Отчёт по лабораторной работе №3

Настройка прав доступа

Акунаева Антонина Эрдниевна

Содержание

1 Цель работы **5**

2 Задание **6**

3 Выполнение лабораторной работы **7**

4 Контрольные вопросы **17**

5 Выводы **23**

Список литературы **24**

Список иллюстраций

3.1 Изменение групп и прав для каталогов 8
3.2 Взаимодействие пользователя с файлами и каталогами из разных групп 9
3.3 Создание файлов в каталоге той же группы, что и пользователь . . 10
3.4 Удаление файлов другого пользователя той же группы 11
3.5 Отображение общей группы пользователя у созданного им файла . 11
3.6 Установка битов идентификатора и sticky-bit для каталога 12
3.7 Удаление чужих файлов при наличии sticky-bit 12
3.8 Изменение прав и получение информации о них через ACL 13
3.9 Создание файлов после взаимодействий с ACL 14
3.10 Установка стандартных прав через ACL 15
3.11 Взаимодействие со старыми и новыми файлами после изменений прав ACL 16

4.1 Контрольный вопрос №1 17
4.2 Контрольный вопрос №2 18
4.3 Контрольный вопрос №3 18
4.4 Контрольный вопрос №5 19
4.5 Контрольный вопрос №6 20
4.6 Контрольный вопрос №8 21
4.7 Контрольный вопрос №9 21

Список таблиц

1 Цель работы

Получение навыков настройки базовых и специальных прав доступа для групп пользователей в операционной системе типа Linux. [1]

2 Задание

1. Прочитайте справочное описание man по командам 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

3.3.1. Управление базовыми разрешениями

Заходим от имени пользователя root в терминал через команду, введя пароль после:

```
su -
```

Создаём каталоги main и third в data и получим информацию о правах пользователей и групп на работу с каталогами:

```
mkdir -p /data/main /data/third
ls - Al /data
```

Изменим группы для этих каталогов на main и third соответственно и снова проверим через ls - убедимся, что третий столбец действительно изменился на новые группы каталогов:

```
chgrp main /data/main
chgrp third /data/third
ls -Al /data
```

Изменим права пользователей и групп для этих каталогов так, чтобы все, кроме “других” (o - others) пользователей могли как угодно обращаться с каталогами и файлами в main и third, установив параметр для chmod - 770, где 7 = rwx = полный доступ, 0 = — = отсутствие прав. Проверим и удостоверимя (рис. 3.1):

```
chmod 770 /data/main
chmod 770 /data/third
ls -Al /data
```

