

Permissões Especiais

Prof. Me Robson Ferreira Lopes

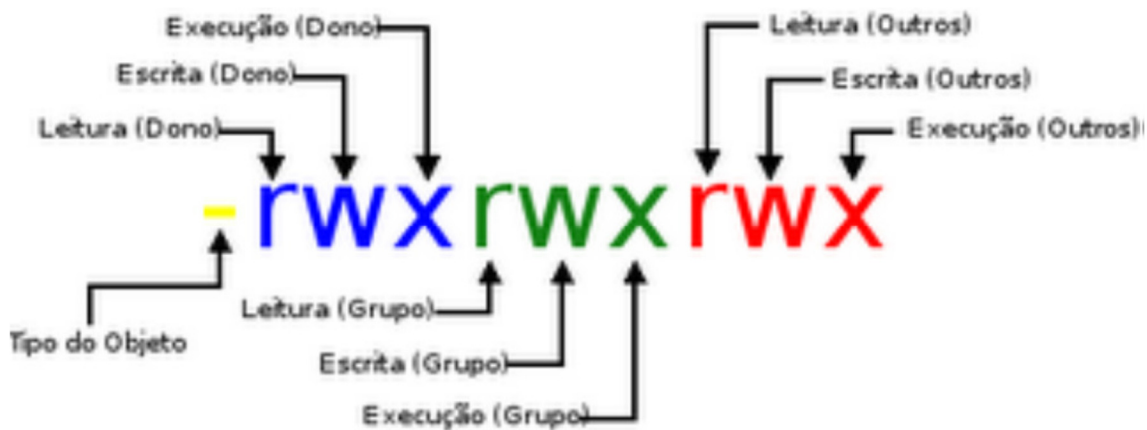
rferreira@ifsp.edu.br

IFSP Câmpus Guarulhos

Permissões Tradicionais

No Linux há três modelos de controle de acesso básicos: Read, Write e Execution.

Veja um exemplo de permissão básica na figura abaixo:



O Tipo do Objeto significa:

d => diretório;

b => arquivo de bloco;

c => arquivo especial de caractere;

p => canal;

s => socket;

- => arquivo normal.

Já os outros caracteres significam:

r => permissão de leitura (read);

w => permissão de gravação (write);

x => permissão de execução (execution);

- => permissão desabilitada.

Explicando um pouco mais:

Leitura (r):

Em arquivos, permite examinar o conteúdo do arquivo.

Em diretórios permite listar conteúdo do diretório.

Escrita (w):

Em arquivos, permite escrever, remover e alterar o arquivo.

Em diretórios, permite criar e remover arquivos dentro do diretório.

Execução (x):

Em arquivos, permite executar um arquivo como um programa.

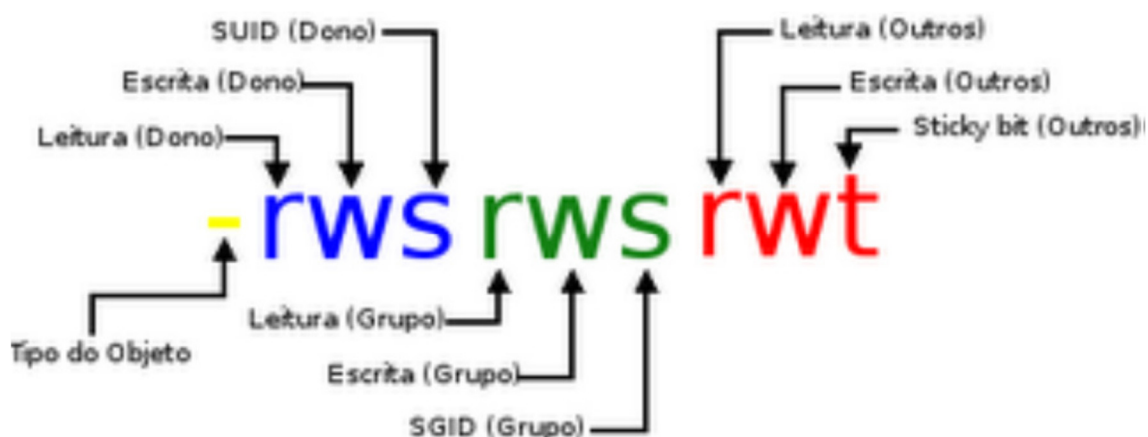
Em diretório, permite ler e escrever em arquivos dentro do diretório.

