



UNIVERSIDADE
VILA VELHA
ESPIRITO SANTO

Discos e Partições

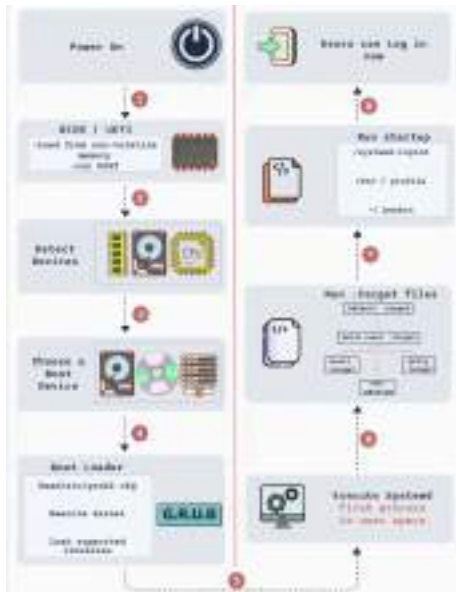
Prof. Jean-Rémi Bourguet

Sistemas Operacionais

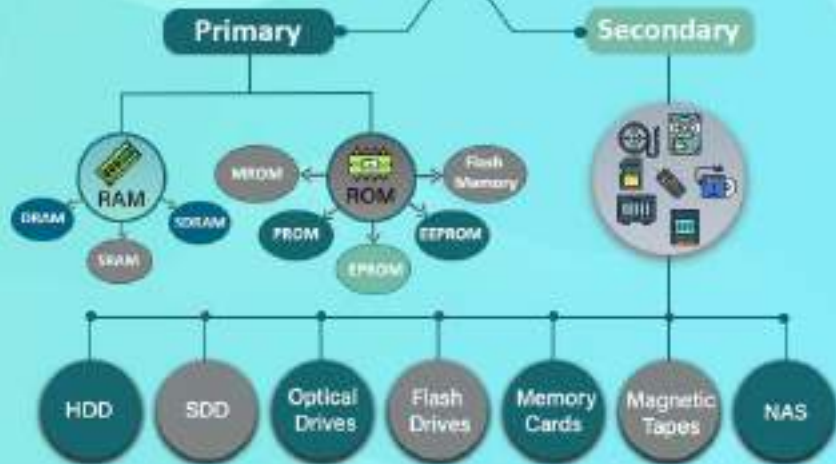
Boot

Pull oneself up by one's bootstraps

Se levantar pelos próprios meios

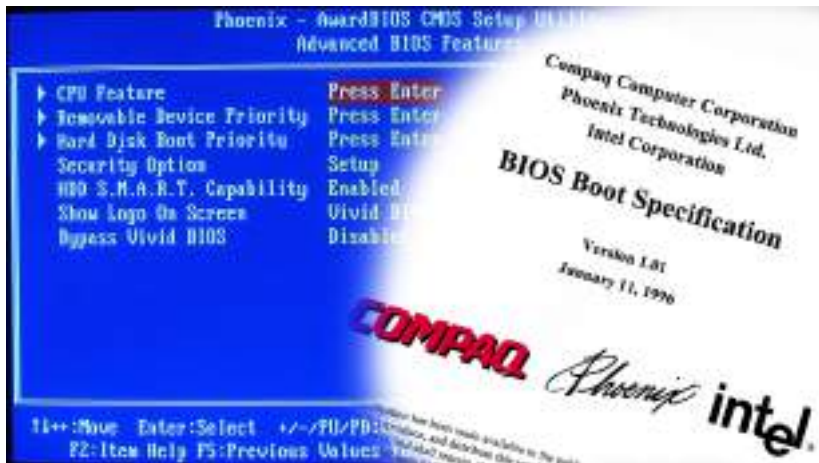


Types of Memory in a Computer



ROM

- **BIOS**: **B**asic **I**nput/**O**utput **S**ystem: gravado em um chip de **ROM**.



- ▶ O **chip CMOS** é uma memória **RAM** alimentado por **bateria** 10 anos.
- ▶ Armazena **configurações do BIOS**: seqüência boot, senha, data/hora.



ROM

- ▶ **PROM**: Pode ser escrita com dispositivos especiais, não modificável (1956).
- ▶ **EPROM**: Erasable PROM. Pode ser apagada - radiação ultravioleta (1971).
- ▶ **EEPROM**: Electrically Erasable PROM. Modificável - carga elétrica (1978).



ROM

- ▶ **Flash** ROM: **memória não-volátil** + moderna derivada da EEPROM.
- ▶ Hoje os firmware são **reprogramável** (e.g. **atualização do BIOS**).



Memória secundária

- **Memória secundária** "fora" do processador e armazena big quantity.



História dos discos rígidos

- ▶ **Disco rígido, HD** ou **memória de massa** ou **memória secundária**.
- ▶ **50 discos magnéticos, 50 000 setores** capacidade 5 MB (IBM, 1956)



305 RAMAC: Random Access Method of Accounting and Control.

História dos discos rígidos

- ▶ Em **1980**, Seagate lançou o **HD de 5MB para microcomputadores**.

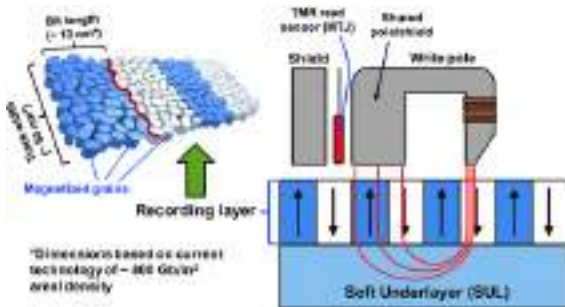


Discos rígidos

- O disco rígido **dominou o mercado até** meados dos **anos 2010**.

