

Отчет по лабораторной работе 3

Генералов Даниил, НПИбд-01-21, 1032202280

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	14
5	Контрольные вопросы	15

Список иллюстраций

3.1	man-страницы	7
3.2	Создание папок	8
3.3	Вход под пользователем bob	8
3.4	Файлы пользователя alice	9
3.5	Удаление файлов alice	9
3.6	Установка sticky-бита	10
3.7	Новые файлы alice	10
3.8	Установка прав доступа с помощью setfacl	11
3.9	Установка прав доступа с помощью setfacl	12
3.10	Удаление и запись в файлы	13
5.1	chown -> chgrp	15
5.2	find	15
5.3	chmod	16
5.4	chmod	16
5.5	chmod	17
5.6	chmod	17
5.7	setfacl	18
5.8	setfacl	19
5.9	umask	19
5.10	chattr	20

Список таблиц

1 Цель работы

В рамках этой лабораторной работы требуется выполнить действия по управлению правами доступа к файлам и папкам.

2 Задание

1. Прочитайте справочное описание man по командам ls, su, chgrp, chmod, getfacl, setfacl.
2. Выполните действия по управлению базовыми разрешениями для групп пользователей (раздел 3.3.1).
3. Выполните действия по управлению специальными разрешениями для групп пользователей (раздел 3.3.2).
4. Выполните действия по управлению расширенными разрешениями с использованием списков ACL для групп пользователей (раздел 3.3.3).

3 Выполнение лабораторной работы

Сначала я открыл man-страницы по всем командам, которые были указаны в задании. Это можно увидеть на скриншоте ниже.

