扇区为30032。由于之前先创建了一个 0 KB 文件，文件系统会分配一个簇占用 64 KB 的空间，但是没有实际数据。它占用簇 2。因此后面创建的文件的起始簇号应该为3。因此，使用上述公式计算，实际有文件的物理起始扇区为 30160，结束扇区为 31822。在Diskgenius中查找扇区检验是否正确。



图 11 输入地址查询

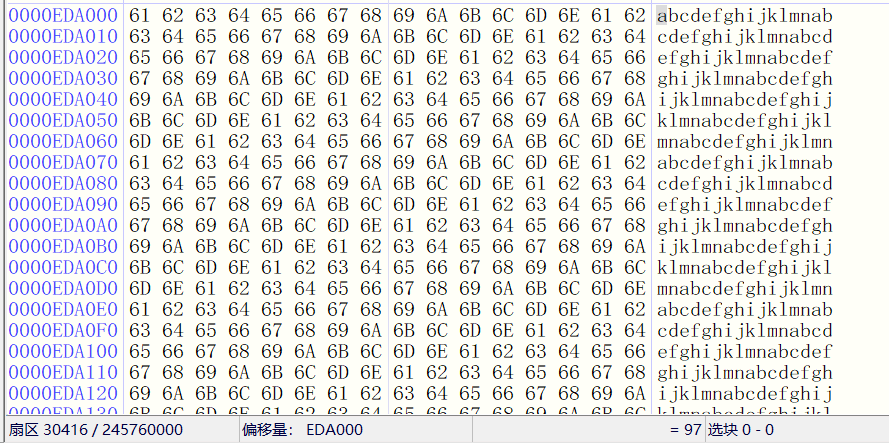


图 12 起始位置扇区查询

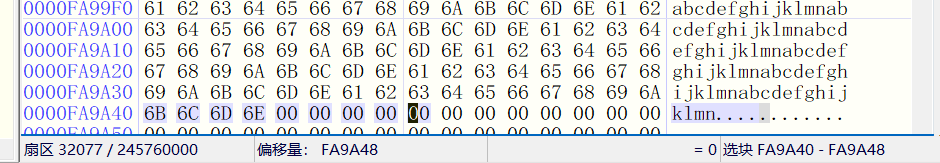


图 13 终止位置扇区查询

起始扇区能够看到abcd...，终止扇区能够看到klmn....。因此能够看到成功地计算了文件的地址信息。

## 2.4删除后尝试恢复文件

操作系统中删除文件时，文件并不会立即从磁盘中物理删除，而是操作系统会将该文件的目录项标记为“已删除”，并且标记相应的簇为可用。只要这些簇没有被其他数据覆盖，理论上是可以恢复文件的。

删除该文件时，操作系统会在 FAT 表中标记文件所占用的簇为“可用”。删除文件的目录项（包括文件名和起始簇号），但是数据并未被清除。

将上述831KB的测试文件删除

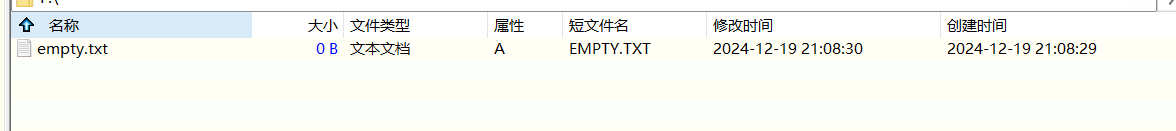


图 14 删除后的文件目录



图 15 删除后已用簇数

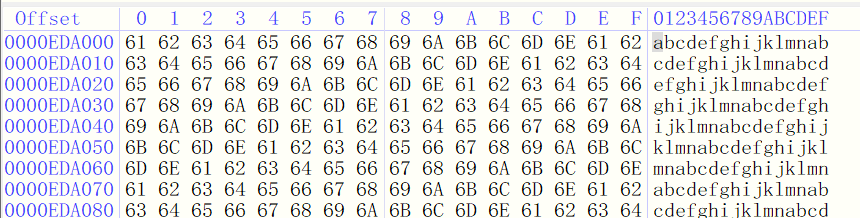
可见该文件的目录项会从根目录（或者相关目录）中移除，已用簇数 会减少，而 空闲簇数 会增加。文件已被删除。但实际上查看刚刚计算出的物理地址并进行跳转查询，能够看到物理上数据并未被删除。

