COMP 5350 / 6350

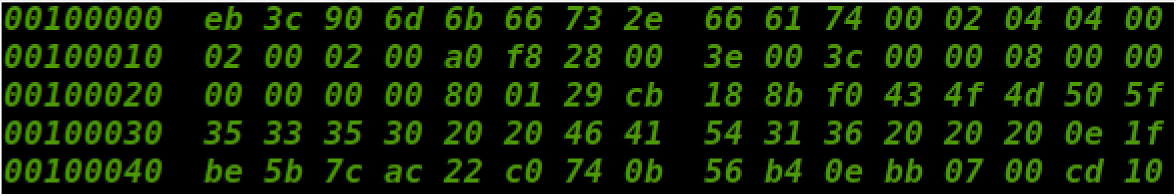
Digital Forensics

Introduction to File Systems
File Allocation Table Analysis

Command Line Notes

Hexadecimal Representation

- To read the contents of a disk volumes, we can use some of the built-in functions introduced in a previous session
  - ✓ hexdump
  - ✓ xxd
- Every position represents 1 byte (8-bits)
- Based on the file system specification, groups of bytes are combined into data structures



```
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10
```

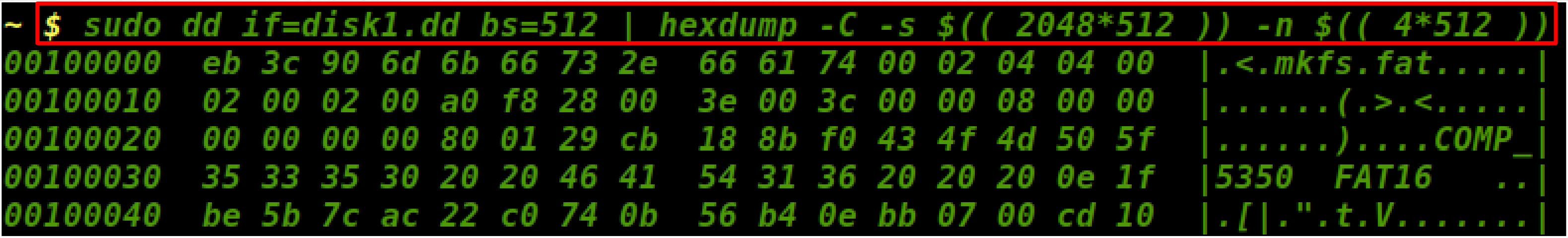
Displaying Disk Contents

- dd
  - ✓ Convert and copy files
    - if – Input File
    - of – Output File
    - bs – Block Size

- hexdump
  - ✓ File dump
    - C – Hexadecimal and ASCII output
    - s – Skip Number of Bytes
    - n – Number of Bytes to Display

- Calculations
  - ✓ $((5+2))

- Pipes
  - ✓ |

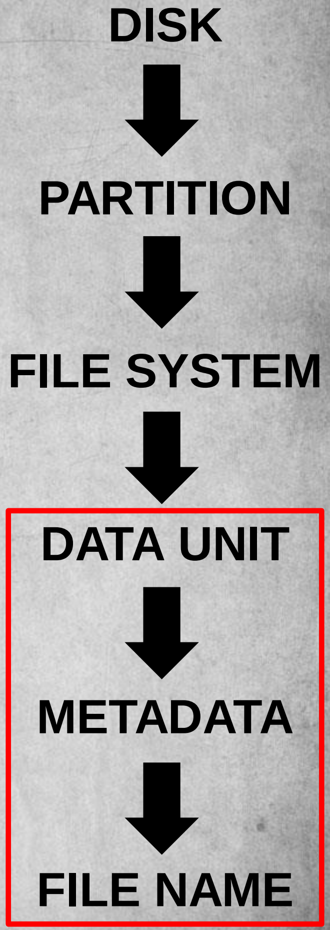


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 4*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......).....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
```

Introduction to File Systems

What is a File System?

- A file system provides hierarchal data structures needed to store and retrieve data that array files and directories
- File system data categories include:
  - Data Unit
    - Containers that hold file contents
    - Cluster / Block
  - Metadata
    - Data that describes a file
      - References to file storage locations
      - File Size
      - Modified, Accessed, and Created (MAC) times
  - Filename
    - Contains data that assigns a name for each file



DISK

PARTITION

FILE SYSTEM

DATA UNIT

METADATA

FILE NAME

File System Types

File System	Operating System	Details
FAT	Windows	File Allocation Table is a basic file system with 12, 16, and 32-bit versions and was the standard Windows file system before NTFS
exFAT	Windows	Extended File Allocation Table builds on FAT32 and offers a lightweight system without the overhead of NTFS
NTFS	Windows	The New Technology File System (NTFS) is the default file system used by modern Windows versions
ext	Linux	The extended file system was the first file system created specifically for the Linux kernel with ext 2, 3, and 4 as versions
HFS	macOS	The Hierarchical File System is the default file system used by macOS operating systems
APFS	macOS, iOS	The Apple File System was developed as a replacement for HFS, with a focus on flash drives, SSDs, and encryption APFS is the manditory file system for iOS 10.3+ and macOS 10.13+

Operating System vs. File System

- The following operating systems generally coincide with the following file systems
- Although extX is the default file system for the Linux OS, FAT and NTFS are also supported
- Although the extX file system is not supported in Windows by default, third-party solutions do exist

Operating System	File System(s)
MS-DOS	FAT16
Windows 95	FAT16
Windows 98	FAT32, NTFS
Windows NT	NTFS
Windows 2000	NTFS
Windows XP	FAT32, NTFS
Windows 7 / 8 / 10	NTFS
MacOS	HFS
Linux – Debian	ext2
Linux – Red Hat 5 / 6 / 7+	ext3, ext4, XFS

File Size Limitations

- Maximum file sizes for the more commonly used file systems:

File System	Max File Size
FAT12	16 MiB
FAT16	2 GiB
FAT32	4 GiB
NTFS	2 TiB
ext2	2 TiB
ext3	2 TiB
ext4	16 TiB
HFS	2 GiB

File System File Access

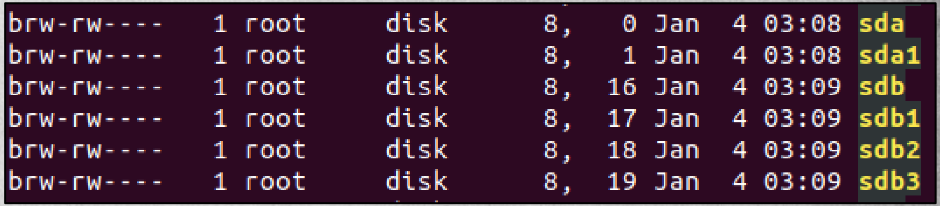
- When a disk is partitioned, data structures are generated based on which file system is selected
- To access files, a file system works on data sectors
  - ✓ A traditional sectors is 512 bytes long, but larger sectors are configurable (i.e. 4K)
  - ✓ A block of data is made up of between 1 to 128 sectors
- Data is placed in a disk volume based on an offset address
- A file system accounts for
  - ✓ Which sectors are occupied and free
  - ✓ File sizes
  - ✓ Filenames

Essential File System Data

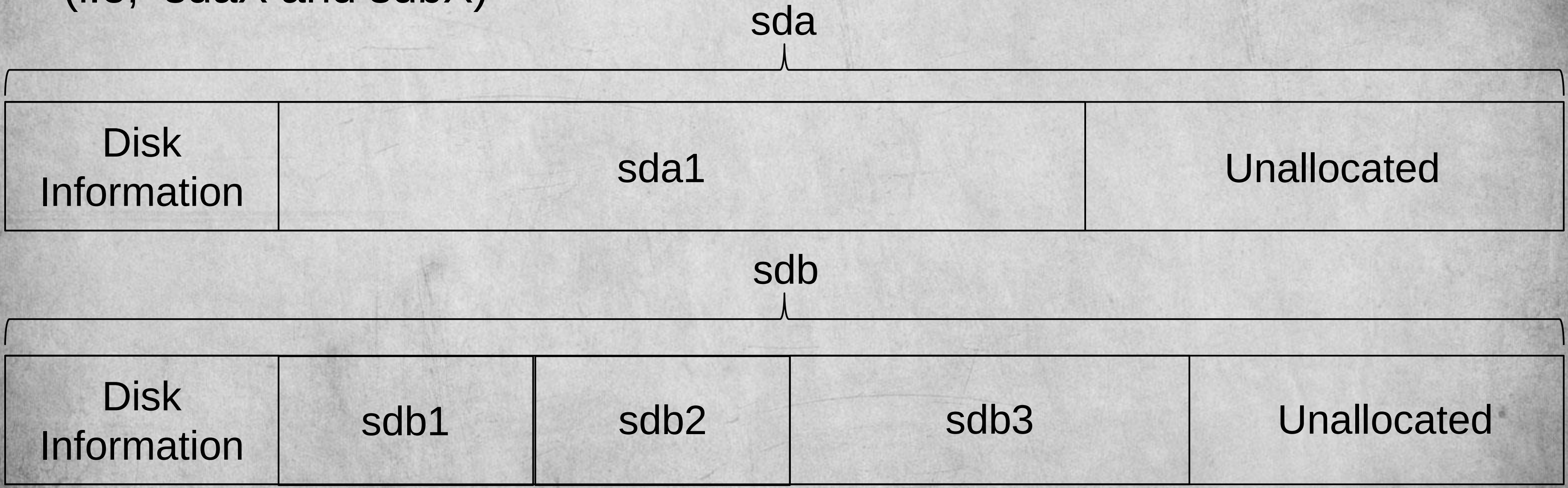
- Essential file system data are elements necessary to saving and retrieving files
  - ✓ Data reference addresses
  - ✓ Filenames
  - ✓ Metadata pointers
- Non-essential file system data are not necessary to saving and retrieving files
  - ✓ File access times
  - ✓ File permissions
- A forensics analysis requires understanding of
  - ✓ Which file system data structures are used to store data
  - ✓ Which operating system wrote the data to the file system

Disks and Partitions

- Disks are made up of formatted partitions and unallocated space
- File systems like FAT and NTFS will be accessible as partitions (i.e, sdaX and sdbX)



```
brw-rw----   1 root     disk      8,   0 Jan  4 03:08 sda
brw-rw----   1 root     disk      8,   1 Jan  4 03:08 sda1
brw-rw----   1 root     disk      8,  16 Jan  4 03:09 sdb
brw-rw----   1 root     disk      8,  17 Jan  4 03:09 sdb1
brw-rw----   1 root     disk      8,  18 Jan  4 03:09 sdb2
brw-rw----   1 root     disk      8,  19 Jan  4 03:09 sdb3
```



sda

Disk Information | sda1 | Unallocated

sdb

Disk Information | sdb1 | sdb2 | sdb3 | Unallocated

File Allocation Table (FAT) File Systems

FAT File System

- The File Allocation Table file system is a basic file system and principally used by legacy Windows OS’s
  - ✓ Microsoft DOS
  - ✓ Windows 9X
- FAT file systems use a simple allocation table to track data based on data clusters
- FAT is supported by all Windows OS’s, most Unix-based OS’s, compact flash cards, and USB drives

FAT Versions

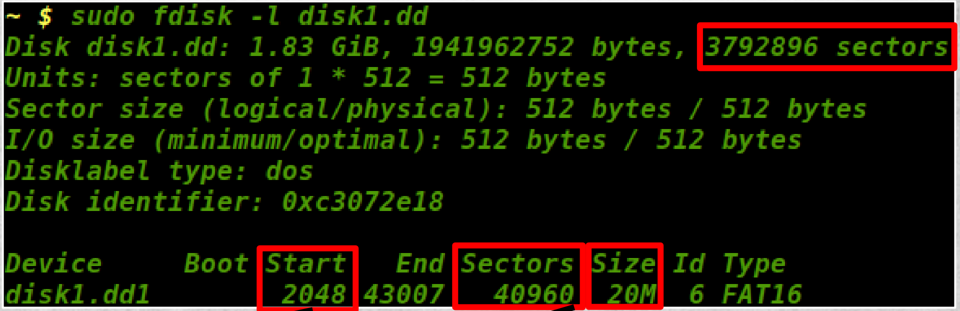
- FAT12: 12-bit FAT entries
  - ✓ Legacy file system for small disks and form factors
- FAT16: 16-bit FAT entries
  - ✓ Improvement on FAT12 by expanding table entries to 16-bits
- FAT32: 28-bit FAT entries
  - ✓ Generally used to format portable devices
  - ✓ Permits file fragmentation

File System	Max File Size	Partition Size	Cluster Size	File Permissions
FAT12	16 MB	< 16 MB	512 bytes - 4 KB	No
FAT16	2 GiB	16 MB - 2 GB	2 KB - 32 KB	No
FAT32	4 GiB	128 MB - 2 TB	2 KB - 32 KB	No

Displaying Disk and Partition Information *

- For this session we will be using disk image "disk1.dd"

1 Sector = 512 bytes