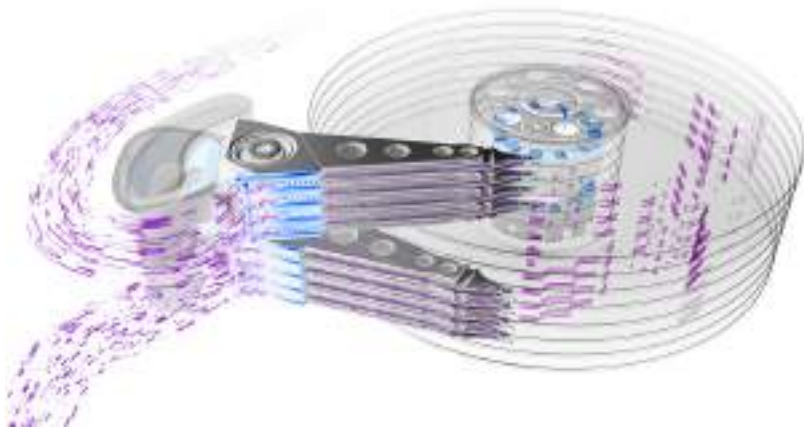


- **Camada magnética fina e cabeça de leitura/gravação (eletroímã).**

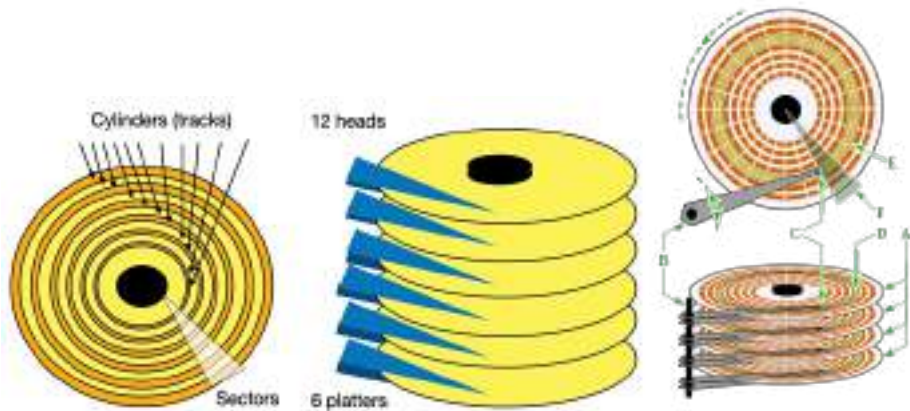


Discos, trilhas, setores

- ▶ **Discos** são as **superfícies físicas** onde os dados são **gravados**.
- ▶ **Trilhas** são **círculos concêntricos** nas **duas superfícies** de **cada disco**.

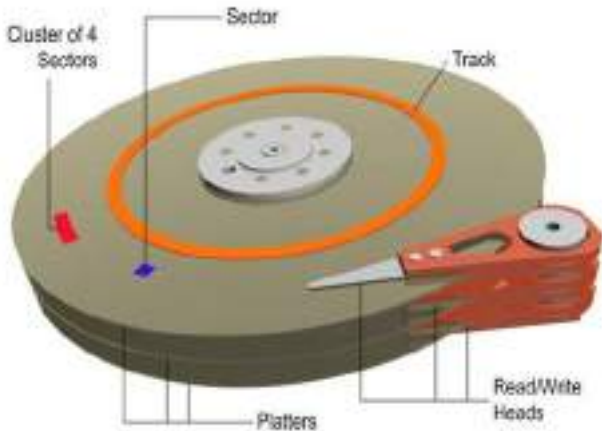


- ▶ Os **discos antigos** utilizavam o método de **endereçamento CHS**.
- ▶ **Cylinder-Heads-Sectors**: Trilha, Cabeças R/W, Campo do ID do setor.



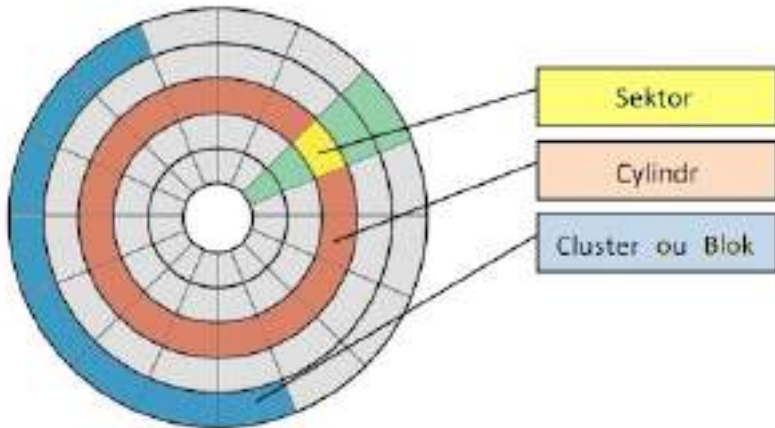
Blocos

- Um **cluster** ou um **bloco** é um grupo de setores.

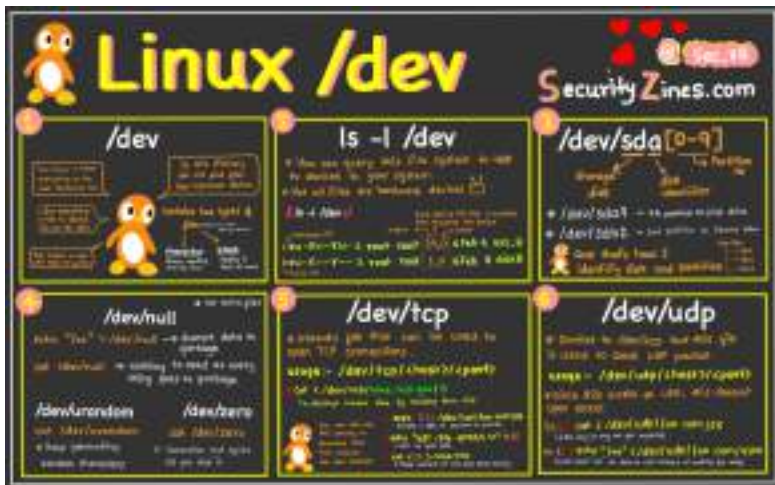


Blocos

- ▶ Nos discos, os **setores** têm **tamanho fixo** (geralmente **512 bytes**).
- ▶ Um **bloco** possui **8 setores** de **512 bytes** ou seja **4096 bytes** (4KB).



