

Ficha 4 – Gestão de utilizadores**Tópicos abordados:**

- Gestão de utilizadores
- Sistemas de ficheiros
- Gestão de quotas

1 Gestão de utilizadores

A gestão de utilizadores (e grupos) de uma organização é uma tarefa muito importante da competência da administração de sistemas. Ou seja, o administrador de sistemas deverá saber o que são as contas de utilizadores e como funcionam, ao nível do sistema operativo. Genericamente, uma conta de utilizador consiste num identificador de um utilizador “real”, com informações relevantes sobre a sua autorização para usar o sistema e, em caso afirmativo, quais os recursos que está habilitado a usar. O administrador de sistemas começa então por criar uma conta de utilizador com as respectivas credenciais (palavra-chave) ao utilizador em questão, para que este se identifique perante a operação de login do sistema operativo. Uma vez autorizado a usar o sistema, o utilizador poderá utilizar os recursos do mesmo com base em permissões. As permissões são também objeto de estudo desta ficha e serão apresentadas mais adiante.

1.1 Utilizadores e identificadores

No sistema operativo Linux, apesar de cada utilizador (incluindo o administrador de sistema) se autenticar com base num nome de utilizador e numa palavra-chave, internamente o sistema identifica esse mesmo utilizador com base num identificador numérico, denominado de User ID ou UID (por questões de desempenho). Isto também quer dizer que sempre que o sistema operativo necessite de comunicar com o utilizador haverá a necessidade de converter o identificador numérico, em informação “confortável” para o utilizador, ou seja, o seu login.

Nota: Regra geral, os UID abaixo de 500 estão reservados para uso do sistema, o que significa que quando se cria o primeiro utilizador no sistema lhe seja atribuído o UID 500.

1.1.1 Criação de contas de utilizadores

[adduser]

```
$ sudo adduser user1 → a forma mais simples de criar uma conta de utilizador
```

Neste exemplo, o próprio comando solicita toda a informação essencial à criação do utilizador no sistema, como a palavra-chave, nome completo, telefones, etc. Contudo, neste caso mais simples, a área de trabalho é criada no diretório por omissão (/home/user1), sendo também criado um grupo, com o mesmo nome que o utilizador (user1), que será o grupo principal a que o utilizador vai pertencer. Se for necessário personalizar mais a criação de determinado utilizador, então devem-se passar argumentos adicionais no comando.

Por exemplo:

```
$ sudo adduser --home /tmp/user2 --uid 5000 --ingroup admin user2
```

Este comando cria o utilizador user2, tal como foi criado o utilizador user1, mas agora o administrador de sistemas especifica que a área de trabalho fique em /tmp, o identificador do utilizador seja o 5000, e que o mesmo tenha como grupo principal o grupo admin. Neste caso é obrigatório que o grupo especificado exista. Para confirmar os detalhes do utilizador user1 e user2, acabados de criar, use o seguinte comando:

```
$ id user1 ; id user2
```

Para explorar mais opções poderá consultar o manual do comando **adduser**.

[useradd]

O comando **useradd** permite também criar uma conta de utilizador, embora seja menos interativo que o **adduser**.

Algumas opções do comando **useradd**:

- d caminho_diretoria – Indicação do caminho para a área de trabalho do utilizador.
- e data_expiração – Indicação da data de expiração da conta (formato YYYY-MM-DD).
- g grupo – Indicação do grupo principal a que o utilizador será associado.
- G grupo1[,grupo2, ...] – Outros grupos aos quais o utilizador passará a pertencer.
- p palavra-chave – Indicação da palavra-chave cifrada (*output* da função **crypt()**).
- s shell – Indicação do nome da *shell* de *login* do utilizador.
- u uid – Indicação do *User ID*.

```
$ sudo useradd user3
```

Neste caso, o utilizador é criado sem que o administrador tenha que fornecer qualquer informação adicional, sendo atribuído ao utilizador a área de trabalho `/home/user3` (por omissão) e um UID por omissão. A grande particularidade deste comando é que cria o utilizador sem palavra-passe, ou seja, é necessário alterar/especificar a palavra-passe explicitamente com o comando **passwd**.

```
$ sudo passwd user3
```

1.1.2 Ficheiros envolvidos na criação de utilizadores

No Linux a informação dos utilizadores é guardada em vários ficheiros dentro da diretoria `/etc`. O ficheiro mais conhecido é o **`/etc/passwd`**. Este ficheiro é um ficheiro de texto, que contém a lista de utilizadores registados no sistema, um por linha, tendo cada linha sete campos separados por “:”, cada um com significado específico:

Campo 1 – nome do utilizador (login)

Campo 2 – palavra passe cifrada ou um “x” se o ficheiro “`/etc/shadow`” (ver **Nota** mais abaixo) estiver em uso no sistema.

