

# ***Отчет по выполнению лабораторной работы***

***Лабораторная работа №3***

Полина Витальевна Барабаш

# ***Содержание***

<b><i>1</i></b>	<b><i>Цель работы</i></b>	<b><i>6</i></b>
<b><i>2</i></b>	<b><i>Выполнение лабораторной работы</i></b>	<b><i>7</i></b>
<b><i>3</i></b>	<b><i>Ответы на контрольные вопросы</i></b>	<b><i>20</i></b>
<b><i>4</i></b>	<b><i>Выводы</i></b>	<b><i>28</i></b>

## Список иллюстраций

2.1	Справочное описание по командам . . . . .	7
2.2	Вход в учетную запись root . . . . .	7
2.3	Создание и проверка владельцев каталогов . . . . .	8
2.4	Изменение владельца каталогов . . . . .	8
2.5	Установка нужных разрешений . . . . .	9
2.6	Вход под учетную запись bob . . . . .	9
2.7	Создание файла в каталоге /data/main пользователем bob . . . . .	10
2.8	Попытка входа в каталог /data/third пользователем bob . . . . .	10
2.9	Создание двух файлов в каталоге /data/main пользователем alice . . . . .	11
2.10	Удаление файлов alice пользователем bob . . . . .	11
2.11	Создание двух файлов пользователем bob . . . . .	12
2.12	Установка бит идентификатора группы и также stiky-бит для разделяемого (общего) каталога группы . . . . .	12
2.13	Создание файлов alice3 и alice4 и проверка их владельца . . . . .	13
2.14	Попытка удалить файлы bob1 и bob2 пользователем alice . . . . .	13
2.15	Установка прав на чтение и выполнение для других групп . . . . .	14
2.16	Проверка правильности разрешений для каталога /data/main . . . . .	14
2.17	Проверка правильности разрешений для каталога /data/third . . . . .	15
2.18	Создание файла newfile1 в каталоге /data/main и проверка текущих полномочий . . . . .	15
2.19	Создание файла newfile1 в каталоге /data/third и проверка текущих полномочий . . . . .	16
2.20	Установка ACL по умолчанию для каталога /data/main и /data/third . . . . .	17
2.21	Создание файла newfile2 в каталоге /data/main и проверка текущих назначенных полномочий . . . . .	17
2.22	Создание файла newfile2 в каталоге /data/third и проверка текущих назначенных полномочий . . . . .	18
2.23	Вход в учетную запись carol . . . . .	18
2.24	Попытка удалить файлы членом группы third в каталоге /data/main . . . . .	19
2.25	Проверка возможности осуществить запись в файл . . . . .	19
3.1	Изменение владельца группы для файла с помощью chown . . . . .	20
3.2	Поиск всех файлов принадлежащих пользователю . . . . .	21
3.3	Применение нужных разрешений . . . . .	22
3.4	Добавление разрешения на исполнение файла . . . . .	22
3.5	Предоставление членам группы права доступа на чтение для всех существующих файлов в текущем каталоге . . . . .	24

3.6	Действия для гарантии того, что члены группы получают 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 . . . . .	25
3.7	Значение <code>umask</code> , чтобы «другие» пользователи не получали какие-либо разрешения на новые файлы . . . . .	26
3.8	Блокировка для удаления файла . . . . .	27

## ***Список таблиц***

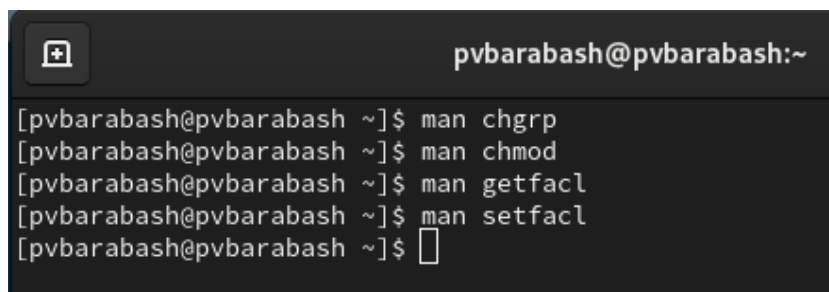
# ***1 Цель работы***

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

## 2 Выполнение лабораторной работы

**Задание 1.** Прочитайте справочное описание man по командам chgrp, chmod, getfacl, setfacl.

Я прочитала справочное описание команд, вводя man + команда (рис. [2.1]).

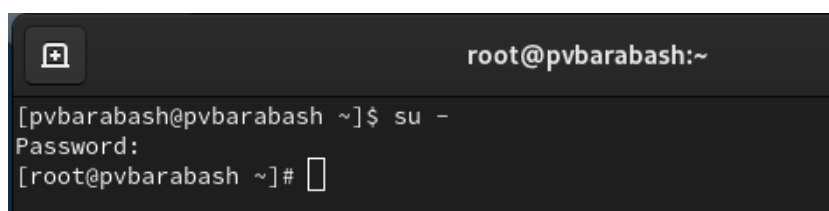


```
pvbarabash@pvbarabash:~$ man chgrp
pvbarabash@pvbarabash:~$ man chmod
pvbarabash@pvbarabash:~$ man getfacl
pvbarabash@pvbarabash:~$ man setfacl
pvbarabash@pvbarabash:~$
```

Рис. 2.1: Справочное описание по командам

**Задание 2.** Откройте терминал с учётной записью root.

Я ввела команду su - и перешла в учетную запись root (рис. [2.2]).

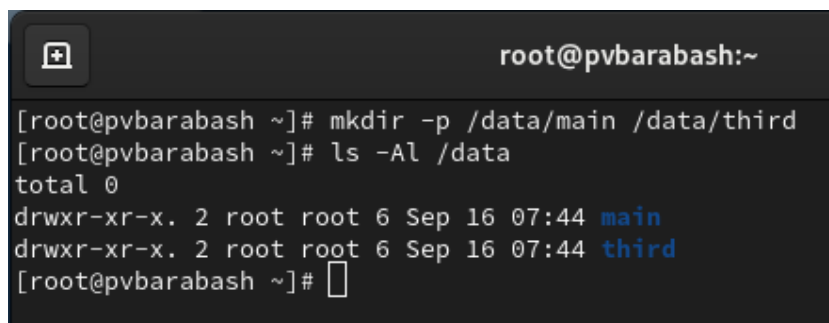


```
root@pvbarabash:~$ su -
Password:
[root@pvbarabash ~]#
```

Рис. 2.2: Вход в учетную запись root

**Задание 3.** В корневом каталоге создайте каталоги /data/main и /data/third. Посмотрите, кто является владельцем этих каталогов.

Я ввела команду `mkdir -p /data/main /data/third`, чтобы создать два каталога. А затем командой `ls -Al /data` вывела информацию о владельцах этих каталогов (рис. [2.3]).

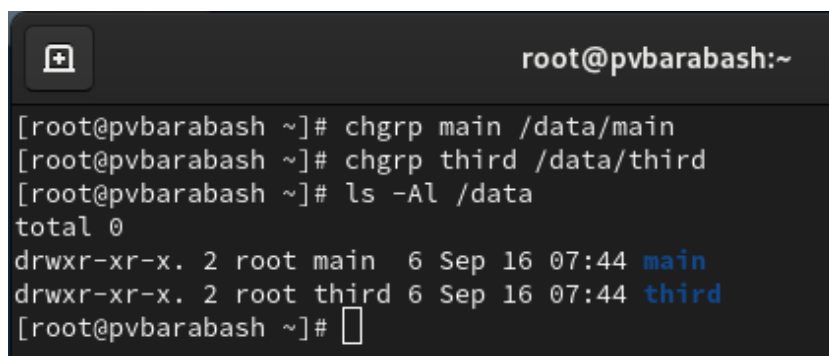
A terminal window titled 'root@pvbarabash:~' with a window icon in the top-left corner. The terminal shows the following commands and output:

```
[root@pvbarabash ~]# mkdir -p /data/main /data/third
[root@pvbarabash ~]# ls -Al /data
total 0
drwxr-xr-x. 2 root root 6 Sep 16 07:44 main
drwxr-xr-x. 2 root root 6 Sep 16 07:44 third
[root@pvbarabash ~]#
```

Рис. 2.3: Создание и проверка владельцев каталогов

**Задание 4.** Прежде чем устанавливать разрешения, измените владельцев этих каталогов с root на main и third соответственно. Посмотрите, кто теперь является владельцем этих каталогов.