Sendo que os três primeiros rwx pertencem ao Dono do arquivo, os outros três rwx pertencem ao Grupo e por fim os últimos três rwx pertencem há outros usuários que não fazem parte do grupo.

Permissões de acesso especiais

Mas somente as três permissões básicas (rwx) não dão toda flexibilidade para controlar acesso aos arquivos e diretórios. Por isso o Linux tem mais três modelos especiais para controle de acesso, chamados suid (set user id), sgid (set group id) e Sticky (Sticky bit).

Veja um exemplo de permissão especial na figura abaixo:



SUID (4) (Set User ID)

A propriedade SUID é somente para arquivos executáveis e não tem efeito em diretórios. Nas permissões básicas, o usuário que executou o programa é dono do processo. Mas em arquivo executável com a propriedade SUID aplicada, o programa rodará com o ID do dono do arquivo, não como ID do usuário que executou o programa. Normalmente o usuário dono do programa executável é também dono do processo sendo executado. Ou seja, quando um arquivo/programa executável tem o controle de acesso SUID, ele é executado como se ele estivesse iniciado pelo dono do arquivo/programa.

A permissão de acesso especial SUID pode aparecer somente no campo Dono.

SGID(2) (Set Group ID)

A propriedade SGID tem a mesma função que o SUID para arquivos executáveis. Mas a propriedade SGID tem um efeito especial para os diretórios. Quando SGID é aplicado em um diretório, os novos arquivos que são criados dentro do diretório assumem o mesmo ID de Grupo do diretório com a propriedade SGID aplicado. A permissão de acesso especial SGID pode aparecer somente no campo Grupo.

Sticky(1) (Sticky bit)

Em arquivos executáveis, a propriedade Sticky faz com que o sistema mantenha uma imagem do programa em memória depois que o programa finalizar. Esta capacidade aumenta o desempenho, pois será feito um cache do programa para a memória e da próxima vez que ele for executado, será carregado mais rápido.

Em diretórios, a propriedade Sticky impede que outros usuários deletem ou renomeiem arquivos dos quais não são donos. Isso normalmente é utilizado para aumentar a segurança, pois o diretório estará em modo append-only (somente incremento). Sendo assim, somente o usuário que é Dono do arquivo, poderá deletar ou renomear os arquivos dentro de um diretório com a propriedade Sticky aplicada. A permissão especial Sticky pode ser especificada somente no campo outros usuários das permissões de acesso.

Usando as personagens e características dos filmes de Harry Potter vamos entender como funciona as permissões especiais

Considerações finais

Na prática quando realizamos uma estrutura de compartilhamento combinado SGID com o sticky bit para que quando um arquivo é criado tenha sim como dono os usuários que o criou e o grupo seja o mesmo atribuído do diretório. Além disso somente o dono poderá apagar esse arquivo. O único usuário que poderá ter privilégios acima desses é o root.

