
Revisão do Tópico 203 – Sistemas de Arquivos e Dispositivos

203.1 – Operando o Sistema de Arquivos Linux

Configurando o /etc/fstab

Cada entrada no arquivo /etc/fstab é composta por 6 campos:

1. Dispositivo (device): Pode ser identificado pelo nome do device, como /dev/sdb1, pelo UUID ou pelo LABEL.
2. Ponto de Montagem (mountpoint): Diretório que será associado ao dispositivo. No caso de swap pode ser identificado como “none” ou “swap”
3. Tipo de FileSystem: Por exemplo ext4, vfat, ou auto para identificação automática
4. Opções: Opções de montagem
5. Dump: Determina se será feito um dump na partição. Em desuso.
6. Checagem do Disco: Determina se a partição será verificada pelo fsck no boot do sistema. 0 para não, 1 para a partição raiz, 2 para as demais partições.

O comando mount

Principais formas de uso:

- mount : lista as partições montadas no momento. A mesma informação pode ser obtida pelos arquivos:
  - /etc/mtab
  - /proc/mounts
- mount -t <tipo-filesystem> : lista apenas as partições do filesystem especificado
- mount <device> <mountpoint> : monta um device em um diretório
- mount -t <tipo-filesystem> <device> <mountpoint> : monta um device em um diretório usando um tipo específico
- mount <device> : faz a montagem de acordo com as configurações do /etc/fstab
- mount <mountpoint> : faz a montagem de acordo com as configurações do /etc/fstab
- mount -a : monta todos os dispositivos configurados com a opção “auto” no /etc/fstab
- mount -U <UUID> / mount UUID=<UUID> : monta um dispositivo pelo seu UUID
- mount -o opções : faz a montagem usando um grupo específico de opções

As **principais opções do mount,** utilizadas também no /etc/fstab:

- rw : Habilitar leitura e escrita na partição
- ro : Habilitar apenas leitura na partição
- auto : Montagem durante o boot e pela opção -a do comando mount
- noauto : Não monta automaticamente
- user : Permite que qualquer usuário possa montar a partição, mas apenas quem montou pode desmontar
- users: Permite que qualquer usuário possa montar e desmontar a partição.
- async / sync : Define se a gravação em disco será feita de forma síncrona ou assíncrona

- owner : Permite que o usuário dono do dispositivo possa montá-lo
- remount : Usado para remontar uma partição já montada, mas com outras opções
- uid / gid : Define o usuário ou grupo que será dono do diretório montado. Disponível apenas em alguns tipos de filesystems como fat e ntfs
- defaults: Habilita as seguintes opções: rw, suid, dev, exec, auto, nouser e async

---

Outros Comandos Úteis

- **blkid** : Mostra informações de todas as partições do sistema, incluindo UUID, Label e Tipo do Filesystem
- lsblk -f : Mostra informações dos discos e partições, incluindo tamanho, UUID, Label, Tipo do Filesystem e Ponto de Montagem
- e2label <device> : Mostra o Label de uma partição
- findfs UUID=<UUID> / LABEL=<LABEL> : Mostra o dispositivo através de seu UUID ou Label
- findmnt : Exibe uma relação de todas as partições montadas.
- **sync** : Efetiva a gravação de dados que estão em cache/buffer da memória para o disco
- umount: Usado para desmontar as partições. Segue o mesmo padrão de uso do comando mount.

Criação e Montagem da Área de Swap

Usando Partições:
- # mkswap /dev/sdX
- # swapon /dev/sdX
- Verificar: # swapon -s ou # cat /proc/swaps
- Configuração no /etc/fstab
  - /dev/sdX swap swap defaults 0 0
  - /dev/sdX none swap sw 0 0
- # swapon -a : Inclui na área de swap as partições/arquivos definidos no /etc/fstab

Usando Arquivos:
- # dd if=/dev/zero of=/arquivoswap bs=1024K count=1000
- # mkswap /arquivoswap
- # swapon /arquivoswap
- Configuração no /etc/fstab
  - /arquivoswap swap swap defaults 0 0
  - /arquivoswap none swap sw 0 0

Systemd Mount

O ponto de montagem pode ser configurado através de uma Unit no Systemd.

