Отчёт по лабораторной работе №3

Настройка прав доступа

Амина Аджигалиева

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Ход выполнения работы

2.1 Управление базовыми разрешениями

Сначала я вошла под суперпользователем с помощью команды `su`. Затем в каталоге `/data` были созданы две директории: **main** и **third**. Проверка их владельцев показала, что они принадлежат пользователю `root`.

```
aradzhigalieva@aradzhigalieva:~$ su
Password:
root@aradzhigalieva:/home/aradzhigalieva# mkdir -p /data/main /data/third
root@aradzhigalieva:/home/aradzhigalieva# ls -Al /data
total 0
drwxr-xr-x. 2 root root 6 Sep 12 13:26 main
drwxr-xr-x. 2 root root 6 Sep 12 13:26 third
root@aradzhigalieva:/home/aradzhigalieva# chgrp main /data/main/
root@aradzhigalieva:/home/aradzhigalieva# chgrp third /data/third/
root@aradzhigalieva:/home/aradzhigalieva# ls -Al /data/
total 0
drwxr-xr-x. 2 root main  6 Sep 12 13:26 main
drwxr-xr-x. 2 root third 6 Sep 12 13:26 third
root@aradzhigalieva:/home/aradzhigalieva# chmod 770 /data/main/
root@aradzhigalieva:/home/aradzhigalieva# chmod 770 /data/third/
root@aradzhigalieva:/home/aradzhigalieva# ls -Al /data/
total 0
drwxrwx---. 2 root main  6 Sep 12 13:26 main
drwxrwx---. 2 root third 6 Sep 12 13:26 third
root@aradzhigalieva:/home/aradzhigalieva# su bob
bob@aradzhigalieva:/home/aradzhigalieva$ cd /data/main/
bob@aradzhigalieva:/data/main$ touch emptyfile
bob@aradzhigalieva:/data/main$ ls -Al
total 0
-rw-r--r--. 1 bob bob 0 Sep 12 13:29 emptyfile
bob@aradzhigalieva:/data/main$ cd /data/third/
bash: cd: /data/third/: Permission denied
bob@aradzhigalieva:/data/main$
```

Рис. 2.1: Создание каталогов main и third

Далее я изменила группы владельцев: каталог **/data/main** был передан группе `main`, а каталог **/data/third** — группе `third`. После этого установлены права до-

ступа `770`, что позволяет только владельцу и его группе работать с каталогами. Другие пользователи не имеют доступа.

Затем я перешла под пользователя **bob** и вошла в каталог **/data/main**. Здесь удалось создать файл *emptyfile*, что подтверждает наличие прав у группы `main`.

После этого была предпринята попытка доступа к каталогу **/data/third**. Однако система вернула ошибку *Permission denied*, так как пользователь `bob` не входит в группу `third` и не имеет прав на работу с этим каталогом.

2.2 Управление специальными разрешениями

Сначала я перешла под пользователем **alice** и в каталоге `/data/main` создала два файла — *alice1* и *alice2*.
Затем вошла под пользователем **bob**, который также состоит в группе `main`. В каталоге `/data/main` он смог просмотреть и удалить файлы, созданные пользователем **alice**.

```
alice@aradzhigalieva:/data/main$ touch alice1
alice@aradzhigalieva:/data/main$ touch alice2
alice@aradzhigalieva:/data/main$
exit
bob@aradzhigalieva:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice alice 0 Sep 12 13:33 alice1
-rw-r--r--. 1 alice alice 0 Sep 12 13:33 alice2
-rw-r--r--. 1 bob   bob   0 Sep 12 13:29 emptyfile
bob@aradzhigalieva:/data/main$ rm -f alice*
bob@aradzhigalieva:/data/main$ ls -l
total 0
-rw-r--r--. 1 bob bob 0 Sep 12 13:29 emptyfile
bob@aradzhigalieva:/data/main$ touch bob1 bob2
bob@aradzhigalieva:/data/main$ su
Password:
root@aradzhigalieva:/data/main# chmod g+s,o+t /data/main/
root@aradzhigalieva:/data/main# su alice
alice@aradzhigalieva:/data/main$ touch alice3
alice@aradzhigalieva:/data/main$ touch alice4
alice@aradzhigalieva:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice main 0 Sep 12 13:34 alice3
-rw-r--r--. 1 alice main 0 Sep 12 13:34 alice4
-rw-r--r--. 1 bob   bob  0 Sep 12 13:34 bob1
-rw-r--r--. 1 bob   bob  0 Sep 12 13:34 bob2
-rw-r--r--. 1 bob   bob  0 Sep 12 13:29 emptyfile
alice@aradzhigalieva:/data/main$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
alice@aradzhigalieva:/data/main$
```

Рис. 2.2: Создание и удаление файлов alice пользователем bob

После этого под пользователем **bob** были созданы собственные файлы *bob1* и *bob2*.

