

Sessão 5: Sistema de arquivos



As atividades desta sessão serão realizadas na máquina virtual *Client_Linux*.



Em algumas atividades, você trabalhará com a conta *root*, o que lhe dará todos os direitos sobre os recursos do sistema. Seja cauteloso antes de executar qualquer comando.

1) Obtendo informações sobre sistemas de arquivos e partições

Verifique quais são as opções do comando *df* e responda:

1. Quais *file systems* foram definidos no seu sistema?

```
$ cat /etc/fstab | grep -v '^#' | awk '{print $3}' | sort | uniq
ext4
swap
udf,iso9660
```

Alternativamente, verifique no arquivo */etc/fstab* o campo *type* de cada partição.

2. Qual partição ocupa maior espaço em disco?

```
$ df -m | awk 'NR>1' | awk '{print $2,$1}' | sort -n | tac | head -n1
29910 /dev/sda1
```

Alternativamente, verifique com o comando *df -h* a partição que possui o maior número de bytes em uso, na coluna *"Used"*.

3. Qual é o *device* correspondente à partição raiz?

```
$ df -h | egrep ' /$' | awk '{print $1}'
/dev/sda1
```

Alternativamente, verifique através do comando *df -h* a linha que possui no campo *"Mounted on"* o caractere */* e em seguida, nesta mesma linha, verificar o *device* correspondente no campo *"Filesystem"*.

4. Os discos do computador que você está utilizando são do tipo *IDE* ou *SCSI*?

```
$ dmesg | egrep 'Attached.*disk'
[ 10.310957] sd 1:0:0:0: [sdb] Attached SCSI disk
[ 10.358641] sd 0:0:0:0: [sda] Attached SCSI disk
```

Alternativamente, verifique através do comando `df -h`, o campo "*Filesystem*". Discos **IDE** são representados pelos dispositivos `/dev/hda`, `/dev/hdb`, `/dev/hdc`, etc. Discos **SCSI** são representados pelos dispositivos `/dev/sda`, `/dev/sdb`, `/dev/sdc`, etc.

5. A que partição pertence o arquivo `/etc/passwd`?

```
$ df -T /etc/passwd | sed -n '1!p' | awk '{print $1}'
/dev/sda1
```

Alternativamente, verifique através do comando `df` em qual partição se encontra o diretório `/etc`.

6. Você faria alguma crítica em relação ao particionamento do disco do computador que você está utilizando? Como você o reparticionaria?

O aluno deve avaliar o esquema de particionamento adotado e responder à pergunta levando em conta as vantagens obtidas com o particionamento, como isolamento de falhas, ganho de performance, etc.

2) Determinando o espaço utilizado por um diretório

1. Que subdiretório do diretório `/var` ocupa maior espaço em disco?

```
# du -sm /var/* | sort -n | tac | head -n1
97      /var/lib
```

Alternativamente, verifique através do comando `du -mcs /var/*` qual diretório ocupa maior espaço em disco.

2. Faça um *script* para monitorar a taxa de utilização das partições de um servidor. Este script deve enviar um e-mail ao usuário `root` caso a taxa de utilização de um ou mais partições ultrapasse 90% de uso. O e-mail deve informar o(s) *filesystem(s)* e sua(s) respectiva(s) taxa(s) de utilização (somente se estiver acima de 90%).

O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

parts=( $( df -h | egrep -e "^/dev" | awk {'print $6'} ) )
partusage=( $( df -h | egrep -e "^/dev" | awk {'print $5'} | tr -d % ) )
out="$( mktemp )"

for (( i=0; i<${#parts[@]}; i++ )); do
    if [ ${partusage[$i]} -gt 90 ]; then
        echo -e "Filesystem ${parts[$i]} over ${partusage[$i]}% capacity." >> $out
    fi
done