Я изменила владельцев каталогов с помощью команды `chgrp`. Затем проверила, что теперь действительно владельцами являются main и third соответственно с помощью той же команды `ls -Al /data` (рис. [2.4]).

A terminal window titled 'root@pvbarabash:~' with a window icon in the top-left corner. The terminal shows the following commands and output:

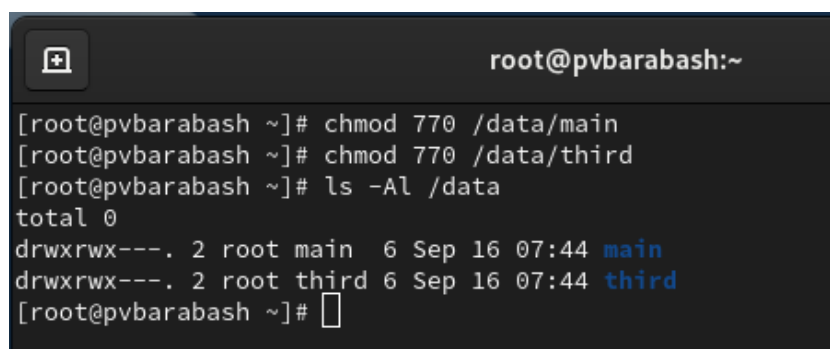
```
[root@pvbarabash ~]# chgrp main /data/main
[root@pvbarabash ~]# chgrp third /data/third
[root@pvbarabash ~]# ls -Al /data
total 0
drwxr-xr-x. 2 root main 6 Sep 16 07:44 main
drwxr-xr-x. 2 root third 6 Sep 16 07:44 third
[root@pvbarabash ~]#
```

Рис. 2.4: Изменение владельца каталогов

**Задание 5.** Установите разрешения, позволяющие владельцам каталогов записывать файлы в эти каталоги и запрещающие доступ к содержимому каталогов всем другим пользователям и группам.



С помощью команды `chmod 770` я установила нужные разрешения и проверила, что разрешения установлены верные с помощью все той же команды `ls -Al /data` (рис. [2.5]).

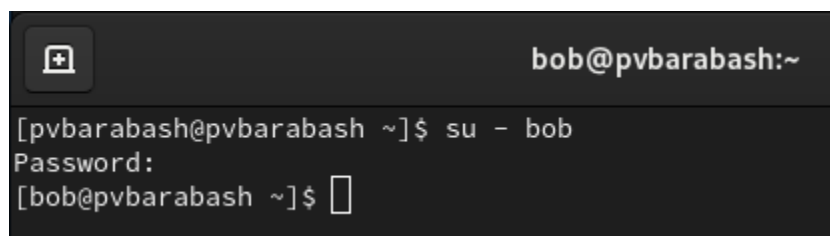
A terminal window titled 'root@pvbarabash:~' showing a series of commands and their output. The user runs 'chmod 770 /data/main', then 'chmod 770 /data/third', and finally 'ls -Al /data'. The output shows two directories, 'main' and 'third', both with permissions 'drwxrwx---', owned by 'root', and created on '6 Sep 16 07:44'.

```
root@pvbarabash:~  
[root@pvbarabash ~]# chmod 770 /data/main  
[root@pvbarabash ~]# chmod 770 /data/third  
[root@pvbarabash ~]# ls -Al /data  
total 0  
drwxrwx---. 2 root main  6 Sep 16 07:44 main  
drwxrwx---. 2 root third 6 Sep 16 07:44 third  
[root@pvbarabash ~]#
```

Рис. 2.5: Установка нужных разрешений

**Задание 6.** В другом терминале перейдите под учётную запись пользователя bob.

Я использовала команду `su - bob`, чтобы войти под учетную запись пользователя bob (рис. [2.6]).

A terminal window titled 'bob@pvbarabash:~' showing the user switching from root to bob. The prompt changes from root@ to pvbarabash@, and the user enters 'su - bob'. A password prompt is shown, and after entering the password, the prompt changes to bob@.

```
bob@pvbarabash:~  
[pvbarabash@pvbarabash ~]$ su - bob  
Password:  
[bob@pvbarabash ~]$
```

Рис. 2.6: Вход под учетную запись bob

**Задание 7.** Под пользователем bob попробуйте перейти в каталог `/data/main` и создать файл `emptyfile` в этом каталоге

Под пользователем bob я перешла в каталог `/data/main` с помощью команды `cd` и с помощью команды `touch` создала файл `emptyfile` (рис. [2.7]).

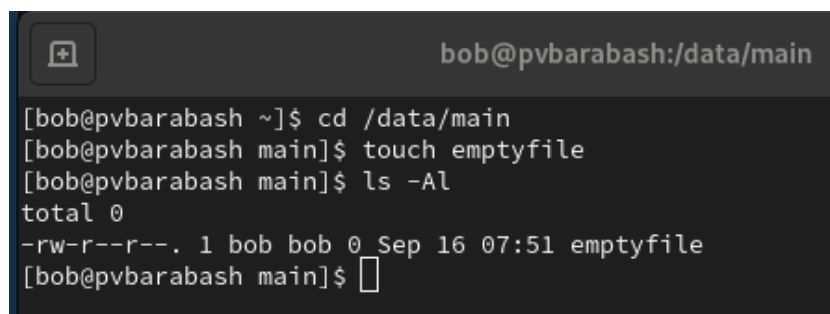
A terminal window with a dark background. The title bar shows a window icon and the text 'bob@pvbarabash:/data/main'. The terminal content shows a series of commands and their outputs: first, 'cd /data/main' is executed; then, 'touch emptyfile' is executed; next, 'ls -Al' is executed, resulting in 'total 0' and a file listing for 'emptyfile' with permissions '-rw-r--r--', owner 'bob', group 'bob', size '0', and timestamp 'Sep 16 07:51'; finally, the prompt returns to '[bob@pvbarabash main]\$'.

Рис. 2.7: Создание файла в каталоге /data/main пользователем bob

Это возможно, так как bob входит в группу main, а мы установили разрешения, позволяющие владельцем каталогов записывать файлы в эти каталоги в задании 5.

**Задание 8.** Под пользователем bob попробуйте перейти в каталог /data/third и создать файл emptyfile в этом каталоге.

Я попробовала перейти в каталог /data/third и получила информацию, что данное действие запрещено (рис. [2.8]).

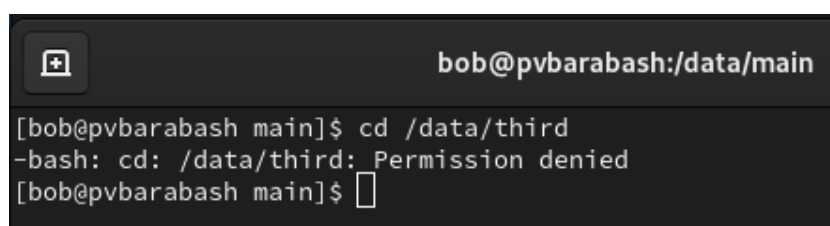
A terminal window with a dark background. The title bar shows a window icon and the text 'bob@pvbarabash:/data/main'. The terminal content shows the command 'cd /data/third' being executed, which results in the error message '-bash: cd: /data/third: Permission denied'. The prompt then returns to '[bob@pvbarabash main]\$'.

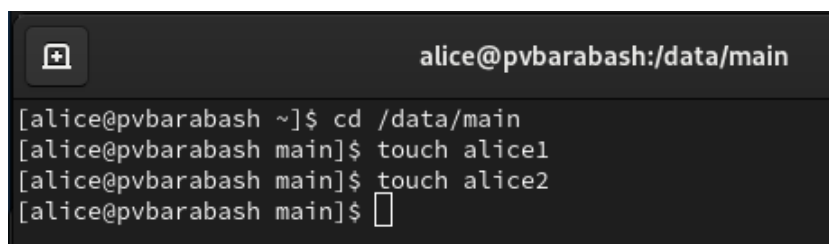
Рис. 2.8: Попытка входа в каталог /data/third пользователем bob

Это произошло, так как bob не входит в группу third, которая является владельцем данного каталога. А в задании 5, мы установили разрешение, запрещающее доступ к содержимому каталогов всем другим пользователям и группам, кроме владельца.

**Задание 9.** Откройте новый терминал под пользователем alice. Перейдите в каталог /data/main. Создайте два файла, владельцем которых является alice.

Я открыла новый терминал и вошла в учетную запись alice с помощью su -

alice. Затем я перешла в каталог /data/main с помощью команды cd и с помощью команды touch создала два файла alice1 и alice2 (рис. [2.9]).