- **Arquivo de dispositivo:** interface para um driver de dispositivo.



🔺 RHA Chapter 13: Acesso a sistemas de arquivos Linux

Dispositivos baseados em caracteres

- Recebe **fluxo de caracteres** (não endereçável) para **E/S sem buffer**.

```
$ ls -l /dev
```

```
crw-rw---- 1 root tty 4, 63 May 4 13:13 tty0
```

```
crw-rw---- 1 root tty 4, 63 May 4 13:13 tty1
```

```
$ ls -l /dev/pts
```

```
crw--w---- 1 tux tty 136, 63 May 4 13:13 0
```

```
crw--w---- 1 tux tty 136, 63 May 4 13:13 1
```

```
c----- 1 root tty 5, 63 May 4 13:13 ptmx
```



Dispositivos baseados em blocos

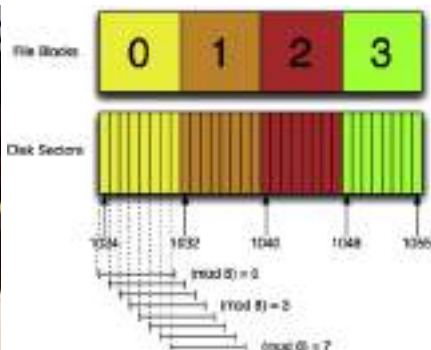
- Liste os dispositivos de bloco não montados em disco loopback.

```
ken@unix:~$ ls -lS /dev | grep "^b" | egrep -v "loop.+"
```

```
brw-rw---- 1 root disk 8, 0 oct. 8 19:20 sda
```

```
brw-rw---- 1 root disk 8, 1 oct. 8 19:20 sda1
```

```
brw-rw---- 1 root disk 8, 2 oct. 8 19:20 sda2
```



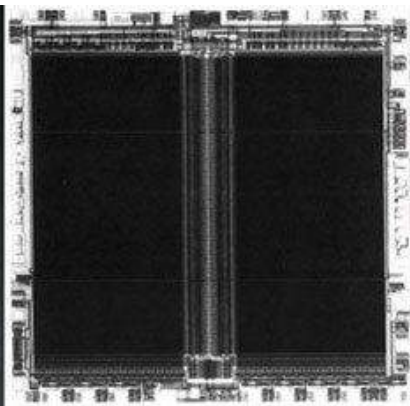
Dispositivos baseados em blocos

- **Arquivo de dispositivo** é montado para **ler a partição** e se tornar **FS**.



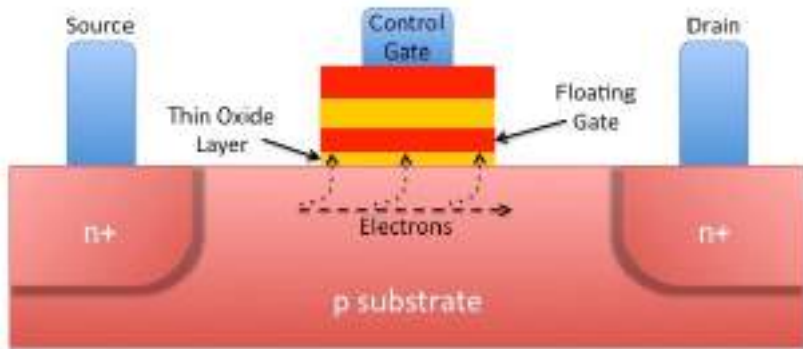
Memória flash

- ▶ Em **1980**, **Dr. Masuoka** (**Toshiba**) inventou e divulgou em 1984 (IEEE)
- ▶ Intel viu o **potencial** e introduziu o **1º chip flash NOR**, em 1988.



Memória flash

- ▶ Normalmente, o transistor **não é condutivo** e **bloqueia** e^- (nível 0).
- ▶ Aplicando **voltagem positiva alta: corrente pode atravessar** (nível 1).



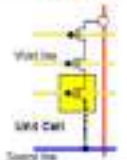
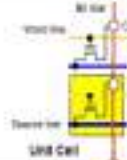
- ▶ **Para apagar**, basta **descarregar todas as cargas** da porta (nível 0).



<https://www.youtube.com/watch?v=gxwE0-hcHTg> (opt)

Memória flash

- ▶ A memória flash NAND permite **alta densidade** de armazenamento.

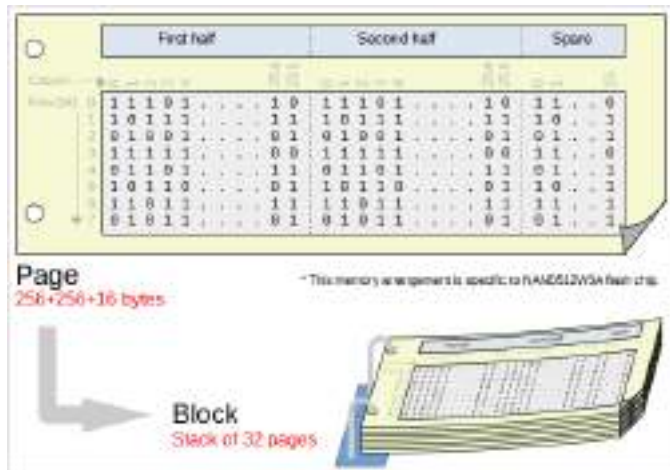
	NAND	NOR
Cell Array		
Cell size	$4F^2$	$10F^2$



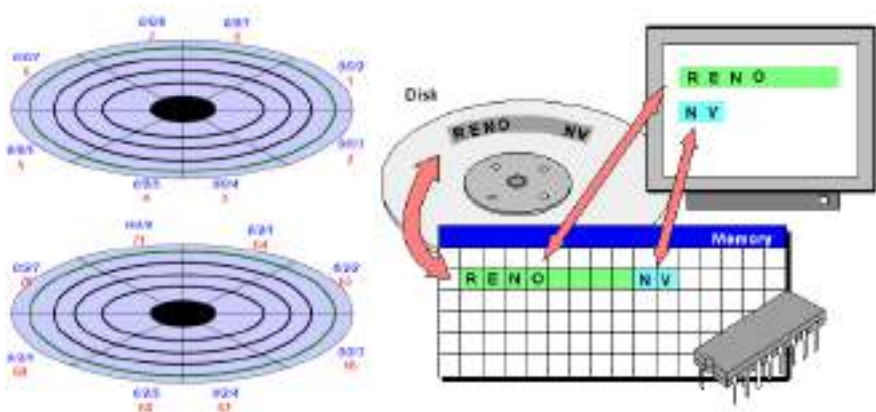
- ▶ A memória flash NOR permite **acesso rápido** aos dados de leitura.