Затем, вернувшись в терминал суперпользователя, я установила для каталога `/data/main` специальные разрешения: бит идентификатора группы (SGID) и sticky-bit.

Снова войдя под пользователем **alice**, я создала новые файлы *alice3* и *alice4*. При проверке содержимого каталога стало видно, что эти файлы принадлежат группе `main`.

При попытке удалить файлы *bob1* и *bob2* доступ был запрещён, так как sticky-bit

защищает файлы пользователей от удаления другими участниками группы. В то же время **alice** может удалять только собственные файлы.

2.3 Управление расширенными разрешениями с использованием списков ACL

Для начала я вошла в систему под суперпользователем и установила дополнительные права: группе **third** были даны разрешения на чтение и выполнение в каталоге `/data/main`, а группе **main** — аналогичные разрешения в каталоге `/data/third`. Проверка с помощью команды `getfacl` показала, что права назначены корректно.

```
alice@aradzhigalieva:/data/main$ su
Password:
root@aradzhigalieva:/data/main# setfacl -m g:third:rx /data/main/
root@aradzhigalieva:/data/main# setfacl -m g:main:rz /data/third/
setfacl: Option -m: Invalid argument near character 9
root@aradzhigalieva:/data/main# setfacl -m g:main:rx /data/third/
root@aradzhigalieva:/data/main# 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::---

root@aradzhigalieva:/data/main# 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::---

root@aradzhigalieva:/data/main#
```

Рис. 2.3: Установка ACL для каталогов main и third

Затем я создала новые файлы *newfile1* в каталогах `/data/main` и `/data/third`.

Анализ прав показал, что по умолчанию для них установлены минимальные разрешения (только для владельца). Это объясняется тем, что при отсутствии правил наследования ACL новые объекты получают стандартные права.

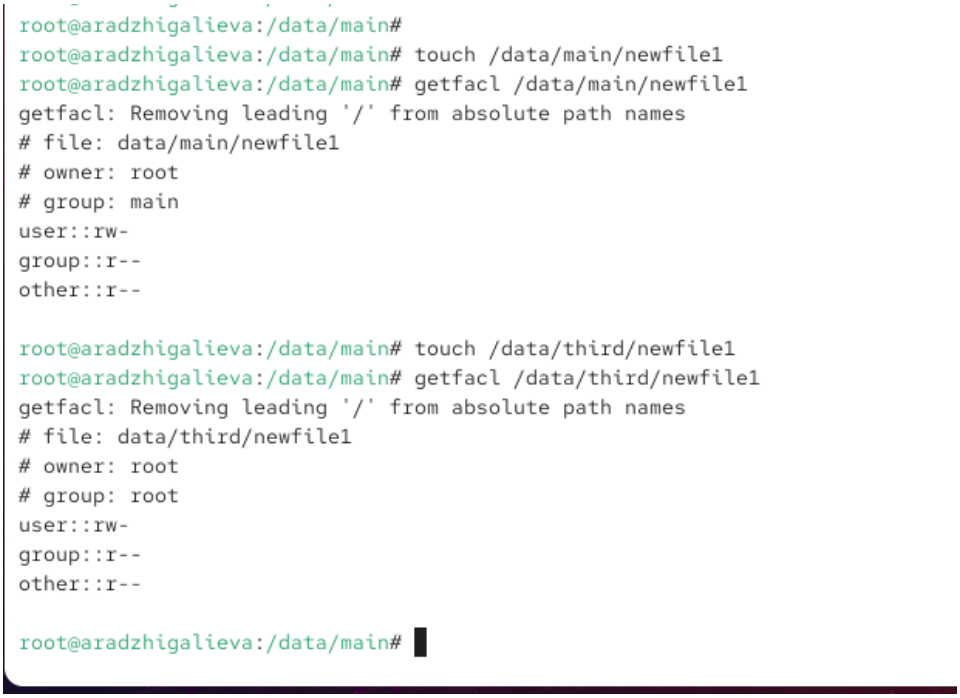

Рис. 2.4: Создание файлов newfile1 и просмотр их ACL

Далее я добавила ACL по умолчанию: для каталога `/data/main` — с правами `rwx` для группы **third**, а для каталога `/data/third` — с правами `rwx` для группы **main**. После этого были созданы новые файлы *newfile2*, которые уже унаследовали дополнительные разрешения.

```
root@aradzhigalieva:/data/main#
root@aradzhigalieva:/data/main# setfacl -m d:g:third:rwx /data/main/
root@aradzhigalieva:/data/main# setfacl -m d:g:main:rwx /data/third/
root@aradzhigalieva:/data/main# touch /data/main/newfile2
root@aradzhigalieva:/data/main# 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::---
```

