# Digital Forensics – Lab05

Ahmad Abdullah i22-1609

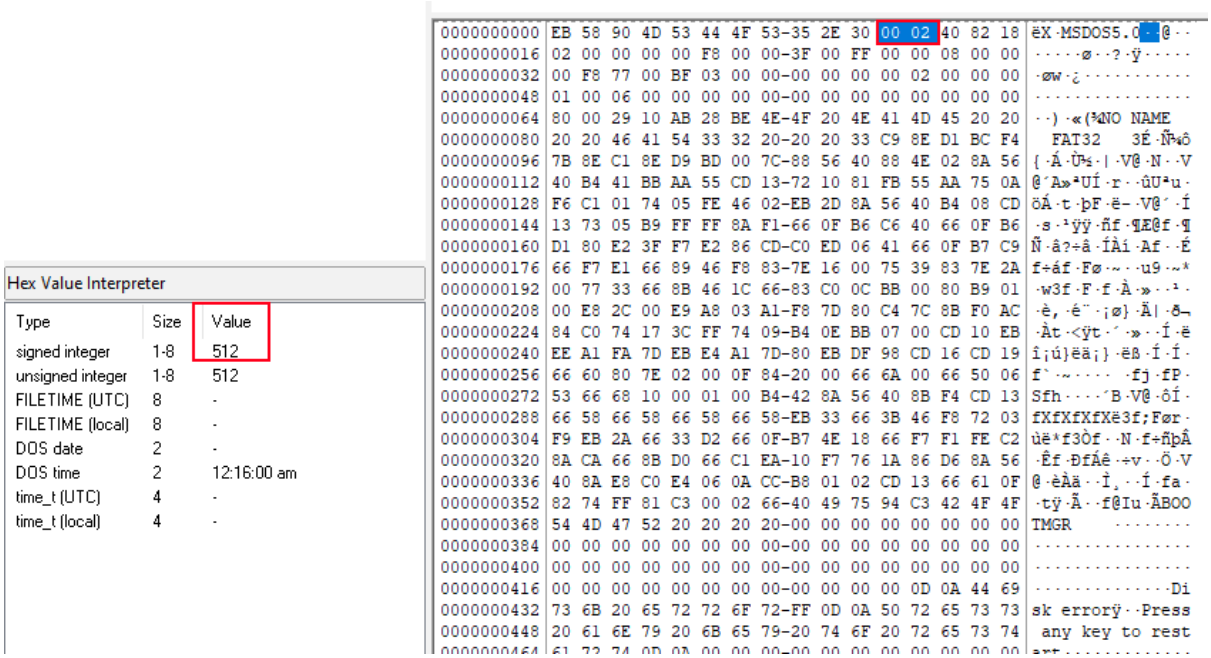
**Task#1: Find Byte per Sector and Sectors per Cluster for Image File? Pay attention to the endianness**

From the Lab manual, we knew where to look.

**0x0B 2 bytes Bytes per Sector**

**0x0D 1 Sectors per Cluster**

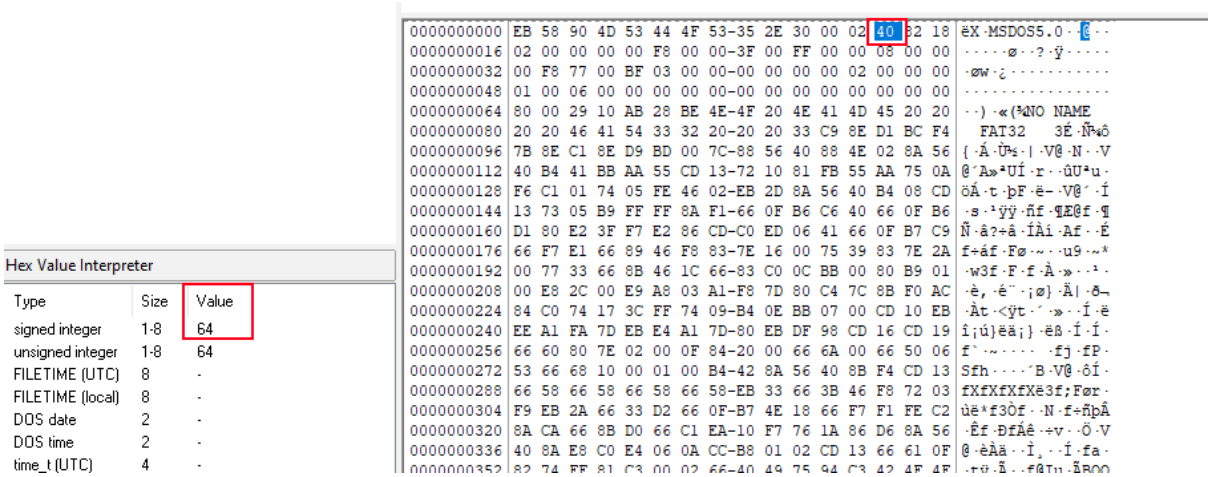
Looking at those offset bits one by one we get the byte size per sector and then sectors per cluster.