图 16 删除后扇区位置的数据

因此只要这些簇没有被其他数据覆盖，理论上是可以恢复文件的。

此处可以使用使用 DiskGenius 或其他数据恢复工具(或者直接手动修改分区表）

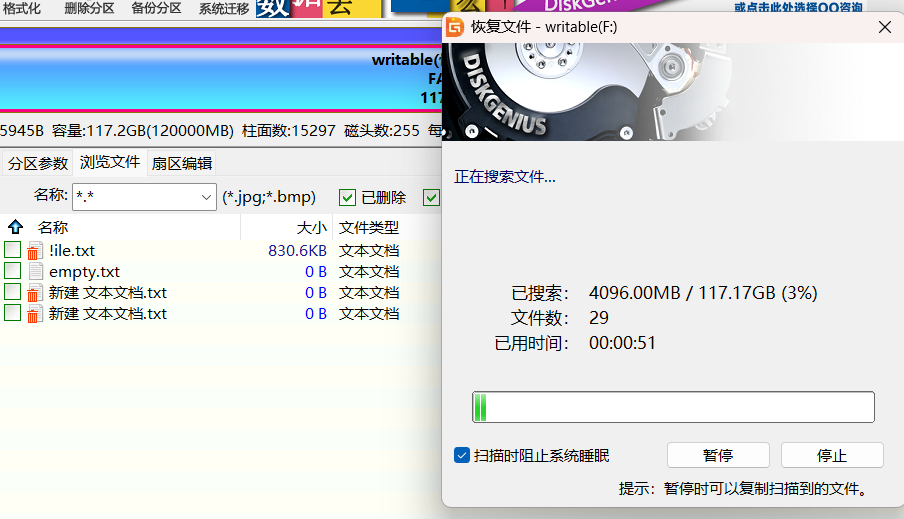


图 17 文件恢复

1.启动 DiskGenius 并选择对应的 U 盘。

2.扫描已删除的文件：使用 DiskGenius 的恢复工具，选择对 U 盘进行深度扫描。扫描文件簇链，并标识出已删除但未被覆盖的数据块。

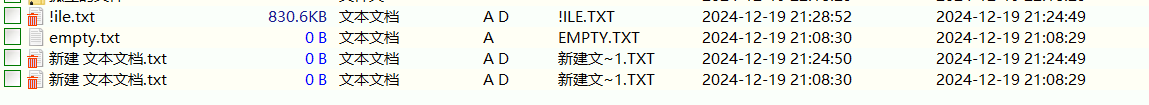


图 18扫描出的可恢复文件

在扫描结果中，DiskGenius 列出已删除的文件和簇。通过查看文件大小、修改时间等信息来确定是否是删除的文件。

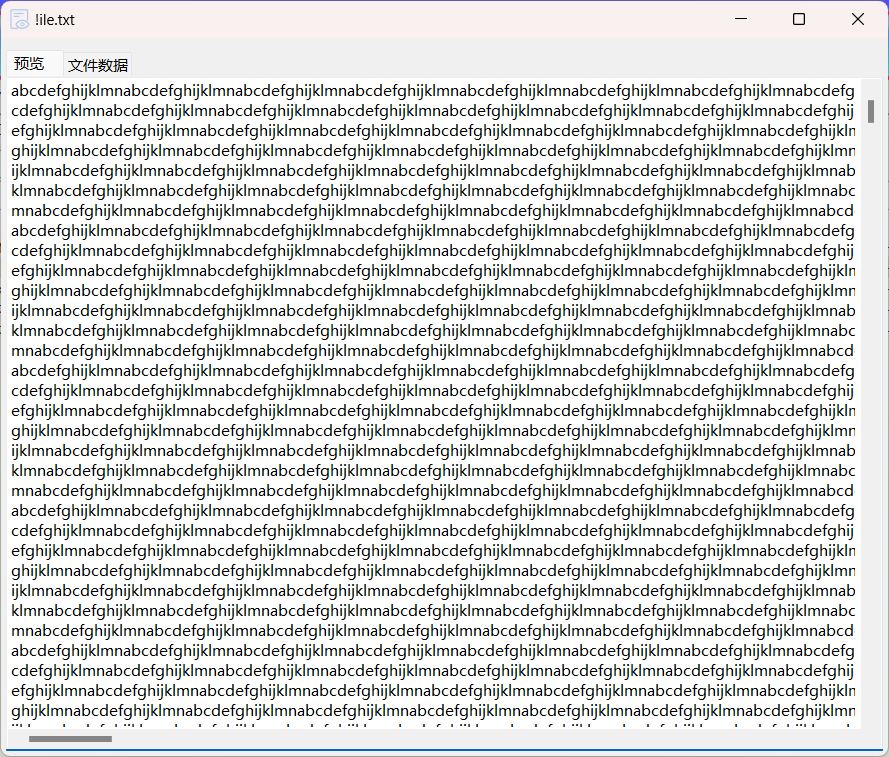


图 19 恢复后的文件

查看恢复后的831KB的测试文件，能够看到文件内容完好无损，但文件名有些许损坏，file变成了！ile，可能是其他内容进行了覆盖，但无伤大雅。

# 可引导盘制作与修改

## 3.1 将 U 盘做成可引导盘

首先，需要将 U 盘制作成一个可引导盘。这通常通过创建一个包含操作系统引导文件的分区来实现。首先使用工具 DiskGenius进行u盘的格式化并下载一个操作系统的镜像。