```
~ $ sudo fdisk -l disk1.dd
Disk disk1.dd: 1.83 GiB, 1941962752 bytes, 3792896 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device     Boot Start   End Sectors Size Id Type
disk1.dd1        2048 43007   40960  20M  6 FAT16
```

3792896 Sectors * 512 Bytes

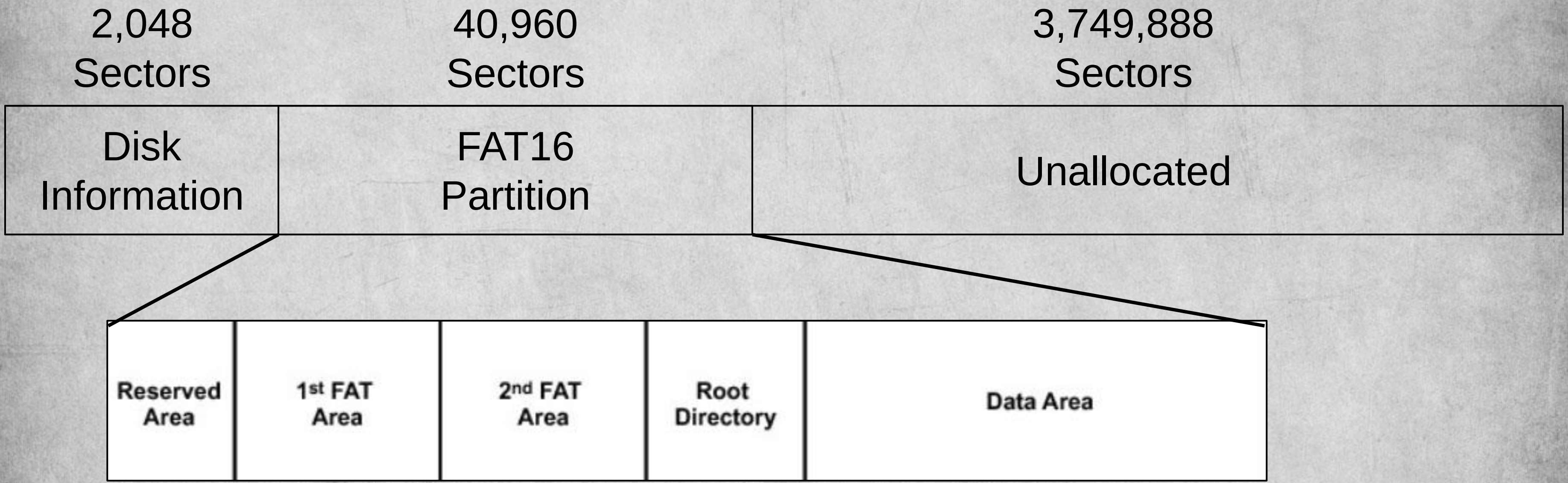
=

1.83 GiB

2,048 Sectors	40,960 Sectors	3,749,888 Sectors
Disk Information	FAT16 Partition	Unallocated

3,792,896 Sectors

FAT16 Partition Architecture



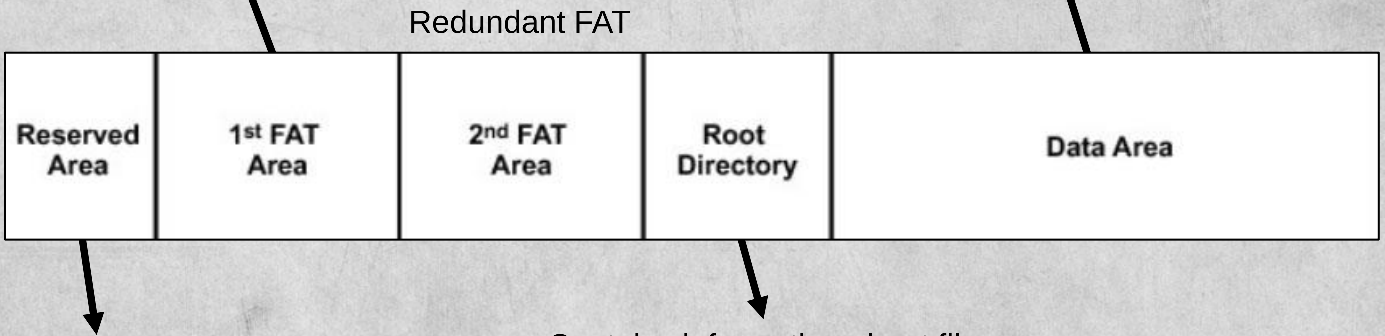
2,048 Sectors	40,960 Sectors	3,749,888 Sectors
Disk Information	FAT16 Partition	Unallocated

Reserved Area	1st FAT Area	2nd FAT Area	Root Directory	Data Area

FAT16 Partition Details

- 16-bit entry with one entry per cluster
  - ✓ Free Cluster (0x0000)
  - ✓ Occupied Cluster
    - ▪ 0x0002 - 0xFFEF
  - ✓ Reserved Cluster (0xFFF6)
  - ✓ Bad Cluster (0xFFF7)
  - ✓ End of Cluster (0xFFFF)

- The contents of files are stored in clusters in the data area
- Provides addresses for files and directories

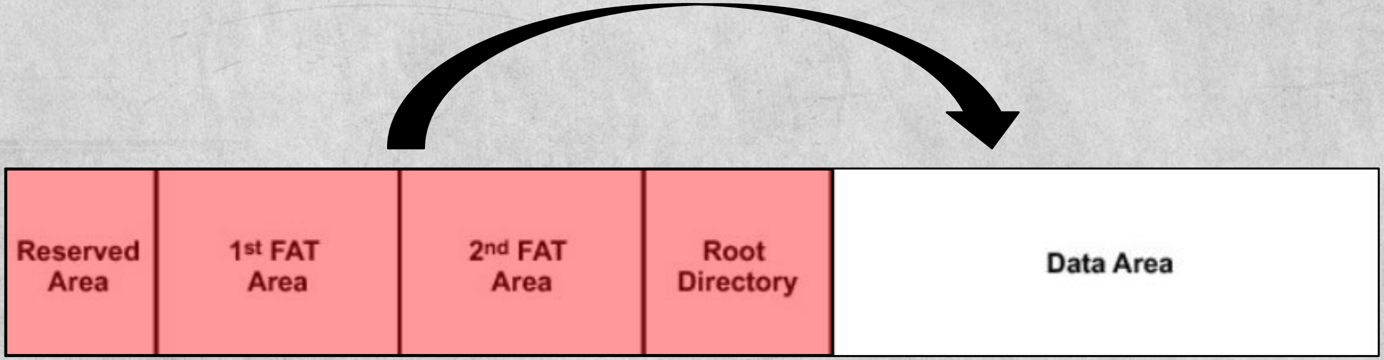


- Boot sector code
  - ✓ # FATs
  - ✓ Sector Size
  - ✓ Sectors / Cluster

- Contains information about files
  - Filenames
  - File Extension
  - File Attributes
  - MAC Times
  - File Starting Cluster
  - File Size

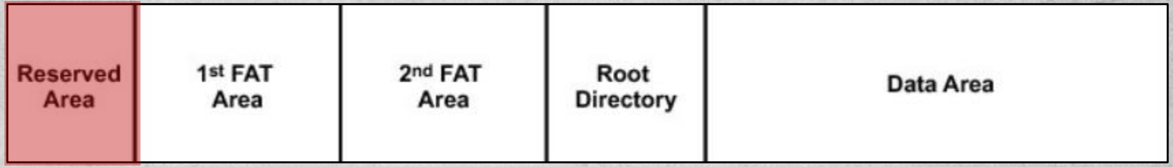
FAT16 Data Recovery Process

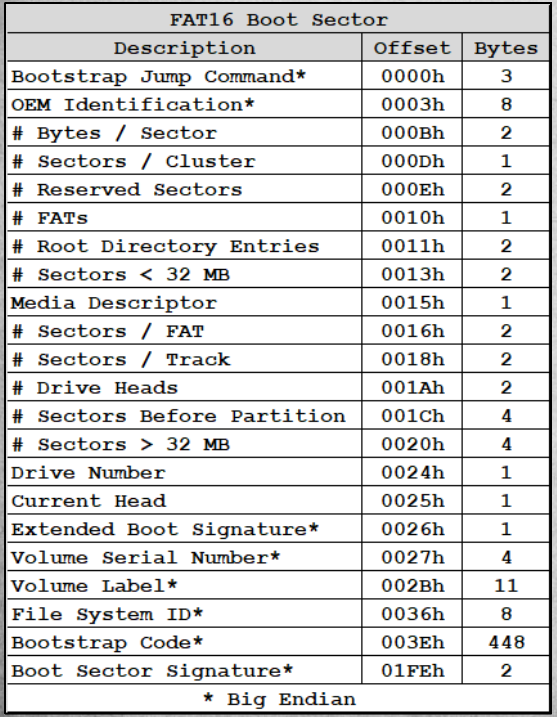
- By collecting information from each of the first 4 data structures in a FAT16 partition, we will be able to recover data in the data area!



FAT16 Reserved Area

FAT16 Reserved Area Structure

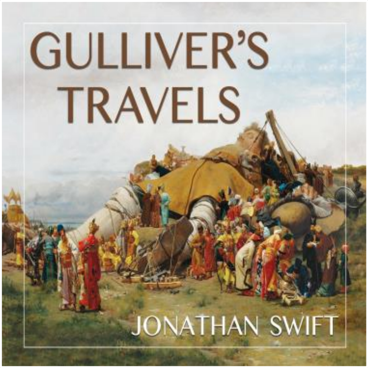




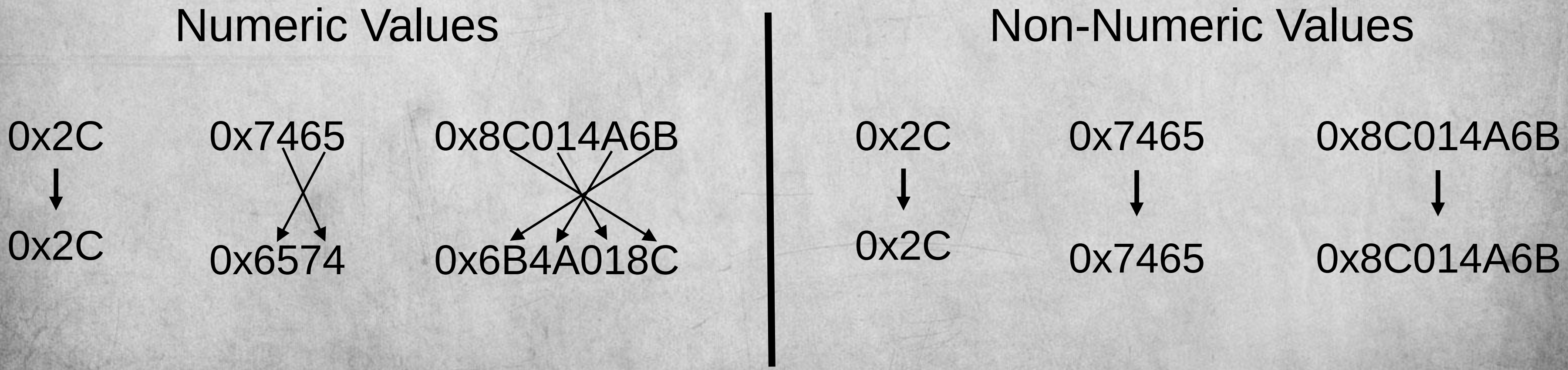
FAT16 Boot Sector		
Description	Offset	Bytes
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature*	0026h	1
Volume Serial Number*	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

- The reserved area of a FAT16 partition is a block of space that includes a boot sector and other protected data generated during the formatting process
- Some of the key parameters specified in the boot sector:
  - ✓ Partition Size
  - ✓ FAT lengths
  - ✓ # FAT’s
  - ✓ # Root Directory Entries
  - ✓ Data Area Length
- When conducting analysis of a partition, it is necessary to have a solid understanding of endian encoding

Endian



- Endian has to do with the byte order of data
  - ✓ Big Endian (BE) – Most Significant Byte First
  - ✓ Little Endian (LE) – Least Significant Byte First
- Numeric values are generally represented “little endian”, whereas non-numeric values are “big endian”



Numeric Values

0x2C → 0x2C

0x7465 → 0x6574

0x8C014A6B → 0x6B4A018C

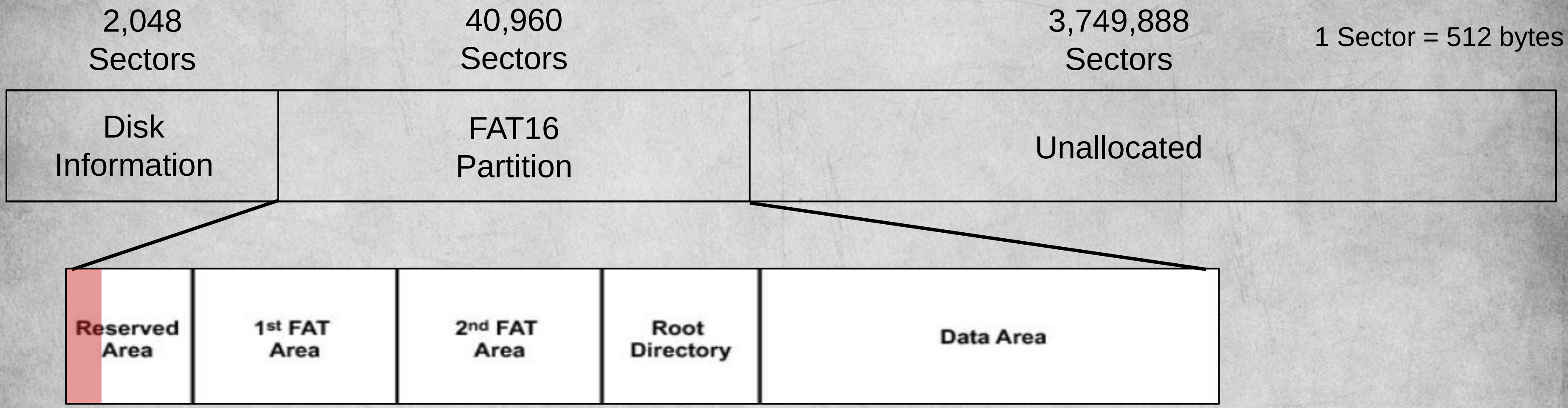
Non-Numeric Values

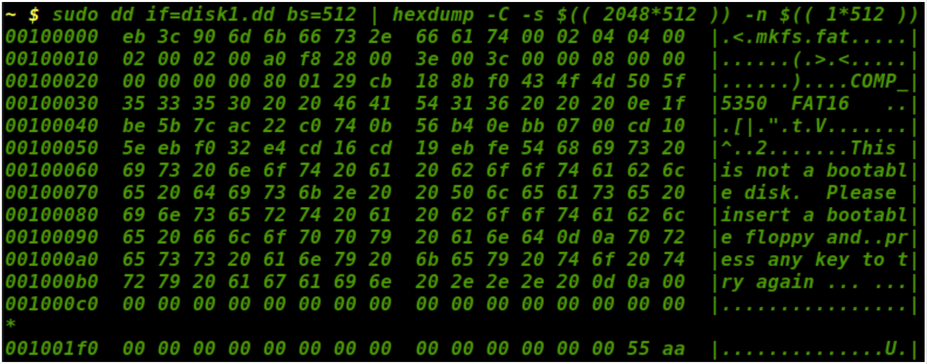
0x2C → 0x2C

0x7465 → 0x7465

0x8C014A6B → 0x8C014A6B

Displaying Boot Sector Contents



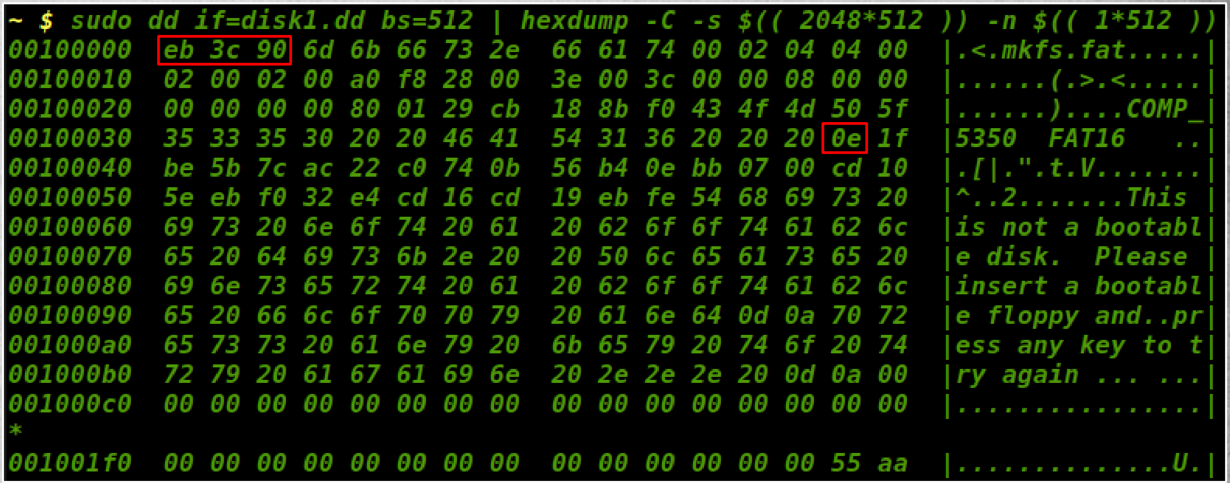


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

- Skip 2048 sectors into the disk and display 1 sector

Boot Sector – Bootstrap Jump Command *

FAT16 Boot Sector		
Description	Offset	Bytes
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		



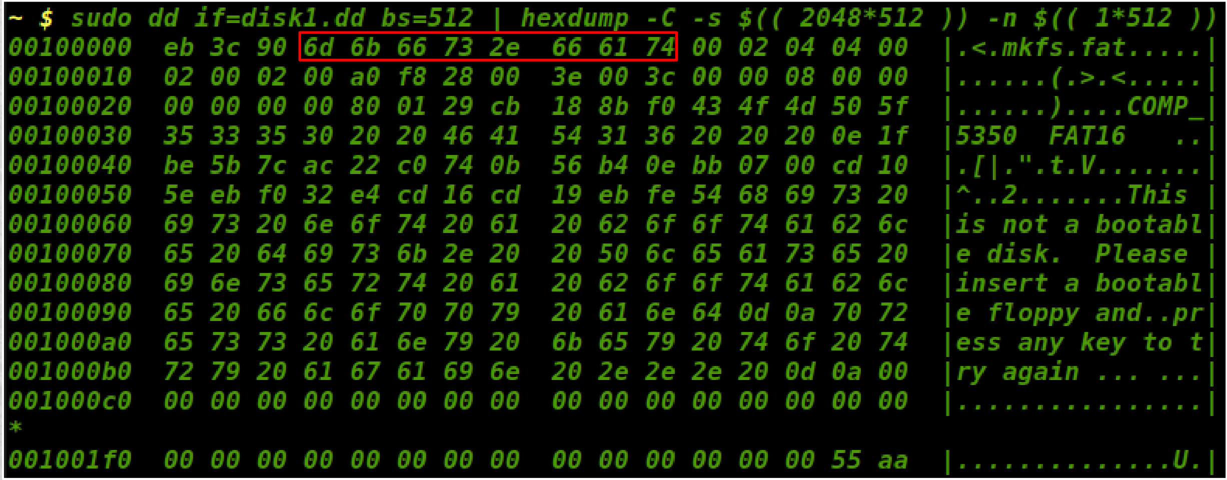
```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0xEB3C90: Jump Short 3C NOP

