East West University Bangladesh
Computing Science and Engineering Department

CSE-325: Operating System

Disk and File System Basics

Objectives:
The goal of this lab is to know the windows File Allocation Table (FAT). In this lab, you will learn:

- Hexadecimal Number System.
- FAT file system and boot sector details.
  - (Part a) tracing information from the given boot sector data (512 bytes)
    - **covered previous assignment**
  - (Part b) FAT, root directory and data area

(Part b) FAT, root directory and data area

Activity Background

Root Directory:

The root directory contains an entry for each file whose name appears at the root of the file system. Other directories can appear within the root directory; they are called subdirectories.

- Space in root directory is static and allocated when the disk is formatted.
- Space in subdirectories can be as large or small as desired.

Each entry in root directory is 32 bytes, so a single block (512 bytes) can contain 16 of them. An example (6 entries on the same MSDOS floppy):

An example (6 entries on the same MSDOS floppy):

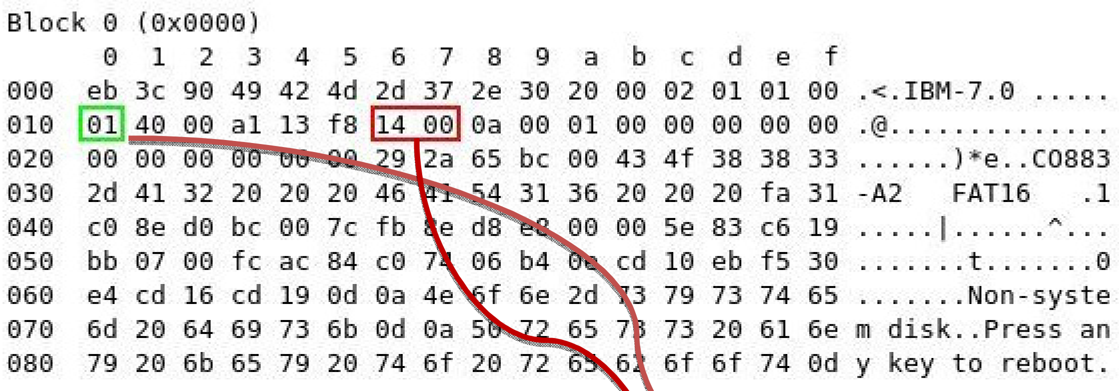
```
0009728 49 4f 20 20 20 20 20 20 53 59 53 27 00 00 00 00  IO      .SYS
0009744 00 00 00 00 00 00 08 5d 62 1b 1d 00 16 9f 00 00
0009760 4d 53 44 4f 53 20 20 20 53 59 53 27 00 00 00 00  MSDOS   .SYS
0009776 00 00 00 00 00 00 08 5d 62 1b 6d 00 38 95 00 00
0009792 43 4f 4d 4d 41 4e 44 20 43 4f 4d 20 00 00 00 00  COMMAND .COM
0009808 00 00 00 00 00 00 07 5d 62 1b b8 00 39 dd 00 00
0009824 44 42 4c 53 50 41 43 45 42 49 4e 27 00 00 00 00  DBLSPACE.BIN
0009840 00 00 00 00 00 00 08 5d 62 1b 27 01 f6 fc 00 00
0009856 4d 53 44 4f 53 20 20 20 20 20 20 28 00 00 00 00  MSDOS
0009872 00 00 00 00 00 00 1a 88 99 1c 00 00 00 00 00 00
0009888 46 44 49 53 4b 20 20 20 45 58 45 20 00 00 00 00  FDISK   .EXE
0009904 00 00 00 00 00 00 36 59 62 1b 02 00 17 73 00 00
```

The following table shows a summary of a single directory entry:

Bytes	Description
(0-7) 8 bytes	Filename
(8-10) 3 bytes	Filename extension
(11) 1 bytes	File attributes bit 0: read only bit 1: hidden bit 2: system file bit 3: volume label (only in root directory) bit 4: subdirectory bit 5: archive flag. Can be set and clear by programmer or user, always set when the file modified. It is used by backup program. bits 6-7: unused
(12-21) 10 bytes	Reserved
(22-23) 2 bytes	Time (5/6/5 bits, for hour/minutes/double seconds)
(24-25) 2 bytes	Date (7/4/5 bits, for year-since 1980/month/day)
(26-27) 2 bytes	Starting cluster # for the file
(28-31) 4 bytes	File size in bytes

Find the Root directory:

