

# **Отчёт по лабораторной работе №3**

## **Настройка прав доступа**

Сулейм Гамбердов

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Ход выполнения</b>	<b>6</b>
2.1 Управление базовыми разрешениями . . . . .	6
2.2 Управление специальными разрешениями . . . . .	8
2.3 Управление расширенными разрешениями с использованием ACL	10
<b>3 Контрольные вопросы</b>	<b>15</b>
<b>4 Заключение</b>	<b>18</b>

# **Список иллюстраций**

2.1	Создание каталогов и проверка владельцев . . . . .	7
2.2	Создание и попытка удаления файлов с sticky bit . . . . .	10
2.3	Назначение ACL и проверка . . . . .	11
2.4	Создание файлов и проверка ACL . . . . .	12
2.5	ACL по умолчанию и проверка наследования . . . . .	13
2.6	Проверка полномочий пользователя carol . . . . .	14

# **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

1. Открыт терминал и выполнен переход под суперпользователя командой `su`.

После ввода пароля активирована учётная запись **root**, что подтверждается приглашением командной строки.

2. В корневом каталоге созданы подкаталоги `/data/main` и `/data/third`.

Проверка содержимого каталога `data` с помощью `ls -Al /data` показала, что владельцем каталогов является **root**.

```
alice@sigamberdov:/home/carol$ su
Password:
root@sigamberdov:/home/carol#
root@sigamberdov:/home/carol# mkdir -p /data/main /data/third
root@sigamberdov:/home/carol# ls -Al /data
total 0
drwxr-xr-x. 2 root root 6 Sep 17 13:53 main
drwxr-xr-x. 2 root root 6 Sep 17 13:53 third
root@sigamberdov:/home/carol# chgrp main /data/main
root@sigamberdov:/home/carol# chgrp third /data/third/
root@sigamberdov:/home/carol# ls -Al /data
total 0
drwxr-xr-x. 2 root main 6 Sep 17 13:53 main
drwxr-xr-x. 2 root third 6 Sep 17 13:53 third
root@sigamberdov:/home/carol# chmod 770 /data/main/
root@sigamberdov:/home/carol# chmod 770 /data/third/
root@sigamberdov:/home/carol# su bob
bob@sigamberdov:/home/carol$ cd /data/main/
bob@sigamberdov:/data/main$ touch emptyfile
bob@sigamberdov:/data/main$ ls -Al
total 0
-rw-r--r--. 1 bob bob 0 Sep 17 13:57 emptyfile
bob@sigamberdov:/data/main$ cd /data/third/
bash: cd: /data/third/: Permission denied
bob@sigamberdov:/data/main$
```

Рис. 2.1: Создание каталогов и проверка владельцев

3. Выполнено изменение групп-владельцев каталогов: **/data/main** передан группе **main**, а **/data/third** – группе **third**.

Повторная проверка вывода *ls -Al /data* подтвердила изменения: теперь каталог **/data/main** принадлежит группе **main**, а каталог **/data/third** – группе **third**.

4. Для каталогов установлены права доступа **770**.

Это означает:

- владелец и группа имеют полный доступ (*rwx*),
- остальные пользователи не имеют прав доступа (-).

5. В другом терминале выполнено переключение на пользователя **bob**.
6. Пользователь **bob** смог перейти в каталог **/data/main** и создать в нём файл

*emptyfile.*

В выводе `ls -Al` видно, что созданный файл принадлежит пользователю **bob**. Это стало возможным, так как группа каталога **main** имеет права на запись, и пользователь **bob** состоит в этой группе.

7. При попытке перейти в каталог **/data/third** и создать файл, система вернула сообщение *Permission denied*.

Доступ запрещён, так как пользователь **bob** не входит в группу **third**, а для остальных пользователей доступ полностью закрыт.

## 2.2 Управление специальными разрешениями

1. Выполнен вход под пользователем **alice** и выполнен переход в каталог **/data/main**.

В этом каталоге созданы два файла — *alice1* и *alice2*, владельцем которых является пользователь **alice**.

2. В другом терминале произведено переключение на пользователя **bob**, который также входит в группу **main**.

После перехода в каталог **/data/main** команда `ls -l` показала наличие файлов, созданных пользователем **alice**.