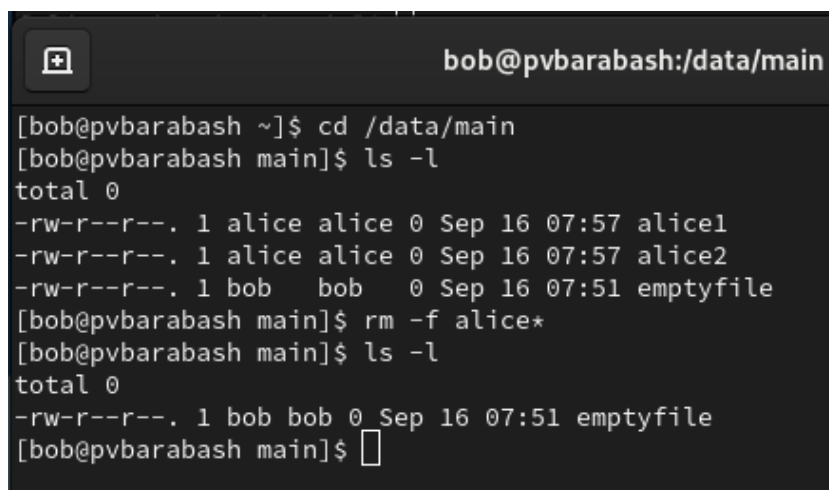


```
alice@pvbarabash:/data/main
[alice@pvbarabash ~]$ cd /data/main
[alice@pvbarabash main]$ touch alice1
[alice@pvbarabash main]$ touch alice2
[alice@pvbarabash main]$
```

Рис. 2.9: Создание двух файлов в каталоге /data/main пользователем alice

**Задание 10.** В другом терминале перейдите под учётную запись пользователя bob. Перейдите в каталог /data/main и в этом каталоге введите ls -l. Попробуйте удалить файлы, принадлежащие пользователю alice. Убедитесь, что файлы будут удалены пользователем bob.

В другом терминале я перешла под учетную запись пользователя bob. Перешла в каталог /data/main с помощью команды cd и ввела ls -l, что позволило мне убедиться, что файлы alice видны и пользователю bob. Затем я использовала команду rm -f alice\*, чтобы удалить файлы, принадлежащие alice. Я вновь ввела команду ls -l и убедилась, что файлы удалены (рис. [2.10]).



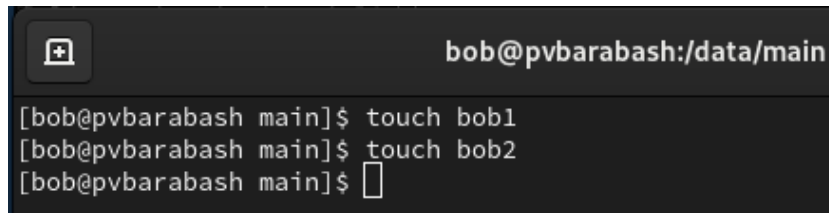
```
bob@pvbarabash:/data/main
[bob@pvbarabash ~]$ cd /data/main
[bob@pvbarabash main]$ ls -l
total 0
-rw-r--r--. 1 alice alice 0 Sep 16 07:57 alice1
-rw-r--r--. 1 alice alice 0 Sep 16 07:57 alice2
-rw-r--r--. 1 bob   bob   0 Sep 16 07:51 emptyfile
[bob@pvbarabash main]$ rm -f alice*
[bob@pvbarabash main]$ ls -l
total 0
-rw-r--r--. 1 bob bob 0 Sep 16 07:51 emptyfile
[bob@pvbarabash main]$
```

Рис. 2.10: Удаление файлов alice пользователем bob

Это возможно, так как bob также входит в группу main и имеет полные права на все файлы, владельцем которых является группа main.

**Задание 11.** Создайте два файла, которые принадлежат пользователю bob.

С помощью команды touch я создала два файла bob1 и bob2 (рис. [2.11]).

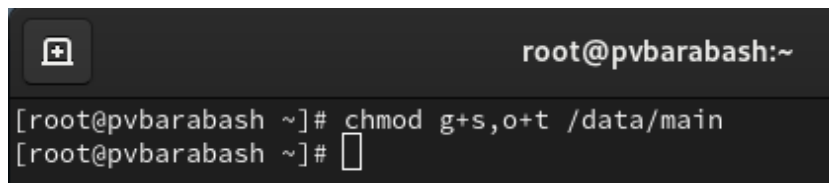


```
bob@pvbarabash:/data/main
[ bob@pvbarabash main]$ touch bob1
[ bob@pvbarabash main]$ touch bob2
[ bob@pvbarabash main]$
```

Рис. 2.11: Создание двух файлов пользователем bob

**Задание 12.** В терминале под пользователем root установите для каталога /data/main бит идентификатора группы, а также sticky-бит для разделяемого (общего) каталога группы.

В терминале под пользователем root я установила для каталога /data/main бит идентификатора группы, а также sticky-бит для разделяемого (общего) каталога группы с помощью команды `chmod g+s,o+t /data/main` (рис. [2.12]).

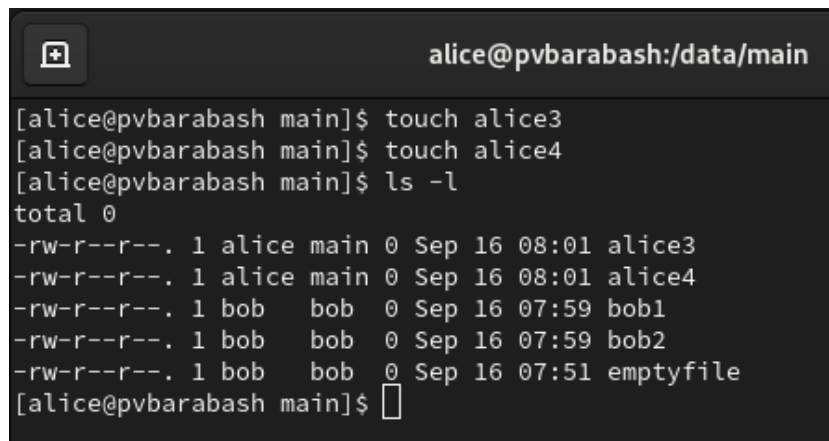


```
root@pvbarabash:~
[ root@pvbarabash ~]# chmod g+s,o+t /data/main
[ root@pvbarabash ~]#
```

Рис. 2.12: Установка бит идентификатора группы и также sticky-бит для разделяемого (общего) каталога группы

**Задание 13.** В терминале под пользователем alice создайте в каталоге /data/main файлы alice3 и alice4.

Под пользователем alice я создала в каталоге /data/main файлы alice3 и alice4 с помощью команды touch. Затем я вывела информацию о файлах каталога с помощью `ls -l` и убедилась, что теперь два созданных файла принадлежат группе main, которая является группой-владельцем каталога /data/main (рис. [2.13]).

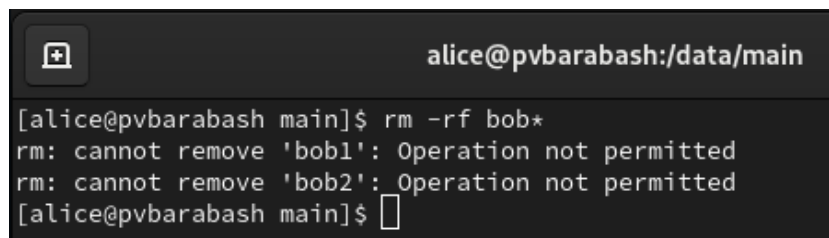
A terminal window titled 'alice@pvbarabash:/data/main'. The user runs the following commands: 'touch alice3', 'touch alice4', and 'ls -l'. The output of 'ls -l' shows a list of files: 'alice3' and 'alice4' owned by 'alice' in the 'main' directory, and 'bob1', 'bob2', and 'emptyfile' owned by 'bob' in the 'main' directory. The permissions for all files are '-rw-r--r--'.

```
alice@pvbarabash:/data/main
[alice@pvbarabash main]$ touch alice3
[alice@pvbarabash main]$ touch alice4
[alice@pvbarabash main]$ ls -l
total 0
-rw-r--r--. 1 alice main 0 Sep 16 08:01 alice3
-rw-r--r--. 1 alice main 0 Sep 16 08:01 alice4
-rw-r--r--. 1 bob   bob   0 Sep 16 07:59 bob1
-rw-r--r--. 1 bob   bob   0 Sep 16 07:59 bob2
-rw-r--r--. 1 bob   bob   0 Sep 16 07:51 emptyfile
[alice@pvbarabash main]$
```

Рис. 2.13: Создание файлов alice3 и alice4 и проверка их владельца

**Задание 14.** В терминале под пользователем alice попробуйте удалить файлы, принадлежащие пользователю bob.

Я попробовала удалить файлы bob1 и bob2 пользователем alice с помощью команды `rm -rf bob*`, однако sticky-bit предотвратит удаление этих файлов пользователем alice, поскольку этот пользователь не является владельцем этих файлов (рис. [2.14]).

