

Sistema de arquivos exFAT (Extended File Allocation Table)

Augusto Ribas, Bruno Nazário, Douglas Sorgatto, Thiago Machado

Sistemas Operacionais

25/10/2014

Visão Geral

1 Introdução

- Introdução
- Origem
- Para que outro sistema de arquivos?
- Mercado

2 Especificações Técnicas

3 Implementação de um sistema de montagem de disco exFAT

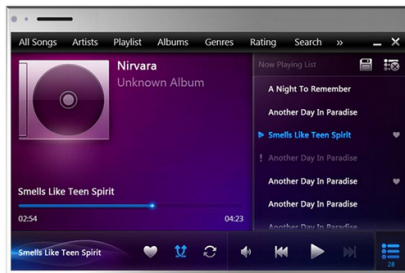
Introdução

O exFAT (Extended File Allocation Table) é o sistema de arquivos da Microsoft otimizado para flash drives. É um sistema proprietário e patentiado.

Esse sistema pode ser usado onde o NTFS não é uma solução viável ou onde o limite do sistema padrão FAT32 não é aceitável.

Origem

exFAT começou como parte do Windows CE (Windows para sistemas mobile, de 1996, precursores do Windows Phone).
Otimizando o uso de discos SSD.



File systems optimized for flash memory, solid state media

Discos de estados solidos (Solid state media), como memorias flash, são similares a discos quanto a suas interfaces, mas possuem problemas diferentes. No nivel mais baixo, eles requerem tratamento especial como diferentes algoritmos de tratamentos de erros. Tipicamente, um dispositivo como um solid-state disk lida com tais operações internamente e assim um sistema de arquivo regular pode ser usado. Entretanto para certas instalações especializadas (como sistemas embarcados), um sistema de arquivos otimizado para o disco é necessario.



Motivação

O exFAT permite arquivos individuais maiores que 4 Gigabytes, facilitando a gravação de vídeos longos em HD, que podem exceder 4Gb em menos de uma hora. Cameras usando Fat32 vão quebrar os arquivos de vídeo em múltiplos segmentos de 2 a 4 Gb. Com o aumento de capacidade e dos dados sendo transferidos, a operação de escrita precisa ser feita de forma mais eficiente.

Motivação

SDXC(Secure Digital eXtended Capacity, especificação de armazenamento de dados para disco sd) cards, rodando UHS-1 tem uma velocidade minima garantida de 10MBps e o exFAT tem um papel importante em reduzir a sobrecarga na alocação de blocos (cluster). Isso é possível através da introdução do blocos(cluster) de bitmap e eliminação (ou redução) das escritas em tabela FAT. Um unico bit (flag) no registro de directorio é usado para informar ao driver de disco exFAT quando um arquivo é contínuo.



U — "UHS Speed Class" mark

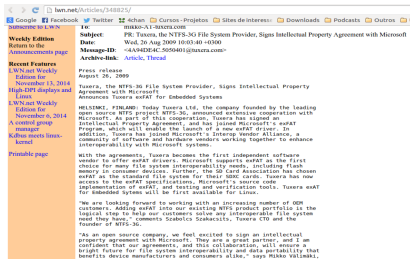


C — "Speed Class" mark

	Marks	Speed	Operable Under...	Applications
Speed Class	C	≥10 MB/s	High Speed Bus I/F	Full HD video recording HD still consecutive recording
	C C	25 MB/s; 34 MB/s	Normal Bus I/F	HD ~ Full HD video recording
	C	≥2 MB/s		SD video recording
new UHS Speed Class	U	≥10 MB/s	UHS-I Bus I/F	Full higher potential of recording real-time broadcasts and capturing large-size HD videos

Comercialização

Alguns vendedores de discos e memórias, incluindo os da pendrives USB, memória flash e solid state drives (SSD) tem mandado de fabrica esses dispositivos pre-formatados com o sistema de arquivos exFAT.



Adoção

exFAT tem suporte no windows (XP, server 2003, CE 6.0, Vista, server 2008, 7, 8), Mac OS X Lion, OS X Mountain Lion, OS X Mavericks e OS X Yosemite.

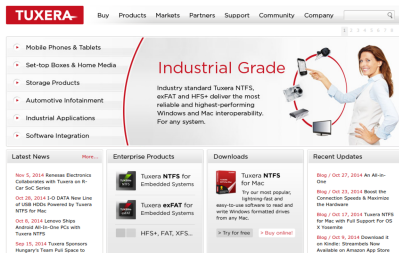
Companias podem integrar o exFAT em um grupo especifico de dispositivos para usuarios, incluindo cameras, filmadoras, porta retrados digitais, smartphones, pcs e redes por um preco variavel. A microsoft entrou com acordos de licenciamento com a BlackBerry, Panasonic, Sanyo, Canon, Aspen Avionics e BMW.

