



# **Título do Curso:**

Administração de Sistemas  
GNU/Linux



# Conteúdo do curso

## ADMINISTRAÇÃO DE SISTEMAS GNU/LINUX

- ✓ Tópico 1: Introdução ao sistema operacional GNU/Linux.
- ✓ Tópico 2: Introdução ao Shell e comandos básicos.
- ✓ Tópico 3: Manipulação de conteúdos com comandos no Shell.
- ✓ Tópico 4: Comandos para gerenciamento do sistema e do Hardware.
- ✓ Tópico 5: Editor de Texto VI.
- ✓ Tópico 6: Administração de usuários e grupos.
- ✓ Tópico 7: Gerenciamento de permissões.
- ✓ Tópico 8: Gerenciamento de processos.
- **Tópico 9: Sistemas de arquivos e particionamento.**
- Tópico 10: Expressões regulares.
- Tópico 11: Introdução ao Shell Script.
- Tópico 12: Gerenciamento de Pacotes.
- Tópico 13: Agendamento de tarefas (cron) e Backup.





**DGP**

Tecnologia da Informação

## **Tópico 9**

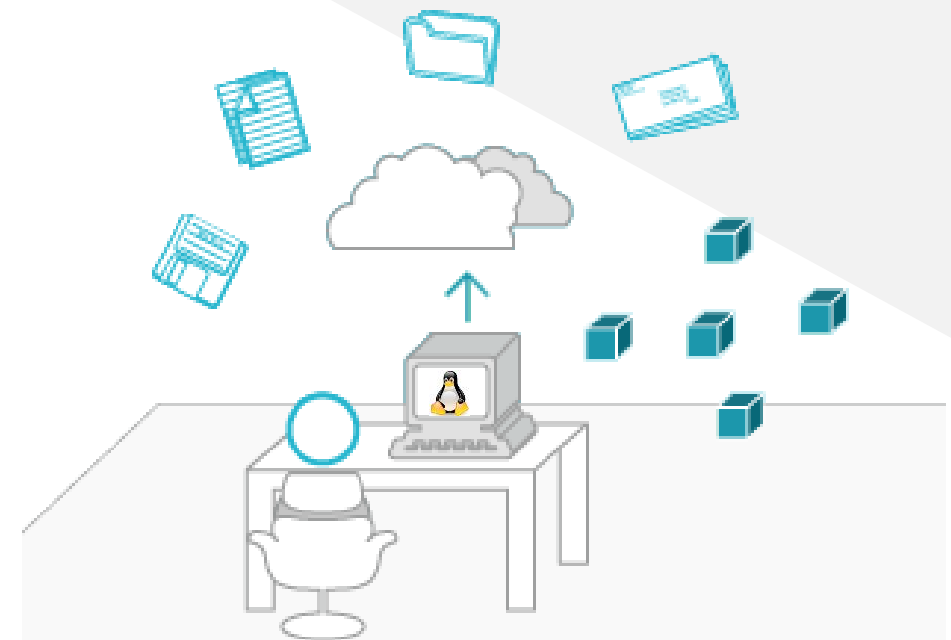
Sistemas de arquivos e  
particionamento.



# Sistemas de Arquivos

e Particionamento.

- Neste slide teremos comandos e conceitos sobre ponto de montagem, partições em disco e sistemas de arquivos:
  - Conceitos sobre sistemas de arquivos, dispositivos e partições;
  - Comandos para gerenciamento de disco e sistemas de arquivos.



# **Conceitos sobre sistemas de arquivos**

Tópico 9: Sistemas de  
arquivos e particionamento.



# Conceitos sobre sistemas de arquivos

- O funcionamento de softwares e sistemas operacionais dependem constantemente de uma estrutura para armazenar e obter dados.
- Todo processo pode armazenar em um espaço da memória (RAM) os dados utilizados no momento, porém, temos três problemas:
  - Apenas um processo pode acessar estes dados;
  - Ao término do processo estes dados são perdidos;
  - Nem sempre o espaço de endereço disponível para o processo possui tamanho suficiente para o volume de dados;

➤ (TANENBAUM, Andrew S. – Sistemas Operacionais Modernos – 3ª Edição).
- Estes problemas tornaram o uso de uma memória não volátil para o armazenamento de dados, algo extremamente necessário.