if [ -e $out ]; then
    mail -s "Filesystem capacity report" root@localhost < $out
    rm -f $out
fi
```

3) Criando uma nova partição e definindo um novo sistema de arquivos

Você, como administrador de um sistema, pode, a qualquer instante, deparar-se com um problema gerado por uma aplicação que necessita de maior espaço em disco para armazenar informações (isso é muito comum em sistemas de banco de dados). Nessas situações, normalmente, um novo disco é adicionado ao sistema.



A execução desta atividade depende da existência de um espaço não alocado no sistema. Caso não exista este espaço e esta atividade esteja sendo executada em um ambiente virtualizado, pode-se ter a facilidade de adicionar um novo disco à máquina virtual. Consulte o instrutor sobre como proceder.

1. Faça login como usuário **root**. Deve haver um espaço não utilizado no disco do seu cliente. Você deve adicionar esse espaço ao sistema, criando uma partição do tipo utilizado pelo Linux.
 - Primeiro, vamos verificar quais discos foram conectados ao sistema durante o *boot*:

```
# dmesg | egrep 'Attached.*disk'
[ 10.310957] sd 1:0:0:0: [sdb] Attached SCSI disk
[ 10.358641] sd 0:0:0:0: [sda] Attached SCSI disk
```

- Vamos checar o estado de uso desses discos, começando pelo **/dev/sda**:

```
# fdisk -l /dev/sda
```

Disco /dev/sda: 40 GiB, 42949672960 bytes, 83886080 setores

Unidades: setor de 1 * 512 = 512 bytes

Tamanho de setor (lógico/físico): 512 bytes / 512 bytes

Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes

Tipo de rótulo do disco: dos

Identificador do disco: 0x27232fb6

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	62500863	62498816	29,8G	83	Linux
/dev/sda2		62502910	83884031	21381122	10,2G	5	Extended
/dev/sda5		62502912	66406399	3903488	1,9G	82	Linux swap / Solaris
/dev/sda6		66408448	83884031	17475584	8,3G	83	Linux

- O disco **/dev/sda** já está sendo utilizado, e aparentemente está cheio. Vamos então verificar o dispositivo **/dev/sdb**:

```
# fdisk -l /dev/sdb
```

Disco /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 setores

Unidades: setor de 1 * 512 = 512 bytes

Tamanho de setor (lógico/físico): 512 bytes / 512 bytes

Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes

- Perfeito, parece estar vazio. Vamos formatá-lo e criar uma única partição Linux ocupando a totalidade do espaço livre:

```
# fdisk /dev/sdb
```

Bem-vindo ao fdisk (util-linux 2.25.2).

As alterações permanecerão apenas na memória, até que você decida gravá-las.
Tenha cuidado antes de usar o comando de gravação.

A unidade não contém uma tabela de partição conhecida.

Created a new DOS disklabel with disk identifier 0x4fa0acac.

Comando (m para ajuda): o

Created a new DOS disklabel with disk identifier 0xb33d8f79.

Comando (m para ajuda): n

Tipo da partição

p primária (0 primárias, 0 estendidas, 4 livre)

e estendida (recipiente para partições lógicas)

Selecione (padrão p):

Usando resposta padrão p.

Número da partição (1-4, padrão 1):

Primeiro setor (2048-16777215, padrão 2048):

Último setor, +setores ou +tamanho{K,M,G,T,P} (2048-16777215, padrão 16777215):

Criada uma nova partição 1 do tipo "Linux" e de tamanho 8 GiB.

Comando (m para ajuda): t

Selecionou a partição 1

Código hexadecimal (digite L para listar todos os códigos): 83

O tipo da partição "Linux" foi alterado para "Linux".

Comando (m para ajuda): w

A tabela de partição foi alterada.

Chamando ioctl() para reler tabela de partição.

Sincronizando discos.

- Finalmente, vamos verificar se o procedimento produziu o resultado esperado:

```
# fdisk -l /dev/sdb
```

Disco /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 setores

Unidades: setor de 1 * 512 = 512 bytes

Tamanho de setor (lógico/físico): 512 bytes / 512 bytes

Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes

Tipo de rótulo do disco: dos

Identificador do disco: 0xb33d8f79

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	16777215	16775168	8G	83	Linux

2. Formate a partição com o sistema de arquivos **ext4**.