Hora da prática

Verificando programas que tem o SUID ativado

```
root@operacao:~# find / -perm -4000 2> /dev/null
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmccrypt-get-device
/usr/lib/xorg/Xorg.wrap
/usr/lib/openssh/ssh-keysign
/usr/bin/su
/usr/bin/ntfs-3g
/usr/bin/gpasswd
/usr/bin/umount
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/bwrap
/usr/bin/fusermount
/usr/bin/mount
/usr/bin/chsh
/usr/bin/passwd
/usr/sbin/pppd
```

Verificando programas que tem o SGid ativado

```
root@operacao:~# find / -perm -2000 2> /dev/null
/usr/bin/chage
/usr/bin/ssh-agent
/usr/bin/dotlockfile
/usr/sbin/unix_chkpwd
/usr/local/share/fonts
/run/log/journal
/run/log/journal/ee2abbb4f5114276b29b2c7121589021
/var/mail
/var/local
...
```

Verificando programas que tem o Sticky Bitativado

```
root@operacao:~# find / -perm -1000 2> /dev/null  
/dev/mqueue  
/dev/shm  
/run/screen  
/run/lock  
/var/spool/cron/crontabs  
/var/tmp  
/tmp  
...
```

Criar os usuários Harry, Hermione, Luna, Severo, Cho, Malfoy e Dumbledore

```
#adduser harry  
#adduser hermione  
#adduser malfoy  
#adduser severo  
#adduser cho  
#adduser luna  
#adduser dumbledore
```

Criar o grupo armada, grifinoria, sonserina e corvinal

```
#addgroup armada  
#addgroup grifinoria  
#addgroup sonserina  
#addgroup corvinal
```

Adicione harry, hermione e luna no grupo armada, Harry e Hermione no grupo grifinoria, cho e a luna no grupo corvinal por fim malfoy e severo no grupo sonserina.

```
#gpasswd -a harry armada  
#gpasswd -a hermione armada  
#gpasswd -a luna armada  
#gpasswd -a harry grifinoria  
#gpasswd -a hermione grifinoria  
#gpasswd -a luna corvinal  
#gpasswd -a cho corvinal
```

```
#gpasswd -a malfoy sonserina
#gpasswd -a severo sonserina
```

Criar os diretórios precisa, sonserina, grifinoria e corvinal em /srv

```
root@operacao:~#mkdir /srv/{precisa,sonserina,grifinoria,corvinal}
```

Resultado:

```
root@operacao:~# cat /etc/passwd | awk -F: '$3 >= 1002 && $3 < 1020 {print $1}'
```

harry

hermione

malfoy

luna

severo

cho

dumbledore

```
root@operacao:~# cat /etc/group | awk -F: '$3 >= 1002 && $3 < 1020 {print $1}'
```

armada

grifinoria

sonserina

corvinal

```
root@operacao:~# cat /etc/group | awk -F: '$3 >= 1002 && $3 < 1020 {print $1FS$4}'
```

armada:harry,hermione,luna

grifinoria:harry,hermione

sonserina:malfoy,severo

corvinal:luna,cho

```
root@operacao:~# tree /srv
```

/srv

├─ corvinal

├─ grifinoria

├─ precisa

| └─ chapeu.sh

| └─ membrosarmada.txt

```
|   └─ nomes.txt
└─ sonserina
```

```
root@operacao:~# ls -lha /srv
total 24K
drwxr-xr-x  6 root root 4,0K jan 22 17:20 .
drwxr-xr-x 20 root root 4,0K dez 12 14:41 ..
drwxr-xr-x  2 root root 4,0K jan 22 17:20 corvinal
drwxr-xr-x  2 root root 4,0K jan 22 17:20 grifinoria
drwxr-xr-x  2 root root 4,0K jan 22 17:03 precisa
drwxr-xr-x  2 root root 4,0K jan 22 17:20 sonserina
```

5. Gerar conteúdo para o diretório /srv/precisa

5.1 Usando a técnica dos redirecionadores de fluxo criar um arquivo membrosarmada.txt com os nomes Harry Potter, Hermione e Ronald

```
root@operacao:~# cat << membros > /srv/precisa/membrosarmada.txt
> Harry Potter
> Hermione
> Ronald
> membros
```

