

Отчет о лабораторной работе

Лабораторная работа №3

Казначеев Сергей Ильич

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Контрольные вопросы	13
4 Выводы	16

Список иллюстраций

2.1 1	6
2.2 2	6
2.3 3	6
2.4 4	7
2.5 5	7
2.6 6	7
2.7 7	8
2.8 8	8
2.9 9	8
2.10 10	9
2.11 11	9
2.12 12	9
2.13 13	10
2.14 14	10
2.15 15	10
2.16 16	11
2.17 17	11
2.18 18	12
2.19 19	12

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Для начала откроем терминал и перейдем в учетную запись root в корневом каталоге создадим каталоги /data/main и /data/third (рис. 2.1).

```
[sikaznacheev@localhost ~]$ su -  
Пароль:  
[root@localhost ~]# mkdir -p /data/main /data/third
```

Рис. 2.1: 1

Затем меняем владельцев каталогов с root на main и third соответственно и проверяем это (рис. 2.2).

```
[root@localhost ~]# chgrp main /data/main  
[root@localhost ~]# chgrp third /data/third  
[root@localhost ~]# ls -Al /data  
итого 0  
drwxr-xr-x. 2 root main 6 сен 15 13:45 main  
drwxr-xr-x. 2 root third 6 сен 15 13:45 third
```

Рис. 2.2: 2

Далее устанавливаем разрешения позволяющие владельцам каталогов записывать файлы в эти каталоги и запрещающие доступ к содержимому каталогов всем другим пользователям и группам 2.3).

```
[root@localhost ~]# chmod 770 /data/main  
[root@localhost ~]# chmod 770 /data/third  
[root@localhost ~]# ls -Al /data  
итого 0  
drwxrwx---. 2 root main 6 сен 15 13:45 main  
drwxrwx---. 2 root third 6 сен 15 13:45 third
```

Рис. 2.3: 3

Затем переходим в учетную запись bob (рис. 2.4).

Переходим в каталог /data/main и создаем файл emptyfile в данном каталоге

```
[sikaznacheev@localhost ~]$ su - bob  
Пароль:  
[bob@localhost ~]$ cd /data/main  
[bob@localhost main]$ touch emptyfile  
[bob@localhost main]$ ls -Al  
итого 0  
-rw-r--r--. 1 bob bob 0 сен 15 13:48 emptyfile
```

Рис. 2.4: 4

Под пользователем bob пробуем перейти в каталог /data/third и создаем файл emptyfile в этом каталоге и у нас не получится это так как bob находится в main и принадлежит группе main (рис. 2.5).

```
[bob@localhost main]$ cd /data/third  
-bash: cd: /data/third: Отказано в доступе
```

Рис. 2.5: 5

Меняем пользователя на alice переходим в каталог и создаем два файла (рис. 2.6).

```
[sikaznacheev@localhost ~]$ su - alice  
Пароль:  
[alice@localhost ~]$ cd /data/main  
[alice@localhost main]$ touch alicel  
[alice@localhost main]$ touch alice2
```

Рис. 2.6: 6

После чего в новом терминале перейдем под учетную запись пользователя bob и пробуем удалить файлы принадлежащие пользователю alice (рис. 2.7).

```
[bob@localhost main]$ ls -l
итого 0
-rw-r--r--. 1 alice alice 0 сен 15 13:52 alice1
-rw-r--r--. 1 alice alice 0 сен 15 13:52 alice2
-rw-r--r--. 1 bob   bob   0 сен 15 13:48 emptyfile
[bob@localhost main]$ rm -f alice*
[bob@localhost main]$ ls -l
итого 0
-rw-r--r--. 1 bob   bob   0 сен 15 13:48 emptyfile
[bob@localhost main]$ █
```

Рис. 2.7: 7

Далее создаем два файла которые принадлежат bob(рис. 2.8).

```
[bob@localhost main]$ touch bob1
[bob@localhost main]$ touch bob2
[bob@localhost main]$ █
```

Рис. 2.8: 8

Потом переходим под пользователя root и устанавливаем для каталога /data/main бит иден-тификатора группы, а также sticky-бит для разделяемого (общего) каталога группы (рис. 2.9).

```
[root@localhost ~]# chmod g+s,o+t /data/main
[root@localhost ~]#
```

Рис. 2.9: 9

Далее в терминале под пользователем alice создаем в каталоге /data/main файлы alice3 и alice4 и проверяем это (рис. 2.10).

```
[alice@localhost main]$ touch alice3
[alice@localhost main]$ touch alice4
[alice@localhost main]$ ls -l
итого 0
-rw-r--r--. 1 alice main 0 сен 15 13:54 alice3
-rw-r--r--. 1 alice main 0 сен 15 13:54 alice4
-rw-r--r--. 1 bob   bob  0 сен 15 13:53 bob1
-rw-r--r--. 1 bob   bob  0 сен 15 13:53 bob2
-rw-r--r--. 1 bob   bob  0 сен 15 13:48 emptyfile
[alice@localhost main]$
```