Para isso basta criar um arquivo do tipo ponto-de-montagem.mount nos diretórios de configuração do systemd (/lib/systemd/system ou /etc/systemd/system/).

A seguir um exemplo para o arquivo opt-teste.mount:
[Unit]
Description=Ponto de Montagem Teste

```
[Mount]
What=/dev/sdb1
Where=/opt/teste
Type=ext4
Options=defaults

[Install]
WantedBy=multi-user.target
```

Após a configuração é importante fazer um reload no systemd pelo comando:

- # systemctl daemon-reload

Desssa forma a partição pode ser administrada pelo comando systemctl:

- # systemctl stop opt-teste.mount
- # systemctl start opt-teste.mount
- # systemctl enable opt-teste.mount

203.2 – Manutenção de um Sistema de Arquivos Linux

Comandos mkfs, mkfs.* e mke2fs

Função: Comandos utilizados para criar a estrutura de um filesystem específico em um dispositivo.

Uso e Principais Opções:

mkfs -t <tipo-fs> <device> : Exemplo: # mkfs -t ext4 /dev/sdb1
mkfs.<tipo-fs> <device> : Exemplo: # mkfs.ext4 /dev/sdb1
mke2fs -t <ext*> <device> : Cria uma partição ext*, se o -t não for utilizado, cria como ext2.
mkfs -c … : A opção opção "-c" faz com que seja realizada uma checagem por badblocks no device antes da criação do fs.

Informações Importantes:

- O comando mkfs é na verdade um frontend que chama os mkfs.* específicos de cada tipo de filesystem
- O comando mke2fs é voltado para filesystems ext2/ext3/ext4
- Em resumo, um sistema de arquivos é um conjunto de estruturas lógicas e funções que determinam como os dados serão armazenados em um dispositvo

---

Comando dumpe2fs

Função: Utilizado para verificar os parâmetros e propriedades de uma partição que utiliza ext2/ext3/ ext4.

Uso e Principais Opções:

dumpe2fs <device> : Exibe todas as informações do dispositivo, inclusive os grupos de blocos
dumpe2fs -h <device> : Exibe apenas as parâmetros de um dispositvo
dumpe2fs -b <device> : Exibe os badblocks, caso hesistam

Comando tune2fs

Função: Utilizado para modificar e ajustar (tunar) os parâmetros de um filesystem ext2/ext3/ext4

Uso e Principais Opções:

tune2fs -l <device> : Lista os parâmetros
tune2fs -L "Label" <device> : Device um nome ou label para o dispositivo
tune2fs -c999 <device> : Define a quantidade máxima de montagens para que o filesystem seja novamente verificado pelo fsck
tune2fs -i30d(w/m) <device> : Define o período máximo de tempo para que o filesystem seja novamente verificado. Pode ser definido em dias (d, padrão), semanas (w) ou meses (m)
tune2fs -mX <device> : Define a %X de blocos reservados para uso apenas de processos privilegiados.
tune2fs -j <device> : Inclui o atributo de journalling a um filesystem do tipo ext2, transformando-o em ext3

Comando debugfs

Função: Realizar alteração de baixo nível em um sistema de arquivos. Um exemplo de uso é recuperar arquivos apagados em sistemas ext2 e ext3.

Uso e Principais Opções:

debugs <device> : Entra em modo de prompt de comando para realização dos procedimentos.

Comando badblocks

Função: Procurar por badblocks em uma partição.

Uso e Principais Opções:

badblocks <device>
badblocks <device> -o arquivo.txt : Resultado será escrito no arquivo.txt

---

Comandos fsck, fsck.* e e2fsck

Função: Comandos utilizados para checar e reparar um filesystem.