При попытке удалить эти файлы с помощью `*rm -f alice**` операция прошла успешно — файлы были удалены, хотя их владельцем был другой пользователь. Это возможно, так как в каталоге ещё не был установлен *sticky bit*, а права группы позволяли удаление.

3. Под пользователем **bob** в каталоге **/data/main** созданы два файла — *bob1* и *bob2*.

Они принадлежат пользователю **bob** и группе **bob**.

4. Для каталога **/data/main** под пользователем **root** установлен бит идентификатора группы (*setgid*) и *sticky bit*.

Команда `chmod g+s,o+t /data/main` обеспечила:

- автоматическое наследование групповой принадлежности для новых файлов;
  - запрет на удаление файлов другими пользователями, даже если у них есть права на запись в каталог.
5. После этого пользователь **alice** снова создал два файла – *alice3* и *alice4*. Проверка содержимого каталога показала, что новые файлы принадлежат группе **main**, то есть унаследовали групповую принадлежность каталога.
6. Попытка удалить файлы, созданные пользователем **bob** (`*rm -rf bob**`), завершилась неудачей.
- Система вернула сообщение *Operation not permitted*, так как был установлен *sticky bit*. Это подтверждает, что теперь удалить файлы в общем каталоге может только их владелец или администратор.

```
bob@sigamberdov:/data/main$ su alice
Password:
alice@sigamberdov:/data/main$ touch alice1
alice@sigamberdov:/data/main$ touch alice2
alice@sigamberdov:/data/main$
exit
bob@sigamberdov:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice alice 0 Sep 17 14:01 alice1
-rw-r--r--. 1 alice alice 0 Sep 17 14:01 alice2
-rw-r--r--. 1 bob   bob   0 Sep 17 13:57 emptyfile
bob@sigamberdov:/data/main$ rm -f alice*
bob@sigamberdov:/data/main$ ls -l
total 0
-rw-r--r--. 1 bob   bob   0 Sep 17 13:57 emptyfile
bob@sigamberdov:/data/main$ touch bob1
bob@sigamberdov:/data/main$ touch bob2
bob@sigamberdov:/data/main$ su
Password:
root@sigamberdov:/data/main# chmod g+s,o+t /data/main/
root@sigamberdov:/data/main# su alice
alice@sigamberdov:/data/main$ touch alice3
alice@sigamberdov:/data/main$ touch alice4
alice@sigamberdov:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice main 0 Sep 17 14:02 alice3
-rw-r--r--. 1 alice main 0 Sep 17 14:02 alice4
-rw-r--r--. 1 bob   bob   0 Sep 17 14:01 bob1
-rw-r--r--. 1 bob   bob   0 Sep 17 14:01 bob2
-rw-r--r--. 1 bob   bob   0 Sep 17 13:57 emptyfile
alice@sigamberdov:/data/main$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
alice@sigamberdov:/data/main$
```

Рис. 2.2: Создание и попытка удаления файлов с sticky bit

## 2.3 Управление расширенными разрешениями с использованием ACL

1. Выполнен вход под пользователем **root**.
2. Для группы **third** установлены права чтения и выполнения в каталоге **/data/main**, а для группы **main** — права чтения и выполнения в каталоге **/data/third**.

Проверка через *getfacl* подтвердила назначение:

- каталог **/data/main** теперь доступен группе **third** с правами **r-x**;
- каталог **/data/third** доступен группе **main** с правами **r-x**.

```
alice@sigamberdov:/data/main$ su
Password:
root@sigamberdov:/data/main#
root@sigamberdov:/data/main# setfacl -m g:third:rx /data/main
root@sigamberdov:/data/main# setfacl -m g:main:rx /data/third/
root@sigamberdov:/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@sigamberdov:/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@sigamberdov:/data/main#
```

Рис. 2.3: Назначение ACL и проверка

### 3. Создан новый файл *newfile1* в каталоге **/data/main**.

При проверке через *getfacl* видно, что файл имеет стандартные права: владелец **root** — **rwx**, группа — **r--**, остальные — **r--**.

Права группы **third** для этого файла отсутствуют, так как ACL были назначены только каталогу, но не распространяются на уже созданные файлы.