Adoção

Uma implementação FUSE-based (Filesystem in Userspace) chamada de fuse-exfat ou exfat-fuse, com funções de leitura e escrita esta disponível para FreeBSD e distribuições linux. Uma implementação para o kernel também foi feita pela Samsung. ela foi liberada no github inicialmente, mas de forma não intencional, sendo posteriormente liberada oficialmente pela Samsung com uma licença GPL. No entanto nenhuma das soluções pode ser incorporada oficialmente ao kernel do linux devido as patentes da Microsoft com relação ao exFAT.

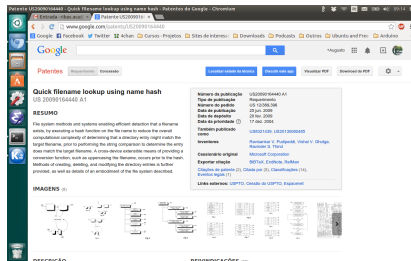
Adoção

Soluções proprietária de escrita e leitura licenciadas e derivada da implementação da Microsoft são disponíveis para Android, Linux e outros sistemas operacionais pela Paragon Software Group e Tuxera.



Localização de nome de arquivo

Em 2012, a microsoft ganhou a patente US8321439, referente ao "Quick File Name Lookup Using Name Hash", que é o algoritmo usado no exFAT para acelerar a procura por nomes de arquivos.

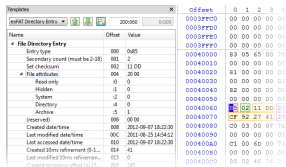


Pré-alocação de Arquivo e Cluster

Como o NTFS, o exFAT pre-aloca espaço em disco para um arquivo apenas marcando um espaço arbitrário no disco como "alocado". Para cada arquivo, o exFAT usa dois campos separados de 64 bits no diretório, o VDL (Tamanho válido de dados, ou Valid Data Length), que indica o tamanho real do arquivo, e o tamanho físico dos dados (physical data length). Para melhorar a alocação de blocos (cluster) para o armazenamento de um novo arquivo, a Microsoft incorporou um método de pré-alocação-contíguo para blocos (cluster) que ignora a tabela FAT.

Pré-alocação de Arquivo e Cluster

Uma característica do exFAT (usada no exFAT para sistemas embarcados) prove transações atômicas para o update os metadados do sistema de arquivos em multiplos passos, essa característica chamada de Transaction Safe FAT (TexFat).



The screenshot shows the Disk Editor application with a File Directory Entry selected. The entry details are as follows:

Name	Offset	Value
File Directory Entry		
Entry type	000	0x05
Secondary count (must be 2-35)	001	2
Set checksum	002	11 00
File attributes	004	20 00
Read only	0	0
Hidden	1	0
System	2	0
Directory	4	0
Archive	5	1
(reserved)	006	00 00
Created date/time	008	2012-08-07 18:22:30
Last modified date/time	00C	2012-08-23 14:04:12
Last accessed date/time	010	2012-08-07 18:22:30
Created 28ms refinement (0-1)	014	41
Last modified 28ms refinement	015	0
Checksum	016	380

The right pane shows the raw hex data for the entry, with the File Directory Entry structure highlighted in blue.

<http://www.disk-editor.org/>

Directory file set

exFAT e o resto da família FAT para sistemas de arquivos não usa índices por nomes de arquivos, diferente do NTFS que usa arvores B para a procura de arquivos. Quando um arquivo é acessado, o diretório precisa ser buscado sequencialmente até que uma busca produza um resultado. Para nomes de arquivos menores que 16 caracteres, um registro de nome é requerido, mas o nome do arquivo é representado por três registros de diretórios de 32 bytes. Isso é chamado de "directory file set", e um diretório de 256 MiB pode ter até 2,796,202 nomes de arquivos. (Se arquivos tiverem nomes maiores, o número de nomes de arquivos diminuirá, mas o mínimo é de 3 arquivos por diretório).

Directory file set

Para ajudar a melhorar a busca sequencia de diretorio (incluindo o raiz), um valor de hash de cada nome de arquivo é calculado e guardado no registro do diretorio. Quando fazemos uma busca por um arquivo, o nome do arquivo é primeiro convertido para letras maiusculas (UPPER CASE) e então usado na função hash (Função hash proprietaria). Cada registro nos diretorios é então vasculhada comparando o valor hash, quando um registro igual é encontrado, os nomes dos arquivos é comparado para garantir que o registro é o certo. Isso aumenta a performance porque apenas dois caracteres precisam ser comparador por arquivo. Isso reduz significativamente o número de ciclos do CPU necessarios, ja que a maioria dos arquivos tem nomes de mais de 2 caracteres e cada comparação fica fixa a 2 caracteres até encontrar um match.