A terminal window titled 'alice@pvbarabash:/data/main'. The user runs the command 'rm -rf bob\*'. The output shows two error messages: 'rm: cannot remove 'bob1': Operation not permitted' and 'rm: cannot remove 'bob2': Operation not permitted'.

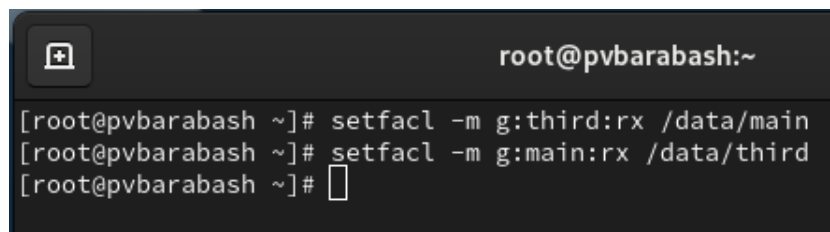
```
alice@pvbarabash:/data/main
[alice@pvbarabash main]$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
[alice@pvbarabash main]$
```

Рис. 2.14: Попытка удалить файлы bob1 и bob2 пользователем alice

**Задание 15.** Откройте терминал с учётной записью root. Установите права на чтение и выполнение в каталоге /data/main для группы third и права на чтение и выполнение для группы main в каталоге /data/third.

Под пользователем root я установила права на чтение и выполнение в каталоге /data/main для группы third и права на чтение и выполнение для группы main в каталоге /data/third с помощью команд `setfacl -m g:third:rx /data/main` и `setfacl`

-m g:main:rx /data/third (рис. [2.15]).

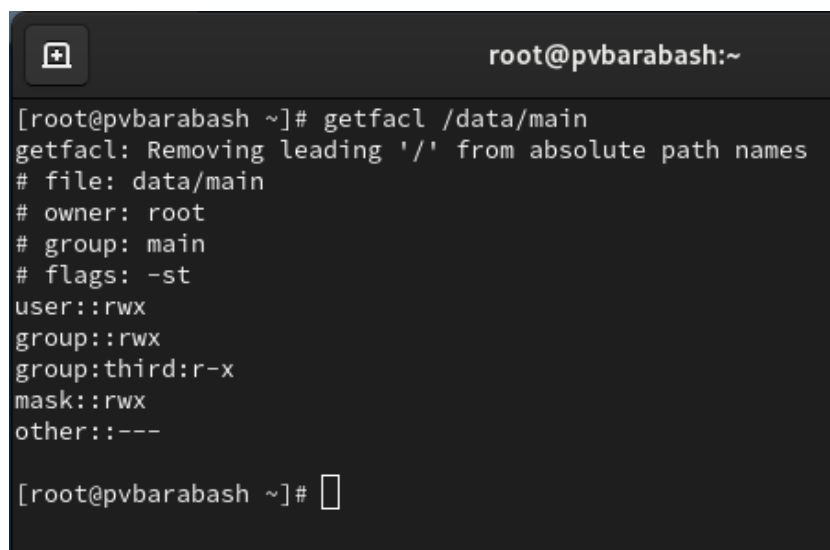
A terminal window titled 'root@pvbarabash:~' showing the execution of two 'setfacl' commands. The first command is 'setfacl -m g:third:rx /data/main' and the second is 'setfacl -m g:main:rx /data/third'. The prompt returns after each command.

```
root@pvbarabash:~  
[root@pvbarabash ~]# setfacl -m g:third:rx /data/main  
[root@pvbarabash ~]# setfacl -m g:main:rx /data/third  
[root@pvbarabash ~]#
```

Рис. 2.15: Установка прав на чтение и выполнение для других групп

**Задание 16.** Используйте команду `getfacl`, чтобы убедиться в правильности установки разрешений.

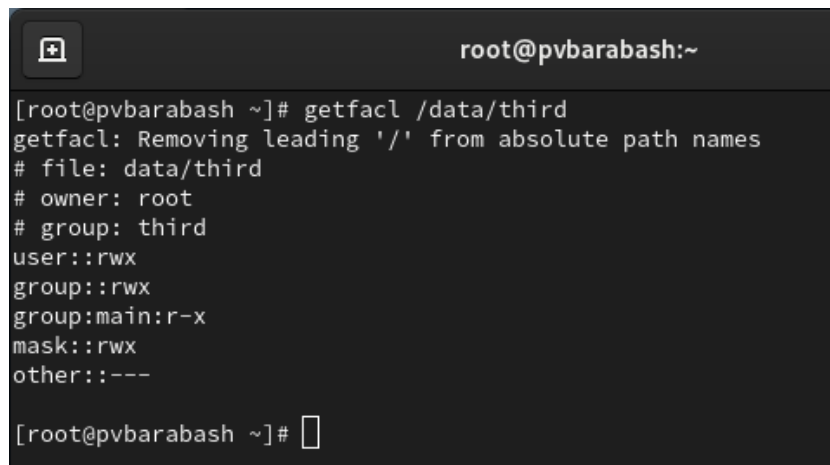
Я использовала команду `getfacl /data/main`, чтобы убедиться в правильности установки разрешений для каталога `/data/main` (рис. [2.16]).

A terminal window titled 'root@pvbarabash:~' showing the output of the 'getfacl /data/main' command. The output lists file details and permissions for user, group, mask, and other.

```
root@pvbarabash:~  
[root@pvbarabash ~]# 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@pvbarabash ~]#
```

Рис. 2.16: Проверка правильности разрешений для каталога `/data/main`

Затем я использовала команду `getfacl /data/third`, чтобы убедиться в правильности установки разрешений для каталога `/data/third` (рис. [2.17]).

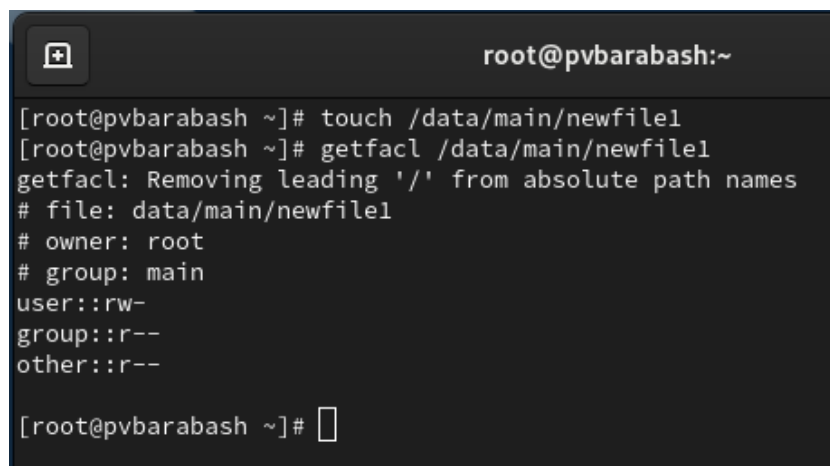


```
root@pvbarabash:~  
[root@pvbarabash ~]# getfacl /data/third  
getfacl: Removing leading '/' from absolute path names  
# file: data/third  
# owner: root  
# group: third  
user::rwx  
group::rwx  
group:main:r-x  
mask::rwx  
other:---  
[root@pvbarabash ~]#
```

Рис. 2.17: Проверка правильности разрешений для каталога /data/third

**Задание 17.** Создайте новый файл с именем newfile1 в каталоге /data/main. Используйте `getfacl /data/main/newfile1` для проверки текущих назначений полномочий. Какие права доступа у этого файла? Объясните, почему.