# Conceitos sobre sistemas de arquivos

- Atualmente temos diversos dispositivos de armazenamento (**pendrive**, **disco rígido**, **SSD**, **CD**, entre outros), porém, *como estes dados são armazenados e disponibilizados nestes diversos tipos de mídia?*
- Podemos exemplificar com uma situação real...:
  - Para possibilitar a leitura/gravação de dados em um novo HD, devemos particioná-lo, formatá-lo com um sistema de arquivos, e por fim, disponibilizá-lo para o sistema (ponto de montagem).
  - O sistema de arquivos é o responsável por gerenciar de que forma os dados serão nomeados, manipulados, protegidos, estruturados e disponibilizados.
- Em resumo, todo sistema de arquivos, apesar de possuírem características diferentes, possuem o mesmo propósito:
  - “Oferecer ao sistema operacional a estrutura necessária para ler/gravar os arquivos/diretórios.” (SILVA, Glaydson Mazioli – Guia Foca v2 – p. 65);



# Principais Sistemas de Arquivos

- O GNU/Linux apresenta uma versatilidade considerável neste quesito, oferecendo suporte a diversos tipos de sistemas de arquivos.
  - Através do “**fdisk**” podemos listar todos os sistemas de arquivos suportados.
- A seguir, temos uma lista dos principais tipos de sistemas de arquivos suportados pela plataforma Linux e todos suportados pela Microsoft:

## GNU/Linux

- ext2 / ext3 / ext4
- ReiserFS
- JFS
- XFS
- FAT... / NTFS
- ISO 9660
- Entre outros...

## Micro\$oft

- FAT16 / FAT32
- NTFS
- ReFS
- ISO 9660



# Acesso aos dispositivos de armazenamento



- Conforme citado no início da disciplina, o diretório “**/dev**” permite acessar diversos dispositivos, dentre eles, dispositivos de armazenamento como **HD**, **CD-ROM**, **disquete**, entre outros.
- No caso do **Windows**, os dispositivos são detectados e montados automaticamente (sendo disponibilizados em unidades representadas por letras, como **C**, **D**, **E**, e assim por diante).
- No GNU/Linux, os dispositivos são detectados, porém, não são montados automaticamente, sendo necessário intervir manualmente para realizar a montagem do dispositivo em um ponto de montagem (veremos como realizar este processo a seguir).



# Acesso aos dispositivos de armazenamento

- Para acessar um dispositivo de armazenamento através do Linux é importante sabermos os tipos de interface existentes e em qual local o dispositivo está conectado.
- Vamos exemplificar através das interfaces mais comuns utilizadas com um disco rígido (**nas literaturas LPI**), que são as interfaces **IDE** e **SCSI** (Também temos **SAS** e **SerialATA**, que serão mencionados a seguir).
- Em cada interface **IDE** podemos conectar dois dispositivos (um **Master** e um **Slave**). Já os dispositivos **SCSI**, **SerialATA** e **SAS**, possuem interfaces independentes (podendo chegar ao limite de **15** dispositivos).
- Cada dispositivo é “endereçoado” no Linux conforme tabela a seguir:



# Acesso aos dispositivos de armazenamento

- Caminho de acesso aos dispositivos de armazenamento no GNU/Linux:

<b>Caminho</b>	<b>Dispositivo de armazenamento conectado em...</b>
/dev/hda	IDE Primária "Master"
/dev/hdb	IDE Primária "Slave"
/dev/hdc	IDE Secundária "Master"
/dev/hdd	IDE Secundária "Slave"
/dev/sda	Primeira interface SCSI / SerialATA / SAS
/dev/sdb	Segunda interface SCSI / SerialATA / SAS
/dev/sdc	Terceira interface SCSI / SerialATA / SAS
/dev/fd0	Disquete conectado na primeira interface
/dev/md0	Volume lógico criado através do RAID via Software
/dev/sr0	Primeiro dispositivo de CD-ROM SCSI / Serial ATA
/dev/sr1	Segundo dispositivo de CD-ROM SCSI / Serial ATA



# Partições em disco – MBR x GPT

- Existem duas formas de armazenar as informações de particionamento em um disco: **MBR** (Master Boot Record) e **GPT** (GUID Partition Table).
- O **GPT** vem substituindo o **MBR** ao longo dos últimos anos, em especial, devido a limitação do tamanho máximo de **2 TB** suportado em uma partição **MBR**. Enquanto que o limite de uma partição **GPT** é de **9,4 ZB** (Zeta Bytes).
- Além disso, ao utilizar o **MBR**, há uma limitação de **4** partições primárias (podendo chegar a **15** com partições lógicas). Enquanto que no **GPT**, o número é ilimitado (concepção da tecnologia), porém, Windows e Linux limitam em **128** partições.
  - Fonte: <https://www.howtogeek.com/193669/whats-the-difference-between-gpt-and-mbr-when-partitioning-a-drive/>
  - Fonte: <https://www.meupositivo.com.br/doseujeito/dicas/diferenca-entre-mbr-e-gpt/>



