

An Introduction to FAT 16/FAT 32 File Systems

Dr. William F. Heybruck, Senior Applications Engineer

Hitachi Global Storage Technologies, Charlotte, NC

bill.heybruck@hitachigst.com

Introduction

This paper is written to introduce the casual PC or digital camera user to the topic of File Allocation Tables, commonly known as FAT. This paper will focus on the issues of FAT and how it relates to Compact Flash (CF) devices that generate large file sizes such as 5+ megapixel digital cameras.

The reason that FAT is now important to CF users is that devices which require, or are optimized by using storage capacity greater than 2.14 Gigabytes (GB) require a different type of FAT than those that use fewer than 2.13GB.

Not since the application of a hard disk into Personal Computers (c1983) has this topic been so important. The next section is written for those who are interested in the historical background relative to FAT. If that is not of interest to you, feel free to continue with the Clusters section.

History

When the first Personal Computers were developed, they used floppy disks with a capacity of 180 kilobytes, soon followed by double-sided diskettes of 360KB. These were the 5 1/4-inch diskettes that you probably noticed in your father's PC! In order to organize data on these storage devices, the initial developers of DOS created tables to indicate which sectors belonged to which files and which sectors were vacant. This allowed the Operating System (DOS) to make maximum use of the limited space on these diskettes. Files could be saved without having a contiguous series of sectors available. The data for the files could be broken up and scattered all over the diskette in an organized manner that is now called fragmentation. However, at that time, it was nice to be able to store your data on the same diskette as the operating system so you didn't have to keep switching diskettes in and out during the operation of the program.

On a 360KB diskette there are 768 sectors. Thus, the table had to be big enough to identify each of the 768 sectors and allow some room for growth since the developers knew that larger diskettes were coming. Using one byte per table entry only permitted 512 possible values so that would not do. Two bytes would permit 16384 entries and would be wasteful of disk space. So the decision was made to use 12 bits (1 1/2 bytes) as the length of the table entries to keep track of sector usage on these first

diskettes. This permitted 4096 possible values and if each entry referenced a single 512-byte sector then this FAT table could handle storage devices of up to 2 GB. However, this resulted in long tables that took time to retrieve and could not be kept in memory since memory sizes were usually less than 64 Kilobytes!

So the concept of clusters was incorporated. This increased the minimum element of storage on a device from a single sector to a fixed number of sectors, called a cluster. For those first 180KB diskettes, the cluster size was 2 or 1024 bytes. Thus if one was saving a 500 byte file, it would occupy 1024 bytes on the diskette. This seems wasteful but in reality the cluster concept works great on larger files which most of the system and user files tend to be. Table 1 illustrates the maximum number of clusters that can be represented by a FAT.

FAT SIZE in bits	Number of Clusters
12	4096
16	65536
32	4,294,967,296

Table 1: Clusters per FAT

Clusters

The smallest element of addressable storage on a disk is called a sector and usually holds 512 bytes (or characters). Since most files are much larger than a sector, an even larger amount of storage should be the smallest amount allocated to any particular file. This allocated amount is called a cluster and depending on the capacity of the storage device (diskette, flash card, hard drive) the size of the cluster will vary to get optimum use of the device. On the early 360KB diskettes, the cluster size was 2 sectors (1024 bytes). Using the first 10MB hard disk, the cluster size was increased to 8 sectors (4096 bytes). Typical cluster sizes for compact flash devices today are 8K (8192 bytes) or 16K (16384 bytes). Hard disks up to 2 Gigabytes have a cluster size of 32K. Table 2 illustrates the maximum size of a partition given a cluster size and a FAT type.

Why clusters? The allocation of space on a storage device is somewhat random. On a new device, files are saved in a sequential manner and knowing the starting sector, ending sector and length would be enough to retrieve the file. However after some time, and some files are erased and some files grow in size, it is easy to comprehend that the file will no longer be contained in a contiguous series of sectors. A method is needed to identify which sectors are allocated to certain files and which sectors are available. Having a method to do this per sector would require a table with an entry per sector. Since most files consist of multiple sectors, this table would be overkill. Hence a table where each entry represented a fixed number of sectors makes more sense. That fixed number of sectors is called a cluster.