Аналогично, в каталоге **/data/third** создан файл *newfile1*. Он также получил стандартные права, без учёта ACL для группы **main**.

```
root@sigamberdov:/data/main#  
root@sigamberdov:/data/main# touch /data/main/newfile1  
root@sigamberdov:/data/main# 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@sigamberdov:/data/main# touch /data/third/newfile1  
root@sigamberdov:/data/main# 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@sigamberdov:/data/main#
```

Рис. 2.4: Создание файлов и проверка ACL

4. Для корректного наследования прав были установлены ACL по умолчанию:

- для каталога **/data/main** – `d:g:third:rwx`;
- для каталога **/data/third** – `d:g:main:rwx`.

После этого созданы новые файлы *newfile2* в каждом каталоге. Проверка показала, что они унаследовали права ACL по умолчанию:

- в каталоге **/data/main** файлы имеют права для группы **third**;
- в каталоге **/data/third** – для группы **main**.

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

root@sigamberdov:/data/main# touch /data/third/newfile2
root@sigamberdov:/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@sigamberdov:/data/main#
```

Рис. 2.5: ACL по умолчанию и проверка наследования

5. Для проверки полномочий был выполнен вход под пользователем **carol**, который входит в группу **third**.

Попытка удалить файлы *newfile1* и *newfile2* в каталоге **/data/main** завершилась ошибкой *Permission denied*.

Аналогично, при попытке записать данные в эти файлы также возник отказ.

Это объясняется тем, что группа **third** получила права на чтение и выполнение, а также наследование прав на новые файлы, но не имеет полномочий на удаление или запись в файлы, владельцем которых является другой пользователь.

```
root@sigamberdov:/data/main#
root@sigamberdov:/data/main# su carol
carol@sigamberdov:/data/main$ rm /data/main/newfile1
rm: remove write-protected regular empty file '/data/main/newfile1'? y
rm: cannot remove '/data/main/newfile1': Permission denied
carol@sigamberdov:/data/main$ rm /data/main/newfile2
rm: cannot remove '/data/main/newfile2': Permission denied
carol@sigamberdov:/data/main$ echo "Hello world" >> /data/main/newfile1
bash: /data/main/newfile1: Permission denied
carol@sigamberdov:/data/main$ echo "Hello world" >> /data/main/newfile2
carol@sigamberdov:/data/main$
```

Рис. 2.6: Проверка полномочий пользователя carol

## 3 Контрольные вопросы

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

- Для изменения владельца группы используется команда **chown** в формате:

```
chown :имя_группы файл
```

- Пример:

```
chown :developers project.txt
```

Файл *project.txt* будет принадлежать группе **developers**.

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

- Используется команда **find**:

```
find / -user alice
```

Найдёт все файлы, владельцем которых является пользователь **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**:

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

Пользователь и группа получают права *rwx*, а остальные не имеют доступа.

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

- Применяется команда:

```
chmod +x script.sh
```

Теперь файл *script.sh* станет исполняемым.

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 **setgid** на каталог:

```
chmod g+s /data/projects
```

Все новые файлы в каталоге **/data/projects** будут принадлежать его группе.

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 используется **sticky bit**:

```
chmod +t /shared
```

В каталоге **/shared** пользователи смогут удалять только свои файлы.

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

- Используется команда:

```
setfacl -m g:developers:r *
```

Все файлы в текущем каталоге становятся доступными группе **developers** для чтения.

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?

логах, а также для всех файлов, которые будут созданы в этом каталоге в будущем? Приведите пример.

- Нужно применить **ACL по умолчанию**:

```
setfacl -R -m g:developers:rx /data
```

```
setfacl -d -m g:developers:rx /data
```

Теперь группа **developers** получит права чтения и выполнения для всех файлов и каталогов, включая новые.

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

- Для этого значение **umask** должно быть **0077**.

- Пример:

```
umask 077
```

Новые файлы будут доступны только владельцу и его группе.

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

- Для защиты файла от удаления можно убрать права на запись каталога или назначить атрибут **immutable**:

```
chattr +i myfile
```

Теперь файл *myfile* нельзя будет удалить или изменить, пока атрибут не будет снят.

## **4 Заключение**

В ходе работы были рассмотрены базовые и специальные разрешения, а также расширенные механизмы управления доступом с помощью ACL.

Это позволило закрепить навыки администрирования прав в Linux и понять, как обеспечивать безопасный совместный доступ пользователей к общим ресурсам.