FAT Filesystem

Assignment 4

Useful resources

Original FAT32 spec by Microsoft: http://msdn.microsoft.com/en-us/library/gg463080.aspx (sometimes hard to read)

Good manual:

http://www.pjrc.com/tech/8051/ide/fat32.html

Wikipedia:

http://en.wikipedia.org/wiki/File_Allocation_Table

FAT32 disk organization - high level

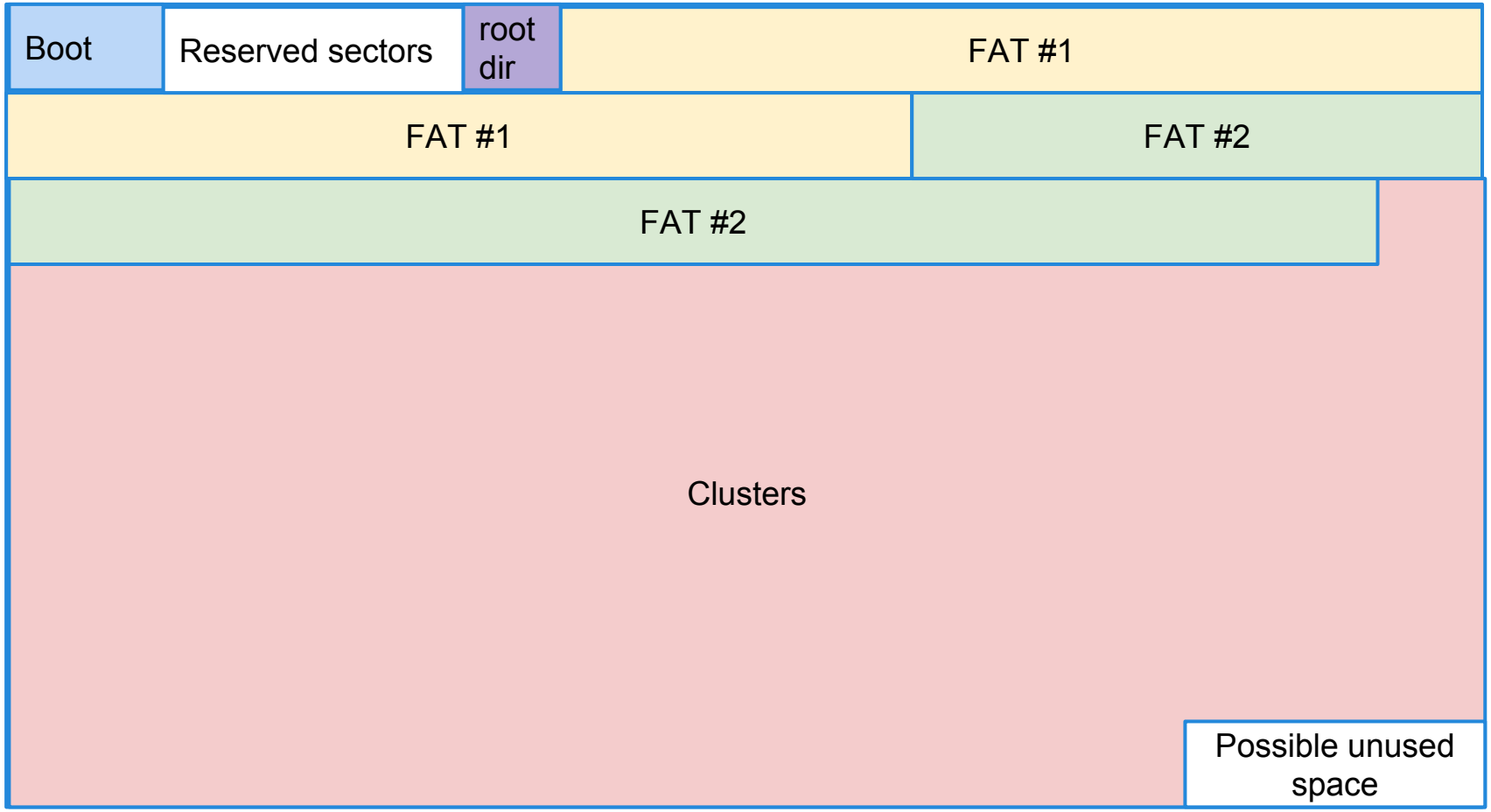
Filesystem =

- files (e.g. data)
- metadata (attributes, filenames, etc.)
- allocation table (e.g. what is where)

