# Analise a dados armazenados

**Conceitos**

- Dados são guardados em suportes de armazenamento
    - Discos, cartões de memoria, pens USB, ...
- Suportes de armazenamentos tem estrutura própria
    - Estrutura guarda em MBR ou GPT
    - Encontram-se divididos em partições
    - Cada partição tem um sistema de ficheiros
- Sistemas de ficheiros organizam a forma como se armazenam os dados
    - FAT (12,16,32), NTFS, EXT (2,3,4), UFS, ...
    - Alguns sistemas de ficheiros são de conhecimentos publico, outros são proprietários.

**Estrutura de um disco**

- Os tipos de discos mais comuns são os discos organizados por
    - Master Boot Record (MBR)
    - GUID Partition Table (GPT)
- O MBR/GPT é armazena no início do disco, contendo a sua estrutura (tabela de partições)
- Cada partição utiliza um sistema de ficheiros
- Reparticionar um disco não apaga dados, apenas a tabela de partições

Master Boot Record (MBR)

- Ocupa os primeiros 512 bytes do disco
    - Inclui apontadores para 4 partições primarias (que podem ou não estar em uso)
- Partições adicionais (alem destas 4)
    - Requer que seja marcada como partição estendida
    - Partição estendida usa Extended Boot Recor (EBR)
    - EBR pode conter apontador para um EBR seguinte
        - Número ilimitado de partições estendidas

Exemplo de MBR:

```
aap@pc:~$ hexdump -n 512 -C usb-mbr.dd

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*
01b0 00 00 00 00 00 00 00 00 60 9d b9 ec 00 00 00 00
01c0 21 00 06 2a ea ca 20 00 00 00 e0 b7 3b 00 00 00
01d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*
01f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 #55 AA#
0200
```

| Endereço | | Descrição | | Tamanho (bytes) |
|---|---|---|---|---|
| Hex | Dec | | | |
| +0x0000 | +0 | Código de arranque (*boot*) | | 446 |
| +0x01BE | +446 | Partição #1 | Tabela de partições primárias | 16 |
| +0x01CE | +462 | Partição #2 | | 16 |
| +0x01DE | +478 | Partição #3 | | 16 |
| +0x01EE | +494 | Partição #4 | | 16 |
| +0x01FE | +510 | 55 | Assinatura | 2 |
| +0x01FF | +511 | AA | | |
| **Tamanho total : 446 + 4×16 + 2** | | | | **512** |

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

Entradas de Partição:

```
aap@pc:~$ hexdump -s 446 -n 16 -C usb-mbr.dd
000001be 00 00 21 00 06 2a ea ca 20 00 00 00 e0 b7 3b 00
000001ce
```

A 1ª estrada de partição inicia na posição 446 e tem 16 bytes de tamanho!

| Posição relativa | Descrição | Tamanho (bytes) |
|---|---|---|
| 0 | Indicador de *boot* (80h) | 1 |
| 1 | Inicio de partição (CHS) | 3 |
| 4 | Tipo de partição | 1 |
| 5 | Fim de partição (CHS) | 3 |
| 8 | Setor inicial (LBA) | 4 |
| 12 | Tamanho da partição (em setores) | 4 |
| | **Tamanho total:** | **16** |

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

Tipos de numeração de setores no disco

- CHS - *Cylinder/Head/Sector* (mais antigo)
- LBA - *Logical Block Addressing* (mais recente)

Exemplo de Partição:

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | **80** | 01 |
| 1C0 | 01 | 00 | **0B** | 1F | 3F | 33 | 3F | 00 | 00 | 00 | 41 | 99 | 01 | 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 | 55 | AA |

http://thestarman.pcministry.com/asm/mbr/PartTables.htm

0x1BE Partição está ativa (valor 80h)

0x1BF Setor inicial da partição CHS(0,1,1)

0x1C2 Tipo de partição (0B → FAT32)

Extended Boot Record:

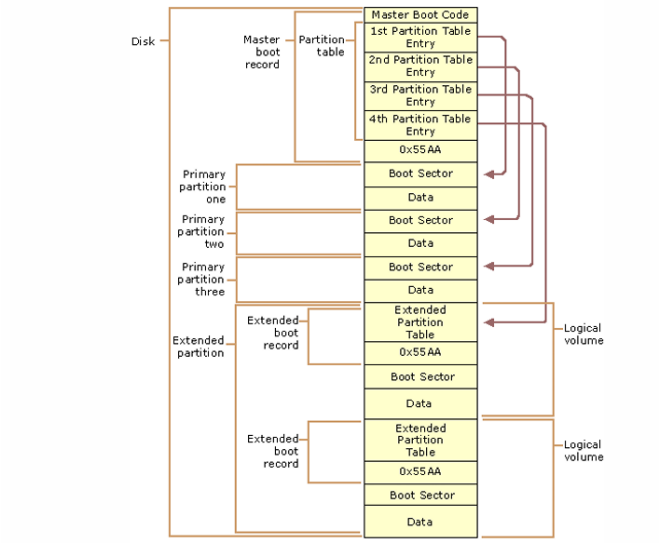
Estrutura:

| Posição relativa ao início do EBR | | Descrição | Tamanho (bytes) |
|---|---|---|---|
| Hex | Dec | | |
| 000 – 1BD | 000 – 445 | Tipicamente vazia (zeros) | **446** |
| 1BE – 1CD | 446 – 461 | **Partição #1** – Descreve a partição atual | **16** |
| 1CE – 1DD | 462 – 477 | **Partição #2** – Descreve a próxima partição | **16** |
| 1DE – 1ED | 478 – 493 | **Partição #3** – Não utilizada (zeros) | **16** |
| 1EE – 1FD | 494 – 509 | **Partição #4** – Não utilizada (zeros) | **16** |
| 1FE - 1FF | 510 - 511 | Assinatura | **2** |
| | | **Tamanho total: 446 +4×16 +2 =** | **512** |

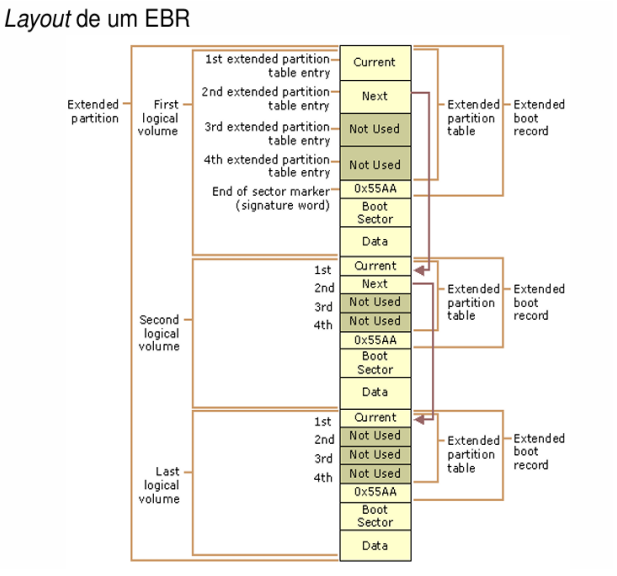
http://en.wikipedia.org/wiki/Extended_boot_record

Têm a mesma estrutura que uma MBR *partition entry*

*Layout* de um disco com MBR




*Layout* de um EBR



GUID Partition Table (GPT)

- GUID Partition Table (GPT) é a forma de organização de discos incluída na norma Unified Extensible Firmware Interface (UEFI)
- Utiliza endereçamento por LBA (blocos de 512 bytes)
- Suporta partições de tamanho superior aos em MBR
- Pode ser utilizada em alguns sistemas com BIOS, desde que suportado pelos sistemas operativos (ex: Linux)
- Primeiro bloco de 512 bytes (LBA-0) é ignorado (por ser o espaço normalmente utilizado pelo MBR) ou inclui Protective MBR