Hex Value Interpreter

| Type | Size | Value |
|---|---|---|
| signed integer | 1-8 | 512 |
| unsigned integer | 1-8 | 512 |
| FILETIME (UTC) | 8 | - |
| FILETIME (local) | 8 | - |
| DOS date | 2 | - |
| DOS time | 2 | 12:16:00 am |
| time_t (UTC) | 4 | - |
| time_t (local) | 4 | - |

```
0000000000 EB 58 90 4D 53 44 4F 53-35 2E 30 00 02 40 82 18 ëX·MSDOS5.0··@··
0000000016 02 00 00 00 00 F8 00 00-3F 00 FF 00 00 08 00 00 ·····ø··?·ÿ·····
0000000032 00 F8 77 00 BF 03 00 00-00 00 00 00 02 00 00 00 ·øw·¿···********
0000000048 01 00 06 00 00 00 00 00-00 00 00 00 00 00 00 00 ********
0000000064 80 00 29 10 AB 28 BE 4E-4F 20 4E 41 4D 45 20 20 ··)·«(¾NO NAME
0000000080 20 20 46 41 54 33 32 20-20 20 33 C9 8E D1 BC F4   FAT32   3É·Ñ¼ô
0000000096 7B 8E C1 8E D9 BD 00 7C-88 56 40 88 4E 02 8A 56 {·Á·Ù½·|·V@·N··V
0000000112 40 B4 41 BB AA 55 CD 13-72 10 81 FB 55 AA 75 0A @´A»ªUÍ·r··ûUªu·
0000000128 F6 C1 01 74 05 FE 46 02-EB 2D 8A 56 40 B4 08 CD öÁ·t·þF·ë-·V@´·Í
0000000144 13 73 05 B9 FF FF 8A F1-66 0F B6 C6 40 66 0F B6 ·s·¹ÿÿ·ñf·¶Æ@f·¶
0000000160 D1 80 E2 3F F7 E2 86 CD-C0 ED 06 41 66 0F B7 C9 Ñ·â?÷â·ÍÀí·Af··É
0000000176 66 F7 E1 66 89 46 F8 83-7E 16 00 75 39 83 7E 2A f÷áf·Fø·~··u9·~*
0000000192 00 77 33 66 8B 46 1C 66-83 C0 0C BB 00 80 B9 01 ·w3f·F·f·À·»··¹·
0000000208 00 E8 2C 00 E9 A8 03 A1-F8 7D 80 C4 7C 8B F0 AC ·è,·é¨·¡ø}·Ä|·ð¬
0000000224 84 C0 74 17 3C FF 74 09-B4 0E BB 07 00 CD 10 EB ·Àt·<ÿt·´·»··Í·ë
0000000240 EE A1 FA 7D EB E4 A1 7D-80 EB DF 98 CD 16 CD 19 î¡ú}ëä¡}·ëß·Í·Í·
0000000256 66 60 80 7E 02 00 0F 84-20 00 66 6A 00 66 50 06 f`·~····· ·fj·fP·
0000000272 53 66 68 10 00 01 00 B4-42 8A 56 40 8B F4 CD 13 Sfh····´B·V@·ôÍ·
0000000288 66 58 66 58 66 58 66 58-EB 33 66 3B 46 F8 72 03 fXfXfXfXë3f;Før·
0000000304 F9 EB 2A 66 33 D2 66 0F-B7 4E 18 66 F7 F1 FE C2 ùë*f3Òf··N·f÷ñþÂ
0000000320 8A CA 66 8B D0 66 C1 EA-10 F7 76 1A 86 D6 8A 56 ·Êf·ÐfÁê·÷v··Ö·V
0000000336 40 8A E8 C0 E4 06 0A CC-B8 01 02 CD 13 66 61 0F @·èÀä··Ì¸··Í·fa·
0000000352 82 74 FF 81 C3 00 02 66-40 49 75 94 C3 42 4F 4F ·tÿ·Ã··f@Iu·ÃBOO
0000000368 54 4D 47 52 20 20 20 20-00 00 00 00 00 00 00 00 TMGR    ********
0000000384 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ********
0000000400 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ********
0000000416 00 00 00 00 00 00 00 00-00 00 00 00 0D 0A 44 69 ··********··Di
0000000432 73 6B 20 65 72 72 6F 72-FF 0D 0A 50 72 65 73 73 sk errorÿ··Press
0000000448 20 61 6E 79 20 6B 65 79-20 74 6F 20 72 65 73 74  any key to rest
0000000464 61 72 74 0D 0A 00 00 00-00 00 00 00 00 00 00 00 art*************
```