Uso e Principais Opções:

fsck -t <tipo-fs> <device> : Exemplo: # fsck -t ext4 /dev/sdb1
fsck.<tipo-fs> <device> : Exemplo: # fsck.ext4 /dev/sdb1
fsck <device> : Dessa forma o tipo do filesystems será identificado automaticamente
e2fsck -t <ext*> <device> : Verifica uma partição do tipo ext*, se o -t não for utilizado, irá identificar automaticamente
fsck -y … : Responde sim para todas as confirmações de correção
fsck -n … : Responde não para todas as confirmações de correção. Na prática apenas verifica um filesystem mas não o corrige.
fsck -f … : Força a execução mesmo que o filesystem esteja indicado como “clean”
fsck -p … : Realiza todos os procedimentos sem pedir nenhuma confirmação do usuário

Informações Importantes:

- Assim como o mkfs, o fsck é um frontend para os fscks específicos de cada filesystem
- O comando e2fsck é voltado para fs do tipo ext2/ext3/ext4

Retornos do fsck:

- 0: Nenhum erro
- 1: Erros encontrados e corrigidos
- 2: O sistema deve ser reiniciado
- 4: Erros encontrados e deixados sem correção
- 8: Erro operacional
- 16 : Erro de uso ou sintaxe
- 128 : Erro de biblioteca compartilhada

BTRFS – Btree File System

- Foco em Tolerância a Falhas e Facilidade de Administração e Reparo
- Implementa RAID 0, 1 e 10 (5 e 6 em desenvolvimento)
- Possibilita a criação de SubVolumes e Snapshots
- Utiliza gravação por CoW (Copy-on-Write)

Comandos - Uso e Principais Opções:

btrfs filesystem show : Exibe informações de partições utilizando btrfs
btrfs subvolume create /ponto/montagem/Subvolume1 : Cria um Subvolume em /ponto/montagem
btrfs subvolume list /ponto/montangem [-t] : Lista os subvolumes criados em /ponto/montagem
btrfs subvolume show /ponto/montagem/Subvolume1 : Exibe detalhes do Subvolume1

mount -o subvol=Subvolume1 /dev/sdb6 /outra/montagem : Monta o Subvolume1 do /dev/sdb6 em outro diretório

btrfs subvolume snapshot /ponto/montagem/Subvolume1 /ponto/montagem/Subvolume1/Snap : Cria um snapshot do Subvolume1

mkfs.btrfs -d raid0/1/10 /dev/sdb1 /dev/sdb2 -f : Faz com que os devices /dev/sdb1 e /dev/sdb2 componham um dispositivo de RAID.

btrfs-convert <device> : Converte um dispositivo de ext* para btrfs

---

XFS

- Desenvolvido pela Silicon Graphics para o IRIX
- Robusto e Escalável
- Trabalha bem com arquivos grandes
- Redimensionamento/Desfragmentação Online
- Tamanho de Blocos Variáveis
- Utiliza Journaling

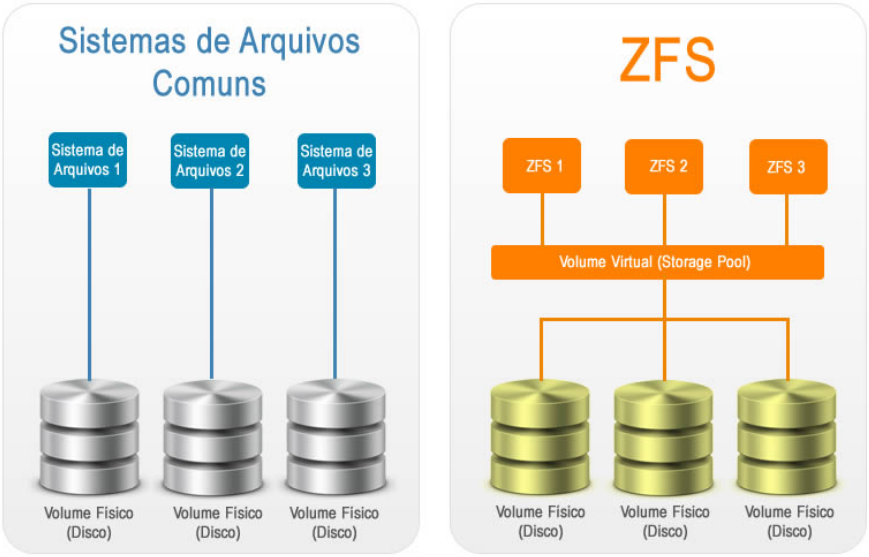
Comandos - Uso e Principais Opções:

xfs_info : Exibe informações do filesystem XFS
xfs_check : Checar por erros e/ou incosistências no filesystem. Substituido pelo xfs_repair.
xfs_repair : Checa e Repara erros e/ou incosistências em filesystems XFS
xfsdump : Cria um dump da partição para ser utilizada para backup
xfsrestore : Restaura um dump criado pelo xfsdump

ZFS (Noções)

- OpenZFS – Alternativa de Código Aberto
- Foco na Integridade dos Dados
- Facilidade de Gerenciamento
- Sistemas de Arquivos em Pools
- Escalabilidade
- Implementa recursos RAID
- Utiliza CoW (Copy-on-Write)

- Principais comandos:
  - zfs
  - zfspool



SMART (Self-Monitoring, Analysis ad Reporting Technology)

Função: Utilizado para verificar a saúde dos discos, erros nos dispositivos e controladoras que não são corrigidos pelo fsck. Utilizado apenas para identificar os erros, que na grande maioria dos casos não pode ser corrigido.

- Daemon: **smartd**
- Configuração: /etc/smartd.conf
- Utilitário: **smartctl**
  - # smartctl -a : Exibe todas as informações para o dispositivo, inclusive testes
  - # smartctl -i : Exibe apenas as informações do dispositivo
  - # smartctl -H : Exibe o resultado do teste de “saúde” (Health) do dispositvo

203.3 – Criando e Configurando Opções dos Sistemas de Arquivos

Montagem Automática via autofs

O recurso de montagem automática permite que uma partição só seja montada quando ela for realmente utilizada, assim como ela será desmontada após um tempo de inatividade.

Uma das formas de implementar esse recurso é através do **autofs**, pelo comando automount.

O processo **automount** roda como um daemon, monitorando os diretórios especificados e realizando a montagem quando forem utilzados.

Arquivos de Configuração para o autofs:

- /etc/auto.master (Master Map) : Arquivo em que são informados os diretórios que devem ser monitorados pelo automount. Exemplos:
  - /mnt/autofs /etc/auto.exemplo --timeout 30
  - /home /etc/auto.home
- /etc/auto.xxxxx : Arquivo de configuração específica de cada ponto de montagem automática. Exemplos:
  - exemplo -fstype=auto :/dev/sdb5
  - usuario -fstype=nfs 10.0.0.50:/dev/sdb5

Montagem Automática via systemd

O systemd também possibilita a criação de Units para a montagem automática. Para isso devem ser criados duas unidades, um .mount e um .automont, conforme o exemplo a seguir:

```
# cat mnt-systemdauto.mount
[Unit]
Description=Exemplo

[Mount]
Where=/mnt/systemdauto
What=/dev/sdb5

[Install]
WantedBy=multi-user.target

# cat mnt-systemdauto.automount
[Unit]
Description=Exemplo

[Automount]
Where=/mnt/systemdauto

[Install]
WantedBy=multi-user.target
```

Para que o systemd leia as novas unidades deve ser utilizado o comando “systemctl daemon-reload”

Criação de Imagens e Gravação de Cds/DVDs

Para criar uma imagem a partir de um CD/DVD:
dd if=/dev/sr0 of=imagem.iso

Para gravar uma imagem em um CD/DVD:
cdrecord dev=/dev/sr0 imagem.iso

Gerando Imagens a Partir de Arquivos e Diretórios

Feito através do comando **mkisofs** (genisoimage). Por padrão, a imagem é gerada usando o filesystem ISO 9660.