# Partições em disco

- Podemos dividir o disco em várias partes de forma lógica, possibilitando a instalação de mais de um sistema no mesmo disco ou apenas separar uma parte para a instalação do sistema operacional e outra para dados.
- Ao utilizar o **MBR**, temos 4 tipos de partições (no **GPT** seriam apenas “duas”):
  - **Partição Primária**: todo disco deve conter pelo menos **uma**, com sistema de arquivos, marcada como ativa para inicialização do sistema operacional. É possível definir no máximo **4** partições primárias.
  - **Partição Estendida**: esse tipo de partição é um contêiner de partições lógicas; não suporta sistemas de arquivos e um disco pode possuir apenas **uma**.
  - **Partições Lógicas**: são subpartições de uma partição estendida. Podem existir até **12** partições lógicas em um disco (identificadas do número **5** ao **16**).
  - **Partições de SWAP**: tem o objetivo de aumentar a performance do sistema, possibilitando que o Linux tenha uma memória virtual em disco. A partição SWAP trabalha como um arquivo de troca de dados entre a RAM e o disco.



# Partições em disco – MBR

- Cada disco pode possuir até **16** partições, com as seguintes ressalvas:
  - Total de **4** partições **primárias**, com suporte a sistemas de arquivos, impossibilitando a criação de outras partições;
  - Total de **15** partições com suporte a sistemas de arquivos, sendo:
    - **3** partições **primárias** com sistemas de arquivos;
    - **1** partição **estendida**, sem sistema de arquivos (**estendida** = contêiner de partições **lógicas**);
    - **12** partições **lógicas**, com sistemas de arquivos, contidas na partição estendida.
- Esta regra se aplica em qualquer plataforma (**Micro\$oft**, **Linux**, **MAC**, etc...), pois estamos tratando de uma propriedade do Hardware.



# Partições em disco

- Para possibilitar o acesso as partições através do sistema, cada uma delas é representada por um número inteiro, conforme tabela a seguir:

Caminho	Partição / Dispositivo
<code>/dev/hda1</code>	1ª partição primária do disco da IDE Primária "Master"
<code>/dev/hda2</code>	2ª partição primária do disco da IDE Primária "Master"
<code>/dev/hdb4</code>	4ª partição primária do disco da IDE Primária "Slave"
<code>/dev/hdb5</code>	1ª partição lógica do disco da IDE Primária "Slave"
<code>/dev/hdd5</code>	1ª partição lógica do disco da IDE Secundária "Slave"
<code>/dev/sda1</code>	1ª partição primária do disco da Primeira interface SCSI/SATA/SAS
<code>/dev/sda3</code>	3ª partição primária do disco da Primeira interface SCSI/SATA/SAS
<code>/dev/sdc16</code>	12ª partição lógica do disco da Terceira interface SCSI/SATA/SAS



# Partições em disco

- Talvez será possível assimilar melhor através das cores...

Caminho	Partição / Dispositivo
/dev/hda1	1ª partição primária do disco da IDE Primária "Master"
/dev/hda2	2ª partição primária do disco da IDE Primária "Master"
/dev/hdb4	4ª partição primária do disco da IDE Primária "Slave"
/dev/hdb5	1ª partição lógica do disco da IDE Primária "Slave"
/dev/hdd5	1ª partição lógica do disco da IDE Secundária "Slave"
/dev/sda1	1ª partição primária do disco da Primeira interface SCSI/SATA/SAS
/dev/sda3	3ª partição primária do disco da Primeira interface SCSI/SATA/SAS
/dev/sdc16	12ª partição lógica do disco da Terceira interface SCSI/SATA/SAS



# **Comandos para gerenciamento de disco e sistemas de arquivos**

Tópico 9: Sistemas de  
arquivos e particionamento.



# Partições em disco – MBR x GPT

- Para realizar o particionamento temos diversas ferramentas como o “**fdisk**”, “**cfdisk**”, “**gdisk**”, “**parted**”, “**gparted**”, “**partimage**”, (“**diskpart**” → **Micro\$oft**);<sup>\*</sup><sup>\*</sup>
- É importante ressaltarmos que temos dois tipos de particionamento:
  - Destrutivo: Exclui todos os dados existentes ao remover e criar novas partições (mais comum e mais utilizado);
    - Ferramentas: “**fdisk**”, “**cfdisk**”, “**gdisk**”, “**diskpart**”, “**Gerenciador de disco**”, entre outros;
  - Não destrutivo (redimensionador): Modifica o tamanho de partições já existentes (mais complexo e de maior risco);
    - Ferramentas: “**parted**”, “**gparted**”, “**partimage**”, “**Partition Magic**”, “**Gerenciador de disco**”, entre outros;
  - OBS.: Neste momento o LVM (Linux Volume Manager) não será abordado, pois não se trata de particionamento.



