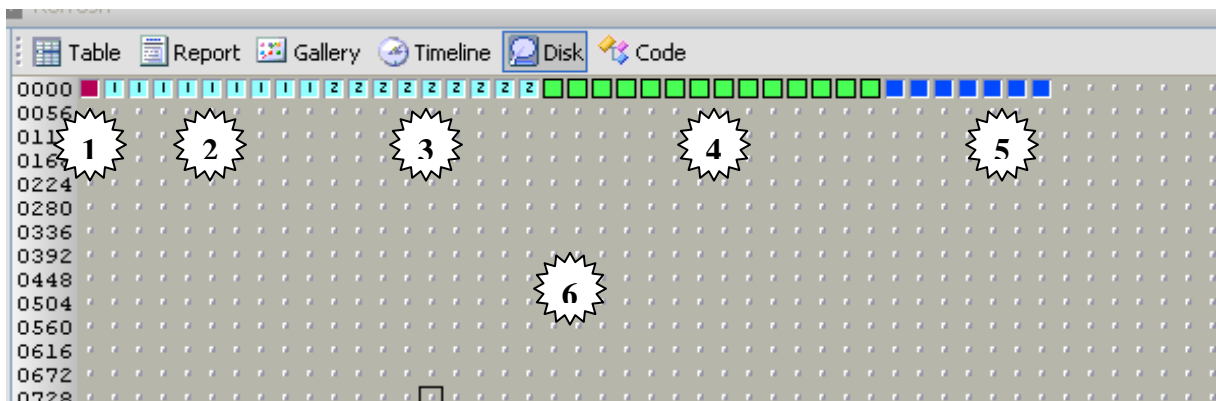


Introduction to Computer Forensics and Security

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

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,

```

000EE 3C 90 4D 53 44 4F 53 35 2E 30 00 02 01 01 00 02 E0 00 40 0B F0 k<...MSDOS5.0.....?P
02209 00 12 00 02 00 00 00 00 00 00 00 00 00 00 00 00 2D DD 16 15 24 4E .....?#N
0444F 20 4E 41 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 FA 33 C0 8E 0 NAME FAT12 z30?
066D0 BC 00 7C 16 07 BB 78 06 C5 37 1E 56 16 53 BF 3F 8E B9 0B 00 P<|...:6E7-V-S>-!9...
08FC F3 A4 06 1F C6 45 FE 0F BB 0E 18 7C 88 4D F9 89 47 02 C7 07 3C |$<.FE.....My G-C>..
1107C FB CD 13 72 79 33 C0 39 06 13 7C 74 08 BB 0E 13 7C 89 0E 20 7C |(M-y3G9...|t....|
132A0 10 7C F7 26 16 7C 03 06 1C 7C 13 16 1E 7C 03 06 0E 7C 83 D2 00 ..wa|.....|...|...R
154A3 50 7C 89 16 52 7C A3 49 7C 89 16 4B 7C BB 20 00 F7 26 11 7C 8B FP|R|#I|K|S-wa|i
1761E 0B 7C 03 C3 48 F7 F3 01 06 49 7C 83 16 4B 7C 00 BB 05 05 8B 16 ..|CHes-I|...K|j...
19852 7C A1 50 7C 88 92 00 72 1D B0 01 E8 AC 00 72 16 8B FB B9 0B 00 R|P|h-x-0-h-r...{9...
220BE E6 7D F3 A6 75 0A 8D 7F 20 B9 0B 00 F3 A6 74 18 BE 9E 7D E5 F5 >f)=6u □ 9...s6t...}h
24200 K3 C0 CD 16 5E 1F 0F 84 0F 44 02 CD 19 58 58 58 58 8E 8B 47 1A .3M.....B-M-XCCTh-G
26448 48 8A 1E OD 7C 32 FF F7 E3 03 06 49 7C 13 16 4B 7C BB 00 07 B9 HH |2|wc-I|...K|j...
28603 00 50 52 51 E8 3A 00 72 D8 B0 01 E8 54 00 5A 5A 58 72 BB 05 01 ..PRQh:rxO-hT-YZxr...
30800 83 D2 00 03 1E 0B 7C E2 8A 2E 15 70 8A 16 24 7C 8B 1E 49 7C ...>...lbb...|...f|...I
330A1 4B 7C EA 00 70 00 AC 0A C0 74 29 BA 0E BB 07 00 CD 10 EB F2 |K|j...p, @t|4...-m-kx
3523B 16 18 7C 73 19 F7 36 18 7C FE C2 8B 16 47 7C 33 D2 F7 36 1A 7C |s|=w6|-B-M-O1GBh-G
37488 16 25 7C A3 4D 7C FC C3 F9 C3 B4 02 8B 16 4D 7C B1 06 D2 E6 0A ..*|H|MxCyC4...M|L-Rf
39636 4D 7C 8B CA 86 E9 8A 16 24 7C 8A 36 25 7C CD 13 C3 OD 0A 4E EF 60|J-i *|6|M-C No
4186E 2D 53 79 73 74 65 6D 20 64 69 73 6B 20 6F 72 20 64 69 73 6B 20 n-system disk or disk
44065 72 72 6F 72 OD 0A 52 65 70 6C 61 63 65 20 61 6E 64 20 70 72 65 error Replace and pre
46273 73 20 61 6E 79 20 6B 65 79 20 77 68 65 6E 20 72 65 61 64 79 OD s any key when ready
4840A 00 49 4F 20 20 20 20 53 59 53 4D 53 44 4F 53 20 20 20 53 .TO SYMSDOS S
50659 53 00 00 55 5A YS..U

