Отчёт по лабораторной работе №3

Настройка прав доступа

Анастасия Мазуркевич

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

2.1 Управление базовыми разрешениями

После входа в систему администратор перешёл в режим **root** командой `su`.

Далее были созданы каталоги **/data/main** и **/data/third**.

По умолчанию владельцем директорий являлся **root**.

Затем группы-владельцы каталогов были изменены на **main** и **third** с помощью `chgrp`.

После проверки команда показала, что каталоги принадлежат соответствующим группам.

Далее были назначены права доступа `770`, что означает полный доступ для владельца и группы, и отсутствие прав для остальных пользователей.

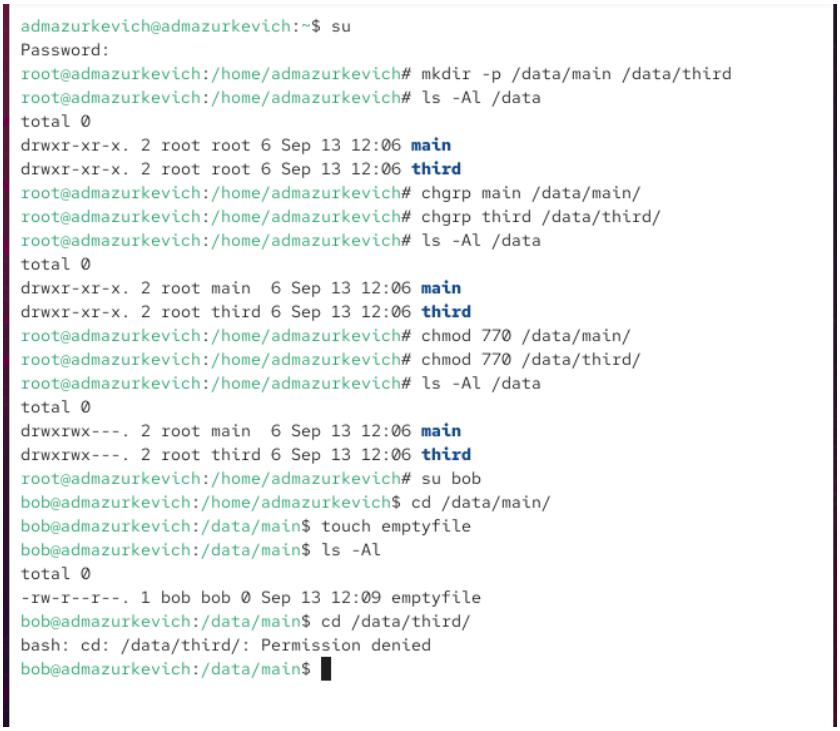

Рис. 2.1: Создание каталогов и настройка прав

После этого произведена проверка под пользователем **bob**.
При переходе в каталог **/data/main** удалось создать файл *emptyfile*, и его владельцем стал **bob**, так как у группы есть права на запись.

Однако при попытке зайти в каталог **/data/third** система выдала сообщение *Permission denied*, так как **bob** не состоит в группе *third* и не имеет доступа.

2.2 Управление специальными разрешениями

Для организации общего каталога с безопасным доступом между пользователями группы был использован **бит идентификатора группы (setgid)** и **sticky-бит**.

Вначале под пользователем **alice** был выполнен переход в каталог **/data/main**, где созданы два файла — *alice1* и *alice2*.
Оба файла принадлежали пользователю alice и его основной группе.

Затем под пользователем **bob** был произведён просмотр содержимого каталога.
В нём отобразились файлы, созданные alice.

Попытка удаления этих файлов завершилась успешно, так как каталог ещё не был защищён специальными атрибутами.

Далее bob создал собственные файлы *bob1* и *bob2*, владельцем которых стал сам bob.

Затем под пользователем **root** для каталога **/data/main** был установлен **бит идентификатора группы** и **sticky-бит**: `chmod g+s,o+t /data/main`

Эта комбинация означает: - **g+s (setgid)** — все новые файлы в каталоге будут принадлежать группе каталога, то есть **main**;

- **o+t (sticky-bit)** — удалять файлы могут только их владельцы или root, даже если у других пользователей есть права записи в каталог.

После этого alice снова создала два файла — *alice3* и *alice4*.