### 3.1.1 镜像文件下载

现代操作系统（尤其是Windows和Linux的安装镜像）大多数已经转向使用 GPT（GUID Partition Table）而非 MBR（Master Boot Record）作为磁盘分区的标准格式。但在某些情况下，MBR 格式依然存在，特别是在旧版操作系统或较旧的硬件上。

此处选择Ubuntu14.04镜像作为实验的测试镜像。选择速度较快的清华源[/ubuntu-releases/14.04.6/ 索引 |清华大学开源软件镜像站 |Tsinghua Open Source Mirror](https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/14.04.6/)



图 20 镜像文件选择

### 3.1.2 可引导盘制作

使用 Win32 Disk Imager 制作启动盘，Win32 Disk Imager 是一个轻量级的工具，主要用于将 ISO 文件 或 镜像文件 写入 U 盘，从而制作启动盘。可以用它来将 Windows 安装镜像或 Linux 系统镜像写入 U 盘，使其成为可引导盘。安装完成后，打开 Win32 Disk Imager。

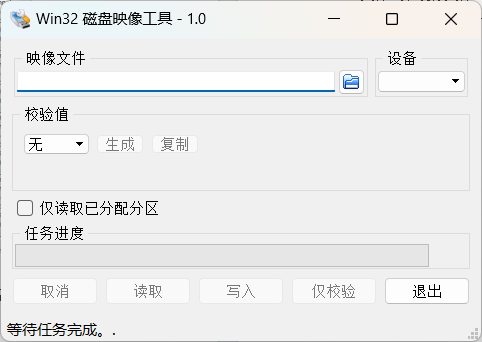


图 21 win32DiskImager 界面

选择 ISO 文件,点击 左上角的文件夹图标，选择下载的操作系统 ISO 文件。在工具界面的下拉框中，选择目标 U 盘设备。选择正确的盘符，避免误操作擦除其他硬盘数据。在 Write 按钮的右侧，确保选择 Write（写入）模式。如果之前已经有数据，程序会提醒清空 U 盘中的内容。点击 Write 按钮，工具将开始将 ISO 文件内容写入到 U 盘中。

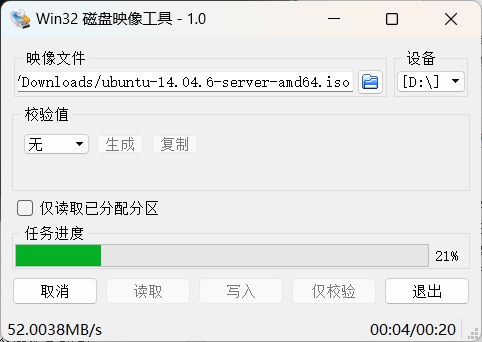
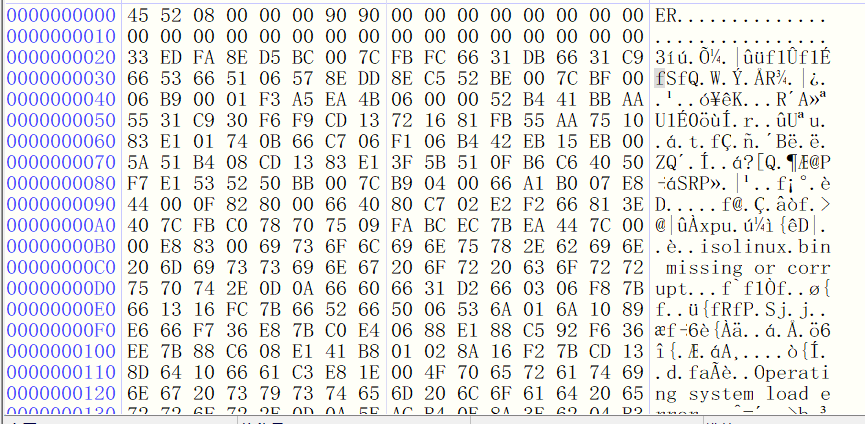


