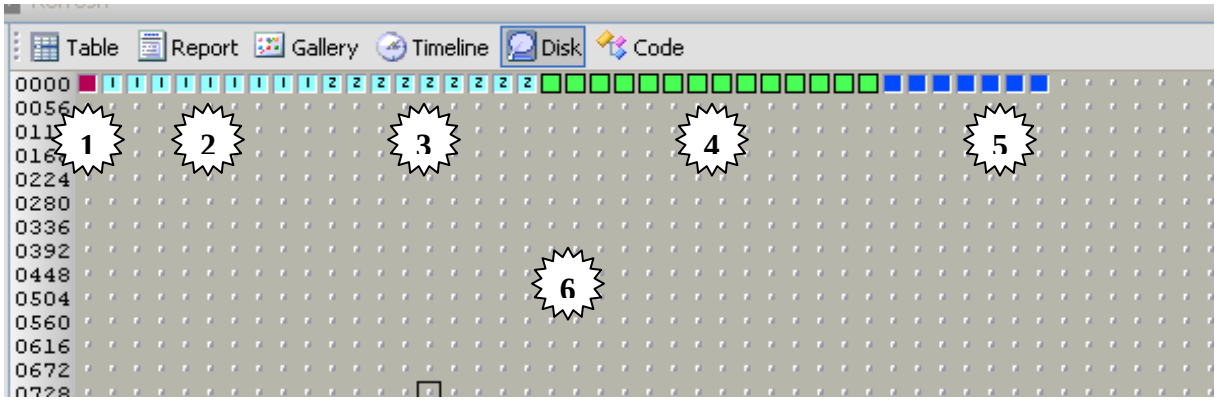
# Introduction to Computer Forensics and Security

## Task 1. Introduction

Through the following tasks we are going to understand the structure of the FAT 12 File System based on a bit by bit image acquired from a floppy disk. You can download three forensic images from Moodle.

The following is an overview of the current FAT12 disk structure:

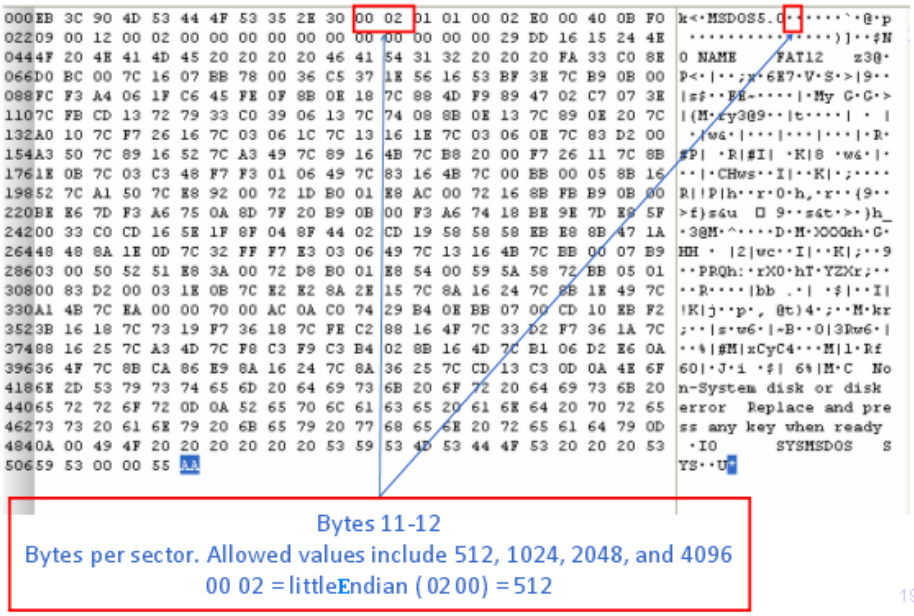



| | |
|---|---|
| 1 | FAT boot sector (1 sector) |
| 2 | FAT 1 (9 sectors) |
| 3 | FAT2 (9 sector) |
| 4 | Root directory (14 sectors) |
| 5 | Allocated sectors |
| 6 | Unallocated sectors |