Рис. 2.5: Добавление ACL по умолчанию и создание файлов newfile2

Проверка содержимого каталогов подтвердила, что новые файлы действительно унаследовали права, назначенные через ACL по умолчанию.

```
root@aradzhigalieva:/data/main# 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::---

root@aradzhigalieva:/data/main#
```

Рис. 2.6: Проверка наследования ACL для newfile2

Наконец, я вошла под пользователем **carol**, который входит в группу **third**, и проверила его возможности.
Удалить файлы *newfile1* и *newfile2* он не смог, так как не является их владельцем. Попытка записи в эти файлы также завершилась ошибкой «Permission denied», что подтверждает ограничение доступа на изменение содержимого.

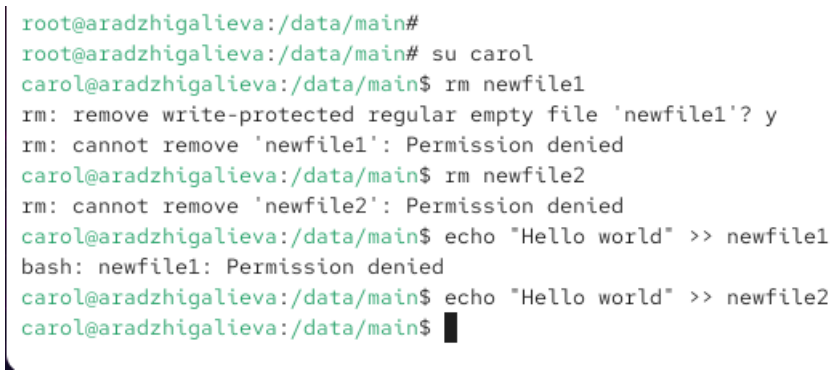

Рис. 2.7: Проверка доступа пользователя carol

3 Контрольные вопросы

1. **Как следует использовать команду chown, чтобы установить владельца группы для файла? Приведите пример.**
   Используется команда `chown`. Например: `chown :main file.txt` — установит группу владельцем файла `file.txt`.

2. **С помощью какой команды можно найти все файлы, принадлежащие конкретному пользователю? Приведите пример.**
   Для этого применяется команда `find`. Например: `find /home -user alice` — покажет все файлы, принадлежащие пользователю *alice*.

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.**
   Используется команда `chmod`. Например: `chmod -R 770 /data` — даст владельцу и группе права `rwx`, а для остальных пользователей права будут отсутствовать.

4. **Какая команда позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым?**
   Применяется команда `chmod +x file.sh`.

5. **Какая команда позволяет убедиться, что групповые разрешения для всех новых файлов, создаваемых в каталоге, будут присвоены владельцу группы этого каталога? Приведите пример.**
   Для этого используется установка SGID на каталог: `chmod g+s /data/main`.

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.**

   Для этого применяется sticky-bit: `chmod +t /data/main`.

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

   Для этого используется команда `setfacl -m g:main:r *`.

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.**

   Нужно установить права ACL по умолчанию: `setfacl -Rm d:g:main:r .`.

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

   Значение `umask 007` гарантирует, что у «других» пользователей не будет прав.

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

    Можно использовать команду `chattr +i myfile`, которая запрещает удаление и изменение файла.

4 Заключение

В ходе выполнения работы были рассмотрены базовые и специальные механизмы управления правами доступа в Linux.
Сначала были настроены стандартные разрешения с помощью команд `chmod` и `chgrp`, что позволило разграничить доступ пользователей к каталогам. Затем с использованием специальных атрибутов — SGID и sticky-bit — была обеспечена возможность совместной работы в общих директориях при сохранении защиты файлов от удаления посторонними пользователями.

Далее было изучено управление расширенными правами через списки ACL. Этот механизм позволил гибко задавать дополнительные правила доступа для отдельных групп и пользователей, а также настраивать наследование прав для новых файлов и каталогов.

В результате работы закреплены навыки администрирования доступа к ресурсам системы:
- использование стандартных и специальных битов прав;
- применение списков ACL для более тонкой настройки;
- организация совместной работы пользователей в общем каталоге с сохранением безопасности данных.