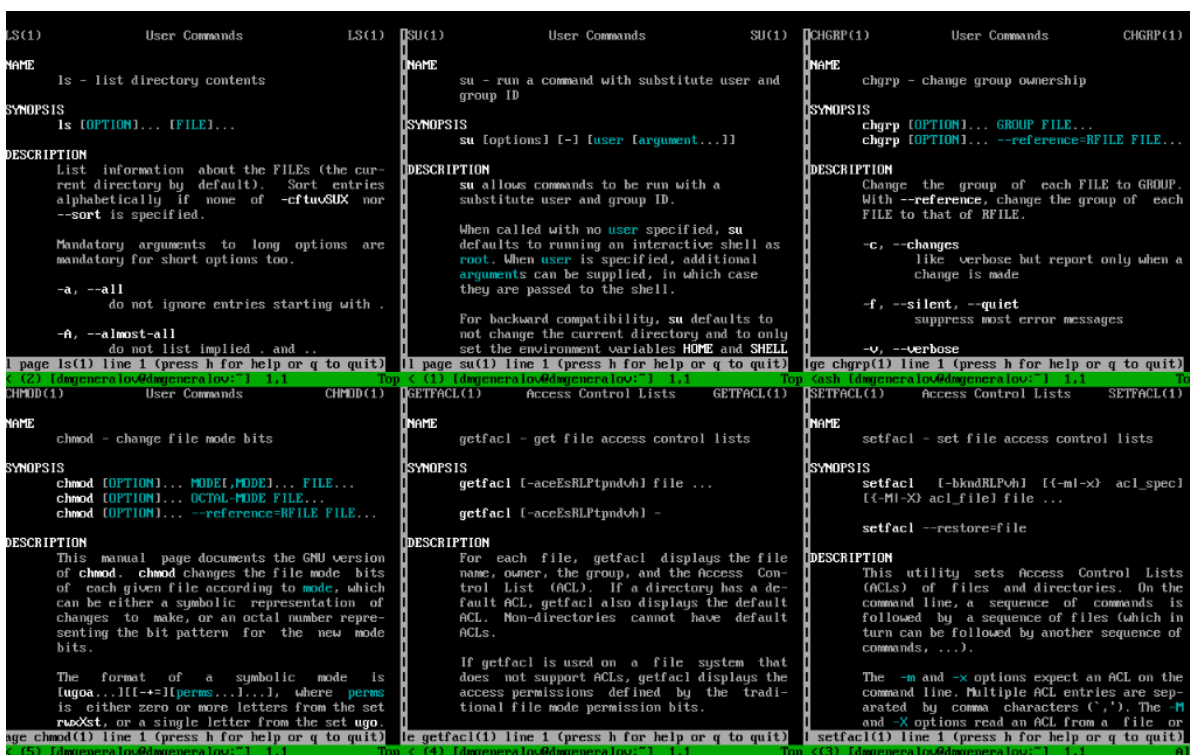


Рис. 3.1: man-страницы

После этого я создал папки /data/main и /data/third, владельцем которых по умолчанию стал root. Затем мы сменили их группы: /data/main теперь стал принадлежать группе main, а /data/third - группе third. Наконец, я добавил группе все разрешения на эти папки, а остальным пользователям – никаких.

```

[dmgeneralov@dmgeneralov ~]$ su -
Password:
Last login: Sat Nov 12 12:24:00 MSK 2022 on tty1
[root@dmgeneralov ~]# mkdir -p /data/main /data/third
[root@dmgeneralov ~]# ls -Al /data
total 0
drwxr-xr-x. 2 root root 6 Nov 18 19:37 main
drwxr-xr-x. 2 root root 6 Nov 18 19:37 third
[root@dmgeneralov ~]# chgrp main /data/main
[root@dmgeneralov ~]# chgrp third /data/third
[root@dmgeneralov ~]# ls -Al /data
total 0
drwxr-xr-x. 2 root main 6 Nov 18 19:37 main
drwxr-xr-x. 2 root third 6 Nov 18 19:37 third
[root@dmgeneralov ~]# chmod 770 /data/main
[root@dmgeneralov ~]# chmod 770 /data/third
[root@dmgeneralov ~]# ls -Al /data
total 0
drwxrwx---. 2 root main 6 Nov 18 19:37 main
drwxrwx---. 2 root third 6 Nov 18 19:37 third
[root@dmgeneralov ~]#

```

Рис. 3.2: Создание папок

После этого я осуществил вход под пользователем bob, который принадлежит к группе main. Из-за этого я смог зайти в директорию /data/main и создать в ней файл, принадлежащий bob и main. Аналогично, я не смог зайти в директорию /data/third или создать в ней файл, так как bob не принадлежит к группе third.

```

Rocky Linux 9.0 (Blue Onyx)
Kernel 5.14.0-70.13.1.el9_0.x86_64 on an x86_64

dmgeneralov login: bob
Password:
[bob@dmgeneralov ~]$ cd /data/main
[bob@dmgeneralov main]$ touch emptyfile
[bob@dmgeneralov main]$ ls -lah
total 0
drwxrwx---. 2 root main 23 Nov 18 19:48 .
drwxr-xr-x. 4 root root 31 Nov 18 19:37 ..
-rw-r--r--. 1 bob bob 0 Nov 18 19:48 emptyfile
[bob@dmgeneralov main]$ cd ../third
-bash: cd: ../third: Permission denied
[bob@dmgeneralov main]$ touch /data/third/emptyfile
touch: cannot touch '/data/third/emptyfile': Permission denied
[bob@dmgeneralov main]$ _

```

Рис. 3.3: Вход под пользователем bob

После этого мы открываем еще один терминал под пользователем alice, и создаем файлы alice1 и alice2 в директории /data/main.


```

Rocky Linux 9.0 (Blue Onyx)
Kernel 5.14.0-70.13.1.el9_0.x86_64 on an x86_64

dmgeneralov login: alice
Password:
Last login: Sat Nov 12 12:26:15 on tty1
[alice@dmgeneralov ~]$ cd /data/main
[alice@dmgeneralov main]$ touch alice1
[alice@dmgeneralov main]$ touch alice2
[alice@dmgeneralov main]$ ls -lha .
total 0
drwxrwx---. 2 root  main  51 Nov 18 19:44 .
drwxr-xr-x. 4 root  root   31 Nov 18 19:37 ..
-rw-r--r--. 1 alice alice   0 Nov 18 19:44 alice1
-rw-r--r--. 1 alice alice   0 Nov 18 19:44 alice2
-rw-r--r--. 1 bob   bob    0 Nov 18 19:40 emptyfile
[alice@dmgeneralov main]$ _