- If bootable, jump 60 bytes to the start of boot code

Boot Sector – OEM ID

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

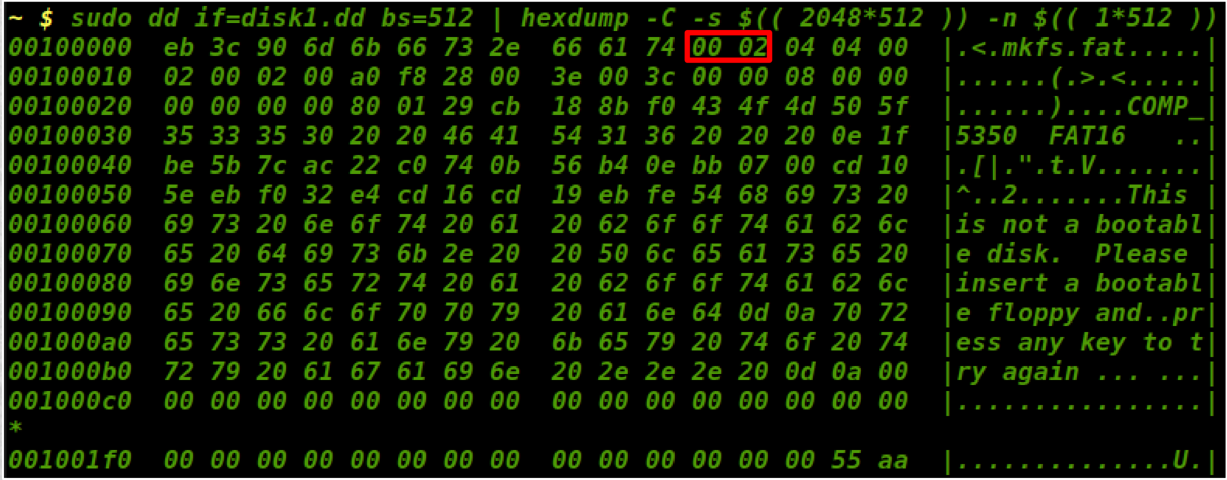


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x6D6B66732E666174: mkfs.fat

Boot Sector – # Bytes / Sector

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

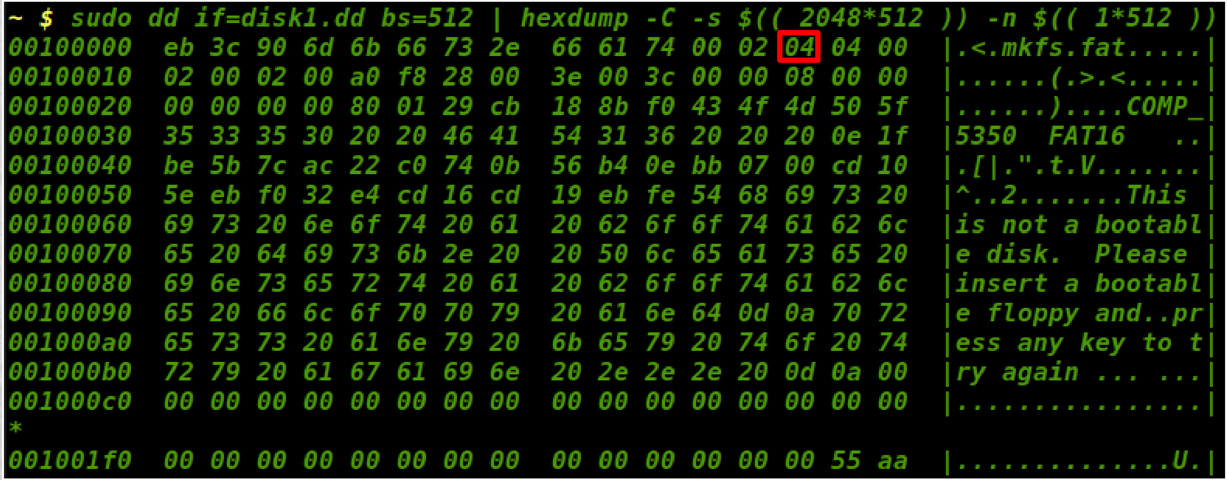
0x200: 512 Bytes / Sector

Boot Sector – # Sectors / Cluster

FAT16 Boot Sector

Description	Offset	Bytes
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	**000Dh**	**1**
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2

* Big Endian



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x4: 4 Sectors / Cluster

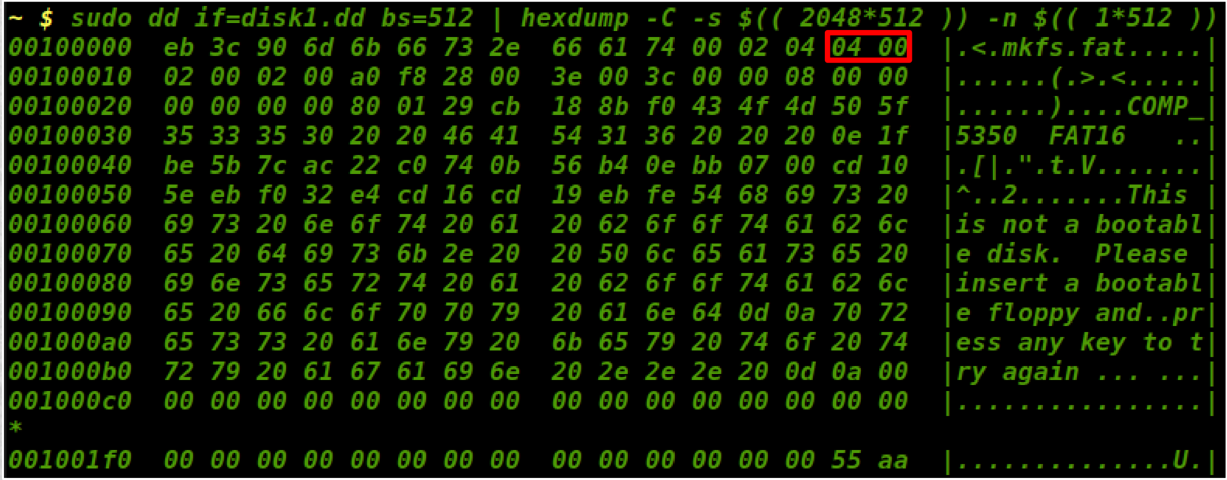
$$512\frac{bytes}{sector} * 4\frac{sectors}{cluster} = 2048\frac{bytes}{cluster}$$

Boot Sector – # Reserved Sectors

FAT16 Boot Sector

Description	Offset	Bytes
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2

* Big Endian

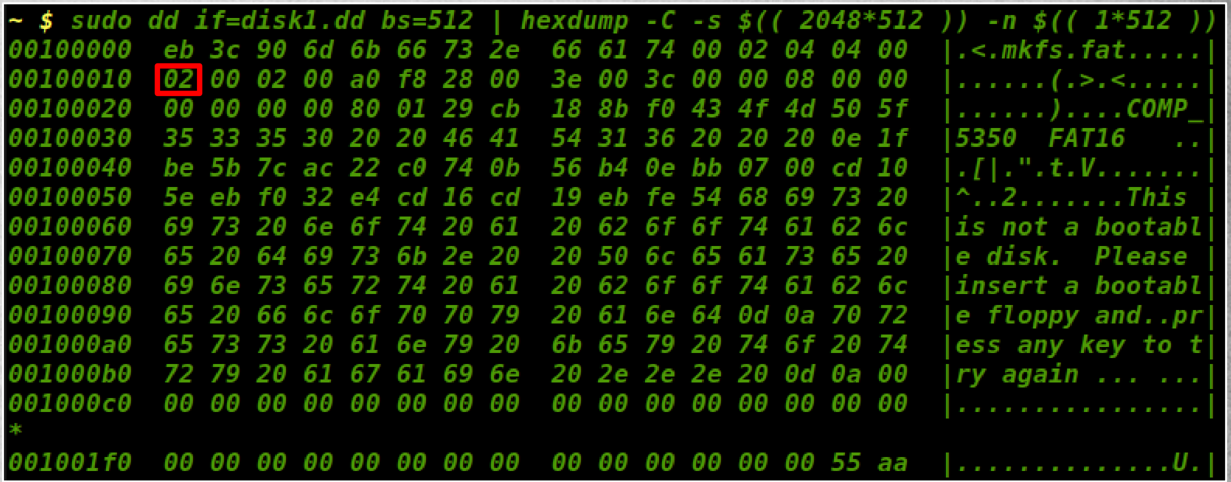


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x4: 4 Reserved Sectors – 2 KB

Boot Sector – # File Allocation Tables

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

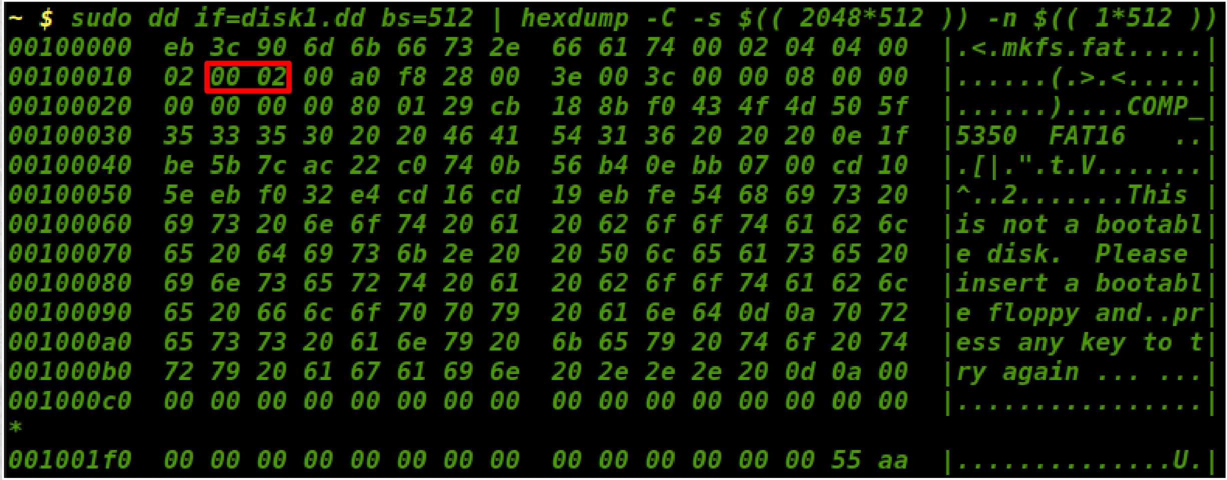


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x2: 2 File Allocation Tables