Я создала новый файл с именем newfile1 в каталоге /data/main с помощью команды `touch`. Затем я использовала команду `getfacl /data/main/newfile1` для проверки текущих назначений полномочий (рис. [2.18]).



```
root@pvbarabash:~  
[root@pvbarabash ~]# touch /data/main/newfile1  
[root@pvbarabash ~]# 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@pvbarabash ~]#
```

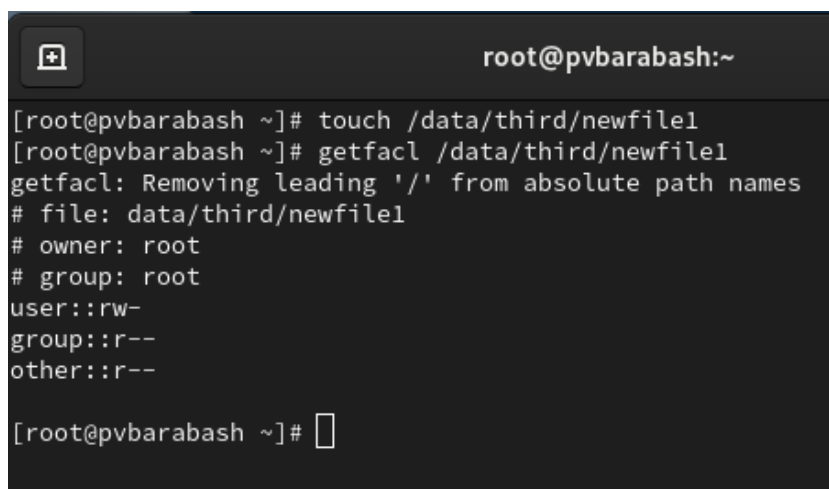
Рис. 2.18: Создание файла newfile1 в каталоге /data/main и проверка текущих полномочий

Как можно видеть на скриншоте, у пользователя стоит разрешение `rw-`, а у

группы и остальных r-. Думаю, это дефолтные разрешения для вновь созданного файла, как можно видеть такие же разрешения были для файлов alice и файлов bob.

**Задание 18.** Выполните аналогичные действия для каталога /data/third. Дайте пояснения.

Я выполнила аналогичные действия для каталога /data/third и получила почти те же самые результаты. Отличие заключается в том, что владеющая группа – root, а не third (рис. [2.19]).



```
root@pvbarabash:~  
[root@pvbarabash ~]# touch /data/third/newfile1  
[root@pvbarabash ~]# 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@pvbarabash ~]#
```

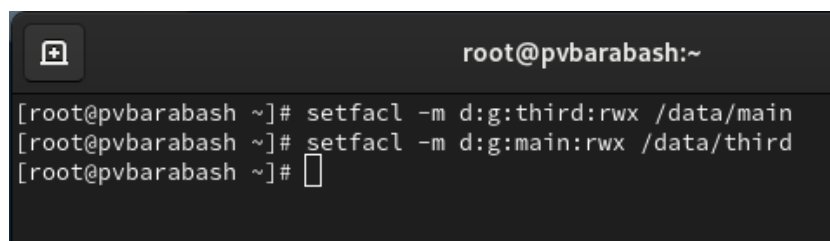
Рис. 2.19: Создание файла newfile1 в каталоге /data/third и проверка текущих полномочий

Так произошло, так как бит идентификатора группы, а также sticky-бит для разделяемого (общего) каталога группы мы добавляли только для группы main, а не third.

**Задание 19.** Установите ACL по умолчанию для каталога /data/main. Добавьте ACL по умолчанию для каталога /data/third.

Я установила ACL по умолчанию для каталога /data/main с помощью команды setfacl -m d:g:third:rwX /data/main и добавила ACL по умолчанию для каталога /data/third с помощью команды setfacl -m d:g:main:rwX /data/third (рис. [2.20]).



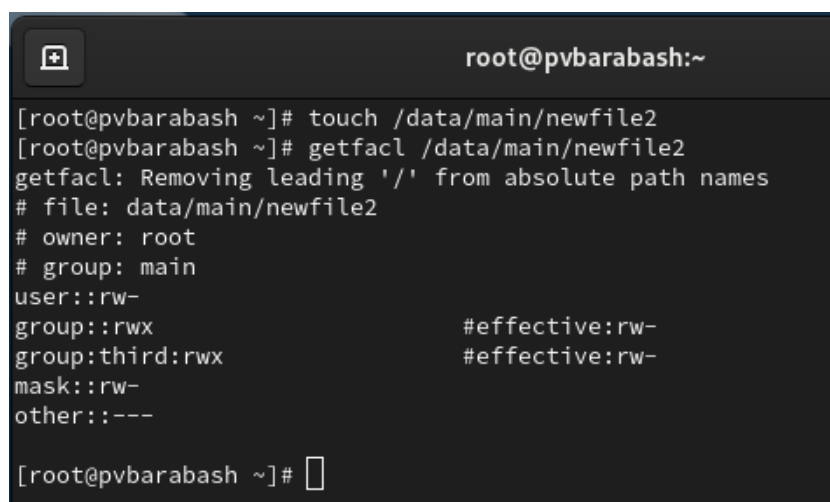
A terminal window with a dark background and light text. The title bar shows a window icon and the text 'root@pvbarabash:~'. The terminal contains three lines of commands and their output:

```
[root@pvbarabash ~]# setfacl -m d:g:third:rwX /data/main
[root@pvbarabash ~]# setfacl -m d:g:main:rwX /data/third
[root@pvbarabash ~]#
```

Рис. 2.20: Установка ACL по умолчанию для каталога /data/main и /data/third

**Задание 20.** Убедитесь, что настройки ACL работают, добавив новый файл в каталог /data/main. Используйте `getfacl /data/main/newfile2` для проверки текущих назначений полномочий.

Я создала файл `newfile2` в каталоге /data/main с помощью команды `touch`, а затем проверила текущие назначенные полномочия с помощью команды `getfacl /data/main/newfile2` (рис. [2.21]).

A terminal window with a dark background and light text. The title bar shows a window icon and the text 'root@pvbarabash:~'. The terminal contains the following commands and output:

```
[root@pvbarabash ~]# touch /data/main/newfile2
[root@pvbarabash ~]# 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::---
```

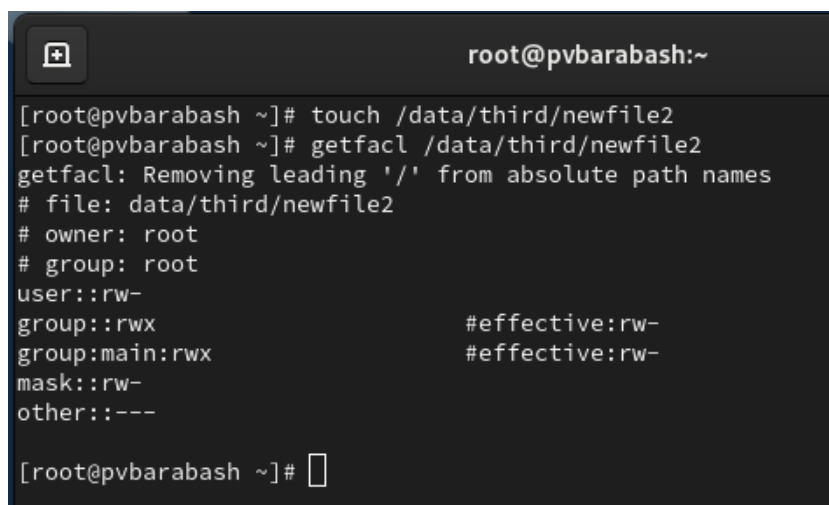
Рис. 2.21: Создание файла `newfile2` в каталоге /data/main и проверка текущих назначенных полномочий

Как можно видеть, настройки ACL работают, все права такие, какими мы их назначили на предыдущем шаге.

**Задание 21.** Выполните аналогичные действия для каталога /data/third.

Я выполнила аналогичные действия для каталога /data/third и также проверила,

что настройки ACL работают (рис. [2.22]).

