

Trabajo Práctico - Sistema de Archivos FAT

1. Al montarlo ¿Para que se ha puesto umask=000 ?

Se usa *umask=000* para setear permisos de lectura, escritura y ejecución públicos de todos los *files* que se están montando.

2. Cargando el MBR

a) **Mostrando el MBR con el Hex Editor:** Muestre los primeros bytes y la tabla de particiones. ¿Cuántas particiones hay? Muestre claramente en qué lugar puede observarlo.

Podemos ver que la estructura del MBR permite hasta 4 particiones primarias:

Structure of a classical generic MBR				
Address		Description		Size (bytes)
Hex	Dec			
+000 _{hex}	+0	Bootstrap code area		446
+1BE _{hex}	+446	Partition entry №1	Partition table (for primary partitions)	16
+1CE _{hex}	+462	Partition entry №2		16
+1DE _{hex}	+478	Partition entry №3		16
+1EE _{hex}	+494	Partition entry №4		16
+1FE _{hex}	+510	55 _{hex}	Boot signature ^[a]	2
+1FF _{hex}	+511	AA _{hex}		
Total size: 446 + 4×16 + 2				512

Empezando la primera a partir del decimal 446 y cada una con una extensión de 16 bytes. Si utilizamos un editor hexadecimal, podemos ver que en los bytes 446 al 461 tenemos declarada una partición primaria. El resto de las particiones están en cero por lo que podemos asumir que están sin uso.

00000000	EB	3C	90	6D	6B	66	73	2E	66	61	74	00	02	04	01	00	02	00	02	00	08	F8	02	00	20	00	40	00	00	00	00	
0000001f	00	00	00	00	00	80	00	29	5F	05	C8	06	4E	4F	20	4E	41	4D	45	20	20	20	20	46	41	54	31	32	20	20	20	
0000003e	0E	1F	BE	5B	7C	AC	22	C0	74	0B	56	B4	0E	BB	07	00	CD	10	5E	EB	F0	32	E4	CD	16	CD	19	EB	FE	54	68	
0000005d	69	73	20	69	73	20	6E	6F	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	64	69	73	6B	2E	20	20	50	6C	65	
0000007c	61	73	65	20	69	6E	73	65	72	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	66	6C	6F	70	70	79	20	61	6E	
0000009b	64	0D	0A	70	72	65	73	73	20	61	6E	79	20	6B	65	79	20	74	6F	20	74	72	79	20	61	67	61	69	6E	20	2E	
000000ba	2E	2E	20	0D	0A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000d9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000f8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000117	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000136	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000155	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000174	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000193	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001b2	00	00	00	00	00	00	00	00	00	00	00	00	80	00	02	00	01	20	20	00	01	00	00	00	FF	07	00	00	00	00	00	00
000001d1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001f0	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	F8	FF	FF	00	F0	FF	06	F0	FF	07	80	00	09	F0	FF	00	
0000020f	FF	0F	00	00	00	00	FF	0F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000022e	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000024d	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000026c	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000028b	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002aa	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002c9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002e8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

b) Lea los datos del punto anterior utilizando código C y muestrelos por pantalla.

El código C por el cual leemos la tabla de particiones y mostramos en pantalla el contenido de cada una se encuentra en el archivo `read_mbr.c`. Los datos que muestra en pantalla son los siguientes:

```
Partition entry 0: First byte 80
  Comienzo de partición en CHS: 00:02:00
  Partition type 0x01
  Fin de partición en CHS: 00:20:20
  Dirección LBA relativa 0x00000001, de tamaño en sectores 2047
Partition entry 1: First byte 00
  Comienzo de partición en CHS: 00:00:00
  Partition type 0x00
  Fin de partición en CHS: 00:00:00
  Dirección LBA relativa 0x00000000, de tamaño en sectores 0
Partition entry 2: First byte 00
  Comienzo de partición en CHS: 00:00:00
  Partition type 0x00
  Fin de partición en CHS: 00:00:00
  Dirección LBA relativa 0x00000000, de tamaño en sectores 0
Partition entry 3: First byte 00
  Comienzo de partición en CHS: 00:00:00
  Partition type 0x00
  Fin de partición en CHS: 00:00:00
  Dirección LBA relativa 0x00000000, de tamaño en sectores 0
```

c) Muestre en el Hex Editor si la primer particion es booteable o no. ¿Lo es?