Команда `ls -l` показала, что владельцем файлов остаётся alice, но группа теперь принудительно устанавливается **main**, как у каталога.

Попытка alice удалить файлы, принадлежащие bob (*bob1* и *bob2*), завершилась ошибкой: *Operation not permitted*.

Это подтвердило корректную работу **sticky-бита**: alice не может удалить чужие файлы, хотя она имеет доступ к каталогу.

```
bob@admazurkevich:/data/main$
bob@admazurkevich:/data/main$ su alice
Password:
alice@admazurkevich:/data/main$ touch alice1
alice@admazurkevich:/data/main$ touch alice2
alice@admazurkevich:/data/main$
exit
bob@admazurkevich:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice alice 0 Sep 13 12:11 alice1
-rw-r--r--. 1 alice alice 0 Sep 13 12:11 alice2
-rw-r--r--. 1 bob   bob   0 Sep 13 12:09 emptyfile
bob@admazurkevich:/data/main$ rm -f alice*
bob@admazurkevich:/data/main$ ls -l
total 0
-rw-r--r--. 1 bob bob 0 Sep 13 12:09 emptyfile
bob@admazurkevich:/data/main$ touch bob1
bob@admazurkevich:/data/main$ touch bob2
bob@admazurkevich:/data/main$ su
Password:
root@admazurkevich:/data/main# chmod g+s,o+t /data/main/
root@admazurkevich:/data/main# su alice
alice@admazurkevich:/data/main$ touch alice3
alice@admazurkevich:/data/main$ touch alice4
alice@admazurkevich:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice main 0 Sep 13 12:12 alice3
-rw-r--r--. 1 alice main 0 Sep 13 12:12 alice4
-rw-r--r--. 1 bob   bob  0 Sep 13 12:12 bob1
-rw-r--r--. 1 bob   bob  0 Sep 13 12:12 bob2
-rw-r--r--. 1 bob   bob  0 Sep 13 12:09 emptyfile
alice@admazurkevich:/data/main$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
alice@admazurkevich:/data/main$
```

Рис. 2.2: Демонстрация работы setgid и sticky-бита

2.3 Управление расширенными разрешениями с использованием списков ACL

Под пользователем **root** были заданы права с помощью команд:

- для группы **third** на каталог `/data/main`: `g:third:rx`

- для группы **main** на каталог `/data/third`: `g:main:rx`

Результаты проверки командой `getfacl` показали, что дополнительные группы действительно получили права на чтение и выполнение.

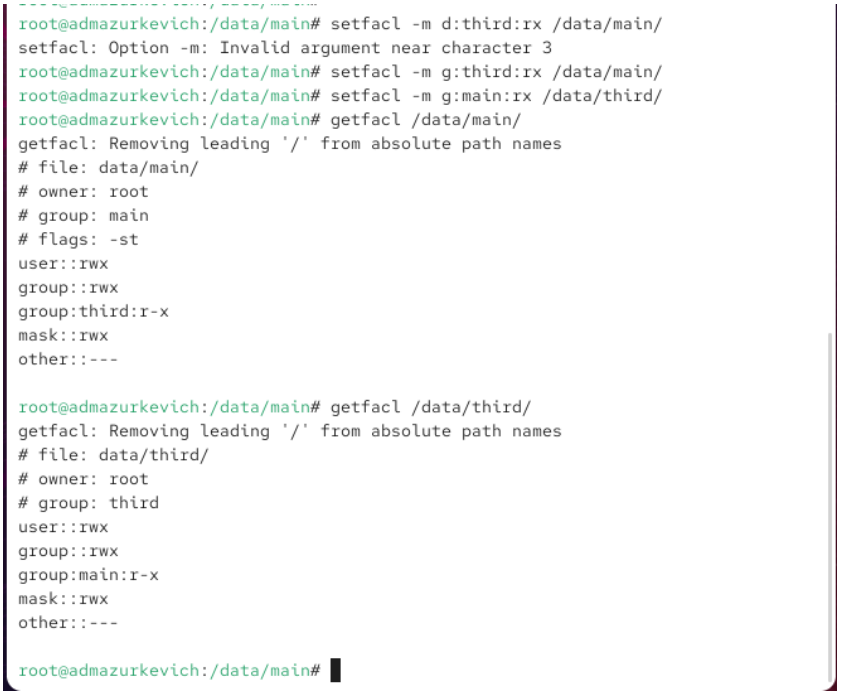

Рис. 2.3: Права ACL для каталогов

В каталоге **/data/main** был создан файл *newfile1*.
Команда `getfacl` показала, что он принадлежит пользователю **root** и группе **main**, но не унаследовал дополнительных прав для группы **third**.