# Partições em disco

- Agora que já sabemos identificar o dispositivo, podemos realizar o particionamento através do “**fdisk**”, “**cfdisk**” ou “**gdisk**”:
  - O comando “**fdisk -l**” lista os dispositivos de armazenamento conectados:

```
[root@server ~]# fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks    Id  System
/dev/sda1   *           1          13       104391    83  Linux
/dev/sda2             14        1044     8281507+   8e  Linux LVM

Disk /dev/sdb: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

→ Disk /dev/sdb doesn't contain a valid partition table
```

- Para iniciar o particionamento, devemos especificar o caminho do dispositivo, conforme abaixo:

```
[root@server ~]# fdisk /dev/sdb
```

```
[root@server ~]# cfdisk /dev/sdb
```

# Definição do tipo de partição



- O sistema de arquivos a ser utilizado em uma nova partição é definido no momento da formatação, através do comando utilizado, porém, durante o particionamento, devemos “preparar” a partição, selecionando o tipo de sistema de arquivos que será utilizado:

0	Empty	1e	Hidden W95 FAT1	80	Old Minix	bf	Solaris
1	FAT12	24	NEC DOS	81	Minix / old Lin	c1	DRDOS/sec (FAT-
2	XENIX root	39	Plan 9	82	Linux swap / So	c4	DRDOS/sec (FAT-
3	XENIX usr	3c	PartitionMagic	83	Linux	c6	DRDOS/sec (FAT-
4	FAT16 <32M	40	Uenix 80286	84	OS/2 hidden C:	c7	Syrinx
5	Extended	41	PPC PReP Boot	85	Linux extended	da	Non-FS data
6	FAT16	42	SFS	86	NTFS volume set	db	CP/M / CTOS / .
7	HPFS/NTFS	4d	QNX4.x	87	NTFS volume set	de	Dell Utility
8	AIX	4e	QNX4.x 2nd part	88	Linux plaintext	df	BootIt
9	AIX bootable	4f	QNX4.x 3rd part	8e	Linux LVM	e1	DOS access
a	OS/2 Boot Manag	50	OnTrack DM	93	Amoeba	e3	DOS R/O
b	W95 FAT32	51	OnTrack DM6 Aux	94	Amoeba BBT	e4	SpeedStor
c	W95 FAT32 (LBA)	52	CP/M	9f	BSD/OS	eb	BeOS fs
e	W95 FAT16 (LBA)	53	OnTrack DM6 Aux	a0	IBM Thinkpad hi	ee	EFI GPT
f	W95 Ext'd (LBA)	54	OnTrackDM6	a5	FreeBSD	ef	EFI (FAT-12/16/
10	OPUS	55	EZ-Drive	a6	OpenBSD	f0	Linux/PA-RISC b
11	Hidden FAT12	56	Golden Bow	a7	NeXTSTEP	f1	SpeedStor
12	Compaq diagnost	5c	Priam Edisk	a8	Darwin UFS	f4	SpeedStor
14	Hidden FAT16 <3	61	SpeedStor	a9	NetBSD	f2	DOS secondary
16	Hidden FAT16	63	GNU HURD or Sys	ab	Darwin boot	fb	VMware VMFS
17	Hidden HPFS/NTF	64	Novell Netware	b7	BSDI fs	fc	VMware VMKCORE
18	AST SmartSleep	65	Novell Netware	b8	BSDI swap	fd	Linux raid auto
1b	Hidden W95 FAT3	70	DiskSecure Mult	bb	Boot Wizard hid	fe	LANstep
1c	Hidden W95 FAT3	75	PC/IX	be	Solaris boot	ff	BBT

OBS.: Listagem do “fdisk” (MBR).