Para saber si una partición es booteable debemos observar el primer byte en la partition table, conocido como el boot flag. Si el byte es 0x80 significa que la partición es booteable, si es otro valor, no lo es.

Observando la partición con el editor hexadecimal podemos ver que el byte 0x1BE, es 0x80. Por lo cual, la primera partición es booteable.

[illegible]

d) Muestre, mediante un programa en C, para la primer partición: el flag de bootable, la dirección Cylinder-head-sector (chs), el tipo de partición y su tamaño en sectores.

El código C por el cual recolectamos la información requerida de la primer particion se encuentra en el archivo **read_fat.c** en el que recopilamos varios ejercicios. Se puede ver la línea del print de la siguiente imagen a partir de la línea 189 y el comienzo de la recopilación de datos de la primer particion a partir de la línea 183.

```
-----PARTITION-----
Booteable: 0x80
CHS: 0x7A748CC1
Type: 0x1
Size in sectors: 2047
```

3. Cargando la tabla de archivos

a) ¿Cuántos y cuáles archivos tiene el filesystem? Muestrelos con Bless y genere el código C para mostrarlos.

El filesystem originalmente tiene un total de dos archivos, el archivo hola.txt y el directorio mi_dir que contiene el archivo vacio.txt.


```
00000991 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000009b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000009cf 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000009ee 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000a0d 00 72 00 00 00 FF FF FF FF FF FF FF FF FF FF 4D 49 5F 44 49 52 20 20 20 20 20 10 .....
00000a2c 00 00 97 BA 7B 4A 7B 4A 00 00 97 BA 7B 4A 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000a4b 0F 00 A0 74 00 78 00 74 00 00 0F FF FF FF 00 00 FF FF FF FF 48 4F 4C 41 20 20 20 20 54 58 .....
00000a6a 54 20 00 64 F3 7C 7A 4A 7A 4A 00 00 F3 7C 7A 4A 05 00 3C 00 00 00 E5 2E 00 73 00 77 00 70 00 .....
00000a89 00 00 0F 00 A1 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000aa8 00 72 00 0F 00 A1 61 00 72 00 6F 00 6E 00 2E 00 74 00 00 00 78 00 74 00 E5 4F 52 52 41 52 7E .....
00000ac7 31 53 57 50 20 00 00 F3 B4 7B 4A 7B 4A 00 00 F3 B4 7B 4A 00 00 00 00 00 00 E5 2E 00 73 00 77 .....
00000ae6 00 78 00 00 00 0F A9 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000b05 6F 00 72 00 72 00 0F A9 61 00 72 00 6F 00 6E 00 2E 00 74 00 00 00 78 00 74 00 E5 4F 52 52 .....
00000b24 41 52 7E 31 53 57 58 20 00 00 F3 B4 7B 4A 7B 4A 00 00 F3 B4 7B 4A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
AR-1SWX .....{J{J.....{J.....
```

```
0000517f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000519e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000051bd 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000051dc 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000051fb 00 00 00 00 00 2E 20 20 20 20 20 20 20 20 20 20 10 00 00 91 BA 7B 4A 7B 4A 00 00 91 BA 7B 4A .....
0000521a 03 00 00 00 00 2E 2E 20 20 20 20 20 20 20 20 20 20 10 00 00 91 BA 7B 4A 7B 4A 00 00 91 BA 7B .....
00005239 4A 00 00 00 00 00 41 76 00 61 00 63 00 69 00 6F 00 0F 00 5B 2E 00 74 00 78 00 74 00 00 00 .....
00005258 FF FF 00 00 FF FF FF FF 56 41 43 49 4F 20 20 20 54 58 54 20 00 00 97 BA 7B 4A 7B 4A 00 00 97 .....
00005277 BA 7B 4A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
..{J.....{J.....{J.....
```