Campo 3 – UID

Campo 4 – GID

Campo 5 – informação adicional sobre o utilizador (nome completo, telefone, etc), separados por vírgula. Comandos como o `finger` servem-se deste tipo de informação

Campo 6 – diretoria de trabalho

Campo 7 – *Shell* que é o programa lançado sempre que o utilizador faz *login*.

Eis um exemplo de uma linha do ficheiro **`/etc/passwd`**:

```
root:x:0:0:root:/root:/bin/bash
```

Esta linha refere-se ao utilizador `root`, o qual tem uma palavra-chave do tipo *shadow*, um UID e GID zero, tem `/root` como diretoria de trabalho e o executável `/bin/bash` como *shell*.

Nota: O mecanismo de palavras-chave *shadow* não é mais do que retirar do ficheiro de utilizadores (`/etc/passwd`) a palavra-chave cifrada de cada utilizador para outro ficheiro (`/etc/shadow`), apenas lido pelo administrador de sistema. O ficheiro `/etc/shadow` guarda as palavras-chave (de forma cifrada) dos utilizadores assim como propriedades adicionais, tais como, a data da última alteração da palavra-chave, data desde a qual a conta foi desligada, etc..

Para saber mais consultar: <http://www.cyberciti.biz/faq/understanding-etcshadow-file/>

1.1.3 Alteração de contas de utilizadores

Depois de ser criada uma conta de utilizador é possível fazer alterações à mesma.

[passwd]

O comando `passwd` permite, para além de alterar/atualizar a palavra-chave de um utilizador, definir a data de expiração de uma palavra-chave, o bloqueio e desbloqueio de uma conta de utilizador, entre outras (ver manual do comando).

```
$ sudo passwd user3
```

[usermod]

Se o objetivo for a alteração de propriedades de uma conta de utilizador então o comando `usermod` será o indicado pois permite, entre outras funcionalidades, alterar a *shell*, indicação de datas para expiração de contas, definir uma nova área de trabalho, etc. Exemplos:

```
$ sudo usermod -s /bin/tcsh user3 -> Alteração da shell
$ sudo usermod -e 2013/12/31 user3 -> Definição da data de expiração da conta
```

Para verificarmos as alterações efetuadas pelos comandos anteriores, temos de consultar os ficheiros **/etc/passwd** e **/etc/shadow**.

Outras opções do comando **usermod**:

- d** diretoria [-m] – Cria nova diretoria de trabalho e move conteúdo da antiga para nova.
- g** grupo – Altera o GroupID (GID) para o grupo especificado.
- G** grupo1[,grupo2, ...] – Outros grupos aos quais o utilizador passará a pertencer.
- l** nome – Altera o *login* do utilizador.
- u** uid – Altera o UID.

Nota: O Linux disponibiliza ainda, o comando **chfn** para alteração do nome completo do utilizador, telefones, etc. (ver manual do comando).

1.1.4 Bloqueio e eliminação de contas de utilizadores

Em determinadas situações e circunstâncias, o administrador de sistema poderá necessitar apenas bloquear uma conta de utilizador, ao invés de a eliminar. Para esse efeito, quer o comando **passwd** (ver ponto 1.1.3) quer o comando **usermod**, através das suas opções (ver manual do comando), permitem-nos bloquear uma conta de utilizador. Analise os seguintes exemplos com o comando **usermod**:

```
$ sudo usermod -L user3 -> bloqueia a conta de utilizador  
$ sudo usermod -U user3 -> desbloqueia a conta de utilizador
```

Ao nível da linha de comandos, o Linux disponibiliza dois comandos semelhantes, para remover contas de utilizadores do sistema: o **userdel** e o **deluser**.