To find the root directory, we need to examine the file system data in the boot sector (block 0). So, let's look again at the following boot sector example:



```
Block 0 (0x0000)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
000  eb 3c 90 49 42 4d 2d 37 2e 30 20 00 02 01 01 00 .<.IBM-7.0 .....
010  01 40 00 a1 13 f8 14 00 0a 00 01 00 00 00 00 00 .@..............
020  00 00 00 00 00 00 29 2a 65 bc 00 43 4f 38 38 33 ......)*e..CO883
030  2d 41 32 20 20 20 46 41 54 31 36 20 20 20 fa 31 -A2   FAT16   .1
040  c0 8e d0 bc 00 7c fb 8e d8 e8 00 00 5e 83 c6 19 .....|......^...
050  bb 07 00 fc ac 84 c0 74 06 b4 0e cd 10 eb f5 30 .......t.......0
060  e4 cd 16 cd 19 0d 0a 4e 6f 6e 2d 73 79 73 74 65 .......Non-syste
070  6d 20 64 69 73 6b 0d 0a 50 72 65 73 73 20 61 6e m disk..Press an
080  79 20 6b 65 79 20 74 6f 20 72 65 62 6f 6f 74 0d y key to reboot.
```

We know that the root directory appears immediately after the last copy of the FAT. So what we need to find out is the size of the FAT, and how many copies there are. Thus, the number of blocks that appear before the root directory is given by:

(size of FAT)*(number of FATs) + 1

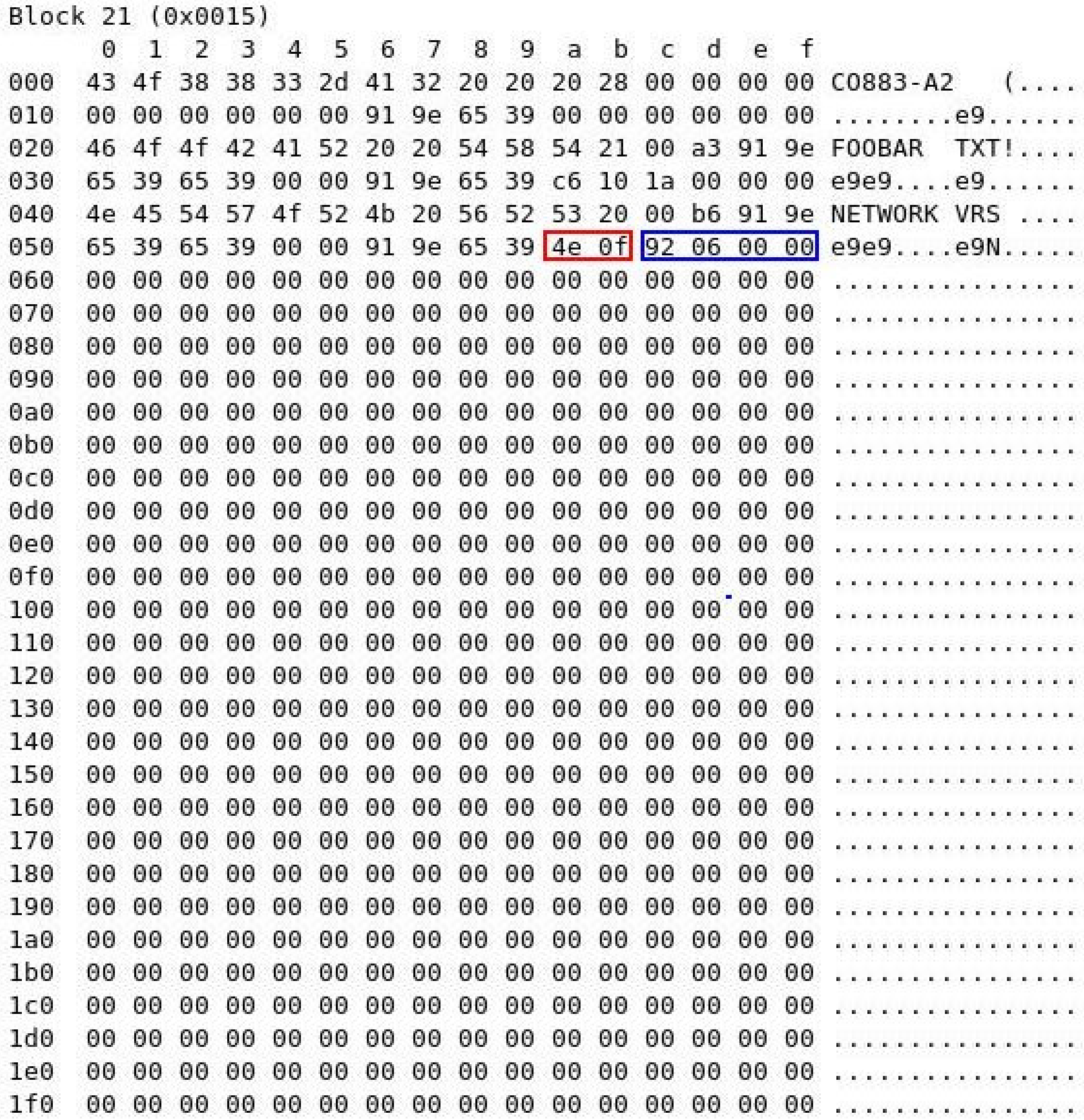
= 0x0014*1 + 1 => 0x0015 (decimal 21)