El código C que se encarga de mostrar los archivos se encuentra en el archivo **read_fat.c** en el que recopilamos varios ejercicios. Comienza a partir de la línea 52 en el método **print_all_files**. En la captura se ven otros archivos que agregamos manualmente para realizar otros ejercicios.

```
-----List Files-----
|- [MI_DIR      ]
|--- VACIO      .TXT
|--- [HOLAS     ]
|----- XELETED .TXT
|- HOLA        .TXT
|- SOURCE      .C
|- *DELETED*   ♦ORRAR~1.SWP
|- *DELETED*   ♦ORRAR~1.SWX
```

b) Montando el filesystem (mediante mount) cree un archivo en la carpeta root / y luego bórralo. Búsquelo por bless y muéstrelo en con el código generado previamente

```
root@osboxes:/mnt#  
root@osboxes:/mnt# touch borrado.txt  
root@osboxes:/mnt#  
root@osboxes:/mnt#  
root@osboxes:/mnt# rm borrado.txt  
root@osboxes:/mnt#
```

Podemos ver el archivo borrado.txt con el editor de hexadecimal

00000ac7	70 00 01 00 0f 00 5f 70 00 01 00 2e 00 74 00 70 00 74 00 00 00 00 ff ff 40 41 5f 50 41 50	p.d...p.d...l.x.c.....LA_PAP
00000ae6	41 20 54 58 54 20 00 64 8e 70 94 4e 94 4e 00 00 8e 70 94 4e 00 00 12 00 00 00 e5 62 00 6f 00	A TXT .d.p.N...p.N.....b.o.
00000b05	72 00 72 00 61 00 0f 00 60 64 00 6f 00 2e 00 74 00 78 00 74 00 00 00 00 00 ff ff e5 4f 52 52	r.r.a...`d.o...t.x.t.....ORR
00000b24	41 44 4f 20 54 58 54 20 00 00 74 73 94 4e 94 4e 00 00 74 73 94 4e 00 00 00 00 00 e5 41 5f	ADO TXT ..ts.N.N..ts.N.....A
00000b43	50 41 50 7e 31 53 57 50 20 00 00 88 70 94 4e 94 4e 00 00 88 70 94 4e 00 00 00 00 00 00 00 00	PAP~1SWP ...p.N.N...p.N.....
AAAA0b67	aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa

Si corremos el código generado previamente podemos ver lo siguiente:

```
-----List Files-----  
|- [MI_DIR      ]  
|--- VACIO      .TXT  
|--- [HOLAS     ]  
|----- XELETED .TXT  
|- HOLA        .TXT  
|- SOURCE      .C  
|- LA_PAPA     .TXT  
|- *DELETED*   ♦ORRADO .TXT  
|- *DELETED*   ♦A_PAP~1.SWP
```

c) Muestre mediante `ls` el archivo que ha sido borrado. Explique cómo lo ha visto. Genere código C para mostrarlos.

El archivo fue mostrado mediante un editor de hexadecimal en el punto anterior al igual que la descripción del código C para mostrarlo. La razón por la que podemos seguir viendo el archivo en el filesystem es debido a que los archivos borrados en fat12 no son borrados realmente, esto se explica en el siguiente punto.

d) ¿Qué puede decir acerca del recupero de archivos?

En el sistema de archivos FAT al eliminar un archivo no lo estamos haciendo realmente, el contenido sigue existiendo en el filesystem, lo único que se hace es marcar la entrada del directorio como borrada. Podemos identificar los archivos borrados observando el primer byte de los 8 bytes que representan el filename del archivo, si el primer byte es 0xE5 significa que el archivo fue borrado. Teniendo en cuenta esto, para recuperar un archivo debemos modificar el primer byte del filename. Esto funcionará solo si no se creó previamente un nuevo archivo que apunte hacia el mismo cluster, en ese caso el contenido del cluster fue modificado y los datos del archivo no pueden recuperarse.