Boot Sector – # Root Directory Entries

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		



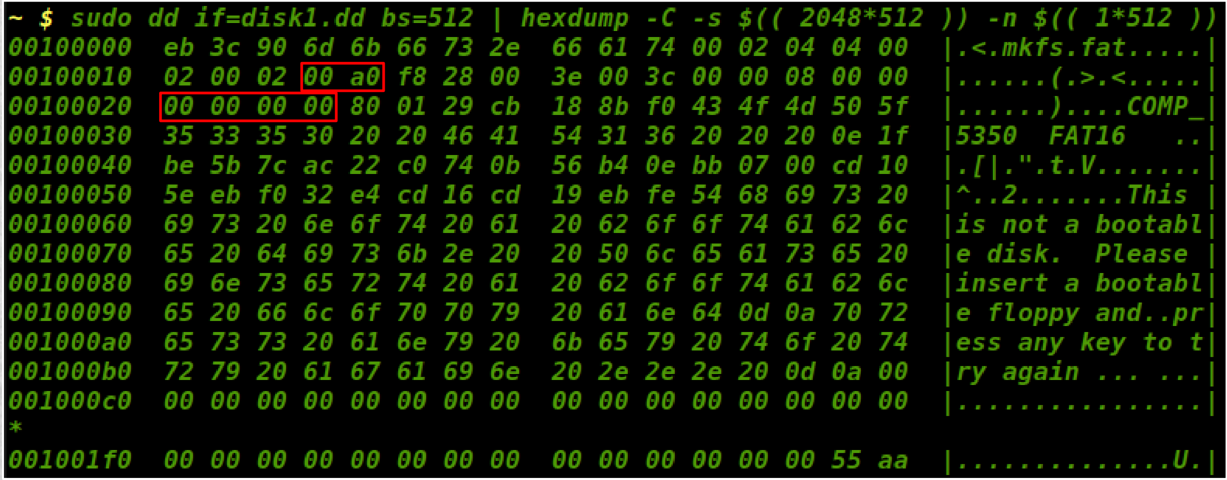
```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x200: 512 Directory Entries

Boot Sector – # Sectors

FAT16 Boot Sector		
Description	Offset	Bytes
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

- Total number of sectors in a partition less than 32 MB



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0xA000: 40960 Sectors

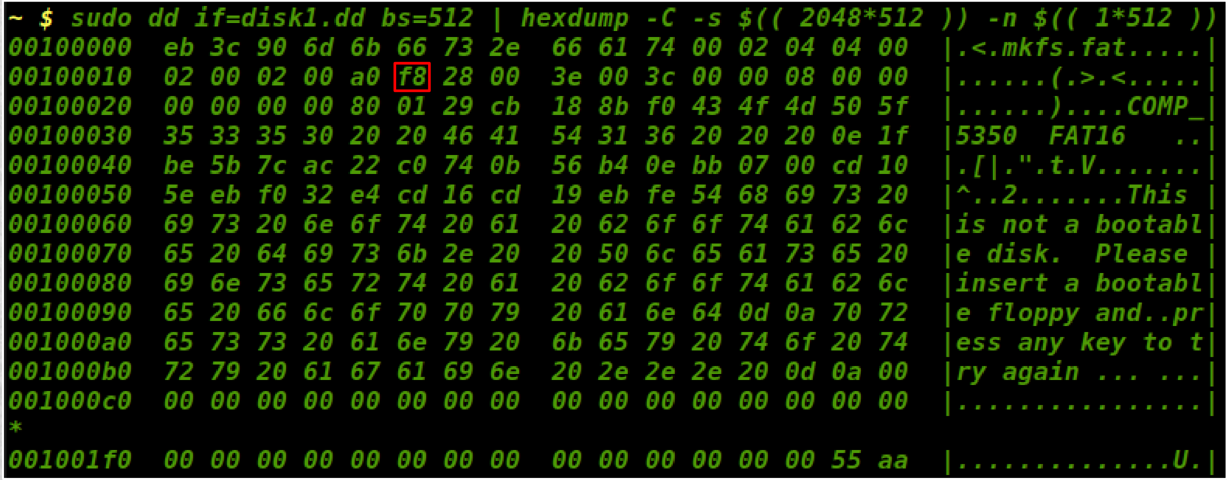
$$40960\sec tors * 512\frac{bytes}{\sec tor} = 20MB$$

Boot Sector – Media Descriptor

FAT16 Boot Sector

Description	Offset	Bytes
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2

*** Big Endian**

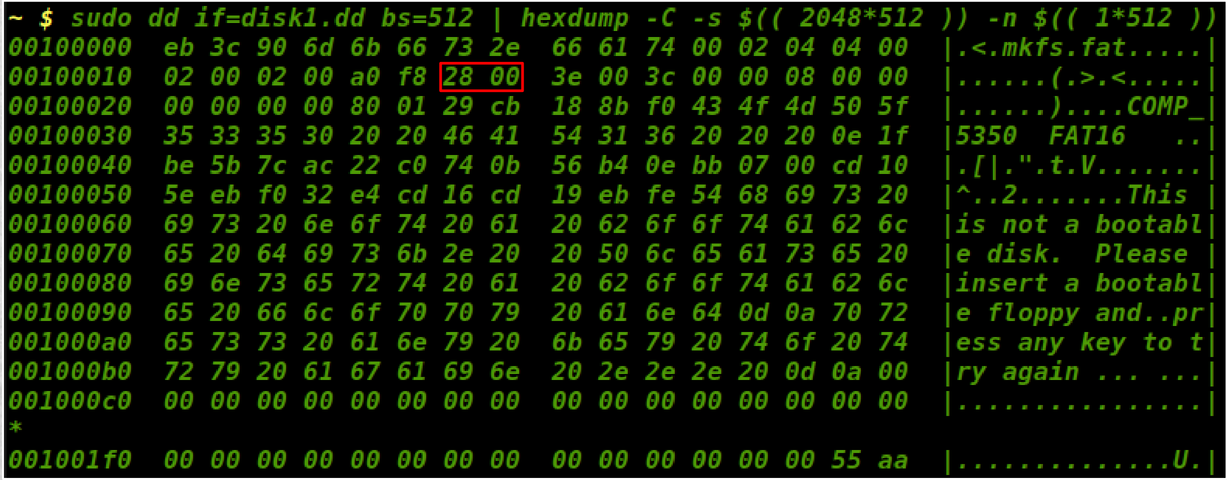


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0xF8: Fixed Disk – Hard Disk Partition

Boot Sector – # Sectors / FAT

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

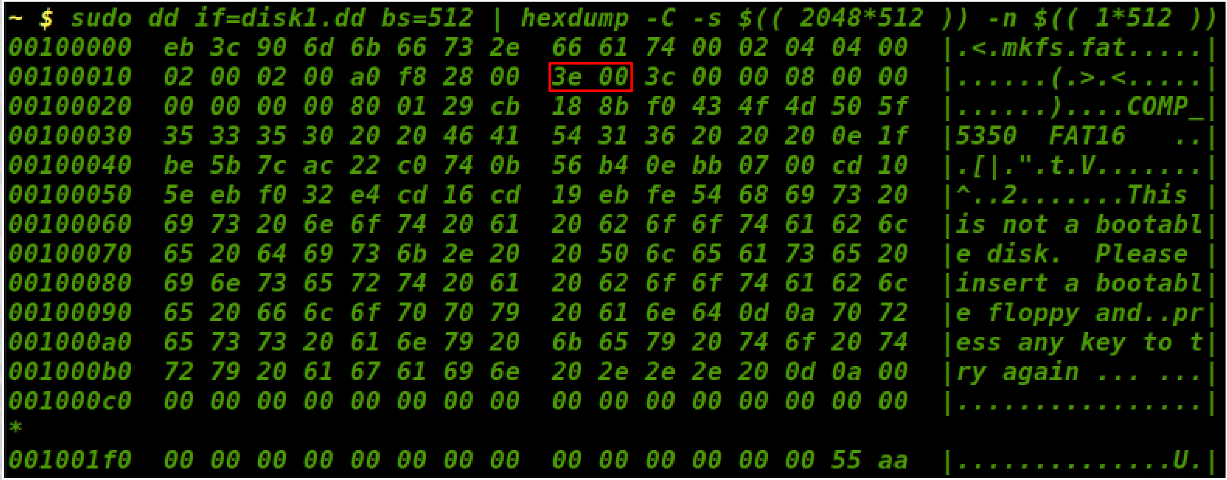


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x28: 40 Sectors / FAT

Boot Sector – # Sectors / Track

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

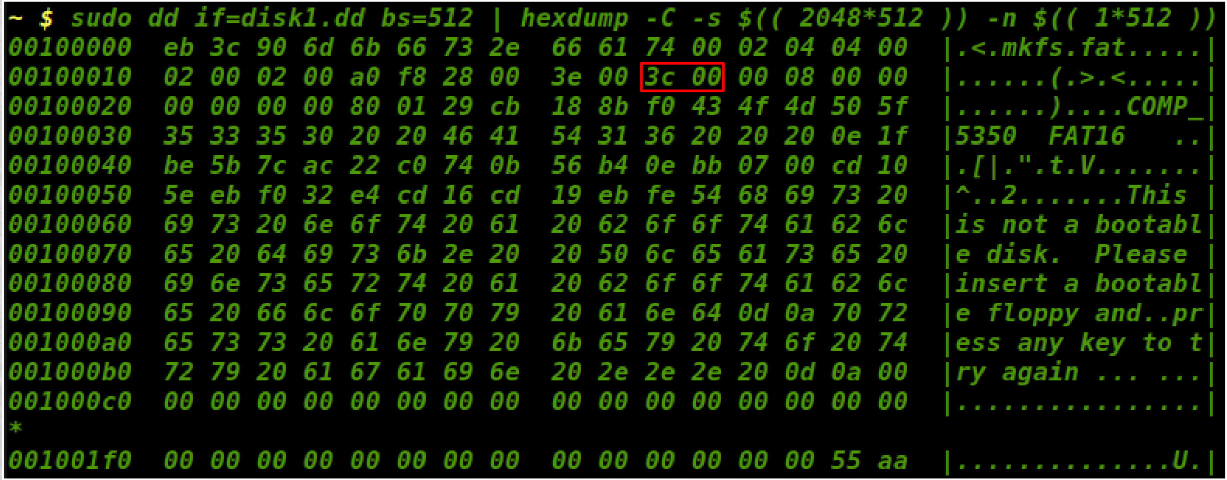


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x3E: 62 Sectors / Track

Boot Sector – # Storage Media Heads

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

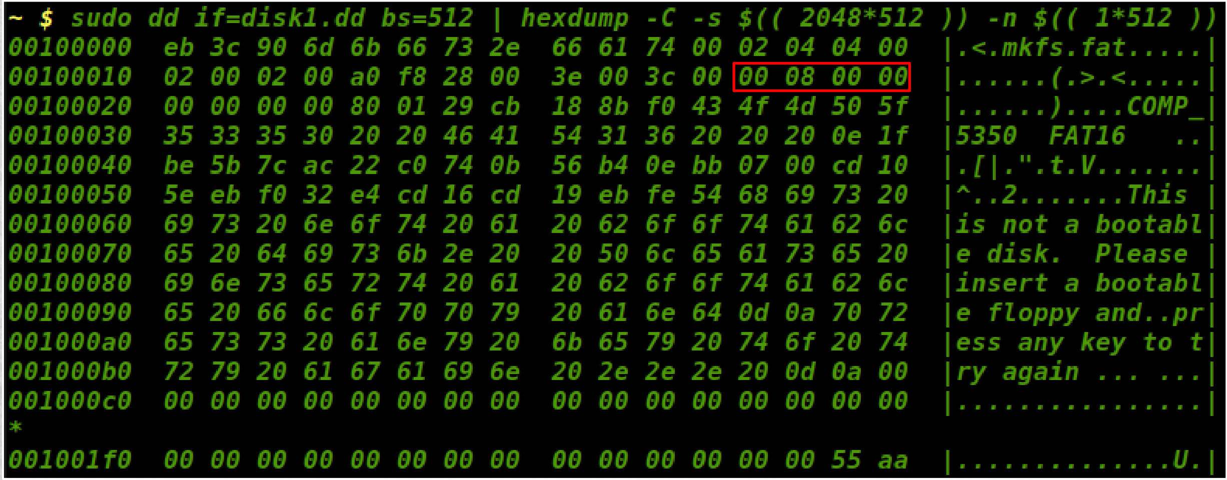


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x3C: 60 Storage Media Heads

Boot Sector – # Sectors Before Partition

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		



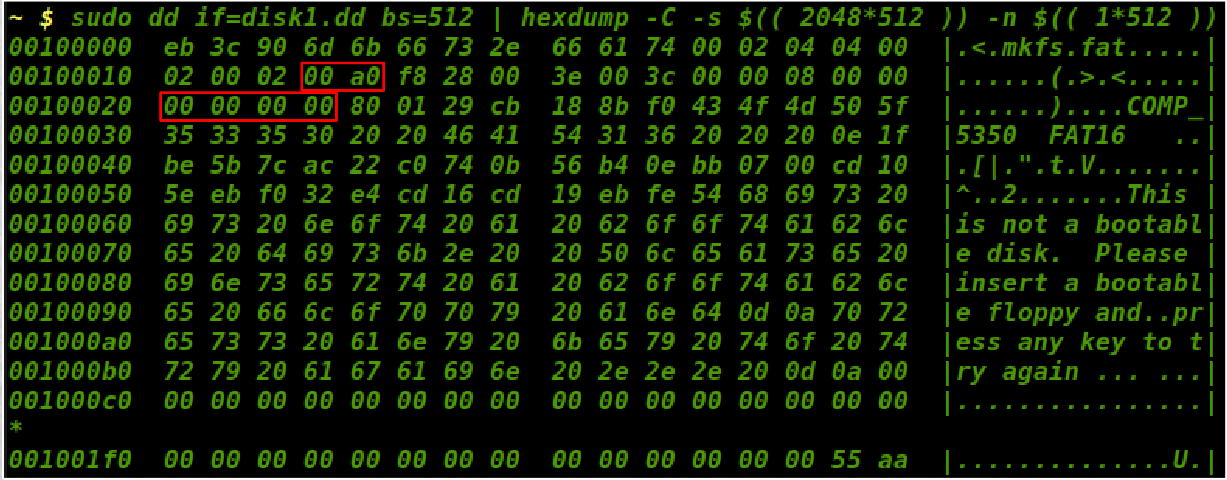
```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x800: 2048 Sectors Before FAT16 Starts

Boot Sector – # Sectors

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	**0000h**	**3**
OEM Identification*	**0003h**	**8**
# Bytes / Sector	**000Bh**	**2**
# Sectors / Cluster	**000Dh**	**1**
# Reserved Sectors	**000Eh**	**2**
# FATs	**0010h**	**1**
# Root Directory Entries	**0011h**	**2**
# Sectors < 32 MB	**0013h**	**2**
Media Descriptor	**0015h**	**1**
# Sectors / FAT	**0016h**	**2**
# Sectors / Track	**0018h**	**2**
# Drive Heads	**001Ah**	**2**
# Sectors Before Partition	**001Ch**	**4**
# Sectors > 32 MB	**0020h**	**4**
Drive Number	**0024h**	**1**
Current Head	**0025h**	**1**
Extended Boot Signature	**0026h**	**1**
Volume Serial Number	**0027h**	**4**
Volume Label*	**002Bh**	**11**
File System ID*	**0036h**	**8**
Bootstrap Code*	**003Eh**	**448**
Boot Sector Signature*	**01FEh**	**2**
*** Big Endian**		

- Total number of sectors in a partition greater than 32 MB
- Since this partition is 20 MB, these 4 bytes will be zero



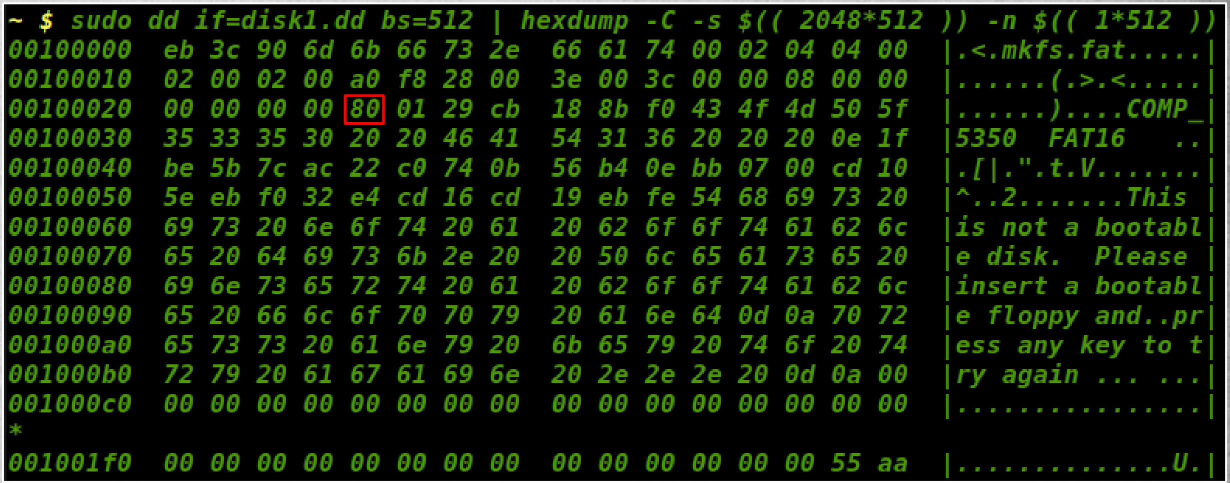
```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x0: 0 Sectors > 32 MB

$$40960\,sec\,tors * 512\frac{bytes}{sec\,tor} = 20MB$$

Boot Sector – Drive Number

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

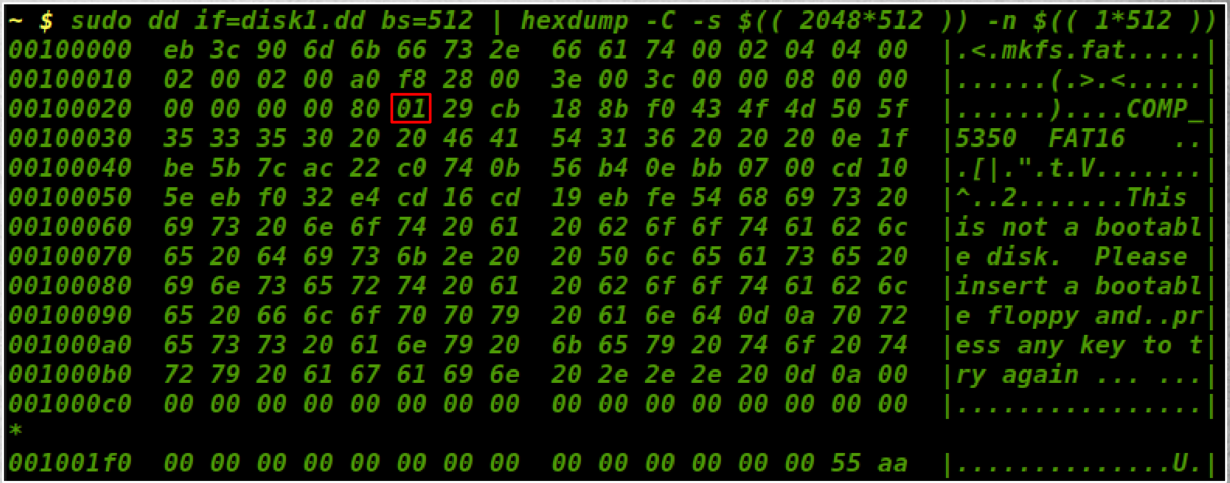


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x80: Physical Hard Disk

Boot Sector – Current Head*

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

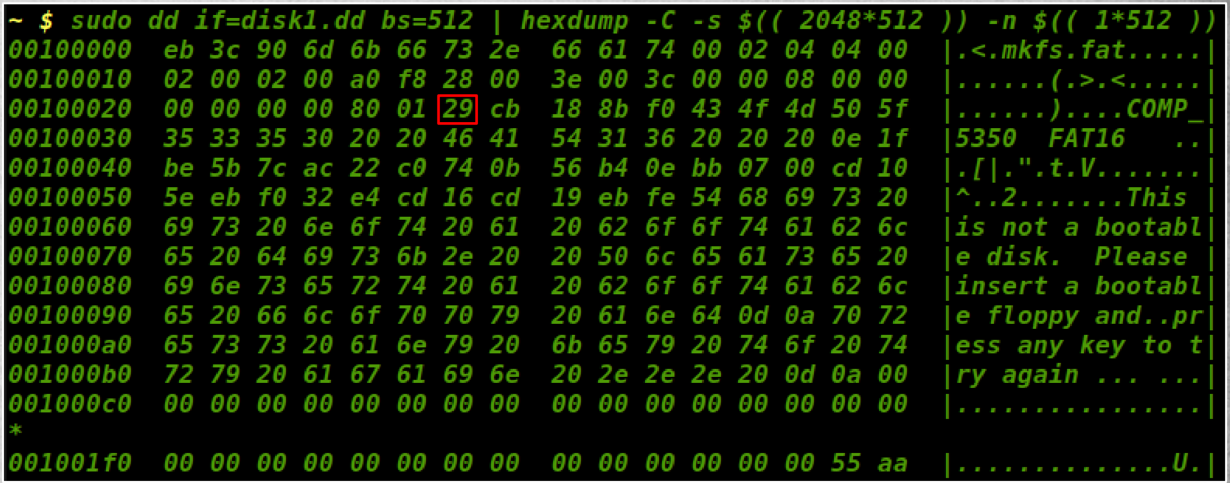


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

* Not Used

Boot Sector – Extended Boot Signature

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

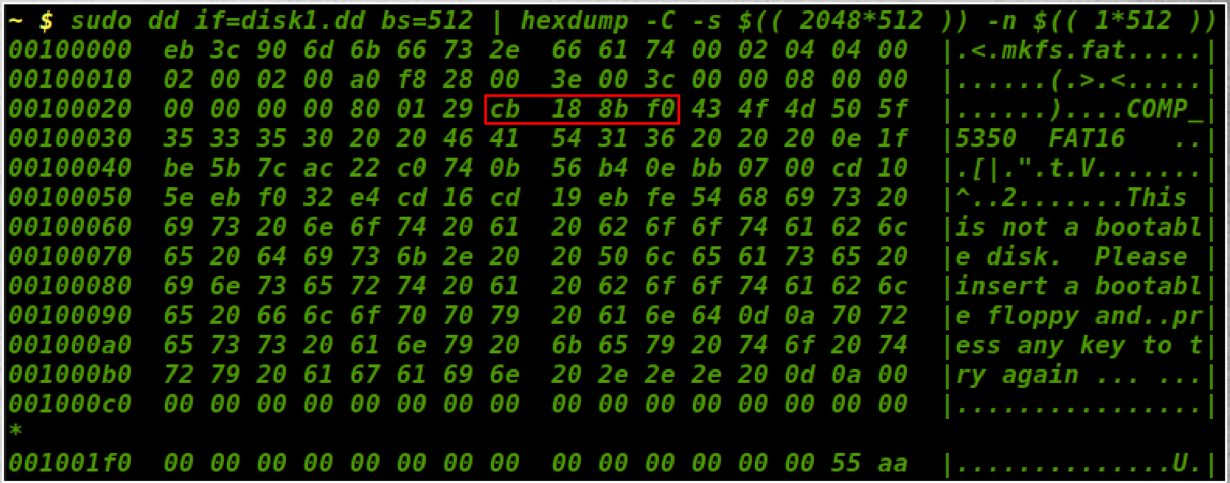


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

Set to either 0x28 or 0x29 for Windows NT

Boot Sector – Volume Serial Number

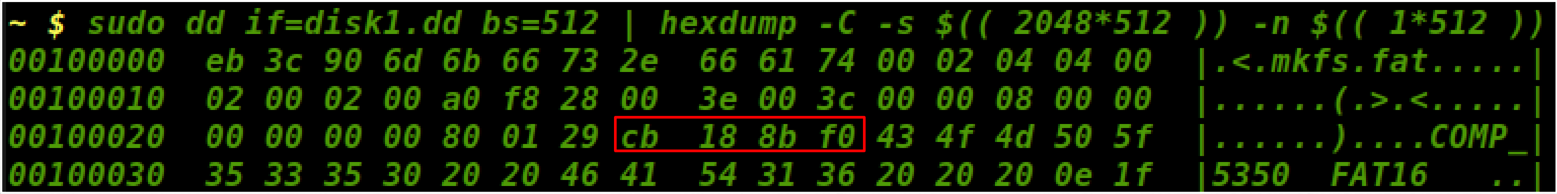
FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

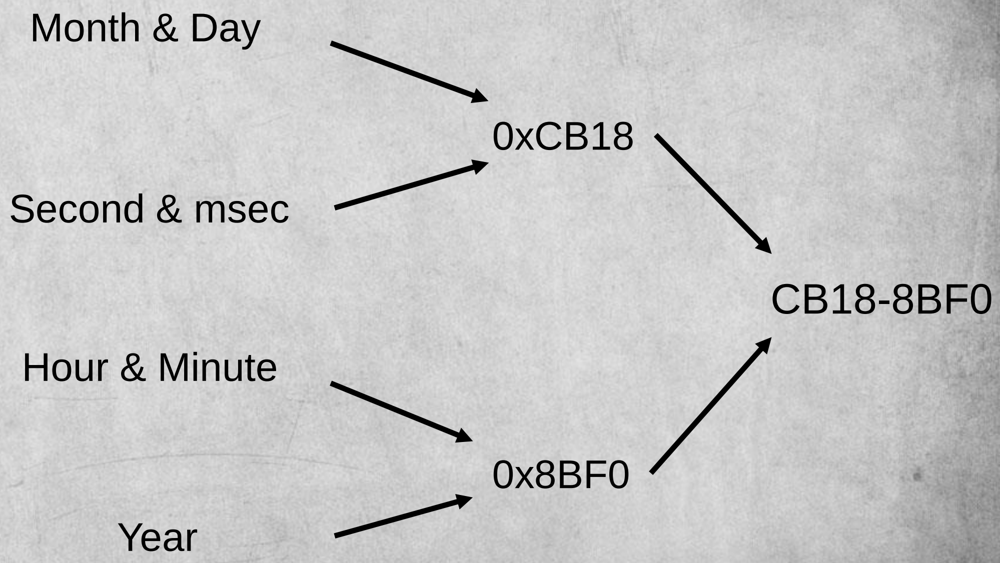
0xCB188BF0: CB18-8BF0

Volume Serial Number Derivation



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
```