图 22 启动盘制作

写入完成后，会弹出提示框，确认写入已完成。此时，U 盘已经制作成了启动盘。

## 3.2读取主引导扇区与空盘进行对比

打开 DiskGenius，在扇区视图中，输入 0 以查看磁盘的第一个扇区（主引导扇区）。



可以看到其包含了关于磁盘布局、引导程序和分区的信息

1.引导程序（Bootloader）：位于MBR的前446字节，用于加载操作系统的引导程序。

2.分区表（Partition Table）：占用MBR的445至510字节，描述磁盘的分区信息。

3.分隔符（Magic number）：MBR的最后2个字节（0x55AA），表示MBR的结束标志。

与空盘MBR的对比：

空盘的MBR分区表通常是空的或没有实际的分区信息。而已分区的U盘的MBR中，分区表会显示有效的分区信息空盘没有分区信息，因此文件系统还未创建，也没有有效的文件或数据；已分区的U盘则有一个或多个有效的分区，且可以在其中存储文件。

· 分区表位置从字节位置446到510，提取出4个分区表项，分别进行分析。每个分区表项包含了分区的起始LBA（逻辑块地址）和分区大小等信息。

|  |  |  |
| --- | --- | --- |
| 字节位置 | 字段名 | 描述 |
| 0–3 | 引导标记（Boot Flag） | 1字节，0x80表示引导分区，0x00表示非引导分区 |
| 4-7 | 起始CHS（Cylinder-Head-Sector） | 3字节，表示分区起始的CHS地址（现在多用LBA） |
| 8–11 | 分区类型（Partition Type） | 1字节，定义分区的类型（例如，0x0C表示FAT32） |
| 12–15 | 结束CHS（Cylinder-Head-Sector） | 3字节，表示分区结束的CHS地址（不常用 |
| 16–19 | 起始LBA（Logical Block Address） | 4字节，分区起始位置的LBA（扇区地址） |
| 20–23 | 分区大小（Partition Size） | 4字节，分区的大小，以扇区为单位 |
| 24–31 | 保留字段 | 固定为0，保留字段 |

表 1 MBR分区表内容

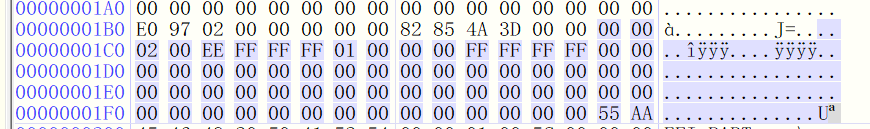


图 23 制作引导盘后分区表内实际内容

起始字节：00000200

这部分是引导标志（Boot Flag）。MBR分区表的引导标志是1个字节。如果为0x80，表示该分区是活动分区，意味着是引导分区；如果为0x00，则该分区不是活动分区。

在这个数据中，0x00表示该分区不是活动分区。

分区类型（Partition Type）：EE

这是分区类型标识符，通常是1字节。0xEE表示这是一个GPT保护分区（GPT Protective Partition）。在MBR磁盘上，通常会用0xEE表示GPT分区表存在，但不是一个实际的操作系统分区。 起始CHS（Cylinder-Head-Sector）：FFFFFF

这部分通常是3字节，表示分区的起始CHS（Cylinder-Head-Sector）地址，但在现代磁盘中，这些字段大多数是无效的，因为LBA（逻辑块地址）已取代它们的作用。这里的FFFFFF表示这个字段未使用（无效）。

结束CHS（Cylinder-Head-Sector）：01000000

这部分是3字节，表示分区的结束CHS地址，同样由于现代磁盘使用LBA，这部分也通常无效。0x01000000是一个默认值，与此分区表条目的分区结束位置无关。

起始LBA（Logical Block Address）：FFFFFFFF

这是4字节的起始LBA，表示分区的起始位置。0xFFFFFFFF表示无效或未设置，通常在一些情况下，0xFFFFFFFF表示该分区表项不可用或错误的LBA。

分区大小（Partition Size）：0000000000000000

这是4字节的分区大小，表示该分区的大小（以扇区为单位）。在此数据中，0x0000000000000000表示分区大小无效或未设置。

保留字段：

这部分是一些未使用的字节，通常是0x00填充的。在这个数据中，剩下的部分是空的，所有字节都是0x00。

## 3.3修改或者破坏主引导扇区并观察结果

3.3.1修改引导代码

修改MBR中的引导代码区域，可能会导致操作系统无法启动。引导代码通常位于MBR 的前446字节。使用十六进制编辑器打开MBR并进行修改。在某些系统（例如使用GRUB的Linux系统）中，环境块（environment block）包含引导加载程序的设置和环境变量。破坏MBR时可能同时破坏了引导配置文件或其引用。