Estrutura genérica

| Posição relativa | Descrição | Tamanho (bytes) |
|---|---|---|
| LBA0 | Não utilizado (MBR) | 512 |
| LBA1 | Cabeçalho GPT principal | 512 |
| LBA2 | Entradas das partições 1 a 4 (128 bytes cada) | 512 |
| LBA3 | Entradas das partições 5 a 128 (128 bytes cada) | 512 |
| . . . | | ... |
| LBA33 | | 512 |
| . . . | Dados / Partições | ... |
| LBA-33 | Cópia das Entradas das partições 5 a 128 (128 bytes cada) | 512 |
| . . . | | ... |
| LBA-3 | | 512 |
| LBA-2 | Cópia das entradas das partições 1 a 4 (128 bytes cada) | 512 |
| LBA-1 | Cópia do cabeçalho GPT | 512 |

LBA-1 refere-se ao último LBA do disco.

Estrutura do cabeçalho GPT

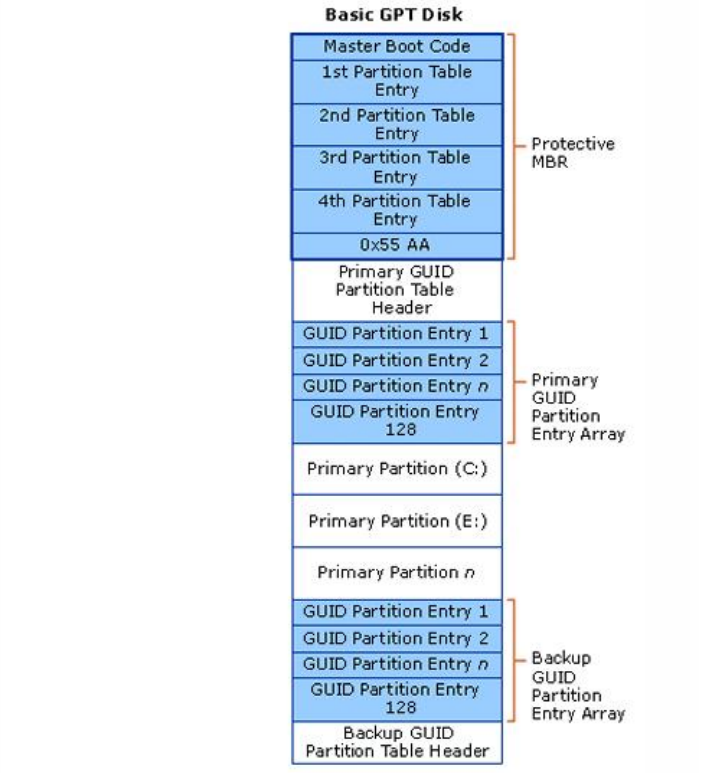
| Posição relativa | Tamanho | Contents |
|---|---|---|
| 0 (0x00) | 8 bytes | Assinatura (45h 46h 49h 20h 50h 41h 52h 54h) |
| 8 (0x08) | 4 bytes | Revisão (00h 00h 01h 00h para GPT ver 1.0) |
| 12 (0x0C) | 4 bytes | Tamanho (5Ch 00h 00h 00h = 92 bytes) |
| 16 (0x10) | 4 bytes | CRC32 do cabeçalho |
| 20 (0x14) | 4 bytes | Reservado (zero) |
| 24 (0x18) | 8 bytes | LBA do cabeçalho |
| 32 (0x20) | 8 bytes | LBA da cópia do cabeçalho |
| 40 (0x28) | 8 bytes | Primeiro LBA útil |
| 48 (0x30) | 8 bytes | Último LBA útil |
| 56 (0x38) | 16 bytes | GUID do disco |
| 72 (0x48) | 8 bytes | LBA da lista de partições |
| 80 (0x50) | 4 bytes | Número de partições na lista |
| 84 (0x54) | 4 bytes | Tamanho da cada entrada de partição (128 bytes) |
| 88 (0x58) | 4 bytes | CRC32 da lista de partições |
| 92 (0x5C) | * | Reservado, zeros até ao fim |