82 = Swap

83 = Linux

# Definição do tipo de partição



- Listagem dos tipos de partições suportados pelo “gdisk” (modo GPT).

```
Command (? for help): l
0700 Microsoft basic data 0c01 Microsoft reserved 2700 Windows RE
3000 ONIE boot 3001 ONIE config 3900 Plan 9
4100 PowerPC PReP boot 4200 Windows LDM data 4201 Windows LDM metadata
4202 Windows Storage Spac 7501 IBM GPPS 7f00 ChromeOS kernel
7f01 ChromeOS root 7f02 ChromeOS reserved 8200 Linux swap
8300 Linux filesystem 8301 Linux reserved 8302 Linux /home
8303 Linux x86 root (/) 8304 Linux x86-64 root (/) 8305 Linux ARM64 root (/)
8306 Linux /srv 8307 Linux ARM32 root (/) 8400 Intel Rapid Start
8e00 Linux LVM a000 Android bootloader a001 Android bootloader 2
a002 Android boot a003 Android recovery a004 Android misc
a005 Android metadata a006 Android system a007 Android cache
a008 Android data a009 Android persistent a00a Android factory
a00b Android fastboot/ter a00c Android OEM a500 FreeBSD disklabel
a501 FreeBSD boot a502 FreeBSD swap a503 FreeBSD UFS
a504 FreeBSD ZFS a505 FreeBSD Uinum/RAID a580 Midnight BSD data
a581 Midnight BSD boot a582 Midnight BSD swap a583 Midnight BSD UFS
a584 Midnight BSD ZFS a585 Midnight BSD Uinum a600 OpenBSD disklabel
a800 Apple UFS a901 NetBSD swap a902 NetBSD FFS
a903 NetBSD LFS a904 NetBSD concatenated a905 NetBSD encrypted
a906 NetBSD RAID ab00 Recovery HD af00 Apple HFS/HFS+
af01 Apple RAID af02 Apple RAID offline af03 Apple label
Press the <Enter> key to see more codes:
af04 AppleTV recovery af05 Apple Core Storage af06 Apple SoftRAID Statu
af07 Apple SoftRAID Scrat af08 Apple SoftRAID Volum af09 Apple SoftRAID Cache
b300 QNX6 Power-Safe bc00 Acronis Secure Zone be00 Solaris boot
bf00 Solaris root bf01 Solaris /usr & Mac Z bf02 Solaris swap
bf03 Solaris backup bf04 Solaris /var bf05 Solaris /home
bf06 Solaris alternate se bf07 Solaris Reserved 1 bf08 Solaris Reserved 2
bf09 Solaris Reserved 3 bf0a Solaris Reserved 4 bf0b Solaris Reserved 5
c001 HP-UX data c002 HP-UX service e100 ONIE boot
e101 ONIE config ea00 Freedesktop $BOOT eb00 Haiku BFS
ed00 Sony system partitio ed01 Lenovo system partit ef00 EFI System
ef01 MBR partition scheme ef02 BIOS boot partition f800 Ceph OSD
f801 Ceph dm-crypt OSD f802 Ceph journal f803 Ceph dm-crypt journa
f804 Ceph disk in creatio f805 Ceph dm-crypt disk i fb00 VMware VMFS
fb01 VMware reserved fc00 VMware kcore crash p fd00 Linux RAID
```

OBS.: Listagem do “gdisk” (GPT).