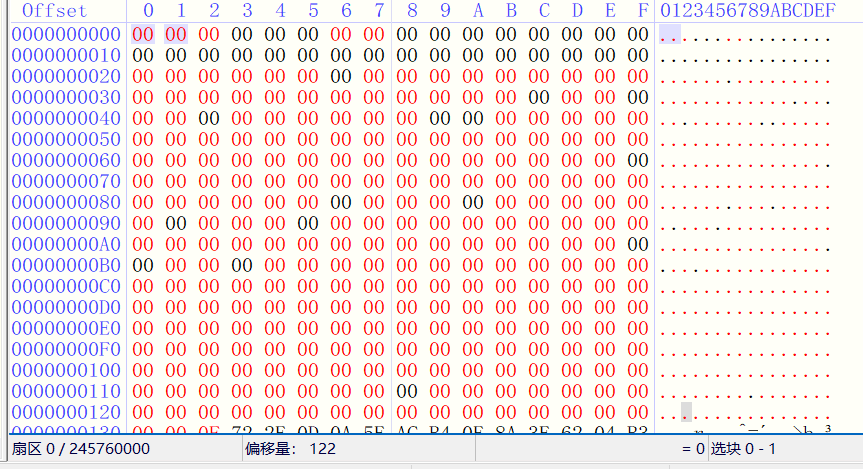


图 24 引导区置0

打开MBR的起始部分，修改引导代码区域（446字节），将它设置为0，模拟引 导代码 损坏。

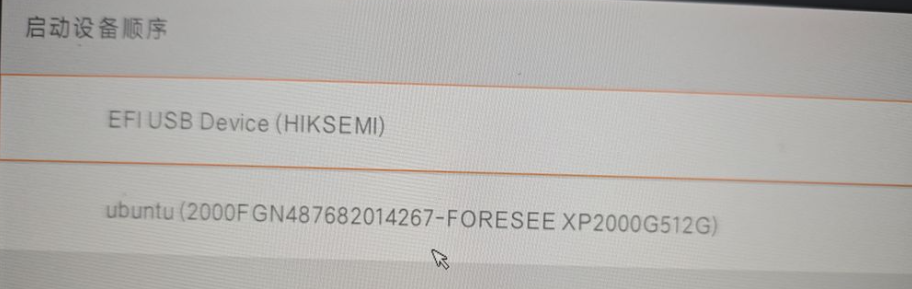


图 25 设置U盘启动

在破坏引导区之后，使用笔记本进入BIOS设置启动盘为U盘如上图所示：

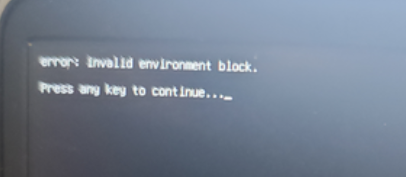


图 26 启动引导报错显示

屏幕上显示Invalid environment block：这通常表明引导程序正在尝试加载或访问一个环境变量，但该变量的 结构不正确或不存在。这是由于引导扇区被修改后，系统无法正确识别相关的引导参数，或者环境数据被损坏导致无法解析。

### 3.3.2更改分区表

更改分区表的条目，修改LBA地址、分区类型等。此处将分区表全部置零。

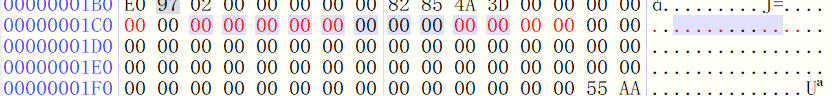


图 27 分区表置零

这会导致分区无法识别或无法启动。更改分区表条目。将某个分区的类型设置为无效的值，或 者修改分区的起始地址。

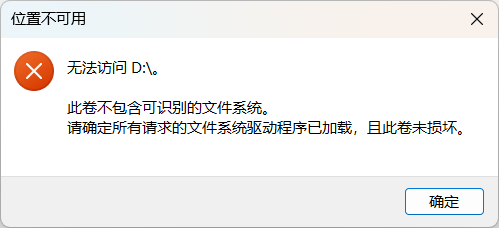


图 28 更改分区表结果

修改了分区表，操作系统无法识别已损坏的分区。操作系统认为该分区是不同类型的文件系统， 导致无法正确访问分区中的数据。

# 设计的U盘数据破环场景并恢复数据

## 4.1创建数据破坏环境

### 4.1.1格式化U盘：

使用DiskGenius工具将U盘分区并格式化为FAT32、NTFS或EXT4文件系统。这里使用FAT32文件系统。在DiskGenius界面中，找到U盘设备，右键选择“格式化”，选择文件系统为FAT32，点击“开始格式化”