FAT32 disk organization

- “Boot records”
  - contains basic information about the filesystem
- Root directory
  - contains root directory entries
  - not present in FAT32!
- FATs
  - Each file is stored in a sequence of clusters
  - FAT has information about these sequences
- Clusters
  - holds data for files (or directories which are just special files)

FAT disk organization



FAT32 Boot sector

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x0B	2 bytes	00 02	**Bytes Per Sector**. The size of a hardware sector. Valid decimal values for this field are 512, 1024, 2048, and 4096.
0x0D	1 byte	10	**Sectors Per Cluster**.The number of sectors in a cluster. The default cluster size for a volume depends on the volume size. Valid decimal values for this field are 1, 2, 4, 8, 16, 32, 64, and 128.

Table taken from http://technet.microsoft.com/en-us/library/cc776720(v=ws.10).aspx

FAT32 Boot sector

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x0E	2 bytes	24 00	**Reserved Sectors**. The number of sectors that precede the start of the first FAT, including the boot sector.
0x10	1 byte	02	**Number of FATs**. The number of copies of the FAT on the volume. The value of this field is always 2.
0x11	2 bytes	00 00	**Root Entries (FAT12/FAT16 only)**. For FAT32 volumes, this field must be set to zero.

FAT32 Boot sector

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x13	2 bytes	00 00	**Small Sectors (FAT12/FAT16 only)**. For FAT32 volumes, this field must be set to zero.
0x15	1 byte	F8	**Media Descriptor**. / Not useful for us/
0x16	2 bytes	00 00	**Sectors Per FAT (FAT12/FAT16 only)**. For FAT32 volumes, this field must be set to zero.

FAT32 Boot sector

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x18	2 bytes	3F 00	**Sectors Per Track**. / Not useful for us /
0x1A	2 bytes	FF 00	**Number of Heads**. / Not useful for us /
0x1C	4 bytes	3F 00 00 00	**Hidden Sectors**. / Not useful for us /

FAT32 Boot sector

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x20	4 bytes	1D 91 11 01	**Large Sectors**. Contains the total number of sectors in the FAT32 volume.
0x24	4 bytes	2A 22 00 00	**Sectors Per FAT (FAT32 only)**. The number of sectors occupied by each FAT on the volume.
0x28	2 bytes	00 00	Unused

FAT32 Boot sector

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x2A	2 bytes	00 00	**File System Version (FAT32 only)**. / Should be zeroes in current FAT32 /
0x2C	4 bytes	02 00 00 00	**Root Cluster Number (FAT32 only)**. The cluster number of the first cluster of the root directory. This value is **typically, but not always, 2.**
0x30	2 bytes	01 00	**File System Information Sector Number (FAT32 only)**. The sector number of the

FAT32 Boot sector

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x32	2 bytes	06 00	**Backup Boot Sector (FAT32 only)**. A value other than zero specifies the sector number in the reserved area of the volume where a copy of the boot sector is stored. The value of this field is typically 6. No other value is recommended.
0x34	12 bytes	00 00 00 00 00 00	**Reserved (FAT32 only)**. Reserved space for future expansion. The value of this field must always be zero.

FAT32 Boot Sector Summary

Contains basic useful information

Check thoroughly if this is FAT32!

- there are many fields that might falsely “indicate” that a filesystem is FAT32
- Microsoft spec has explicit way to do this

File Allocation Table

Linked list of clusters:

FAT[cluster_i] -> next cluster of a file

Example (files starting with clusters 2 and 4)