[deluser]

```
$ sudo deluser user2
```

Neste exemplo, o comando **deluser** remove o registo do utilizador do sistema. Contudo, a área de trabalho permanece e o grupo, criado para servir de grupo principal ao utilizador user2, também permanece no sistema. Para remover a área de trabalho do utilizador deve-se obter a designação da mesma, a partir do ficheiro **/etc/passwd** (penúltima coluna), antes de apagar o utilizador e utilizar o comando que se segue:

```
$ sudo rm -r /home/user2
```

Se preferir, poderá dar indicação ao comando **deluser** para também remover a área de trabalho do utilizador:

```
$ sudo deluser --remove-home user2
```

Ou ainda, se desejar remover determinado utilizador, a sua área de trabalho e todos os ficheiros, cujo dono é esse utilizador, poderá usar o comando da seguinte forma:

```
$ sudo deluser --remove-home --remove-all-files user2
```

Nota: Esta operação revela-se demorada, uma vez que o sistema operativo tem que identificar todos os ficheiros do utilizador para os poder remover. É recomendável a remoção prévia de todos os ficheiros do sistema cujo "owner" seja o utilizador que estamos a remover.

[userdel]

```
$ sudo userdel user3
```

O comando **userdel** tem o mesmo objetivo que o comando **deluser**, mas suporta operações mais “abruptas”, as quais, sempre que possível, devem ser evitadas (disponível com a opção *force*). Por exemplo, com o comando **userdel** é possível remover um utilizador, enquanto este está “ligado”. Permite também apagar a diretoria de trabalho, mesmo que outro utilizador também partilhe essa diretoria de trabalho, apagar o grupo primário do utilizador, mesmo que outros utilizadores pertençam ao mesmo grupo, etc.. Ora, este tipo de operações podem deixar o sistema operativo inconsistente e, por isso, estas operações, sempre que possível, não devem ser executadas (opção *force*).

1.2 Criação de utilizadores em lote

A criação de utilizadores em lote é uma das tarefas que facilmente pode ser automatizada. Assuma que a utilização do sistema operativo Linux nas aulas é feita de forma a centralizar as contas de acesso de todos os utilizadores. Ou seja, poderá ser necessário criar dezenas ou até centenas de contas de utilizadores. Assim, torna-se necessário automatizar o procedimento de criação das contas em lote, em alternativa à criação de cada conta separadamente. Poderá desenvolver-se um *script* em Perl para o efeito.

Assumindo que existe um ficheiro que contém os números dos alunos (neste caso, numa coluna única), o código seguinte, permite criar a conta individual de cada aluno, recebendo por linha de comandos, o ficheiro de alunos.

Já em relação à palavra-chave, pode ser utilizada a função **crypt()** num *script* Perl (ver caixa em baixo) ou, se quisermos criar uma palavra-chave cifrada na linha de comandos, podemos utilizar o comando **echo "password"|openssl passwd -1 -stdin**. Este comando vai gerar uma cifra da *string* “password” utilizando o comando openssl. Outra alternativa à função **crypt()** é o comando **pwgen**. Este comando, que poderá não estar instalado, gera palavras-chave aleatórias, ideal para situações de criação de contas de utilizador em bloco.

```
#!/usr/bin/perl
use strict;
use warnings;

my $conta = "";
my $FInput;
my @arrContas_L = ();
```

```

my $strEncPwd = "";
my $strSalt = "LX";

if(@ARGV != 1)
{
    print("Sintaxe: $0 <ficheiro_de_logins>\n");
    exit(1);
}
if(!open($FInput, "$ARGV[0]"))
{
    print("Erro durante a abertura do ficheiro de entrada\n");
    exit(2);
}
@arrContas_L = <$FInput>;
close($FInput);
foreach $conta (@arrContas_L)
{
    chomp($conta);
    $strEncPwd = crypt($conta, $strSalt);
    `sudo useradd -p '$strEncPwd' -m $conta`;
    if($? == 0) # $? Devolve o "código de erro" da última chamada ao sistema
    {
        print("Conta $conta criada com sucesso\n");
    }
    else
    {
        print("Erro ao criar a conta $conta\n");
    }
}

```

IMPORTANTE: O *script* anterior cria as contas de utilizadores, com palavra-chave igual ao *login*, sendo da responsabilidade do administrador de sistemas “obrigar” os utilizadores a mudarem a palavra-chave no primeiro *login*, utilizando para isso o comando **chage -d 0 user** (ver manual do comando).

