

# **Отчёт по лабораторной работе 3**

## **Настройка прав доступа**

Вишняков Родион Сергеевич

# **Содержание**

<b>Цель работы</b>	<b>5</b>
<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>Вывод</b>	<b>16</b>
<b>Контрольные вопросы</b>	<b>17</b>

# Список иллюстраций

1	Справочное описание . . . . .	6
2	Справочное описание . . . . .	7
3	Справочное описание . . . . .	7
4	Справочное описание . . . . .	8
5	Учетная запись root . . . . .	8
6	Создание каталогов . . . . .	8
7	Замена владельцев . . . . .	9
8	Установка разрешений . . . . .	9
9	Учетная запись bob . . . . .	9
10	Попытка создать файл . . . . .	9
11	Попытка создать файл . . . . .	10
12	Учетная запись alice . . . . .	10
13	Создание файлов . . . . .	10
14	Учётная запись bob . . . . .	10
15	Попытка удалить файлы . . . . .	11
16	Создание файлов . . . . .	11
17	Установка битов . . . . .	11
18	Создание файлов . . . . .	12
19	Попытка удалить файлы . . . . .	12
20	Учётная запись root . . . . .	12
21	Установка прав . . . . .	12
22	Проверка правильности установки разрешений . . . . .	13
23	Создание файла и проверка . . . . .	14
24	Установка ACL . . . . .	14
25	Установка ACL . . . . .	14
26	Проверка ACL . . . . .	15
27	Проверки . . . . .	15

# **Список таблиц**

# **Цель работы**

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

# Выполнение лабораторной работы

Читаю справочное описание man по командам: chgrp, chmod, getfacl, setfacl

```
rodvish@rodvishh:~ - man chgrp
+
CHGRP(1)                               User Commands                               CHGRP(1)
NAME
    chgrp - change group ownership

SYNOPSIS
    chgrp [OPTION]... GROUP FILE...
    chgrp [OPTION]... --reference=RFILE FILE...

DESCRIPTION
    Change the group of each FILE to GROUP. With --reference, change the
    group of each FILE to that of RFILE.

    -c, --changes
        like verbose but report only when a change is made

    -f, --silent, --quiet
        suppress most error messages

    -v, --verbose
        output a diagnostic for every file processed

    --dereference
        affect the referent of each symbolic link (this is the default),
        rather than the symbolic link itself

    -h, --no-dereference
        affect symbolic links instead of any referenced file (useful
        only on systems that can change the ownership of a symlink)

    --from=CURRENT_OWNER:CURRENT_GROUP
        change the ownership of each file only if its current owner
        and/or group match those specified here. Either may be omitted,
        in which case a match is not required for the omitted attribute

    --no-preserve-root
        do not treat '/' specially (the default)

    --preserve-root
        fail to operate recursively on '/'


Manual page chgrp(1) line 1 (press h for help or q to quit)
```

Рис. 1: Справочное описание

rodvish@rodvish:~ - man chmod

---

**CHMOD(1)**

User Commands

**NAME**  
chmod - change file mode mode bits

**SYNOPSIS**  
**chmod** [**OPTION**]... **MODE**[,**FILE**]...  
**chmod** [**OPTION**]... **OCTAL-MODE** **FILE**...  
**chmod** [**OPTION**]... **-reference=RFILE** **FILE**...

**DESCRIPTION**  
This manual page documents the GNU version of **chmod**. **chmod** changes the file mode bits of each given file according to **MODE**, either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

The format of a symbolic mode is [**ugo**...][[-\*][**perms**...]]..., where **perms** is either zero or more letters from the set **rwxXs** letter from the set **ugo**. Multiple symbolic modes can be given, separated by commas.

A combination of the letters **ugoa** controls which users' access to the file will be changed: the user who owns it (**u**), other file's group (**g**), other users not in the file's group (**o**), or all users (**a**). If none of these are given, the effect is as given, but bits that are set in the umask are not affected.

