|  |  |
| --- | --- |
| **Digital Forensics**  Diploma in CSF/IT  Year 2/3 (2022/23) Semester 4/6 | Week 7 |
| Tutorial 6 |
| **NTFS Analysis** | |

**OBJECTIVES**

After completing this topic, you should be able to

1. Perform Hexadecimal number addition without using calculator;
2. Perform forensic analysis on NTFS file system;
3. Explain the difference between different timestamps in MFT.

Q1: Perform addition on the hexadecimal numbers using Table 1.

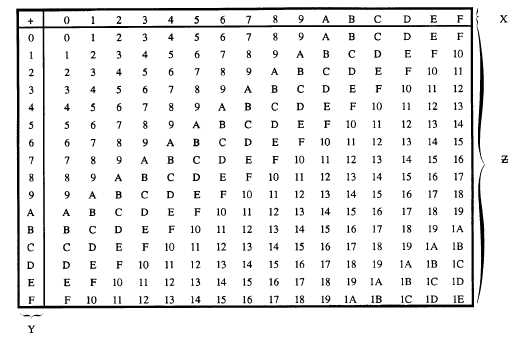


Figure 1: Hexadecimal Addition Table



|  |  |  |  |
| --- | --- | --- | --- |
|  | 4 | 5 | 616 |
| + | 7 | 8 | 416 |
|  | B | D | A |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 7 | 8 | 416 |
| + | B | D | A16 |
| 1 | 3 | 5 | E |

|  |  |  |
| --- | --- | --- |
|  | 9 | 816 |
| + | 7 | 816 |
| 1 | 1 | 0 |

Q2. Figure 2-1 shows the MFT record of an NTFS file system.

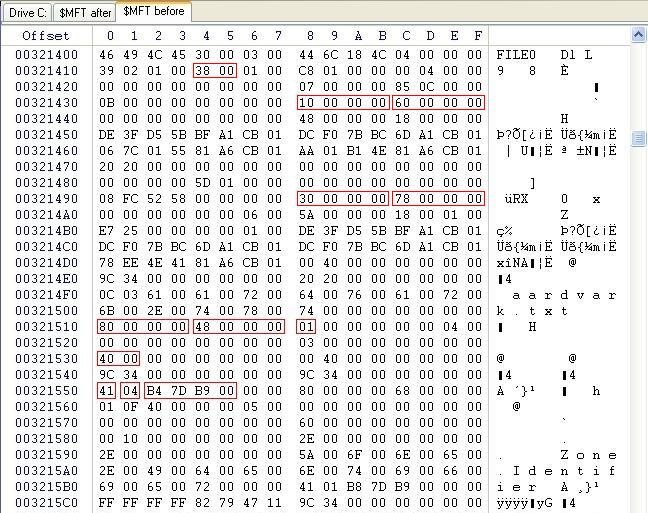


Figure 2-1: MFT File Record

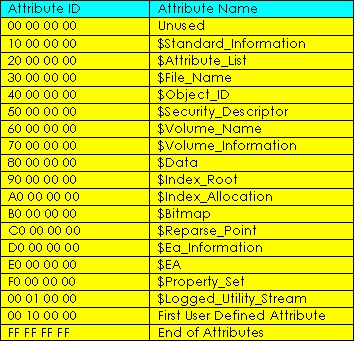


Figure 2-2: NTFS Attributes

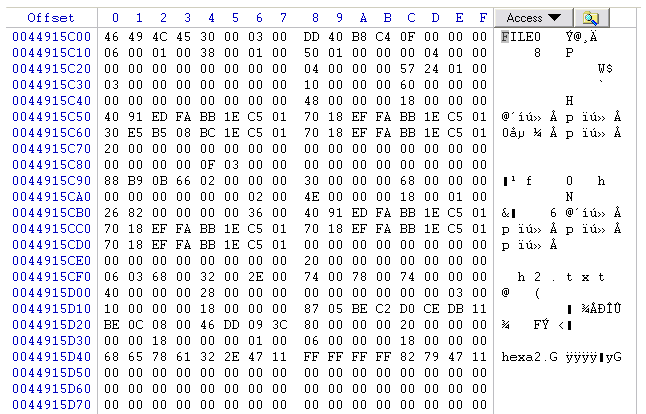
Answer the following with reference to Figure 2-1 and 2-2. You may use Hex addition table or calculator to perform the addition.