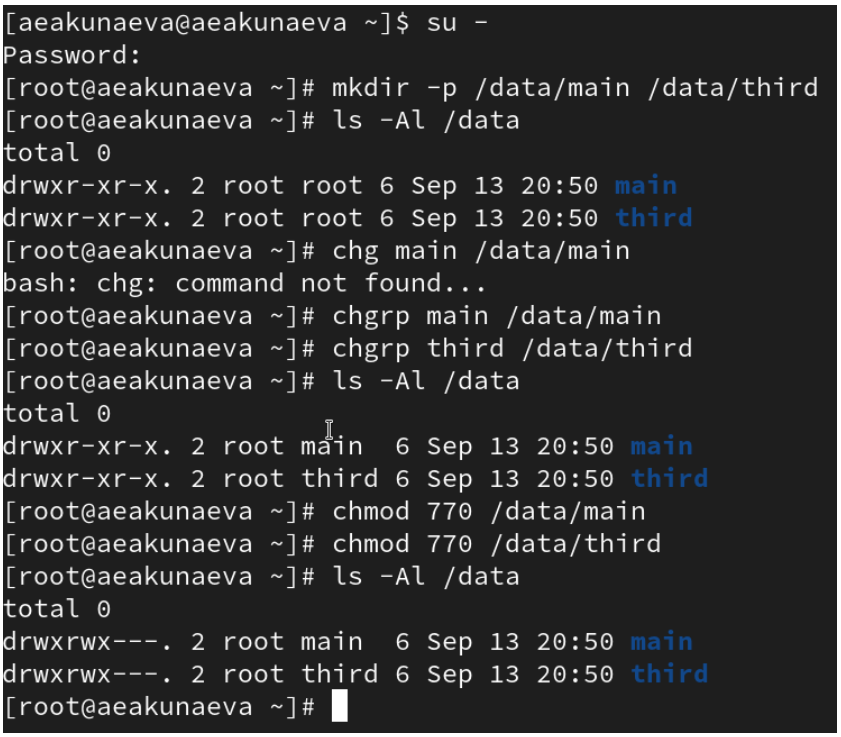

Рис. 3.1: Изменение групп и прав для каталогов

Перейдём во второй терминал и зайдём от пользователя bob из ЛР №2 (рис. 3.2):

```
su - bob
```

Перейдём в новый каталог main и создадим там файл emptyfile, проверив также сведения о правах на него. Так как файл был создан пользователем bob, то группа и владелец (создатель) файла будет bob:

```
cd /data/main
touch emptyfile
ls -Al
```

Однако, если мы попытаемся повторить операцию с каталогом /data/third и создать в нём файл emptyfile2, то получим отказ. Это объясняется тем, что ранее мы установили группу и права на пользование каталогами, и если bob может взаимодействовать с файлами в /data/main, т.к. он состоит в группе main (из ЛР №2), то в third он не сможет делать то же самое:

```
cd /data/third
touch /data/third/emptyfile2
```

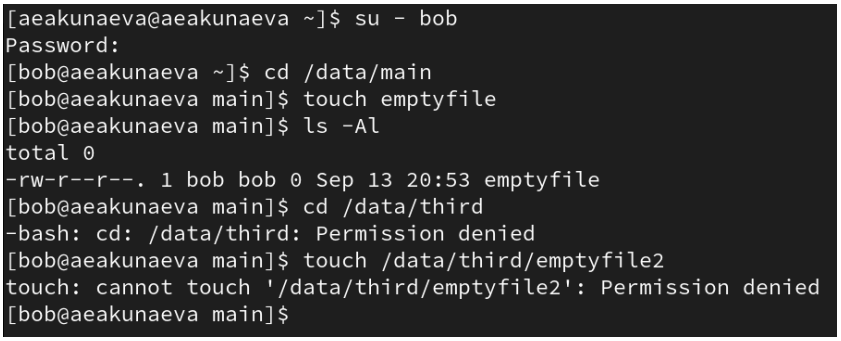

Рис. 3.2: Взаимодействие пользователя с файлами и каталогами из разных групп

3.3.2. Управление специальными разрешениями

В новом терминале зайдём как пользователь alice и перейдём в новый каталог /data/main (рис. 3.3):

```
su - alice
cd /data/main
```

В каталоге создадим файлы alice1 и alice2 и проверим их наличие через ls. Файлы будутсозданы, т.к. alice состоит в main, они будут иметь владельца и группу alice:

```
touch alice1 alice2
ls -Al
```

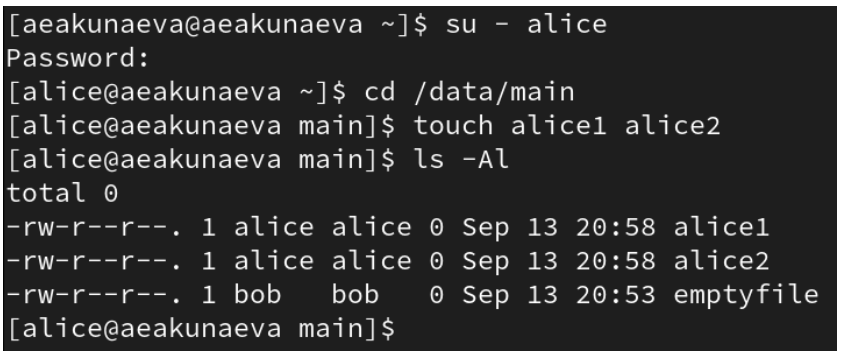

Рис. 3.3: Создание файлов в каталоге той же группы, что и пользователь

Теперь откроем терминал как bob, перейдём в тот же каталог и снова проверим, какие файлы есть в каталоге - действительно, файлы alice1 и alice2 были созданы и имеют ту же инфомрацию о правах и группах (рис. 3.4):

```
su - bob
cd /data/main
ls -l
```