```

Bytes 11-12

Bytes per sector. Allowed values include 512, 1024, 2048, and 4096
 00 02 = littleEndian (02 00) = 512

15

Task 2.2: Looking at the FAT boot sector

Method Looking at the byte 13:

Byte 13

Sectors per cluster (data unit). Allowed values are powers of 2
01 = 1

Task 2.3: Looking at the FAT boot sector

Method Looking at the byte 16:

Byte 16
Number of FATs. Typically two for redundancy.
02 = 2

23

Task 2.4: Looking at the FAT boot sector

Method Looking at the byte 17-18:

Bytes 17 - 18
Maximum number of files in the root directory for FAT12 and FAT16. This is 0 for FAT32 and typically 512 for FAT16.
E0 00= littleEndian (00 E0) =224

24

Task 2.5: Looking at the FAT boot sector

Method Looking at the byte 19-20:

```
000EB 3C 90 4D 53 44 4F 53 35 2E 30 00 02 01 01 00 02 E0 00 40 0B F0 k<MSDOS5.0.....Gp
02209 00 12 00 02 00 00 00 00 00 00 00 00 00 00 00 29 DD 16 15 24 4E .....}#N
0444F 20 4E 41 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 FA 33 C0 8E 0 NAME FAT12 z30
066D0 BC 00 7C 16 07 BB 78 00 36 C5 37 1E 56 16 53 BF 3E 7C B9 0B 00 P<|...x*6E7*V*S>|9..
088FC F3 A4 06 1F C6 45 FE 0F 8B 0E 18 7C 88 4D F9 89 47 02 C7 07 3E |$*FE.....|My G.C>
1107C FB CD 13 72 79 33 C0 39 06 13 7C 74 08 8B 0E 13 7C 89 0E 20 7C |(M*ry309...|t....|..|
132A0 10 7C F7 26 16 7C 03 06 1C 7C 13 16 1E 7C 03 06 0E 7C 83 D2 00 *|w6*|...|...|...|R
154A3 50 7C 89 16 52 7C A3 49 7C 89 16 4B 7C B8 20 00 F7 26 11 7C 8B #P|*R|*I|*K|8*w6*|
1761E 0B 7C 03 C3 48 F7 F3 01 06 49 7C 83 16 4B 7C 00 B8 00 05 8B 16 ...|CHws*I|...K|;....
19852 7C A1 50 7C E8 92 00 72 1D B0 01 E8 AC 00 72 16 8B FB B9 0B 00 R|P|h*x*0*h,x*(9..
220BE E6 7D F3 A6 75 0A 8D 7F 20 B9 0B 00 F3 A6 74 18 BE 9E 7D E8 5F >f)s4u 0 9...s4t>.)h_
24200 33 C0 CD 16 5E 1F 8F 04 8F 44 02 CD 19 58 58 58 EB E8 8B 47 1A *30M^.....D*M*000h.G
26448 48 8A 1E 0D 7C 32 FF F7 E3 03 06 49 7C 13 16 4B 7C BB 00 07 B9 HH * |2|w6*I|...K|;..9
28603 00 50 52 51 E8 3A 00 72 D8 B0 01 E8 54 00 59 5A 58 72 BB 05 01 ..PRQh:xX0*hT*YZXr;..
30800 83 D2 00 03 1E 0B 7C E2 E2 8A 2E 15 7C 8A 16 24 7C 8B 1E 49 7C ..R....|bb..|..|...I|
330A1 4B 7C EA 00 00 70 00 AC 0A C0 74 29 B4 0E BB 07 00 CD 10 EB F2 |K|j*p*,(t)4;..M*kr
3523B 16 18 7C 73 19 F7 36 18 7C FE C2 88 16 4F 7C 33 D2 F7 36 1A 7C ;..|s*w6*|B*0|3Rw6*|
37488 16 25 7C A3 4D 7C F8 C3 F9 C3 B4 02 8B 16 4D 7C E1 06 D2 E6 0A **|#M|xCyC4...M|l.Rf
39636 4F 7C 8B CA 86 E9 8A 16 24 7C 8A 36 25 7C CD 13 C3 0D 0A 4E 6F 60|J*i*|6|M*C No
4186E 2D 53 79 73 74 65 6D 20 64 69 73 6B 20 6F 72 20 64 69 73 6B 20 n-System disk or disk
44065 72 72 6F 72 0D 0A 52 65 70 6C 61 63 65 20 61 6E 64 20 70 72 65 error Replace and pre
46273 73 20 61 6E 79 20 6B 65 79 20 77 68 65 5E 20 72 65 61 64 79 0D ss any key when ready
4840A 00 49 4F 20 20 20 20 20 20 20 53 59 53 4B 53 44 4F 53 20 20 20 53 .IO SYSMSDOS S
50659 53 00 00 55 BA TS..U
```

Bytes 19 - 20
number of sectors in file system
40 0B = littleEndian (0B 40) = 2880