```

Рис. 3.4: Файлы пользователя alice

После этого мы снова входим под пользователем bob, и видим, что эти файлы можно удалить, так как bob принадлежит к группе main и поэтому имеет права редактирования этой директории.

```

[bob@dmgeneralov main]$ ls -lha
total 0
drwxrwx---. 2 root  main  51 Nov 18 19:44 .
drwxr-xr-x. 4 root  root   31 Nov 18 19:37 ..
-rw-r--r--. 1 alice alice   0 Nov 18 19:44 alice1
-rw-r--r--. 1 alice alice   0 Nov 18 19:44 alice2
-rw-r--r--. 1 bob   bob    0 Nov 18 19:40 emptyfile
[bob@dmgeneralov main]$ rm -f alice*
[bob@dmgeneralov main]$ ls -lha
total 0
drwxrwx---. 2 root  main  23 Nov 18 19:46 .
drwxr-xr-x. 4 root  root   31 Nov 18 19:37 ..
-rw-r--r--. 1 bob   bob    0 Nov 18 19:40 emptyfile
[bob@dmgeneralov main]$ touch bob1
[bob@dmgeneralov main]$ touch bob2
[bob@dmgeneralov main]$ ls -lha
total 0
drwxrwx---. 2 root  main  47 Nov 18 19:46 .
drwxr-xr-x. 4 root  root   31 Nov 18 19:37 ..
-rw-r--r--. 1 bob   bob    0 Nov 18 19:46 bob1
-rw-r--r--. 1 bob   bob    0 Nov 18 19:46 bob2
-rw-r--r--. 1 bob   bob    0 Nov 18 19:40 emptyfile
[bob@dmgeneralov main]$

```

Рис. 3.5: Удаление файлов alice

Теперь мы под пользователем root задаем sticky-бит, что запрещает удаление

файлов, если они не принадлежат пользователю, который их удаляет. Мы также устанавливаем `setgid`-бит, из-за которого новые файлы в директории будут принадлежать группе, к которой принадлежит директория, а не группе пользователя, который создал файл.

```
[root@dmgeneralov ~]# chmod g+s,o+t /data/main
[root@dmgeneralov ~]# ls -lha /data
total 0
drwxr-xr-x. 4 root root 31 Nov 18 19:37 .
dr-xr-xr-x. 19 root root 247 Nov 18 19:37 ..
drwxrws--T. 2 root main 47 Nov 18 19:46 main
drwxrwx---. 2 root third 6 Nov 18 19:37 third
[root@dmgeneralov ~]#
```

Рис. 3.6: Установка sticky-бита

Теперь мы создаем файлы `alice3` и `alice4` в директории `/data/main`, и видим, что они принадлежат группе `main`, а не `alice`. Файлы `bob1` и `bob2` также нельзя удалить, потому что они не принадлежат `alice`.

```
[alice@dmgeneralov main]$ touch alice3
[alice@dmgeneralov main]$ touch alice4
[alice@dmgeneralov main]$ ls -lha
total 0
drwxrws--T. 2 root main 75 Nov 18 19:53 .
drwxr-xr-x. 4 root root 31 Nov 18 19:37 ..
-rw-r--r--. 1 alice main 0 Nov 18 19:53 alice3
-rw-r--r--. 1 alice main 0 Nov 18 19:53 alice4
-rw-r--r--. 1 bob bob 0 Nov 18 19:46 bob1
-rw-r--r--. 1 bob bob 0 Nov 18 19:46 bob2
-rw-r--r--. 1 bob bob 0 Nov 18 19:48 emptyfile
[alice@dmgeneralov main]$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
[alice@dmgeneralov main]$ _
```

Рис. 3.7: Новые файлы alice

После этого мы выполняем более тонкую настройку прав доступа к файлам и директориям с помощью команды `setfacl`. В частности, мы добавляем возможность группе `main` читать файлы в директории `/data/third`, а группе `third` читать файлы в директории `/data/main`. Проверив разрешения, мы видим, что обе директории имеют полные права доступа для пользователя `root` и своей группы, а

также права `gx` для другой группы.