[illegible]

4. Leyendo archivos

a) Montando el filesystem cree un archivo llamado lapapa.txt y póngale algún texto como contenido. Hágalo en la carpeta root /. . Busquelo por bless y muéstrelo con el código generado previamente.

Creamos el archivo `lapapa.txt` con el contenido “Hola! Soy la papa”

```
root@osboxes:/mnt# ls
hola.txt  la_papa.txt  mi_dir  source.c
root@osboxes:/mnt# cat la_papa.txt
hola! Soy la papa
```


Vemos su contenido usando un editor hexadecimal:

```
00000a89 63 00 0F 00 72 65 00 2E 00 63 00 00 00 FF FF FF FF 00 00 FF FF FF FF 53 4F 55 52 43 45 20 20 c...re...c.....SOURCE
00000aa8 43 20 20 20 00 64 A1 98 87 4E 87 4E 00 00 A1 98 87 4E 04 00 B8 22 00 00 41 6C 00 61 00 5F 00 C...d...N.N....N...".Al.a...
00000ac7 70 00 61 00 0F 00 5F 70 00 61 00 2E 00 74 00 78 00 74 00 00 00 00 00 FF FF 4C 41 5F 50 41 50 p.a...p.a...t.x.t.....LA_PAP
00000ae6 41 20 54 58 54 20 00 64 8E 70 94 4E 94 4E 00 00 8E 70 94 4E 00 12 00 00 00 E5 73 00 77 00 A TXT .d.p.N.N...p.N.....s.w.
00000b05 70 00 00 00 FF FF 0F 00 70 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF E5 2E 00 6C p.....p.....1
00000b24 00 61 00 5F 00 70 00 0F 00 70 61 00 70 00 61 00 2E 00 74 00 78 00 00 00 74 00 2E 00 E5 41 5F .a...p...pa.p.a...t.x.t....A_
00000b43 50 41 50 7E 31 53 57 50 20 00 00 88 70 94 4E 94 4E 00 00 88 70 94 4E 00 00 00 00 00 00 00 PAP~1SWP ...p.N.N...p.N.....

0000a988 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000a9a7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000a9c6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000a9e5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 68 6F 6C 61 .....hola
0000aa04 21 20 53 6F 79 20 6C 61 20 70 61 70 61 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ! Soy la papa.....
0000aa00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

Lo vemos por el código utilizado previamente:

```
-----List Files-----
|- [MI_DIR      ]
|--- VACIO      .TXT
|--- [HOLAS     ]
|----- XELETED .TXT
|- HOLA        .TXT
|- SOURCE      .C
|- LA_PAPA     .TXT
|- *DELETED*   A_PAP~1.SWP
```

b) Muestre, mediante el hex editor y mediante código C lo que hay en el archivo no borrado.

Para este punto decidimos crear un archivo llamado xeleted.txt en el directorio mi_dir/holas. Este archivo tiene como contenido un lorem ipsum.

```
root@osboxes:/mnt/mi_dir/holas# ls
XELETED.TXT
```

```
root@osboxes:/mnt/mi_dir/holas# cat xeleted.txt
```

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras efficitur mauris a
c justo blandit rhoncus. Nullam ligula magna, fermentum et nunc et, commodo effi
citur augue. Nullam ligula ex, iaculis dignissim velit vel, fringilla aliquet le
ctus. Fusce sit amet fermentum ipsum. Donec facilisis lorem non dui pretium, eu
eleifend erat volutpat. Duis a elit quis eros euismod volutpat quis eget lacus.
Curabitur tempor mi in massa pulvinar venenatis. Proin semper fermentum mi, a co
nvallis turpis ultricies non. Sed sed blandit nisi, a porttitor urna. Praesent t
ristique lorem quam, tristique ornare nibh hendrerit id. Donec eleifend eleifend
dui, at mattis purus elementum aliquam. Ut imperdiet commodo aliquam. Donec mol
estie commodo mauris, et consequat eros auctor non. Nullam dolor tortor, rhoncus
vel tincidunt id, placerat in tortor. Mauris interdum ligula vel leo dignissim
placerat.
```

```
Suspendisse at magna mattis, dignissim ex vel, faucibus purus. In consequat arcu
```