After getting both byte size and cluster size we can easily determine the cluster size by the formual

Byte Size = 512 bytes

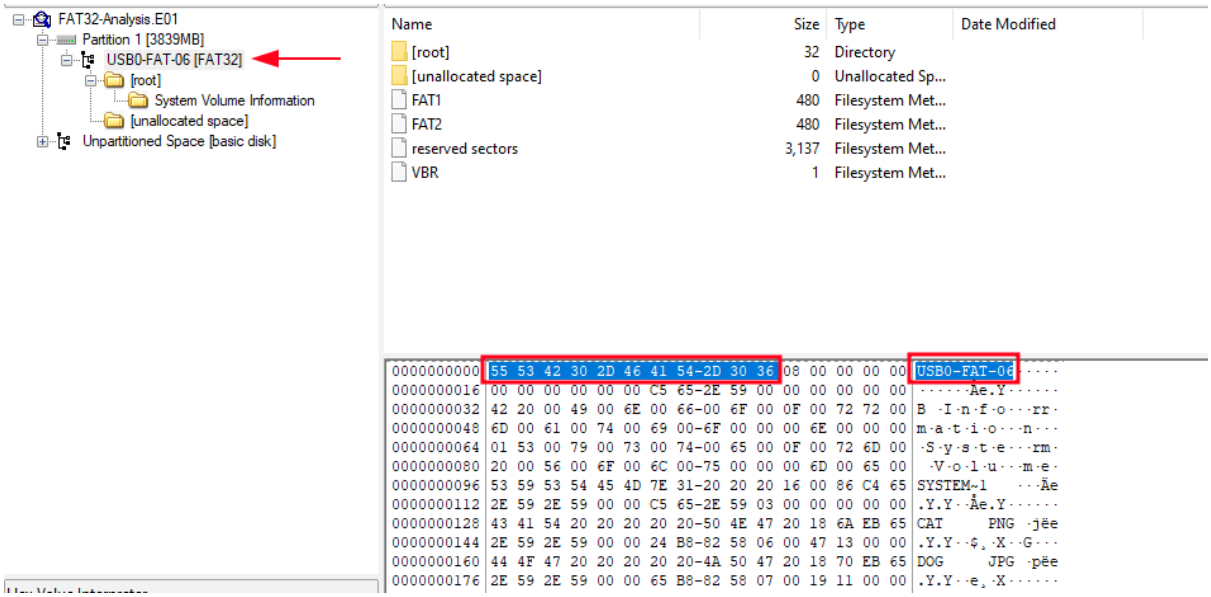
Sector Size = 64 Bytes

Cluster Size = byte size X Sector Size

Cluster Size = 512 x 64 = 32,768 bytes

**Task#2: Show and explain where the FAT file system stores the volume label.**

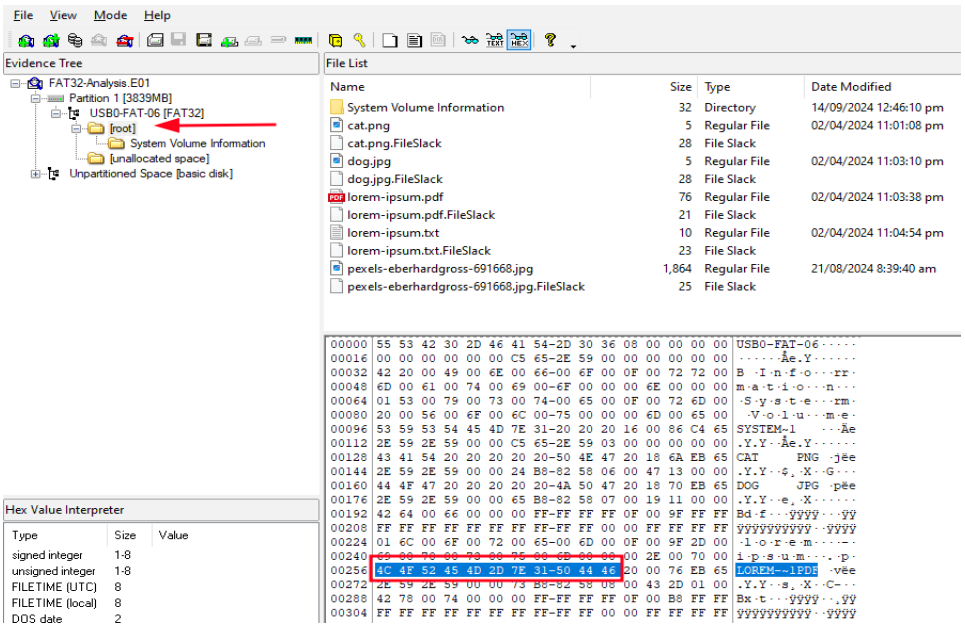