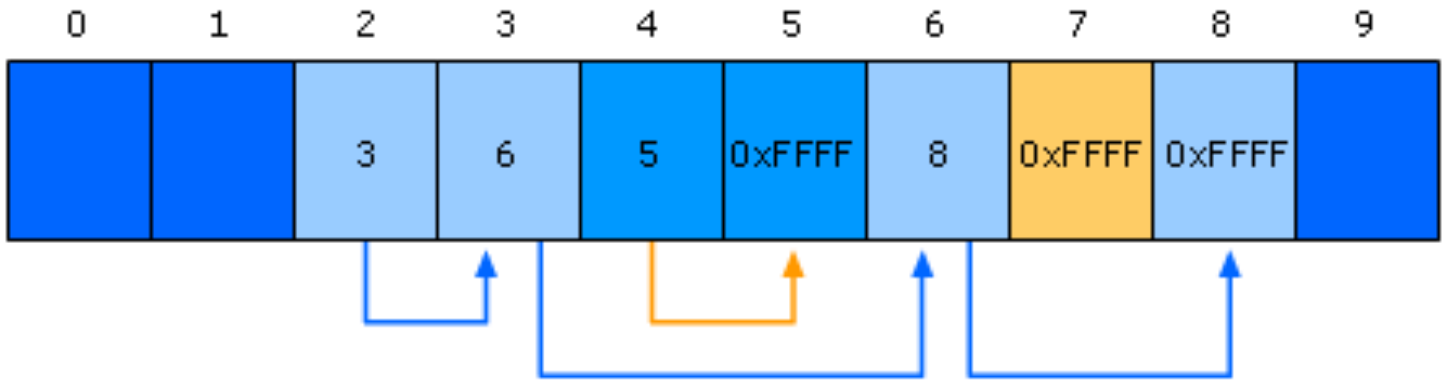


Image taken from http://technet.microsoft.com/en-us/library/cc776720(v=ws.10).aspx

File Allocation Table

Cluster chains

- 0xFFFFFFFx == end of chain
- other special values - bad sector

Note:

- upper 4 bits of value should be masked and are reserved

Directory entries

Directory = special file

- contains set of 32-byte long entries
  - name of subdirectory/file
  - attribute
  - size
  - first cluster of a file

Complications:

- MS-DOS 8.3 vs Windows long names

Directory entries MSDOS 8.3

Field	Microsoft's Name	Offset	Size
Short Filename	DIR_Name	0x00	8+3=11 Bytes
Attrib Byte	DIR_Attr	0x0B	8 Bits
First Cluster High	DIR_FstClusHI	0x14	16 Bits
First Cluster Low	DIR_FstClusLO	0x1A	16 Bits
File Size	DIR_FileSize	0x1C	32 Bits

Directory Entry Attributes

Attrib Bit	Function	LFN	Comment
0 (LSB)	Read Only	1	Should not allow writing
1	Hidden	1	Should not show in dir listing
2	System	1	File is operating system
3	Volume ID	1	Filename is Volume ID
4	Directory	x	Is a subdirectory (32-byte records)
5	Archive	x	Has been changed since last backup
6	Ununsed	0	Should be zero
7 (MSB)	Ununsed	0	Should be zero

Directory entries

Complications:

- size attribute of directory is always 0
  - “end-of-directory”
    - a) special entry, or
    - b) reach end of cluster chain

- MS-DOS 8.3 vs Windows long names

Directory Entries - Long filenames

Hack

- prepend each short entry with several special entries storing long name
- special = attributes: volume id + read-only + system + hidden
- DOS will ignore these entries