Аналогично, в каталоге **/data/third** был создан файл *newfile1*, который унаследовал группу **root**, без наследования ACL для группы **main**.

Это объясняется тем, что **по умолчанию новые файлы наследуют только стандартные UNIX-права, а не настройки ACL каталога**.

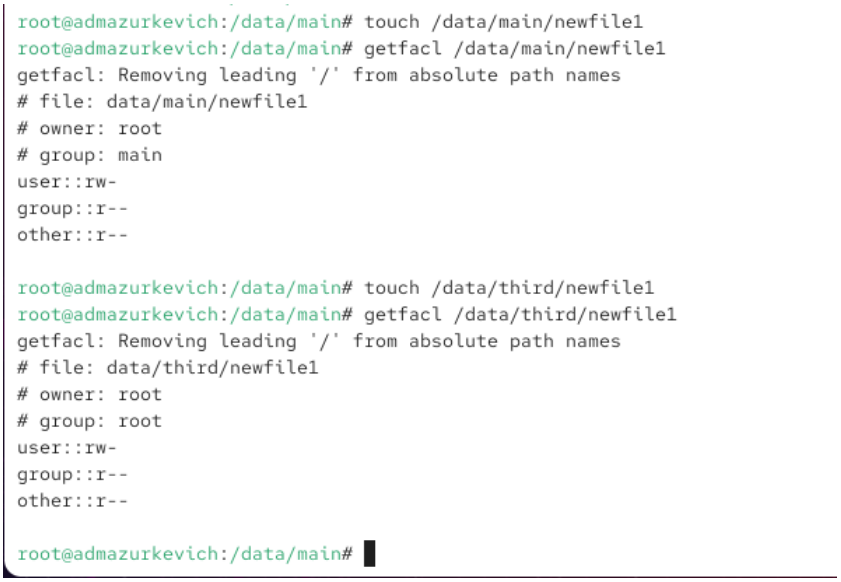

Рис. 2.4: Создание файлов без ACL по умолчанию

Чтобы исправить ситуацию, были заданы права наследования:

- для каталога `/data/main`: `d:g:third:rwx`
- для каталога `/data/third`: `d:g:main:rwx`

После этого в обоих каталогах были созданы новые файлы (*newfile2*). Проверка `getfacl` показала, что они унаследовали права для дополнительных групп (**third** и **main** соответственно).

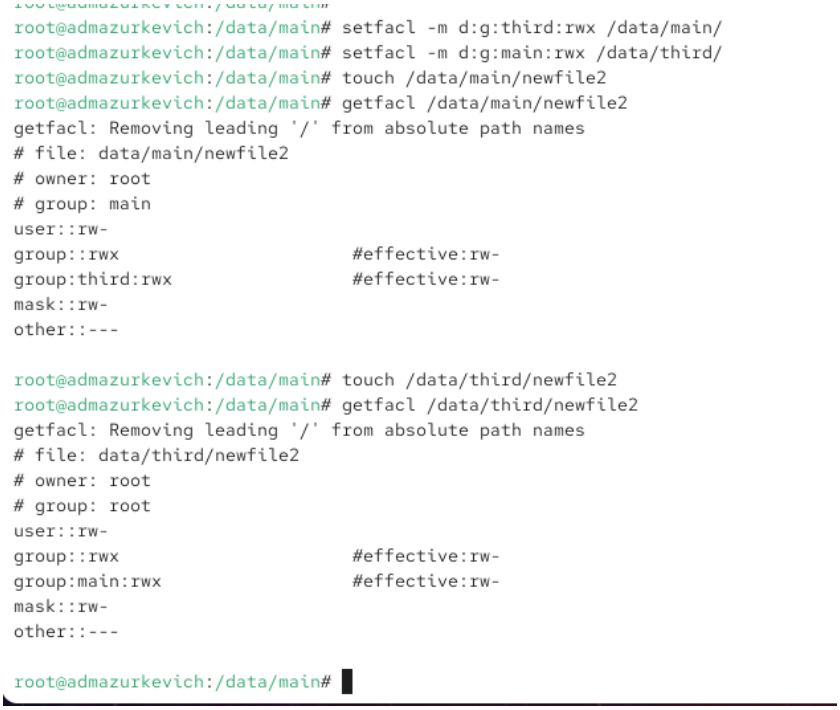

Рис. 2.5: Наследование ACL по умолчанию

Затем тест был выполнен под пользователем **carol**, который входит в группу **third**.