图 29 U盘格式化

选择文件系统为FAT32，点击“开始格式化”，确认并等待格式化完成

### 4.1.2 模拟创建数据环境

在格式化后的U盘中创建几个文本文件（file1.txt、file2）。

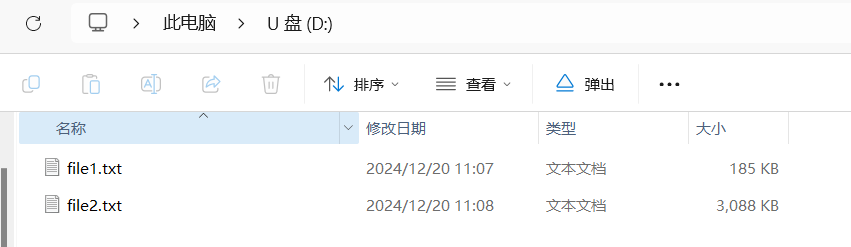


图 30 建立数据环境测试文件

并在其中输入一些内容

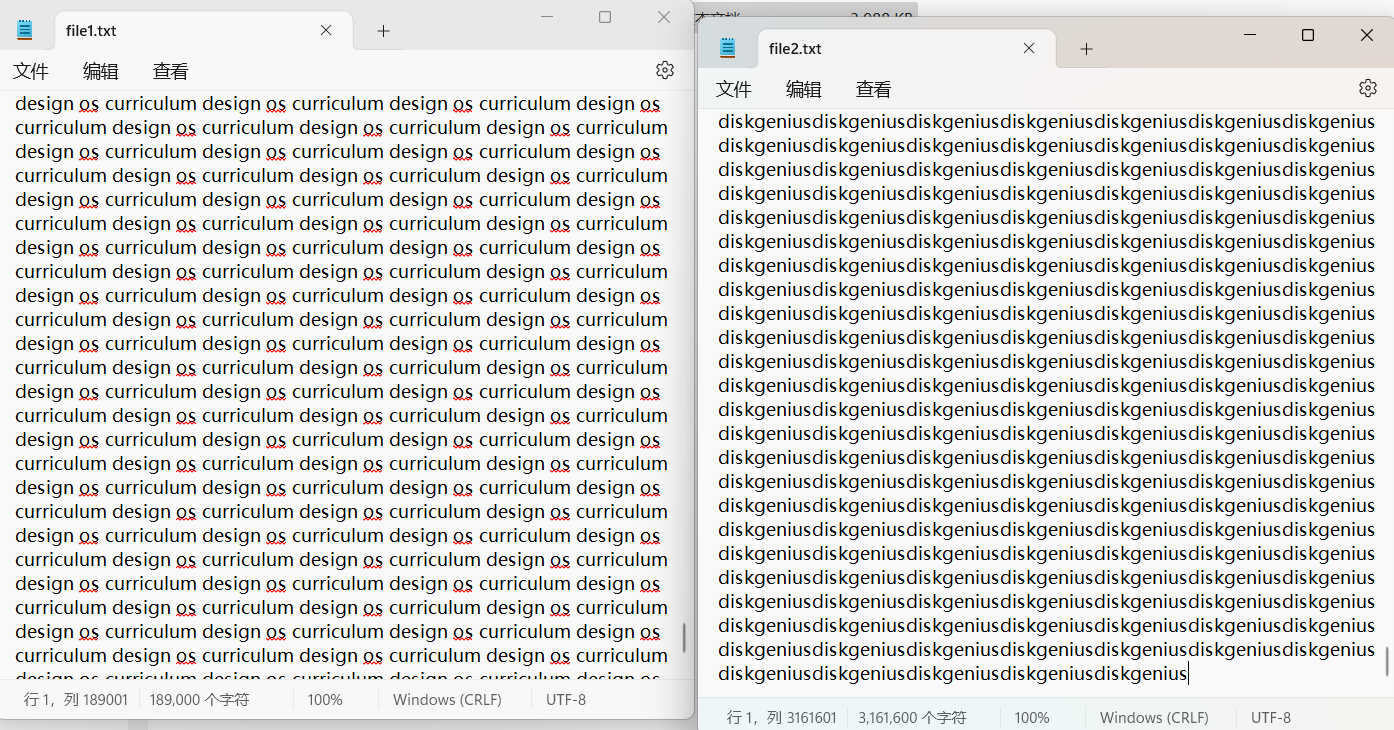


图 31 在测试文件中输入一定数据

### 4.1.3 模拟破坏数据环境

1.删除文件

在文件管理器中删除file1.txt和file2.txt。。

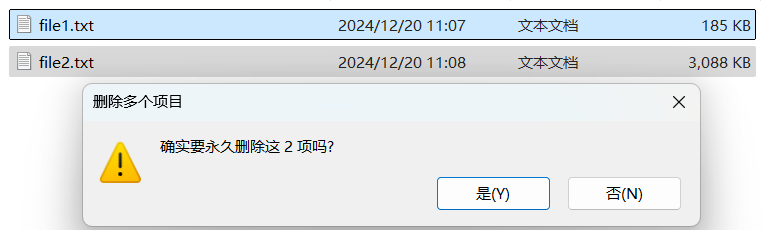


图 32 删除数据文件

此时文件在操作系统中已经删除，但磁盘上的数 据仍然存在，直到被新数据覆盖。

2.损坏U盘分区表

使用DiskGenius故意修改U盘的分区。

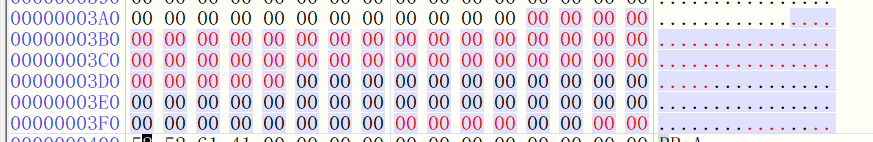


图 33 破坏U盘分区表

如图将分区表所有置0将其损坏。

3.破坏引导信息

这种破坏会导致整个磁盘变得不可用，即使在没有损坏数据的情况下，磁盘也无法被正确识别。使用磁盘编辑工具打开U盘的主引导扇区，修改或删除MBR

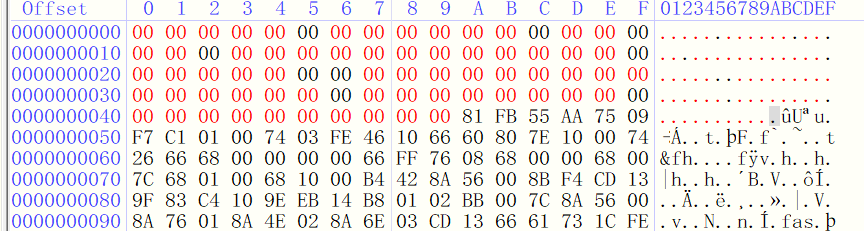


图 34 破坏引导信息

如图将MBR中引导信息全部置0，将其损坏。

1. 伪造磁盘序列号和硬件ID

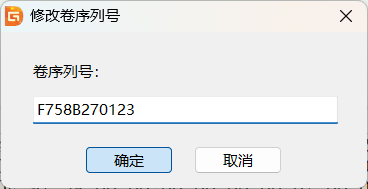