При этом, если мы удалим файлы пользователя alice, указав ключ -f и имя пользователя* со звёздочкой для обозначения владельца файлов и проверив, мы не обнаружим их наличие - т.к. пользователь bob не имеет ограничений на удаление файлов других пользователей. Останется только emptyfile:

```
rm -f alice*
ls -l
```

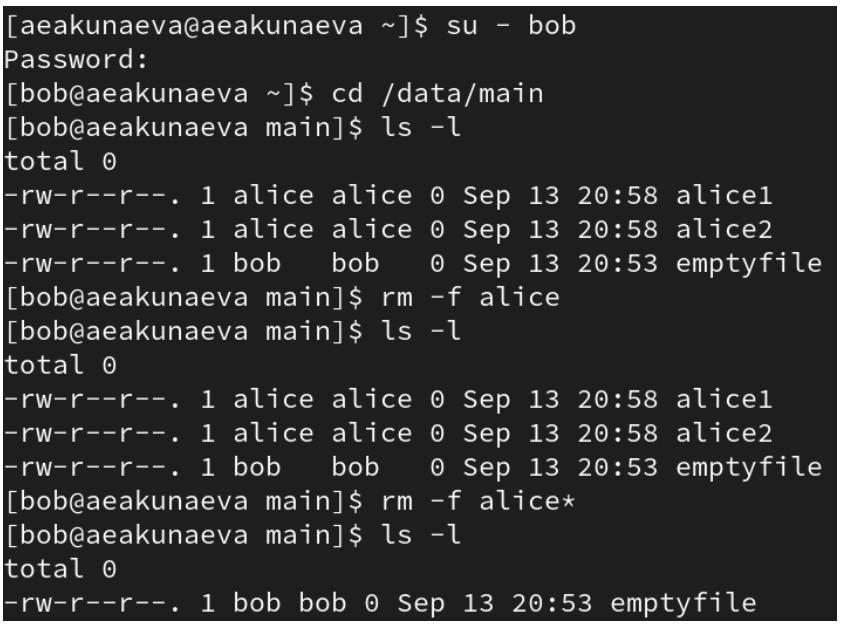

Рис. 3.4: Удаление файлов другого пользователя той же группы

Создадим файлы bob1 и bob2 как bob и проверим. Новые файлы будут иметь группу main (рис. 3.5):

```
touch bob1 bob2
ls -l
```

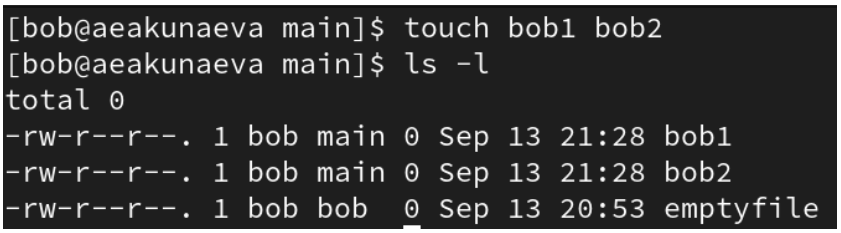

Рис. 3.5: Отображение общей группы пользователя у созданного им файла

От лица пользователя root в прошлом терминале изменим права на каталог /data/main, обозначив его бит идентификатора для группы и sticky-bit для пользователей, указав ключи соответственно +s и +t (рис. 3.6):

```
su -
chmod g+s,o+t /data/main
```

```
[aeakunaeva@aeakunaeva ~]$ su -
Password:
[root@aeakunaeva ~]# chmod g+s,o+t /data/main
```