We should thus find the root directory in block 0x0015 (block 21)

ACTIVITY

From the below Root Directory (block 21) entries trace the following information for **FOOBAR.TXT** and **NETWORK.VRS** file entries:

1. Filename
2. File extension
3. File attributes value
4. Time
5. Date
6. Starting cluster
7. File size in bytes



```
Block 21 (0x0015)
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
000  43 4f 38 38 33 2d 41 32 20 20 20 28 00 00 00 00 CO883-A2   (....
010  00 00 00 00 00 00 91 9e 65 39 00 00 00 00 00 00 ........e9......
020  46 4f 4f 42 41 52 20 20 54 58 54 21 00 a3 91 9e FOOBAR  TXT!....
030  65 39 65 39 00 00 91 9e 65 39 c6 10 1a 00 00 00 e9e9....e9......
040  4e 45 54 57 4f 52 4b 20 56 52 53 20 00 b6 91 9e NETWORK VRS ....
050  65 39 65 39 00 00 91 9e 65 39 4e 0f 92 06 00 00 e9e9....e9N.....
060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0a0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0c0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0d0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0e0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0f0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1a0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1c0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1d0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1e0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1f0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

Data Blocks for a file:

Now we have known the root directory entries for NETWORK.VRS file. However we would also like to see the contents of the particular file.

Lets review what we know so far ..

- **The root directory starts at block 0x15 (block 21)**
- **The starting cluster of the file is 0x0f4e**
- **The first allocation unit starts at the first block after the root directory.**

But we don’t know:

- **Where the root directory ends.**

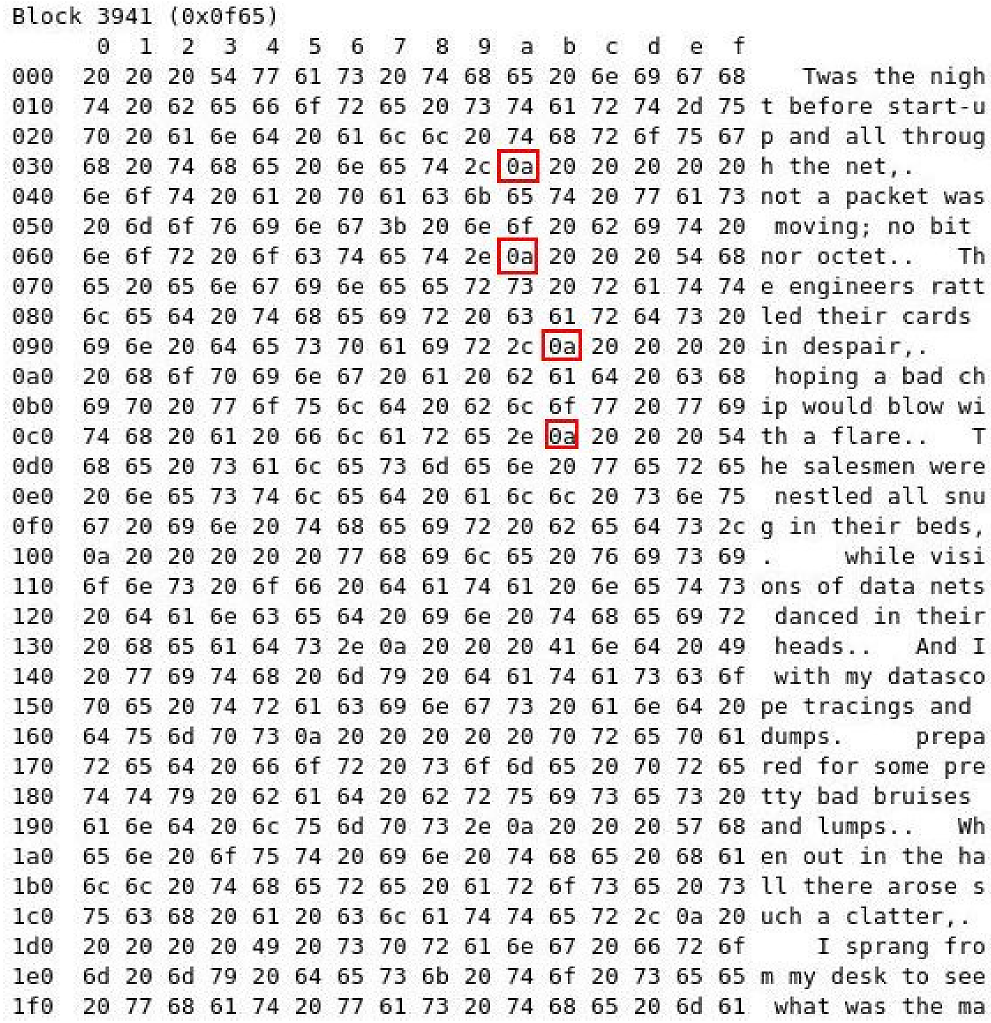
Thus, we need to know the **total size of root directory** and to trace that we need to find **the total number of root directories**.

The number of total root directories is in boot sector, lets that is 0x0040 (64 in decimal); we have known that each root entry consists 32 bytes. Thus total bytes hold by root directory is 64x32=2048 bytes =2048/512 blocks =4 blocks**.**

Root directory starts at block 0x15. Thus the first allocation unit starts at ox15 +4 or 0x19.

And to convert a cluster number (which is what appears in the root directory) to a block number, we need to add 0x17, to allow for that strange offset of 2.

We now know that the first data block of the file is at cluster number 0xf4e (see above). Adding the constant we have discovered, we find that this is block number 0xf4e+0x17, or 0xf65. Let's look at block 0xf65:



```
Block 3941 (0x0f65)
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
000  20 20 20 54 77 61 73 20 74 68 65 20 6e 69 67 68    Twas the nigh
010  74 20 62 65 66 6f 72 65 20 73 74 61 72 74 2d 75 t before start-u
020  70 20 61 6e 64 20 61 6c 6c 20 74 68 72 6f 75 67 p and all throug
030  68 20 74 68 65 20 6e 65 74 2c 0a 20 20 20 20 20 h the net,.
040  6e 6f 74 20 61 20 70 61 63 6b 65 74 20 77 61 73 not a packet was
050  20 6d 6f 76 69 6e 67 3b 20 6e 6f 20 62 69 74 20  moving; no bit
060  6e 6f 72 20 6f 63 74 65 74 2e 0a 20 20 20 54 68 nor octet..   Th
070  65 20 65 6e 67 69 6e 65 65 72 73 20 72 61 74 74 e engineers ratt
080  6c 65 64 20 74 68 65 69 72 20 63 61 72 64 73 20 led their cards
090  69 6e 20 64 65 73 70 61 69 72 2c 0a 20 20 20 20 in despair,.
0a0  20 68 6f 70 69 6e 67 20 61 20 62 61 64 20 63 68  hoping a bad ch
0b0  69 70 20 77 6f 75 6c 64 20 62 6c 6f 77 20 77 69 ip would blow wi
0c0  74 68 20 61 20 66 6c 61 72 65 2e 0a 20 20 20 54 th a flare..   T
0d0  68 65 20 73 61 6c 65 73 6d 65 6e 20 77 65 72 65 he salesmen were
0e0  20 6e 65 73 74 6c 65 64 20 61 6c 6c 20 73 6e 75  nestled all snu
0f0  67 20 69 6e 20 74 68 65 69 72 20 62 65 64 73 2c g in their beds,
100  0a 20 20 20 20 20 77 68 69 6c 65 20 76 69 73 69 .     while visi
110  6f 6e 73 20 6f 66 20 64 61 74 61 20 6e 65 74 73 ons of data nets
120  20 64 61 6e 63 65 64 20 69 6e 20 74 68 65 69 72  danced in their
130  20 68 65 61 64 73 2e 0a 20 20 20 41 6e 64 20 49  heads..   And I
140  20 77 69 74 68 20 6d 79 20 64 61 74 61 73 63 6f  with my datasco
150  70 65 20 74 72 61 63 69 6e 67 73 20 61 6e 64 20 pe tracings and
160  64 75 6d 70 73 0a 20 20 20 20 20 70 72 65 70 61 dumps.     prepa
170  72 65 64 20 66 6f 72 20 73 6f 6d 65 20 70 72 65 red for some pre
180  74 74 79 20 62 61 64 20 62 72 75 69 73 65 73 20 tty bad bruises
190  61 6e 64 20 6c 75 6d 70 73 2e 0a 20 20 20 57 68 and lumps..   Wh
1a0  65 6e 20 6f 75 74 20 69 6e 20 74 68 65 20 68 61 en out in the ha
1b0  6c 6c 20 74 68 65 72 65 20 61 72 6f 73 65 20 73 ll there arose s
1c0  75 63 68 20 61 20 63 6c 61 74 74 65 72 2c 0a 20 uch a clatter,.
1d0  20 20 20 20 49 20 73 70 72 61 6e 67 20 66 72 6f     I sprang fro
1e0  6d 20 6d 79 20 64 65 73 6b 20 74 6f 20 73 65 65 m my desk to see
1f0  20 77 68 61 74 20 77 61 73 20 74 68 65 20 6d 61  what was the ma
```