Metadata and checksums

O exFAT introduz a integridade de metadados através do uso de checksums. Existem três checksums atualmente em uso.

Volume Boot Record (VBR), que é uma região de 12 setores que contém os registros de boot. BIOS Parameter Block (BPB) OEM parameters and the checksum sector, este é uma verificação dos 11 setores anteriores. Com exceção de três bytes no setor de boot (Flags e porcentagem usada).

Isso garante a integridade do VBR ao determinar se o VBR foi modificado. Seu problema mais comum pode ser um vírus que altera o setor de boot, mas isso também pode ser causado por outros problemas no VBR.

Metadata and checksums

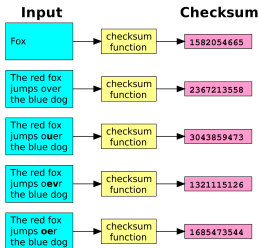
Um segundo checksum é usado para a "upcase table". Esta é uma tabela estática e nunca deve ser alterada. Qualquer dano a essa tabela pode impedir que arquivos sejam localizados já que esta tabela é usada para converter os arquivos para o upper case quando a busca por um arquivo é realizada.

Metadata and checksums

Uma terceira checksum é a do arquivos de diretório. Múltiplos registros de diretório são usados para definir um único arquivo e isso é chamado de file set. Esse file set tem metadados incluindo o nome do arquivo, time stamps, atributos, endereço do da localização do primeiro cluster e tamanho do arquivo. Um checksum é feito para todo o fileset, podendo detectar alterações acidentais ou maliciosas. Quando o sistema de arquivos é montado, as verificações de integridade são conduzidas e todos os hashes são verificados. A montagem também inclui comparações de versão do sistema de arquivos exFAT pelo driver para garantir que o driver é compatível com o sistema de arquivo que está tentando montar, e para garantir que nenhum dos registros dos diretórios estão perdidos (ou corrompidos). Se algum dessas verificações falhar, o sistema de arquivo não deve ser montado, apesar de que em alguns casos, ele pode ser montado como apenas para leitura.

Exemplo

Checksum são funções de hash que transformam um conjunto de dados de qualquer tamanho em um conjunto de bits (ou número) de tamanho determinado. Ele é útil para verificar a integridade de dados de forma eficiente, já que a verificação se torna um número de bits constante. Sendo raro o evento onde dois arquivos vão produzir exatamente o mesmo número.



exFAT root

```
struct exfat
{
    struct exfat_dev* dev;
    struct exfat_super_block* sb;
    le16_t* upcase;
    size_t upcase_chars;
    struct exfat_node* root;
    struct
    {
        cluster_t start_cluster;
        uint32_t size; /* in bits */
        bitmap_t* chunk;
        uint32_t chunk_size; /* in bits */
        bool dirty;
    }
    cmap;
    char label[UTF8_BYTES(EXFAT_ENAME_MAX) + 1];
    void* zero_cluster;
    int dmask, fmask;
    uid_t uid;
    gid_t gid;
    int ro;
    bool noatime;
};
```

<https://code.google.com/p/exfat/>

Node de exFAT

```
struct exfat_node {  
    struct exfat_node* parent;  
    struct exfat_node* child;  
    struct exfat_node* next;  
    struct exfat_node* prev;  
  
    int references;  
    uint32_t fptr_index;  
    cluster_t fptr_cluster;  
    cluster_t entry_cluster;  
    off_t entry_offset;  
    cluster_t start_cluster;  
    int flags;  
    uint64_t size;  
    time_t mtime, atime;  
    le16_t name[EXFAT_NAME_MAX + 1];  
};
```

<https://code.google.com/p/exfat/>

Restrictive licensing and software patents

A micro\$oft não liberou oficialmente as especificações do sistema de arquivos exFAT e deixou este com uma licença restritiva, a qual é necessária para fazer e distribuir implementações completas do sistema exFAT. A micro\$oft ainda reivindica algumas patentes de softwares que torna difícil reimplementar todas as funcionalidades requeridas pelo sistema de arquivos exFAT sem violar uma destas patentes.

Isso torna a implementação, distribuição e uso do exFAT como parte da comunidade livre de sistemas de código livre e softwares comerciais difícil, para vendedores que não conseguem obter a licença da micro\$oft, especialmente em países que reconhecem as patentes de Software dos Estados Unidos.

Restrictive licensing and software patents

Apesar do sistema exFAT ser amplamente suportado atualmente, em dispositivos eletrônicos e Mac OS X, inicialmente eles só podiam lidar com formatos FAT12/FAT16/FAT32, o que tornava o exFAT impraticável como um sistema universal de armazenamento de dados.

O suporte para o exFAT em sistemas Linux ainda é limitado devido a essas barreiras.