Memória flash

- ▶ Nas unidades SSD, **não há trilha ou setor, apenas há blocos.**
- ▶ Um **bloco** é uma **sequencia de páginas** (setor é abstração do SSD).



- ▶ Logical **B**lock **A**ddressing usado em HD (com LBA) e SSD.
- ▶ Moderno, **linear** e **sequencial**, numerando setores de **forma contínua**.



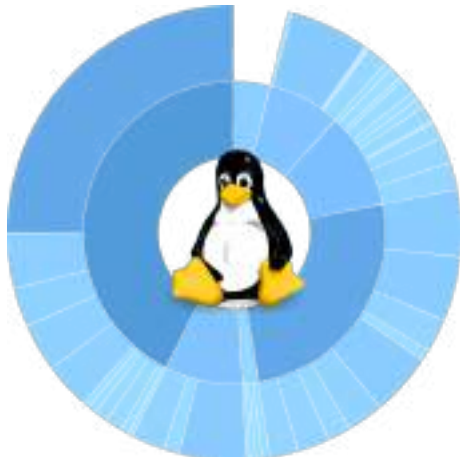
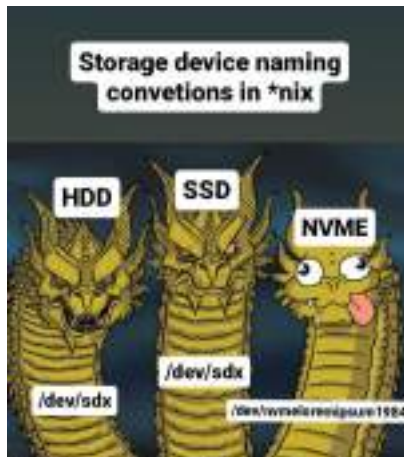
NVMe

- ▶ **NVMe**: **N**on-**V**olatile **M**emory Express é um meio **super-rápido**!
- ▶ Acessa os dados em **até 7 vezes mais rápido** do que **SSDs SATA**.



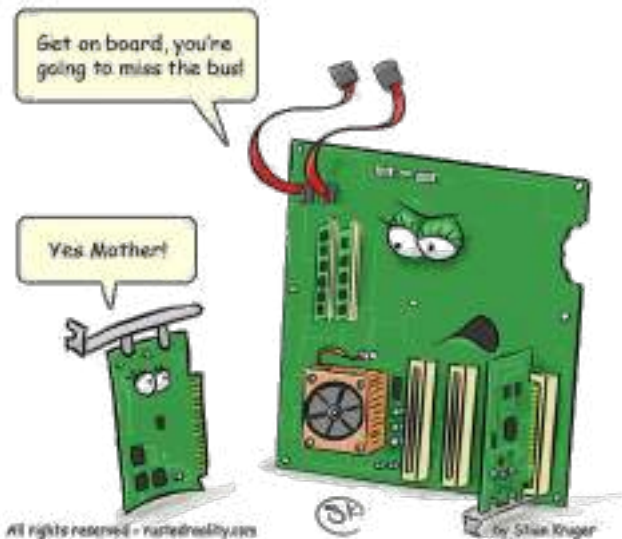
Nomenclatura

- ▶ Existe uma **nomenclatura** para **referenciar discos e partições**.
- ▶ **Tipos de interfaces** para "discos": IDE/ATA, Serial ATA, SCSI, SAS.



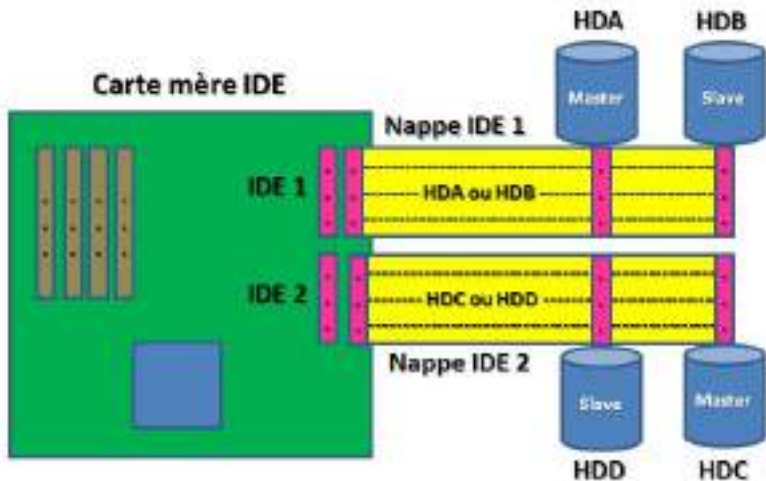
Nomenclatura

- **Placas-mãe** possuíam **2 interfaces IDE** (Integrated **D**rive **E**lectronics).