5.2 Elaborar o script logo a seguir com o nome de chapéu.sh

```
root@operacao:/srv/precisa# vim chapéu.sh
```

```
#!/bin/bash
echo "Preparando para execução"
sleep 3
echo "Quem é você? "
read NOME
echo $NOME >> nomes.txt 2> /dev/null
if [ $? = 0 ]
then
    clear
    echo "$NOME incluído"
    sleep 5
    cat nomes.txt
```

```
    exit 0
else
    clear
    echo "$NOME não incluído"
    exit 1
fi
```

5.2.1 Criar o arquivo nomes.txt dentro de /srv/precisa

```
#echo ***** Lista de nomes ***** > /srv/precisa/nomes.txt
```

5.2.2 Dar permissão de execução ao script chapeu.sh

```
#chmod 755 /srv/precusa/chapeu.sh
```

5.3 Mudar o grupo dos arquivos e do diretório /srv/precisa em modo recursivo para armada.

```
#chgrp -R armada /srv/precisa
```

5.4 Mudar o grupo dos diretórios grifinoria, sonserina e corvinal para o respectivo grupo.

```
#chgrp -R grifinoria /srv/grifinoria
```

```
#chgrp -R sonserina /srv/sonserina
```

```
#chgrp -R corvinal /srv/corvinal
```

5.5 Ajuste a permissão total para o dono e grupo e retire a permissão dos outros usuários acessarem esse arquivo.

```
#chmod 770 /srv/precisa
```

```
#chmod 770 /srv/grifinoria
```

```
#chmod 770 /srv/sonserina
```

```
#chmod 770 /srv/corvinal
```

Resultado 5.3, 5.4 e 5.5

```
root@operacao:~# ls -lhaR /srv
```

```
/srv:
```

```
total 24K
```

```
drwxr-xr-x  6 root root          4,0K jan 22 17:20 .
```

```
drwxr-xr-x 20 root root          4,0K dez 12 14:41 ..
```



```
drwxr-xr-x  2 root corvinal    4,0K jan 22 17:20 corvinal
drwxr-xr-x  2 root grifinoria  4,0K jan 22 17:20 grifinoria
drwxr-xr-x  2 root armada      4,0K jan 22 17:03 precisa
drwxr-xr-x  2 root sonserina   4,0K jan 22 17:20 sonserina
```

/srv/corvinal:

total 8,0K

```
drwxr-xr-x 2 root corvinal 4,0K jan 22 17:20 .
drwxr-xr-x 6 root root      4,0K jan 22 17:20 ..
```

/srv/grifinoria:

total 8,0K

```
drwxr-xr-x 2 root grifinoria 4,0K jan 22 17:20 .
drwxr-xr-x 6 root root        4,0K jan 22 17:20 ..
```

/srv/precisa:

total 20K

```
drwxr-xr-x 2 root armada 4,0K jan 22 17:03 .
drwxr-xr-x 6 root root   4,0K jan 22 17:20 ..
-rwxr-xr-x 1 root armada 264 jan 22 17:03 chapeu.sh
-rw-rw-r-- 1 root armada  29 jan 22 17:24 membrosarmada.txt
-rw-rw-r-- 1 root armada  83 jan 22 17:03 nomes.txt
```

/srv/sonserina:

total 8,0K

```
drwxr-xr-x 2 root sonserina 4,0K jan 22 17:20 .
drwxr-xr-x 6 root root      4,0K jan 22 17:20 ..
```

Agora que o cenário foi preparado vamos começar a testar as permissões especiais

O dumbledore é o administrador e será quem ira administrar o castelo. Assim terá alguns poderes adicionais como por exemplo alterar permissões dos usuários assim...