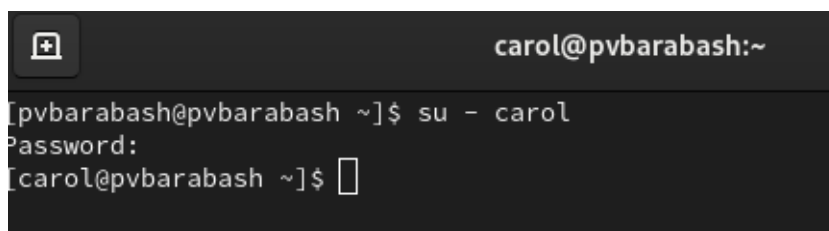


```
root@pvbarabash:~  
[root@pvbarabash ~]# touch /data/third/newfile2  
[root@pvbarabash ~]# getfacl /data/third/newfile2  
getfacl: Removing leading '/' from absolute path names  
# file: data/third/newfile2  
# owner: root  
# group: root  
user::rw-  
group::rwx          #effective:rwx  
group:main:rwx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

Рис. 2.22: Создание файла newfile2 в каталоге /data/third и проверка текущих назначенных полномочий

**Задание 22.** Для проверки полномочий группы third в каталоге /data/third войдите в другом терминале под учётной записью члена группы third.

В новом терминале я вошла в учетную запись пользователя carol, так как она член группы third (рис. [2.23]).

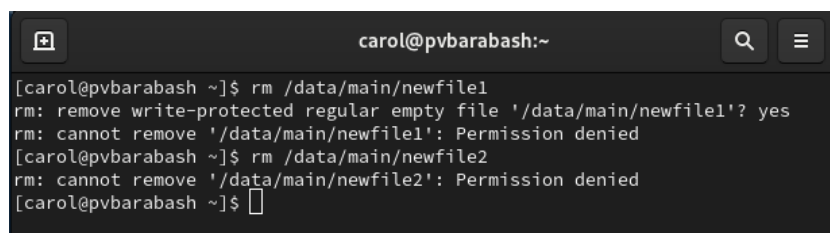


```
carol@pvbarabash:~  
[pvbarabash@pvbarabash ~]$ su - carol  
Password:  
[carol@pvbarabash ~]$
```

Рис. 2.23: Вход в учетную запись carol

**Задание 23.** Попробуйте удалить файлы newfile1 и newfile2 в каталоге /data/main.

Я попробовала удалить файлы newfile1 и newfile2 в каталоге /data/main с помощью команд `rm /data/main/newfile1` и `rm /data/main/newfile2` (рис. [2.24]).

A terminal window titled 'carol@pvbarabash:~' with a search icon and a menu icon in the top right. The terminal shows the following commands and output:

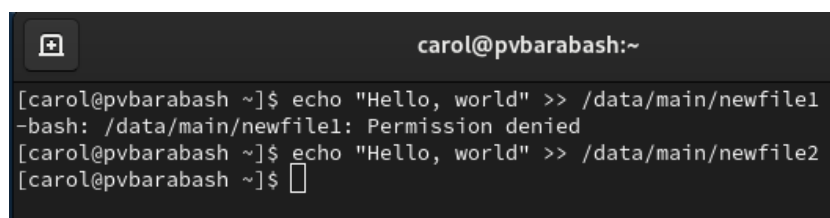
```
[carol@pvbarabash ~]$ rm /data/main/newfile1
rm: remove write-protected regular empty file '/data/main/newfile1'? yes
rm: cannot remove '/data/main/newfile1': Permission denied
[carol@pvbarabash ~]$ rm /data/main/newfile2
rm: cannot remove '/data/main/newfile2': Permission denied
[carol@pvbarabash ~]$
```

Рис. 2.24: Попытка удалить файлы членом группы third в каталоге /data/main

Для членов группы third нет таких полномочий, поэтому действие не было выполнено.

**Задание 24.** Проверьте, возможно ли осуществить запись в файл.

Я проверила, возможно ли осуществить запись в файл с помощью команд `echo "Hello, world" >> /data/main/newfile1` и `echo "Hello, world" >> /data/main/newfile2` (рис. [2.25]).

A terminal window titled 'carol@pvbarabash:~' with a search icon and a menu icon in the top right. The terminal shows the following commands and output:

```
[carol@pvbarabash ~]$ echo "Hello, world" >> /data/main/newfile1
-bash: /data/main/newfile1: Permission denied
[carol@pvbarabash ~]$ echo "Hello, world" >> /data/main/newfile2
[carol@pvbarabash ~]$
```

Рис. 2.25: Проверка возможности осуществить запись в файл

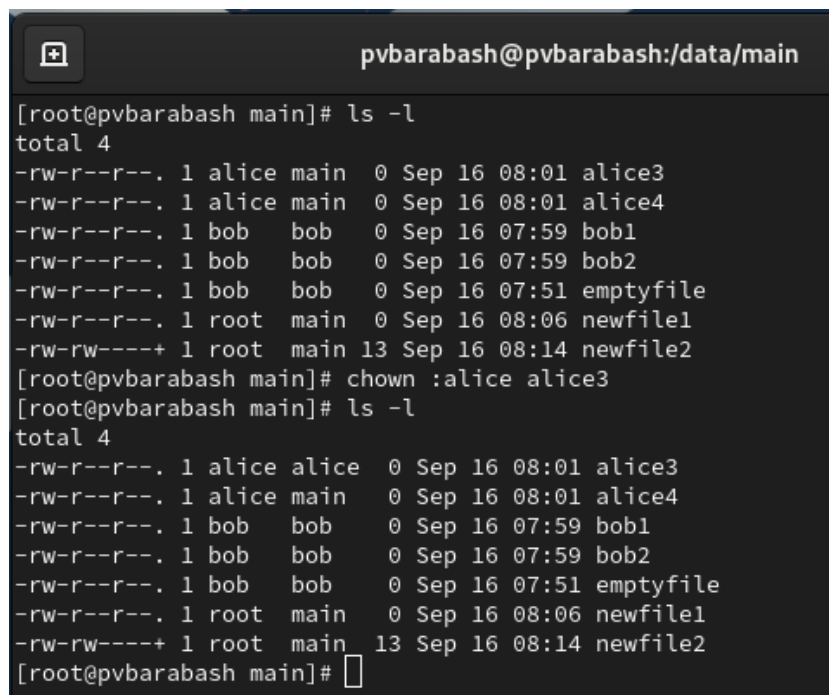
В первый файл не удалось ничего записать, а во второй файл удалось. Это связано с тем, что до создания newfile2 мы установили настройки ACL, позволяющие группе third записывать информацию в файлы группы main, поэтому к файлу newfile2 эти настройки были применены.

### 3 Ответы на контрольные вопросы

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

Команда `chown` используется для изменения владельца и группы файла. Чтобы установить только группу, можно использовать двоеточие.

Пример: файл `alice3` принадлежал группе `main` (первая строка вывода `ls -l`). С помощью команды `chown :alice alice3` я изменила владельца группы, теперь, как можно видеть по новому выводу `ls -l`, владельцем группы файла `alice3` является группа `alice` (рис. [3.1]).



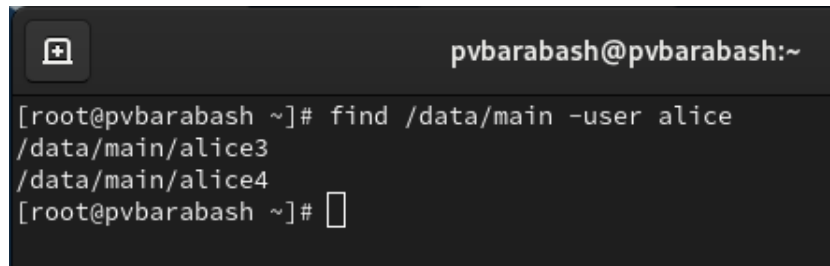
```
pvbarabash@pvbarabash:/data/main
[root@pvbarabash main]# ls -l
total 4
-rw-r--r--. 1 alice main  0 Sep 16 08:01 alice3
-rw-r--r--. 1 alice main  0 Sep 16 08:01 alice4
-rw-r--r--. 1 bob  bob   0 Sep 16 07:59 bob1
-rw-r--r--. 1 bob  bob   0 Sep 16 07:59 bob2
-rw-r--r--. 1 bob  bob   0 Sep 16 07:51 emptyfile
-rw-r--r--. 1 root  main  0 Sep 16 08:06 newfile1
-rw-rw----+ 1 root  main 13 Sep 16 08:14 newfile2
[root@pvbarabash main]# chown :alice alice3
[root@pvbarabash main]# ls -l
total 4
-rw-r--r--. 1 alice alice  0 Sep 16 08:01 alice3
-rw-r--r--. 1 alice main  0 Sep 16 08:01 alice4
-rw-r--r--. 1 bob  bob   0 Sep 16 07:59 bob1
-rw-r--r--. 1 bob  bob   0 Sep 16 07:59 bob2
-rw-r--r--. 1 bob  bob   0 Sep 16 07:51 emptyfile
-rw-r--r--. 1 root  main  0 Sep 16 08:06 newfile1
-rw-rw----+ 1 root  main 13 Sep 16 08:14 newfile2
[root@pvbarabash main]#
```

Рис. 3.1: Изменение владельца группы для файла с помощью `chown`

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

Можно использовать команду `find`, добавив опцию `-user`.

Пример: я использовала команду `find /data/main -user alice`, чтобы узнать, какие файлы принадлежат `alice` в каталоге `/data/main`. Команда вывела список файлов (рис. [3.2]).

A screenshot of a terminal window with a dark background. The title bar at the top shows a window icon on the left and the text 'pvbarabash@pvbarabash:~' on the right. The terminal content shows a root prompt '[root@pvbarabash ~]#', followed by the command 'find /data/main -user alice'. The output consists of two lines: '/data/main/alice3' and '/data/main/alice4'. Below the output, the prompt '[root@pvbarabash ~]#' is shown again with a cursor, indicating the command has finished execution.

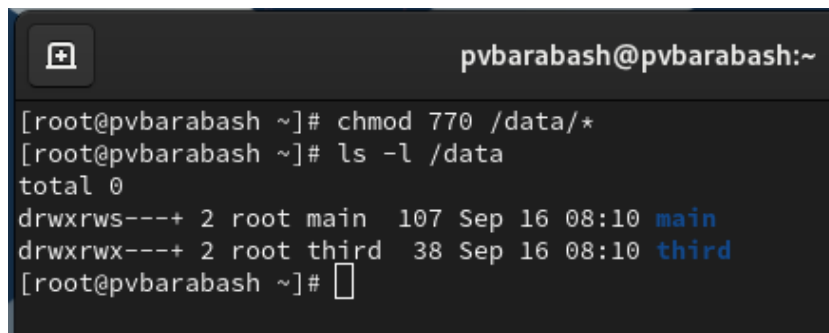
```
[root@pvbarabash ~]# find /data/main -user alice
/data/main/alice3
/data/main/alice4
[root@pvbarabash ~]#
```

Рис. 3.2: Поиск всех файлов принадлежащих пользователю

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.

Мы выполняли аналогичное задание в пункте 5, для этого нужно использовать команду `chmod 770`, которая устанавливает нужные права.

Пример: я использовала команду `chmod 770 /data/*`, чтобы применить разрешения на чтение, запись и выполнение для всех файлов в каталоге `/data` для пользователей и владельцев групп, не устанавливая никаких прав для других (рис. [3.3]).



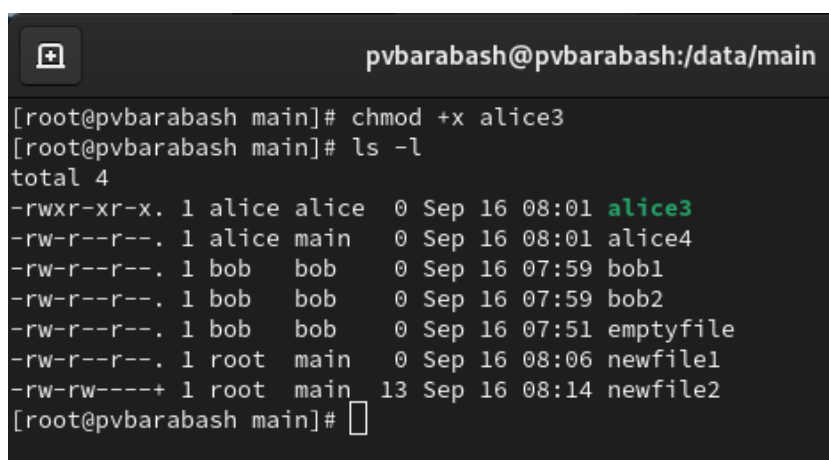
```
pvbarabash@pvbarabash:~  
[root@pvbarabash ~]# chmod 770 /data/*  
[root@pvbarabash ~]# ls -l /data  
total 0  
drwxrws---+ 2 root main 107 Sep 16 08:10 main  
drwxrwx---+ 2 root third 38 Sep 16 08:10 third  
[root@pvbarabash ~]#
```

Рис. 3.3: Применение нужных разрешений

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

Команда `chmod +x` позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым.

Пример: я добавила всё тому же файлу `alice3` разрешение на выполнение с помощью команды `chmod +x alice3` (рис. [3.4]).



