SEG12 - Atividades - Semana 1

Francisco Marcelo, Marcelo Karam e Felipe Scarel

06-08-2018

Introdução ao sistema operacional Linux

1) Identificando bits de permissão

1. Verifique as permissões do diretório `/tmp`. O que você percebe de diferente em relação às permissões de *outros*?

   ```
   $ ls -lha / | grep 'tmp$'
   drwxrwxrwt  7 root root 4,0K Ago  7 01:01 tmp
   ```

   O sticky bit está definido: `t`.

2. Considerando que há permissão de escrita no diretório para todos, o que o impediria de remover um arquivo de outra pessoa?

   ```
   $ rm -f /tmp/file_root
   rm: não foi possível remover “/tmp/file_root”: Operação não permitida
   ```

   Com o sticky bit definido somente o dono de um arquivo pode removê-lo.

2) Identificando e entendendo *hard links*

O número de *links* (*link counter*) que apontam para um arquivo é mantido em seu *inode*. Esse contador é utilizado pelo sistema para controlar a liberação dos blocos do disco alocados ao arquivo quando o contador atingir o valor zero ,ou seja, quando nenhum outro arquivo estiver apontando para o *inode*.

1. Qual o número de *links* do seu diretório *home*?

   ```
   $ ls -lha /home/ | egrep ' aluno$'
   drwxr-xr-x  2 aluno  aluno  4,0K Ago  7 01:45 aluno
   ```

   Como visto acima, `2`. Esse número não é fixo, mas depende do conteúdo do diretório. Um diretório recém criado, que não tenha nenhum conteúdo possui dois *links* (um referente ao próprio diretório e outro referente à entrada especial ".").

2. Crie o arquivo `arqses1ex3` no seu diretório *home*. Utilize o comando *touch*.

   ```
   $ touch ~/arqses1ex3
   $ ls /home/aluno
   arqses1ex3
   ```

3. Verifique o número de *links* do arquivo `arqses1ex3` e anote o resultado. Você pode utilizar o

redirecionamento de saída para registrar esse resultado no próprio arquivo criado. Essa informação será necessária para uma atividade posterior.

```
$ mytemp=$(mktemp) && ls -lha ~/arqses1ex3 | tee nlinks && awk '{print $2}' nlinks
> $mytemp && mv $mytemp nlinks
-rw-r--r-- 1 aluno aluno 0 Ago  7 01:52 /home/aluno/arqses1ex3
$ cat nlinks
1
```

O arquivo `arqses1ex3` possui apenas um link.

4. Verifique se mudou o número de *links* do seu diretório *home*.

```
$ ls -lha /home/ | egrep ' aluno$'
drwxr-xr-x  2 aluno  aluno  4,0K Ago  7 02:05 aluno
```

O número de *links* continuou o mesmo.

5. Crie um diretório com o nome de `dirses1ex3`, também no seu diretório *home*.

```
$ mkdir /home/aluno/dirses1ex3
$ ls ~
arqses1ex3  dirses1ex3  nlinks
```

6. Mais uma vez, verifique o número de *links* do seu diretório *home*. Ele mudou? Você saberia dizer por quê?

```
$ ls -lha /home/ | egrep ' aluno$'
drwxr-xr-x  3 aluno  aluno  4,0K Ago  7 02:11 aluno
```

O número de *links* aumentou em uma unidade, por conta de entrada especial ".." presente no diretório `/home/aluno/dirses1ex3`, que aponta para o diretório `/home/aluno`.

7. Qual o número de links do diretório `dirses1ex3`?

```
$ ls -lha ~ | egrep ' dirses1ex3$'
drwxr-xr-x 2 aluno aluno 4,0K Ago  7 02:11 dirses1ex3
```

Como visto acima, `2`.

8. Verifique qual opção deve ser passada ao comando `ls` para que ele liste as informações do diretório `dirses1ex3` e não o seu conteúdo.

```
$ ls -dl ~/dirses1ex3/
drwxr-xr-x 2 aluno aluno 4096 Ago  7 02:11 /home/aluno/dirses1ex3/
```

Devem ser passadas as opções `-d` e `-l`.

9. Você saberia explicar por que o número de *links* do diretório `dirses1ex3` é maior que um?

   Os dois *links* são relativos ao próprio diretório. Um aponta o caminho direto `/home/aluno` → `/home/aluno/dirses1ex3` e o outro corresponde à entrada especial ".", presente no próprio diretório `/home/aluno/dirses1ex3`.

3) Conhecendo diferenças entre *hard link* e *symbolic link*

Foi explicada a importância dos *links* criados com o comando `ln`. Para criar um *symbolic link*, a opção `-s` deve ser informada na linha de comando. Consulte as páginas do manual para conhecer outras opções.

1. No seu diretório de trabalho, crie um *hard link* para o arquivo `arqses1ex3`. O nome do arquivo criado deverá ser `hosts.hard`.

```
$ ln /home/aluno/arqses1ex3 /home/aluno/hosts.hard
$ ls ~
arqses1ex3  dirses1ex3  hosts.hard  nlinks
```

2. Verifique agora o número de links do arquivo `arqses1ex3` e compare com aquele obtido na atividade 2. Explique a diferença.

```
$ ls -lha /home/aluno/arqses1ex3 | awk '{print $2}'
2
$ cat nlinks
1
```

   O número de *links* foi aumentado de 1 para 2 devido à criação do *link* `hosts.hard`.

3. Crie um *symbolic link* para o arquivo `arqses1ex3`, que deverá se chamar `hosts.symbolic`.

```
$ ln -s /home/aluno/arqses1ex3 /home/aluno/hosts.symbolic
$ ls
arqses1ex3  dirses1ex3  hosts.hard  hosts.symbolic  nlinks
```

4. O número de *links* do arquivo `arqses1ex3` aumentou?

```
$ ls -lha /home/aluno/arqses1ex3
-rw-r--r-- 2 aluno aluno 0 Ago  7 01:52 /home/aluno/arqses1ex3
```

Não, não aumentou.

5. Caso não tenha aumentado, por que isso aconteceu, considerando que foi criado um *link* para ele?

   Porque o *symbolic link* aponta para outro *inode*.

6. Qual o tamanho do arquivo *hosts.symbolic*?

   ```
   $ du -sb ~/hosts.symbolic
   22      /home/aluno/hosts.symbolic
   ```

   Como mostrado acima, 22 bytes.

7. Você percebe alguma correlação entre o tamanho e o arquivo para o qual ele aponta?

   ```
   $ ls -d /home/aluno/arqses1ex3 | tr -d '\n' | wc -c
   22
   ```

   Esse tamanho representa o número de caracteres presentes no *path* completo do arquivo original linkado, sendo cada caractere representado por 1 byte.

4) Trabalhando com *hard link* e *symbolic link*

1. Se o arquivo original `arqses1ex3` fosse removido, o que aconteceria se tentássemos acessá-lo pelo *hard link*? E pelo *symbolic link*?

   Pelo *hard link* conseguiríamos acessar o conteúdo do arquivo normalmente. Já pelo *symbolic link* não conseguiríamos acessar o conteúdo do arquivo, uma vez que o mesmo é somente uma referência para o arquivo original.

2. Depois de responder a essas questões, remova o arquivo criado (`arqses1ex3`) e verifique se as suas respostas estão corretas.

```
$ rm arqses1ex3

$ ls -l hosts.hard
-rw-r--r-- 1 aluno aluno 0 Ago  7 01:52 hosts.hard
$ ls -l hosts.symbolic
lrwxrwxrwx 1 aluno aluno 22 Ago  7 02:38 hosts.symbolic -> /home/aluno/arqses1ex3

$ cat hosts.hard
$ cat hosts.symbolic
cat: hosts.symbolic: Arquivo ou diretório não encontrado
```

As respostas acima estão corretas.

5) Conhecendo algumas limitações do *hard link*

1. Crie um arquivo chamado `arqses1ex6`. Em seguida, crie um *hard link* para esse arquivo com o nome `link-arqses1ex6` no diretório `/tmp`. O que aconteceu? Por quê? Como resolver esse problema?

Para que esta atividade tenha efeito, o diretório `/tmp` deverá ter sido criado numa partição diferente da partição onde se encontra o *home* do usuário. Caso essa situação não ocorra, verifique se existe o diretório `/var/tmp` e veja se ele está em outra partição. Se for o caso, use este último para fazer o exercício.

```
$ touch ~/arqses1ex6
$ ln ~/arqses1ex6 /tmp/link-arqses1ex6
ln: failed to create hard link “/tmp/link-arqses1ex6” => “/home/aluno/arqses1ex6”:
Link entre dispositivos inválido

$ df -h | sed -n '1!p' | egrep -v '^tmpfs |^udev ' | awk '{printf "%s\t mounted on:
%s\n", $6, $1}'
/        mounted on: /dev/sda1
/tmp     mounted on: /dev/sda6
```

Não foi possível criar o *hard link*, porque o diretório `/tmp` está em outra partição.