Inclua o dumbledore no grupo do root e no grupo armada

```
#gpasswd -a dumbledore root
```

```
#gpasswd -a dumbledore root
```

Aplique o comando ls para exibir com detalhes o binário do chmod

```
root@operacao:~# ls -lha /usr/bin/chmod
-rwxr-xr-x 1 root armada 63K fev 28 2019 /usr/bin/chmod
```

Em outro terminal com o usuário dumbledore (use o comando su -)

```
$su - dumbledore
```

Entre no diretório /srv/precisa e liste seu conteúdo

```
$cd /srv/precisa
dumbledore@operacao:/srv/precisa$ ls -la
total 28
drwxrwx--- 2 root armada 4096 jan 22 21:47 .
drwxr-xr-x 6 root root 4096 jan 22 17:20 ..
-rwxr-xr-x 1 root armada 264 jan 22 17:03 chapeu.sh
-rw-rw-r-- 1 luna users 30 jan 22 21:46 cidades.txt
-rw-rw-rw- 1 luna users 29 jan 22 20:43 frutas.txt
-rw-rw-r-- 1 root armada 29 jan 22 17:24 membrosarmada.txt
-rw-r--r-- 1 luna users 0 jan 22 21:47 mensagem.txt
-rw-rw-rw- 1 root armada 100 jan 22 20:40 nomes.txt
```

Teste a criação de um arquivo vazio com o nome de dumbledore.txt em seguida tente trocar a permissão para 666

```
dumbledore@operacao:/srv/precisa$ > dumbledore.txt
dumbledore@operacao:/srv/precisa$ chmod 666 dumbledore.txt
```

Resultado:

```
dumbledore@operacao:/srv/precisa$ ls -l dumbledore.txt
-rw-rw-rw- 1 dumbledore users 0 jan 22 21:56 dumbledore.txt
```

Comentário: Como o dono é o próprio dumbledore ele troca a permissão mas no arquivo nomes.txt que tem o dono o root ele não permite alteração da permissão

como: alterando permissões de 'nomes.txt': Operação não permitida

No terminal do root aplique a permissão SUID

```
root@operacao:~# chmod 4755 /usr/bin/chmod
```

Retorne ao terminal do dumbledore

```
dumbledore@operacao:/srv/precisa$ chmod 664 nomes.txt
```

Comenário: Veja que agora com o usuário dumbledore mesmo o arquivo sendo do root ele consegue alterar a permissão

```
dumbledore@operacao:/srv/precisa$ ls -la
```

```
total 28
```

```
drwxrwx--- 2 root          armada 4096 jan 22 21:56 .
drwxr-xr-x 6 root          root    4096 jan 22 17:20 ..
-rwxr-xr-x 1 root          armada  264 jan 22 17:03 chapeu.sh
-rw-rw-r-- 1 luna         users    30  jan 22 21:46 cidades.txt
-rw-rw-r-- 1 dumbledore   users     0  jan 22 21:56 dumbledore.txt
-rw-rw-rw- 1 luna         users    29  jan 22 20:43 frutas.txt
-rw-rw-r-- 1 root         armada   29  jan 22 17:24 membrosarmada.txt
-rw-r--r-- 1 luna         users     0  jan 22 21:47 mensagem.txt
-rw-rw-r-- 1 root         armada  100  jan 22 20:40 nomes.txt
```

Para essa segunda parte do experimento vamos usar combinados os bits de SGid com Sticky bit assim podemos aplicar em um exemplo pratico

Aplicar a permissão de SGid com a permissão total para o dono e para o grupo e nenhuma permissão para outros de modo recursivo no diretório /srv/sonserina

```
root@operacao:~# chmod -R 2770 /srv/sonserina
```

Criar o diretório /srv/comunal

```
root@operacao:~# #mkdir /srv/comunal
```

Criar o grupo hogwarts

```
root@operacao:~# #addgroup hogwarts
```

Alterar o dono para dumbledore e o grupo para hogwarts do diretório /srv/comunal

```
root@operacao:~# #chown dumbledore:hogwarts /srv/comunal
```

Alterar a permissão do diretório para total para dono, grupo e outros com o sticky bit ativado

```
root@operacao:~# #chmod 1777 /srv/comunal
```