- When a FAT volume is created, a “unique” value is generated based on computer local time
- The volume serial number (VSN) is generated combining local timestamp information
  - ✓ Month and Day
  - ✓ Second and msec
  - ✓ Hour and Minute
  - ✓ Year
- VSN values are non-essential



Month & Day

Second & msec

0xCB18

Hour & Minute

Year

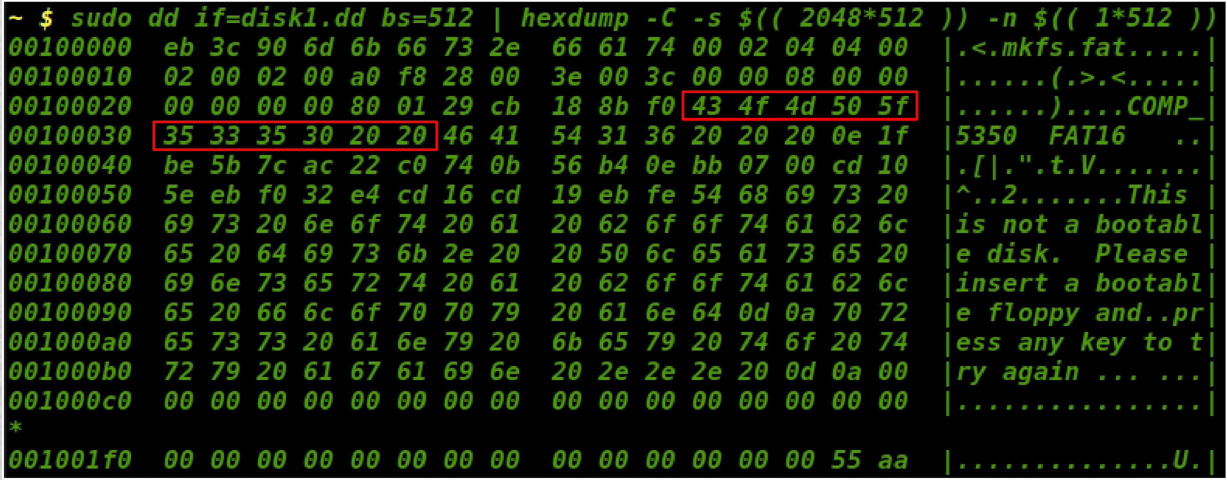
0x8BF0

CB18-8BF0

Boot Sector – Volume Label

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

- The volume label is a maximum of 11 characters
- Any label less than 11 characters is padded with %20

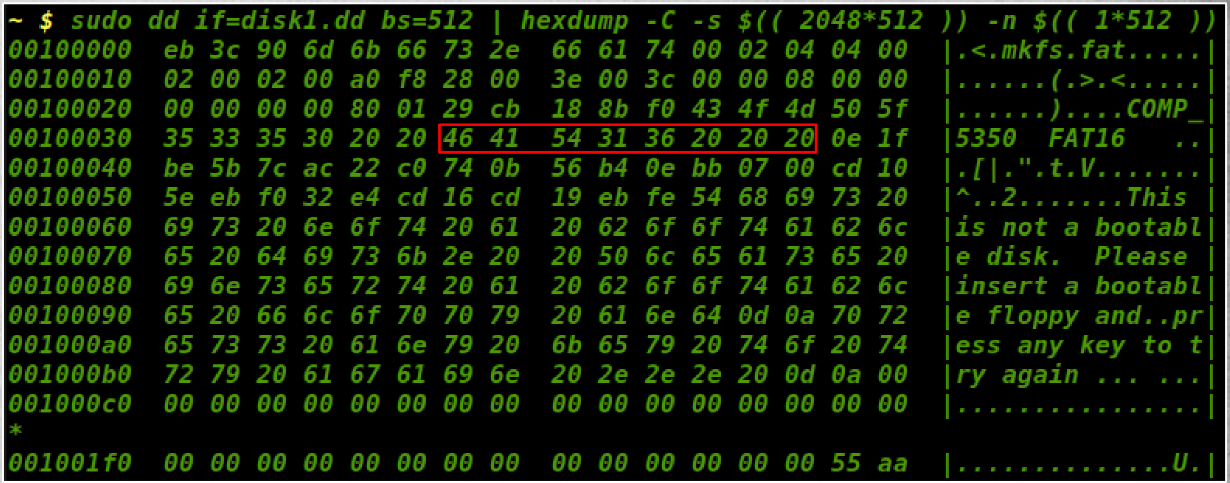


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x434F4D505F353335302020: “COMP_5350 "

Boot Sector – File System Identifier

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

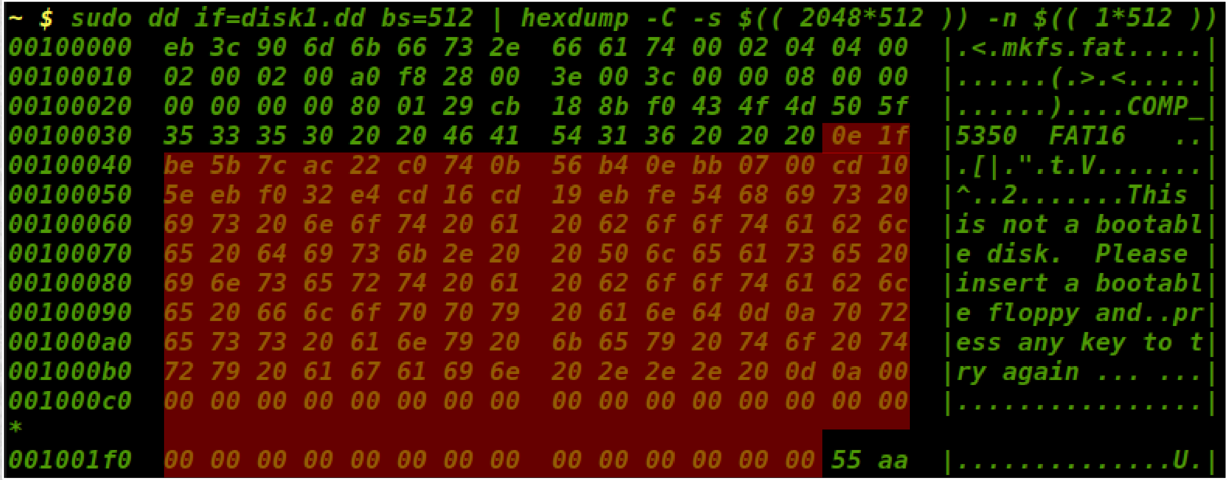


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x4641543136202020: "FAT16   "

Boot Sector – Boot Code

FAT16 Boot Sector		
Description	Offset	Bytes
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		

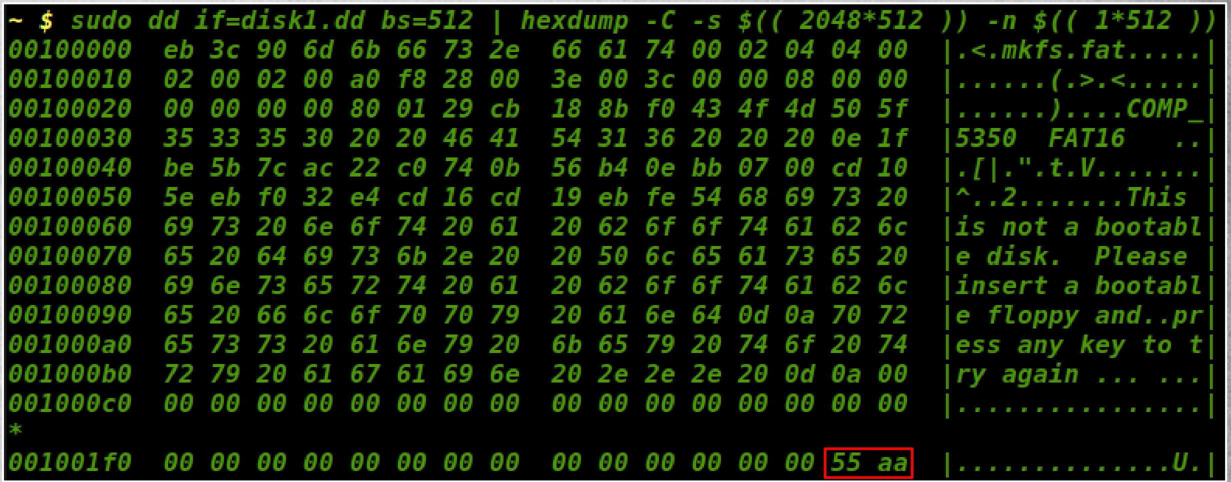


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

- If the partition is bootable, then this code will run

Boot Sector – End of Sector Signature

FAT16 Boot Sector		
Description	**Offset**	**Bytes**
Bootstrap Jump Command*	0000h	3
OEM Identification*	0003h	8
# Bytes / Sector	000Bh	2
# Sectors / Cluster	000Dh	1
# Reserved Sectors	000Eh	2
# FATs	0010h	1
# Root Directory Entries	0011h	2
# Sectors < 32 MB	0013h	2
Media Descriptor	0015h	1
# Sectors / FAT	0016h	2
# Sectors / Track	0018h	2
# Drive Heads	001Ah	2
# Sectors Before Partition	001Ch	4
# Sectors > 32 MB	0020h	4
Drive Number	0024h	1
Current Head	0025h	1
Extended Boot Signature	0026h	1
Volume Serial Number	0027h	4
Volume Label*	002Bh	11
File System ID*	0036h	8
Bootstrap Code*	003Eh	448
Boot Sector Signature*	01FEh	2
* Big Endian		



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x55AA: End of Sector Signature

