Отчёт по лабораторной работе №3

Настройка прав доступа

Максат Хемраев

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 Отчёт по выполнению работы

2.1 Управление базовыми разрешениями

1. Вошёл в систему под пользователем **root**. Затем создал структуру каталогов */data/main* и */data/third*.

   Проверка показала, что владельцем по умолчанию стал суперпользователь **root**, а владельцем группы — также **root**.

```
mhemraev@mhemraev:~$ su
Password:
root@mhemraev:/home/mhemraev# mkdir -p /data/main /data/third
root@mhemraev:/home/mhemraev# ls -Al /data
total 0
drwxr-xr-x. 2 root root 6 Sep 12 10:58 main
drwxr-xr-x. 2 root root 6 Sep 12 10:58 third
root@mhemraev:/home/mhemraev# chgrp main /data/main/
root@mhemraev:/home/mhemraev# chgrp third /data/third/
root@mhemraev:/home/mhemraev# ls -Al /data/
total 0
drwxr-xr-x. 2 root main  6 Sep 12 10:58 main
drwxr-xr-x. 2 root third 6 Sep 12 10:58 third
root@mhemraev:/home/mhemraev# chmod 770 /data/main/
root@mhemraev:/home/mhemraev# chmod 770 /data/third/
root@mhemraev:/home/mhemraev# ls -Al /data/
total 0
drwxrwx---. 2 root main  6 Sep 12 10:58 main
drwxrwx---. 2 root third 6 Sep 12 10:58 third
root@mhemraev:/home/mhemraev#
root@mhemraev:/home/mhemraev#
```

Рис. 2.1: Создание каталогов main и third

2. Изменил владельцев групп каталогов: для */data/main* установил группу **main**, а для */data/third* — группу **third**.

После проверки стало видно, что каталоги теперь принадлежат соответствующим группам.

3. Задал права доступа к каталогам так, чтобы только владельцы и их группы имели возможность чтения, записи и выполнения. Всем остальным пользователям доступ был запрещён. Для этого применил команды *chmod 770 /data/main* и *chmod 770 /data/third*.
   Повторная проверка подтвердила корректное применение прав: теперь доступ имеют только владельцы и группы.

4. Вошёл в систему под пользователем **bob**. Перешёл в каталог */data/main* и создал там пустой файл **emptyfile**.
   Проверка содержимого показала, что файл успешно создан, его владельцем стал пользователь **bob**, а права доступа ограничены для других.

```
root@mnemraev:/nome/mnemraev#
root@mhemraev:/home/mhemraev# su bob
bob@mhemraev:/home/mhemraev$ cd /data/main/
bob@mhemraev:/data/main$ touch emptyfile
bob@mhemraev:/data/main$ ls -Al
total 0
-rw-r--r--. 1 bob bob 0 Sep 12 11:02 emptyfile
bob@mhemraev:/data/main$ cd /data/third/
bash: cd: /data/third/: Permission denied
bob@mhemraev:/data/main$
```

Рис. 2.2: Создание файла emptyfile пользователем bob

5. После этого под пользователем **bob** была предпринята попытка перейти в каталог */data/third*. Доступ оказался запрещён, так как группа каталога установлена **third**, а пользователь **bob** в неё не входит. Соответственно, bob не имеет прав на выполнение операций в этом каталоге.

2.2 Управление специальными разрешениями

1. Вошёл в систему под пользователем **alice** и перешёл в каталог */data/main*. Создал два файла — **alice1** и **alice2**, владельцем которых является alice.

```
bob@mhemraev:/data/main$ su alice
Password:
alice@mhemraev:/data/main$ cd /data/main/
alice@mhemraev:/data/main$ touch alice1
alice@mhemraev:/data/main$ touch alice2
alice@mhemraev:/data/main$ su bob
Password:
bob@mhemraev:/data/main$ cd /data/main/
bob@mhemraev:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice alice 0 Sep 12 11:04 alice1
-rw-r--r--. 1 alice alice 0 Sep 12 11:04 alice2
-rw-r--r--. 1 bob   bob   0 Sep 12 11:02 emptyfile
bob@mhemraev:/data/main$ rm -f alice*
bob@mhemraev:/data/main$ ls -l
total 0
-rw-r--r--. 1 bob bob 0 Sep 12 11:02 emptyfile
bob@mhemraev:/data/main$
```

Рис. 2.3: Создание файлов alice1 и alice2

2. В другом терминале перешёл под учётную запись **bob**. В каталоге */data/main* выполнил просмотр содержимого и убедился в наличии файлов alice. Затем удалил их, что оказалось возможным из-за отсутствия дополнительных ограничений.