6) Criando *links* para diretórios

Crie, no seu diretório *home*, um *link* simbólico para o diretório `/usr/bin` com o nome de `link-bin`. Com o *link* criado, execute o seguinte:

1. Mude para o diretório `link-bin`.

```
$ ln -s /usr/bin /home/aluno/link-bin ; cd link-bin
$ pwd
/home/aluno/link-bin
```

2. Agora, vá para o diretório pai (utilize a notação ".."). Você saberia explicar por que se encontra no seu diretório *home* e não no diretório `/usr`?

```
$ cd ..
$ pwd
/home/aluno
```

   Porque o *link* simbólico é apenas uma referência para o diretório.

7) Alterando permissões de arquivos e diretórios

O comando `chmod` é utilizado para modificar as permissões de um arquivo. Utilizando a notação octal, execute a seguinte sequência:

1. Modifique a permissão do seu diretório *home* de modo a retirar a permissão de escrita do seu dono.

```
$ chmod 555 /home/aluno
$ ls -ld /home/aluno
dr-xr-xr-x 3 aluno aluno 4096 Ago  7 03:38 /home/aluno
```

2. Verifique as permissões associadas ao arquivo `arqses1ex6`. Você tem permissão para escrever nesse arquivo? O grupo tem?

```
$ ls -lha ~/arqses1ex6
-rw-r--r-- 1 aluno aluno 0 Ago  7 02:55 /home/aluno/arqses1ex6
```

   Somente o dono do arquivo tem permissão para escrever no mesmo.

3. Tente remover o arquivo `arqses1ex6`. Você conseguiu? Em caso negativo, você sabe explicar o motivo?

```
$ rm ~/arqses1ex6
rm: não foi possível remover “/home/aluno/arqses1ex6”: Permissão negada
```

   Não, porque o diretório `/home/aluno` está sem permissão de escrita para o dono.

4. Modifique as permissões do arquivo `arqses1ex6` de forma a retirar a permissão de escrita para o dono e colocá-la para o grupo.

```
$ chmod 464 ~/arqses1ex6
$ ls -ld ~/arqses1ex6
-r--rw-r-- 1 aluno aluno 0 Ago  7 02:55 /home/aluno/arqses1ex6
```

5. Com o uso de redirecionamento, tente copiar o conteúdo do seu diretório *home* para dentro do arquivo `arqses1ex6`.

```
$ ls -lha /home/aluno > /home/aluno/arqses1ex6
-bash: /home/aluno/arqses1ex6: Permissão negada
```

   Apresentou erro de permissão de gravação no diretório por parte do dono.

6. Torne a colocar a permissão para escrita no seu diretório *home* para o dono.

```
$ chmod 755 /home/aluno
$ ls -ld ~
drwxr-xr-x 3 aluno aluno 4096 Ago  7 03:38 /home/aluno
```

8) Atribuindo as permissões padrão

1. Crie arquivos (`arq1ses1ex9`, `arq2ses1ex9`, etc.) e diretórios (`dir1ses1ex9`, `dir2ses1ex9`, etc.) em seu diretório *home*, após definir cada uma das seguintes *umasks*: `000`; `002`; `003`; `023`; `222`; `022`. Em seguida, observe as permissões que foram associadas a cada um dos arquivos e diretórios.

```
$ umask 000 ; touch arq1ses1ex9 ; mkdir dir1ses1ex9
$ umask 002 ; touch arq2ses1ex9 ; mkdir dir2ses1ex9
$ umask 003 ; touch arq3ses1ex9 ; mkdir dir3ses1ex9
$ umask 023 ; touch arq4ses1ex9 ; mkdir dir4ses1ex9
$ umask 222 ; touch arq5ses1ex9 ; mkdir dir5ses1ex9
$ umask 022 ; touch arq6ses1ex9 ; mkdir dir6ses1ex9

$ ls -lha /home/aluno | egrep 'arq[1-6]ses1ex9|dir[1-6]ses1ex9'
-rw-rw-rw- 1 aluno aluno    0 Ago  7 03:50 arq1ses1ex9
-rw-rw-r-- 1 aluno aluno    0 Ago  7 03:50 arq2ses1ex9
-rw-rw-r-- 1 aluno aluno    0 Ago  7 03:50 arq3ses1ex9
-rw-r--r-- 1 aluno aluno    0 Ago  7 03:52 arq4ses1ex9
-r--r--r-- 1 aluno aluno    0 Ago  7 03:52 arq5ses1ex9
-rw-r--r-- 1 aluno aluno    0 Ago  7 03:52 arq6ses1ex9
drwxrwxrwx 2 aluno aluno 4,0K Ago  7 03:50 dir1ses1ex9
drwxrwxr-x 2 aluno aluno 4,0K Ago  7 03:50 dir2ses1ex9
drwxrwxr-- 2 aluno aluno 4,0K Ago  7 03:50 dir3ses1ex9
drwxr-xr-- 2 aluno aluno 4,0K Ago  7 03:52 dir4ses1ex9
dr-xr-xr-x 2 aluno aluno 4,0K Ago  7 03:52 dir5ses1ex9
drwxr-xr-x 2 aluno aluno 4,0K Ago  7 03:52 dir6ses1ex9
```

9) Entendendo as permissões padrões

1. Na execução do exercício anterior, você saberia explicar por que, ainda que utilizando a mesma *umask*, as permissões associadas ao arquivo criado diferem das do diretório?

   O comando `umask` trabalha de forma diferente com arquivos e diretórios. Por motivos de segurança um um novo arquivo nunca recebe a permissão de execução quando da sua criação.

Usuários e grupos

1) Criando contas de usuários

Uma das atividades que fazem parte da rotina diária de um administrador de sistemas é o gerenciamento de contas de usuários. Frequentemente, usuários são criados, modificados, desabilitados ou excluídos do sistema.

1. Descubra se o sistema faz uso de *shadow passwords* ou se ainda utiliza o esquema tradicional.

   ```
   $ ls -ld /etc/gshadow /etc/shadow
   -rw-r----- 1 root shadow  666 Ago  5 16:52 /etc/gshadow
   -rw-r----- 1 root shadow 1125 Ago  5 16:51 /etc/shadow
   ```

   O aluno deve verificar se os arquivos `/etc/shadow` e `/etc/gshadow` existem.

2. Crie uma conta para você no sistema, seguindo os passos descritos na aula teórica e no material didático.
   - Editar o arquivo `/etc/group` e inserir uma nova linha com os parâmetros relativos ao grupo do novo usuário:
     - Nome do grupo;
     - Senha ("x");
     - GID;
     - Membros do grupo.

     ```
     marcelo:x:1001:
     ```

   - Editar o arquivo `/etc/gshadow` e inserir uma nova linha com os parâmetros relativos ao grupo do novo usuário:
     - Nome do grupo;
     - Senha criptografada do grupo ("!");
     - Administradores do grupo;
     - Membros do grupo.

     ```
     marcelo:!::
     ```

   - Editar o arquivo `/etc/passwd` e inserir uma nova linha com os parâmetros relativos à conta do novo usuário:
     - Nome do usuário;
     - Senha ("x");

    - UID;
    - GID;
    - GECOS: campo com comentários informativos do usuário;
    - Diretório *home*;
    - Shell de login.

```
marcelo:x:1001:1001:,,,:/home/marcelo:/bin/bash
```

- Editar o arquivo `/etc/shadow` e inserir uma nova linha os parâmetros relativos à conta do novo usuário:
    - Nome do usuário;
    - Senha criptografada: inserir valor "*", que será alterado a seguir;
    - *last_change*: número de dias desde a última alteração de senha;
    - *minimum*: número mínimo de dias até que senha possa ser alterada novamente;
    - *maximum*: número máximo de dias até que a senha deva ser alterada;
    - *warning*: número de dias para aviso de expiração de senha;
    - *inactive*: número de dias após expiração em que a senha será aceita;
    - *expire*: data para expiração da senha.

```
marcelo:*:16846:0:99999:7:::
```

- Definir uma senha para a nova conta, utilizando o comando `passwd`:

```
# passwd marcelo
```

- Copiar os arquivos de inicialização contidos no diretório `/etc/skel` para o diretório *home* do usuário.

```
# cp -r /etc/skel /home/marcelo
```

- Alterar o usuário e grupo donos dos arquivos na pasta *home* do novo usuário:

```
# chown -R marcelo.marcelo /home/marcelo
```