Затем мы создали два файла, `/data/main/newfile1` и `/data/third/newfile1`. Файл в `third` принадлежит `root:root`, потому что эта директория не имеет `setgid`-бита, а файл в `main` принадлежит `root:main`, потому что эта директория имеет `setgid`-бит. Помимо этого, файлы по умолчанию доступны для чтения и записи своему пользователю, и для чтения всем остальным.

```
[root@dmgeneralov ~]# setfacl -m g:third:rx /data/main
[root@dmgeneralov ~]# setfacl -m g:main:rx /data/third
[root@dmgeneralov ~]# getfacl /data/main
getfacl: Removing leading '/' from absolute path names
# file: data/main
# owner: root
# group: main
# flags: -st
user::rwx
group::rwx
group:third:r-x
mask::rwx
other::---
default:user::rwx
default:group::rwx
default:group:third:rwx
default:mask::rwx
default:other::---

[root@dmgeneralov ~]# getfacl /data/third
getfacl: Removing leading '/' from absolute path names
# file: data/third
# owner: root
# group: third
user::rwx
group::rwx
group:main:r-x
mask::rwx
other::---
default:user::rwx
default:group::rwx
default:group:main:rwx
default:mask::rwx
default:other::---

[root@dmgeneralov ~]# _
```

Рис. 3.8: Установка прав доступа с помощью `setfacl`

```

[root@dmgeneralov ~]# touch /data/main/newfile1
[root@dmgeneralov ~]# getfacl /data/main/newfile1
getfacl: Removing leading '/' from absolute path names
# file: data/main/newfile1
# owner: root
# group: main
user::rw-
group::r--
other::r--

[root@dmgeneralov ~]# touch /data/main/newfile1
[root@dmgeneralov ~]# touch /data/third/newfile1
[root@dmgeneralov ~]# getfacl /data/third/newfile1
getfacl: Removing leading '/' from absolute path names
# file: data/third/newfile1
# owner: root
# group: root
user::rw-
group::r--
other::r--

[root@dmgeneralov ~]# setfacl -m d:g:third:rwx /data/main
[root@dmgeneralov ~]# setfacl -m d:g:main:rwx /data/third
[root@dmgeneralov ~]# touch /data/main/newfile2
[root@dmgeneralov ~]# getfacl /data/main/newfile2
getfacl: Removing leading '/' from absolute path names
# file: data/main/newfile2
# owner: root
# group: main
user::rw-
group::rwx
group:third:rwx
mask::rw-
other::---
#effective:rw-
#effective:rw-

[root@dmgeneralov ~]# _

```

Рис. 3.9: Установка прав доступа с помощью setfacl

Наконец, мы пробуем удалить, а затем записать в файлы newfile1 и newfile2 от имени пользователя carol. Файл newfile1 не получается удалить или записать, потому что он принадлежит root:root и доступен только для чтения. Файл newfile2 не получается удалить, потому что он находится в директории с установленным sticky-битом, и пользователь carol не является владельцем файла, но он может записать в него, потому что для группы third установлены права на запись с помощью setfacl.

```

[root@dmgeneralov ~]# setfacl -m d:g:third:rwX /data/main
[root@dmgeneralov ~]# setfacl -m d:g:main:rwX /data/third
[root@dmgeneralov ~]# touch /data/main/newfile2
[root@dmgeneralov ~]# getfacl /data/main/newfile2
getfacl: Removing leading '/' from absolute path names
# file: data/main/newfile2
# owner: root
# group: main
user::rw-
group::rwX                    #effective:rw-
group:third:rwX               #effective:rw-
mask::rw-
other::---

[root@dmgeneralov ~]# su -
Last login: Fri Nov 18 19:36:47 MSK 2022 on tty1
[root@dmgeneralov ~]# su - carol
[carol@dmgeneralov ~]$ rm /data/main/newfile1
rm: remove write-protected regular empty file '/data/main/newfile1'? y
rm: cannot remove '/data/main/newfile1': Permission denied
[carol@dmgeneralov ~]$ rm /data/main/newfile2
rm: cannot remove '/data/main/newfile2': Permission denied
[carol@dmgeneralov ~]$ echo "Hello, world" >> /data/main/newfile1
-bash: /data/main/newfile1: Permission denied
[carol@dmgeneralov ~]$ echo "Hello, world" >> /data/main/newfile2
[carol@dmgeneralov ~]$ cat /data/main/newfile1
[carol@dmgeneralov ~]$ cat /data/main/newfile2
Hello, world
[carol@dmgeneralov ~]$

```

Рис. 3.10: Удаление и запись в файлы

4 Выводы

Я получил опыт работы с разрешениями файлов и директорий в Linux.

5 Контрольные вопросы

1. Как следует использовать команду `chown`, чтобы установить владельца группы для файла? Приведите пример.