Рис. 2.10: 10

Пробуем удалить файлы bob под пользователем alice и убеждаемся что sticky-bit предотвратит удаление этих файлов пользователем alice (рис. 2.11).

```
[alice@localhost main]$ rm -rf bob*
rm: невозможно удалить 'bob1': Операция не позволена
rm: невозможно удалить 'bob2': Операция не позволена
[alice@localhost main]$
```

Рис. 2.11: 11

Открываем новый терминал и переходим в пользователя root и устанавливаем права для чтения и выполнения в каталоге /data/main для группы third и права на чтение и выполнение для группы main в каталоге /data/third (рис. 2.12).

```
[sikaznacheev@localhost ~]$ su -
Пароль:
[root@localhost ~]# setfacl -m g:third:rx /data/main
[root@localhost ~]# setfacl -m g:main:rx /data/third
```

Рис. 2.12: 12

После чего используем команду getfacl, чтобы убедиться в правильности установки разрешений main и third (рис. 2.13).

```
[root@localhost ~]# 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::---
```

Рис. 2.13: 13

```
[root@localhost ~]# 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::---
```

Рис. 2.14: 14

После проверки создаем новый файл с именем newfile1и проверяем текущее назначение полномочий у нас оказывается что права доступа рис. 2.15).

1. Владелец -root- чтение и запись
2. Группа владелец - group main - только чтение
3. Все остальные - other - только чтение

Причина почему права именно такие в том что отсутствуют наследуемые acl

```
[root@localhost ~]# touch /data/main/newfile1
[root@localhost ~]# 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--
```

Рис. 2.15: 15

Далее устанавливаем ACL по умолчанию для каталога /data/main и /data/third, проверяем что настройки acl работают добавив новые файлы в каталог /data/main и проверяем текущее назначение полномочий для main(рис. 2.16).

```
[root@localhost ~]# setfacl -m d:g:third:rwx /data/main
[root@localhost ~]# setfacl -m d:g:main:rwx /data/third
[root@localhost ~]# touch /data/main/newfile2
[root@localhost ~]# getfacl /data/main/newfile2
> ^C
[root@localhost ~]# 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. 2.16: 16

Затем проверяем назначение полномочий для third (рис. 2.17).

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

Рис. 2.17: 17

После чего переходим под учетную запись группы third - carol и проверяем операции с файлами (рис. 2.18).

1. rm /data/main/newfile1
2. rm /data/main/newfile2

```
[sikaznacheev@localhost ~]$ su - carol  
Пароль:  
[carol@localhost ~]$ rm /data/main/newfile1  
rm: удалить защищенный от записи пустой обычный файл '/data/main/newfile1'? у  
рм: невозможно удалить '/data/main/newfile1': Отказано в доступе  
[carol@localhost ~]$ rm /data/main/newfile2  
rm: невозможно удалить '/data/main/newfile2': Отказано в доступе  
[carol@localhost ~]$ █
```

Рис. 2.18: 18

Затем проверяем возможно ли осуществить запись в файл

1. echo "Hello, world" » /data/main/newfile1
2. echo "Hello, world" » /data/main/newfile2

```
[carol@localhost ~]$ echo "Hello, world" >> /data/main/newfile1  
-bash: /data/main/newfile1: Отказано в доступе  
[carol@localhost ~]$ echo "Hello, world" >> /data/main/newfile2  
[carol@localhost ~]$ █
```

Рис. 2.19: 19

3 Контрольные вопросы

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

Ответ - чтобы установить владельца группы для файла нужно использовать команду chown user:group file Пример: chown sergey:developers report.txt

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

Ответ - с помощью команды find / -user user_name Пример: find /home -user ivan

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

Ответ - для того чтобы применить разрешения на чтение, запись и выполнение для всех файлов в каталоге /data для пользователей нужно выдать права чтение, запись и выполнение только владельцу группе а для других убрать Пример: chmod 770 /data/file1

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

Ответ - 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.

Ответ - команда chmod g+s каталог Пример: chmod g+s /projects

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.

Ответ - команда chmod +t каталог Пример: chmod +t /projects

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

Ответ - команда setfacl -m g:groupname: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.

Ответ - чтобы гарантировать что члены группы всегда будут иметь доступ на чтение к файлам в текущем каталоге его подкаталогах и ко всем будущим файлам нужно использовать acl

Пример: setfacl -R m g:groupname:rX - это команда для установки прав чтения
Пример: setfacl -d -m g:groupname:rX. - это для установки прав по умолчанию для будущих прав файлов и каталогов

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

Ответ - чтобы другие пользователи не получали никаких прав на новые файлы нужно выставить umask обнуляющий все разрешения для категорий others.

Пример: umask 007

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

Ответ команда chattr +i myfile гарантирует что никто не сможет удалить файл myfile случайно.

4 Выводы

В результате выполнения лабораторной работы я получил опыт работы с настройками базовых и специальных прав доступа для групп пользователей в операционной системе типа linux