FAT16 Partition Map

Bytes / Sector
0x200 → 512

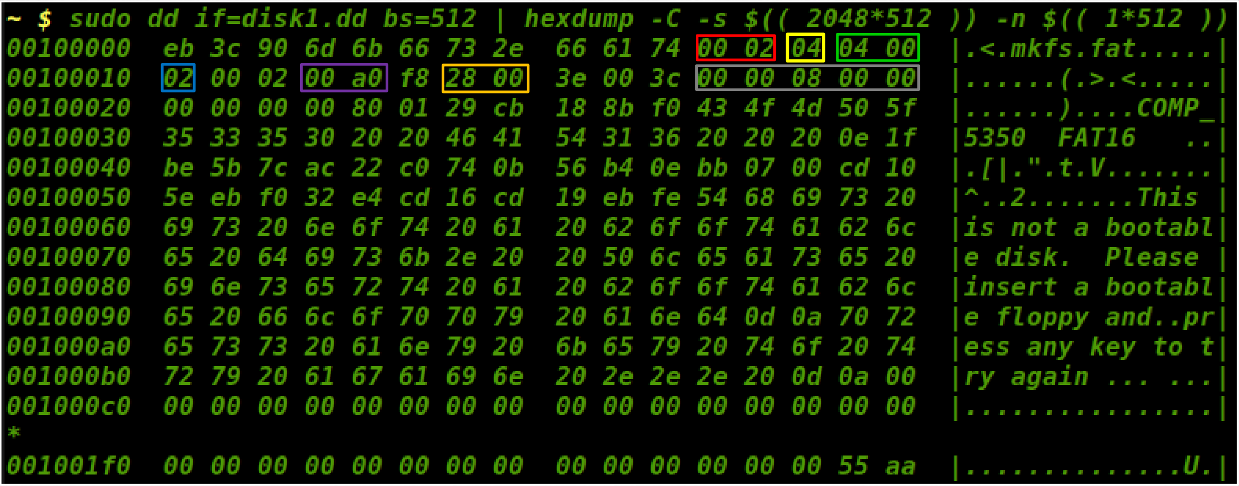
Sectors / Cluster
0x4 → 4

Reserved Sectors
0x4 → 4

# FAT's
0x2 → 2

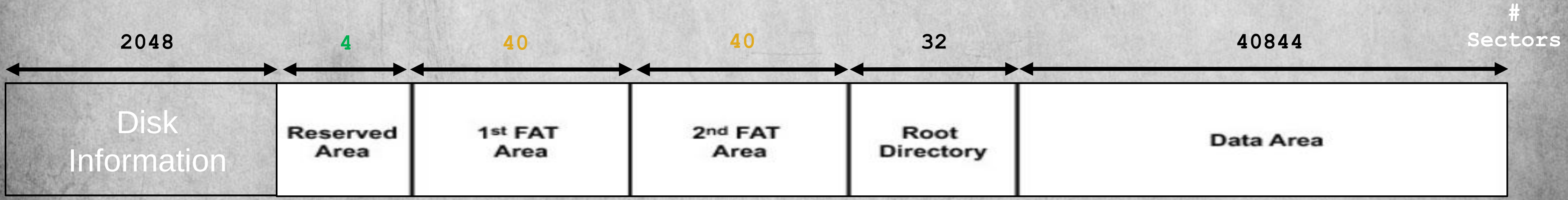
# Sectors
0xA000 → 40960

# Sectors / FAT
0x0028 → 40

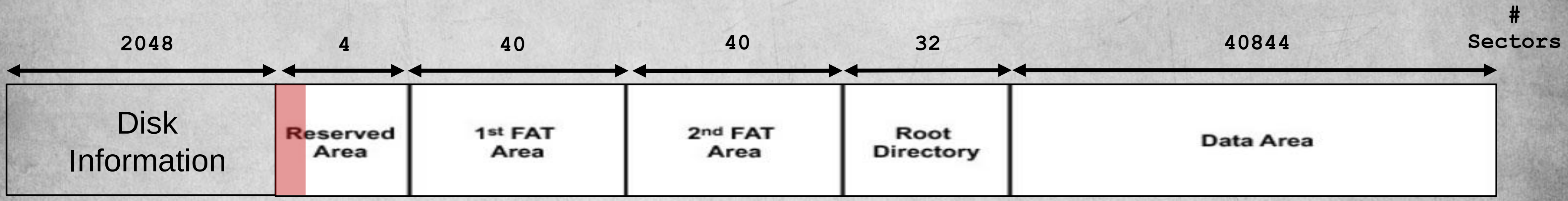


```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

# Sectors Before Partition
0x800 → 2048

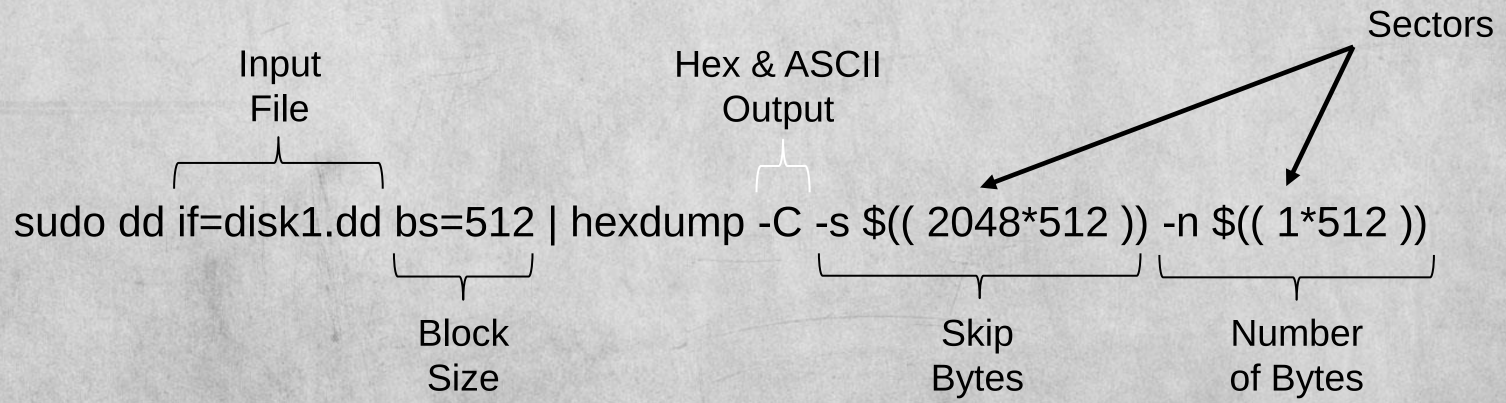


Accessing FAT16 Partitions



2048 | 4 | 40 | 40 | 32 | 40844 | # Sectors

Disk Information | Reserved Area | 1st FAT Area | 2nd FAT Area | Root Directory | Data Area



```
sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
```

Input File

Block Size

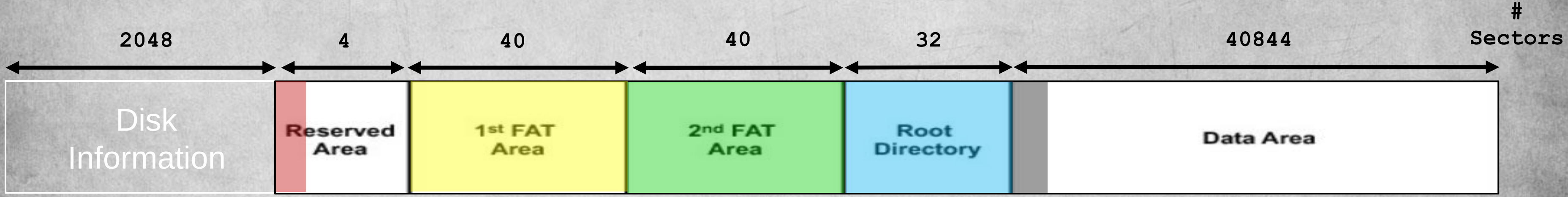
Hex & ASCII Output

Skip Bytes

Number of Bytes

Sectors

Accessing FAT16 Partitions



Reserved Area: sudo dd if=disk1.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))

1st FAT: sudo dd if=disk1.dd bs=512 | hexdump -C -s $((2052*512)) -n $((40*512))

2nd FAT: sudo dd if=disk1.dd bs=512 | hexdump -C -s $((2092*512)) -n $((40*512))

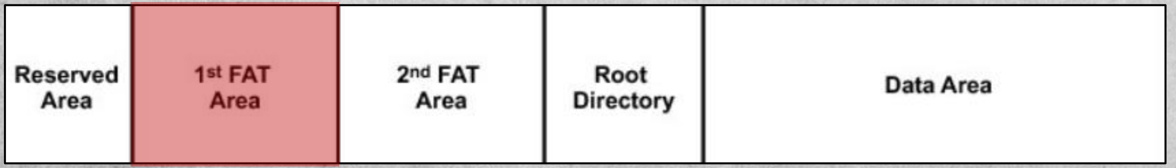
Root Directory: sudo dd if=disk1.dd bs=512 | hexdump -C -s $((2132*512)) -n $((32*512))

Data Area: sudo dd if=disk1.dd bs=512 | hexdump -C -s $((2164*512)) -n $((4*512))

FAT16 File Allocation Table

File Allocation Table

- A file allocation table provides a lists of data clusters that hold file contents
- The FAT only provides data cluster information and does not provide critical metadata such as filenames, extensions, and file sizes, which are provided in the root directory
- The clusters in a FAT will provide consecutive values until an end of file (EOF) marker is found
  - ✓ 0xFFFF
- The FAT file system has a redundant FAT to ensure integrity



Reserved Area

1st FAT Area

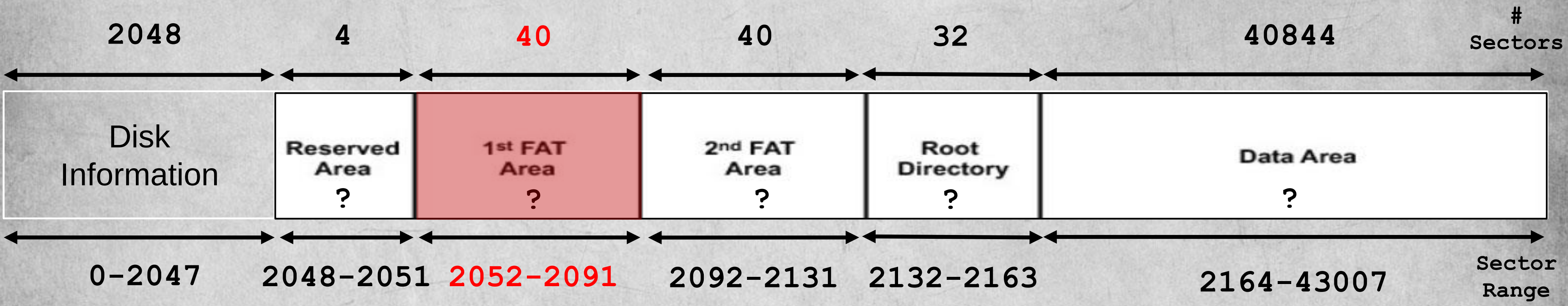
2nd FAT Area

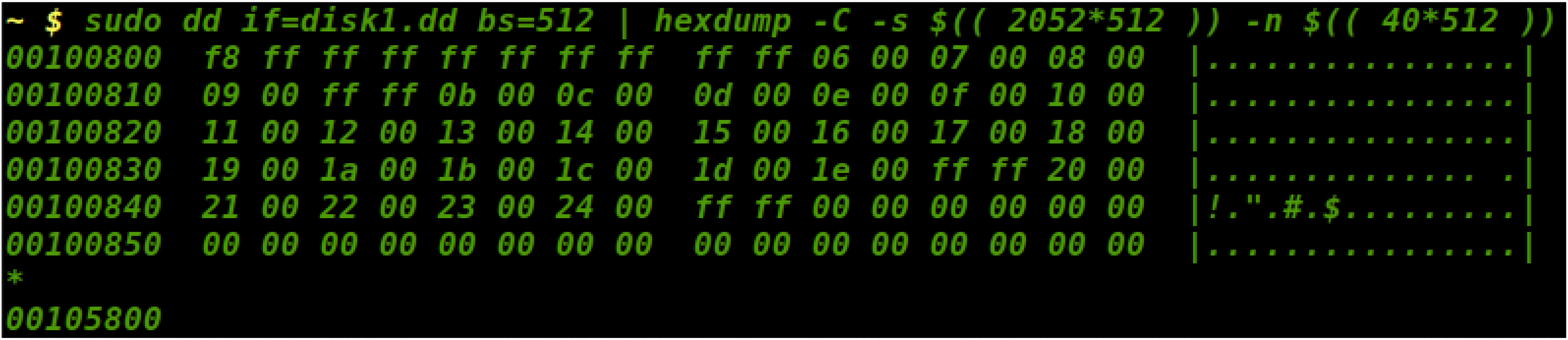
Root Directory

Data Area

FAT Contents

- What does a FAT look like?

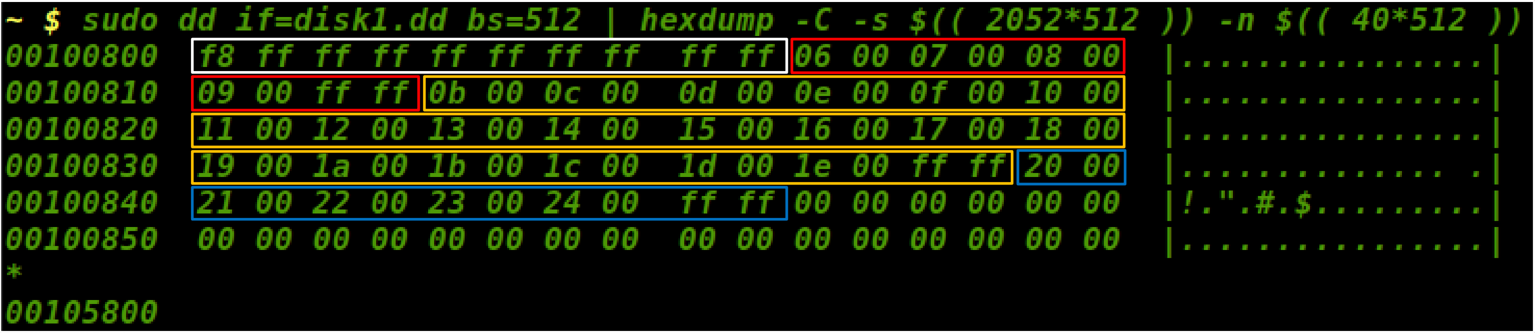




```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))
00100800  f8 ff ff ff ff ff ff ff  ff ff 06 00 07 00 08 00  |................|
00100810  09 00 ff ff 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 ff ff 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  ff ff 00 00 00 00 00 00  |!.".#.$.........|
00100850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00105800
```

FAT Values

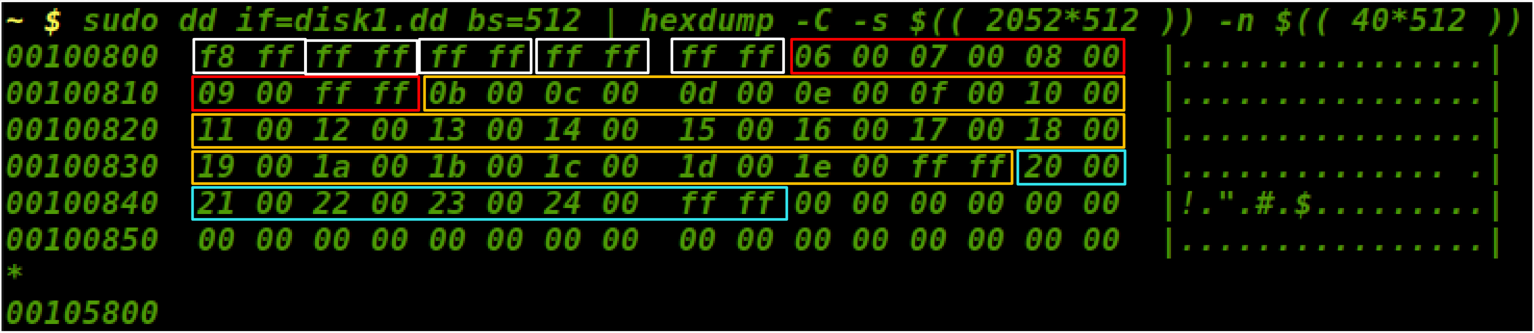
- By looking at the FAT we can determine where in the data area the contents of files are
- However, we also need the FAT16 root directory to provide other critical information including filenames, file sizes, or metadata
- Cluster values to keep in mind when evaluating a FAT
  - ✓ Free Cluster (0x0000)
  - ✓ Occupied Cluster (0x0002 – 0xFFEF)
  - ✓ Reserved Cluster (0xFFF6)
  - ✓ Bad Cluster (0xFFF7)
  - ✓ Last Cluster in a File (0xFFF8 – 0xFFFF)



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))
00100800  f8 ff ff ff ff ff ff ff  ff ff 06 00 07 00 08 00  |................|
00100810  09 00 ff ff 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 ff ff 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  ff ff 00 00 00 00 00 00  |!.".#.$.........|
00100850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00105800
```