Assim aplicar a permissão de SGid e Sticky bit com a permissão total para o dono e para o grupo e nenhuma permissão para outros de modo recursivo em /srv/precisa, /srv/sonserina, /srv/grifinoria e /srv/corvinal

```
root@operacao:~# chmod -R 3770 /srv/precisa
```

```
root@operacao:~# chmod -R 3770 /srv/grifinoria
```

```
root@operacao:~# chmod -R 3770 /srv/corvinal
```

Tudo ajustado

```
root@operacao:~# ls -lha /srv/
```

total 28K

drwxr-xr-x	7	root	root	4,0K	jan 22 22:41	.
drwxr-xr-x	20	root	root	4,0K	dez 12 14:41	..
drwxrwxrwt	2	dumbledore	hogwarts	4,0K	jan 22 22:41	comunal
drwxrws--T	2	root	corvinal	4,0K	jan 22 17:20	corvinal
drwxrws--T	2	root	grifinoria	4,0K	jan 22 22:32	grifinoria
drwxrws--T	2	root	armada	4,0K	jan 22 21:56	precisa
drwxrws---	2	root	sonserina	4,0K	jan 22 17:20	sonserina

```
root@operacao:~# ls -lhaR /srv/
```

/srv/:

total 28K

drwxr-xr-x	7	root	root	4,0K	jan 22 22:41	.
drwxr-xr-x	20	root	root	4,0K	dez 12 14:41	..
drwxrwxrwt	2	dumbledore	hogwarts	4,0K	jan 22 22:41	comunal
drwxrws--T	2	root	corvinal	4,0K	jan 22 17:20	corvinal
drwxrws--T	2	root	grifinoria	4,0K	jan 22 22:32	grifinoria
drwxrws--T	2	root	armada	4,0K	jan 22 21:56	precisa

```
drwxrws--- 2 root          sonserina  4,0K jan 22 17:20 sonserina
```

```
/srv/comunal:
```

```
total 8,0K
```

```
drwxrwxrwt 2 dumbledore hogwarts 4,0K jan 22 22:41 .
```

```
drwxr-xr-x 7 root          root       4,0K jan 22 22:41 ..
```

```
/srv/corvinal:
```

```
total 8,0K
```

```
drwxrws--T 2 root corvinal 4,0K jan 22 17:20 .
```

```
drwxr-xr-x 7 root root      4,0K jan 22 22:41 ..
```

```
/srv/grifinoria:
```

```
total 12K
```

```
drwxrws--T 2 root  grifinoria 4,0K jan 22 22:32 .
```

```
drwxr-xr-x 7 root  root       4,0K jan 22 22:41 ..
```

```
-rw-r--r-- 1 harry grifinoria 17 jan 22 22:32 harry.txt
```

```
/srv/precisa:
```

```
total 28K
```

```
drwxrws--T 2 root          armada 4,0K jan 22 21:56 .
```

```
drwxr-xr-x 7 root          root    4,0K jan 22 22:41 ..
```

```
-rwxrws--T 1 root          armada 264 jan 22 17:03 chapeu.sh
```

```
-rwxrws--T 1 luna          users   30 jan 22 21:46 cidades.txt
```

```
-rwxrws--T 1 dumbledore users     0 jan 22 21:56 dumbledore.txt
```

```
-rwxrws--T 1 luna          users   29 jan 22 20:43 frutas.txt
```

```
-rwxrws--T 1 root          armada 29 jan 22 17:24 membrosarmada.txt
```

```
-rwxrws--T 1 luna          users     0 jan 22 21:47 mensagem.txt
```

```
-rwxrws--T 1 root          armada 100 jan 22 20:40 nomes.txt
```

```
/srv/sonserina:
```

```
total 8,0K
```

```
drwxrws--- 2 root sonserina 4,0K jan 22 17:20 .
```

```
drwxr-xr-x 7 root root      4,0K jan 22 22:41 ..
```

```
root@operacao:~# tree /srv/
```

```
/srv/  
├─ comunal  
├─ corvinal  
├─ grifinoria  
|   └─ harry.txt  
├─ precisa  
|   ├─ chapeu.sh  
|   ├─ cidades.txt  
|   ├─ dumbledore.txt  
|   ├─ frutas.txt  
|   ├─ membrosarmada.txt  
|   ├─ mensagem.txt  
|   └─ nomes.txt  
└─ sonserina
```

