CS-458: Computer & Network Forensics

Winter 2021

Midterm Exam

Name: Colin Quinn

Total Points: 100

**Instructions (Read them!):**

1. The final is a take-home exam.
2. The final exam will be due on **2/20/2021 @ 11:00 PM**.
3. The exam has a total of 8 questions. Total of 13 pages
4. Read each question carefully. Complete **all** problems.
5. The answers can be typed or handwritten. Your responses should be legible. They won't be graded if I can't read them.
6. Upload your answers to the blackboard dropbox in a single PDF file.
7. Show your work and **state your assumptions** clearly, if any; partial credit may be awarded. Assumptions must be valid.
8. **If you do not understand the question, please clarify the question with the instructor.**
9. If any student is caught cheating, he/she will be awarded zero, and action will be taken according to Kettering University Code of Student Conduct

|  |  |  |
| --- | --- | --- |
| **Section** |  | **Points** |
|  |  |  |
| NTFS Forensics | 30 | |
|  |  |  |
| LINUX Forensics | 35 | |
|  |  |  |
| FAT32 Forensics | 30 | |
|  |  |  |
| FREE Points :-) | 5 | |
|  |  |  |
|  |  |  |
| Total |  | 100 |
|  |  |  |

All the best

**NTFS Forensics**

**1.** Refer to the Master boot record of a Hard Drive. Given that the sector size is 0x200 bytes.

**[8 points]**

**Offset** **|** **00 01 02 03 04 05 06 07** **08 09 0A 0B 0C 0D 0E 0F |** **ASCII**

-----------------------------------------------------------------------------------

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 000000000 | | | 33 | C0 | 8E | D0 | BC 00 7C 8E | | | C0 | 8E | D8 | BE 00 7C | | | BF 00 | | | 3..... | |...... | |.. |
| 000000010 | | | 06 | B9 | 00 | 02 | FC F3 A4 50 | | | 68 | 1C | 06 | CB FB B9 04 | | | | 00 | | ....... | Ph....... |  |
| 000000020 | | | BD BE 07 | | | 80 7E 00 | | 00 | 7C | 0B | 0F | 85 | 0E | 01 83 | | C5 | 10 | | ....~..|........ | |  |
| 000000030 | | | E2 | F1 | CD | 18 | 88 56 | 00 | 55 | C6 | 46 | 11 | 05 | C6 46 | | 10 | 00 | | .....V.U.F... | | F.. |
| **. . . . . . . . . . . . . . . . BOOT CODE . . . . . . . . . . . . . . . . .** | | | | | | | | | | | | | | | | | | | |
| 000000180 | | | 20 | 6C | 6F | 61 | 64 69 | 6E | 67 | 20 | 6F | 70 | 65 | 72 61 | | 74 | 69 | | loading operati | | |
| 000000190 | | | 6E | 67 | 20 | 73 | 79 73 | 74 | 65 | 6D | 00 | 4D | 69 | 73 73 | | 69 | 6E | | ng system.Missin | | |
| 0000001A0 | | | 67 | 20 | 6F | 70 | 65 72 | 61 | 74 | 69 6E 67 20 73 | | | | | 79 73 74 | | | | g operating syst | | |
| 0000001B0 | | | 65 | 6D | 00 | 00 | 00 63 | 7B | 9A | 1A | 46 | 36 | F9 | 00 00 | | 80 | 20 | | em... | c{..F6.... |  |
| 0000001C0 | | | 21 | 00 | 07 | DD 1E 3F 00 | | | 08 | 10 00 00 A0 0F | | | | | 00 00 DD | | | | !.... | ?.......... |  |
| 0000001D0 | | | 1F | 3F | 83 | FE FF FF 00 A8 | | | | 0D 10 00 50 B0 | | | | | 03 00 00 | | | | .?......... |  | P.... |
| 0000001E0 | | | 00 | 00 | 00 | 00 | 00 00 | 00 00 | | 00 00 00 00 00 | | | | | 00 80 EE | | | | !.... | ?.......... |  |
| 0000001D0 | | | 6F | 7F | AB FE FF FF 0F 99 | | | | | 9F 00 00 50 B0 | | | | | 03 55 AA | | | | .............. |  | U. |