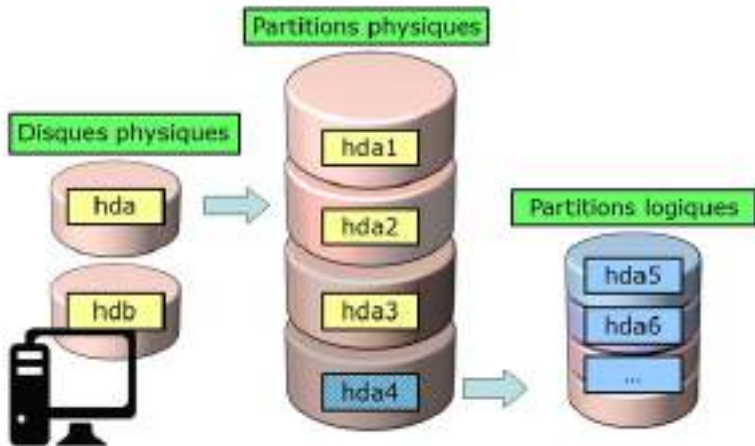
Dispositivos IDE

- ▶ Aparecem no **/dev**: /dev/hda, /dev/hdb, /dev/hdc e /dev/hdd.
- ▶ **/dev/hdi** não se referem aos hard disks, mas ao **dispositivo IDE**.



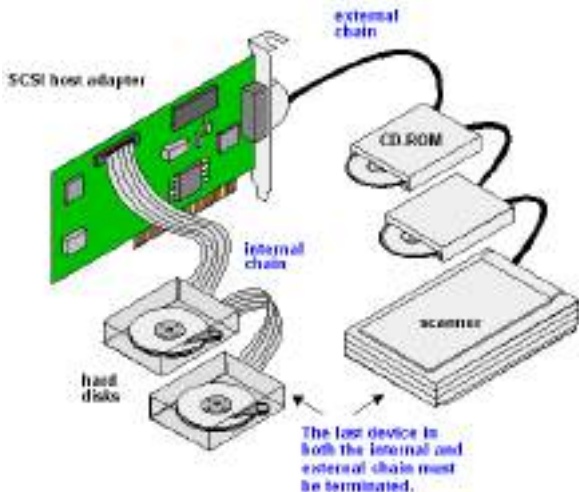
Tipo de partições

- ▶ As **partições primárias** e a **estendida** são **numeradas de 1 a 4**.
- ▶ As **partições lógicas** recebem **numeração de 5 em diante**.



Dispositivos SCSI, SAS, SATA e USB

- **SCSI**: Small Computer System Interface: barramento para periféricos.
- Pode **conectar** vários dispositivos **na mesma linha** (daisy chain).



Dispositivos SCSI, SAS, SATA e USB

- **sd**: sCSI dISK : **não** são **tratados por hd**, como no caso dos IDE.



Comando df (disk free)

- ▶ Mostra o **uso dos discos**: número de blocos utilizados e livres.
- ▶ df mostrará **também o espaço ocupado pela área de controle**.

Chave	Função
-h	Mostra resultados em notações humanas.
-T	Mostra os filesystems utilizados pelas partições.



lsblk: identificação do dispositivo de blocos

- **lsblk** lista os **detalhes** de um determinado **dispositivo de blocos**.

\$ **lsblk**

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
loop1	7:39	0	44,5M	1	loop	/snap/mari0/6
sda	259:0	0	238,5G	0	disk	
—sda1	259:1	0	512M	0	part	/boot/efi
—sda2	259:2	0	238G	0	part	/
sdb	8:0	1	3,6G	0	disk	/media/tux/lab

LSBLK COMMAND IN LINUX

Comando fdisk

- Este comando permite **gerenciar partições**.

Chave	Função
-l /dev/[disk part]	lista as partições (do disco atual por padrão).
-s /dev/[disk part]	lista o tamanho da partição em blocos.
-b tamanho	fixa tamanho do setor (512, 1025, 2048, 4096).
-C tamanho	fixa número de cilindros do disco.

6 PROJETOS CRIADOS
POR LINUS TORVALDS

FDISK, MKFS, FSCK,
MKSWAP, SWAPON, SWAPOFF



git



A LEI DE LINUS

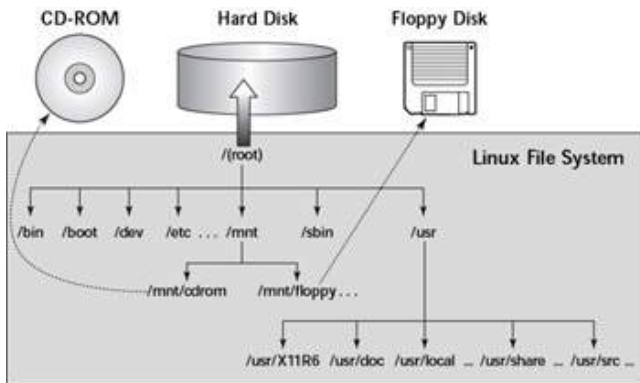


Comando mount

- ▶ 1º argumento defina o **sistema de arquivos** de /dev a ser montado.
- ▶ 2º argumento defina o diretório existente do **ponto de montagem**.

touch /mnt/file

mount /dev/sdb1 /mnt



Montagem por nome de dispositivo de blocos

- ▶ Para montar um filesystem, o **diretório de destino** já deve **existir**.
- ▶ **/mnt** ou **mount** são padrão mas você pode montar em outro lugar!

mount /dev/sdb1 /cdrom

Chave	Função
-w	monta a partição para leitura/gravação (é o padrão).
-r	monta a partição somente para leitura.
-a	monta todos os sistemas de arquivos do /etc/fstab



Desmontagem de sistemas de arquivos

- **Antes de remover** o dispositivo, deve **desmontá-lo manualmente**.

umount / **mnt**



Desmontagem de sistemas de arquivos

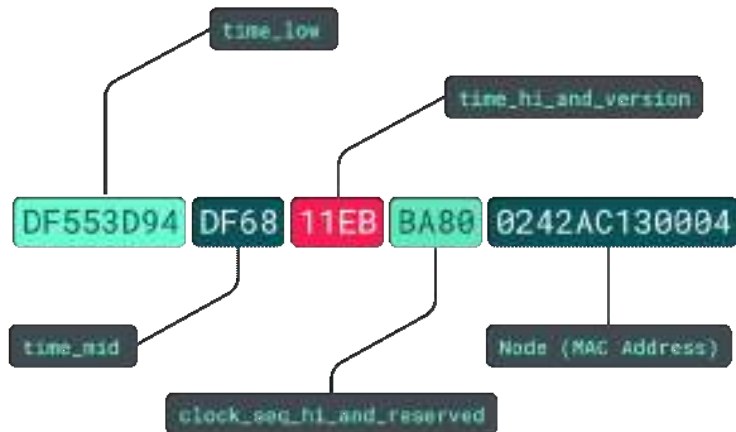
- ▶ **Alterações no FS** gravadas temporariamente em **cache de escrita***.
- ▶ **umount** garante a **sincronização** dos dados antes de liberar a unidade.