Vamos agora validar o trabalho

Validando o SGid

Em outro terminal logar com usuário severo

```
sysadmin@operacao:~$ su - severo
```

Tente entrar em outros diretórios que não tem permissão para validar

```
severo@operacao:~$ cd /srv/corvinal/  
-bash: cd: /srv/corvinal/: Permissão negada  
severo@operacao:~$ cd /srv/precisa/  
-bash: cd: /srv/precisa/: Permissão negada  
  
severo@operacao:~$ cd /srv/grifinoria/  
-bash: cd: /srv/grifinoria/: Permissão negada  
severo@operacao:~$ cd /srv/sonserina/  
severo@operacao:/srv/sonserina$ ls
```

Criando um arquivo texto severo.txt

```
severo@operacao:/srv/sonserina$ echo "sou o principe mestiço" >  
severo.txt
```

Conferindo a criação do arquivo severo.txt

```
severo@operacao:/srv/sonserina$ ls -la
total 12
drwxrws--- 2 root    sonserina 4096 jan 22 23:11 .
drwxr-xr-x 7 root    root      4096 jan 22 22:41 ..
-rw-r--r-- 1 severo sonserina   24 jan 22 23:11 severo.txt
```

Comentário:

Podemos verificar que a permissão SGid funcionou, pois o arquivo criado por severo tem ele como dono, mas o grupo é o do diretório e não o dele que é o users

Faça logoff do usuário severo

```
severo@operacao:/srv/sonserina$ exit
sair
```

Efetuar o login com o usuário malfoy

```
sysadmin@operacao:~$ su - malfoy
```

Entra no diretório /srv/sonserina e verifica se o arquivo está lá

```
malfoy@operacao:~$ cd /srv/sonserina/
malfoy@operacao:/srv/sonserina$ ls
severo.txt
```

Depois tente apagar o arquivo criado pelo severo.txt

```
malfoy@operacao:/srv/sonserina$ rm severo.txt
rm: remover arquivo comum 'severo.txt' protegido contra escrita? s
malfoy@operacao:/srv/sonserina$ ls
malfoy@operacao:/srv/sonserina$
```

Observe que o programa dá o aviso que o documento está protegido contra escrita, mesmo assim apagou o arquivo. Porque a permissão de SGid somente vai garantir que o grupo dos novos arquivos seja o mesmo do diretório pai.

Testando o Sticky bit

Efetuar o login como luna

```
sysadmin@operacao:~$ su - luna
```

Entrar no diretório /srv/comunal que tem o sticky bit ativado

```
luna@operacao:~$ cd /srv/comunal/
```

Gere um arquivo texto com a mensagem "afasta os nargoles" guarde no arquivo luna.txt

```
luna@operacao:/srv/comunal$ echo "afastar os nargoles" > luna.txt
```

```
luna@operacao:/srv/comunal$ ls -lha
```

```
total 12K
```

```
drwxrwxrwt 2 dumbledore hogwarts 4,0K jan 22 23:21 .
```

```
drwxr-xr-x 7 root          root      4,0K jan 22 22:41 ..
```

```
-rw-r--r-- 1 luna          users      20 jan 22 23:21 luna.txt
```

Faça logoff da usuária luna

```
luna@operacao:/srv/comunal$ exit
```

```
sair
```