Just selecting the volume the first bytes we see are of volume label/name of the volume.



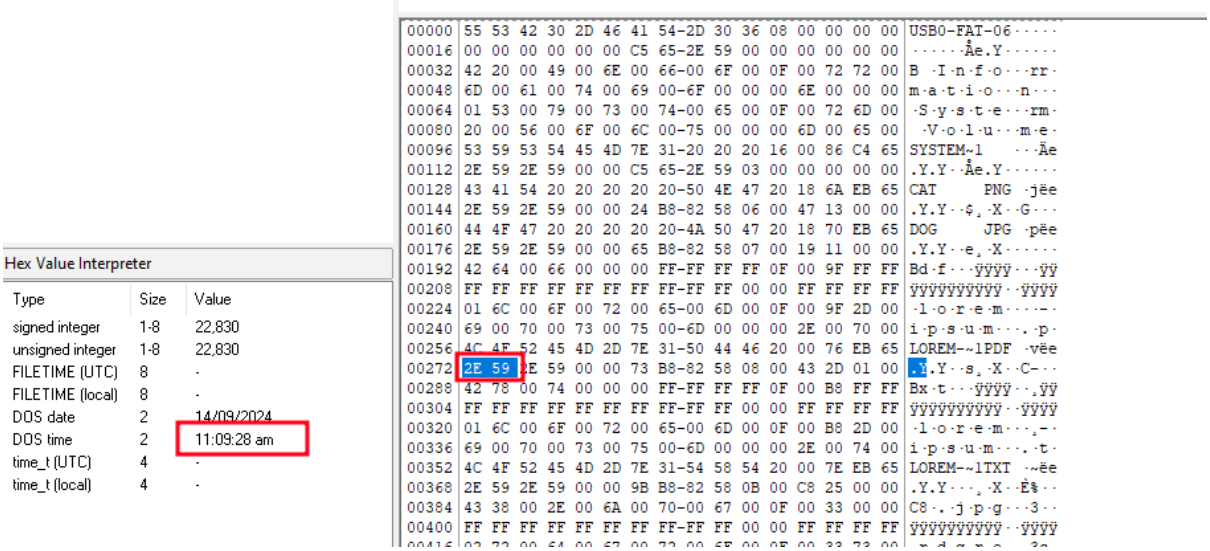

**Task#3: Check the FAT root directory, explain how the filename and extension can be extracted from these entries.**

We were required to find the file name and extension of lorem ipsum. PDF file along with its metadata.