```
pvbarabash@pvbarabash:/data/main  
[root@pvbarabash main]# chmod +x alice3  
[root@pvbarabash main]# ls -l  
total 4  
-rwxr-xr-x. 1 alice alice 0 Sep 16 08:01 alice3  
-rw-r--r--. 1 alice main 0 Sep 16 08:01 alice4  
-rw-r--r--. 1 bob bob 0 Sep 16 07:59 bob1  
-rw-r--r--. 1 bob bob 0 Sep 16 07:59 bob2  
-rw-r--r--. 1 bob bob 0 Sep 16 07:51 emptyfile  
-rw-r--r--. 1 root main 0 Sep 16 08:06 newfile1  
-rw-rw----+ 1 root main 13 Sep 16 08:14 newfile2  
[root@pvbarabash main]#
```

Рис. 3.4: Добавление разрешения на исполнение файла

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

Команда `chmod` с флагом `g+s` позволяет убедиться, что групповые разрешения для всех новых файлов, создаваемых в каталоге, будут присвоены владельцу группы этого каталога. Пример можно найти в задании 12.

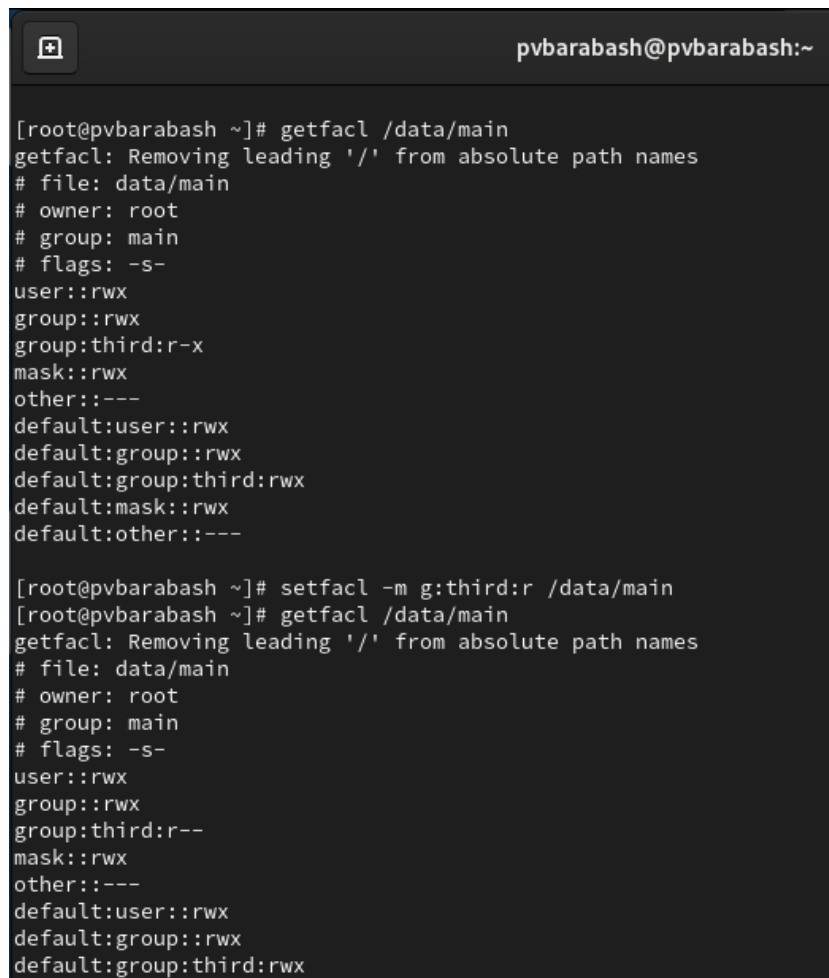
6. Необходимо, чтобы пользователи могли удалять только те файлы, владельцами которых они являются, или которые находятся в каталоге, владельца которого они являются. С помощью какой команды можно это сделать? Приведите пример.

Это позволяет сделать установка `sticky`-бит для каталога (`chmod o+t`). Пример всё то же задание 12 и последующие за ним.

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

Команда `setfacl -m g:groupname:r` предоставляет членам группы права доступа на чтение для всех существующих файлов в текущем каталоге.

Пример: для каталога `/data/main` были установлены разрешения `r-x` для группы `third`, с помощью команды `setfacl -m g:third:r /data/main` я установила разрешение только на чтение и проверила, что действительно права изменились (рис. [3.5]).