25

Task 2.6: Looking at the FAT boot sector

Method Looking at the byte 22-23:

```
000EB 3C 90 4D 53 44 4F 53 35 2E 30 00 02 01 01 00 02 E0 00 40 0B F0 k<MSDOS5.0.....Gp
02209 00 12 00 02 00 00 00 00 00 00 00 00 00 00 00 29 DD 16 15 24 4E .....}#N
0444F 20 4E 41 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 FA 33 C0 8E 0 NAME FAT12 z30
066D0 BC 00 7C 16 07 BB 78 00 36 C5 37 1E 56 16 53 BF 3E 7C B9 0B 00 P<|...x*6E7*V*S>|9..
088FC F3 A4 06 1F C6 45 FE 0F 8B 0E 18 7C 88 4D F9 89 47 02 C7 07 3E |$*FE.....|My G.C>
1107C FB CD 13 72 79 33 C0 39 06 13 7C 74 08 8B 0E 13 7C 89 0E 20 7C |(M*ry309...|c....|..|
132A0 10 7C F7 26 16 7C 03 06 1C 7C 13 16 1E 7C 03 06 0E 7C 83 D2 00 *|w6*|...|...|...|R
154A3 50 7C 89 16 52 7C A3 49 7C 89 16 4B 7C B8 20 00 F7 26 11 7C 8B #P|*R|*I|*K|8*w6*|
1761E 0B 7C 03 C3 48 F7 F3 01 06 49 7C 83 16 4B 7C 00 B8 00 05 8B 16 ...|CHws*I|...K|;....
19852 7C A1 50 7C E8 92 00 72 1D B0 01 E8 AC 00 72 16 8B FB B9 0B 00 R|P|h*x*0*h,x*(9..
220BE E6 7D F3 A6 75 0A 8D 7F 20 B9 0B 00 F3 A6 74 18 BE 9E 7D E8 5F >f)s4u 0 9...s4t>.)h_
24200 33 C0 CD 16 5E 1F 8F 04 8F 44 02 CD 19 58 58 58 EB E8 8B 47 1A *30M^.....D*M*000h.G
26448 48 8A 1E 0D 7C 32 FF F7 E3 03 06 49 7C 13 16 4B 7C BB 00 07 B9 HH * |2|w6*I|...K|;..9
28603 00 50 52 51 E8 3A 00 72 D8 B0 01 E8 54 00 59 5A 58 72 BB 05 01 ..PRQh:xX0*hT*YZXr;..
30800 83 D2 00 03 1E 0B 7C E2 E2 8A 2E 15 7C 8A 16 24 7C 8B 1E 49 7C ..R....|bb..|..|...I|
330A1 4B 7C EA 00 00 70 00 AC 0A C0 74 29 B4 0E BB 07 00 CD 10 EB F2 |K|j*p*,(t)4;..M*kr
3523B 16 18 7C 73 19 F7 36 18 7C FE C2 88 16 4F 7C 33 D2 F7 36 1A 7C ;..|s*w6*|B*0|3Rw6*|
37488 16 25 7C A3 4D 7C F8 C3 F9 C3 B4 02 8B 16 4D 7C E1 06 D2 E6 0A **|#M|xCyC4...M|l.Rf
39636 4F 7C 8B CA 86 E9 8A 16 24 7C 8A 36 25 7C CD 13 C3 0D 0A 4E 6F 60|J*i*|6|M*C No
4186E 2D 53 79 73 74 65 6D 20 64 69 73 6B 20 6F 72 20 64 69 73 6B 20 n-System disk or disk
44065 72 72 6F 72 0D 0A 52 65 70 6C 61 63 65 20 61 6E 64 20 70 72 65 error Replace and pre
46273 73 20 61 6E 79 20 6B 65 79 20 77 68 65 5E 20 72 65 61 64 79 0D ss any key when ready
4840A 00 49 4F 20 20 20 20 20 20 20 53 59 53 4B 53 44 4F 53 20 20 20 53 .IO SYSMSDOS S
50659 53 00 00 55 BA TS..U
```

Byte 22-23
16-bit size in sectors of each FAT for FAT12 and FAT16. For FAT32,
this field is 0.
09 00 = littleEndian (00 09) = 9

27

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,

```
00000000: f0ff ffff ffff 0560 0007 8000 ff0f 0000 .....