```
# mkfs.ext4 /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 2096896 4k blocks and 524288 inodes
Filesystem UUID: 2464c725-9356-4abb-8a9f-a2de3d64e7ac
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

3. Crie um *mount point* chamado **/dados** e monte nele a nova partição.

```
# mkdir /dados
# mount -t ext4 /dev/sdb1 /dados
# mount | egrep '^/dev/sdb1'
/dev/sdb1 on /dados type ext4 (rw,relatime,data=ordered)
```

4. Qual a quantidade de espaço em disco que foi reservada para armazenar os dados dos *inodes*? E da partição em si?

Para calcular o espaço solicitado, o primeiro passo é descobrir quantos *inodes* foram criados, e qual o tamanho de cada um deles:

```
$ sudo tune2fs -l /dev/sdb1 | egrep -i 'inode count|inode size'
Inode count:          524288
Inode size:           256
```

Feito isso, basta multiplicar os dois valores e, opcionalmente, mostrar o resultado em um formato mais legível, já que o **tune2fs** mostra o tamanho dos *inodes* em bytes:

```
# s=( $(tune2fs -l /dev/sdb1 | egrep -i 'inode count|inode size' | awk '{print $3}') ); echo "$(( ${s[0]} * ${s[1]} / 1048576 )) MB"
128 MB
```

5. Cheque a partição criada com o comando apropriado. Que tipos de checagens foram realizados?

```

# umount /dev/sdb1
# e2fsck /dev/sdb1 -fv
e2fsck 1.42.12 (29-Aug-2014)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information

    11 inodes used (0.00%, out of 524288)
    0 non-contiguous files (0.0%)
    0 non-contiguous directories (0.0%)
    # of inodes with ind/dind/tind blocks: 0/0/0
    Extent depth histogram: 3
70287 blocks used (3.35%, out of 2096896)
    0 bad blocks
    1 large file

    0 regular files
    2 directories
    0 character device files
    0 block device files
    0 fifos
    0 links
    0 symbolic links (0 fast symbolic links)
    0 sockets

-----
    2 files

```

6. Tome as medidas necessárias para que essa partição seja montada toda vez que o sistema for reiniciado, e verifique se isso acontece de fato.

Deve-se inserir a linha abaixo ao final do arquivo `/etc/fstab`.

```

/dev/sdb1  /dados/  ext4  defaults,errors=remount-ro  0  2

```

Feito isso, reinicie o sistema e verifique a montagem do *filesystem*.

Atualmente, é muito comum sistemas Linux indicarem os *filesystems* no arquivo `/etc/fstab` através de seu UUID (*Universally Unique Identifier*), em lugar de nome de dispositivo, já que a ordem em que os discos são detectados pelo kernel não é determinística — em uma instância de *boot* um disco pode ser detectado como `/dev/sda`, e na próxima, como `/dev/sdb`. Para identificar a partição que acabamos de criar através do seu UUID, siga os passos abaixo:



```
# ls -l /dev/disk/by-uuid/ | egrep 'sdb1$' | awk '{print $9}'
2464c725-9356-4abb-8a9f-a2de3d64e7ac

# uuid="$(ls -l /dev/disk/by-uuid/ | egrep 'sdb1$' | awk '{print $9}')" ; echo "UUID=$uuid /dados ext4 defaults,errors=remount-ro 0 2" >> /etc/fstab

# egrep ' /dados ' /etc/fstab
UUID=2464c725-9356-4abb-8a9f-a2de3d64e7ac /dados ext4
defaults,errors=remount-ro 0 2
```

4) Trabalhando com o sistema de *quotas*

Em sistemas compartilhados por muitos usuários, a competição por espaço em disco costuma gerar conflitos que acabam prejudicando o desempenho do sistema e os próprios usuários, caso não haja controle de uso dos recursos. Neste exercício, veremos como habilitar e configurar o sistema de *quotas* do Linux.

1. Faça login com a conta do usuário `root`. Verifique se o sistema de *quotas* está instalado. Se ainda não estiver, execute a instalação.

Verifique se o pacote `quota` está instalado no sistema com o comando `dpkg -l | grep quota`. Caso não esteja, instale-o usando o `apt-get`:

```
# dpkg -l | grep ' quota '
# apt-get -y install quota quotatool
```

2. O próximo passo é habilitar o sistema de *quotas* para a partição raiz. Faça isso seguindo os procedimentos descritos na parte teórica dessa sessão de aprendizagem.