# Task 2.1: Looking at the FAT boot sector

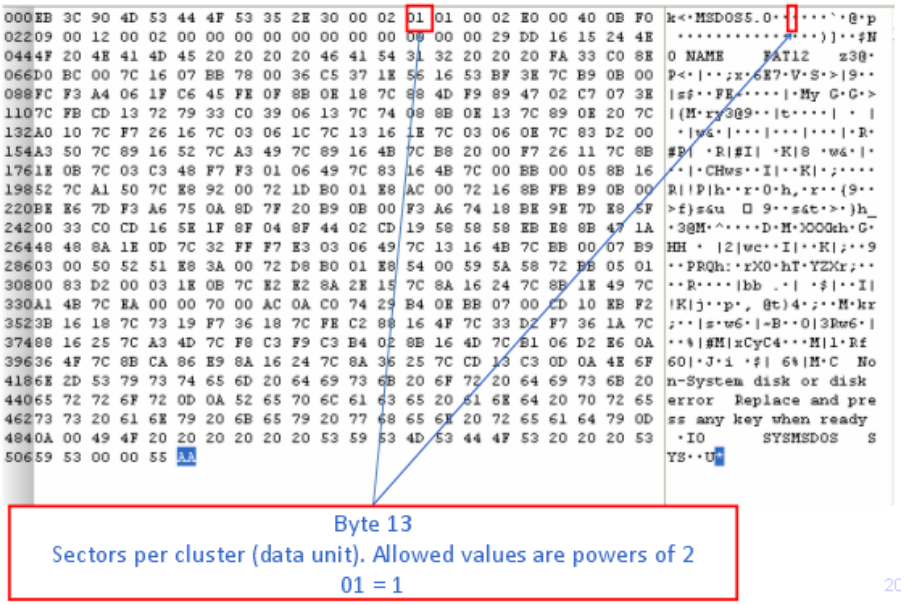
***Note: Through Task 2: Re-do the calculations yourself and compare your results.***