Based on the record shown above fill in the following table.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Partition** |  | **Physical Location** | |  | **Active** |  |  |
|  |  |  |  |  |  |
| **Type** | **in dec** |  |  | **in hex** | **Yes/No** |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| NTFS, HPFS, |  |  | Start sector | = 0x100800 | Yes |  |  |
| exFat | 1,050,624 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Linux | 269,330,432 |  | Start sector | = 0x100DA800 | No |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Unused | 0 |  | Start sector | = 0x00000000 | No |  |  |
|  |  |  |  |  |  |  |  |
| Mac OSX boot | 10,459,407 |  | Start sector | = 0x9F990F | Yes |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  | |  |  |  |
|  |  |  | Rough work available in the next page | | | |  |



2



Rough work if needed

~~Sector size = 0x200 = 512 bytes~~

~~Bytes per sector = 0xBE~~

~~Sectors per cluster = 0x7C~~

~~LCN of $MFT: 0x550056568818CDF1E2~~

Answer derived from partition table breakdown on slide 4. Relative start sector is the little-endian form of the 4 bytes starting at 0x1C6, 0x1D6… I feel like there is more math to do for this but cannot find notes for it.

|  |  |
| --- | --- |
| **2.** Refer to the NTFS boot record. | **[7 points]** |
|  |  |
| Offset | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F | | ASCII |

---------------------------------------------------------------------------------

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 00000000 | | | EB 52 90 | | | 4E | 54 | 46 | 53 | 20 | 20 | 20 | 20 | 00 | 02 | 20 | 00 | 00 | | | .R.NTFS | .. .. |
| 00000010 | | | 00 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | 3F | 00 | FF 00 | | 80 | 00 | 00 | 00 | | | ........ | ?....... |
| 00000020 | | | 00 | 00 | 00 | 00 | 80 | 00 | 80 | 00 | FF E7 0F 00 | | | | 00 | 00 | 00 | 00 | | | ................ |  |
| 00000030 | | | 6A | 2A | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | | j\*.............. | |  |
| 00000040 | | | F6 | 00 | 00 | 00 | F4 | 00 | 00 | 00 | 5F | 55 | F4 | 8C | 65 | F4 | 8C | 82 | | | ........ | \_U..e... |
| 00000050 | | | 00 | 00 | 00 | 00 | FA 33 C0 | | | 8E | D0 | BC 00 7C | | | FB 68 C0 | | | 07 | | .....3..... | | |.h.. |
| 00000060 | | | 1F | 1E | 68 | 66 | 00 | CB 88 | | 16 | 0E | 00 | 66 | 81 | 3E | 03 | 00 | 4E | | | ..hf...... | f.>..N |
| 00000070 | | | 54 | 46 | 53 | 75 | 15 | B4 | 41 | BB | AA 55 CD 13 | | | | 72 | 0C | 81 | FB | | | TFSu..A..U..r... | |
| 00000080 | | | 55 | AA 75 | | 06 | F7 | C1 | 01 | 00 | 75 | 03 | E9 | DD | 00 | 1E | 83 | EC | U.u..... | | | u....... |
| **. . . . .** | **. . . . . .** | | | | **. . . . . .** | | | | **. BOOT CODE . . . . . . . . . . . . . . . . . .** | | | | | | | | | | | |
| 000001F0 | | | 00 | 00 | 00 | 00 | 00 | 00 | 8A | 01 | A7 | 01 | BF 01 | | 00 | 00 | 55 | AA | | | .............. | U. |



3



Based on the record shown above answer the following questions

Bytes per sector = 0x200 = 512 bytes

a) Number of sectors per cluster in decimal.