Рис. 3.6: Установка битов идентификатора и sticky-bit для каталога

Снова зайдём как alice в каталог main и создадим теперь файлы alice3 и alice4, проверим. Файлы также будут теперь иметь группу main (рис. 3.7):

```
su - alice
cd /data/main
touch alice3 alice4
ls -l
```

При этом, попытавшись удалить теперь файлы пользователя bob, мы получим отказ, т.к. sticky-bit запрещает воздействие на чужие файлы, даже если пользователи находятся в одной группе:

```
rm -rf bob*
```

```
[bob@aeakunaeva main]$ su - alice
Password:
[alice@aeakunaeva ~]$ cd /data/main
[alice@aeakunaeva main]$ touch alice3 alice4
[alice@aeakunaeva main]$ ls -l
total 0
-rw-r--r--. 1 alice main 0 Sep 13 21:30 alice3
-rw-r--r--. 1 alice main 0 Sep 13 21:30 alice4
-rw-r--r--. 1 bob   main 0 Sep 13 21:28 bob1
-rw-r--r--. 1 bob   main 0 Sep 13 21:28 bob2
-rw-r--r--. 1 bob   bob  0 Sep 13 20:53 emptyfile
[alice@aeakunaeva main]$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
[alice@aeakunaeva main]$
```

Рис. 3.7: Удаление чужих файлов при наличии sticky-bit

3.3.3. Управление расширенными разрешениями с использованием списков ACL

Как root-пользователь, установим права на чтение и выполнение для каталогов main и third как для групп third и main соответственно, указав ключ -m для внесения изменений в права командой setfacl и проверим результат через getfacl - действительно, у каталогов будет подпункт с группами с названиями друг друга и установка прав r-x (рис. 3.8):

```
su -
setfacl -m g:third:rx /data/main
setfacl -m g:main:rx /data/third
getfacl /data/main
```

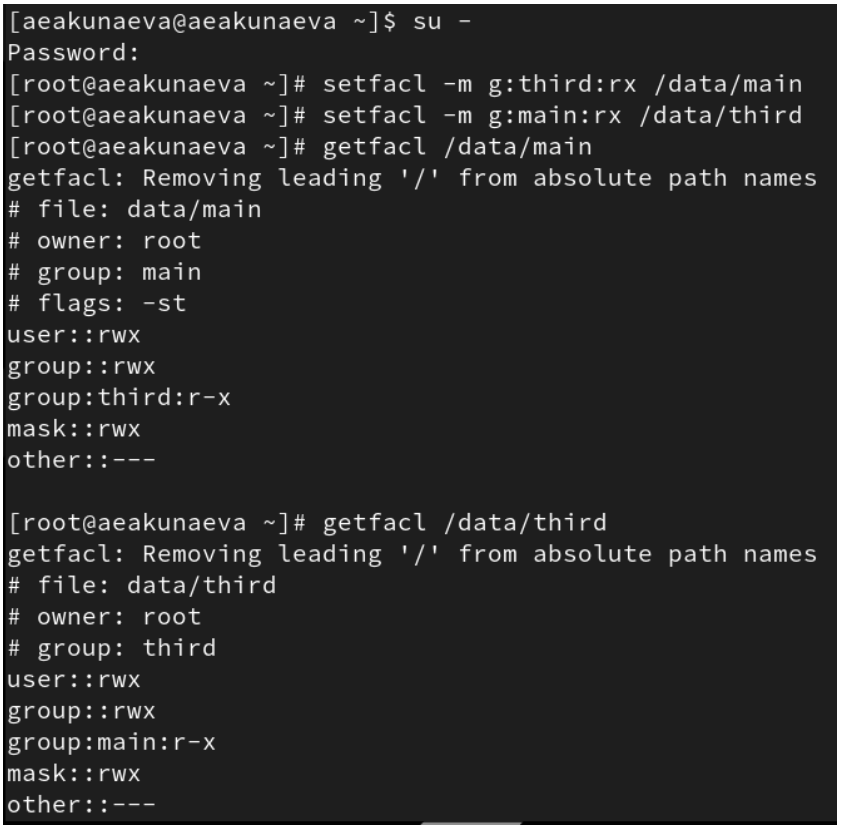

Рис. 3.8: Изменение прав и получение информации о них через ACL

Создадим новый файл в каталоге main, проверим его данные о правах и группах - пользователем будет являться root (т.к. мы его создавали), а в группе будет

