More on file systems/exFAT

Ravi Pudipeddi

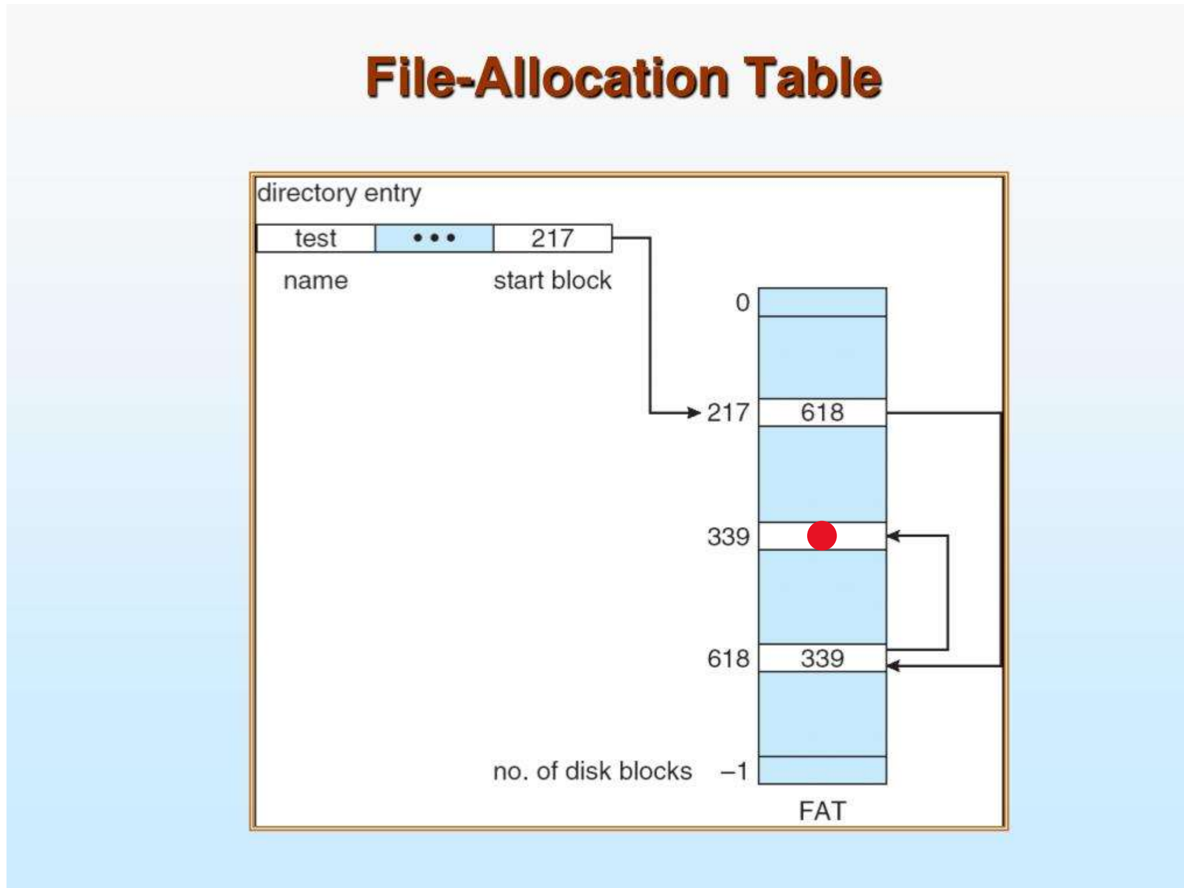
Table of some file systems

FS	FAT32	exFAT	NTFS	APFS
Journaled ?	No	No	Yes	Yes
Purpose	Interoperability	Interoperability	Primary storage (internal)	Primary storage (internal)
Capacity	2TB	128PB	8PB	8TB
Max file size	4GB	128PB	~8PB	8EB

Brief foray into FAT32

FAT File System Structure

FAT32 File System Structure



Limitations of FAT32

- File size limit of 4gb and outlived usefulness for large video files.
- FAT32 has reached limits of extensibility – no more features can be added
- SD card association foresaw >=64gb portable drives with large files
- Not vendor customizable
- No implementation features targeting SSDs