FAT Table Header
Clusters for File #1
Clusters for File #2
Clusters for File #3

Reading a File Allocation Table



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))
00100800  f8 ff ff ff ff ff ff ff  ff ff 06 00 07 00 08 00  |................|
00100810  09 00 ff ff 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 ff ff 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  ff ff 00 00 00 00 00 00  |!.".#.$.........|
00100850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00105800
```

FAT Table Header
Clusters for File #1
Clusters for File #2
Clusters for File #3

***Note: There are 4 sectors per cluster in this example!**

The 1st file represented in the FAT resides from cluster 5 to cluster 9 in the data area

- ✓ The maximum size of the 1st file is 10,240 bytes (i.e. 5 * 4 * 512)

The 2nd file represented in the FAT resides from cluster 10 to cluster 30 in the data area

- ✓ The maximum size of the 2nd file is 43,008 bytes (i.e. 21 * 4 * 512)

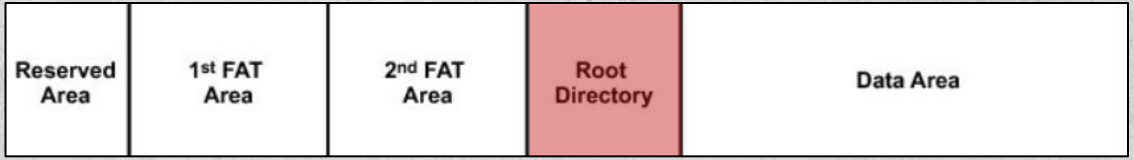
The 3rd file represented in the FAT resides from cluster 31 to cluster 36 in the data area

- ✓ The maximum size of the 3rd file is 12,288 bytes (i.e. 6 * 4 * 512)

FAT16 Root Directory

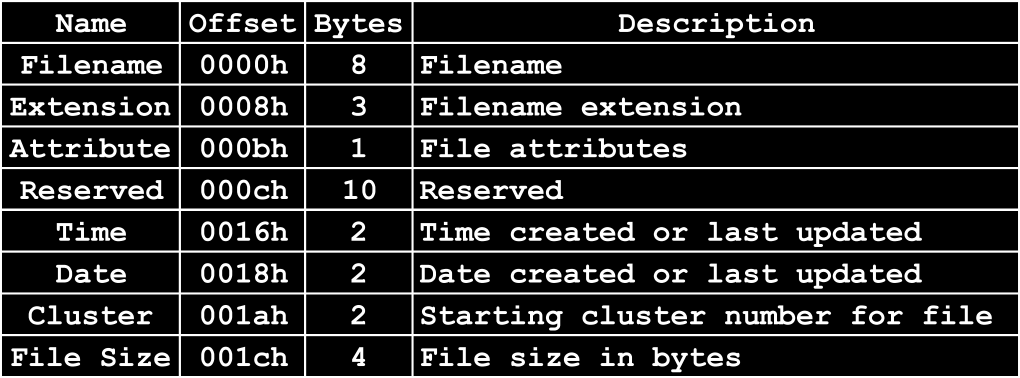
FAT16 Root Directory

- The root directory of a FAT16 partition contains critical data about files including filenames, extensions, attributes, and starting data clusters
- Each FAT16 root directory entry is 32 bytes and contains 512 entries



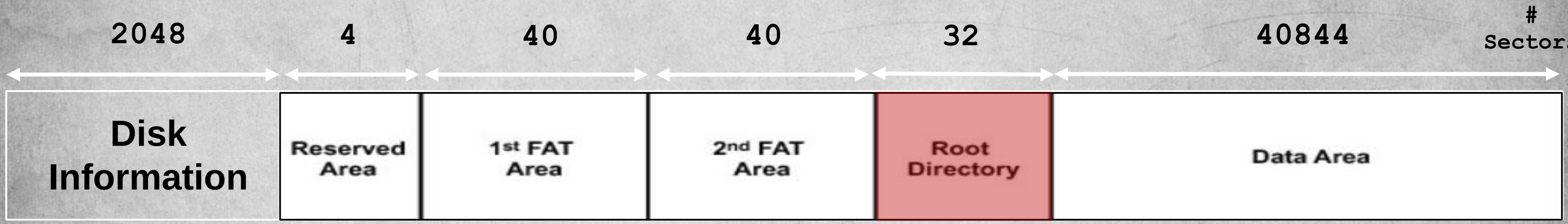
Reserved Area | 1st FAT Area | 2nd FAT Area | Root Directory | Data Area

- Each entry in the root directory will contain a 32-byte structure that includes:

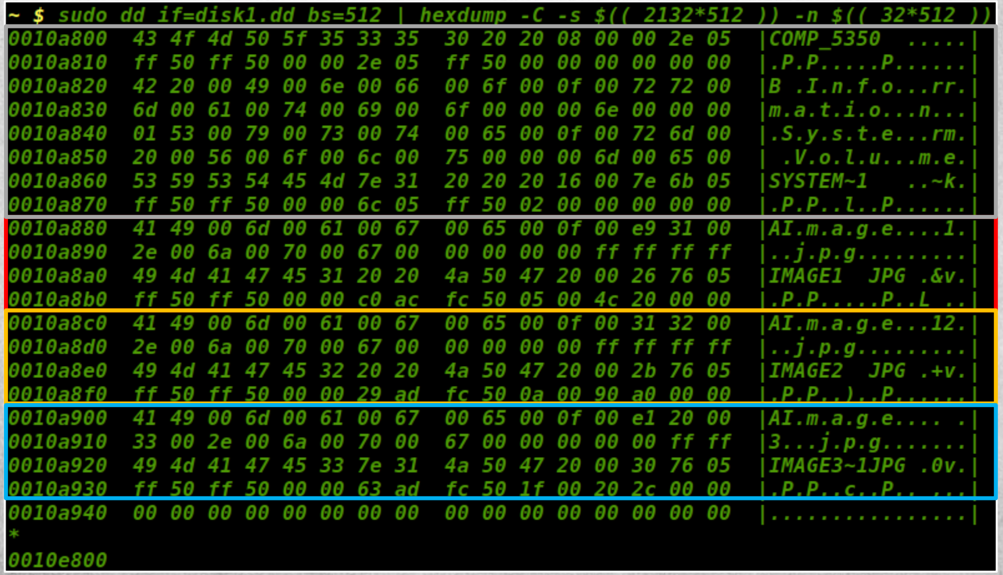


Name	Offset	Bytes	Description
Filename	0000h	8	Filename
Extension	0008h	3	Filename extension
Attribute	000bh	1	File attributes
Reserved	000ch	10	Reserved
Time	0016h	2	Time created or last updated
Date	0018h	2	Date created or last updated
Cluster	001ah	2	Starting cluster number for file
File Size	001ch	4	File size in bytes

Root Directory Contents



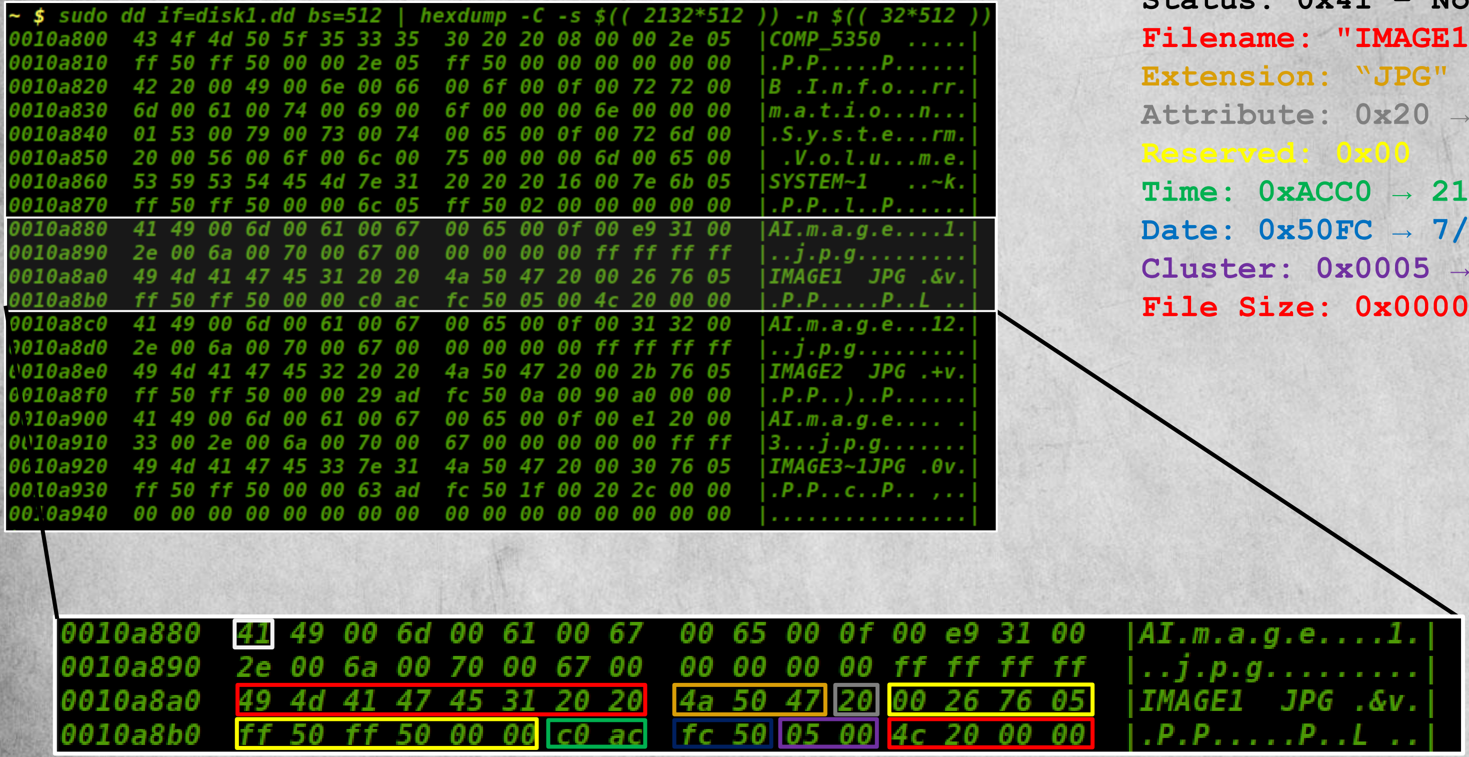
- This partition has only 3 files on the file system



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2132*512 )) -n $(( 32*512 ))
0010a800  43 4f 4d 50 5f 35 33 35  30 20 20 08 00 00 2e 05  |COMP_5350  .....|
0010a810  ff 50 ff 50 00 00 2e 05  ff 50 00 00 00 00 00 00  |.P.P.....P......|
0010a820  42 20 00 49 00 6e 00 66  00 6f 00 0f 00 72 72 00  |B .I.n.f.o...rr.|
0010a830  6d 00 61 00 74 00 69 00  6f 00 00 00 6e 00 00 00  |m.a.t.i.o...n...|
0010a840  01 53 00 79 00 73 00 74  00 65 00 0f 00 72 6d 00  |.S.y.s.t.e...rm.|
0010a850  20 00 56 00 6f 00 6c 00  75 00 00 00 6d 00 65 00  | .V.o.l.u...m.e.|
0010a860  53 59 53 54 45 4d 7e 31  20 20 20 16 00 7e 6b 05  |SYSTEM~1   ..~k.|
0010a870  ff 50 ff 50 00 00 6c 05  ff 50 02 00 00 00 00 00  |.P.P..l..P......|
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
0010a8c0  41 49 00 6d 00 61 00 67  00 65 00 0f 00 31 32 00  |AI.m.a.g.e...12.|
0010a8d0  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8e0  49 4d 41 47 45 32 20 20  4a 50 47 20 00 2b 76 05  |IMAGE2  JPG .+v.|
0010a8f0  ff 50 ff 50 00 00 29 ad  fc 50 0a 00 90 a0 00 00  |.P.P..)..P......|
0010a900  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e1 20 00  |AI.m.a.g.e.... .|
0010a910  33 00 2e 00 6a 00 70 00  67 00 00 00 00 00 ff ff  |3...j.p.g.......|
0010a920  49 4d 41 47 45 33 7e 31  4a 50 47 20 00 30 76 05  |IMAGE3~1JPG .0v.|
0010a930  ff 50 ff 50 00 00 63 ad  fc 50 1f 00 20 2c 00 00  |.P.P..c..P.. ,..|
0010a940  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
0010e800
```

System Volume Information
Root Directory Entry #1
Root Directory Entry #2
Root Directory Entry #3

Root Directory Entries



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2132*512 )) -n $(( 32*512 ))
0010a800  43 4f 4d 50 5f 35 33 35  30 20 20 08 00 00 2e 05  |COMP_5350  .....|
0010a810  ff 50 ff 50 00 00 2e 05  ff 50 00 00 00 00 00 00  |.P.P.....P......|
0010a820  42 20 00 49 00 6e 00 66  00 6f 00 0f 00 72 72 00  |B .I.n.f.o...rr.|
0010a830  6d 00 61 00 74 00 69 00  6f 00 00 00 6e 00 00 00  |m.a.t.i.o...n...|
0010a840  01 53 00 79 00 73 00 74  00 65 00 0f 00 72 6d 00  |.S.y.s.t.e...rm.|
0010a850  20 00 56 00 6f 00 6c 00  75 00 00 00 6d 00 65 00  | .V.o.l.u...m.e.|
0010a860  53 59 53 54 45 4d 7e 31  20 20 20 16 00 7e 6b 05  |SYSTEM~1   ..~k.|
0010a870  ff 50 ff 50 00 00 6c 05  ff 50 02 00 00 00 00 00  |.P.P..l..P......|
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
0010a8c0  41 49 00 6d 00 61 00 67  00 65 00 0f 00 31 32 00  |AI.m.a.g.e...12.|
0010a8d0  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8e0  49 4d 41 47 45 32 20 20  4a 50 47 20 00 2b 76 05  |IMAGE2  JPG .+v.|
0010a8f0  ff 50 ff 50 00 00 29 ad  fc 50 0a 00 90 a0 00 00  |.P.P..)..P......|
0010a900  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e1 20 00  |AI.m.a.g.e.... .|
0010a910  33 00 2e 00 6a 00 70 00  67 00 00 00 00 00 ff ff  |3...j.p.g.......|
0010a920  49 4d 41 47 45 33 7e 31  4a 50 47 20 00 30 76 05  |IMAGE3~1JPG .0v.|
0010a930  ff 50 ff 50 00 00 63 ad  fc 50 1f 00 20 2c 00 00  |.P.P..c..P.. ,..|
0010a940  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

Status: 0x41 – Normal File

Filename: "IMAGE1 "

Extension: “JPG"

Attribute: 0x20 → Archive

Reserved: 0x00

Time: 0xACC0 → 21:38:00

Date: 0x50FC → 7/28/2020

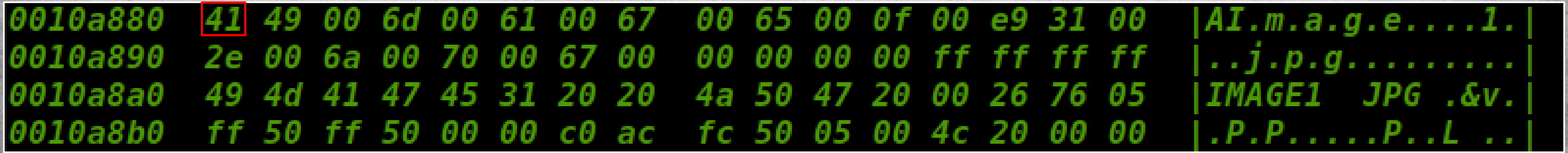
Cluster: 0x0005 → Cluster 5

File Size: 0x0000204C → 8268 bytes

Name	Offset	Bytes
Filename	0000h	8
Extension	0008h	3
Attribute	000bh	1
Reserved	000ch	10
Time	0016h	2
Date	0018h	2
Cluster	001ah	2
File Size	001ch	4

FAT16 File Status

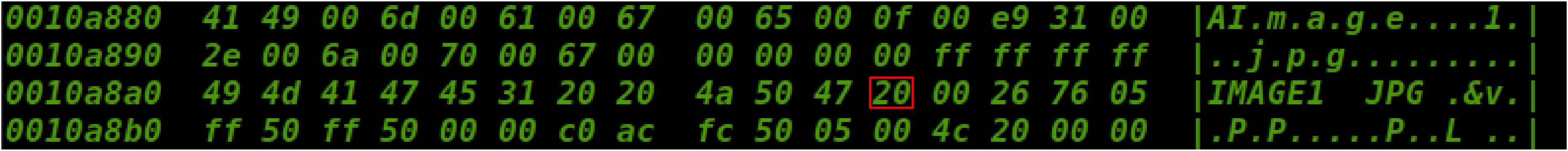
- FAT16 specifies file status:
  - ✓ 0x00 – Filename Never Used
  - ✓ 0xe5 – Filename Used, But Deleted
  - ✓ 0x2e – Directory
  - ✓ 0x41 – Normal File



```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

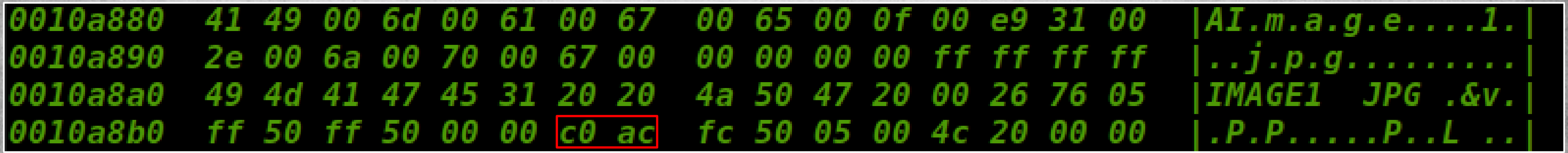
FAT16 File Attributes

- FAT16 specifies file attributes:
  - ✓ 0x01 – Read Only
  - ✓ 0x02 – Hidden
  - ✓ 0x04 – System
  - ✓ 0x08 – Volume Label
  - ✓ 0x10 – Directory
  - ✓ 0x20 – Archive
  - ✓ 0x0F – Long File Name

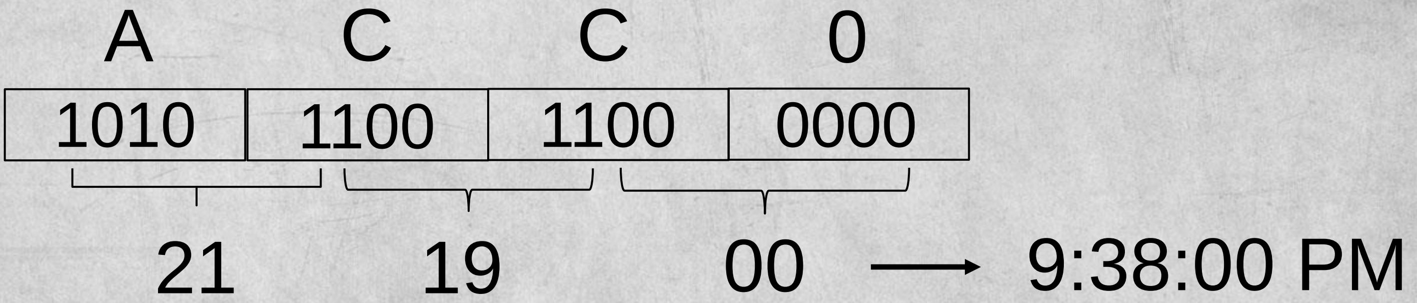