указана main. Повторив то же самое для каталога third, получим, что группа будет также root, как и владелец, т.к. мы не изменяли данных для third (рис. 3.9):

```
touch /data/main/newfile1
getfacl /data/main/newfile1

touch /data/third/newfile2
getfacl /data/third/newfile2
```

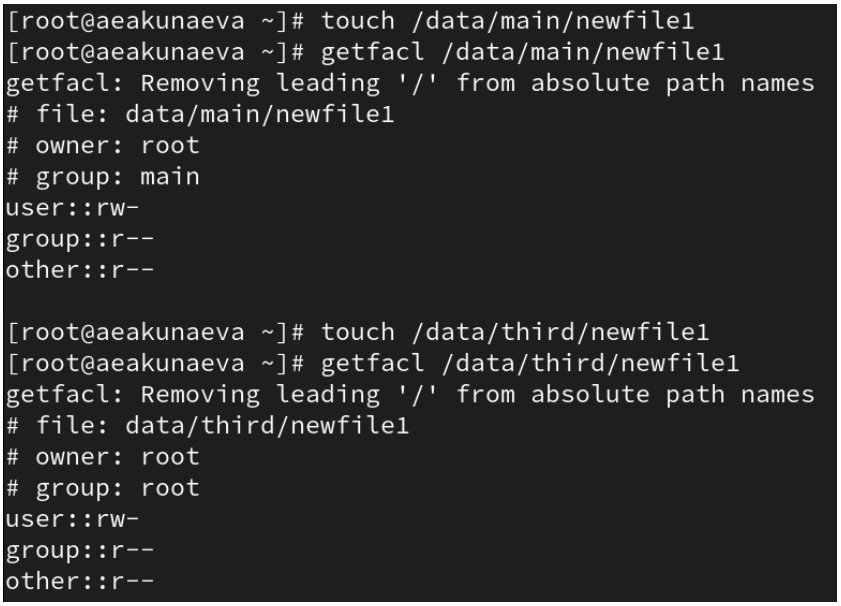

Рис. 3.9: Создание файлов после взаимодействий с ACL

Установим теперь для этих каталогов базовые значения для групп (default - d) - rwx, создадим в каждом из них файлы и проверим данные новых файлов. Заметим, что добавится у новых файлов пункт на права для групп third и main соответственно с полным набором rwx, т.к. базовая установка позволяет влиять на новые файлы и каталоге, созданные в этом каталоге (рис. 3.10):

```
setfacl -m d:g:third:rwx /data/main
setfacl -m d:g:main:rwx /data/third
touch /data/main/newfile2
```

```
getfacl /data/main/newfile2
touch /data/third/newfile2
getfacl /data/third/newfile2
```

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

Рис. 3.10: Установка стандартных прав через ACL

Теперь зайдём как пользователь carol из группы third. Попытавшись удалить файлы newfile1-2, мы получим отказ, т.к. sticky-bit не позволяет удалить файлы, не являясь их владельцем. При этом, попытавшись сделать запись в файл newfile1 каталога main, столкнёмся также с запретом, хотя записать в newfile2 выйдет - т.к. второй файл был создан после установки новых прав, пользователи группы third могут на него воздействовать, тогда как для старого файла newfile1 права не изменились (рис. 3.11):

```
su - carol
rm /data/main/newfile1 /data/main/newfile2
ls /data/main
```

```
echo "Hello, world" >> /data/main/newfile1
echo "Hello, world" >> /data/main/newfile2
cat /data/main/newfile2
```

Рис. 3.11: Взаимодействие со старыми и новыми файлами после изменений прав ACL

4 Контрольные вопросы

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

Команду chown нужно использовать по схеме (рис. 4.1):

```
chown [group] [file/catalogue]
```

- где group - новый владелец группы, а затем указан файл или каталог, для которого он меняется.