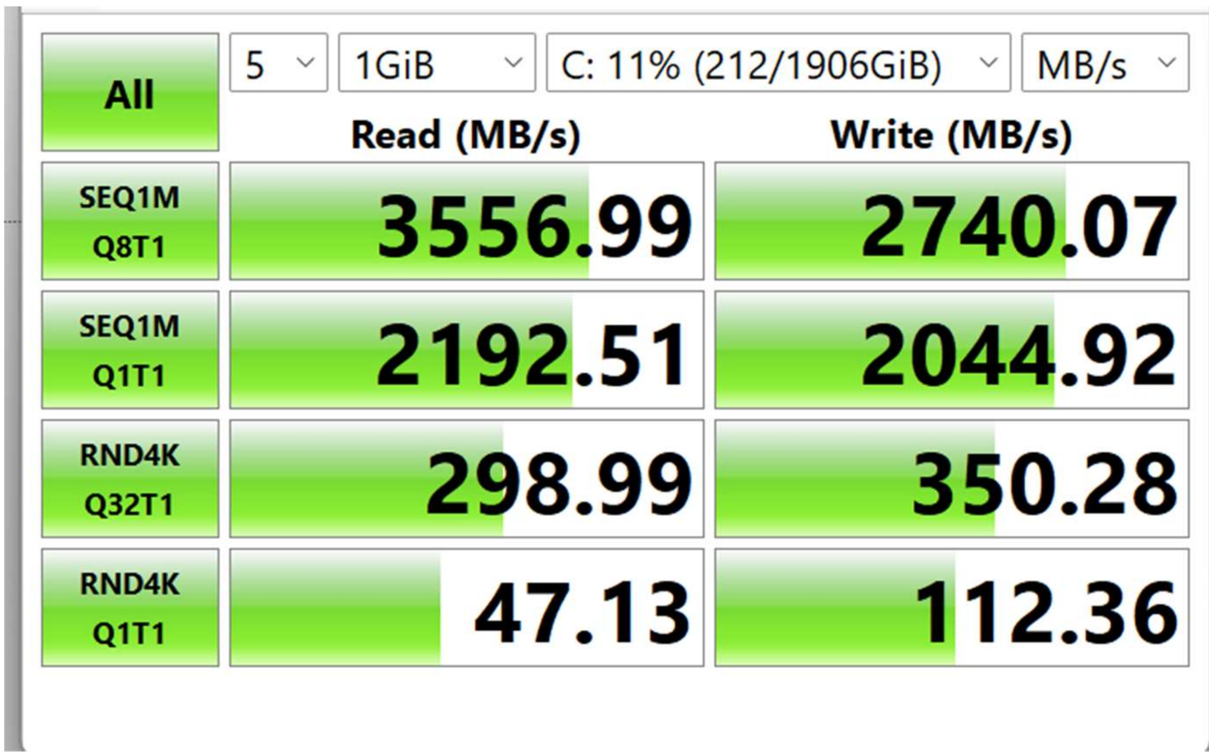
Brief overview of SSDs at h/w layer

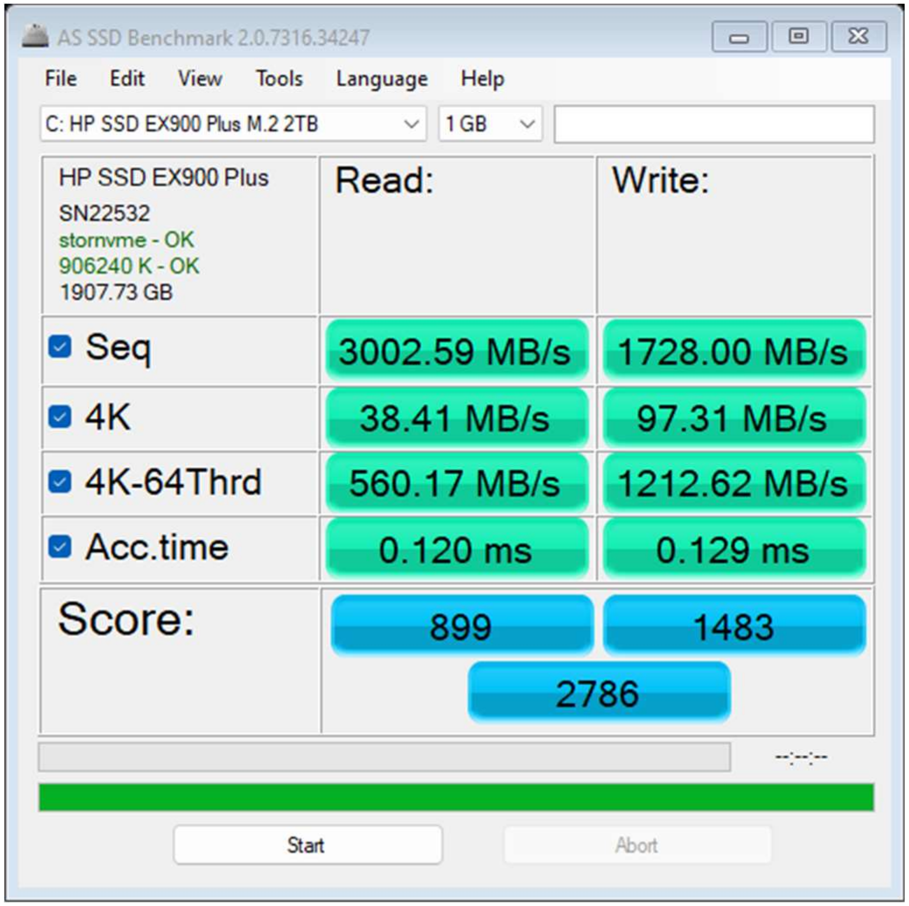
- Flash memory is divided into large **blocks** – 128kb
- If a byte is written to in a block, it cannot be overwritten until the entire block is erased
- Erase (called an erase cycle) is expensive
- Hence blocks are divided into smaller **pages** – say 4k each
- Page is the minimum unit size that is writeable
- Once a page is written to it can be written until entire block is erased and is marked 'dirty'
- Flash translation layer (FTL) remaps writes to pages that are already dirty to a clean page from a different block
- Once a threshold is reached for dirty pages it is taken out and put in a pool for asynchronous erase cycle
- It is important to have a healthy amount of clean blocks so writes can be satisfied immediately