Exercício 2

Altere o *script* anterior para que o ficheiro passado por parâmetro inclua também a diretoria base. Em relação à palavra-chave, deve ser utilizado um gerador aleatório de palavras-chave (ex: **pwgen**). O *script* deve também validar se o ficheiro tem linha vazias ou comentadas quando iniciadas por “!”. O formato de cada linha do ficheiro, que deve ser validado através de REGEX, é o seguinte:

```
USER#DIRETORIA
```

1.3 Remoção de utilizadores em lote

Assim, como existe a necessidade de criar utilizadores em lote, poderá também ser necessário o contrário, ou seja, eliminar contas de utilizadores em lote. O *script* seguinte faz precisamente isso, recorrendo ao utilitário **userdel**.

```
#!/usr/bin/perl
use strict;
use warnings;

my $conta = "";
my $FInput;
my @arrContas_L = ();

if(@ARGV != 1)
{
    print("Sintaxe: $0 <ficheiro_de_logins>\n");
    exit(1);
}
if(!open($FInput, "$ARGV[0]"))
{
    print("Erro durante a abertura do ficheiro de entrada\n");
    exit(2);
}
@arrContas_L = <$FInput>;
close($FInput);
foreach $conta (@arrContas_L)
{
    chomp($conta);
    `sudo userdel --remove $conta`;
    if($? == 0) # $? Devolve o "código de erro" da última chamada ao sistema
    {
        print("Conta $conta eliminada com sucesso\n");
    }
    else
    {
        print("Erro ao eliminar a conta $conta\n");
    }
}
}
```

Exercício 3

Como é notório, o *script* anterior apaga as contas juntamente com a diretoria de área de trabalho. E os ficheiros do utilizador “espalhados” pelo sistema de ficheiros? Altere o *script*, de modo a que seja feito o *backup* de todos os ficheiros do utilizador a eliminar, antes de se apagarem todos os dados relativos ao mesmo.

2 Gestão de grupos

Similarmente ao que sucede com as contas dos utilizadores, em Linux os grupos criados no sistema são guardados no ficheiro **/etc/group**. Para além da informação de grupos, este ficheiro regista também que utilizadores pertencem a cada grupo. O formato deste ficheiro de texto tem a seguinte estrutura:

```
NomeGrupo:PwdGrupo:Gid:user1,user2,userN
cdrom:x:24:haldaemon,futre
floppy:x:25:haldaemon,futre
```



```
tape:x:26:
...
video:x:44:
...
user1:x:1000:
user2:x:1001:
```

Ou seja, cada linha regista o nome do grupo, a definição da palavra-chave, o identificador do grupo (numérico) e, por fim, os utilizadores que lhe estão associados.

Nota: Não é obrigatório, que cada grupo tenha uma palavra-chave associada. Contudo, se um determinado grupo tiver uma palavra-chave associada, permite que um determinado utilizador que não pertence ao grupo se possa juntar ao mesmo, fornecendo a palavra-chave (ver, por exemplo, o comando **newgrp**).

2.1 Criação de grupos

O Linux disponibiliza dois comandos para criação de grupos: **groupadd** e **addgroup**. Dadas as suas semelhanças, neste documento apenas será abordado o **addgroup**.

[addgroup]

```
$ sudo addgroup grupo1
```

Neste caso, o grupo grupo1 é criado e o sistema atribui o identificador de grupo GID. Para verificar os detalhes do grupo, consultar ficheiro **/etc/group**.

```
$ sudo addgroup --gid 10000 grupo2
```

Já neste exemplo, é o administrador de sistemas que define o identificador de grupo para o novo grupo.

2.2 Eliminação de grupos

A eliminação de grupos pode ser feita usando um de dois comandos: **groupdel** ou **delgroup**.

[delgroup]

```
$ sudo delgroup grupo2
```

Esta linha de comandos permite eliminar o grupo grupo2, caso este não seja o grupo primário de nenhum utilizador, caso contrário o comando falha. Sendo assim, será necessário remover em primeiro lugar o utilizador ou então retirar os utilizadores do grupo que se pretende eliminar.

2.3 Operações envolvendo permissões

De um modo geral em Linux, as permissões que são dadas aos utilizadores, relativamente aos recursos, são organizadas com base em três operações e com base em três conjuntos de utilizadores. As operações resumem-se, respetivamente, à leitura(r), escrita(w) e execução(x). Já os utilizadores alvo dividem-se no dono do recurso, os utilizadores que pertencem ao grupo afeto ao recurso e todos os restantes utilizadores que não se enquadram nos dois conjuntos anteriores.