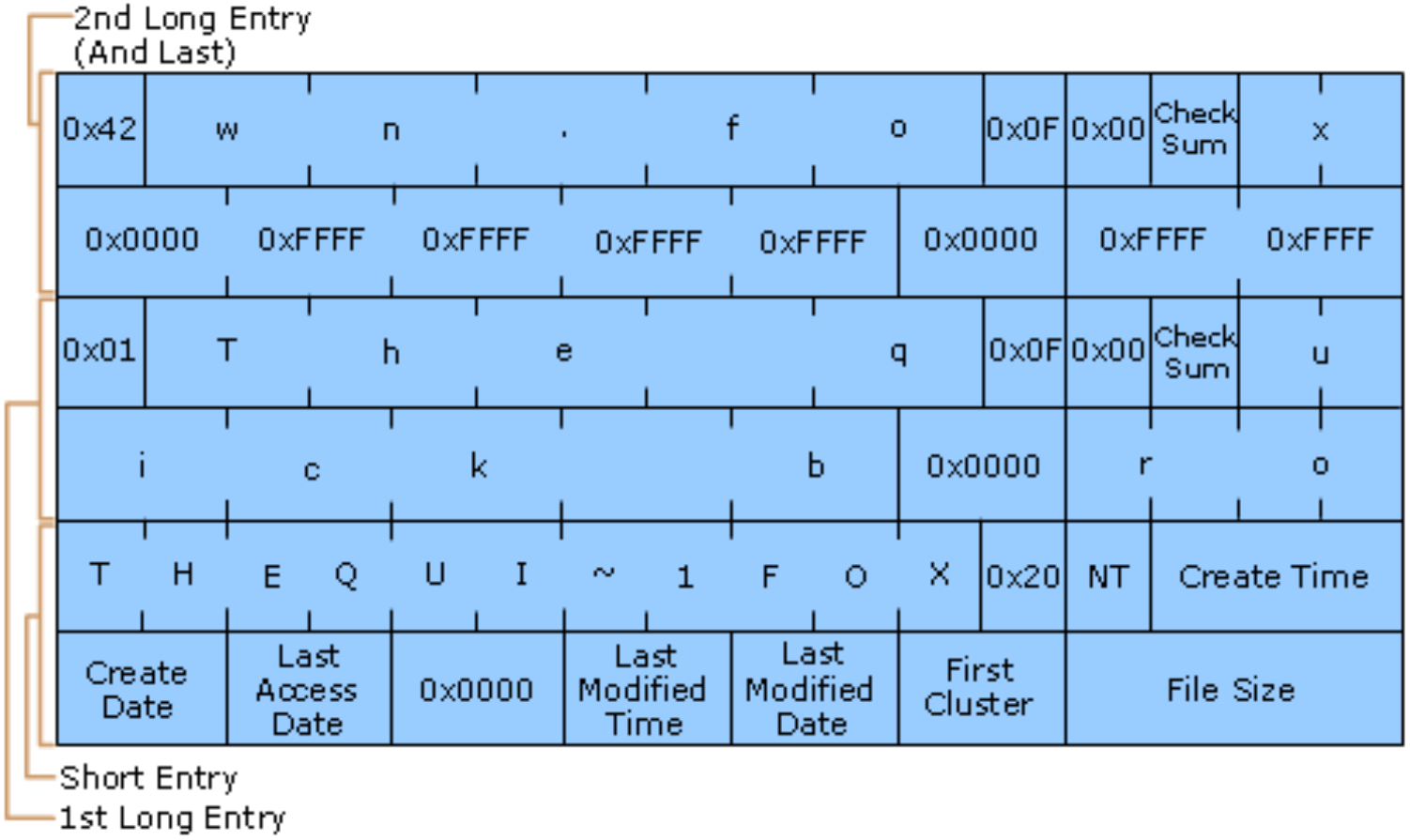
Directory Entries - Long Filenames

Basic layout:

- Long entry last (n)
- Long entry n-1
- ....
- Long entry 2
- Long entry 1
- Short entry

Directory Entries - Long Filenames

Example: “The quick brown.fox”



Taken from http://technet.microsoft.com/en-us/library/cc776720(v=ws.10).aspx

Directory Entry - Long

Byte Offset	Length (bytes)	Description
0x00	1	Sequence Number (bit 6 =last logical, first physical LFN entry; bit 5: 0; bits 4-0: number 0x01..0x14 (0x1F), deleted entry: 0xE5)
0x01	10	Name characters (five UCS-2 characters)
0x0B	1	Attributes (always 0x0F)
0x0C	1	Type (always 0x00 for VFAT LFN, other values reserved for future use; for special usage of bits 4 and 3 in SFNs see below)

Directory Entry - Long

Byte Offset	Length (bytes)	Description
0x0D	1	Checksum of DOS file name
0x0E	12	Name characters (six UCS-2 characters)
0x1A	2	First cluster (always 0x0000)
0x1C	4	Name characters (two UCS-2 characters)

Directory Entry - Long

- be sure to validate that long entries are correct
  - start with last one
  - continue in order
  - are not marked as deleted
  - have same checksum as short entry

Questions?