**Method** Looking at the bytes 11-12, Note: Make sure you start counting the bytes from 0,



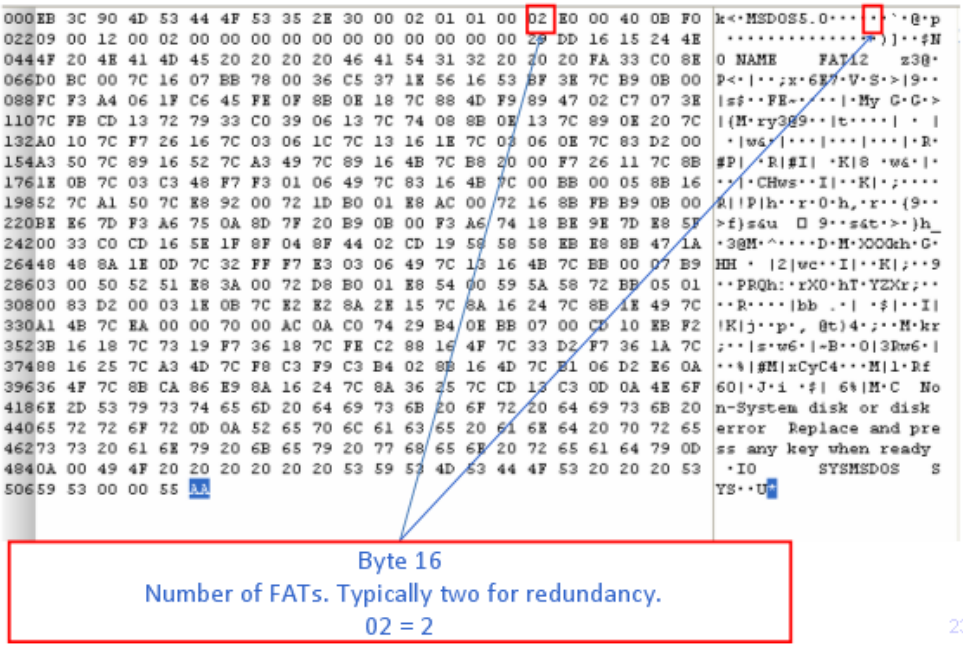

# Task 2.2: Looking at the FAT boot sector

**Method** Looking at the byte 13:



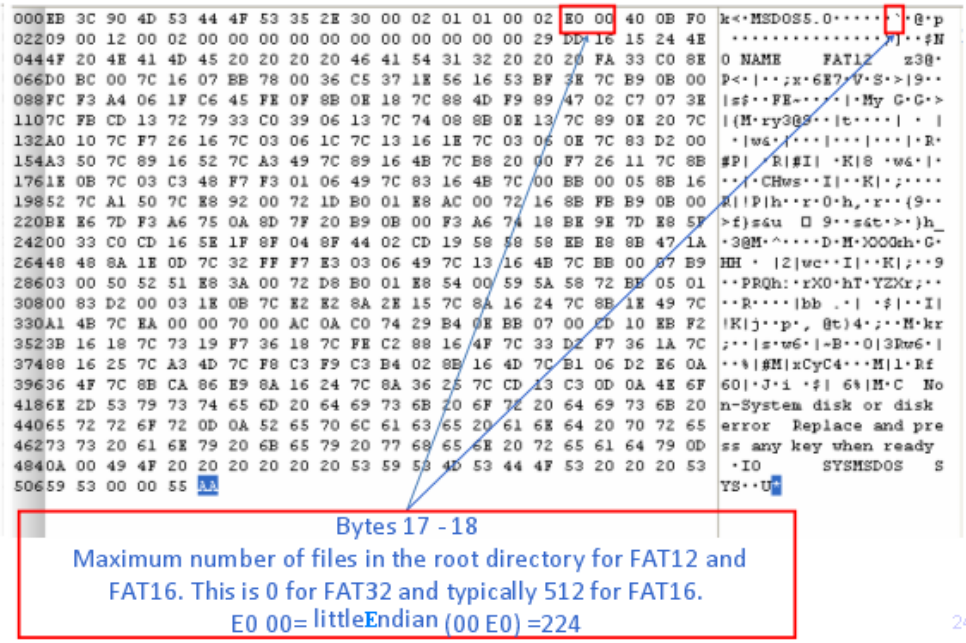
# Task 2.3: Looking at the FAT boot sector