图 35 伪造序列号

修改U盘的序列号或硬件ID，伪造磁盘的身份。这将导致U盘被认为是一个新的磁盘，可能导致操作系统无法识别文件系统和数据。复杂性：这种操作会直接影响磁盘的硬件层面标识，通常需要使用低级别的工具进行操作。



图 36 破坏后U盘状态

如上图所示，U盘已经无法访问，且文件已被删除。

## 4.2已破坏数据恢复

### 4.2.1恢复分区

使用 DiskGenius 扫描并恢复分区。启动 DiskGenius，插入受损的U盘，在DiskGenius界面中，选择U盘磁盘。右键点击U盘磁盘，选择 “恢复分区”。扫描完成后，DiskGenius会列出所有找到的分区和文件。选择想要恢复的分区或文件。



图 37 恢复分区

### 4.2.2恢复文件内容

在DiskGenius界面中，选择U盘磁盘，并右键点击选择 “文件恢复”。在弹出的恢复界面中，选择扫描方式。



图 38 文件恢复

可以选择 “快速扫描”（适用于轻度删除）或 “深度扫描”（适用于数据丢失较久或文件系统损坏）。这里选择深度扫描。

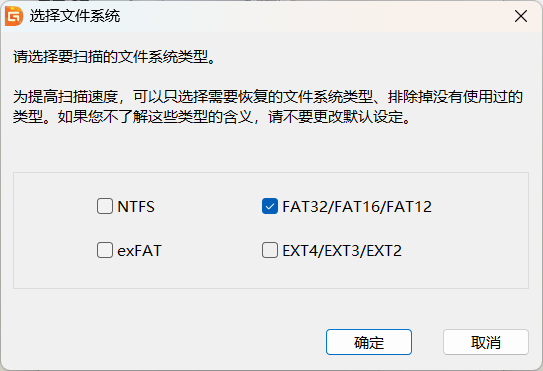


图 39 选择恢复文件类型

这里已经知道了需要恢复文件的类型，选择FAT32以加快速度。

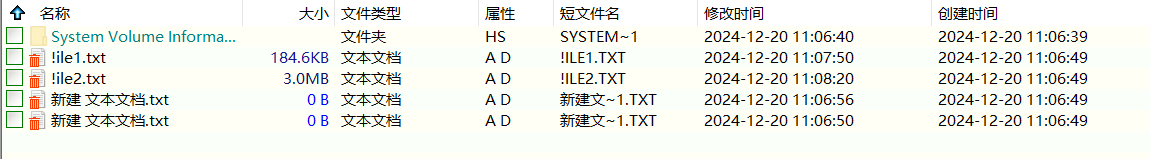


图 40 恢复后的文件

能够看到成功恢复了刚刚创建的两份文件，但其文件名有损坏，file变为!ile，可能是由于文件系统损坏或恢复过程中元数据丢失导致的。恢复工具只能恢复文件的实际数据内容，但无法正确恢复文件的名称和路径。