00000010: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000060: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000080: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000090: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000e0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000f0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000100: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000110: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000120: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000130: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000140: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000150: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000160: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000170: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000180: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000190: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001a0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001b0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001c0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001d0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001f0: 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,

```
00000000: 4174 0065 0073 0074 0066 000f 008c 6f00 At.e.s.t.f....o.
00000010: 6c00 6400 6500 7200 0000 0000 ffff ffff l.d.e.r.....
00000020: 5445 5354 464f 7e31 2020 2010 003c 2396 TESTF0~1 ..<#.
00000030: 393b 393b 0000 2496 393b 0200 0000 0000 9;9;..$.9;.....
00000040: 4174 0065 0073 0074 0066 000f 002c 6f00 At.e.s.t.f...,o.
00000050: 6c00 6400 6500 7200 3100 0000 0000 ffff l.d.e.r.1.....
00000060: 5445 5354 464f 7e32 2020 2010 0040 2b96 TESTF0~2 ..@+.
00000070: 393b 393b 0000 2c96 393b 0300 0000 0000 9;9;...9;.....
00000080: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000090: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000e0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000f0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001000: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001100: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001200: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001300: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001400: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001500: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001600: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001700: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001800: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001900: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001a00: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001b00: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001c00: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001d00: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001e00: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001f00: 0000 0000 0000 0000 0000 0000 0000 0000 .....
```

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