8200 = Swap

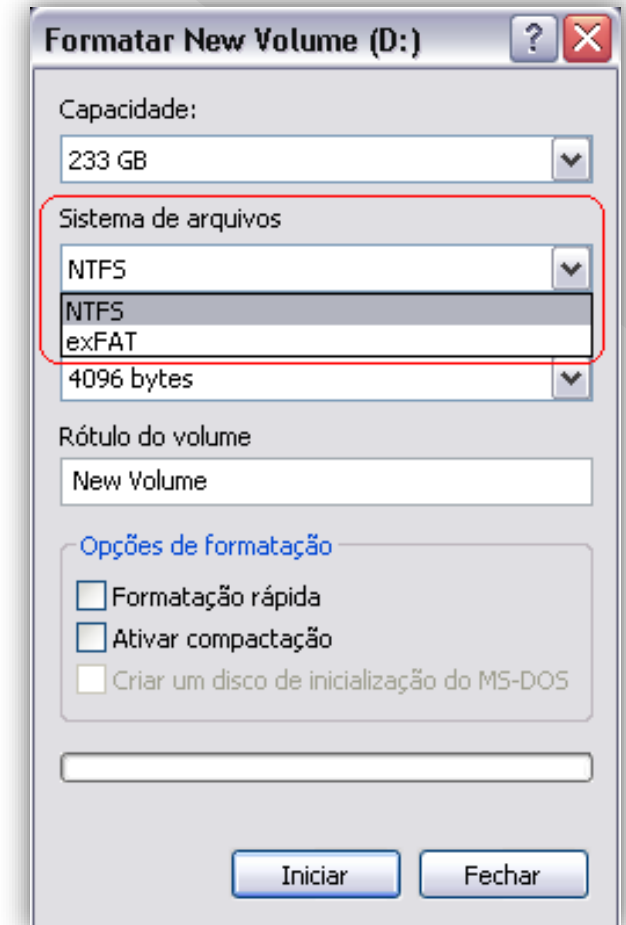
8300 = Linux

# Formatação e definição do Sistema de arquivos



- Após definir o tipo de partição, gravamos as novas informações de partição no disco e realizamos o processo de formatação.
- No Windows, selecionamos qual o sistema de arquivos desejado entre o **NTFS** e o **FAT**.
- No Linux, cada distribuição pode oferecer suporte a sistemas de arquivos distintos.
  - Dentre os principais, temos:
    - ReiserFS;
    - ext2 / ext3 / ext4;
    - XFS;
    - JFS;

```
root@ninja:~# mkfs
mkfs          mkfs.cramfs    mkfs.ext3      mkfs.minix     mkfs.xfs
mkfs.bfs      mkfs.ext2      mkfs.jfs       mkfs.reiserfs
```





# Formatação e definição do Sistema de arquivos

- Podemos realizar a formatação e definição do sistema de arquivos de duas formas:
  - Comando “`mkfs -t ext4 /dev/sda2`”
  - Comando “`mkfs.ext4 /dev/sda2`”
    - OBS.: Ambos os comandos acima estão formatando a segunda partição do disco SCSI/SATA conectado na primeira interface SCSI/SATA, além de definir o “ext4” como sistema de arquivos.
  - Comando “`mkfs -t ntfs /dev/sdb1`”
  - Comando “`mkfs.ntfs /dev/sdb1`”
    - OBS.: Ambos os comandos acima estão formatando a primeira partição do disco SCSI/SATA conectado na segunda interface SCSI/SATA, além de definir o “ntfs” como sistema de arquivos.

# **Conceitos sobre sistemas de arquivos Parte 2**

Tópico 9: Sistemas de  
arquivos e particionamento.





# Conceito de “Ponto de Montagem”

- Na plataforma Microsoft, uma nova partição pode ser disponibilizada e acessada através de “**unidades**” como “**C:**”, “**D:**”, e assim por diante;
- Esta “**unidade**” representada por uma letra nada mais é do que o “**ponto de montagem**” que possibilita o acesso a partição do dispositivo.
- No GNU/Linux não temos “**unidades**” disponíveis para serem utilizadas como “**ponto de montagem**”, portanto, como acessar os dispositivos?
- Simples, qualquer local (**diretório**) do sistema pode ser utilizado como um “**ponto de montagem**”, proporcionando alta escalabilidade.
  - Podemos ter em um HD o sistema operacional, diretório raiz e subdiretórios.
  - Outro HD (ou partição) apenas para os arquivos dos usuários → “/home”.
  - Outro HD (ou partição) apenas para os Logs de sistema, caso seja um servidor WEB, ou Proxy, que gera muitos Logs → “/var”.
  - Outro HD (ou partição) apenas para o **banco de dados** do **ERP** da empresa.



# Conceito de “Ponto de Montagem”

- Desta forma, podemos separar a partição do sistema de outras partições (ou disco) que contém dados em constante crescimento, evitando instabilidades, queda de desempenho (pela grande demanda de I/O ao mesmo disco), ou até mesmo o travamento do sistema por falta de espaço em disco (ou melhor, na partição raiz).
- Portanto, um “**ponto de montagem**”, nada mais é do que: “Diretório de onde a unidade de disco/partição será acessado. O diretório deve estar vazio para montagem de um sistema de arquivo. Normalmente é usado o diretório “/mnt” para armazenamento de pontos de montagem temporários.” → (SILVA, Glaydson Mazioli – Guia Foca v2 – p. 86);
- OBS.: Este processo também pode ser feito no **Windows**... *Como assim?*
  - ***Vamos ver na prática!?***

# **Comandos para gerenciamento de disco e sistemas de arquivos Parte 2**

Tópico 9: Sistemas de  
arquivos e particionamento.



# Comandos “mount” e “umount”

- mount → Monta um sistema de arquivos, tornando-o disponível para operações de I/O (entrada e saída / leitura e escrita).
  - Ex.: `mount -t [file_system] [/caminho/dispositivo] [ponto/montagem]`
  - OBS.: O comando acima montou um sistema de arquivos “ext3”;
  - OBS.2: Sempre deve ser utilizado um diretório vazio para montagem;
- umount → Desmonta um sistema de arquivos.
  - Ex.: `umount <opções> [“dispositivo” ou “caminho_ponto_de_montagem”]`
  - OBS.: É importante executar o “`umount`” para remover um dispositivo como **pendrive** ou **CD-ROM**, para evitar que os dados fiquem corrompidos (no caso do **pendrive**) ou para possibilitar a abertura do drive (no caso do **CD-ROM**).