Flash performance issues

- Flash storage comes with a surplus of spare blocks which are ‘hidden’
- As long as there is a healthy amount of clean blocks flash memory operates in ‘burst mode’ for writes – highest perf mode for device
- When the threshold for clean blocks drops, FTL has to slow down writes to increase erase cycles to make more clean blocks under duress
- As one uses up more and more capacity in a flash drive write performance slows down
- Hence small capacity flash drives – usb sticks which are filled up quickly and overwritten frequently degrade over time
- There are a limited number of erase cycles each block can withstand – which is a durability concern
- Bottomline : reduce fragmentation within blocks so ideally an entire block is filled up with dirty pages and can be cleaned

Some sample flash performance numbers – 2TB fresh ssd – what is the anomaly? ☺




AS SSD Benchmark 2.0.7316.34247
File Edit View Tools Language Help
C: HP SSD EX900 Plus M.2 2TB
1 GB
HP SSD EX900 Plus
SN22532
stornvme - OK
906240 K - OK
1907.73 GB
Read:
Write:
Seq
3002.59 MB/s
1728.00 MB/s
4K
38.41 MB/s
97.31 MB/s
4K-64Thrd
560.17 MB/s
1212.62 MB/s
Acc.time
0.120 ms
0.129 ms
Score:
899
1483
2786
Start
Abort

Types of file systems considered for portable SSDs

- “Log structured” file systems so writes are always sequential and flash-friendly – however this is redundant since the FTL already does that below the file system
- A new journaled file system (JFFS was proposed)
- Interoperability is paramount
- Simplicity of the file system so many kinds of devices can read and write – mp3 players, cameras, navigation systems ….
- The challenge was primarily tradeoffs.
- Simplicity ruled them all : for thousands of different devices to implement something as complicated as an FS it has to be at the lowest common denominator
- Extensibility : vendors need proprietary extensions like adding permissions or key vendor specific metadata

Enter exFAT

- exFAT primary goals were to summarize
  - Simplicity
  - Extensibility – vendors can add support for ‘multiple streams’ for a file, permissions etc.
  - Permit sequential file allocation
  - Easily specified upfront so that many vendors can implement the specification with few ‘bugs’
  - Performance : FAT32 allocation performance suffered due to lookups for free clusters
  - Faster file name matching for lookups
  - Detection of corruption
  - 64-bit file sizes

Key technical elements of exFAT

- Bitmap for allocation
- File allocation units are blocks (clusters) of 512-byte sectors with default size 4k
- Bitmap aggressively reserves space for new files so that new allocations for the file are contiguous.
- Vendors can choose their ‘contiguity’ of allocation
- Extensibility is supported by *typing* directory/file entries
- **Type** of an entry indicates if it is necessary for a vendor to implement that type or can be ignored if it is just benign metadata and such

Design details

- Each directory consists of a series of directory entries. Each entry is exactly 32-bytes
- Directory entries are classified as critical/benign and primary/secondary as follows:
  - Critical Primary
    - Allocation bitmap
    - Up-case table
    - Volume label
    - File
  - Benign primary
    - Volume GUID
  - Critical secondary
    - Stream extension
    - File Name
  - Benign primary
    - Vendor extension
    - Vendor allocation
- A linear file allocation table with 64-bit entries used for indicating clusters belonging to a file. Each entry represents a cluster (allocation unit) of the file. Each allocated cluster entry points to the next entry for the allocated unit. End of file is a special value.