3. Под пользователем **bob** создал в каталоге два собственных файла — **bob1** и **bob2**.

4. Далее под пользователем **root** для каталога */data/main* установил бит идентификатора группы и sticky-бит с помощью команды *chmod g+s,o+t /data/main*. Это позволило назначить наследование группы **main** для всех создаваемых файлов, а также запретило удаление чужих файлов пользователями.

5. Вернувшись под пользователем **alice**, создал в каталоге ещё два файла — **alice3** и **alice4**. Проверка показала, что оба файла принадлежат группе **main**, которая является группой-владельцем каталога.

```
bob@mhemraev:/data/main$
bob@mhemraev:/data/main$ touch bob1
bob@mhemraev:/data/main$ touch bob2
bob@mhemraev:/data/main$ su
Password:
root@mhemraev:/data/main# chmod g+s,o+t /data/main/
root@mhemraev:/data/main# su alice
alice@mhemraev:/data/main$ touch alice3
alice@mhemraev:/data/main$ touch alice4
alice@mhemraev:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice main 0 Sep 12 11:06 alice3
-rw-r--r--. 1 alice main 0 Sep 12 11:06 alice4
-rw-r--r--. 1 bob   bob  0 Sep 12 11:06 bob1
-rw-r--r--. 1 bob   bob  0 Sep 12 11:06 bob2
-rw-r--r--. 1 bob   bob  0 Sep 12 11:02 emptyfile
alice@mhemraev:/data/main$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
alice@mhemraev:/data/main$
```

Рис. 2.4: Создание файлов alice3 и alice4 с установленными спецразрешениями

6. Под alice была предпринята попытка удалить файлы bob (**bob1** и **bob2**). Система отклонила запрос, выдав сообщение *Operation not permitted*. Sticky-бит сработал корректно, сохранив файлы bob от удаления другим пользователем.

2.3 Управление расширенными разрешениями с использованием списков ACL

1. Вошёл в систему под пользователем **root**. Для каталога */data/main* установил права на чтение и выполнение для группы **third**, а для каталога */data/third* — права на чтение и выполнение для группы **main**.

```
root@mhemraev:/data/main# setfacl -m g:third:rx /data/main/
root@mhemraev:/data/main# setfacl -m g:main:rx /data/third/
root@mhemraev:/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@mhemraev:/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@mhemraev:/data/main#
```

Рис. 2.5: Установка ACL для групп main и third

2. Проверил применённые разрешения с помощью команды *getfacl*. Каталоги отобразили новые записи для дополнительных групп.

3. Создал файл **newfile1** в каталоге */data/main*. Проверка с помощью *getfacl* показала, что права доступа у файла остались стандартными: владелец имеет доступ к чтению и записи, группа получила только права на чтение. Это объясняется тем, что новые ACL были назначены только на каталог, а не как наследуемые по умолчанию для новых файлов.
   Аналогичные действия выполнил в каталоге */data/third* и получил тот же результат.