Como é sabido, um grupo é um mecanismo lógico útil para agrupar recursos, sejam eles utilizadores, ficheiros, diretórios, periféricos, etc. Por exemplo, utilizando grupos, o administrador não tem que definir para cada utilizador o que cada um pode fazer e o que não pode fazer. Em vez disso, associa-se um grupo para “controlo” de determinado recurso e todos os utilizadores que pertencerem a esse grupo ficam automaticamente habilitados a tirar partido do recurso em causa. Desta forma, os grupos facilitam muito trabalho ao administrador de sistemas simplificando a administração do mesmo. De seguida são apresentadas operações que envolvem grupos e utilizadores.

2.4 Adicionar/remover um utilizador de um grupo

Por omissão, qualquer utilizador está sempre associado a um grupo, mesmo que não se especifique esse grupo durante a criação da conta do utilizador no sistema. O próprio sistema operativo encarrega-se de criar um grupo para esse novo utilizador, se for caso disso.

Faz parte das operações de administração associar utilizadores a grupos, assim como retirar utilizadores de grupos, tendo estas operações básicas a finalidade de definir privilégios para os utilizadores em causa.

Vamos supor a existência de uma impressora topo de gama com impressão de alta qualidade. Se apenas determinado grupo restrito de utilizadores podem usar a impressora, então poderá ser criado o grupo impressora, ao qual serão adicionados esses utilizadores. Se por algum motivo determinado utilizador deixe de poder utilizar a impressora, podemos operacionalizar essa alteração simplesmente através da remoção desse utilizador do grupo associado à impressora.

[usermod]

```
$ sudo usermod -g grupo5 user1
```

Este comando define um novo grupo primário para o utilizador user1.

```
$ sudo usermod -G grupo6,grupo7 user1
```

Este comando adiciona o utilizador user1 ao grupo6 e grupo7 (grupos secundários).

```
$ sudo usermod -G grupo7 user1
```

Se este terceiro comando for executado (em sequência relativamente aos dois anteriores), remove o utilizador do grupo secundário grupo6.

Exercício 4

Que comando utilizaria para verificar o resultado dos comandos anteriores?

2.5 Alterar o grupo e/ou o dono de um ficheiro/diretoria

As operações de mudança de dono e de grupo de determinado recurso, constituem também formas de definir privilégios desse mesmo recurso.

[chown]

```
$ sudo chown user3 ficheiro.txt
```

Este comando define o utilizador user3 como novo dono do ficheiro “ficheiro.txt”.

```
$ sudo chown user3 diretoria1
```

Este comando define o utilizador user3 como novo dono da diretoria diretoria1. Neste caso, apenas a diretoria mudou de dono. Para que a operação seja executada para todo o conteúdo da diretoria, em causa deve-se usar o argumento **-R**.

```
$ sudo chown user3:grupo5 ficheiro.txt
```

Este comando permite alterar o dono e o grupo ao ficheiro “ficheiro.txt”.

```
$ sudo chown :grupo4 diretoria1
```

O comando anterior apenas altera o grupo à diretoria diretoria1 (não recursivo). Apesar do comando chown permitir alterar o dono e o grupo a um ficheiro ou diretoria, o Linux tem um comando para ser usado apenas para grupos, cujo nome é **chgrp**.

[chgrp]

```
$ sudo chgrp grupo5 diretoria1
```

O comando anterior permite alterar o grupo à diretoria diretoria1 para grupo5. Mais uma vez, para que a operação tenha efeito no conteúdo da diretoria em causa, deve-se usar o argumento **-R** (modo recursivo).

De forma a complementar as operações de alteração de dono e de grupo, o Linux disponibiliza o comando **chmod** que permite ao administrador ou utilizador definir as permissões em termos de leitura, escrita e execução para o dono do recurso, para o grupo e para todos os outros utilizadores que não pertençam ao universo do dono nem do grupo especificado.