```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

FAT16 File Time



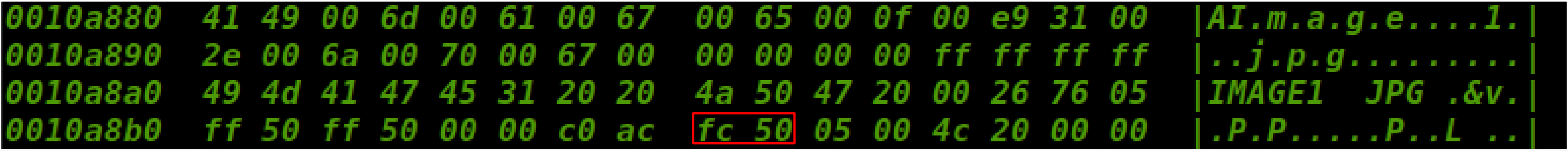
```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```



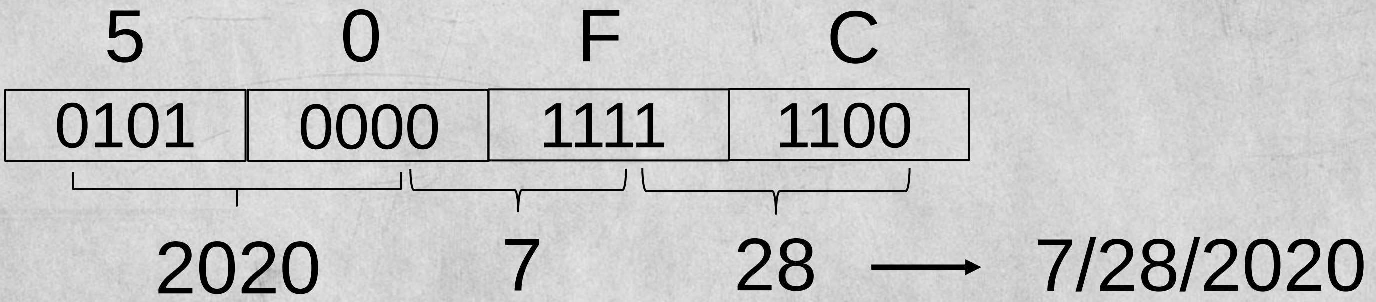
19*2 = 38

- File dates and times are not essential information for files. Why?

FAT16 File Date



```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```



5 0 F C

0101 0000 1111 1100

2020 7 28 → 7/28/2020

1980 + 40 = 2020

Directory Entries

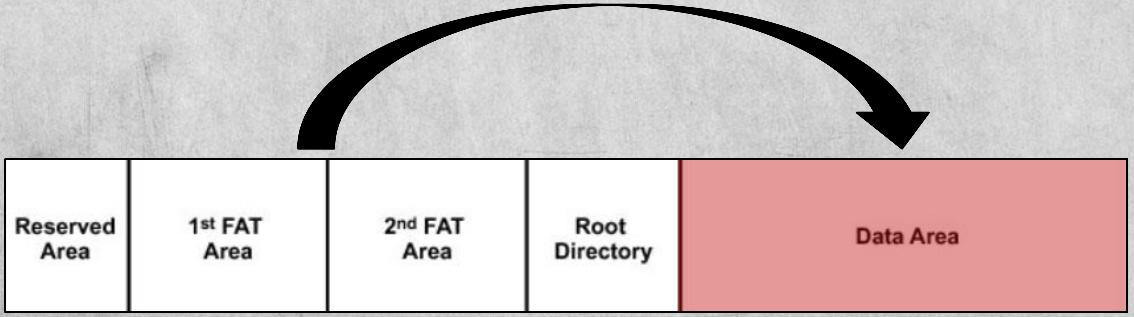
- Following the same process for each file in the previous example, we can generate a table for all root directory entries
- The root directory indicates that three files currently exist, and their contents can be found by looking into the FAT16 data area

Filename	Extension	Attribute	Time	Date	File Start (Cluster)	File Start (Sectors)	File Size (Bytes)	File Size (Sectors)	Status
Image1	jpg	Archive	9:38:00	7/28/2020	0x0005	20	8268	17	Normal File
Image2	jpg	Archive	9:41:18	7/28/2020	0x000A	84	41104	81	Normal File
Image3	jpg	Archive	9:43:06	7/28/2020	0x001F	24	11296	23	Normal File

FAT16 Data Area

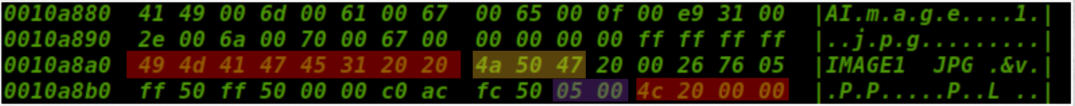
FAT16 Data Area

- The data area of a FAT16 partition is where file contents are stored
- As we have seen from the FAT16 partition to find files we will need information found in each of the following data structures
  - File Allocation Table
  - Root Directory
  - Data Area



Finding Data Area File Contents

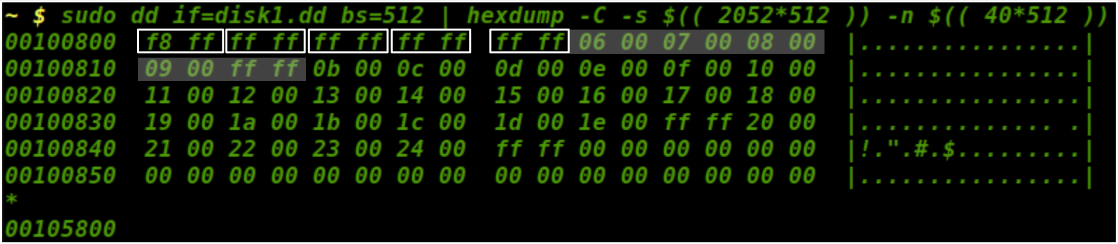
- The root directory specifies starting cluster and file size



```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

Filename	Start Cluster	File Size
Image1.jpg	0005	8268 bytes

- The FAT points to follow-on clusters or to EOF (i.e. 0xFFFF)



```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))
00100800  f8 ff ff ff ff ff ff ff  ff ff 06 00 07 00 08 00  |................|
00100810  09 00 ff ff 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 ff ff 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  ff ff 00 00 00 00 00 00  |!.".#.$.........|
00100850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00105800
```

File System Values

512 bytes / sector

1 cluster = 4 sectors

2048 bytes / cluster

List Directory Entries

- Now we can combine the FAT and directory entries to find the location of file contents

Note: The FAT is going to specify the starting location of file contents in the data area! In this case, there is an offset of 12 sectors before file contents are found.

File Allocation Table

Data Area Cluster	Next Cluster
0x0000	0xFFF8
0x0001	0xFFFF
0x0002	0xFFFF
0x0003	0xFFFF
0x0004	0xFFFF
0x0005	0x0006
0x0006	0x0007
0x0007	0x0008
0x0008	0x0009
0x0009	0xFFFF

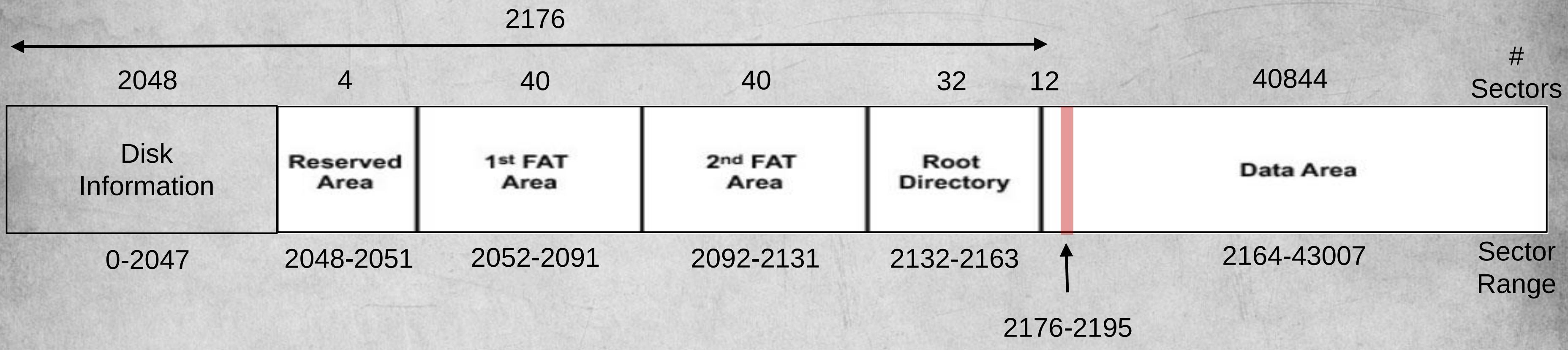
3 Clusters = 12 Sectors

The file named Image1.jpg is 20 sectors long

Root Directory Entries (Partial)

Filename	Cluster	File Size (Clusters)	File Size (Sectors)	File Size
Image1	0x0005	5	20	8268 bytes
Image2	0x000A	21	84	41104 bytes
Image3	0x001F	6	24	11296 bytes

File Recovery *



2176

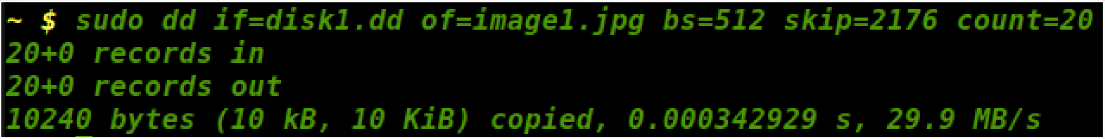
# Sectors	2048	4	40	40	32	12	40844
	Disk Information	Reserved Area	1st FAT Area	2nd FAT Area	Root Directory		Data Area
Sector Range	0-2047	2048-2051	2052-2091	2092-2131	2132-2163		2164-43007

2176-2195

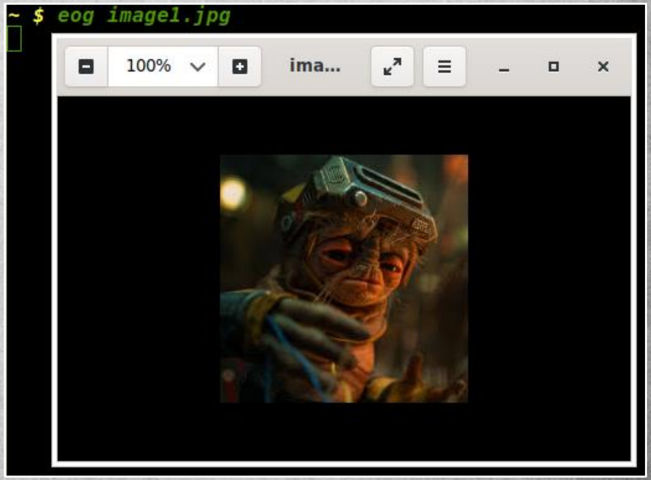
sudo dd if=disk1.dd of=image1.jpg bs=512 skip=2176 count=20

skip=2176: # Sectors to Skip

count=20: Max File Size



```
~ $ sudo dd if=disk1.dd of=image1.jpg bs=512 skip=2176 count=20
20+0 records in
20+0 records out
10240 bytes (10 kB, 10 KiB) copied, 0.000342929 s, 29.9 MB/s
```

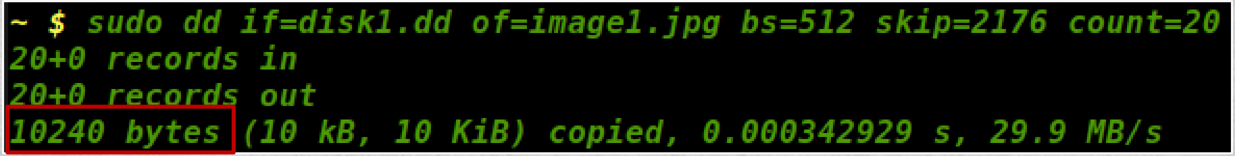


```
~ $ eog image1.jpg
```

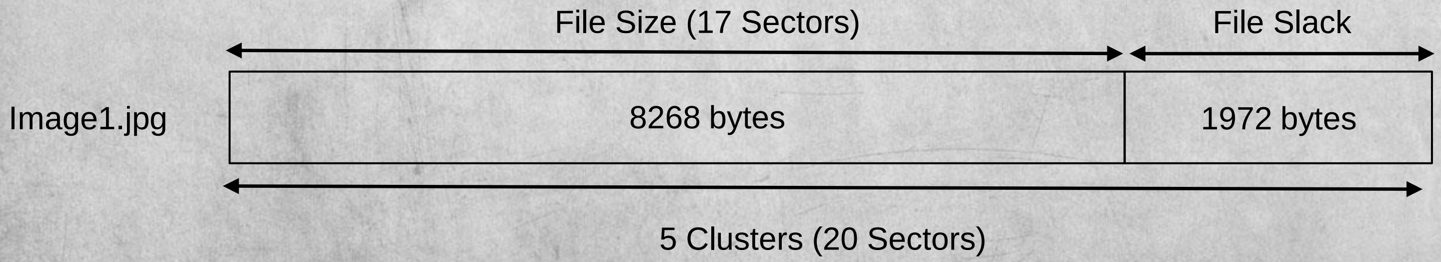
File Slack

- Why is there a discrepancy between the specified size of the file in the root directory and the file recovered?

Filename	Extension	Attribute	Time	Date	File Start (Cluster)	File Start (Sectors)	File Size (Bytes)	File Size (Sectors)	Status
Image1	jpg	Archive	9:38:00	7/28/2020	0x0005	20	8268	17	Normal File
Image2	jpg	Archive	9:41:18	7/28/2020	0x000A	84	41104	81	Normal File
Image3	jpg	Archive	9:43:06	7/28/2020	0x001F	24	11296	23	Normal File



```
~ $ sudo dd if=disk1.dd of=image1.jpg bs=512 skip=2176 count=20
20+0 records in
20+0 records out
10240 bytes (10 kB, 10 KiB) copied, 0.000342929 s, 29.9 MB/s
```



File Size (17 Sectors)

File Slack

Image1.jpg

8268 bytes

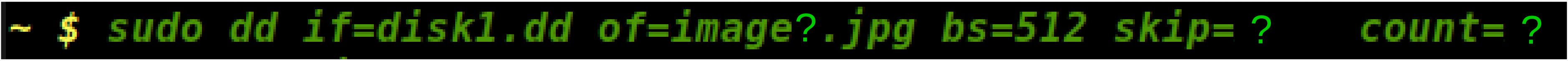
1972 bytes

5 Clusters (20 Sectors)

Assignment

- Using the same process discussed in this session, recover the other 2 files in the FAT16 file system
- The next quiz will ask questions about which skip and count values were necessary to recover image2.jpg and image3.jpg
- You may work with your project teams on Quiz #2

Filename	Extension	Attribute	Time	Date	File Start (Cluster)	File Start (Sectors)	File Size (Bytes)	File Size (Sectors)	Status
Image1	jpg	Archive	9:38:00	7/28/2020	0x0005	20	8268	17	Normal File
Image2	jpg	Archive	9:41:18	7/28/2020	0x000A	84	41104	81	Normal File
Image3	jpg	Archive	9:43:06	7/28/2020	0x001F	24	11296	23	Normal File



```
~ $ sudo dd if=disk1.dd of=image?.jpg bs=512 skip= ?    count= ?
```

References

- **File System Forensic Analysis, Carrier, 2005**

- **Endian**
  - ✓ **https://bit.ly/30lIv3y**

- **FAT Volume Serial Numbers**
  - ✓ **https://www.digital-detective.net/documents/Volume%20Serial%20Numbers.pdf**

- **File Allocation Tables**
  - ✓ **https://www.ntfs.com/fat-partition-sector.htm**
  - ✓ **http://pierre.baudu.in/other/clustersize.en.html**
  - ✓ **https://slideplayer.com/slide/9087283**
  - ✓ **https://www.slideshare.net/shashikantpabari/file-system-and-file-allocation-tablesystem**
  - ✓ **https://doc.micrium.com/display/TECHOV/FAT+Organization**

- **Digital Forensics with Open Source Tools, Chapter 3**