Insira no arquivo `/etc/fstab` o suporte à *quota* de disco na partição raiz com as opções apropriadas:

```
# grep ' / ' /etc/fstab | grep -v '^#'
UUID=6d035549-c33d-4f72-a751-1e7ddc602dbe / ext4 errors=remount-ro,usrquota,grpquota 0 1
```

Feito isso, reinicie o sistema e verifique se o suporte a *quotas* foi habilitado através do comando

mount:

```
# mount | egrep '^/dev/sda1'
/dev/sda1 on / type ext4 (rw,relatime,quota,usrquota,grpquota,errors=remount-
ro,data=ordered)
```

3. Crie uma conta de usuário para teste e configure o limite desse novo usuário para 200 MB, utilizando o comando `edquota`.

Primeiro, vamos criar o usuário. Em seguida, editar seu arquivo de *quota*:

```
# useradd -m pedro
# edquota -u pedro
```

O comando `edquota` irá invocar um editor (indicado pela variável de ambiente `$EDITOR`) para que as *quotas* sejam ajustadas. Vamos editar os campos *soft* e *hard* da seção *block* do arquivo — note que os valores devem ser informados em *kBytes*. Pode-se, opcionalmente, também setar um limite para *inodes* que o usuário pode criar.

```
Disk quotas for user pedro (uid 1005):
  Filesystem      blocks      soft      hard      inodes      soft
  hard
  /dev/sda1        16    100000    200000         4         0
  0
```

4. Saia do sistema e entre novamente como o usuário de teste que acaba de ser criado. Como pode ser verificado, a partir dessa conta, as *quotas* de uso de disco? E o espaço efetivamente utilizado?

```
# su - pedro

$ quota -u
Disk quotas for user pedro (uid 1005):
  Filesystem blocks quota limit grace files quota limit grace
  /dev/sda1   16 100000 200000         4      0      0
```

Na listagem acima, pode-se observar que o usuário `pedro` está utilizando 16 kB de espaço em disco, com um *soft limit* de 100 MB e um *hard limit* de 200 MB.

5. Crie dois arquivos no diretório, utilizando os comandos `cp` e `ln` (criando um link simbólico). Há diferença na forma como o espaço ocupado por esses dois arquivos é contabilizado no sistema de quotas?

```

$ pwd
/home/pedro

$ quota -u
Disk quotas for user pedro (uid 1005):
    Filesystem  blocks    quota  limit  grace  files   quota  limit  grace
    /dev/sda1    16  100000 200000         4      0      0

```

```

$ cp /boot/vmlinuz-3.16.0-6-amd64 ~
$ ls
vmlinuz-3.16.0-6-amd64
$ quota -u
Disk quotas for user pedro (uid 1005):
    Filesystem  blocks    quota  limit  grace  files   quota  limit  grace
    /dev/sda1   3116  100000 200000         5      0      0

```

```

$ ln -s /boot/vmlinuz-3.16.0-6-amd64 ~/kernel-link
$ ls
kernel-link  vmlinuz-3.16.0-6-amd64
$ quota -u
Disk quotas for user pedro (uid 1005):
    Filesystem  blocks    quota  limit  grace  files   quota  limit  grace
    /dev/sda1   3116  100000 200000         6      0      0

```

A forma de contabilização é diferente: o tamanho do link simbólico corresponde apenas ao tamanho em bytes do *path* completo até o arquivo apontado; já o arquivo criado com o comando **cp** possui o mesmo tamanho do arquivo original.

6. Como determinar se o sistema de *quotas* está habilitado na inicialização do sistema? E, se não estiver como habilitá-lo?