Mostramos su contenido utilizando el código `read_fat.c` en el que recopilamos varios ejercicios, el print de lo mostrado en pantalla se encuentra en la línea 127.

```
----- Content of XELETED .TXT -----
```

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras efficitur mauris a
c justo blandit rhoncus. Nullam ligula magna, fermentum et nunc et, commodo effi
citur augue. Nullam ligula ex, iaculis dignissim velit vel, fringilla aliquet le
ctus. Fusce sit amet fermentum ipsum. Donec facilisis lorem non dui pretium, eu
eleifend erat volutpat. Duis a elit quis eros euismod volutpat quis eget lacus.
Curabitur tempor mi in massa pulvinar venenatis. Proin semper fermentum mi, a co
nvallis turpis ultricies non. Sed sed blandit nisi, a porttitor urna. Praesent t
ristique lorem quam, tristique ornare nibh hendrerit id. Donec eleifend eleifend
dui, at mattis purus elementum aliquam. Ut imperdiet commodo aliquam. Donec mol
estie commodo mauris, et consequat eros auctor non. Nullam dolor tortor, rhoncus
vel tincidunt id, placerat in tortor. Mauris interdum ligula vel leo dignissim
placerat.
```

```
Suspendisse at magna mattis, dignissim ex vel, faucibus purus. In consequat arcu
non velit commodo, id hendrerit quam imperdiet nullam.
```

```
----- End of Content -----
```

c) Cree código C para que dado un archivo (o una parte), lo busque y si lo encuentra y el mismo se encuentra borrado, lo recupere.

Borramos el archivo `xeleted.txt`

```
root@osboxes:/mnt/mi_dir/holas# ls
XELETED.TXT
root@osboxes:/mnt/mi_dir/holas# rm xeleted.txt
root@osboxes:/mnt/mi_dir/holas# ls
root@osboxes:/mnt/mi_dir/holas#
```


Ejecutamos el archivo **read_fat.c** y vemos lo siguiente en pantalla:

```
----- Content of ✖ELETED .TXT -----  
  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras efficitur mauris a  
c justo blandit rhoncus. Nullam ligula magna, fermentum et nunc et, commodo effi  
citur augue. Nullam ligula ex, iaculis dignissim velit vel, fringilla aliquet le  
ctus. Fusce sit amet fermentum ipsum. Donec facilisis lorem non dui pretium, eu  
eleifend erat volutpat. Duis a elit quis eros euismod volutpat quis eget lacus.  
Curabitur tempor mi in massa pulvinar venenatis. Proin semper fermentum mi, a co  
nvallis turpis ultricies non. Sed sed blandit nisi, a porttitor urna. Praesent t  
ristique lorem quam, tristique ornare nibh hendrerit id. Donec eleifend eleifend  
dui, at mattis purus elementum aliquam. Ut imperdiet commodo aliquam. Donec mol  
estie commodo mauris, et consequat eros auctor non. Nullam dolor tortor, rhoncus  
vel tincidunt id, placerat in tortor. Mauris interdum ligula vel leo dignissim  
placerat.  
  
Suspendisse at magna mattis, dignissim ex vel, faucibus purus. In consequat arcu  
non velit commodo, id hendrerit quam imperdiet nullam.  
----- End of Content -----  
Was deleted  
Restored as XELETED .TXT
```

El código que restaura el archivo xeleated.txt se encuentra en **read_fat.c**, línea 96, en el método `find_and_restore`. La forma de encontrar el archivo borrado xeleated.txt que utilizamos fue buscando una porción del lorem ipsum.