The operator **+** causes the selected file mode bits to be added to the existing file mode bits of each file; **-** causes them to be removed; **=** causes them to be added and causes unmentioned bits to be removed except that a directory's unmentioned set user and group affected.

The letters **rwxXst** select file mode bits for the affected users: read (**r**), write (**w**), execute (or search for directory) or search for directory only if the file is a directory or already has execute permission for some user (**X**), set user or group ID on execute or restricted deletion flag or sticky bit (**t**). Instead of one or more of these letters, you can specify exactly one of the letter permissions granted to the user who owns the file (**u**), the permissions granted to other users who are members of the file's group or the permissions granted to users that are in neither of the two preceding categories (**o**).

A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Omitted digits are leading zeros. The first digit selects the set user ID (4) and set group ID (2) and restricted deletion or sticky (1); a second digit selects permissions for the user who owns the file: read (4), write (2), and execute (1); the third selects **p** other users in the file's group, with the same values; and the fourth for other users not in the file's group, with the same values.

**chmod** doesn't change the permissions of symbolic links; the **chmod** system call cannot change their permissions on most systems ignore permissions of symbolic links. However, for each symbolic link listed on the command line, **chmod** changes the permissions granted to file. In contrast, **chmod** ignores symbolic links encountered during recursive directory traversals. Options that havior are described in the **OPTIONS** section.

Manual page chmod(1) line 1 (press h for help or q to quit)

Рис. 2: Справочное описание

rodvish@rodvish:~ - man getfacl

---

**GETFACL(1)**

Access Control Lists

**GETFACL(1)**

**NAME**  
getfacl - get file access control lists

**SYNOPSIS**  
**getfacl** [-aceEsRLPtpndvh] **file** ...  
**getfacl** [-aceEsRLPtpndvh] -

**DESCRIPTION**  
For each **file**, **getfacl** displays the file name, owner, the group, and the Access Control List (ACL). If a directory has a default ACL, **getfacl** also displays the default ACL. Non-directories cannot have default ACLs.

If **getfacl** is used on a file system that does not support ACLs, **getfacl** displays the access permissions defined by the traditional file mode permission bits.

The output format of **getfacl** is as follows:

```

1: # file: setuid/
2: # file: setgid/
3: # group: staff
4: # flags: -
5: user::rwx
6: user:joe:rwx      #effective:r-x
7: group:rwx         #effective:r-x
8: group:cool:r-x
9: mask::r-x
10: other::r-
11: default:user:rwx
12: default:user:joe:rwx   #effective:r-x
13: default:group:r-x
14: default:mask:r-x
15: default:other:---

```

Lines 1-3 indicate the file name, owner, and owning group.

Line 4 indicates the setuid (s), setgid (s), and sticky (t) bits: either the letter representing the bit, or else a dash (-). This line is included if any of those bits is set and left out otherwise, so it will not be shown for most files. (See CONFORMANCE TO POSIX 1003.1e DRAFT STANDARD 17 below.)

Lines 5, 7 and 10 correspond to the user, group and other fields of the file mode permission bits. These three are called the base ACL entries.

Manual page getfacl(1) line 1 (press h for help or q to quit)