Можно передать название группы после символа `:`, как показано на скриншоте.

```
[root@dmgeneralov tmp]# touch test1
[root@dmgeneralov tmp]# ls -lha test1
-rw-r--r--. 1 root root 0 Nov 18 20:14 test1
[root@dmgeneralov tmp]# chown :main test1
[root@dmgeneralov tmp]# ls -lha test1
-rw-r--r--. 1 root main 0 Nov 18 20:14 test1
[root@dmgeneralov tmp]#
```

Рис. 5.1: `chown -> chgrp`

2. С помощью какой команды можно найти все файлы, принадлежащие конкретному пользователю? Приведите пример.

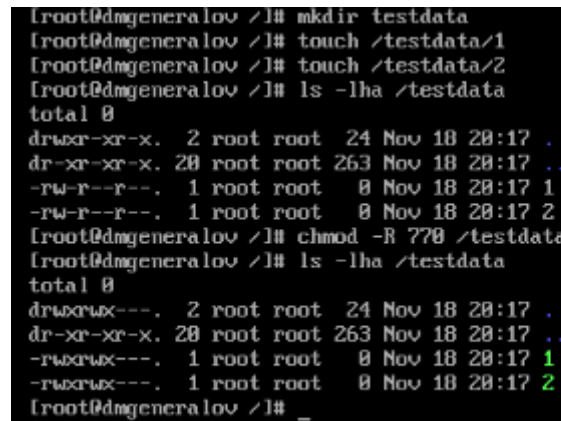
`find / -user <username>`

```
[root@dmgeneralov /]# find -user alice | head
./dev/tty3
./proc/1537
./proc/1537/task
./proc/1537/task/1537
./proc/1537/task/1537/fd
./proc/1537/task/1537/fd/0
./proc/1537/task/1537/fd/1
./proc/1537/task/1537/fd/2
./proc/1537/task/1537/fd/3
./proc/1537/task/1537/fd/4
[root@dmgeneralov /]# ls -lha /dev/tty3
crw--w----. 1 alice tty 4, 3 Nov 18 19:53 /dev/tty3
[root@dmgeneralov /]# _
```

Рис. 5.2: `find`

3. Как применить разрешения на чтение, запись и выполнение для всех файлов в каталоге /data для пользователей и владельцев групп, не устанавливая никаких прав для других? Приведите пример.

```
chmod -R 770 /data
```

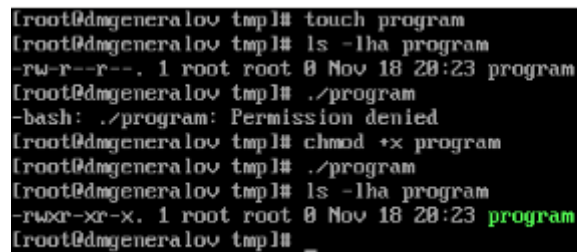


```
[root@dmgeneralov ~]# mkdir testdata
[root@dmgeneralov ~]# touch /testdata/1
[root@dmgeneralov ~]# touch /testdata/2
[root@dmgeneralov ~]# ls -lha /testdata
total 0
drwxr-xr-x. 2 root root 24 Nov 18 20:17 .
dr-xr-xr-x. 20 root root 263 Nov 18 20:17 ..
-rw-r--r--. 1 root root 0 Nov 18 20:17 1
-rw-r--r--. 1 root root 0 Nov 18 20:17 2
[root@dmgeneralov ~]# chmod -R 770 /testdata
[root@dmgeneralov ~]# ls -lha /testdata
total 0
drwxrwx---. 2 root root 24 Nov 18 20:17 .
dr-xr-xr-x. 20 root root 263 Nov 18 20:17 ..
-rwxrwx---. 1 root root 0 Nov 18 20:17 1
-rwxrwx---. 1 root root 0 Nov 18 20:17 2
[root@dmgeneralov ~]# _
```

Рис. 5.3: chmod

4. Какая команда позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым?

```
chmod +x <filename>
```



```
[root@dmgeneralov tmp]# touch program
[root@dmgeneralov tmp]# ls -lha program
-rw-r--r--. 1 root root 0 Nov 18 20:23 program
[root@dmgeneralov tmp]# ./program
-bash: ./program: Permission denied
[root@dmgeneralov tmp]# chmod +x program
[root@dmgeneralov tmp]# ./program
[root@dmgeneralov tmp]# ls -lha program
-rwxr-xr-x. 1 root root 0 Nov 18 20:23 program
[root@dmgeneralov tmp]# _
```

Рис. 5.4: chmod

5. Какая команда позволяет убедиться, что групповые разрешения для всех новых файлов, создаваемых в каталоге, будут присвоены владельцу группы этого каталога? Приведите пример.