- Configurar a *quota* de disco para o usuário, se o sistema utilizar *quotas*.
- Testar se a conta foi criada corretamente, fazendo login no sistema e verificando se o diretório corrente é o diretório *home* do usuário, definido no arquivo `/etc/passwd`.
- O *script shell* abaixo mostra uma maneira como os comandos executados manualmente

nesta atividade poderiam ser automatizados por um administrador de sistemas:

```
#!/bin/bash

usage() {
  echo "  Usage: $0 -u USER -p PASSWORD"
  exit 1
}


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

while getopts ":u:p:" opt; do
  case "$opt" in
    u)
      user=${OPTARG}
      ;;
    p)
      pass=${OPTARG}
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $user ] && { echo "  [*] No user?"; usage; }
[ -z $pass ] && { echo "  [*] No password?"; usage; }

if egrep "^${user}:" /etc/passwd &> /dev/null; then
  echo "  [*] User exists!"
  exit 1
fi

lastgid=$( getent group | grep -v 'nogroup' | cut -d':' -f3 | sort -n | tail -n1
)
((lastgid++))

echo "$user:x:$lastgid:" >> /etc/group
echo "$user:!::" >> /etc/gshadow

lastuid=$( getent passwd | grep -v 'nobody' | cut -d':' -f3 | sort -n | tail -n1
)
((lastuid++))

echo "$user:x:$lastuid:$lastgid:,,,:/home/$user:/bin/bash" >> /etc/passwd
```

```
salt="$( cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 8 | head -n 1 )"
hpass="$( mkpasswd -m sha-512 -S $salt -s <<< $pass )"
echo "$user:$hpass:16842:0:99999:7:::" >> /etc/shadow

cp -r /etc/skel /home/$user
chown -R ${user}.${user} /home/$user
```

3. Agora, crie uma conta para o instrutor, utilizando, desta vez, o comando `useradd`. Faça com que a conta criada tenha sete dias de duração e com que o seu diretório de trabalho seja `/NOME`, onde `NOME` é o nome de usuário para o qual a conta deve ser aberta.

Consulte a página de manual do comando `useradd` e procure as informações necessárias para incluir a data de expiração (*expire date*) e criar o diretório de trabalho (*homedir*) em um local diferente do padrão, que é `/home/NOME`. Ainda, não se deve esquecer de escolher e atribuir uma senha para as contas que obedeça aos padrões de segurança apresentados no texto. Observe, ainda, que o diretório *home* não é criado automaticamente pelo comando `useradd`.

```
# useradd instrutor -d /instrutor -m -e 2018-08-07
# passwd instrutor
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: senha atualizada com sucesso
```

Ao usar o comando `useradd`, o shell escolhido pelo sistema é o `/bin/sh`, por padrão. Para alterar o shell do usuário, pode-se editar o arquivo `/etc/passwd` diretamente, ou executar o comando `chsh`, mostrado abaixo:

```
# getent passwd | grep '^instrutor:'
instrutor:x:1002:1002::/home/instrutor:/bin/sh

# chsh instrutor
Mudando o shell de login para instrutor
Informe o novo valor ou pressione ENTER para aceitar o padrão
        Shell de Login [/bin/sh]: /bin/bash

# getent passwd | grep '^instrutor:'
instrutor:x:1002:1002::/home/instrutor:/bin/bash
```

4. O comando `useradd` não é uma boa opção para informar a senha do usuário. Por quê?

   Porque a senha criptografada deve ser digitada diretamente na linha de comando, podendo ser lida posteriormente via logs ou histórico do shell.

5. Faça um *script* que simule o comando `newusers`. Para isso, você deve criar um arquivo texto contendo as informações a respeito dos usuários, mantendo o mesmo padrão dos arquivos lidos pelo comando `newusers` (para descobrir o formato, consulte a página de manual: `$ man 8`

`newusers`). Como este arquivo conterá as senhas dos usuários, é importante removê-lo logo após a criação das contas.

Utilize a variável de sistema `IFS` (*Internal Field Separator*) em seu *script* para definir o caractere ":" como campo que separa as informações sobre as contas.

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

```
#!/bin/bash

IFS=':'
useradd="$( which useradd )"
groupadd="$( which groupadd )"


usage() {
  echo "  Usage: $0 -f NEWUSERS_FILE"
  echo "  File syntax: username:password:uid:gid:gecos:homedir:shell"
  exit 1
}


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

while getopts ":f:" opt; do
  case "$opt" in
    f)
      file=${OPTARG}
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $file ] && { echo "  [*] No file?"; usage; }

while read username password uid gid gecos homedir shell; do
  if egrep "^${username}:" /etc/passwd &> /dev/null; then
    echo "  [*] User $username already exists, skipping..."
  elif getent passwd | cut -d':' -f3 | grep "$uid" &> /dev/null; then
    echo "  [*] UID $uid already exists, skipping..."
  elif getent group | cut -d':' -f3 | grep "$gid" &> /dev/null; then
    echo "  [*] GID $gid already exists, skipping..."
  else
    hpass="$( mkpasswd -m sha-512 -s <<< $pass )"
    $groupadd $username -g $gid
    $useradd $username -p $( mkpasswd -m sha-512 -s <<< $password)  -u $uid -g $gid
-c "$gecos" -d $homedir -s $shell
    cp -r /etc/skel $homedir
    chown -R $username:$username $homedir
  fi
done < "$file"
```

Um arquivo de entrada com sintaxe válida para o *script* acima seria como se segue:

```
usuario1:rnpesr:1101:1101::/home/usuario1:/bin/bash
usuario2:rnpesr:1102:1102::/home/usuario2:/bin/bash
usuario3:rnpesr:1103:1103::/home/usuario3:/bin/bash
```

2) Verificando e modificando informações de contas de usuário

Após a criação de uma conta, é fundamental que o administrador verifique se ela foi criada corretamente.

1. Entre no sistema com o usuário criado no item 3 da atividade 1 e execute os comandos indicados para verificação de uma conta.

```
$ ssh instrutor@localhost
instrutor@localhost's password:

$ id
uid=1002(instrutor) gid=1002(instrutor) grupos=1002(instrutor)
$ pwd
/instrutor
$ ls -la
total 8
drwxr-xr-x  2 instrutor instrutor 4096 Ago  7 14:42 .
drwxr-xr-x 23 root      root      4096 Ago  7 14:42 ..
```

2. Seria possível inserir o número de telefone de trabalho desse mesmo usuário, junto com a informação de quem ele é? Faça isso e torne a checar se a sua mudança surtiu efeito.

```
# chfn -w 6198765432 instrutor
# finger -l instrutor
Login: instrutor                        Name:
Directory: /instrutor                   Shell: /bin/sh
Office Phone: 619-876-5432
Last login Tue Aug  7 14:44 (-03) on pts/1 from localhost
No mail.
No Plan.
```

3) Criando grupos de usuários

O recurso de grupos de usuários é muito útil para compartilhar informações. No momento em que a conta `instrutor` foi criada, no item 3 da atividade 1 deste roteiro, o grupo primário ficou sendo o seu próprio nome de usuário. Isso ocorre sempre que não é atribuído um valor para o grupo primário, no momento da criação de um novo usuário. Como o usuário criado não faz parte de outro grupo, a não ser do seu próprio, ele somente poderá acessar seus arquivos ou aqueles

arquivos para os quais haja permissão de acesso para outros usuários.

1. Use o comando apropriado para criar um grupo chamado `grupoteste`.

```
# addgroup grupoteste
Adicionando grupo 'grupoteste' (GID 1003) ...
Concluído.
```

2. Liste o arquivo `/etc/group` e anote o `GID` que foi atribuído ao grupo criado.

```
# getent group | egrep '^grupoteste:' | cut -d':' -f3
1003
```

3. Aproveite para observar, no arquivo `/etc/group`, quais são os outros grupos existentes no sistema. Qual o grupo associado ao usuário `root`?

```
# getent group | grep root
root:x:0:
```

   O grupo `root`, que é o grupo primário do superusuário do sistema.

4. Altere o grupo primário do usuário `instrutor`, de modo que este passe a ser o grupo criado no item 1 da atividade 3, `grupoteste`.

```
# usermod -g grupoteste instrutor
# getent passwd | egrep '^instrutor:'
instrutor:x:1002:1003:,,6198765432,:/instrutor:/bin/sh
```

5. Se autentique no sistema utilizando a sua conta e inclua seu usuário como administrador do grupo `grupoteste`. Em seguida inclua o usuário `instrutor` no grupo `grupoteste`. Você conseguiu executar as tarefas propostas? Por quê? Como você deve fazer para realizar as tarefas?

```
$ gpasswd -a instrutor grupoteste
gpasswd : Permissão negada.
```

   Não, porque somente o usuário `root` pode cadastrar administradores em um grupo. Os comandos para viabilizar essa tarefa seriam:

```
# gpasswd -A aluno grupoteste
# logout

$ whoami
aluno
$ gpasswd -a instrutor grupoteste
Adicionando usuário instrutor ao grupo grupoteste
```

6. Altere novamente o grupo primário do usuário `instrutor` para o grupo `instrutor`.

```
# usermod -g instrutor instrutor
# getent passwd | egrep '^instrutor:'
instrutor:x:1002:1002:,,6198765432,:/instrutor:/bin/sh
```

4) Incluindo usuários em grupos secundários

1. Editando o arquivo `/etc/group`, inclua, no grupo `grupoteste`, o usuário criado no terceiro item da atividade 1 desse roteiro (`instrutor`). Note que o grupo primário do usuário não deve mudar; continua sendo o nome do usuário.