1. Locate the file's short name. Usually, the 0x00 and 0x10 sectors are for files with short name and above files short name, the hex usually contains files long name .

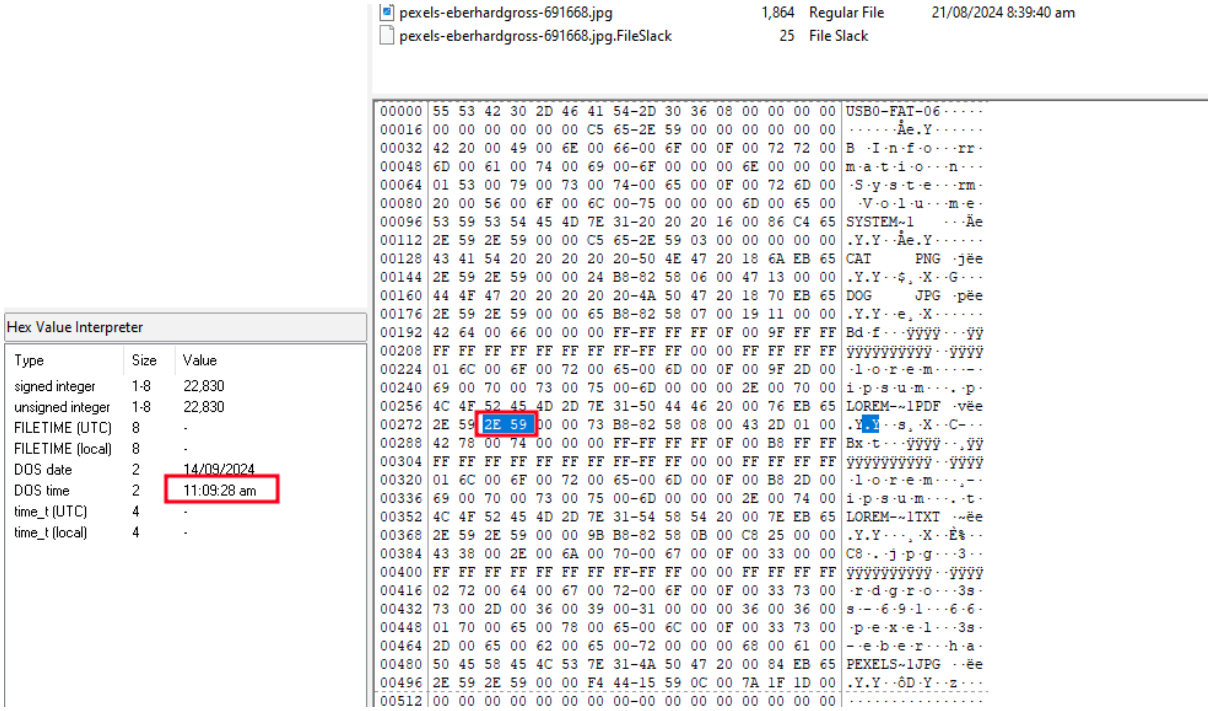


**Task#4: Determine the date and time when the file “lorem-ipsum.pdf” was created / modified based on the root entry hex data?**



Hex Value Interpreter

| Type | Size | Value |
|---|---|---|
| signed integer | 1-8 | 22,830 |
| unsigned integer | 1-8 | 22,830 |
| FILETIME (UTC) | 8 | - |
| FILETIME (local) | 8 | - |
| DOS date | 2 | 14/09/2024 |
| DOS time | 2 | 11:09:28 am |
| time_t (UTC) | 4 | - |
| time_t (local) | 4 | - |