Exemplos e Opções de Uso:

Gerar uma imagem contendo todos os arquivos dos diretórios informados:
mkisofs -o imagem.iso /home/usuario /etc/config/

Gerar uma imagem contendo de maneira separada os arquivos de cada um dos diretórios informados:
mkisofs -o imagem.iso -graft-point dir1=/home/usuario dir2=/etc/config

O uso do ISO 9660 traz uma série de limitações na geração das imagens, principalmente relacionada à compatibilidade entre sistemas operacionais. Entre esses limitações temos:

- Uso do padrão de nomes 8.3, por exemplo, arquivos.pdf
- Limitações relacionados aos caracteres especiais (Unicode)
- Não mantém permissões e propriedades
- Não copia links simbólicos

Para solucionar essas limitações foram desenvolvidas extensões aos ISO 9660, entre elas:

- Joliet, opção -J
  - Habilita nomes longos, não ficando restritos ao formato 8.3
  - Suporte à codificação Unicode, possibilitando espaços e caracteres especiais
  - Indicado para quando o CD deva ser lido tanto em sistemas Linux como Windows
- Rock Ridge, opção -R
  - Além dos recursos do Joliet, possibilita a cópia de links simbólicos
  - Preserva permissões e propriedades dos arquivos

Além disso, o mkisofs tem as seguintes opções:

- El Torito
  - Utilizado para criar CDs Bootáveis
- HFS (-hfs)
  - Habilitar o uso do CD em sistemas Macintosh, que utiliza filesystems do tipo HFS
- UDF (-udf)
  - Cria a imagem usando o filesystem UDF, ao invés do ISO 9660
  - O UDF (Universal Disk Format) visa ser um padrão único, que engloba os padrões de nomes e atributos de todos os sistemas operacionais
  - Muito utilizado para criação de DVDs

Criptografia do Sistema de Arquivos

Os dados de um sistema de arquivos podem ser totalmente critptografados de forma que apenas através de uma senha ele possa ser montado para uso.

Para esse processos, são utilizados os seguintes recursos e ferramentas:

- LUKS (Linux Verified Key Setup) : Método de encriptação de dados
- dm-crypt : Sistema que faz parte do Device Mapper para realizar a criptografia dos devices
- Device Mapper : framework que cria devices virtuais para mapear blocos físicos, muito utilizado pro exemplo com LVD e RAID
- cryptsetup : Comando que irá utilizar os recursos do dm-crypt e LUKS para criar e gerenciar os devices criptografados

Processo de Criação e Montagem de um Dispositivo Criptografado:

- Preparar o device e definir uma senha:
  - # cryptsetup -v –verify-passphrase luksFormat /dev/sdXY
- Ativar a criptografia do device
  - # cryptsetup luksOpen /dev/sdXY nome-virtual
  - [ou] # cryptsetup open --type luks /dev/sdXY nome-virtual

  Formatar a partição com algum sistema de arquivos
  - # mkfs.<fs> /dev/mapper/nome-virtual
- Montar a partição
  - # mount /dev/mapper/nome-vitual /ponto/montagem
- Para montagem durante o boot, configurar o arquivo /etc/crypttab
  - nome-virtual /dev/sdXY none luks
  - nome-virtual /dev/sdXY /arquivosenha luks
- Configurar o /etc/fstab
  - /dev/mapper/nome-virtual /ponto/montagem auto defaults 0 0

Outras opções:

- luksDump – Exibe informações do cabeçalho e informações de um dispositivo que está utilizando o LUKS
  - # cryptsetup luksDump /dev/sdXY