   Inserir após o último caractere ":" na linha referente ao grupo `grupoteste`, o *username* do usuário `instrutor`.

```
# getent group | egrep '^grupoteste:'
grupoteste:x:1003:instrutor
# groups instrutor
instrutor : instrutor grupoteste
```

2. Agora, utilize um comando apropriado para inserir nesse mesmo grupo o usuário criado para você no primeiro item da atividade 1.

```
# groups marcelo
marcelo : marcelo

# usermod -a -G grupoteste marcelo
# groups marcelo
marcelo : marcelo grupoteste
```

5) Bloqueando contas de usuários

No Linux, é possível impedir temporariamente o acesso ao sistema mesmo que o usuário esteja utilizando uma conta com acesso liberado a este.

1. Utilizando um comando apropriado, bloqueie a conta criada para o instrutor e teste se obteve

sucesso no bloqueio.

```
# passwd -l instrutor
passwd: informação de expiração de senha alterada.

# ssh instrutor@localhost
instrutor@localhost's password:
Permission denied, please try again.
```

2. Agora desbloqueie a conta e faça o teste de acesso para verificar se sua alteração surtiu efeito.

```
# passwd -u instrutor
passwd: informação de expiração de senha alterada.

# ssh instrutor@localhost
instrutor@localhost's password:
$ pwd
/instrutor
```

Também pode-se utilizar o comando `# usermod -U USERNAME` para atingir o mesmo objetivo.

6) Removendo uma conta de usuário manualmente

No Linux, é possível executar uma mesma tarefa de diversas maneiras. Para um administrador de sistemas, é importante conhecer essas alternativas, porque elas podem ser úteis em situações específicas em que não seja possível utilizar um dado recurso ou ferramenta do sistema.

1. Sem utilizar o comando `userdel`, remova a conta criada para você no segundo item da atividade 1.

   Em ordem, deve-se executar as atividades espelho das que foram feitas anteriormente, quais sejam:

   - Remover entradas referente à conta nos arquivos:
     - `/etc/group`
     - `/etc/gshadow`
     - `/etc/passwd`
     - `/etc/shadow`
   - Remover o diretório *home* do usuário;
   - Remover as configurações de *quota*, caso tenham sido configuradas anteriormente.
   - O *script shell* abaixo mostra uma maneira como os comandos executados manualmente nesta atividade poderiam ser automatizados por um administrador de sistemas:

```
#!/bin/bash

BACKUP_DIR="/root/user_backups"

usage() {
  echo "  Usage: $0 -u USER [-b]"
  echo "  Use [-b] to backup user dir to /root before deletion."
  exit 1
}


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

backup=false
while getopts ":u:b" opt; do
  case "$opt" in
    u)
      user=${OPTARG}
      ;;
    b)
      backup=true
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $user ] && { echo "  [*] No user?"; usage; }

if ! egrep "^${user}:" /etc/passwd &> /dev/null; then
  echo "  [*] User does not exist!"
  exit 1
fi

homedir=$( getent passwd | egrep "^$user:" | cut -d':' -f6 )

if $backup; then
  [ ! -d $BACKUP_DIR ] && mkdir $BACKUP_DIR
  tar czf $BACKUP_DIR/${user}.tar.gz $homedir
fi
rm -rf /home/$user

sed -i "/^$user:/d" /etc/group
sed -i "/^$user:/d" /etc/gshadow
sed -i "/^$user:/d" /etc/passwd
sed -i "/^$user:/d" /etc/shadow
```

```
# remove user from secondary groups
sed -r -i "s/,?${user},?/,/ ; s/:,/:/ ; s/,$//" /etc/group
```

2. Certifique-se de que esse usuário foi realmente excluído do sistema, utilizando um dos comandos que fornecem informações sobre os usuários.

```
# finger marcelo
finger: marcelo: no such user.
```

3. Faça um backup de seus dados de modo que o instrutor possa ter sobre eles o mesmo tipo de acesso que você.

   O *script* apontado no primeiro item desta atividade já faz o backup de arquivos (via opção `-b`). Caso o usuário tenha sido removido sem que seu *home* tenha sido apagado (por exemplo, via comando `userdel`), pode-se fazer o backup dos dados da seguinte forma:

```
# tar czf /instrutor/marcelo.tar.gz /home/marcelo && rm -rf /home/marcelo
tar: Removendo `/' inicial dos nomes dos membros
# ls /instrutor/
marcelo.tar.gz
```

7) Obtendo informações sobre usuários

Muitas vezes, é necessário obter informações sobre os usuários de um sistema. Dois comandos que fornecem informações sobre usuários são `finger` e `id`.

1. Verifique os parâmetros do usuário criado na atividade 1 utilizando esses comandos, e descreva a diferença entre os dois a partir dos resultados obtidos. Consulte as páginas de manual para verificar as opções disponíveis nestes comandos.

```
$ id instrutor
uid=1002(instrutor) gid=1002(instrutor) grupos=1002(instrutor),1003(grupoteste)

$ finger instrutor
Login: instrutor                        Name:
Directory: /instrutor                   Shell: /bin/sh
Office Phone: 619-876-5432
Last login Tue Aug  7 15:45 (-03) on pts/1 from localhost
No mail.
No Plan.
```

   O comando `id` mostra os grupos do usuário e seu UID enquanto o comando `finger` mostra informações como: diretório *home*, shell, *username*, GECOS, terminal utilizado pelo usuário, etc.

8) Removendo contas de usuários

1. Utilizando os comandos apropriados, remova a conta criada para o instrutor. Não se esqueça de que um grupo foi especialmente criado para ele e que ele também possui um grupo secundário.

```
# userdel -r instrutor
# getent passwd | egrep '^instrutor:'
# getent group  | egrep ',?instrutor,?'
#
```

9) Alterando o grupo a que um arquivo pertence

O arquivo `/etc/passwd` contém informações importantes sobre os usuários do sistema. Esse arquivo pertence ao usuário `root` e ao grupo `root`. As permissões de acesso desse arquivo definem que ele só poderá ser modificado pelo usuário `root`.

1. Faça com que esse arquivo pertença ao grupo `grupoteste`, criado na atividade 3. Com isso, os usuários desse grupo, incluindo o usuário criado na atividade 1 poderão acessar esse arquivo por meio das permissões definidas para os usuários do grupo.

```
# chgrp grupoteste /etc/passwd
# ls -ld /etc/passwd
-rw-r--r-- 1 root grupoteste 1612 Ago  7 16:12 /etc/passwd
```

10) Alterando permissões de acesso de arquivos

É muito comum o administrador ter que modificar a permissão de arquivos para possibilitar ou impedir que eles sejam lidos ou modificados por diferentes categorias de usuários. A melhor forma de fazer isso é utilizando o comando `chmod`.

1. O arquivo `/etc/passwd` tem apenas permissão de leitura para os usuários do seu grupo proprietário. Use o comando `chmod` para atribuir permissão de escrita ao grupo proprietário desse arquivo. A permissão de escrita nesse arquivo é inicialmente atribuída apenas ao usuário proprietário do arquivo.

```
# chmod 664 /etc/passwd
# ls -ld /etc/passwd
-rw-rw-r-- 1 root grupoteste 1612 Ago  7 16:12 /etc/passwd
```

   Alternativamente, pode-se usar também o comando `# chmod g+w /etc/passwd` para atingir o mesmo objetivo.

2. O setor de controladoria de uma empresa só possuía um funcionário, que pediu demissão. Como não há um diretório específico para armazenar os arquivos do setor, todos os seus

arquivos de trabalho estão armazenados em seu diretório *home*. Que passos você deve fazer para disponibilizar estes arquivos para o novo funcionário que será contratado e para que este tipo de problema não volte a ocorrer?