Sectors per cluster = 0x02 = 2 sectors.

b) Size of each cluster in bytes (in dec)

Cluster size = 0x02 \* 0x200 = 0x400 = 1024 bytes

c) Physical location of $MFT record (in hex).

LCN of $MFT = 0x2A6A

Location of $MFT = 0x2A6A \* 0x400 = 0xA9A800

**3.** The following is the data attribute (**0x80**) of one of the user file record of the NTFS file

|  |  |
| --- | --- |
| system shown above. | **[15 points]** |
|  |  |
| **Offset | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F |** | **ASCII** |

---------------------------------------------------------------------------------

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0A9B3110 | | |  |  |  |  |  |  |  | 80 | 00 | 00 | 00 | 48 | 00 00 | 00 | | | ....H ... |
| 0A9B3120 | | | 01 | 00 | 00 | 00 | 00 | 00 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 00 | 00 | | | ................ |
| 0A9B3130 | | | 8F | 0D | 00 | 00 | 00 | 00 00 | 00 | 40 | 00 | 00 | 00 | 00 | 00 00 00 | ........ ....... | | | @ |
| 0A9B3140 | | | 00 | 00 | 64 | 03 | 00 | 00 00 | 00 | 00 | D6 | E3 | 89 | 2A | 00 00 | 00 | | ............... |  |
| 0A9B3150 | | | 00 | D6 | E3 | 89 | 2A | 00 00 | 00 | 32 | 90 | 0D | AC 0B FF 32 00 | ....... ....... | | | | | " |
| 0A9B3160 | | | DD AD EF BE 00 00 00 00 | | | | | | | FF FF FF FF FF 00 00 | | | | | | 00 | | ............... |  |



4



a) Determine if the file is a resident or a non-resident

Non-resident, the size on disk is greater than the file size and offset 0x0A9B3120 is set to 0x01.

1. Provide the physical location of the data and the size of the data. The data is divided into chunks, identify the LCN portion and VCN portion and also provide the size of each chunk and provide the command to extract each portion. Show all the steps.

|  |  |  |  |
| --- | --- | --- | --- |
| i) | LCN | LCN: 32 | 90 | 0D | AC | 0B | FF |  |
|  |  |  |

Cluster runs: 0xAC0D90

First Cluster Location: 0xFF0B

Decimal: 65,291

Byte offset: 267,431,936

Cluster location = 0xFF0B000

1. VCN

Assuming 0x00 means that we take the bytes available, in this case they are: DD AD EF BE

This assumption could be wrong. I couldn't find any notes on this case as we use the first byte to determine the run length and offset.

VCN: 00 | DD | AD | EF | BE

Cluster Runs: 0xADDD

First cluster location: 0xBEEF

Decimal: 48,879

VCN offset = 48,879 + 65,291 = 114,170

Byte offset: 467,640,320

Cluster Location = 0x1BDFA000

empty space in the next page



5



**LINUX Forensics**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **4.** Given below is the long listing of a directory named | | | | | | “**foobar”**. Refer to the output below | | |
| and answer the questions: | | | |  |  |  |  | **[15 Points]** |
| **student@autobot:~/foobar$ ls** | | | | **-ila** |  |  |  |  |
| total 16 | |  |  |  |  |  |  |  |
| 296539 | drwxrwxr-x | 2 | student student 4096 | | | Feb 11 16:40 . | |  |
| 311299 | drwxr-xr-x 27 | | student student 4096 | | | Feb 11 16:35 .. | |  |
| 296538 | -rw-rw-r-- 2 | | student student | | 21 | Feb 11 16:02 foo | |  |
| 296538 | -rw-rw-r-- 2 | | student student | | 21 | Feb 11 16:02 link1 | |  |
| 296541 | lrwxrwxrwx | 2 | student student | | 5 | Feb 11 16:03 link2 | | -> link1 |
| 296541 | lrwxrwxrwx | 2 | student student | | 5 | Feb 11 16:03 link3 | | -> link1 |
| 296542 | lrwxrwxrwx | 1 | student student | | 5 | Feb 11 | 16:35 link4 | -> link3 |
| 296543 | lrwxrwxrwx | 1 | student student | | 3 | Feb 11 | 16:39 link5 | -> foo |
| 296566 | lrwxrwxrwx | 1 | student student | | 5 | Feb 11 | 16:40 link6 | -> link5 |
| 296566 | lrwxrwxrwx | 2 | student student | | 5 | Feb 11 | 16:40 link7 | -> link5 |