`chmod g+s <directory>`

```
(root@dmgeneralov tmp1# mkdir test
(root@dmgeneralov tmp1# chgrp main test
(root@dmgeneralov tmp1# touch test/1
(root@dmgeneralov tmp1# ls -lha test
total 4.0K
drwxr-xr-x. 2 root main 15 Nov 18 20:25 .
drwxrwxrwt. 10 root root 4.0K Nov 18 20:25 ..
-rw-r--r--. 1 root root 0 Nov 18 20:25 1
(root@dmgeneralov tmp1# chmod g+s test
(root@dmgeneralov tmp1# touch test/2
(root@dmgeneralov tmp1# ls -lha test
total 4.0K
drwxr-sr-x. 2 root main 24 Nov 18 20:25 .
drwxrwxrwt. 10 root root 4.0K Nov 18 20:25 ..
-rw-r--r--. 1 root root 0 Nov 18 20:25 1
-rw-r--r--. 1 root main 0 Nov 18 20:25 2
(root@dmgeneralov tmp1# _
```

Рис. 5.5: chmod

6. Необходимо, чтобы пользователи могли удалять только те файлы, владельцами которых они являются, или которые находятся в каталоге, владельцами которого они являются. С помощью какой команды можно это сделать? Приведите пример.

`chmod o+t <directory>`

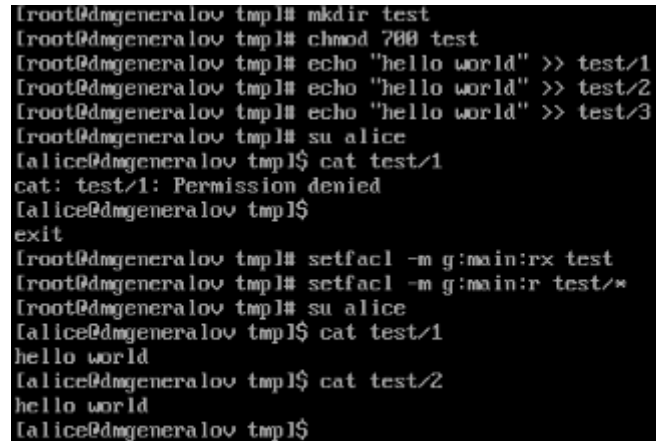
```
(root@dmgeneralov tmp1# mkdir test
(root@dmgeneralov tmp1# chmod 777 test
(root@dmgeneralov tmp1# chmod o+t test
(root@dmgeneralov tmp1# su alice
(alice@dmgeneralov tmp1)$ cd test
(alice@dmgeneralov test)$ touch alice
(alice@dmgeneralov test)$
exit
(root@dmgeneralov tmp1# su bob
(bob@dmgeneralov tmp1)$ cd test
(bob@dmgeneralov test)$ rm alice
rm: remove write-protected regular empty file 'alice'? y
rm: cannot remove 'alice': Operation not permitted
(bob@dmgeneralov test)$ ls -lha .
total 4.0K
drwxrwxrwt. 2 root root 19 Nov 18 20:26 .
drwxrwxrwt. 10 root root 4.0K Nov 18 20:26 ..
-rw-rw-r--. 1 alice alice 0 Nov 18 20:26 alice
(bob@dmgeneralov test)$ _
```

Рис. 5.6: chmod

7. Какая команда добавляет ACL, который предоставляет членам группы права доступа на чтение для всех существующих файлов в текущем каталоге?

```
setfacl -m g:groupname:rx <directory>
```

```
setfacl -m g:groupname:r <directory>/*
```



```
root@dmgeneralov tmp1# mkdir test
root@dmgeneralov tmp1# chmod 700 test
root@dmgeneralov tmp1# echo "hello world" >> test/1
root@dmgeneralov tmp1# echo "hello world" >> test/2
root@dmgeneralov tmp1# echo "hello world" >> test/3
root@dmgeneralov tmp1# su alice
[alice@dmgeneralov tmp1]$ cat test/1
cat: test/1: Permission denied
[alice@dmgeneralov tmp1]$ exit
root@dmgeneralov tmp1# setfacl -m g:main:rx test
root@dmgeneralov tmp1# setfacl -m g:main:r test/*
root@dmgeneralov tmp1# su alice
[alice@dmgeneralov tmp1]$ cat test/1
hello world
[alice@dmgeneralov tmp1]$ cat test/2
hello world
[alice@dmgeneralov tmp1]$
```