Estrutura de entrada de partição em GPT

| Posição relativa | Descrição | Tamanho (bytes) |
|---|---|---|
| 0 | GUID do tipo de partição | 16 |
| 16 | GUID da partição (único) | 16 |
| 32 | LBA inicial | 8 |
| 40 | LBA final | 8 |
| 48 | Atributos | 8 |
| 56 | Nome da partição (UTF-16LE) | 72 |
| | **Tamanho total:** | **128** |

https://en.wikipedia.org/wiki/GUID_Partition_Table

*Layout* de um disco com GPT



## FAT: *File Allocation Table*

- Sistema de ficheiros simples, muito popular
- Utilizado primeiramente em sistemas DOS, Windows
- Atualmente é utilizado em *pens* USB, cartões de memória, *smartphones*

## FAT12, FAT16, FAT32

- Versão indica o número de bits utilizado para referenciar *clusters* no disco.
  - FAT12 $\rightarrow$ 12 *bits* $\rightarrow$ Max: $2^{12} = 4,096$ *clusters*
  - FAT16 $\rightarrow$ 16 *bits* $\rightarrow$ Max: $2^{16} = 65,536$ *clusters*
  - FAT32 $\rightarrow$ 32 *bits* $\rightarrow$ Max: $2^{32} = 4,294,967,296$ *clusters*
- *Cluster* é um conjunto de sectores
- Sector é a unidade mínima de armazenamento de dados

## Utilização do disco

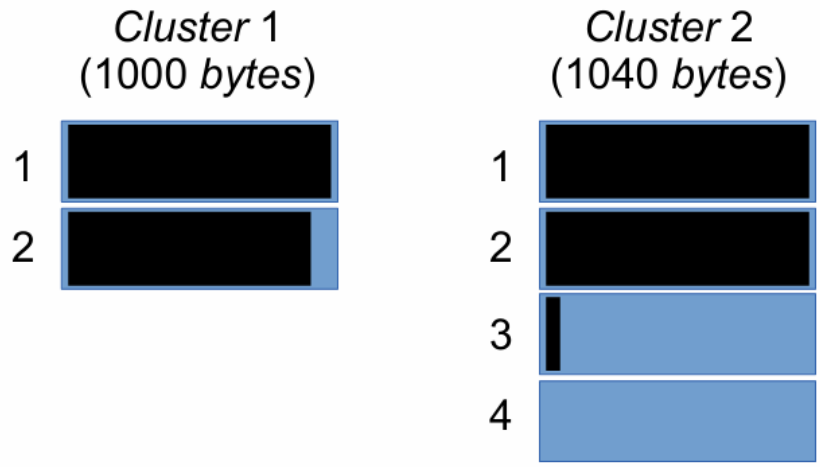
- Sector tem usualmente 512 *bytes* de tamanho
- Sector é o tamanho mínimo para operações de leitura/escrita no disco
- Sempre que se utiliza um sector, este é considerado como totalmente ocupado
  - Se se guardar 10 *bytes* num sector, restantes 502 *bytes* são *desperdiçados*
  - *Bytes* não sobrescritos, mantém dados anteriores

## Utilização do disco (2)

- Espaço em disco é alocado a ficheiros em conjuntos de sectores
- Número de sectores por *cluster* tem de ser uma potência de 2 (1,2,4,...)
- *Cluster* é a unidade mínima de alocação de ficheiros