**Method** Looking at the byte 16:



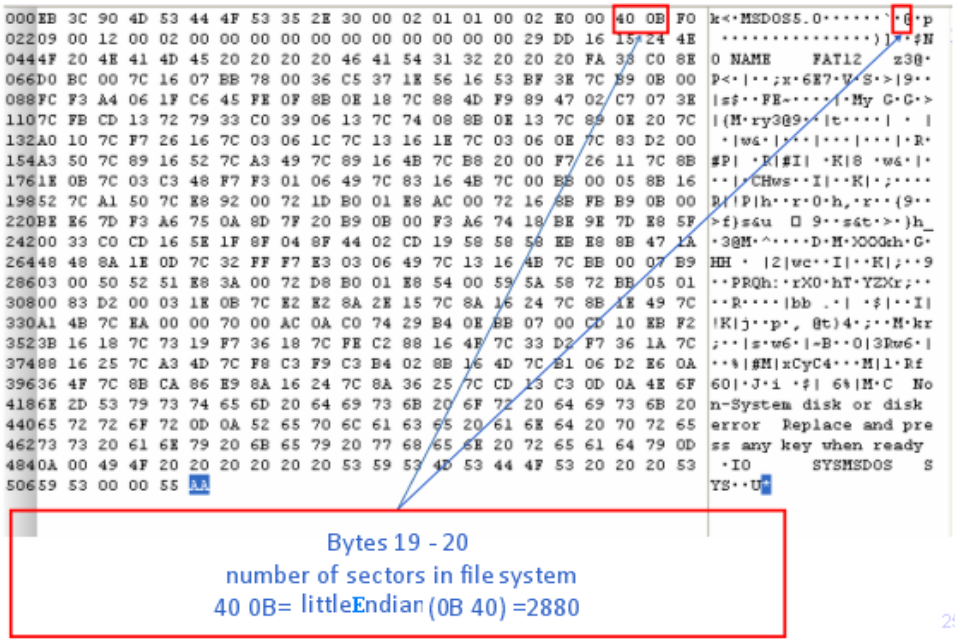

# Task 2.4: Looking at the FAT boot sector

**Method** Looking at the byte 17-18:



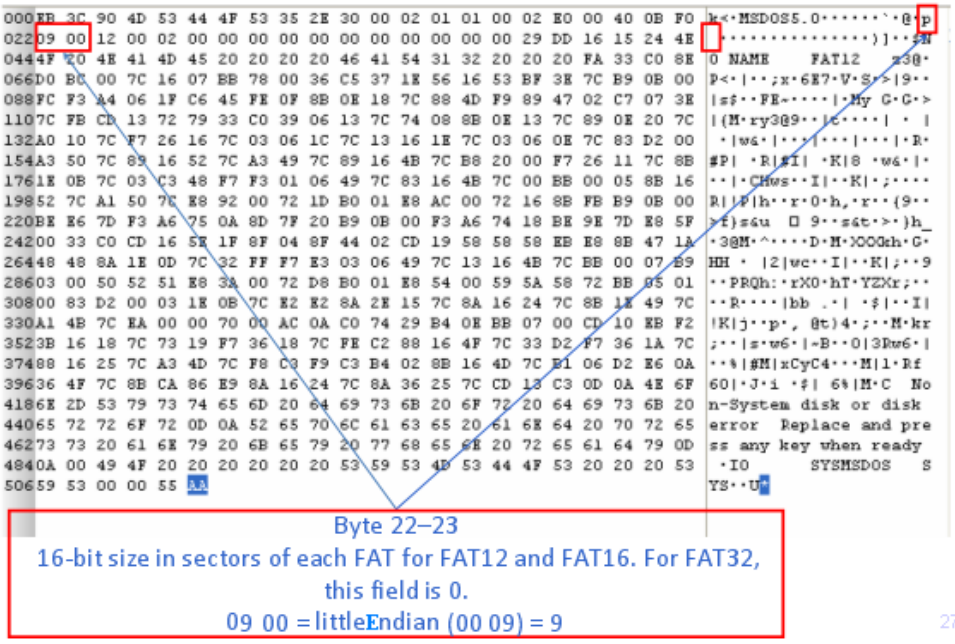
# Task 2.5: Looking at the FAT boot sector

**Method** Looking at the byte 19-20:




# Task 2.6: Looking at the FAT boot sector

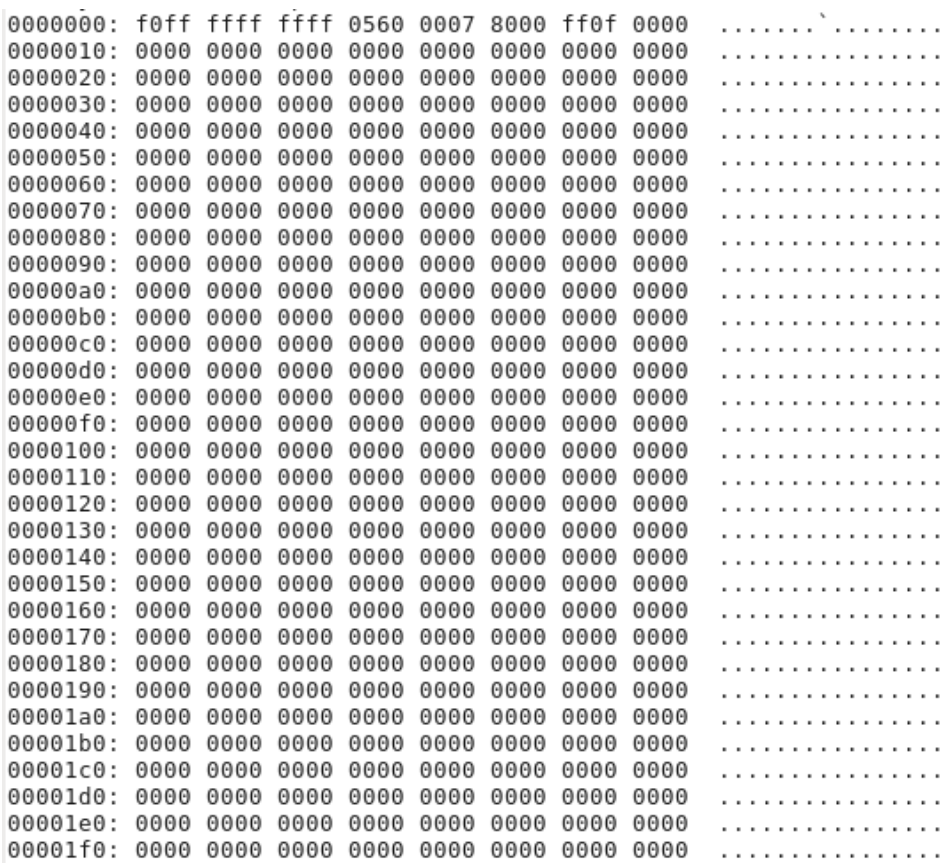
**Method** Looking at the byte 22-23:



# Task 3.1: Looking at FAT1

**Method**

You should be able to see the following,

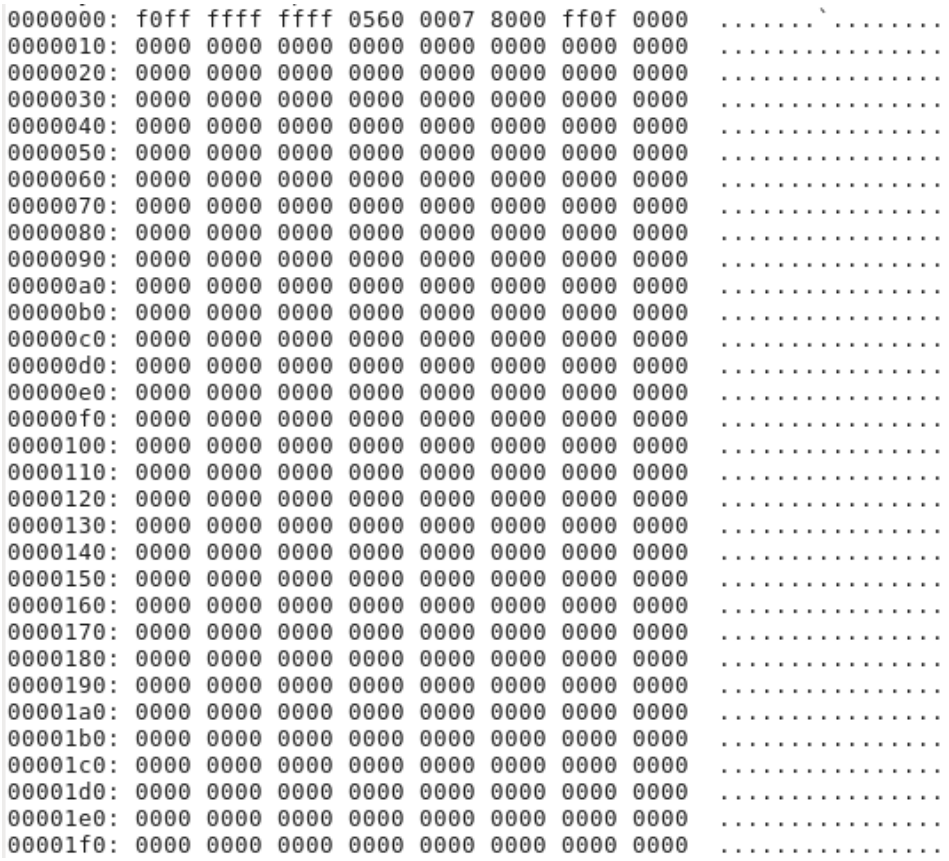


```
0000000: f0ff ffff ffff 0560 0007 8000 ff0f 0000  .......`........
0000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000100: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000120: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000130: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000140: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000150: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000160: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000170: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000180: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000190: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

# Task 3.2: Looking at FAT2

## Method

You should be able to see the following,



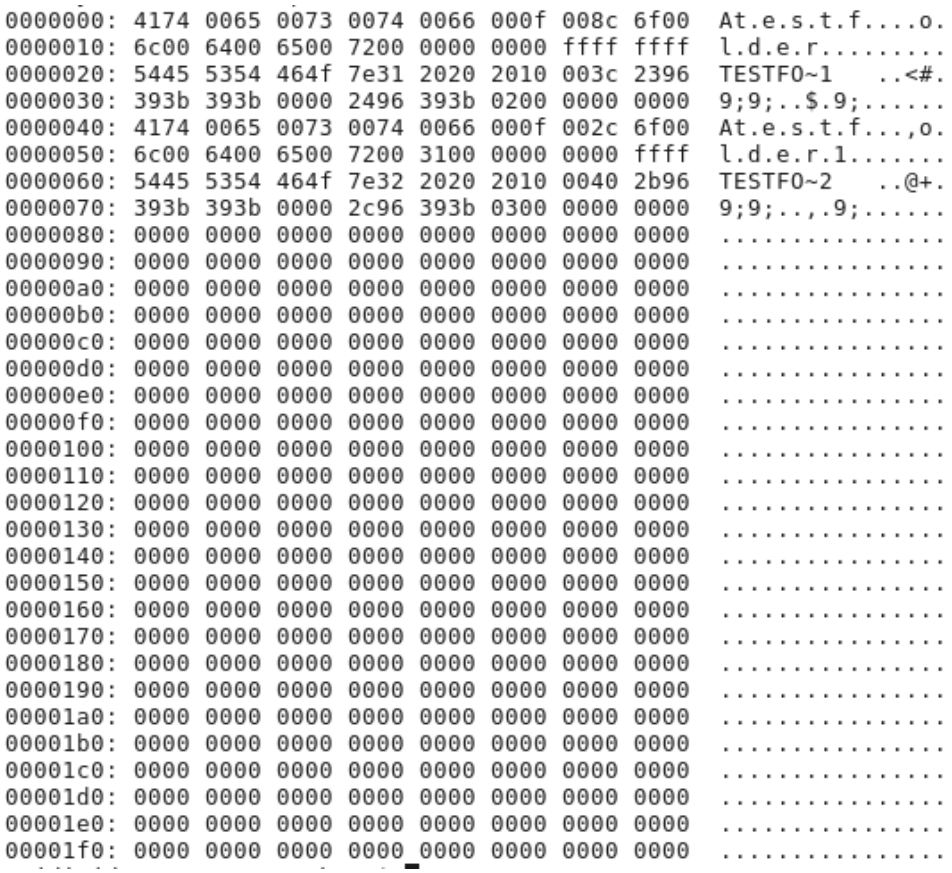
```
0000000: f0ff ffff ffff 0560 0007 8000 ff0f 0000  .......`........
0000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000100: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000120: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000130: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000140: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000150: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000160: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000170: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000180: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000190: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