- Crie o grupo controladoria:

```
# addgroup controladoria
Adicionando grupo 'controladoria' (GID 1002) ...
Concluído.
```

- Crie a conta do novo funcionário e defina o grupo `controladoria` como seu grupo primário:

```
# useradd -m -g controladoria funcionario
# ls -lha /home/ | egrep ' funcionario$'
drwxr-xr-x  2 funcionario controladoria 4,0K Ago  7 16:22 funcionario
```

- Crie o diretório `/home/controladoria`:

```
# mkdir /home/controladoria
# chgrp controladoria /home/controladoria
# chmod g+w /home/controladoria/
# ls -lha /home/ | egrep ' controladoria$'
drwxrwxr-x  2 root        controladoria 4,0K Ago  7 16:24 controladoria
```

- Habilite o *sticky bit* para o diretório `/home/controladoria`, de forma que todos os membros do grupo `controladoria` possam criar arquivos ali, mas apenas o dono de cada arquivo possa apagá-los:

```
# chmod +t /home/controladoria/
# ls -lha /home/ | egrep ' controladoria$'
drwxrwxr-t  2 root        controladoria 4,0K Ago  7 16:24 controladoria
```

- Mova os arquivos do antigo funcionário para o diretório `/home/controladoria`:

```
# cp -a /home/antigo_funcionario /home/controladoria
# ls /home/controladoria
antigo_funcionario
```

  Redefina as permissões dos arquivos do antigo funcionário:

```
# chown -R root.controladoria /home/controladoria
```

- Remova a conta do antigo funcionário:

```
# userdel -r antigo_funcionario
```

- Oriente o novo funcionário para que ele só armazene os arquivos relacionados ao setor de controladoria no diretório `/home/controladoria`, e seus arquivos pessoais em `/home/funcionario`.

Por motivos de segurança, ao final das atividades, retorne a permissão e o grupo do arquivo `/etc/passwd` para os valores originais.

```
# chown root.root /etc/passwd
# chmod 644 /etc/passwd
# ls -lh /etc/passwd
-rw-r--r-- 1 root root 1,7K Ago  7 16:22 /etc/passwd
```

Processos

1) Descobrindo o número de processos em execução

1. Quantos processos estão sendo executados na máquina no momento? Use o comando `wc` para contá-los.

```
# ps aux | sed -n '1!p' | wc -l
71
```

2. Faça um *script* que liste o número de processo que cada usuário está executando.

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

```
#!/bin/bash

users=( $( ps aux | awk '{ if (NR>1) print $1 }' | sort | uniq ) )

for (( i=0; i<${#users[@]}; i++ )); do
  nproc=$( ps aux | grep "${users[$i]}" | wc -l )
  echo "User ${users[$i]} has $nproc active processes"
done
```

2) Descobrindo o PID e o PPID de um processo

1. Quais os valores de `PID` e `PPID` do shell que você está utilizando no sistema?

```
$ echo -e "PID: $$\nPPID: $PPID"
PID: 1016
PPID: 1015
```

2. Faça um *script* que liste todos os processos que foram iniciados pelo processo `init`. A lista não deve conter mais de uma ocorrência do mesmo processo.

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

```
#!/bin/bash

pinit=( $( ps -eo ppid,comm | egrep -e "^ *1 " | sort | uniq | awk {'print $2'} ) )
pinit_count=${#pinit[@]}

echo "$pinit_count processes started by init (1):"

for (( i=0; i<$pinit_count; i++ )); do
  echo "  ${pinit[$i]}"
done
```

3) Estados dos processos

1. Qual o status mais frequente dos processos que estão sendo executados no sistema? Você saberia explicar por quê?

```
$ ps aux | awk '{print $8}' | sort | uniq -c | sort -n | tac
     24 S
     23 S<
     16 Ss
      4 S+
      1 STAT
      1 Ssl
      1 Ss+
      1 SN
      1 R+
      1 D+
```

O estado mais frequente é *sleep*, porque apenas um processo pode estar sendo executado pela CPU em um dado momento.

4) Alternando a execução de processos

1. Execute o comando `$ sleep 1000` diretamente do terminal.

```
$ sleep 1000
```

2. Pare o processo e mantenha-o em memória.

   Basta digitar a combinação de teclas `CTRL + Z`.

```
$ sleep 1000
^Z
[1]+  Parado
```

3. Liste os processos parados.

```
$ jobs
[1]+  Parado                 sleep 1000
```

4. Coloque-o em *background*.

```
$ bg
[1]+ sleep 1000 &
$ jobs
[1]+  Executando             sleep 1000 &
```

5. Verifique se o comando sleep 1000 está rodando.

```
$ ps ax | egrep 'sleep 1000$'
 2178 pts/0    S      0:00 sleep 1000
```

6. É possível cancelar a execução desse comando quando ele está rodando em *background*? Caso seja possível, faça-o.

```
$ kill 2178
$ ps ax | egrep 'sleep 1000$'
[1]+  Terminado              sleep 1000
```

5) Identificando o RUID e o EUID de um processo

1. Logado como o usuário `aluno`, execute o comando `passwd` no seu terminal. Antes de mudar a senha, abra uma segunda console e autentique-se como `root`. Verifique o `RUID` e o `EUID` associados ao processo `passwd`. Esses valores são iguais ou diferentes? Você saberia explicar por quê? Por fim, cancele a execução do processo `passwd`.

   Na primeira console, execute:

```
$ passwd
Mudando senha para aluno.
Senha UNIX (atual):
```

   Antes de digitar a senha, abra uma segunda console como `root` e execute:

```
# ps -eo user,ruser,comm | egrep '^USER | passwd$'
USER     RUSER    COMMAND
root     aluno    passwd

# which passwd
/usr/bin/passwd
# ls -lh /usr/bin/passwd
-rwsr-xr-x 1 root root 53K Mai 17  2017 /usr/bin/passwd
```

Os valores são diferentes porque o binário `passwd` possui o bit *SUID* ativado. O `RUID` (*real uid*) é do usuário que está executando o comando e o EUID (*effective uid*) é o do usuário `root`, que é o dono do arquivo.

6) Definindo a prioridade de processos

1. Verifique as opções do comando `nice` e em seguida, execute o comando abaixo, verificando sua prioridade, utilizando o comando `ps`:

   ```
   # nice -n -15 sleep 1000 &
   [1] 2289
   ```

   Basta executar o comando `# ps lax` e buscar o processo relevante, verificando o valor da quinta coluna. Em uma única linha e de forma mais específica, podemos fazer:

   ```
   # ps lax | egrep ' sleep 1000$' | awk '{print $5}'
   2289 5
   ```

2. Repita o comando do primeiro item, passando para o comando `nice` o parâmetro `-n -5`. Verifique como isso afeta a prioridade do processo. Ela aumentou, diminuiu ou permaneceu a mesma?

   ```
   # nice -n -5 sleep 1000 &
   [2] 2312
   # ps lax | egrep ' sleep 1000$' | awk '{print $3, $5}'
   2289 5
   2312 15
   ```

   A prioridade diminuiu, porque quanto maior o valor na coluna `PRI`, menor a prioridade do processo.

7) Editando arquivos crontab para o agendamento de tarefas

Neste exercício, trabalharemos com o comando `crontab`, utilizado para editar os arquivos `cron` do agendador de tarefas do sistema. Esses arquivos serão verificados pelo *daemon* `cron` periodicamente em busca de tarefas para serem executadas pelo sistema.

Para entender o funcionamento do `crontab`, o primeiro passo é ler as páginas do manual relevantes. Para o comando `crontab` em si, consulte a seção 1 do manual:

```
$ man 1 crontab
```

Para o formato de um arquivo de configuração `crontab`, consulte a seção 5:

```
$ man 5 crontab
```

1. Existe alguma entrada de `crontab` para o seu usuário?

```
$ crontab -l
no crontab for aluno
```

2. Que opção deve ser usada para editar o seu arquivo de `crontab`?

```
$ crontab -e
no crontab for aluno - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny

Choose 1-3 [1]: 1
No modification made
```

8) Agendando uma tarefa no daemon cron

Neste exercício, será necessário enviar mensagens de correio eletrônico. Para isso, você deverá utilizar o comando `mail`; o instrutor pode fornecer as informações básicas sobre ele. Um exemplo do uso desse comando para enviar uma mensagem ao endereço `fulano@dominio` com o assunto *Mensagem de teste* é:

```
$ mail fulano@dominio -s "Mensagem de teste" < /dev/null
```

1. Configure o `crontab` para que uma mensagem de correio eletrônico seja enviada automaticamente pelo sistema, sem interferência do administrador às 20:30 horas.

   Utilize o comando `$ crontab -e` para editar o `crontab` e inserir a linha:

   ```
   30 20 * * * mail fulano@dominio -s "Mensagem de teste" < /dev/null
   ```

2. Como verificar se a configuração foi feita corretamente?

   ```
   $ crontab -l | egrep -v '^#'
   30 20 * * * mail fulano@dominio -s "Mensagem de teste" < /dev/null
   ```

3. Qual o requisito fundamental para garantir que a ação programada será executada?

   O daemon do `cron` deve estar em execução e a sintaxe do `crontab`, incluindo a linha de comando utilizada, deve estar correta.