6



1. Identify all the hardlinks. You answer should be of the form (‘a’ is a hardlink to file ‘b’ (or) link ‘c’; where a, b and c are the names of links/files)

link1 is a hardlink to foo link3 is a hardlink to link1 link6 is a hardlink to link5 link7 is a hardlink to link5

1. Identify all the symbolic links. You answer should be of the form (‘a’ is a symbolic link to file ‘b’ (or) link ‘c’; where a, b and c are the names of links/files)

link2 is a softlink to link1 link4 is a softlink to link3 link5 is a softlink to foo

Softlinks are determined by having unique inode values.

c) What happens when the file **“foo”** is renamed to “bar”.

Nothing should change. The only direct link to "foo" is the softlink link5. Though link6 and link7 are hardlinks to link5, the inode to link5 will not be impacted.

d) What happens when **“link1”** is deleted

If link1 is deleted, the link2, link3, and link4 will also lose their connection to "foo".



7



**5.** Given below is the command to check the number of inodes used and the number of inodes available. Assume there is a file called **“temp.txt”** present. Refer to the output below and

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| answer the questions: |  |  |  |  | **[5 Points]** |
| **student@autobot:~$ df** | | **-i /dev/sdb1** | |  |  |
| Filesystem | Inodes | IUsed | IFree | IUse% | Mounted on |
| udev | 10000 | 9900 | 100 | 99% | /dev |

a) How many new hard links can be created to the file **“temp.txt”**

The limit to hardlinks is determined by the OS since they do not require a unique inode value

b) How many new symbolic links can be created to the file **“temp.txt”**

Assuming IFree is the amount of free inodes, there would be 100 possible symbolic links to temp.txt

c) How many new copies of **“temp.txt”** can be created***.***

Based on the same logic that a new file has a unique inode, there would be 100 possible copies of temp.txt

**6.** The following is an output of an ext4 directory entry named **“dir”**. Based on the contents

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| of the directory, fill in the table. Include the | | | | | | | | **“.”** and | | **“..** | **”** in the table. | | | **[15 Points]** |
|  | 00 | 01 02 | | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | <--- | organization |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00000000 | 0C | 00 | 00 | 00 | 0C | 00 | 01 | 02 | 2E | 00 | 00 | 00 | ............ |  |
| 0000000C | 02 | 00 | 00 | 00 | 0C | 00 | 02 | 02 | 2E | 2E | 00 | 00 | ............ |  |
| 00000018 | 0E | 00 | 00 | 00 | 10 | 00 | 08 | 01 | 74 | 65 | 73 | 74 | ........ | test |
| 00000024 | 66 | 69 | 6C | 65 | 0F | 00 | 00 | 00 | 0C | 00 | 04 | 02 | file........ |  |
| 00000030 | 74 | 65 | 6D | 70 | 10 | 00 | 00 | 00 | 10 | 00 | 05 | 07 | temp........ |  |
| 0000003C | 6C | 69 | 6E | 6B | 31 | 00 | 00 | 00 | 0E | 00 | 00 | 00 | link1....... | |
| 00000048 | 10 | 00 | 05 | 01 | 6C | 69 | 6E | 6B | 32 | 00 | 00 | 00 | ....link2... | |
| 00000054 | 11 | 00 | 00 | 00 | 10 | 00 | 05 | 07 | 6C | 69 | 6E | 6B | ........ | link |
| 00000060 | 33 | 00 | 00 | 00 | 12 | 00 | 00 | 00 | 9C | 03 | 05 | 07 | 3........... |  |
| 0000006C | 6C | 69 | 6E | 6B | 34 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | link4....... | |
| 00000078 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ............ |  |
| 00000084 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ............ |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