Em sistemas com o sistema de *init* **systemd**, como é o caso do Debian e da maioria das distribuições Linux atuais, podemos usar o comando **# systemctl is-enabled** para determinar o estado de um *daemon* durante a inicialização do sistema:

```

# systemctl is-enabled quota
enabled

```

Para desabilitar um serviço, basta usar a palavra-chave **disable**. Ao contrário, para habilitá-lo, utilize **enable**:

```
# systemctl disable quota
Synchronizing state for quota.service with SysVinit using update-rc.d...
Executing /usr/sbin/update-rc.d quota defaults
Executing /usr/sbin/update-rc.d quota disable
insserv: warning: current start runlevel(s) (empty) of script 'quota' overrides LSB
defaults (S).
insserv: warning: current stop runlevel(s) (0 6 S) of script 'quota' overrides LSB
defaults (0 6).
# systemctl is-enabled quota
disabled

# systemctl enable quota
Synchronizing state for quota.service with SysVinit using update-rc.d...
Executing /usr/sbin/update-rc.d quota defaults
insserv: warning: current start runlevel(s) (empty) of script 'quota' overrides LSB
defaults (S).
insserv: warning: current stop runlevel(s) (0 6 S) of script 'quota' overrides LSB
defaults (0 6).
Executing /usr/sbin/update-rc.d quota enable
# systemctl is-enabled quota
enabled
```

7. Teste a efetividade do sistema de *quotas*:

```
# su - pedro

$ quota -u
Disk quotas for user pedro (uid 1005):
    Filesystem blocks quota limit grace files quota limit grace
    /dev/sda1   20  100000 200000          5      0      0

$ du -sk /boot/vmlinuz-3.16.0-6-amd64
3100    /boot/vmlinuz-3.16.0-6-amd64

$ for i in {1..1000}; do cp /boot/vmlinuz-3.16.0-6-amd64 ~/kernel-$i; done
sda1: warning, user block quota exceeded.
sda1: write failed, user block limit reached.
cp: erro escrevendo "/home/pedro/kernel-65": Disk quota exceeded
```

Através do comando acima, o usuário **pedro** conseguiu copiar para seu diretório *home* a imagem do kernel Linux, copiada do **/boot** e com tamanho de 3100 kB, por 64 vezes até que o *hard limit* de *quota* fosse ativado, e novas cópias fossem desabilitadas.

8. Faça um *script* que defina o esquema de *quota* para todos os usuários do sistema baseado nas cotas de um usuário passado como parâmetro para esse *script*.

O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

if [[ $EUID -ne 0 ]]; then
    echo "  [*] Not root!" 1>&2
    exit 1
fi

for user in $( getent shadow | awk -F: '$2 != "*" && $2 !~ /^!/ { print $1 }' ); do
    edquota -u ${user} -p $1
done
```

Note, no entanto, que apesar de o *script* acima ser minimamente funcional, há alguns parâmetros importantes que não sendo testados no momento:

- O usuário passado como parâmetro para o *script* existe?
- Está sendo removido o usuário **root** da lista de usuários para aplicação de **quota**?
- Está sendo removido o próprio usuário passado como parâmetro da lista de usuários para aplicação de **quota**?

A resposta para todos esses itens, evidentemente, é não. Poderíamos estender o script para fazer essas funções, mas no intuito de mostrar uma abordagem diferente para o problema, veja abaixo uma solução equivalente, mais completa, usando a linguagem Python:

```
#!/usr/bin/python

import os, sys, subprocess, pwd, spwd

if os.geteuid() != 0:
    exit(' Not root?')

if len(sys.argv) <= 1:
    exit(' Usage: ' + sys.argv[0] + ' TEMPLATE_USER')

try:
    pwd.getpwnam(sys.argv[1])
except KeyError:
    exit('No such \' + sys.argv[1] + \' user')

qusers = []

for user in pwd.getpwall():
    if user[0] == 'root' or user[0] == sys.argv[1]:
        continue

    phash = spwd.getspnam(user[0]).sp_pwd

    if phash != '*' and not phash.startswith('!'):
        qusers.append(user[0])

for user in qusers:
    subprocess.call(['edquota', '-u', user, '-p', sys.argv[1]])
```

O que você achou da solução acima? Mais fácil, mais difícil ou apenas diferente? Lembre-se, ao atuar como um administrador de redes e sistemas não se deve ficar preso a um único tipo de ferramenta ou solução, mas sim utilizar a melhor alternativa possível para resolver o problema.