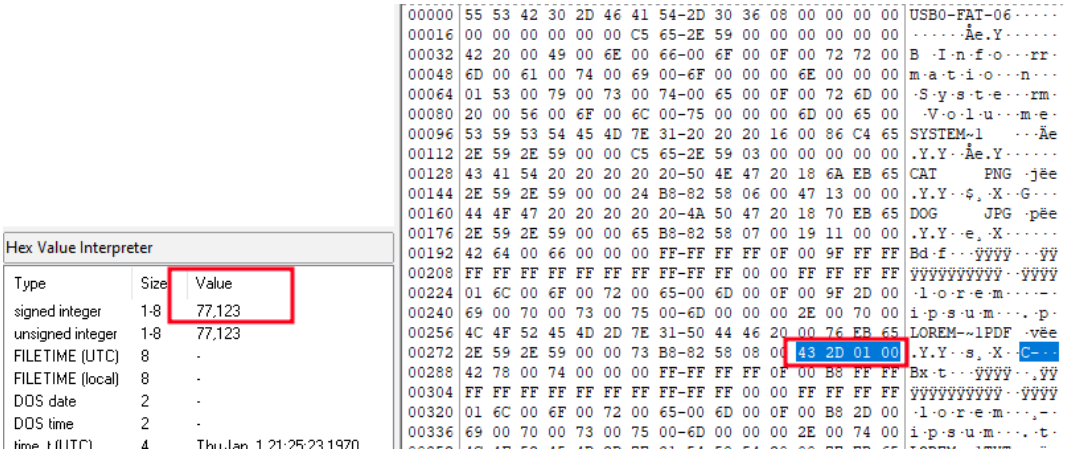


pexels-eberhardgross-691668.jpg 1,864 Regular File 21/08/2024 8:39:40 am
pexels-eberhardgross-691668.jpg.FileSlack 25 File Slack

Hex Value Interpreter

| Type | Size | Value |
|---|---|---|
| signed integer | 1-8 | 22,830 |
| unsigned integer | 1-8 | 22,830 |
| FILETIME (UTC) | 8 | - |
| FILETIME (local) | 8 | - |
| DOS date | 2 | 14/09/2024 |
| DOS time | 2 | 11:09:28 am |
| time_t (UTC) | 4 | - |
| time_t (local) | 4 | - |

**Task#5: Determine the date and time when the file "lorem-ipsum.pdf" was created / modified based on the root entry hex data?**




## RAM, Drive and File Slack:

### RAM Slack

- File Size: 77,123 bytes

- Sector Size: 512 bytes

File Size modulo Sector Size = File Size % Sector Size

= 77123 % 512

= 323

RAM Slack = Sector Size - (File Size modulo Sector Size)

= 512 - 323

= 189 bytes

### Drive Slack

- File Size: 77,123 bytes

- Sector Size: 512 bytes

- Cluster Size: 32,768 bytes

Drive Slack = Cluster Size - (File Size % Cluster Size)

= 32768 – (77123%32,768)

=32768 - 11587

= 21181 bytes

**File Slack**

- File Size: 77,123 bytes

- Cluster Size: 32,768 bytes

Step 1: Determine the Number of Clusters Needed

Number of Clusters Needed = File Size / Cluster Size

Number of Clusters Needed = 77,123/ 32,768 ≈ 2 cluster

Step 2: Check for Additional Cluster Needed

If (File Size modulo Cluster Size <> 0), add 1 additional cluster needed

File Size modulo Cluster Size = 45,334 mod 32,768 = 12,566

Since the remainder is not zero, an additional cluster is needed.