Cluster Size (bytes)	FAT 12	FAT 16	FAT 32
1024 (1K)	4.1MB	67MB	4.398TB
4096 (4K)	16.7MB	268MB	17.59TB
8192 (8K)	33.4MB	536MB	35.18TB
16384 (16K)	66.8MB	1.07GB	70.36TB
32768 (32K)	134.2MB	2.14GB	140.7TB

Table 2: Drive size

Allocating Sectors to Files

Given that the smallest amount of storage one can allocate to a file (a cluster) depends on the media the file is stored upon, we must have a method that will indicate that a cluster is either available or allocated to a file. If a cluster is allocated to a file, then we must know which part of the file this cluster is allocated to. This is accomplished with a linked list table called the File Allocation Table (FAT). The FAT is merely a list of N numbers, where N is the maximum number of clusters on the device. The number of bits in each entry is called the FAT size and is one of three values, 12, 16 or 32. Early storage devices used a 12-bit (byte and a half) FAT to reduce waste, as 12 bits would accommodate up to 4096 clusters. Given a cluster size of 2 sectors (1024 bytes) a storage device of nearly 4.2 Megabytes could be represented. Larger devices can be handled with a larger cluster size. However, more space would be wasted on smaller files and clusters that were not full. The Operating System determines what size FAT to use based upon the number of clusters it takes to represent the entire disk space. If the disk requires less than 4096 clusters, then a 12-bit fat is used. If it is less than 65536 but more than 4096 then a 16-bit FAT is used. Otherwise a 32-bit FAT is used for the device. The cluster size is specified

in the first sector of the device in a list of numbers called the Bios Parameter Block. This sector can be read by the Operating System during boot so it can configure how to read files from the storage device. The FAT is a linked list table. A linked list table is one where related entries point to one another. For the FAT, the device directory contains the name, size of the file and the number of the first cluster allocated to that file. The entry in the table for that first cluster of that particular file contains the number of the second cluster in that file. This continues until the last cluster entry for that file which will contain all F's indicating it is used and the last cluster of the file. The first file on a new device will use all sequential clusters. Hence the first cluster will point to the second, which will point to the third and so on.

FAT Examples

An empty 16 bit FAT

0	1	2	3	4	5	6	7	8
0000	0000	0000	0000	0000	0000	0000	0000	0000

A 16 bit FAT with a single file entry

0001	0002	0006	0000	0000	0000	0007	FFFF	0000
------	------	------	------	------	------	------	------	------

This FAT illustrates that the first cluster is number 0, the second is number 1, the third is number 2, the fourth is number 6, and the last is number 7.

This FAT has 2 files. One has 3 clusters (0, 1, 8) and the other 2 clusters (2, 6).

0	1	2	3	4	5	6	7	8
0001	0008	0006	0000	0000	0000	FFFF	0000	FFFF

Cluster 0

In reality, the first cluster (cluster 0) is always reserved for operating system information, the root directory and two copies of the FAT. The system creates two copies of the FAT so that should the system be interrupted (crash) while modifying one of the copies, the other copy is still intact and a recovery program may check the FAT and recover it using the copy.

Big Deal?

So what is the big deal about FAT? Up to now, the Compact Flash storage devices have used either a 12-bit or 16-bit FAT. Digital cameras and digital music players have been designed to read and write using this table structure. Since the introduction of the 4GB Microdrive in 2003, 16-bit FAT is not large enough to handle all the sectors. With 16 bits, only 65536 clusters can be represented and with the maximum cluster size of 32K bytes, only 2.14 GB can be accommodated in a single FAT. There are two solutions. Increase the FAT size to 32 bits or split the drive into multiple partitions.

Partitions

A storage device may be split up into multiple partitions (logical entities where each partition appears as a separate storage device). Each partition would then have its own directory and FAT. You may have experienced multiple partitions on your PC hard disk when you only have one hard drive but have a “C:” and “D:” drive not counting your CD-ROM drive.

Most portable devices such as cameras, and digital music players are not programmed to accept partitions and expect the storage device to have a single partition. PDAs on the other hand are really computers and their file system routines are designed for partitions.