***Note: Compare what you have seen here from what you have seen on previous task (3.1). Why?***

## Task 3.1: Looking at root directory entries

### Method

You should be able to see the following,




## Task 3.2: List files names/folders names if any, in the root directory. Use the tables 10.5 and 10.6.

---

---

## Task 3.3: Why do we need a long file name directory entry?

### Table 10.1. Data structure for the first 36 bytes of the FAT boot sector.

| Byte Range | Description | Essential |
|---|---|---|
| 0–2 | Assembly instruction to jump to boot code. | No (unless it is a bootable file system) |
| 3–10 | OEM Name in ASCII. | No |
| 11–12 | Bytes per sector. Allowed values include 512, 1024, 2048, and 4096. | Yes |
| 13–13 | Sectors per cluster (data unit). Allowed values are powers of 2, but the cluster size must be 32KB or smaller. | Yes |
| 14–15 | Size in sectors of the reserved area. | Yes |
| 16–16 | Number of FATs. Typically two for redundancy, but according to Microsoft it can be one for some small storage devices. | Yes |
| 17–18 | Maximum number of files in the root directory for FAT12 and FAT16. This is 0 for FAT32 and typically 512 for FAT16. | Yes |
| 19–20 | 16-bit value of number of sectors in file system. If the number of sectors is larger than can be represented in this 2-byte value, a 4-byte value exists later in the data structure and this should be 0. | Yes |
| 21–21 | Media type. According to the Microsoft documentation, 0xf8 should be used for fixed disks and 0xf0 for removable. | No |
| 22–23 | 16-bit size in sectors of each FAT for FAT12 and FAT16. For FAT32, this field is 0. | Yes |
| 24–25 | Sectors per track of storage device. | No |
| 26–27 | Number of heads in storage device. | No |
| 28–31 | Number of sectors before the start of partition.[1] | No |
| 32–35 | 32-bit value of number of sectors in file system. Either this value or the 16-bit value above must be 0. | Yes |

### Table 10.5. Data structure for a basic FAT directory entry.

| Byte Range | Description | Essential |
|---|---|---|
| 0–0 | First character of file name in ASCII and allocation status (0xe5 or 0x00 if unallocated) | Yes |
| 1–10 | Characters 2 to 11 of file name in ASCII | Yes |
| 11–11 | File Attributes (see Table 10.6) | Yes |
| 12–12 | Reserved | No |
| 13–13 | Created time (tenths of second) | No |
| 14–15 | Created time (hours, minutes, seconds) | No |
| 16–17 | Created day | No |
| 18–19 | Accessed day | No |
| 20–21 | High 2 bytes of first cluster address (0 for FAT12 and FAT16) | Yes |
| 22–23 | Written time (hours, minutes, seconds) | No |
| 24–25 | Written day | No |
| 26–27 | Low 2 bytes of first cluster address | Yes |
| 28–31 | Size of file (0 for directories) | Yes |

### Table 10.6. Flag values for the directory entry attributes field.

| Flag Value (in bits) | Description | Essential |
|---|---|---|
| 0000 0001 (0x01) | Read only | No |
| 0000 0010 (0x02) | Hidden file | No |
| 0000 0100 (0x04) | System file | No |
| 0000 1000 (0x08) | Volume label | Yes |
| 0000 1111 (0x0f) | Long file name | Yes |
| 0001 0000 (0x10) | Directory | Yes |
| 0010 0000 (0x20) | Archive | No |