Рис. 5.7: setfacl

8. Что нужно сделать для гарантии того, что члены группы получают разрешения на чтение для всех файлов в текущем каталоге и во всех его подкаталогах, а также для всех файлов, которые будут созданы в этом каталоге в будущем? Приведите пример.

```
setfacl -dm g:groupname:rx <directory>
```

```
[root@dmgeneralov tmp]# mkdir test
[root@dmgeneralov tmp]# echo "hello world" >> test/1
[root@dmgeneralov tmp]# setfacl -dm g:main:rx test
[root@dmgeneralov tmp]# mkdir test/test2
[root@dmgeneralov tmp]# echo "hello world" >> test/test2/1
[root@dmgeneralov tmp]# su alice
[alice@dmgeneralov tmp]# ls test
1 test2
[alice@dmgeneralov tmp]# cat test/1
hello world
[alice@dmgeneralov tmp]# cat test/test2/1
hello world
[alice@dmgeneralov tmp]#
```

Рис. 5.8: setfacl

9. Какое значение umask нужно установить, чтобы «другие» пользователи не получали какие-либо разрешения на новые файлы? Приведите пример.

umask 007

```
[root@dmgeneralov tmp]# echo "hello world" >> test1
[root@dmgeneralov tmp]# umask 007
[root@dmgeneralov tmp]# echo "hello world" >> test2
[root@dmgeneralov tmp]# su alice
[alice@dmgeneralov tmp]# cat test1
hello world
[alice@dmgeneralov tmp]# cat test2
cat: test2: Permission denied
[alice@dmgeneralov tmp]# ls -lha test*
-rw-r--r--. 1 root main 12 Nov 18 20:40 test1
-rw-rw----. 1 root root 12 Nov 18 20:40 test2
[alice@dmgeneralov tmp]#
```

Рис. 5.9: umask

10. Какая команда гарантирует, что никто не сможет удалить файл myfile случайно?

chattr +i myfile

```

[root@dmgeneralov ~]# mkdir test
[root@dmgeneralov ~]# chmod 777 test
[root@dmgeneralov ~]# cd test
[root@dmgeneralov test]# touch 1
[root@dmgeneralov test]# touch 2
[root@dmgeneralov test]# touch 3
[root@dmgeneralov test]# chmod 777 *
[root@dmgeneralov test]# chattr +i 1
[root@dmgeneralov test]# chattr +i 2
[root@dmgeneralov test]# ls -lha
total 0
drwxrwxrwx. 2 root root 33 Nov 18 20:45 █
dr-xr-xr-x. 28 root root 259 Nov 18 20:45 ..
-rwxrwxrwx. 1 root root 0 Nov 18 20:45 1
-rwxrwxrwx. 1 root root 0 Nov 18 20:45 2
-rwxrwxrwx. 1 root root 0 Nov 18 20:45 3
[root@dmgeneralov test]# su alice
[alice@dmgeneralov test]$ sl
bash: sl: command not found
[alice@dmgeneralov test]$ ls
1 2 3
[alice@dmgeneralov test]$ rm *
rm: cannot remove '1': Operation not permitted
rm: cannot remove '2': Operation not permitted
[alice@dmgeneralov test]$ ls -lha
total 0
drwxrwxrwx. 2 root root 24 Nov 18 20:45 █
dr-xr-xr-x. 28 root root 259 Nov 18 20:45 ..
-rwxrwxrwx. 1 root root 0 Nov 18 20:45 1
-rwxrwxrwx. 1 root root 0 Nov 18 20:45 2
[alice@dmgeneralov test]$ rm 1
rm: cannot remove '1': Operation not permitted
[alice@dmgeneralov test]$ exit
[root@dmgeneralov test]# rm 1
rm: remove regular empty file '1'? y
rm: cannot remove '1': Operation not permitted
[root@dmgeneralov test]#

```

Рис. 5.10: chattr