All that being said, for the digital camera market, storage devices have to have a single partition to be compatible. Figure 1 illustrates a drive split into one, two or three different partitions. To the user, each partition will have its own logical drive letter and will usually be assigned drives C:, D:, E: etc.

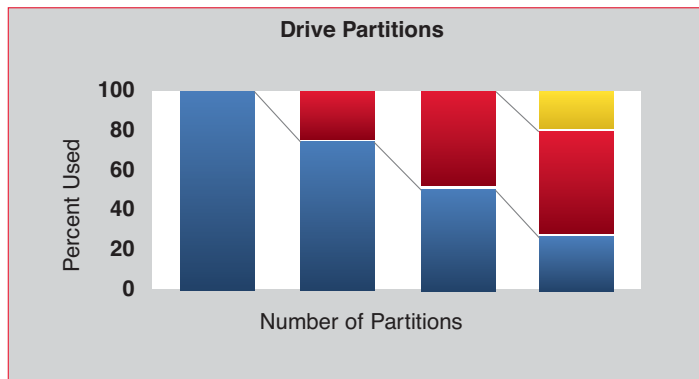


Figure 1

FAT Inter-compatibility

Using a FAT16 storage device in a host compatible with FAT32 will operate seamlessly because operating systems that support FAT32 will also support FAT16. The only caution is when you want to re-format the FAT16 device. Some FAT32 compatible hosts (e.g. some PDAs) will

default to a FAT32 format unless specifically selected. When doing this for the first time, watch the prompts carefully to insure you will keep your FAT16 format unless your camera/PDA/digital music player supports FAT32.

Using a FAT32 device in a FAT16 compatible host will result in several possible outcomes. The most common is that the ubiquitous CF ERROR appears in the display. Should a 4GB or larger capacity device be formatted in a FAT16 only compatible host, the drive formats without error but should indicate a 2GB capacity. Note the word “should”. Different programmers can use different methods to handle this issue and for this unusual case, the result is not always predictable. If a 4GB device ever is formatted with FAT16, one can re-format the device in a FAT32 compliant host, as long as the FAT32 is specified during the format session. This will enable the user to regain the full capacity in a FAT32 compliant host.

Summary

Since cameras and digital music players have difficulty with multiple partitions, FAT 32 is the only answer to provide them with storage in excess of 2.14GB.

Therefore the new Hitachi 6GB and greater capacity Microdrives will be shipped pre-formatted for FAT 32. To provide for compatibility with more devices, the 2GB Microdrive will continue to use FAT 16. Other Compact Flash devices must jump on the FAT 32 bandwagon as their capacities exceed the 2GB barrier.

The good news is that several companies are already supporting FAT 32 in their cameras and digital music players. Most PDAs that have a CF slot support FAT 32 today.

If you are planning to purchase a new digital camera or digital music player it would be a good idea to verify that it does support FAT 32 for future compatibility with the latest CF storage devices.



Hitachi Global Storage Technologies trademarks are intended and authorized for use only in countries and jurisdictions in which Hitachi Global Storage Technologies has obtained the rights to use, market and advertise the brand. The Travelstar trademark is authorized for use in the Americas, EMEA, and the following Asia-Pacific countries and jurisdictions: Australia, Hong Kong, Japan, New Zealand, South Korea and Taiwan. Contact Hitachi Global Storage Technologies for additional information. Hitachi Global Storage Technologies shall not be liable to third parties for unauthorized use of this document or unauthorized use of its trademarks.

References in this publication to Hitachi Global Storage Technologies' products, programs or services do not imply that Hitachi Global Storage Technologies intends to make these available in all countries in which it operates.

Product specifications provided are sample specifications and do not constitute a warranty. Information is true as of the date of publication and is subject to change. Actual specifications for unique part numbers may vary. Please visit the Support section of our website, www.hitachigst.com/support, for additional information on product specifications. Photographs may show design models.

© 2007 Hitachi Global Storage Technologies

Hitachi Global Storage Technologies
3403 Yerba Buena Road
San Jose, CA 95135 USA

Produced in the United States 11/07.
All rights reserved.

Microdrive® is a registered trademark of Hitachi Global Storage Technologies.