Рис. 3: Справочное описание

```

rodvish@rodvishh:~ - man setfACL
+                                         rodvish@rodvishh:~ - man setfACL
SETFACL(1)                                         Access Control Lists                                         SETFACL(1)

NAME
    setfACL - set file access control lists

SYNOPSIS
    setfACL [-bkndRLPvh] [[{-n|-x} acl_spec] [[{-M|-X} acl_file] file ...]
    setfACL --restore=[file|-]

DESCRIPTION
    This utility sets Access Control Lists (ACLs) of files and directories. On the command line, a sequence of commands is followed by a sequence of files (which in turn can be followed by another sequence of commands, ...).

    The -n and -x options expect an ACL on the command line. Multiple ACL entries are separated by comma characters (','). The -M and -X options read an ACL from a file or from standard input. The ACL entry format is described in Section ACL ENTRIES.

    The --set and --set-file options set the ACL of a file or a directory. The previous ACL is replaced. ACL entries for this operation must include permissions.

    The -m (--modify) and -M (--modify-file) options modify the ACL of a file or directory. ACL entries for this operation must include permissions.

    The -x (--remove) and -X (--remove-file) options remove ACL entries. It is not an error to remove an entry which does not exist. Only ACL entries without the perm field are accepted as parameters, unless POSIXLY_CORRECT is defined.

    When reading from files using the -M and -X options, setfACL accepts the output getfACL produces. There is at most one ACL entry per line. After a Pound sign ('#'), everything up to the end of the line is treated as a comment.

    If setfACL is used on a file system which does not support ACLs, setfACL operates on the file mode permission bits. If the ACL does not fit completely in the permission bits, setfACL modifies the file mode permission bits to reflect the ACL as closely as possible, writes an error message to standard error, and returns with an exit status greater than 0.

PERMISSIONS
    The file owner and processes capable of CAP_FOWNER are granted the right to modify ACLs of a file. This is analogous to the permissions required for accessing the file mode. (On current Linux systems, root is the only user with the CAP_FOWNER capability.)

OPTIONS
    -b, --remove-all
        Remove all extended ACL entries. The base ACL entries of the owner, group and others are retained.

Manual page setfACL(1) line 1 (press h for help or q to quit)

```

Рис. 4: Справочное описание

Открываю терминал с учетной записью root

```

rodvish@rodvishh:~$ sudo -i
[sudo] пароль для rodvish:
root@rodvishh:~#

```

Рис. 5: Учетная запись root

Создаю каталоги и проверяю владельца каталогов

```

root@rodvishh:~# mkdir -p /data/main /data/third
root@rodvishh:~# ls -Al /data
итого 0
drwxr-xr-x. 2 root root 6 сен 20 18:40 main
drwxr-xr-x. 2 root root 6 сен 20 18:40 third
root@rodvishh:~#

```

Рис. 6: Создание каталогов

Меняю владельцев каталогов

```
root@rodvishh:~# chgrp main /data/main
root@rodvishh:~# chgrp third /data/third
root@rodvishh:~# ls -Al /data
итого 0
drwxr-xr-x. 2 root main 6 сен 20 18:40 main
drwxr-xr-x. 2 root third 6 сен 20 18:40 third
root@rodvishh:~# █
```

Рис. 7: Замена владельцев

Устанавливаю необходимые разрешения

```
root@rodvishh:~# chmod //0 /data/main
root@rodvishh:~# chmod 770 /data/third
root@rodvishh:~# ls -ld /data/main /data/third
drwxrwx---. 2 root main 6 сен 20 18:40 /data/main
drwxrwx---. 2 root third 6 сен 20 18:40 /data/third
root@rodvishh:~# █
```

Рис. 8: Установка разрешений

Перехожу на учетную запись bob

```
rodvishh@rodvishh:~$ su - bob
Пароль:
Последняя неудачная попытка входа в систему: Сб сен 20 18:48:01 MSK 2025 на pts/2
Число неудачных попыток со времени последнего входа: 4.
bob@rodvishh:~$ █
```

Рис. 9: Учетная запись bob

Пытаюсь создать файл в каталоге main

```
bob@rodvishh:~$ cd /data/main
bob@rodvishh:/data/main$ touch emptyfile
bob@rodvishh:/data/main$ ls -Al
итого 0
-rw-r--r--. 1 bob bob 0 сен 20 18:50 emptyfile
bob@rodvishh:/data/main$
```

Рис. 10: Попытка создать файл

Пытаюсь создать файл в каталоге third

```
bob@rodvishh:~$ cd /data/third  
-bash: cd: /data/third: Отказано в доступе
```