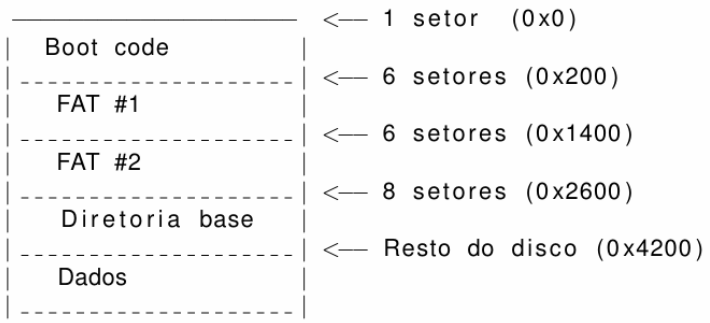
## Exemplos de *clusters*

*Cluster* 1 usa 2 sectores, já o *cluster* 2 usa 4 sectores




Espaço não utilizado no fim de cada ficheiro é chamado de folga (ou *slack*)

## *Layout* de uma partição FAT

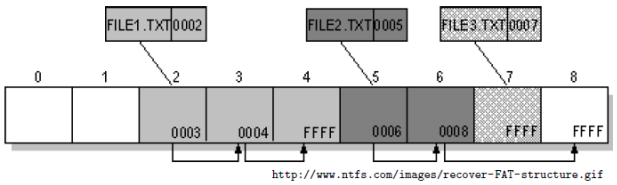



- *Boot code* costuma estar vazio
- Diretoria base conhecida como *root directory*
- FAT #2 é uma cópia de segurança da FAT #1

## Tipos de entradas em *FAT #1*

- Não utilizado (0x0000 0000)
- *Cluster* com erro (0xFFFF FFF7)
- Último *cluster* de um ficheiro (0xFFFF FFF8)
- Número do próximo *cluster* de um ficheiro

## *Exemplo de FAT #1*



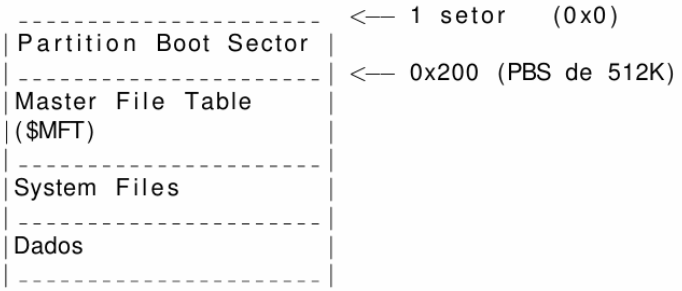

- FILE1.TXT: Ocupa *clusters* 2, 3 e 4
- FILE2.TXT: Ocupa *clusters* 5, 6 e 8 (Fragmentado)
- FILE3.TXT: Ocupa *cluster* 7

# Sistemas de ficheiros NTFS

NTFS: *New Technologies File System*

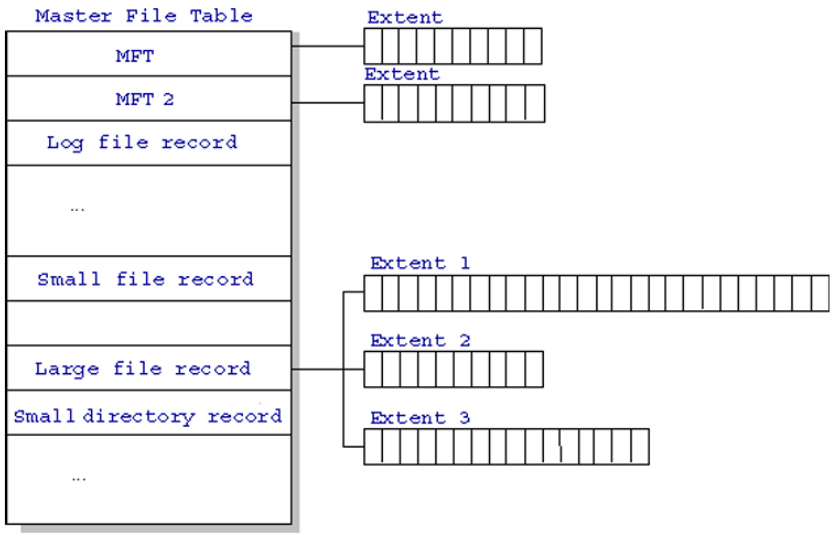
- Sistema de ficheiros utilizado em Windows NT, 2000, ...
- Sistema proprietário da Microsoft
- Suporte para ficheiros superiores a 4GB
- Todos os registos são ficheiros (mesmo a própria $MFT)