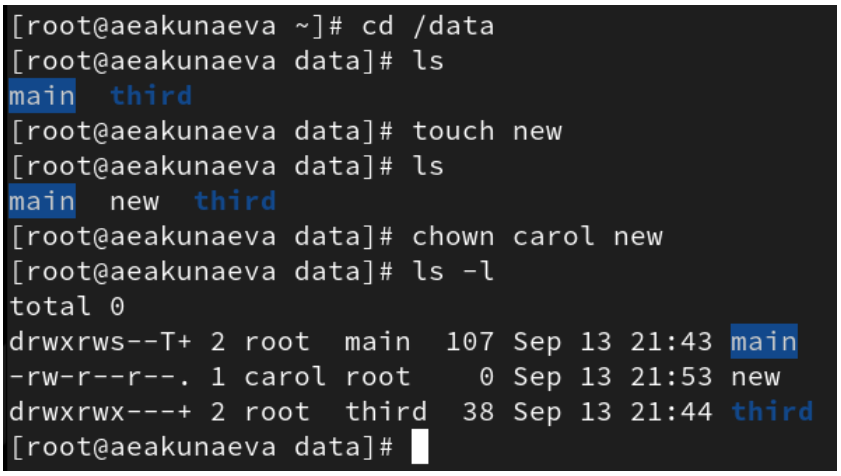

Рис. 4.1: Контрольный вопрос №1

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

При помощи команды (рис. 4.2):

```
find -user [username]
```

- где username - имя пользователя, чьи файлы нужно найти, а -user - ключ, для обозначения имени пользователя как ключа поиска.

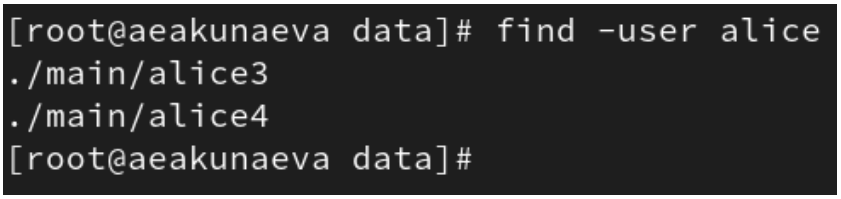

Рис. 4.2: Контрольный вопрос №2

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.**

Благодаря команде chmod можно изменить права для файла или каталога /data, установив значение 770 (где 7 - rwx, полный набор прав, а 0 - отсутсвие прав) (рис. 4.3):

```
chmod 770 test
ls -l
```

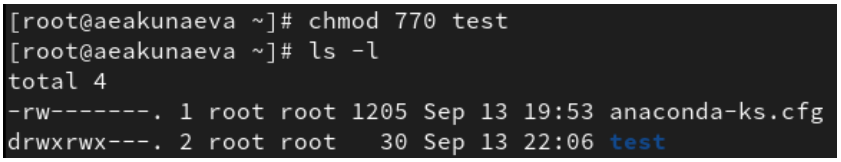

Рис. 4.3: Контрольный вопрос №3

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

При помощи команды chmod можно поставить параметр x для файла, чтобы сделать его исполняемым для какой-либо группы или пользователя:

```
chmod x [file/catalogue]
```

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

Установив бит идентификатора владельца группы для каталога через ключ +s для параметра g (group) (что будет видно при проверке getfacl в строке flags: -s-) (рис. 4.4):

```
chmod g+s test
```