Allocation/Performance

- A cluster bitmap for faster allocation
- Scanning for free clusters no longer requires traversal of the FAT table itself
- A per-file contiguous bit that allows non fragmented files to bypass the FAT table entirely
  - -Useful for recording movies or writing photos which can be sequentially written and contiguously allocated
- Better alignment of the FAT table and cluster heap
- On-disk storage of file Valid Data Length (VDL)

Metadata integrity

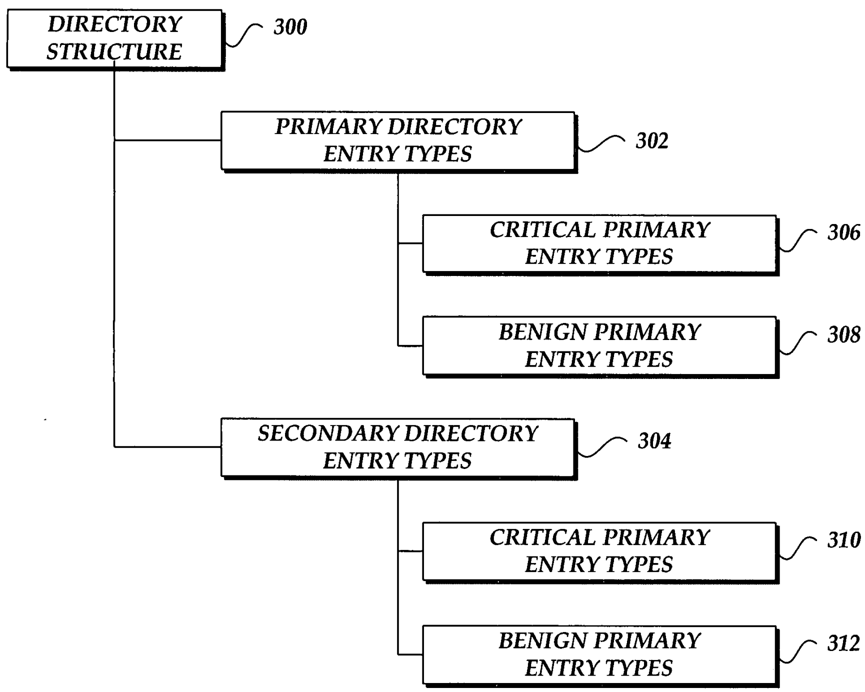
- Checksum of directory file sets. Multiple directory records are used to define a single file – a file set.
- File set has metadata including the file name, time stamps, attributes, address of first cluster location of the data, file lengths, and the file name.
  - A checksum is taken over the entire file set, and a mismatch would occur if the directory file set was accidentally or maliciously changed.
- Other metadata has individual checksums
- Checksums compensate for journaling by detecting ‘torn writes.’ The idea is that only a part of data is lost if checksum files instead of the entire volume

File name lookup

- File names are hashed with checksums for quick compare
- Still a linear search is needed to create new files (detect collisions) or opening a new file
- For large portable flash drives with ‘lots of files’ creation/lookup empirical median perf improvement was anywhere from 40-60%

Conclusions

- Different file systems needed for different purposes
- File systems are complex
- A widely deployed and *developed* file system needs to be as simple as possible but no simpler
- exFAT is now the industry standard for all portable storage devices >=64gb (SDXC mandated)


DIRECTORY STRUCTURE
300
PRIMARY DIRECTORY ENTRY TYPES
302
CRITICAL PRIMARY ENTRY TYPES
306
BENIGN PRIMARY ENTRY TYPES
308
SECONDARY DIRECTORY ENTRY TYPES
304
CRITICAL PRIMARY ENTRY TYPES
310
BENIGN PRIMARY ENTRY TYPES
312

Bonus: ReFS – FS for SMR drives

- SMR drives are *shingled magnetic drives* – for the same hard drive platter SMR records 20% more capacity by ‘shingling’ blocks. 8TB drives or greater are typically SMR drives
- SMR drives are like SSDs in terms of write limitation: drive is divided into 256k blocks – once written to it has to be erased before another write happens
- Extremely expensive if used natively
- ReFS is a log-structured file system that always does append only writing.