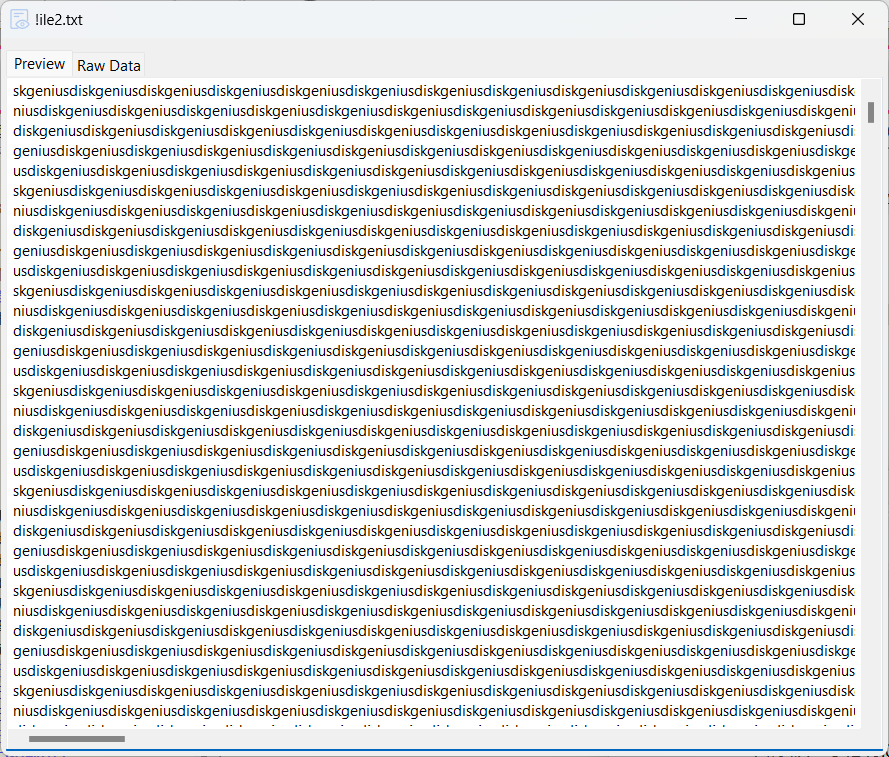
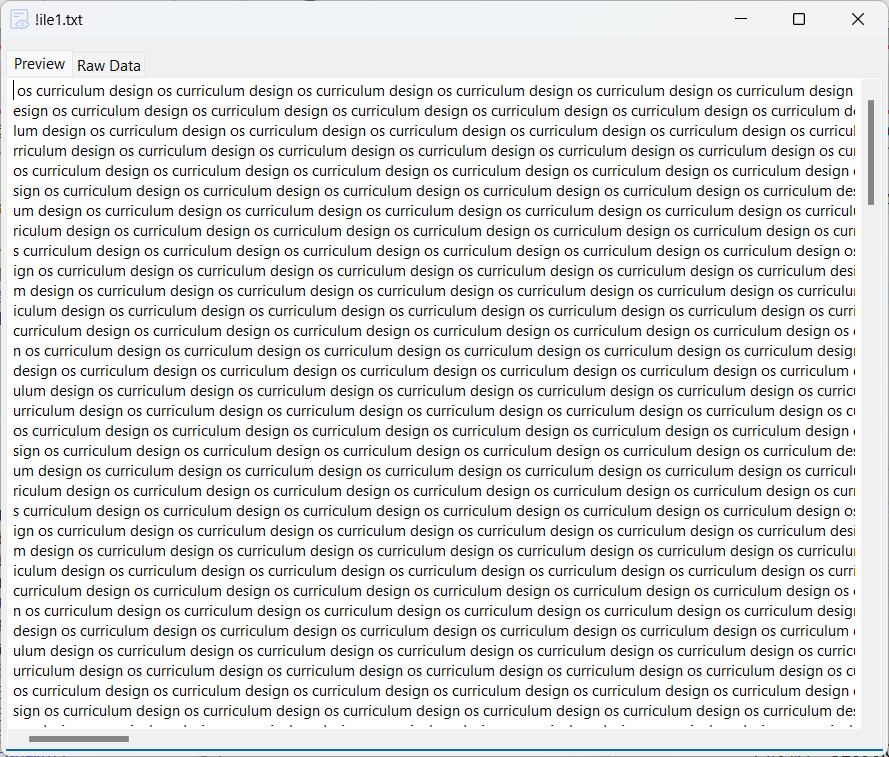


图 41 恢复数据内容

能够明显，数据恢复成功。原来的数据内容完好无损，仅是文件名有损坏。