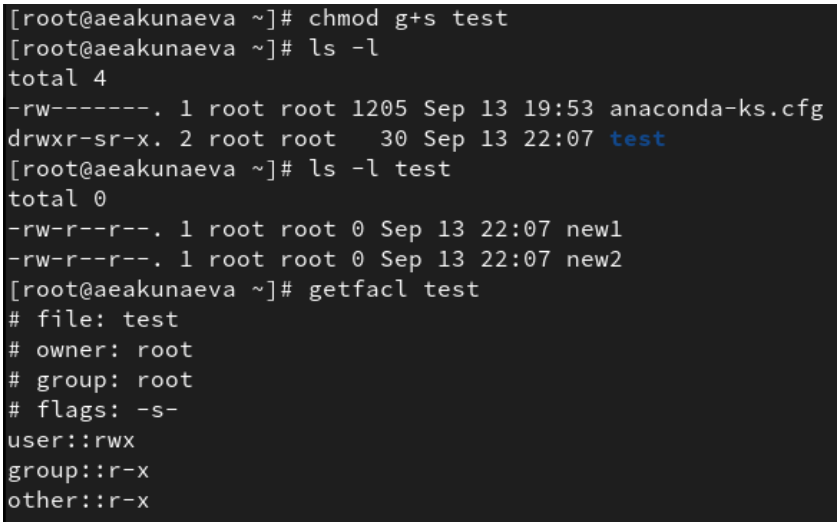

Рис. 4.4: Контрольный вопрос №5

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.**

Чтобы пользователи удаляли только свои файлы,нужно поставить sticky-bit для каталога через ключ +t (будет отображаться как –t или подобное во flags после проверки getfacl) (рис. 4.5):

```
chmod o+t test
```

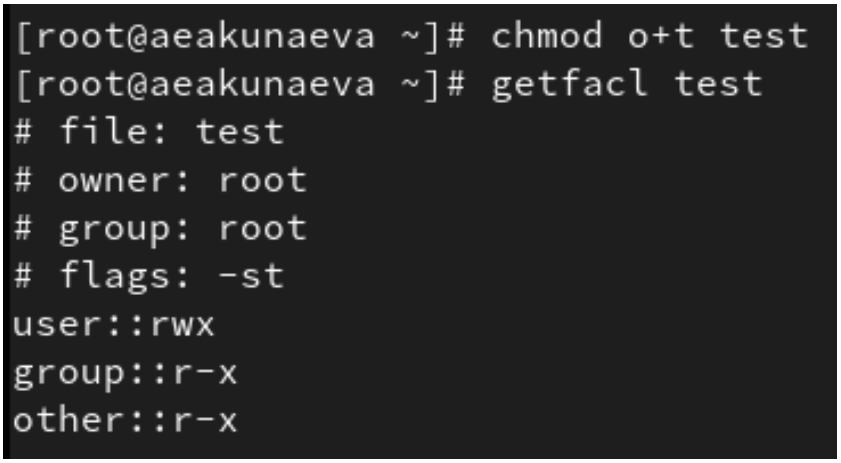

Рис. 4.5: Контрольный вопрос №6

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

```
setfacl -m g:[groupname]:r [catalogue/.]
```

8. **Что нужно сделать для гарантии того, что члены группы получат 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.**

Чтобы все будущие файлы и каталоги в текущем каталоге сохраняли те же права доступа, нужно установить бит идентификатора группы для него (рис. 4.6):

```
chmod g+s test
```

```
[root@aeakunaeva ~]# chmod g+s test
[root@aeakunaeva ~]# getfacl test
# file: test
# owner: root
# group: root
# flags: -st
user::rwx
group::r-x
other::r-x
```

Рис. 4.6: Контрольный вопрос №8

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

Можно установить значение umask на 077 (где 7 - rwx, 0 - —) (рис. 4.7):

```
umask 077
```

```
[root@aeakunaeva ~]# umask 077
[root@aeakunaeva ~]# umask
0077
[root@aeakunaeva ~]# touch test/new4
[root@aeakunaeva ~]# getfacl test/new4
# file: test/new4
# owner: root
# group: root
user::rw-
group::---
other::---
```

Рис. 4.7: Контрольный вопрос №9

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

Команда chmod с указанием параметра -r или -w, но не -x:

```
chmod -rw myfile
```

5 Выводы

Я получила навыки настройки базовых и специальных прав доступа для групп пользователей в операционной системе типа Linux.

Список литературы

1. Кулябов Д.С. Настройка прав доступа. ТУИС РУДН.