1. What is the byte offset to the first attribute? 0x38
2. What is the attribute name of this first attribute? $Standard\_Information

The next 4 bytes immediately after attribute type is the length (in bytes) of that attribute.

1. What is the length of first attribute? 0x60 (hex)
2. What is the byte offset of second(THIRD?) attribute? 0x98
3. What is the attribute name of second(THIRD?) attribute? $File\_Name
4. What is the name of the file in this entry? Aardvark.txt
5. What is the byte offset of the file content ($Data)? 0x110

Q3. (a) An investigator was examining a hard disk drive that was formatted with NTFS file system. Figure 3(a)-1 shows an entry/record in the Master File Table (MFT) and Figure 3(a)-2 and Figure 3(a)-3 show the list of NTFS attributes and Attribute header respectively.



**Byte offset 0x38**

Figure 3(a)-1: An Entry/Record in Master File Table

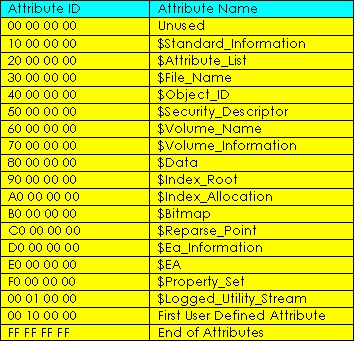


Figure 3(a)-2: NTFS Attributes

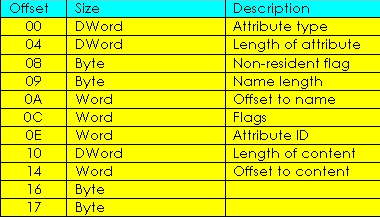


Figure 3(a)-3: Attribute Header

Given that the byte offset to the first attribute is **0x38** (hex), based on Figure 3(a)-1 to 3(a)-3 above, answer the following questions:

* 1. How do you tell if this MFT entry is a valid/usable entry? What represents an unusable entry?

Check the first 4 bytes and see if it combines to form the identifier “FILE”. This would signify a valid entry while seeing a “BAAD” would signify an unusable entry.

* 1. What is the name of the file/folder represented by this entry?

H2.txt

* 1. Complete the following table for this MFT entry.

|  |  |
| --- | --- |
|  | Attribute Name |
| 1st Attribute | $Standard\_Information |
| 2nd Attribute | $Attribute\_List |
| 3rd Attribute | $File\_Name |
| 4th Attribute | $Object\_ID |

* 1. Is the 4th attribute a Resident or Non-resident attribute? Explain.

Resident. The byte with an offset of 08 from the the 4th attribute is 00 indicating that it is a resident.

* 1. Which location on the hard disk can you find the content of this file? Explain.

(b) Briefly explain how Windows operating system stores files of the following sizes in disk drive configured with NTFS with respect to the Master File Table (MFT). In your answers, explain how the Non-Resident flag is set.

* 1. 100 Bytes
  2. 5 Kbytes

1. As the file is very small, Windows will store the file in the attribute in the MFT record. The non-resident flag will be set to 0
2. As the file too large to be stored in the MFT record, it will be stored in another cluster in the hard disk. Thus, it will be stored elsewhere on the disk and the non-resident flag will be set to 1.

Q4. Where is the Volume Boot Record (VBR) located in the NTFS partition?

Stored at logical sector zero as the $Boot file.

Q5. Both $Standard\_Information and $File\_Name contain MACE timestamps. Explain the difference between the timestamps found in these 2 attributes.

$Standard\_Information timestamps show Modified, Accessed, Created timestamps but not entry modified timestamp. But $File\_Name contains all 4 timestamps.

- End -