4. Há como confirmar se a mensagem foi efetivamente enviada, sem consultar o destinatário?

   Verifique no arquivo `/var/log/syslog` se a tarefa foi executada no horário correto com sucesso. Você deve ver uma entrada do tipo:

   ```
   /var/log/syslog:Aug  7 17:40:01 cliente CRON[2524]: (aluno) CMD (COMMAND)
   ```

   Dependendo da distribuição Linux em uso, as mensagens relativas ao `cron` podem estar em `/var/log/syslog`, `/var/log/cron.log`, `/var/log/daemon.log` ou outros arquivos. Verifique na documentação do fabricante/mantenedor.

5. Dê dois exemplos de utilização desse mecanismo para apoiar atividades do administrador de sistemas.

   Podemos, por exemplo, utilizar o `cron` para agendamento de backups e limpeza de diretórios temporários.

6. Faça um script que liste os arquivos sem dono do sistema e envie a lista por e-mail ao usuário root.

   O *script shell* abaixo mostra um exemplo de solução para o problema proposto, com a característica adicional de guardar os logs enviados por e-mail em um diretório dentro do *home* do `root`:

```
#!/bin/bash

LOGDIR="/root/nouser_logs"

[ ! -d $LOGDIR ] && mkdir $LOGDIR

curlog="$LOGDIR/nouser_$( date +%Y%m%d ).log"
find / -nouser -print > $curlog
mail -s "Files without ownership for $( date )" root < $curlog
```

7. Agende no crontab do usuário `root` o script do item 6, de modo que ele seja executado de segunda a sexta às 22:30 horas.

   Logado como usuário `root`, digite o comando `# crontab -e` para editar o `crontab` e insira a linha a seguir:

```
30 22 * * 1-5 /root/scripts/find_nouser.sh
```

9) Listando e removendo arquivos crontab

1. Liste o conteúdo do seu arquivo de `crontab` e, em seguida, remova-o. Quais as opções utilizadas para executar as ações demandadas?

```
$ crontab -l | egrep -v '^#'
30 20 * * * mail fulano@dominio -s "Mensagem de teste" < /dev/null

$ crontab -r
$ crontab -l
no crontab for aluno
```

10) Entendendo o comando exec

1. Execute o comando `$ exec ls -l`. Explique o que aconteceu.

```
# whoami
root
# exec ls -l /mnt/
total 0

$ whoami
aluno
```

   O shell corrente foi finalizado. Sempre que um comando é executado, um novo processo é criado. Já quando um comando é executado como argumento do comando `exec`, a imagem do

shell corrente é substituída pela do processo invocado, e quando esse processo encerra sua execução já não há mais shell de retorno.

Sistema de arquivos

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.

Registro de eventos

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) Registrando os eventos do kernel

1. Configure seu sistema de modo que os eventos gerados pelo kernel sejam registrados em um arquivo chamado `kernel.log`, no diretório `/var/log`.

```
# echo "kern.*        -/var/log/kernel.log" >> /etc/rsyslog.conf
# systemctl restart rsyslog.service

# cat /var/log/kernel.log
cat: /var/log/kernel.log: Arquivo ou diretório não encontrado
```

   Mesmo após reiniciar o *daemon* `rsyslog`, o arquivo não será criado de imediato. Para testar o funcionamento da diretiva, precisamos gerar alguma mensagem para a *facility* apropriada:

```
# modprobe lp
# cat /var/log/kernel.log
Aug  9 11:15:45 cliente kernel: [  447.128333] lp: driver loaded but no devices
found
```

2) Analisando os arquivos de log do sistema

Para esta atividade você terá que ter acesso `ssh` à máquina em que está configurando o sistema de logs para que você possa acompanhar, em tempo real, os registros gravados nos arquivos de log. Cada aluno deve executar os passos abaixo em seu próprio servidor.

1. Crie, em seu servidor, uma conta com senha para acesso via `ssh`.

```
# useradd -m aluno2
# passwd aluno2
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: senha atualizada com sucesso
```

2. A partir de uma máquina remota faça o login no seu servidor utilizando a conta criada no passo anterior. Utilize o comando `tail` com a opção `-f` para verificar em tempo real os registros gerados pelo `syslog` no arquivo `/var/log/auth.log`.

   No servidor, execute:

```
# tail -f -n0 /var/log/auth.log
```

De outra máquina, faça login via `ssh` no servidor com a conta criada anteriormente:

```
$ ssh aluno2@192.168.0.25
aluno2@192.168.0.25's password:

aluno2@cliente:~$
```

Monitore o que aconteceu no arquivo `/var/log/auth.log`:

```
# tail -f -n0 /var/log/auth.log
Aug  9 11:26:24 cliente sshd[1050]: Accepted password for aluno2 from 192.168.0.12
port 50325 ssh2
Aug  9 11:26:24 cliente sshd[1050]: pam_unix(sshd:session): session opened for user
aluno2 by (uid=0)
```

3. Faça um *script* que contabilize o número de tentativas de login mal sucedidas através do `ssh`, listando os IPs de origem e quantas tentativas foram feitas por cada IP.

   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

while read -r line; do
  s=( $( echo $line ) )
  echo -e "Host ${s[1]}: ${s[0]} failed logins"
done < <( grep "(sshd.auth): authentication failure.*rhost=" /var/log/auth.log |
awk '{print $14}' | cut -d'=' -f2 | sort -n | uniq -c )
```

3) Analisando os arquivos de log binários do sistema

Nesta atividade, você irá trabalhar com os arquivos de log binários armazenados no diretório `/var/log`.

1. Verifique quais foram os dois últimos usuários a efetuarem login em seu computador.

```
$ last | head -n2
aluno2   pts/1        192.168.0.12     Thu Aug  9 11:26 - 11:27  (00:01)
aluno    pts/0        192.168.0.12     Thu Aug  9 11:10   still logged in
```

2. Como você poderia verificar as contas existentes em seu computador que nunca efetuaram login?

```
$ lastlog | grep '**Nunca logou**' | sort
avahi-autoipd                                **Nunca logou**
backup                                       **Nunca logou**
bin                                          **Nunca logou**
daemon                                       **Nunca logou**
Debian-exim                                  **Nunca logou**
funcionario                                  **Nunca logou**
games                                        **Nunca logou**
gnats                                        **Nunca logou**
irc                                          **Nunca logou**
list                                         **Nunca logou**
lp                                           **Nunca logou**
mail                                         **Nunca logou**
man                                          **Nunca logou**
marcelo                                      **Nunca logou**
messagebus                                   **Nunca logou**
news                                         **Nunca logou**
nobody                                       **Nunca logou**
pedro                                        **Nunca logou**
proxy                                        **Nunca logou**
sshd                                         **Nunca logou**
statd                                        **Nunca logou**
sync                                         **Nunca logou**
sys                                          **Nunca logou**
systemd-bus-proxy                             **Nunca logou**
systemd-network                              **Nunca logou**
systemd-resolve                              **Nunca logou**
systemd-timesync                             **Nunca logou**
uucp                                         **Nunca logou**
www-data                                     **Nunca logou**
```

3. Qual a maneira mais fácil de identificar um login remoto efetuado em seu computador?

   Através do comando `last`. A terceira coluna mostra o *host* de origem do login, seja ele local ou remoto:

```
$ last | head -n20 | grep -v '^reboot'
aluno2   pts/1        192.168.0.12     Thu Aug  9 11:26 - 11:27  (00:01)
aluno    pts/0        192.168.0.12     Thu Aug  9 11:10   still logged in
root     tty1                          Thu Aug  9 03:25 - down   (00:00)
aluno    pts/0        192.168.0.12     Thu Aug  9 02:32 - 03:25  (00:53)
aluno    pts/0        192.168.0.12     Thu Aug  9 02:25 - down   (00:05)
aluno    pts/0        192.168.0.12     Thu Aug  9 01:47 - down   (00:37)
root     tty1                          Wed Aug  8 19:05 - down   (00:00)
aluno    pts/0        192.168.0.12     Wed Aug  8 18:19 - 19:05  (00:46)
root     tty1                          Tue Aug  7 18:18 - down   (00:00)
aluno    pts/0        192.168.0.12     Tue Aug  7 17:56 - 18:17  (00:21)
aluno    pts/1        192.168.0.12     Tue Aug  7 17:07 - 17:15  (00:07)
instruto pts/1        localhost        Tue Aug  7 15:45 - 16:01  (00:15)
instruto pts/1        localhost        Tue Aug  7 14:44 - 14:46  (00:01)
instruto pts/1        localhost        Tue Aug  7 14:42 - 14:42  (00:00)
instruto pts/1        localhost        Tue Aug  7 14:39 - 14:39  (00:00)
```

4. Faça um *script* que mostre o tempo total que cada usuário ficou logado no sistema utilizando as informações obtidas com o comando `last`.

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