8



**Note: If a file is a link identify to which file is it linking to**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **offset** | **Inode\_number** | | **Rec** | **File type** |  |  |  |
| **length** |  |  |  |
|  |  |  |  | **Linked to** | **File Name** |  |
|  |  |  |  |  |  |
| (in hex) | (in hex) | (in dec) | (in dec) | Sym/hard |  |
|  |  |  |
| /file/dir |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 0x00 | 0xC |  | 12 | Directory | Not linked | . dot |  |
| 12 | Unknown |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 0x0C | 0x2 | 2 | 12 | Directory | Not linked | .. (dot dot) |  |
| 0x18 | 0xE | 14 | 16 | Regular File | Not linked | testfile |  |
| 0x28 | 0xF | 15 | 12 | Directory | Not linked | temp |  |
| 0x34 | 0x10 | 16 | 16 | Sym link |  | link1 |  |
| 0x44 | 0xE | 14 | 16 | Hard link | testfile | link2 |  |
| 0x54 | 0x11 | 17 | 16 | Sym link |  | link3 |  |
| 0x64 | 0x12 | 18 | 924 | Sym link |  | link4 |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

I don’t remember going over how to determine what symbolic links are attached to when given the hex dump.

**FAT Forensics**

|  |  |  |
| --- | --- | --- |
| **7.** Refer to the following FAT32 boot record and answer the following questions | | **[7 Points]** |
|  |  |  |
| **Offset | 00 01 02 03 04 05 06 07** | **08 09 0A 0B 0C 0D 0E 0F |** | **ASCII** |

---------------------------------------------------------------------------------

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 00010000 | | | EB 58 90 4D 53 | | | | 44 | 4F | 53 | 35 | 2E | 30 | 00 | 02 | 01 6E | 10 | | .X.MSDOS5.0..... | |
| 00010010 | | | 02 00 | 00 | 00 | 00 | F8 | 00 | 00 | 3F | 00 | 10 | 00 | 80 | 00 00 | 00 | | ........ | ?....... |
| 00010020 | | | 00 E8 | 07 | 00 | E5 | 03 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 00 | 00 | | ................ |  |
| 00010030 | | | 01 00 | 06 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 00 | 00 | | ................ |  |
| 00010040 | | | 80 00 | 29 | 85 | B3 | EA 1C 4E | | | 4F | 20 | 4E | 41 | 4D | 45 20 | 20 | | ..).... | NO NAME |
| 00010050 | | | 20 20 | 46 | 41 | 54 | 33 | 32 | 20 | 20 | 20 |  |  |  |  |  | | FAT32 |  |



9



1. Number of sectors per cluster in decimal.

Located at 0x0001000D, is 0x01 which is 1 in decimal. There is 1 sector per cluster.

1. Size of each cluster in bytes (in dec)

0x200 = 512 bytes per cluster.

1. Identify the physical location of the Root directory. Show the steps

MBR = 0x00010000

Reserved region = 0x16E \* 0x200 = 0x2DC00

Fat Region = 0x3E5 \* 0x200 = 0x7CA00

Root location = 0x00010000 + 0x2DC00 + 0x7CA00 = 0xAB600

d) Identify the physical location of the first FAT table.