Рис. 11: Попытка создать файл

Перехожу на учетную запись alice

```
bob@rodvishh:~$ su - alice  
Пароль:  
Последний вход в систему: Пт сен 19 22:50:29 MSK 2025 на pts/2  
alice@rodvishh:~$
```

Рис. 12: Учетная запись alice

Создаю два файла в каталоге main

```
alice@rodvishh:~$ cd /data/main  
alice@rodvishh:/data/main$ touch alice1  
alice@rodvishh:/data/main$ touch alice2  
alice@rodvishh:/data/main$ █
```

Рис. 13: Создание файлов

Перехожу на учетную запись bob

```
rodvish@rodvishh:~$ su - bob  
Пароль:  
Последний вход в систему: Сб сен 20 18:49:06 MSK 2025 на pts/2  
bob@rodvishh:~$
```

Рис. 14: Учётная запись bob

Попытка удалить файлы

```
bob@rodvishh:~$ cd /data/main
bob@rodvishh:/data/main$ ls -l
итого 0
-rw-r--r--. 1 alice alice 0 сен 20 18:55 alice1
-rw-r--r--. 1 alice alice 0 сен 20 18:56 alice2
-rw-r--r--. 1 bob   bob   0 сен 20 18:50 emptyfile
bob@rodvishh:/data/main$ rm -f alice*
bob@rodvishh:/data/main$ ls -l
итого 0
-rw-r--r--. 1 bob   bob   0 сен 20 18:50 emptyfile
bob@rodvishh:/data/main$
```

Рис. 15: Попытка удалить файлы

Создаю два файла

```
bob@rodvishh:/data/main$ touch bob1
bob@rodvishh:/data/main$ touch bob2
bob@rodvishh:/data/main$
```

Рис. 16: Создание файлов

Устанавливаю бит идентификатора группы и sticky-бит

```
rodvish@rodvishh:~$ sudo -i
[sudo] пароль для rodvish:
root@rodvishh:~# chmod g+s,o+t /data/main
root@rodvishh:~#
```

Рис. 17: Установка битов

Создаю файлы

```
alice@rodvishh:/data/main$ touch alice3
alice@rodvishh:/data/main$ touch alice4
alice@rodvishh:/data/main$ ls -l
итого 0
-rw-r--r--. 1 alice main 0 сен 20 19:01 alice3
-rw-r--r--. 1 alice main 0 сен 20 19:01 alice4
-rw-r--r--. 1 bob    bob   0 сен 20 18:58 bob1
-rw-r--r--. 1 bob    bob   0 сен 20 18:58 bob2
-rw-r--r--. 1 bob    bob   0 сен 20 18:50 emptyfile
alice@rodvishh:/data/main$
```

Рис. 18: Создание файлов

Пытаюсь удалить файлы

```
alice@rodvishh:/data/main$ rm -rf bob*
rm: невозможно удалить 'bob1': Операция не позволена
rm: невозможно удалить 'bob2': Операция не позволена
alice@rodvishh:/data/main$
```

Рис. 19: Попытка удалить файлы

Перехожу на учетную запись root

```
rodvish@rodvishh:~$ sudo -i
[sudo] пароль для rodvish:
root@rodvishh:~#
```

Рис. 20: Учётная запись root

Устанавливаю необходимые права

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

Рис. 21: Установка прав

Убеждаюсь в правильности установки разрешений

```
root@rodvishh:~# 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@rodvishh:~# 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@rodvishh:~#
```

Рис. 22: Проверка правильности установки разрешений

Создаю файл и проверяю текущие назначения полномочий

```
root@rodvishh:~# touch /data/main/newfile1
root@rodvishh:~# 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@rodvishh:~# getfacl /data/third/newfile1
getfacl: /data/third/newfile1: Нет такого файла или каталога
root@rodvishh:~# touch /data/third/newfile1
root@rodvishh:~# 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@rodvishh:~#
```