```
#!/bin/bash

users=( $( last -w | egrep '(tty|pts)' | awk '{print $1}' | sort | uniq ) )

for user in "${users[@]}"; do
  times=( $( last -w | egrep "^$user " | egrep '(tty|pts)' | egrep -v 'still logged
in *$' | sed 's/  *$//' | awk -F '[ :()]' '{printf "%s:%s\n", $(NF-2), $(NF-1)}' )
)

  h=0
  m=0
  for time in "${times[@]}" ; do
    s=( $( echo $time | tr ':' ' ' ) )
    ((h+=${s[0]}))
    ((m+=${s[1]}))
  done

  mh=$(($m/60))
  mr=$(($m%60))
  ((h+=$mh))

  echo "User \"$user\" logged time: $h hours, $mr minutes"
done
```

4) Servidor de log remoto

1. Este exercício deve ser feito utilizando duas máquinas virtuais Linux. Configure na máquina virtual *Server_Linux* um servidor de logs; posteriormente, configure a máquina virtual *Client_Linux* para enviar os registros dos eventos gerados para esse servidor de logs.

   Na máquina *Server_Linux*, edite o arquivo `/etc/rsyslog.conf` e descomente as linhas que se seguem. Em seguida, reinicie o serviço do `rsyslog`.

   ```
   # grep -A1 'imudp' /etc/rsyslog.conf
   $ModLoad imudp
   $UDPServerRun 514

   # systemctl restart rsyslog.service
   ```

   Na máquina *Client_Linux*, configure o envio de logs para o servidor remoto editando o arquivo `/etc/rsyslog.conf` e inserindo a linha que se segue ao final do arquivo, substituindo o endereço IP `192.168.0.10` pelo IP da máquina *Server_Linux*. Em seguida, reinicie o serviço do `rsyslog`.

   ```
   # tail -n1 /etc/rsyslog.conf
   *.*                     @192.168.0.10

   # systemctl restart rsyslog.service
   ```

2. Após terminar a configuração, efetue um login na máquina *Client_Linux* em um terminal qualquer e verifique onde foi registrado esse evento no servidor de logs *Server_Linux*.

   Tendo em vista que o evento gerado na máquina *Client_Linux* será de login, o registro deverá ser enviado para o arquivo onde eventos de autenticação são enviados, na *facility* `authpriv`:

   ```
   # grep '^auth,authpriv' /etc/rsyslog.conf
   auth,authpriv.*                 /var/log/auth.log
   ```

   Sabendo que o arquivo a ser monitorado é o `/var/log/auth.log`, usaremos o comando `tail` para fazê-lo:

   ```
   # tail -f -n0 /var/log/auth.log
   ```

   Após gerar um evento de login via `ssh` na máquina *Client_Linux*, imediamente a mesma mensagem aparece replicada nos logs da máquina *Server_Linux*:

```
# tail -f -n0 /var/log/auth.log
Aug  9 15:18:07 cliente sshd[3285]: Accepted password for aluno from 192.168.0.12
port 50854 ssh2
Aug  9 15:18:07 cliente sshd[3285]: pam_unix(sshd:session): session opened for user
aluno by (uid=0)
```

Evidentemente, é muito confuso ter todas as mensagens de log de uma máquina remota sendo colocadas nos mesmos arquivos que registram os eventos do servidor local. Para tratar esses logs com mais clareza, é interessante separar os logs de cada *host* remoto em seus próprios arquivos e pastas para facilitar o processamento e entendimento. A seguinte configuração pode ser útil para atingir esse objetivo.

Primeiro, note que o `rsyslog` inclui arquivos customizados pelo usuário terminados com a extensão `.conf` no diretório `/etc/rsyslog.d`:

```
# grep '^\$IncludeConfig' /etc/rsyslog.conf
$IncludeConfig /etc/rsyslog.d/*.conf
```

Vamos criar um arquivo novo nessa pasta indicando que os logs da máquina *Client_Linux* devem ser enviados para o arquivo `/var/log/client_linux.log`, e nenhum outro arquivo (palavra-chave `stop`). Feito isso, reinicia-se o *daemon* `rsyslog`:

```
# cat /etc/rsyslog.d/client_linux.conf
if $fromhost-ip == '192.168.0.25'     then /var/log/client_linux.log
& stop

# systemctl restart rsyslog.service
```

Pronto! Agora, novos eventos gerados pela máquina *Client_Linux* serão enviados exclusivamente para o arquivo `/var/log/client_linux.log`, sem se misturar com os eventos locais do servidor de logs.

```
# tail -f -n0 /var/log/client_linux.log
Aug  9 15:34:33 cliente sshd[3340]: Accepted password for aluno from
192.168.0.12 port 50902 ssh2
Aug  9 15:34:33 cliente sshd[3340]: pam_unix(sshd:session): session
opened for user aluno by (uid=0)
```

3. Cite três vantagens obtidas com o uso de um servidor de logs.
    - Facilita o gerenciamento dos arquivos de log, já que estão centralizados em um único servidor.
    - Aumenta a segurança no armazenamento dos arquivos de log, pois o servidor pode estar em

outra rede, com regras diferenciadas, dificultando o acesso de possíveis invasores.

- Facilita o backup dos arquivos de log.

5) Utilizando o logger

Nesta atividade, você irá verificar uma funcionalidade importante do comando `logger`.

1. Na máquina *Server_Linux*, inclua uma nova regra no arquivo `/etc/rsyslog.conf`, de modo que qualquer evento gerado pelo daemon `cron` seja registrado no arquivo `/var/log/cron.log`.

```
# tail -n1 /etc/rsyslog.conf
cron.*                  /var/log/cron.log

# systemctl restart rsyslog.service
```

2. Utilize o comando `logger` para testar se a alteração feita no passo anterior produziu o efeito esperado.

```
# logger -p cron.info "teste"

# tail /var/log/cron.log
Aug  9 15:52:26 servidor aluno: teste
```

6) Rotacionando arquivos de log do sistema

Nesta atividade, você irá configurar o rotacionamento dos arquivos de log de seu computador.

1. Realize o rotacionamento mensal do arquivo recém-criado `/var/log/cron.log`, mantendo uma cópia dos dois últimos arquivos compactados e criando, automaticamente, um novo arquivo vazio após o rotacionamento.

   No arquivo `/etc/logrotate.conf` estão as configurações globais para o rotacionamento dos arquivos de log. Ao configurar o rotacionamento de um arquivo ou um grupo de logs podemos editar diretamente esse arquivo ou, opcionalmente, incluir novas configurações dentro do diretório `/etc/logrotate.d`.

```
# grep '^include' /etc/logrotate.conf
include /etc/logrotate.d

# ls /etc/logrotate.d/
apt  aptitude  dpkg  exim4-base  exim4-paniclog  iptraf  rsyslog
```

   Vamos criar um arquivo `/etc/logrotate.d/cron` para configurar os aspectos de rotacionamento de logs desse arquivo de acordo com os parâmetro especificados no exercício.

```
# cat /etc/logrotate.d/cron
/var/log/cron.log
{
  rotate 2
  monthly
  missingok
  notifempty
  create 640 root adm
  delaycompress
  compress
  postrotate
    systemctl reload cron.service > /dev/null
  endscript
}
```

7) Aplicativos para análise de arquivos de log

1. Na máquina *Server_Linux*, instale o pacote `logwatch` através do comando `apt-get` e configure-o para enviar um relatório diário do sistema para o usuário `root`. Um exemplo do arquivo de configuração está disponível em `/usr/share/logwatch/default.conf/logwatch.conf`.

   Primeiro, vamos instalar o pacote:

   ```
   # apt-get install logwatch
   ```

   A seguir, vamos copiar o modelo do arquivo de configuração em `/usr/share/logwatch/default.conf/logwatch.conf` para o diretório `/etc/logwatch/conf`:

   ```
   # cp /usr/share/logwatch/default.conf/logwatch.conf /etc/logwatch/conf/
   ```

   Edite o arquivo para que o período e opções de envio fiquem de acordo com o solicitado pela atividade:

   ```
   # grep -v '^ *#' /etc/logwatch/conf/logwatch.conf  | sed '/^$/d'
   LogDir = /var/log
   TmpDir = /var/cache/logwatch
   Output = mail
   Format = text
   Encode = none
   MailTo = root
   MailFrom = Logwatch
   Range = All
   Detail = Low
   Service = All
   mailer = "/usr/sbin/sendmail -t"
   ```

Lembre-se de criar o diretório `/var/cache/logwatch`, que ainda não existe:

```
# mkdir /var/cache/logwatch
```

Finalmente, observe que por padrão o Debian já habilita a execução diária do `logwatch` através de um *script* instalado pelo próprio pacote no diretório `/etc/cron.daily`:

```
# ls /etc/cron.daily/ | grep 'logwatch'
00logwatch

# cat /etc/cron.daily/00logwatch | grep -v '^#' | sed '/^$/d'
test -x /usr/share/logwatch/scripts/logwatch.pl || exit 0
/usr/sbin/logwatch --output mail
```

2. Ainda na máquina *Server_Linux*, crie uma regra para o `swatch` que envie um e-mail de notificação ao administrador quando alguma tentativa de login via `ssh`, ou `su` para o usuário `root`, falharem.

   Primeiro, vamos instalar o `swatch` via `apt-get`:

   ```
   # apt-get install swatch
   ```

   A configuração do `swatch` é um tanto quanto arcana, mas a página de manual do programa (`$ man 1p swatch`) nos dá algum direcionamento através da seção *CONFIGURATION EXAMPLE*. Um dos requisitos é criar um arquivo de configuração com a expressão regular que casa com o erro de autenticação do daemon do `sshd`. Primeiro, precisamos conhecer o formato da mensagem:

   ```
   Aug  9 16:39:56 servidor sshd[4113]: Failed password for aluno from 192.168.0.12
   port 51230 ssh2
   ```

   Outro ponto de atenção é a tentativa de `su` para o usuário `root` com falha, possivelmente por senha incorreta. Vamos verificar o formato da mensagem de log:

   ```
   Aug  9 16:46:29 servidor su[4175]: FAILED su for root by aluno
   ```

   Sabendo os formatos objetivados, vamos agora elaborar expressões regulares que casem com os padrões acima, extraiam informação relevante, e executem uma ação apropriada — enviar e-mail de notificação ao usuário `root` em caso de violação desses padrões:

```
# cat /etc/swatch.conf
watchfor /^(.*sshd\[[0-9]*\]\: Failed password for [A-Za-z0-9]* from ([0-9:.]*).*)/
  exec "echo '$1' | mail root -s '[swatch][ssh]:\ $2' "
  echo

watchfor /^(.*su\[[0-9]*\]\: FAILED su for root by ([A-Za-z0-9]*))/
  exec "echo '$1' | mail root -s '[swatch][su]:\ $2' "
  echo
```

Vamos rodar o `swatch` manualmente e testar se os padrões estão sendo capturados. Serão realizadas duas ações de violação — um login `ssh` com senha incorreta e uma tentativa de `su` para `root` com senha incorreta.

```
# swatch --tail-file=/var/log/auth.log --config-file=/etc/swatch.conf --pid
-file=/var/run/swatch.pid

*** swatch version 3.2.3 (pid:5011) started at Qui Ago  9 17:29:51 -03 2018

Aug  9 17:32:35 servidor sshd[5093]: Failed password for aluno from 192.168.0.12
port 51460 ssh2
Aug  9 17:32:43 servidor su[5117]: FAILED su for root by aluno
```

Aparentemente, tudo funcionou. Vamos verificar se os e-mails estão sendo de fato enviados:

```
$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/aluno": 2 messages 2 new
>N  1 root@servidor.emp  Thu Aug 09 17:32   16/705   [swatch][ssh]: 192.168.0.12
 N  2 root@servidor.emp  Thu Aug 09 17:32   16/663   [swatch][su]: aluno
& 1
Message 1:
From root@servidor.empresa.com.br Thu Aug 09 17:32:35 2018
Envelope-to: root@servidor.empresa.com.br
Delivery-date: Thu, 09 Aug 2018 17:32:35 -0300
To: root@servidor.empresa.com.br
Subject: [swatch][ssh]: 192.168.0.12
From: root <root@servidor.empresa.com.br>
Date: Thu, 09 Aug 2018 17:32:35 -0300

Aug  9 17:32:35 servidor sshd[5093]: Failed password for aluno from 192.168.0.12
port 51460 ssh2

& 2
Message 2:
From root@servidor.empresa.com.br Thu Aug 09 17:32:43 2018
Envelope-to: root@servidor.empresa.com.br
Delivery-date: Thu, 09 Aug 2018 17:32:43 -0300
To: root@servidor.empresa.com.br
Subject: [swatch][su]: aluno
From: root <root@servidor.empresa.com.br>
Date: Thu, 09 Aug 2018 17:32:43 -0300

Aug  9 17:32:43 servidor su[5117]: FAILED su for root by aluno
```

Excelente! Para que o `swatch` não tenha que ser iniciado manualmente, e continue operando mesmo após o reinício do sistema, é necessário que ele possua um *initscript* correspondente. Infelizmente, a versão instalada pelo apt-get não disponibiliza tal facilidade nem em formato legado (no diretório `/etc/init.d`) nem em arquivo de serviço para o `systemd` (que ficam no diretório `/etc/systemd/system`). Felizmente, é relativamente fácil criar um arquivo de serviço para o `systemd` manualmente:

```
# cat /etc/systemd/system/swatch.service
[Unit]
Description=Swatch Log Monitoring Daemon
After=syslog.target network.target auditd.service sshd.service

[Service]
ExecStart=/usr/bin/swatch --config-file=/etc/swatch.conf --tail
-file=/var/log/auth.log --pid-file=/var/run/swatch.pid --daemon
ExecStop=/bin/kill -s KILL $(cat /var/run/swatch.pid)
Type=forking
PIDFile=/var/run/swatch.pid

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

Uma vez criado, deve-se instruir o `systemd` a carregar o arquivo:

```
# systemctl daemon-reload
```

Pronto! Agora é possível habilitar/desabilitar o `swatch` durante o *boot* do sistema, e iniciar/parar/reiniciar e verificar o estado do serviço normalmente:

```
# systemctl enable swatch.service
Created symlink from /etc/systemd/system/multi-user.target.wants/swatch.service to
/etc/systemd/system/swatch.service.

# systemctl is-enabled swatch.service
enabled

# systemctl start swatch.service

# systemctl status swatch.service
● swatch.service - Swatch Log Monitoring Daemon
   Loaded: loaded (/etc/systemd/system/swatch.service; enabled)
   Active: active (running) since Qui 2018-08-09 17:37:57 -03; 4s ago
  Process: 5216 ExecStart=/usr/bin/swatch --config-file=/etc/swatch.conf --tail
-file=/var/log/auth.log --pid-file=/var/run/swatch.pid --daemon (code=exited,
status=0/SUCCESS)
 Main PID: 5218 (/usr/bin/swatch)
   CGroup: /system.slice/swatch.service
           ├──5218 /usr/bin/swatch --config-file=/etc/swatch.conf --tail
-file=/var/log/auth...
           └──5219 /usr/bin/tail -n 0 -F /var/log/auth.log

Ago 09 17:37:57 servidor systemd[1]: Starting Swatch Log Monitoring Daemon...
Ago 09 17:37:57 servidor systemd[1]: PID file /var/run/swatch.pid not readable
(yet?) a...rt.
Ago 09 17:37:57 servidor systemd[1]: Started Swatch Log Monitoring Daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

3. Ainda na máquina *Server_Linux*, habilite o `logcheck` para enviar relatórios ao usuário `root` de 30 em 30 minutos (ex: 1:00, 1:30, etc.).

   Primeiro, vamos instalar o `logcheck` via `apt-get`:

   ```
   # apt-get install logcheck
   ```

   O `logcheck` já vem com envio de e-mails habilitado por padrão, então a única configuração necessária é alterar a periodicidade de envio de relatórios. O arquivo `/etc/cron.d/logcheck` vem configurado para envios de hora em hora. Edite a linha:

   ```
   2 * * * *       logcheck    if [ -x /usr/sbin/logcheck ]; then nice -n10
   /usr/sbin/logcheck; fi
   ```

   Alterando-a para:

   ```
   0,30 * * * *       logcheck    if [ -x /usr/sbin/logcheck ]; then nice -n10
   /usr/sbin/logcheck; fi
   ```

O `logcheck` fará um *scan* dos logs de sistema e enviará por e-mail linhas consideradas "interessantes" — note que o programa envia apenas os registros ocorridos desde a sua última execução.

8) Recomendações básicas de segurança

1. O que você faria para aumentar o nível de segurança em um servidor de logs centralizado? Cite duas opções.
    - Desabilitar o serviço `sshd` no servidor de logs, permitindo acesso somente pela console.
    - Configurar o firewall de *host* para permitir apenas tráfego de pacotes UDP na porta 514.
    - Utilizar uma rede isolada para a troca de mensagens de log.
    - Desinstalar todos os serviços que não estão sendo utilizados ou são desnecessários à função do servidor.
    - Manter o sistema operacional rigorosamente atualizado.