# *Layout* de uma partição NTFS



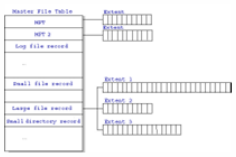

- *Partition Boot Sector* poder ocupar de 1 a 16 setores
- Usa vários ficheiros de sistema (com metadados), como o $MFT, $Bitmap, $LogFile, ...
- Mantém assinatura 0x55AA (posição 0x1FE)

# Estrutura do *MFT*



## Estrutura do *MFT* (2)

- Registo ocupa 1KB
- Cada registo pode conter um directório ou ficheiro (até aprox.512 *bytes*)
- Primeiro registo é a própria $MFT
- Segundo registo é uma cópia de segurança do $MFT ($MftMirr)

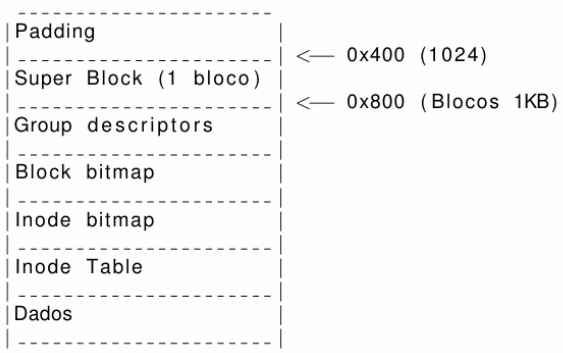


## Sistemas de ficheiros EXT

EXT: *Extended File System*

- Vai atualmente na versão 4 (EXT4)
- Estrutura tem se mantido ao longo da sua evolução
- Novas versões traduzem novas funcionalidades

## *Layout* de uma partição EXT




- Organizado em blocos (de 1KB até 64KB)
- Bloco inicial de 1024 *bytes* é ignorado

## Representação de números

- Binário - Representação digital (0-1)
- Octal - Representação digital compacta (0-7)
- Decimal - Representação humana (0-9)
- Hexadecimal - Representação digital compacta (0-F)

## Representação de números

Valor depende da posição

Numero $25102_d$

- $2 \times 10^4 + 5 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 2 \times 10^0$
  - Número 2 à esquerda vale mais (**mais significativo**)
  - Número 2 à direita vale menos (**menos significativo**)
- **Big-endian** - *byte* mais significativo em primeiro lugar
- **Litle-endian** - *byte* menos significativo em primeiro lugar

## Representação de números

Ordem dos dados

- Processadores *big-endian*
  - Sparc, PowerPC, MIPS . . .
- Processadores *litle-endian*
  - z80, x86, x86-64, adm64 . . .
- Processadores programáveis (*big/litle*)
  - ARM . . .
- Redes *big-endian*
  - Redes IP (com excepções)

## Representação de carateres

Codificação ASCII

(American Standard Code for Information Interchange)

- Caracter ocupa um byte (sem problemas de *endianness*)
- Versão original usa apenas 7 bits
- Ocupa menos espaço que *unicode*
- Múltiplas versões estendidas (8 bits)
- ISO-8859 (latin-1) é mais comum

## Representação de carateres

Codificação *Unicode*

- Representa carateres da generalidade das línguas
- Várias versões
  - UTF-8: 1 a 4 *bytes*, compatível com ASCII
  - UTF-16: 2 *bytes* ou 4 bytes
  - UTF-32: 4 *bytes* (fixo)

| Posição | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| ISO-8859 | 4F | 6C | **E1** | |
| UTF-8 | 4F | 6C | **C3** | **A1** |
| Texto | O | l | á | |