[chmod]

```
$ sudo chmod 700 ficheiro1
```

Este comando permite definir que no sistema operativo apenas o dono pode ler, escrever e executar o ficheiro `ficheiro1`. Os restantes universos de utilizadores não terão qualquer privilégio sobre o recurso. Neste caso, foi usada uma máscara octal (700) para definir os privilégios. Contudo, também é possível usar o comando `chmod` de uma forma mais intuitiva para o utilizador, empregando mnemónicas em vez dos dígitos, na base octal.

```
$ sudo chmod ugo+x ficheiro1
```

Neste caso, pretende-se ativar a permissão de execução para os três grupos de utilizadores (dono, grupo e outros). Neste caso, o “a” em vez de “**ugo**” teria o mesmo efeito.

```
$ sudo chmod u+rx,g+rx,g-w,o+rx,o-w ficheiro1
```

Eis um exemplo, mais complexo onde se ativam e desativam permissões para o mesmo grupo de utilizadores.

[umask]

O comando **umask** é usado para definir ou obter as permissões por omissão relativas à criação de ficheiros no sistema de ficheiros. Apesar de empregar também a notação octal, este comando usa o complemento octal da permissão desejada, ou seja, deve-se negar a permissão que se pretende definir.

Permissões por omissão:

777 – para ficheiros executáveis

666 – para ficheiros de texto (a execução não faz sentido)

A permissão a calcular para a criação de um novo ficheiro executável é definida pela subtração do valor **umask** ao valor por omissão. Por exemplo, vamos supor que se definiu o seguinte valor **umask** e que se pretende criar um ficheiro de texto:

```
$ umask 022
```

Então, quando se criar o ficheiro este terá as permissões:

A máscara invertida 022 é o valor mais utilizado. Por norma, está definido no ficheiro **/etc/profile**, o que quer dizer, que é aplicado a todos os utilizadores.

Exercício 5

- a) Crie três utilizadores, com logins alunoA, alunoB e alunoC.
- b) Faça login com o utilizador alunoA, visualize o respetivo UID, GID e grupos secundários a que pertence o utilizador.
- c) Com o mesmo utilizador, crie um ficheiro de texto (fich1) e uma diretoria (dir1).
- d) Altere as permissões do ficheiro, fich1 para **rwX, r-- , ---**
- e) Crie o grupo “partilha” e associe o ficheiro, fich1 a este grupo.
- f) Que operações seriam necessárias para dar permissões de leitura ao utilizador alunoC (e apenas este) relativamente ao ficheiro, fich1?
- g) Altere o UID do utilizador alunoA para 4500.
- h) Remova do sistema o utilizador alunoB e todos os seus ficheiros.

3 Quotas

O sistema de quotas é um mecanismo que pode ser utilizado para controlar e/ou limitar a quantidade de dados que cada utilizador e/ou grupo de utilizadores podem armazenar nos nossos sistemas de ficheiros (ou partições). Estes controlos ou limitações serão definidos para as contas de utilizadores ou mesmo para grupos de utilizadores, que irão partilhar quotas num determinado sistema de ficheiros. Note-se que as quotas aplicam-se a cada um dos sistemas de ficheiros individualmente.

Se pretender efectuar restrições mais avançadas (RAM, CPU, LB, etc.) terá de utilizar o *accounting*.

Antes de iniciarmos a configuração e utilização de quotas poderá ser necessário instalar a aplicação de gestão de quotas:

```
$ sudo apt-get install quota
```

A ativação do sistema de quotas num sistema de ficheiros passa por vários passos que serão descritos e explicados de seguida:

1. Editar e acrescentar opções na linha correspondente ao sistema de ficheiros, existente no ficheiro **/etc/fstab**, no qual queremos implementar um sistema de quotas para utilizadores e/ou grupos de utilizadores.

```
#aspeto da linha referente à partição /dev/sda2 antes da alteração
/dev/sda2 /home ext3 defaults 0 2

#aspeto da linha referente à partição /dev/sda2 depois da alteração
/dev/sda2 /home ext3 defaults,
usrjquota=aquota.user,grpjquota=aquota.group,jqfmt=vfsv0 0 2
```

2. A fim de evitar a reiniciação do sistema, fazer remount do sistema de ficheiros por forma a atualizá-lo com as alterações efetuadas até ao momento.

```
$ sudo mount -o remount /home

# Alternativa
$ sudo mount -o
remount,usrjquota=aquota.user,grpjquota=aquota.group,jqfmt=vfsv0 /home
```

3. Inicializar/criar as bases de dados e respetivos índices para as quotas definidas na tabela de partições com o comando **quotacheck** (ver opções no manual do comando). Ao mesmo tempo serão criadas as bases de dados referentes às quotas do respetivos sistemas de ficheiros. Este comando permite também atualizar o estado atual das quotas.

```
$ sudo quotacheck -avugmc

# -c → cria ficheiros de quotas vazios
#verificar estado dos ficheiros com as BD das quotas da partição
$ ls -la /home/aquota.*
```