Faça login com a usuária cho

```
sysadmin@operacao:~$ su - cho
```

Entrar no diretório /srv/comunal

```
cho@operacao:~$ cd /srv/comunal
```

Tente apagar o arquivo da luna.txt

```
cho@operacao:/srv/comunal$ rm luna.txt
```

```
rm: remover arquivo comum 'luna.txt' protegido contra escrita? s
```

```
rm: não foi possível remover 'luna.txt': Operação não permitida
```

Criar o arquivo com a mensagem "eu sou a Cho" e guarde em cho.txt

```
cho@operacao:/srv/comunal$ echo "eu sou a Cho" > cho.txt
```

Faça o logoff da usuária cho

```
cho@operacao:/srv/comunal$ exit
```

Faça login com a usuária hermione


```
sysadmin@operacao:~$ su - hermione
```

Tente apagar todos os arquivos do diretório /srv/comunal

```
hermione@operacao:$cd /srv/comunal
hermione@operacao:/srv/comunal$ rm *
rm: remover arquivo comum 'cho.txt' protegido contra escrita? s
rm: não foi possível remover 'cho.txt': Operação não permitida
rm: remover arquivo comum 'luna.txt' protegido contra escrita? s
rm: não foi possível remover 'luna.txt': Operação não permitida
hermione@operacao:/srv/comunal$ echo "você não lêem não?" >
hermione.txt
hermione@operacao:/srv/comunal$ ls -la
total 20
drwxrwxrwt 2 dumbledore hogwarts 4096 jan 22 23:25 .
drwxr-xr-x 7 root          root    4096 jan 22 22:41 ..
-rw-r--r-- 1 cho          users    13 jan 22 23:23 cho.txt
-rw-r--r-- 1 hermione     users    23 jan 22 23:25 hermione.txt
-rw-r--r-- 1 luna        users    20 jan 22 23:21 luna.txt
```

Agora vamos analisar a aplicação SGid com Sticky bit ativados

Faça login com usuário harry

```
sysadmin@operacao:~$ su - harry
```

Entre no diretório /srv/grifinoria

```
harry@operacao:~$ cd /srv/grifinoria
harry@operacao:/srv/grifinoria$ vim harry.txt
Sou Harry Potter
Hermione e Ronald são meus amigos
<salvar e sair>
```

Faça logoff no usuário harry

```
harry@operacao:/srv/grifinoria$ exit
sair
```

Faça login no usuário hermione

```
sysadmin@operacao:~$ su - hermione
hermione@operacao:/srv/grifinoria$ id
```

```
uid=1003(hermione)                                gid=100(users)
grupos=100(users),20(dialout),24(cdrom),25(floppy),29(audio),44(vi
deo),46(plugdev),1002(armada),1003(grifinoria)
```

Entrar e em seguida listar o conteúdo do diretório /srv/grifinoria

```
hermione@operacao:~$ cd /srv/grifinoria
hermione@operacao:/srv/grifinoria$ ls -lha
total 12K
drwxrws--T 2 root  grifinoria 4,0K jan 22 23:30 .
drwxr-xr-x 7 root  root      4,0K jan 22 22:41 ..
-rw-r--r-- 1 harry grifinoria  52 jan 22 23:28 harry.txt
hermione@operacao:/srv/grifinoria$ cat harry.txt
Sou Harry Potter
Hermione e Ronald são meus amigos
hermione@operacao:/srv/grifinoria$ vim harry.txt
```

Tente apagar o arquivo do diretório

```
hermione@operacao:/srv/grifinoria$ rm *
rm: remover arquivo comum 'harry.txt' protegido contra escrita? s
rm: não foi possível remover 'harry.txt': Operação não permitida
```

Comentário:

Observe que quando foi aplicado as duas permissões de SGid e Sticky bit o dono sempre será o usuário que cria o arquivo e o grupo será igual ao grupo do diretório e somente o dono pode apagar o arquivo. Claro excluindo o root que pode tudo.