# Comandos “mount” e “umount”

- Para montar uma mídia de CD-ROM ou DVD-ROM, devemos utilizar o sistema de arquivos “**iso9660**”;

```
[root@server ~]# mount -t iso9660 /dev/hdc /mnt/cdrom/  
mount: block device /dev/hdc is write-protected, mounting read-only
```

- Podemos visualizar os dispositivos montados e local do ponto de montagem através do comando “**mount -l**”:

```
[root@server ~]# mount -l  
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)  
proc on /proc type proc (rw)  
sysfs on /sys type sysfs (rw)  
devpts on /dev/pts type devpts (rw,gid=5,mode=620)  
/dev/sda1 on /boot type ext3 (rw) [/boot]  
tmpfs on /dev/shm type tmpfs (rw)  
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)  
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)  
/dev/sdb1 on /mnt/hd01 type ext3 (rw)
```



# Automount – Arquivo “/etc/fstab”

- O arquivo “/etc/fstab” é utilizado pelos sistemas Linux para determinar quais partições serão montadas automaticamente na inicialização e qual o ponto de montagem.
- Segue abaixo um print do conteúdo do arquivo “/etc/fstab”:

```
/dev/hda1      swap          swap          defaults      0      0
/dev/hda2      /             reiserfs      defaults      1      1
#/dev/cdrom    /mnt/cdrom    auto          noauto,owner,ro 0      0
/dev/fd0       /mnt/floppy   auto          noauto,owner   0      0
devpts        /dev/pts      devpts        gid=5,mode=620 0      0
proc          /proc         proc          defaults      0      0
```



# Automount – Arquivo “/etc/fstab”

- Cada uma das colunas do arquivo “/etc/fstab” possuem os parâmetros sobre as partições que são lidos pelo comando “mount”, sendo:
  - 1ª coluna = **Dispositivo** → Partição a ser montada. Ex: “/dev/sda2”
  - 2ª Coluna = **Ponto de Montagem** → Especifica o diretório em que a partição será montada. Ex: “/mnt/hd02”.
  - 3ª Coluna = **Tipo** → Determina o tipo de sistema de arquivos que será usado na partição a ser montada. (ext2, ext3, reiserfs, msdos, vfat, iso9660, nfs, ntfs, swap, proc, entre outras).
  - 4ª Coluna = **Opções** → Especifica as opções usadas com o sistema de arquivos (veremos a lista de opções a seguir).

```
/dev/hda1      swap      swap      defaults      0      0
/dev/hda2      /         reiserfs   defaults      1      1
#/dev/cdrom    /mnt/cdrom auto       noauto,owner,ro 0      0
/dev/fd0       /mnt/floppy auto       noauto,owner    0      0
devpts         /dev/pts  devpts     gid=5,mode=620  0      0
proc           /proc     proc       defaults      0      0
```



# Automount – Arquivo “/etc/fstab”

- Continuação, arquivo “/etc/fstab”...:
  - 5ª coluna = **Frequência de Backup** → O comando “**dump**” consulta o arquivo “/etc/fstab” para saber quais sistemas de arquivos devem ser copiados. Se o valor for **1** ele faz o backup, se o valor for **0** ele assumirá que o sistema de arquivos não precisa ser copiado.
  - 6ª coluna = **Checagem de disco** → Determina se o dispositivo deve ou não ser checado na inicialização do sistema pelo “**fsck**”. Se o valor for **0**, o sistema de arquivos não será checado, o número **1** deve ser usado para checar a partição raiz ( / ), a partir do **2**, outros sistemas de arquivos montados em subdiretórios.