* write-back caching melhora o desempenho e a eficiência das gravações.

Montagem por identificador estável de FS

- **UUID**: **U**niversally **U**nique **ID**entifier é um número de 128 bits.
- **GUID**: **G**lobally **U**nique **ID**entifier utilizado sobretudo na Microsoft.



- ▶ **lsblk** pode listar o **caminho completo do dispositivo** com **UUIDs**.

Chave	Função
-f	exibe informação sobre os sistemas de arquivos.
-p	exibe o caminho completo do dispositivo.

```
$ lsblk -fp
```

NAME	FSTYPE	LABEL	UUID	FSAVAIL	FSUSE	MOUNTPOINT
/dev/sda						
-/dev/sda1	vfat		C6E0-9B61	505M	1%	/boot/efi
-/dev/sda2	ext4		7f669e0f ...	85G	58%	/
/dev/sdb	vfat	DISK	103B-E5B5	30,6M	99%	/media/tux/DISK

- ▶ Monte o **sistema de arquivos pela UUID** do sistema de arquivos.

```
# mount UUID="103B-E5B5" /mnt
```

Montagem durante o boot do sistema

- ▶ **fstab** file systems **table**: inicializa a montagem das partições
- ▶ **/etc/fstab** define como **partições dos discos** devem ser **montados**.



Gerenciadores de boot

LILO (**L**inux **L**Oader) e **G**RUB (**G**Rand **U**nified **B**ootloader)

1. Irá procurar por um **boot loader** no início do disco.
2. Se não existir, **procura** a **partição primária ativa** e monta na /.
3. Se não existir, verificará **outro dispositivo** na **sequência de boot**.
4. Se não existir, lançará a **mensagem na tela...**



Master Boot Record (MBR)

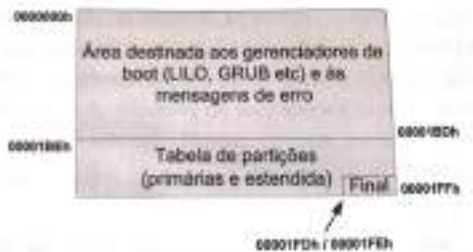
- ▶ O **M**aster **B**oot **R**ecord (**MBR**) é um **espaço** de **512 bytes**.
- ▶ Ele existe no **início do HD**, que acomoda **gerenciadores de boot**.



Master Boot Record (MBR)

MBR é dividido em **três partes** em uma **página de 512 bytes**:

1. A área dos **gerenciadores de boot**: 446 primeiros bytes.
2. A área de 64 bytes da **tabela de partições** (00001BEh a 00001FDh).
3. Última área com 2 bytes assinatura 55AAh (00001FEh e 00001FFh).



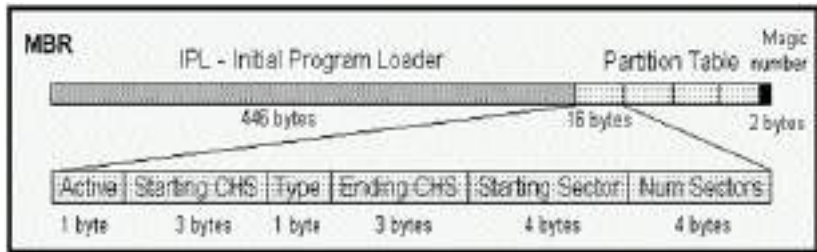
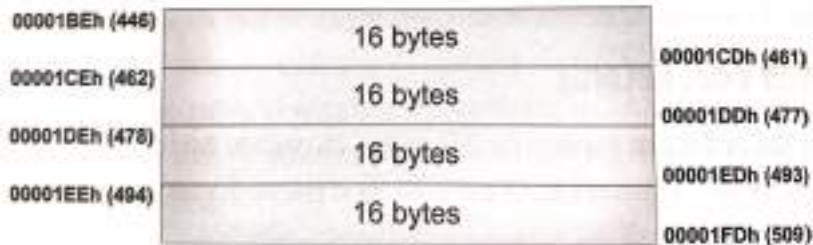
Master Boot Record (MBR)

- ▶ **Parte inicial do bootloader** localiza/**carrega 2º estágio** do bootloader.
- ▶ Resto incluindo **interface de seleção**, fora do MBR, **na partição ativa**.



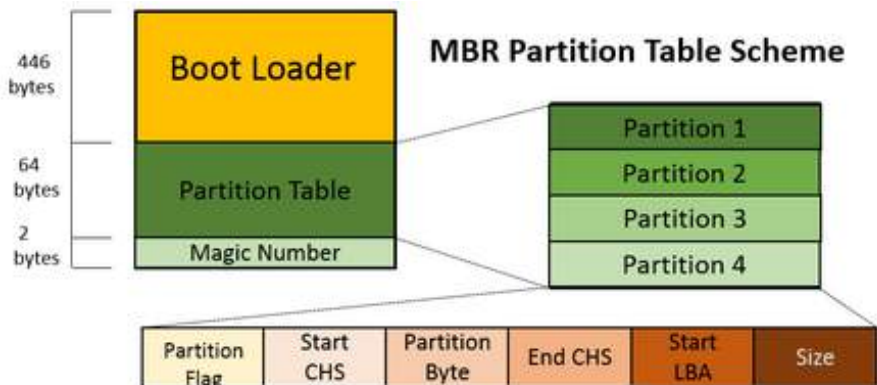
Master Boot Record (MBR)

- A **tabela de partições** (64 bytes) está dividida em **4 áreas de 16 bytes**.