FAT table 1 = MBR + reserved region

= 0x10000 + 0x2DC00 = 0x3DC00

10

**8.** Following is the **Root directory entry** information of a FAT32 file system.

Offset | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F | ASCII

---------------------------------------------------------------------------------

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 00400000 | | | 46 | 41 | 54 | 33 | 32 20 | 20 | 20 | 20 | 20 | 20 | 08 | 00 | 00 | 00 | 00 | | FAT32 |  | ..... |
| 00400010 | | | 00 | 00 | 00 | 00 | 00 00 | 38 | 98 | 4D | 52 | 00 | 00 | 00 | 00 | 00 | 00 | | ...... | 8.MR...... |  |
| 00400080 | | | E5 | 4E 4F 4E | | | 59 20 | 20 | 20 | 4A | 50 | 47 | 20 | 18 | 8E | 3A | 98 | | .NONY | JPG ..:. | |
| 00400090 | | | 4D | 52 4D 52 | | | 00 00 | 86 | 4E | 3D | 4E | 06 | 00 | 69 | 28 | 00 | 00 | | MRMR... | N=N..i(.. | |
| 004000A0 | | | 41 | 74 00 65 | | | 00 73 | 00 | 74 | 00 | 2E | 00 | 0F | 00 | E5 | 74 | 00 | | At.e.s.t...... | | t. |
| 004000B0 | | | 78 | 00 74 00 | | | 2E 00 | 74 | 00 | 78 | 00 | 00 | 00 | 74 | 00 | 00 | 00 | | x.t... | t.x... | t... |
| 004000C0 | | | 54 | 45 53 54 | | | 54 58 | 7E | 31 | 54 58 | | 54 | 20 | 00 | 8E | 3A 98 | | | TESTTX~1TXT ..:. | | |
| 004000D0 | | | 4D | 52 4D 52 | | | 00 00 | 4B | 8E | 41 | 4E | 1B | 00 | 1F | 00 | 00 | 00 | | MRMR..K.AN...... | |  |
| 004000E0 | | | 24 | 52 45 43 | | | 59 43 | 4C | 45 | 42 | 49 | 4E | 16 | 00 | AF 3A | | 98 | | $RECYCLEBIN | | ...:. |
| 004000F0 | | | 4D | 52 4D 52 | | | 00 00 3B 98 | | | 4D 52 1C 00 00 00 00 00 | MRMR..;.MR...... | | | | | | | | | |  |
| 00400100 | | | 42 | 78 00 74 | | | 00 2E | 00 | 74 | 00 | 78 | 00 | 0F | 00 | 32 | 74 | 00 | | Bx.t... | t.x... | 2t. |
| 00400110 | | | 00 | 00 FF FF | | | FF FF FF FF | | | FF FF | | 00 | 00 | FF | FF FF | | FF | ................ | |  |  |
| 00400120 | | | 01 | 6C 00 61 | | | 00 72 | 00 | 67 | 00 65 00 0F 00 32 5F 00 | .l.a.r.g.e... | | | | | | | | | | 2\_. |
| 00400130 | | | 5F | 00 66 00 | | | 69 00 | 6C | 00 | 65 | 00 | 00 | 00 | 2E | 00 | 74 | 00 | | \_.f.i.l.e..... | | t. |
| 00400140 | | | 4C | 41 52 47 | | | 45 5F | 7E | 31 | 54 | 58 | 54 | 20 | 00 | 02 | EE | A1 | | LARGE\_~1TXT .... | | |
| 00400150 | | | 4D | 52 4D 52 | | | 00 00 | E4 | A1 | 4D | 52 1E 00 00 C0 17 00 | MRMR.... | | | | | | | | MR...... |  |
| 00400160 | | | E5 | 74 00 78 | | | 00 74 | 00 | 2E | 00 | 74 | 00 | 0F | 00 | 96 | 78 | 00 | | .t.x.t | ...t.... | x. |
| 00400170 | | | 74 | 00 00 00 | | | FF FF FF FF | | | FF FF | | 00 | 00 | FF | FF FF | | FF | t............... | |  |  |
| 00400180 | | | E5 | 73 00 65 | | | 00 63 | 00 | 72 | 00 | 65 | 00 | 0F 00 96 74 00 | .s.e.c.r.e.... | | | | | | | t. |
| 00400190 | | | 5F | 00 5F | | 00 | 66 00 | 69 | 00 | 6C | 00 | 00 | 00 | 65 | 00 | 2E | 00 | | \_.\_.f.i.l... | | e... |
| 004001A0 | | | E5 | 45 43 52 | | | 45 54 | 7E | 31 | 54 | 58 | 54 | 20 | 00 | 21 | F8 | A1 | | .ECRET~1TXT .!.. | | |
| 004001B0 | | | 4D | 52 4D | | 52 | 00 00 | E4 | A1 | 4D | 52 | FE 0B 00 | | | C0 17 00 | MRMR.... | | | | MR...... |  |
| 004001C0 | | | 42 | 74 00 78 | | | 00 74 | 00 | 2E | 00 | 74 | 00 | 0F | 00 | 39 | 78 | 00 | | Bt.x.t | ...t... | 9x. |
| 004001D0 | | | 74 | 00 00 00 | | | FF FF FF FF | | | FF FF | | 00 | 00 | FF | FF FF | | FF | t............... | |  |  |
| 004001E0 | | | 01 | 61 00 | | 6E | 00 6F | 00 | 74 | 00 | 68 | 00 | 0F | 00 | 39 | 65 | 00 | | .a.n.o.t.h... | | 9e. |
| 004001F0 | | | 72 | 00 5F | | 00 | 66 00 | 69 | 00 | 6C | 00 | 00 | 00 | 65 | 00 | 2E | 00 | | r.\_.f.i.l... | | e... |