/dev/hda1	swap	swap	defaults	0	0
/dev/hda2	/	reiserfs	defaults	1	1
#/dev/cdrom	/mnt/cdrom	auto	noauto,owner,ro	0	0
/dev/fd0	/mnt/floppy	auto	noauto,owner	0	0
devpts	/dev/pts	devpts	gid=5,mode=620	0	0
proc	/proc	proc	defaults	0	0





# Automount – Arquivo “/etc/fstab”

- Dentre as opções do sistema de arquivos da quarta coluna, temos:
  - **auto** → Dispositivo deve ser montado na inicialização do sistema.
  - **noauto** → Dispositivo não deve ser montado na inicialização do sistema.
  - **ro** → Montar sistema de arquivos com permissões de somente de leitura.
  - **rw** → Montar sistema de arquivos com permissões de leitura e gravação.
  - **exec** → Montar sistema de arquivos com permissão de execução de arquivos.
  - **noexec** → Montar sistema de arquivos sem permissão de execução.
  - **user** → Permite que qualquer usuário monte o dispositivo, mas proíbe outros de desmontá-lo.
  - **users** → Permite que qualquer usuário monte e desmonte os sistemas de arquivos.
  - **nouser** → Somente o superusuário pode montar e desmontar.
  - **owner** → Permite que o proprietário do dispositivo realize a montagem.



# Automount – Arquivo “/etc/fstab”

- Continuação... Opções da quarta coluna:
  - **sync** → Habilita a transferência de dados síncrona no dispositivo.
  - **async** → Habilita a transferência de dados assíncrona no dispositivo.
  - **dev** → Dispositivo especial de caracteres.
  - **suid** → Habilita que os executáveis tenham permissões **SUID** e **SGID**.
  - **nosuid** → Especifica que os executáveis não terão permissões **SUID** e **SGID**.
  - **defaults** → Especifica as opções de montagem padrão, como **rw**, **suid**, **exec**, **auto**, **nouser** e **async**.
- Após este *overview* já podemos analisar o conteúdo do arquivo:

```
/dev/hda1      swap      swap      defaults      0      0
/dev/hda2      /          reiserfs   defaults      1      1
#/dev/cdrom    /mnt/cdrom auto       noauto,owner,ro 0      0
/dev/fd0       /mnt/floppy auto       noauto,owner    0      0
devpts         /dev/pts   devpts     gid=5,mode=620  0      0
proc           /proc      proc       defaults      0      0
```

# Área de troca

## Partição de SWAP (Memória Virtual)



- Este tipo de partição (SWAP) é utilizado para prover memória virtual ao sistema GNU/Linux;
- Durante a execução de Softwares, cada processo carrega os dados necessários para sua execução na memória RAM, porém, ao esgotar o espaço de memória física (RAM), o sistema pode utilizar parte do disco (memória virtual – SWAP) para otimizar o funcionamento do sistema e permitir a execução de novos processos;
- Como a troca de dados entre a RAM e a memória virtual é constante (de acordo com a necessidade de uso dos aplicativos), esta área também é conhecida como “área de troca”;



# Área de troca

## Partição de SWAP (Memória Virtual)

- Após criar a partição, devemos defini-la como partição de SWAP:
  - Código 82 (MBR) → Linux SWAP / Solaris;
  - Código 8200 (GPT) → Linux SWAP;
- mkswap → Formata uma partição para ser utilizada como SWAP, preparando o dispositivo para ser usado como área de memória virtual.
  - Ex.: `mkswap [caminho/dispositivo]`
- swapon → Ativa a partição SWAP.
  - Ex.: `swapon [caminho/dispositivo]`

```
[root@server ~]# mkswap /dev/sdb5  
Setting up swspace version 1, size = 551055 kB
```

```
[root@server ~]# swapon /dev/sdb5
```



# Journaling

- O recurso de “*Journaling*” realiza gravações de diversas alterações em uma área da própria partição do disco (chamada de “*journal*”), com o objetivo de prover alta disponibilidade e maior tolerância a falhas;
- Caso ocorra um problema como falha de energia, todo os dados gerados pelo “*journaling*” são analisados durante a montagem do sistema de arquivos;
- Após a verificação, os dados (que não foram gravados devido o processo não ter sido finalizado corretamente) podem ser desfeitos ou finalizados (recuperados);
- Os sistemas de arquivos para Linux em sua grande maioria possuem o recurso de “*Journaling*”, no Windows, temos o recurso no NTFS.



# No próximo slide...

- Tópico 10: Expressões regulares:
  - Conceitos e aplicabilidade;
  - Metacaracteres e suas funções.



# Referências

- BONAN, Adilson Rodrigues. **LINUX – Fundamentos, Prática & Certificação LPI**. Editora: Alta Books. RJ. 2010;
- PEREIRA, Guilherme Rodrigues. **Slides para aula expositiva**. Centro Universitário UNA.
- SILVA, Gleydson Mazioli. **Guia Foca GNU/Linux**. Disponível em: <https://guiafoca.org/>



**DGP**

Tecnologia da Informação

**Obrigado!**



Guilherme Rodrigues