Total Clusters Needed = 3 clusters

Step 3: Calculate File Slack

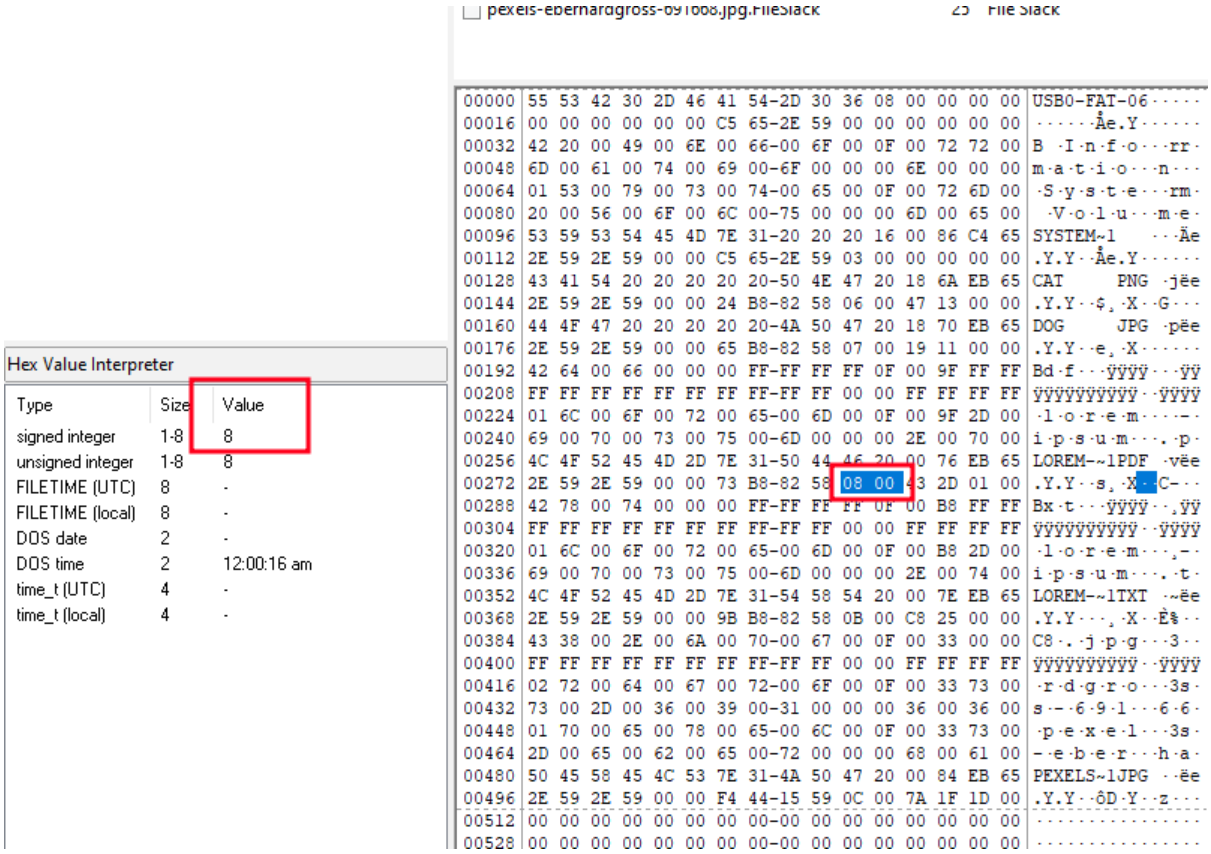
File Slack = (Clusters Needed * Cluster Size) - File Size

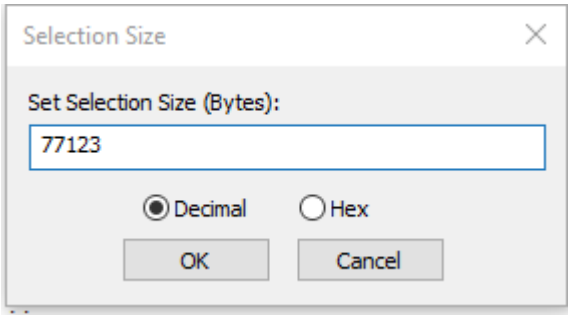
File Slack = (3 * 32,768) - 77123

File Slack = 98304 - 77123

File Slack = 21181 bytes

**Task#6: Analyze the root directory entry, compute the start and end offsets where the data of the file is located and manually extract the file using a hex editor. Compute hash values for the original file (i.e., original copy that you still have on your laptop PC) and the manually extracted file (i.e., from the USB) and verify if they match.**



Both Files have same Hash which tells us that file was not modified.