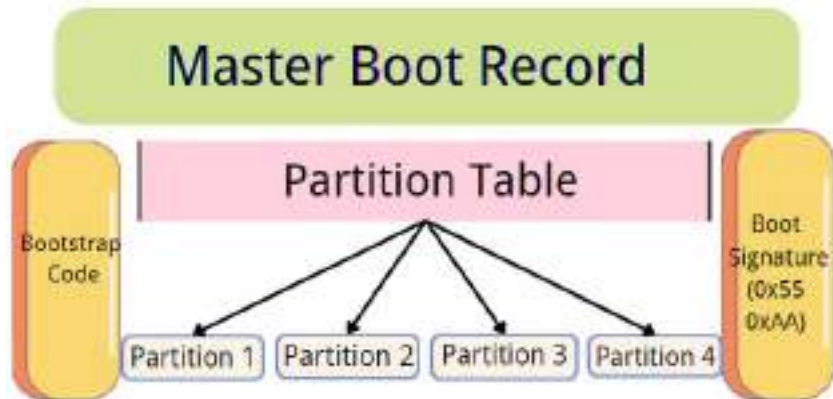
Master Boot Record (MBR)

- ▶ Endereços de início e fim de partição **tanto em LBA** quanto **em CHS**.
- ▶ Pode haver entradas CHS com **valores padrão para compatibilidade**.



Master Boot Record (MBR)

- ▶ Cada uma das **4 áreas** da tabela de partições é uma **partição primária**.
- ▶ Ocorre que **antigamente** os HDs **só** podiam ter **quatro partições**!



Master Boot Record (MBR)

- ▶ Com passar dos tempos os HDs foram **aumentando em tamanho**.
- ▶ Assim foi **concebida** a **ideia** de **partições estendidas**!

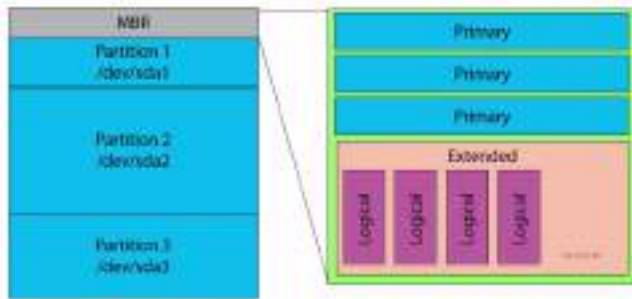


Master Boot Record (MBR)

- A seguir, foi criada uma **participação estendida** (com **valor 05h**).

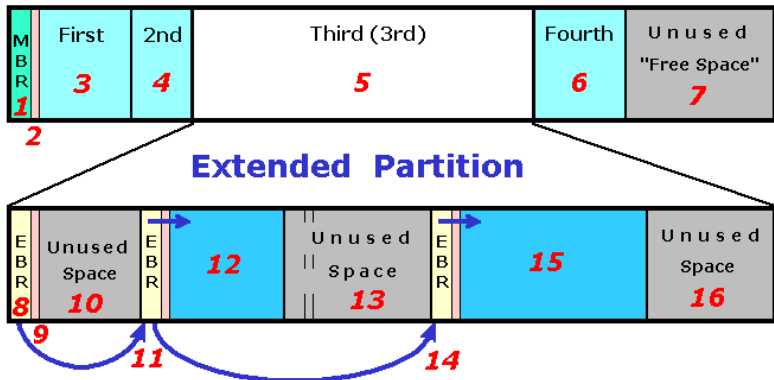
```
000001b0  00 00 00 00 00 00 00 00 00 00 00 00 00 01 | .....|
000001c0  01 00 83 fe 3f 49 3f 00 00 00 8b 23 12 00 00 00 | ...?I?...#...|
000001d0  01 4a 83 fe 3f 62 ca 23 12 00 d9 20 06 00 00 00 | .J..?b.#... ..|
000001e0  01 63 05 fe bf 09 a3 44 18 00 e7 b0 67 00 00 00 | .c....D...g...|
000001f0  00 00 00 00 00 00 00 00 00 00 00 00 00 55 aa | .....U.|
```

- Os dados especificando a **partição estendida** ficam guardados **no MBR**.



Extended Boot Record (EBR)

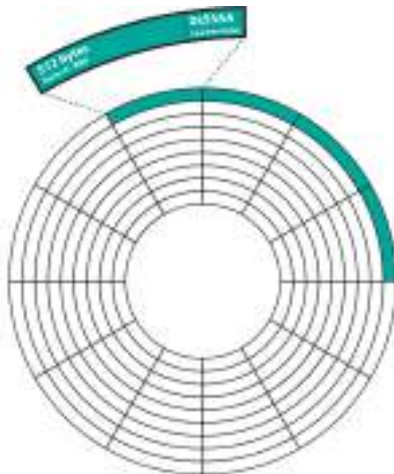
- **EBR** = **E**xtended **P**artition **T**able (**EPT**) + **assinatura**: $4 \times 16 + 2 = 66$ bytes
- Cada **EBR** apontará para a próxima **EBR** formando uma **corrente**.



Copyright(C)2007 by Daniel B. Sedory

Decadência do MBR

- ▶ Apenas **32 bits** estão disponíveis para representar setores lógicos.
- ▶ **Discos** compostos por **setores de 512 bytes** vão **até 2^{32} setores (2TB)**!



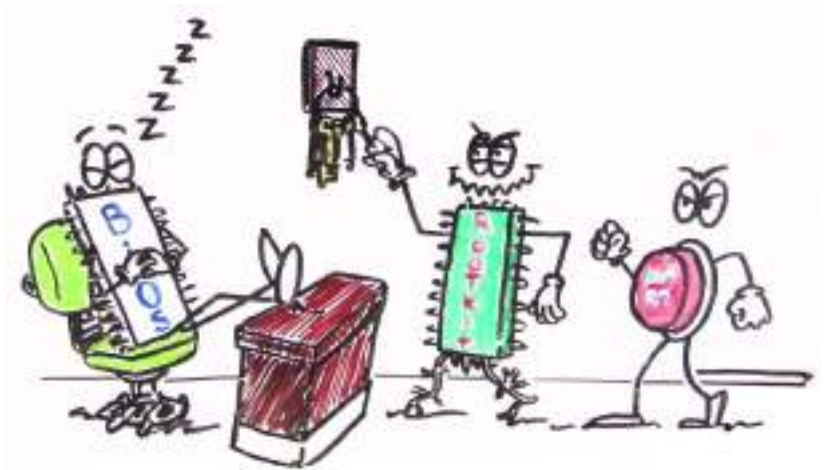
Decadência do MBR

- ▶ Você não pode usar **partições lógicas**, como **volumes de inicialização**.