11



**FAT TABLE 1**

**Offset** **|** **00 01 02 03 04 05 06 07** **08 09 0A 0B 0C 0D 0E 0F |** **ASCII**

---------------------------------------------------------------------------------

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0001BC00 | | | F8 | FF FF | 0F FF | | FF FF | FF | DE 17 00 00 FF FF FF 0F | ................ | | | | |
| 0001BC10 | | | FF FF FF | | 0F | FF | FF FF | 0F | 07 | 00 00 00 | 0A | 00 00 00 | | ................ |
| 0001BC20 | | | 11 | 00 00 | 00 | FF FF FF 0F | | | 0F | 00 00 00 | 0C | 00 00 00 | | ................ |
| 0001BC30 | | | 0E | 00 00 | 00 | 13 | 00 00 00 | | 0D | 00 00 00 | 10 | 00 00 00 | | ................ |
| 0001BC40 | | | 08 | 00 00 | 00 | 0B | 00 00 00 | | FF FF FF 0F | | FF FF FF 0F | | | ................ |

1. The root directory is split into 2 cluster locations. The first cluster is at physical sector **0x400000**. With the help of the FAT table, identify the second physical location in hex.Given that the logical cluster location of root directory is 2.

**[5 points]**

FAT table 2 = MBR + reserved region + FAT table 1

Bytes per sector = 0xFF = 255 bytes

MBR = 0x400000

Reserved region = 0xFFF \* 0xFF = 0xEF1

FAT table size = 0x0FFFFFFF \* 0xFF \* 0xFF = 0xFE00FFF01FF

Table 2 = 0x400000 + 0xEF1 + 0xFE00FFF01FF = 0xFE0103F10F0



12



b) Identify the 2 deleted files and fill in the table **[8 points]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Type** | **Physical** | **Size of the file** |  |
| **File Name** | **location** |  |
| **(txt/jpg/png)** | **(in dec)** |  |
|  | **(in hex)** |  |
|  |  |  |  |
|  |  |  |  |  |
| NONY.jpg | Jpg |  |  |  |
| Bx.txt | Txt |  |  |  |



I know that deleted files have an inode that is 0. This means that the first 4 bytes of the file entry needs to be 0x00000000.

c) With the help of the FAT table, provide all the logical and physical cluster locations of the file whose starting cluster number (logical) is 6. Show all the steps. **[10 points]**

**Note: This is a tricky question. If you get it correct, you get 5 additional points as bonus!**



Not sure where to begin with this.

13