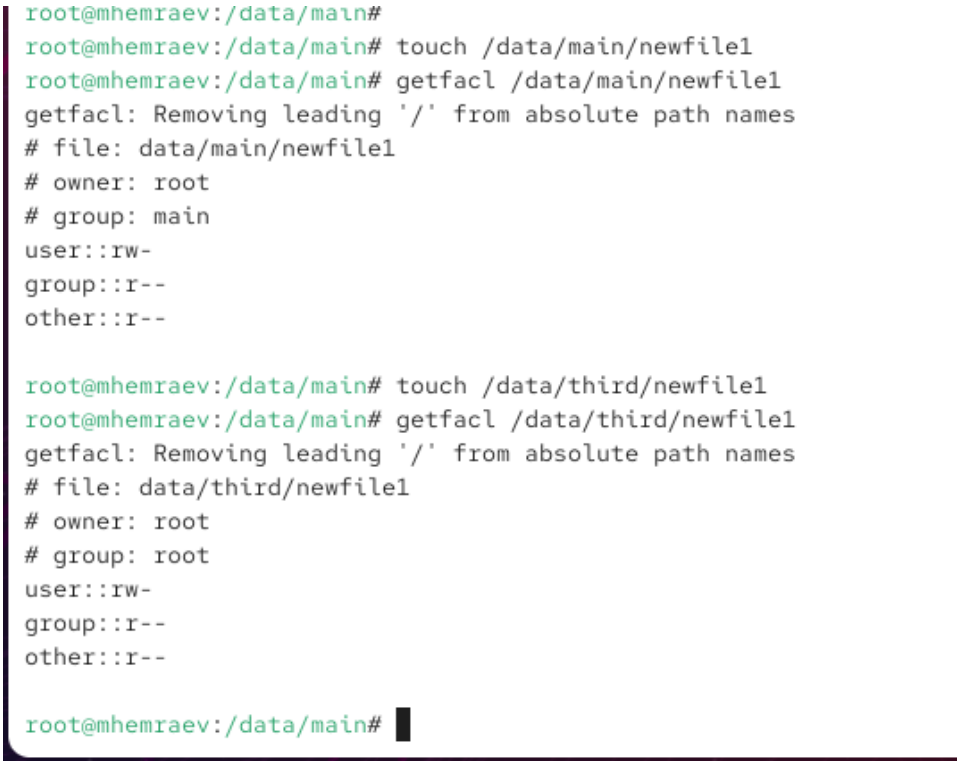

Рис. 2.6: Создание и проверка файла newfile1

4. Установил ACL по умолчанию: для группы **third** в каталоге */data/main* — *rwx*, а для группы **main** в каталоге */data/third* — *rwx*.

5. Создал новые файлы **newfile2** в каталогах */data/main* и */data/third*. Теперь при проверке с помощью *getfacl* было видно, что права по умолчанию успешно применились: группы **third** и **main** получили доступ к чтению и записи.

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

Рис. 2.7: Создание и проверка файла newfile2

6. Для проверки полномочий вошёл в систему под пользователем **carol**, который входит в группу **third**.

   Попытка удалить файлы **newfile1** и **newfile2** завершилась ошибкой *Permission denied*, что подтвердило ограничение на удаление.

   Также при попытке записи данных в файл **newfile1** система отказала, так как у него не были заданы наследуемые ACL. Однако в файл **newfile2** запись оказалась возможна, так как он был создан уже после установки ACL по умолчанию.

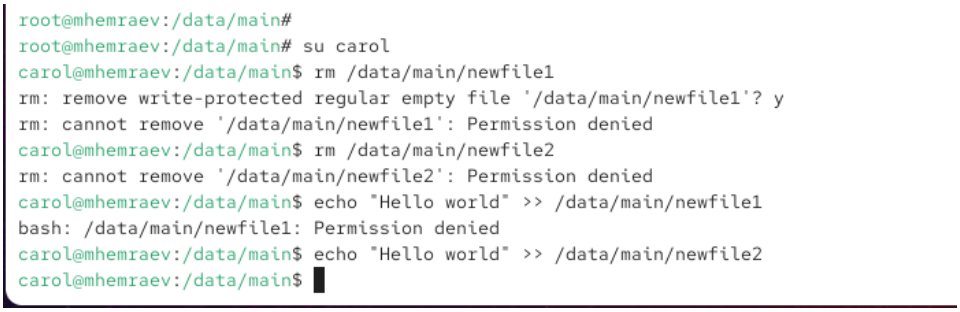

Рис. 2.8: Проверка работы ACL под пользователем carol

3 Контрольные вопросы

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

- Команда chown позволяет изменить владельца и/или группу файла.
- Синтаксис: chown :<имя_группы> <файл> — изменяет только группу.
- Пример:

```
chown :main /data/file1
```

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

- Для поиска используется команда find.
- Пример:

```
find /home -user alice
```

3.0.3 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`.
- Пример:

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

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

- Используется команда `chmod +x`.
- Пример:

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

3.0.5 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 применяется **SGID-бит**.
- Пример:

```
chmod g+s /data/main
```

3.0.6 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-бит**.
- Пример:

```
chmod +t /data/main
```

3.0.7 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 ./*
```

3.0.8 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 применить ACL с опцией наследования `-d`.

- Пример:

```
setfacl -R -m g:main:r /data
setfacl -R -d -m g:main:r /data
```

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

- Нужно установить `umask 007`.
- Пример:

```
umask 007
```

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

- Нужно убрать права на запись и установить атрибут защиты от удаления.
- Пример:

```
chmod a-w myfile
chattr +i myfile
```

4 Заключение

В ходе работы были изучены и применены базовые, специальные и расширенные разрешения в Linux. Создавались каталоги, назначались права доступа, использовались sticky-бит, SGID и ACL, что позволило реализовать разграничение доступа и обеспечить безопасное совместное использование ресурсов пользователями разных групп.