Carol попыталась удалить файлы *newfile1* и *newfile2* в каталоге **/data/main** — система выдала сообщение *Permission denied*. Sticky-бит и ограничения владельца защитили файлы.

Далее были предприняты попытки записи в файлы:

- при добавлении текста в *newfile1* и *newfile2* права группы **third** позволили выполнить операцию только для тех файлов, где ACL это разрешал.

В итоге **carol** смогла дописать данные в *newfile2* (так как у него сработали унаследованные ACL), но не смогла изменить *newfile1*, созданный до настройки ACL по умолчанию.

```
root@admazurkevich:/data/main# su carol
carol@admazurkevich:/data/main$ rm newfile1
rm: remove write-protected regular empty file 'newfile1'? y
rm: cannot remove 'newfile1': Permission denied
carol@admazurkevich:/data/main$ rm newfile2
rm: cannot remove 'newfile2': Permission denied
carol@admazurkevich:/data/main$ echo "Hello world !" >> newfile1
bash: newfile1: Permission denied
carol@admazurkevich:/data/main$ echo "Hello world !" >> newfile2
carol@admazurkevich:/data/main$ ls -l
total 4
-rw-r--r--. 1 alice main  0 Sep 13 12:12 alice3
-rw-r--r--. 1 alice main  0 Sep 13 12:12 alice4
-rw-r--r--. 1 bob   bob   0 Sep 13 12:12 bob1
-rw-r--r--. 1 bob   bob   0 Sep 13 12:12 bob2
-rw-r--r--. 1 bob   bob   0 Sep 13 12:09 emptyfile
-rw-r--r--. 1 root  main  0 Sep 13 12:16 newfile1
-rw-rw----+ 1 root  main 14 Sep 13 12:22 newfile2
carol@admazurkevich:/data/main$
```

Рис. 2.6: Проверка доступа пользователем carol

3 Контрольные вопросы

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

Команда `chown` позволяет задать владельца и группу. Например: `chown alice:main file1` изменяет владельца файла на alice и группу на main.

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

Для поиска используется команда `find` с параметром `-user`. Пример: `find / -user bob` выведет все файлы, владельцем которых является bob.

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.

Необходимо использовать команду `chmod` с правами 770. Пример: `chmod 770 /data/*`.

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

Для этого применяется команда `chmod +x`. Пример: `chmod +x script.sh`.

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

Используется установка бита setgid. Пример: `chmod g+s /data/main`.

6. Необходимо, чтобы пользователи могли удалять только те файлы, владельцами которых они являются, или которые находятся в каталоге, вла-

дельцами которого они являются. С помощью какой команды можно это сделать? Приведите пример.

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

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

Используется команда `setfacl`. Пример: `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.

Следует использовать рекурсивную установку ACL и ACL по умолчанию. Пример: `setfacl -R -m g:main:r .` и `setfacl -R -d -m g:main:r .`.

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

Нужно установить значение umask 007. Пример: `umask 007`.

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

Для этого используется атрибут immutable. Пример: `chattr +i myfile`.

4 Заключение

В ходе выполненной работы были рассмотрены базовые и расширенные механизмы управления правами доступа в Linux.
Были изучены стандартные права пользователей и групп, использование команд `chmod`, `chown`, `chgrp` для изменения владельцев и назначения разрешений, а также специальные атрибуты — **setgid** и **sticky-бит**, позволяющие реализовать безопасное совместное использование каталогов. Отдельное внимание уделено управлению доступом с помощью **ACL (Access Control Lists)**. Этот инструмент позволяет гибко задавать права для отдельных пользователей и групп, а также наследование прав для вновь создаваемых файлов и каталогов.