Se os ficheiros não forem criados, eventualmente terão de ser criados:

```
# criar os ficheiros
$ sudo touch /home/aquota.user /home/aquota.group

#verificar a criação dos ficheiros e que estão vazios
$ ls -la /home/aquota.*

#alterar permissões
$ sudo chmod 600 /home/aquota.user /home/aquota.group
```

Após a ativação do sistema de quotas no(s) sistema(s) de ficheiros, poderão ser utilizados os seguintes comandos para gestão de quotas de utilizadores e grupos de utilizadores:

- **edquota**: este comando permite definir/alterar/copiar quotas e respetivos parâmetros (ver manual do comando, especialmente as opções 'u', 'g', 't', 'p')

```
$ sudo edquota -u user1
Filesystem blocks soft hard inodes soft hard
```

```
/dev/sda2    6576    61440 81920 456      0      0
$ sudo edquota -g group1
```

- **setquota**: Uma alternativa ao editor de quota consiste em alterar os valores diretamente através da linha de comando, através do comando **setquota**.

Explicação dos parâmetros de configuração no comando **edquota**:

```
Filesystem: indica o filesystem onde serão aplicadas as quotas
Blocks: quantidade de blocos em Kilobytes que o utilizador está a utilizar
Soft: limite flexível a partir do qual o utilizador recebe avisos referente
ao uso de espaço em blocos(1024Bytes = 1KB cada)
Hard: limite máximo que o utilizador pode utilizar em disco para
armazenamento
Inodes: quantidade de inodes ou ficheiros que o utilizador pode ter
Soft: limite flexível relativo à quantidade de inodes ou ficheiros
Hard: limite máximo relativo à quantidade de inodes ou ficheiros

Nota: Se os parâmetros Soft ou Hard estiverem a '0'(zero) é porque não está
definido nenhum limite
```

- **quota**: este comando permite ver o estado das quotas de utilizadores e grupos (ver manual do comando, especialmente as opções '**u**', '**g**', '**q**', '**s**')

```
$ sudo quota -u user1
$ sudo quota -g group1
```

- **repquota**: este comando permite ver relatórios acerca do estado das quotas de utilizadores e grupos (ver manual do comando, especialmente as opções '**u**', '**g**', '**a**')

```
$ sudo repquota -uv /home
$ sudo repquota -av
```

- **quotaon** e **quotaoff**: estes comandos permitem ativar ou desativar quotas de utilizadores e grupos (ver manual dos comandos, especialmente as opções '**u**', '**g**', '**a**', '**v**')

```
#ativa as quotas de utilizadores e grupos apenas na diretoria /home
$ sudo quotaon -ugv /home

#ativa/desativa as quotas em todos os sistemas de ficheiros previamente
configurados
$ sudo quotaon -a
$ sudo quotaoff -a
```

- **quotacheck**: este comando permite examinar e compilar as bases de dados das quotas e deve ser executado com alguma regularidade pois é aconselhado a sua automatização através do

cron para ser executado todas as semanas (ver manual do comando, especialmente as opções '**u**', '**g**', '**a**', '**v**')

```
$ sudo quotacheck -a  
$ sudo quotacheck -u user1
```

Exercício 6:

- Crie um utilizador com o nome “teste” e configure a sua quota com 10MB de *Soft limit* e 13MB de *Hard limit*.
- Construa um *script* em Perl que receba por argumento de linha de comandos os *usernames* de dois utilizadores, copie as definições da quota do primeiro e as atribua ao segundo.
- Adapte o script anterior para que seja possível introduzir um ficheiro de nomes no segundo parâmetro.

Créditos

©2016-17: mario.antunes@ipleiria.pt

©2014-15: {mario.antunes, carlos.antunes, leonel.santos, miguel.frade, joana.costa, nuno.veiga}@ipleiria.pt

©2013-14: {carlos.antunes, leonel.santos, gustavo.reis, miguel.frade, joana.costa, mario.antunes}@ipleiria.pt

©2013: {carlos.antunes, mario.antunes}@ipleiria.pt

©2012: {carlos.antunes, miguel.frade, mario.antunes, paulo.loureiro}@estg.ipleiria.pt