Decadência do MBR

- Pode **substituir o código original** do MBR por um **código malicioso**.



 <https://www.youtube.com/watch?v=PI1atKjmDsc>

Decadência do MBR

- ▶ Secure boot com **assinaturas digitais** para **OS/driver** ser **aprovado**.
- ▶ Aprovado por **entidade confiável** (fabricante do hardware ou dev).



Padrão UEFI

- ▶ **90's: Intel substitui BIOS** da IBM por **Extensible Firmware Interface**.
- ▶ Com **AMD, Apple, Dell, HP, IBM e Lenovo: Unified EFI** em **2009**.



Partição EFI (/boot/efi)

- ▶ UEFI armazena na NVRAM como encontrar os bootloaders no disco
- ▶ Partição EFI (FAT32): bootmanager/loader, boot utilities (initramfs).



Partição EFI (/boot/efi)

- ▶ Cada OS tem sua **pasta** na partição para armazenar seu bootloader.
- ▶ UEFI carrega **bootloader** (.efi) responsável por **carregar o kernel** em RAM.

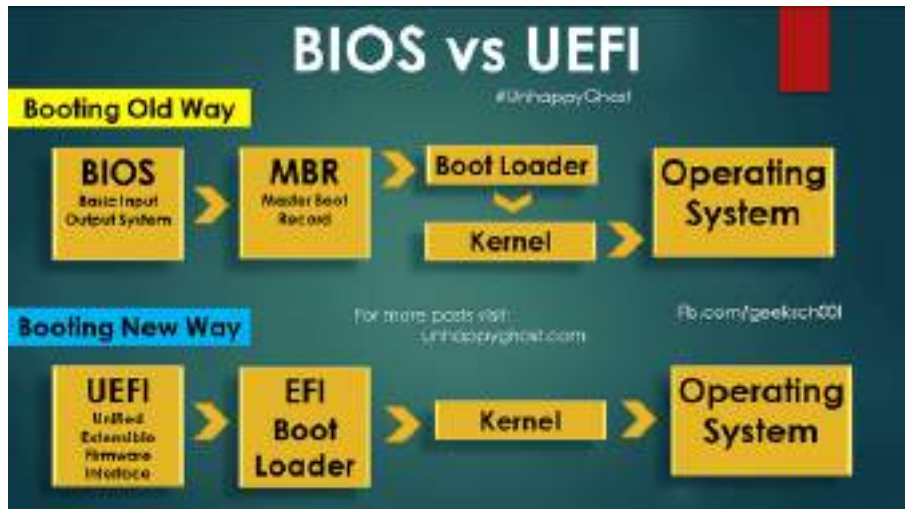


Kernel

- ▶ Bootloader **transfere o controle/info** pra kernel (UUID da partição raiz)
- ▶ **Kernel usa initramfs** do BL (FS temporário com mount, init scripts).



Diferença de boot entre BIOS e UEFI



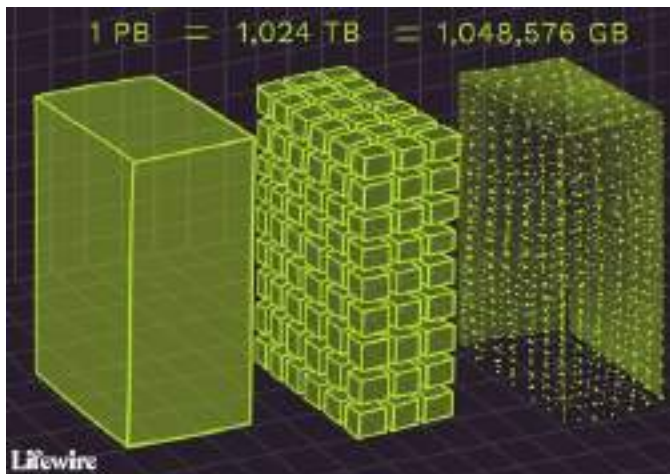
GPT (Globally Unique Identifier Partition Table)

- UEFI recebeu um **novo modelo** de **tabela de particionamento**: **GPT**.



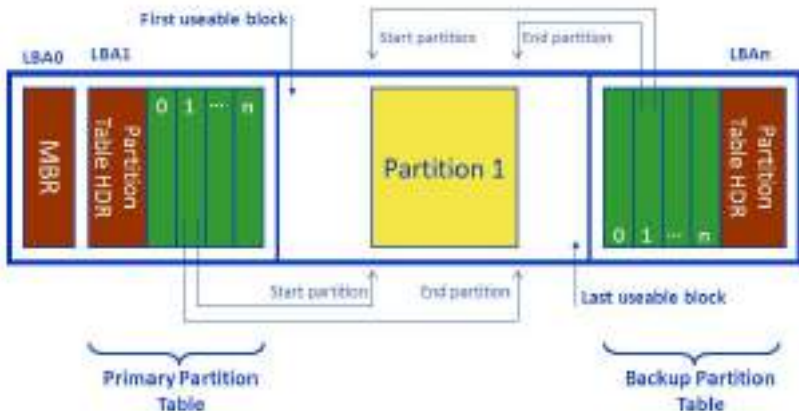
GPT (Globally Unique Identifier Partition Table)

- GPT permite **64 bits** com limitação de armazenamento de **até 9,4ZB!**



GPT (Globally Unique Identifier Partition Table)

- GPT está localizada no **início do dispositivo** preservando o MBR!



- Firmware UEFI suporta ambos os esquemas de particionamento.