Рис. 23: Создание файла и проверка

#### Устанавливаю ACL

```
root@rodvishh:~# setfacl -m d:g:third:rwx /data/main
root@rodvishh:~#
```

Рис. 24: Установка ACL

```
root@rodvishh:~# setfacl -m d:g:main:rwx /data/third
root@rodvishh:~#
```

Рис. 25: Установка ACL

#### Проверяю работоспособность ACL

```

root@rodvishh:~# touch /data/main/newfile2
root@rodvishh:~# 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@rodvishh:~# touch /data/third/newfile2
root@rodvishh:~# 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@rodvishh:~#

```

Рис. 26: Проверка ACL

### Осуществляю заданные проверки

```

root@rodvishh:~# su - carol
Последний вход в систему: Пт сен 19 22:50:03 MSK 2025 на pts/2
carol@rodvishh:~$ rm /data/main/newfile1
rm: удалить защищенный от записи пустой обычный файл '/data/main/newfile1'?
carol@rodvishh:~$ rm /data/main/newfile2
rm: невозможно удалить '/data/main/newfile2': Отказано в доступе
carol@rodvishh:~$ echo "Hello, world" >> /data/main/newfile1
-bash: /data/main/newfile1: Отказано в доступе
carol@rodvishh:~$ echo "Hello, world" >> /data/main/newfile2
carol@rodvishh:~$ █

```

Рис. 27: Проверки

## **Вывод**

В ходе работы были получены практические навыки управления правами доступа в Linux. Освоены базовые механизмы (`chmod`, `chown`), специальные атрибуты (`setuid`, `setgid`, `sticky bit`) и расширенные списки доступа ACL (`setfacl`, `getfacl`). Приобретён опыт настройки прав для групп пользователей, проверки наследования разрешений и обеспечения безопасности данных. Цель работы достигнута.

# Контрольные вопросы

1. Вопрос: Как следует использовать команду chown, чтобы установить владельца группы для файла? Приведите пример. Ответ: Для установки владельца группы используется синтаксис `chown :GROUPNAME FILENAME`. Пример: `chown :developers myfile.txt`.
2. Вопрос: С помощью какой команды можно найти все файлы, принадлежащие конкретному пользователю? Приведите пример. Ответ: Команда `find / -user USERNAME`. Пример: `find /home -user ivanov`.
3. Вопрос: 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? Ответ: Команда `chmod -R ug=rwX, o= /data`. Флаг X устанавливает выполнение только для каталогов.
4. Вопрос: Какая команда позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым? Ответ: Команда `chmod +x FILENAME`. Пример: `chmod +x script.sh`.
5. Вопрос: Какая команда гарантирует наследование групповых разрешений для новых файлов в каталоге? Ответ: Команда `chmod g+s DIRECTORY`. Пример: `chmod g+s /shared/data`.
6. Вопрос: Как разрешить пользователям удалять только свои файлы в каталоге? Ответ: Команда `chmod +t DIRECTORY` устанавливает sticky bit. Пример: `chmod +t /tmp`.
7. Вопрос: Какая команда добавляет ACL для чтения членам группы для всех файлов в каталоге? Ответ: Команда `setfacl -m g:GROUPNAME:r *`. Пример: `setfacl -m g:readers:r *`.

8. Вопрос: Как гарантировать права на чтение для группы для всех существующих и будущих файлов в каталоге? Ответ: Команды: `setfacl -R -m g:GROUPNAME:r DIRECTORY`
9. Вопрос: Какое значение umask запретит права для “других” пользователей? Ответ: Значение `umask 007`. Новые файлы: 660 (rw-rw—), каталоги: 770 (rwxrwx—).
10. Вопрос: Как защитить файл от случайного удаления? Ответ: Команда `chmod a-w myfile` снимает права на запись у всех пользователей.