```
pvbarabash@pvbarabash:~  
[root@pvbarabash ~]# getfacl /data/main  
getfacl: Removing leading '/' from absolute path names  
# file: data/main  
# owner: root  
# group: main  
# flags: -s-  
user::rwx  
group::rwx  
group:third:r-x  
mask::rwx  
other:---  
default:user::rwx  
default:group::rwx  
default:group:third:rwx  
default:mask::rwx  
default:other:---  
  
[root@pvbarabash ~]# setfacl -m g:third:r /data/main  
[root@pvbarabash ~]# getfacl /data/main  
getfacl: Removing leading '/' from absolute path names  
# file: data/main  
# owner: root  
# group: main  
# flags: -s-  
user::rwx  
group::rwx  
group:third:r--  
mask::rwx  
other:---  
default:user::rwx  
default:group::rwx  
default:group:third:rwx
```

Рис. 3.5: Предоставление членам группы права доступа на чтение для всех существующих файлов в текущем каталоге

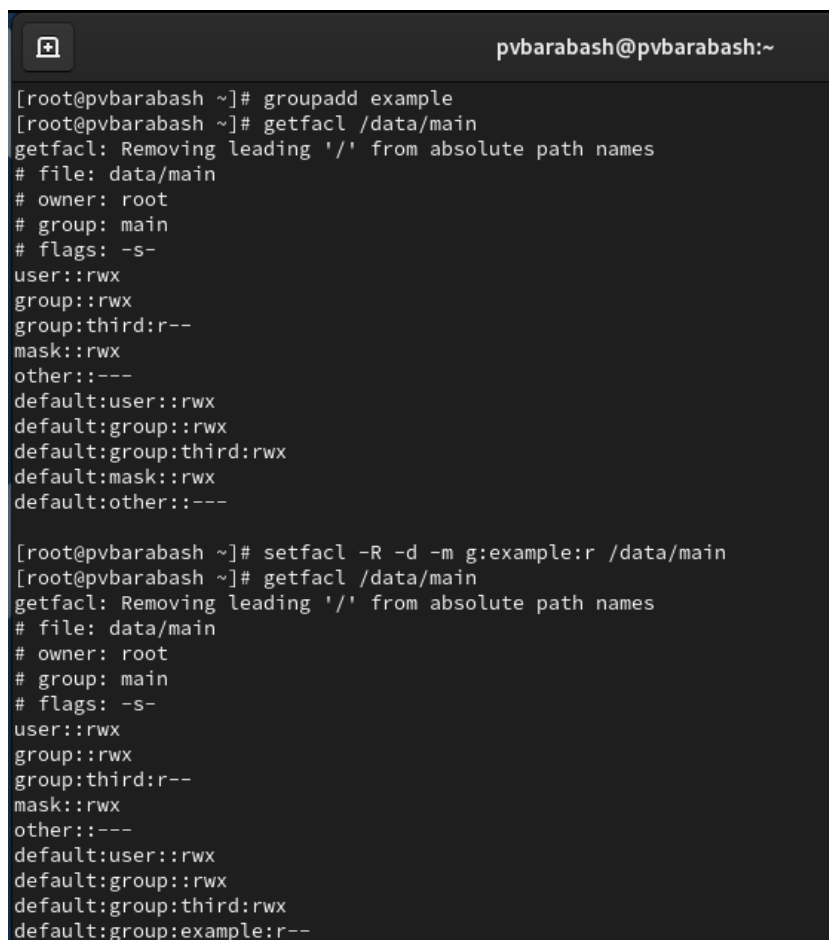
8. Что нужно сделать для гарантии того, что члены группы получают разрешения на чтение для всех файлов в текущем каталоге и во всех его подкаталогах, а также для всех файлов, которые будут созданы в этом каталоге в будущем? Приведите пример.

Для этого нужно установить стандартные ACL. Чтобы применить разрешения ACL на каталог и все его подкаталоги и файлы, нужно использовать опцию -R, что означает “рекурсивно”. Затем нужно использовать опцию -d, чтобы для новых файло ACL устанавливалась по умолчанию таким, как для группы. Затем нужно использовать опцию -m, чтобы изменить текущий ACL для файла. Если собрать



всё вместе, то получится следующая команда: `setfacl -R -d -m g:groupname:r`.

Пример: я создала группу `example`, затем добавила дефолтные права на чтение для этой группы для каталога `/data/main` и проверила, что новые разрешения вступили в силу (рис. [3.6]).



```
pvbarabash@pvbarabash:~  
[root@pvbarabash ~]# groupadd example  
[root@pvbarabash ~]# getfacl /data/main  
getfacl: Removing leading '/' from absolute path names  
# file: data/main  
# owner: root  
# group: main  
# flags: -s-  
user::rwx  
group::rwx  
group:third:r--  
mask::rwx  
other:---  
default:user::rwx  
default:group::rwx  
default:group:third:rwx  
default:mask::rwx  
default:other:---  
  
[root@pvbarabash ~]# setfacl -R -d -m g:example:r /data/main  
[root@pvbarabash ~]# getfacl /data/main  
getfacl: Removing leading '/' from absolute path names  
# file: data/main  
# owner: root  
# group: main  
# flags: -s-  
user::rwx  
group::rwx  
group:third:r--  
mask::rwx  
other:---  
default:user::rwx  
default:group::rwx  
default:group:third:rwx  
default:group:example:r--
```

Рис. 3.6: Действия для гарантии того, что члены группы получают 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

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

Нужно установить `umask` на `007`.

Пример: я перешла в каталог `/data/main` и установила `umask` на `007`, затем

создала новый файл aaa и проверила, что для “других” пользователей нет прав на новый созданный файл (рис. [3.7]).

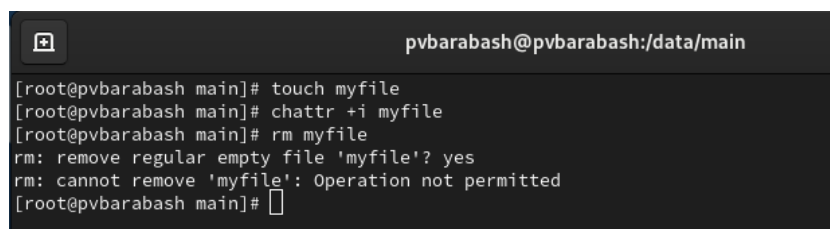
```
pvbarabash@pvbarabash:/data/main
[root@pvbarabash ~]# cd /data/main
[root@pvbarabash main]# ls -l
total 4
-rwxr-xr-x. 1 alice alice 0 Sep 16 08:01 alice3
-rw-r--r--. 1 alice main 0 Sep 16 08:01 alice4
-rw-r--r--. 1 bob bob 0 Sep 16 07:59 bob1
-rw-r--r--. 1 bob bob 0 Sep 16 07:59 bob2
-rw-r--r--. 1 bob bob 0 Sep 16 07:51 emptyfile
-rw-r--r--. 1 root main 0 Sep 16 08:06 newfile1
-rw-rw----+ 1 root main 13 Sep 16 08:14 newfile2
[root@pvbarabash main]# umask 007
[root@pvbarabash main]# ls -l
total 4
-rwxr-xr-x. 1 alice alice 0 Sep 16 08:01 alice3
-rw-r--r--. 1 alice main 0 Sep 16 08:01 alice4
-rw-r--r--. 1 bob bob 0 Sep 16 07:59 bob1
-rw-r--r--. 1 bob bob 0 Sep 16 07:59 bob2
-rw-r--r--. 1 bob bob 0 Sep 16 07:51 emptyfile
-rw-r--r--. 1 root main 0 Sep 16 08:06 newfile1
-rw-rw----+ 1 root main 13 Sep 16 08:14 newfile2
[root@pvbarabash main]# touch aaa
[root@pvbarabash main]# ls -l
total 4
-rw-rw----+ 1 root main 0 Sep 16 10:51 aaa
-rwxr-xr-x. 1 alice alice 0 Sep 16 08:01 alice3
-rw-r--r--. 1 alice main 0 Sep 16 08:01 alice4
-rw-r--r--. 1 bob bob 0 Sep 16 07:59 bob1
-rw-r--r--. 1 bob bob 0 Sep 16 07:59 bob2
-rw-r--r--. 1 bob bob 0 Sep 16 07:51 emptyfile
-rw-r--r--. 1 root main 0 Sep 16 08:06 newfile1
-rw-rw----+ 1 root main 13 Sep 16 08:14 newfile2
[root@pvbarabash main]#
```

Рис. 3.7: Значение umask, чтобы «другие» пользователи не получали какие-либо разрешения на новые файлы

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

Чтобы никто не смог удалить файл myfile случайно, нужно установить атрибут “immutable” с помощью команды chattr.

Пример: я создала файл myfile, затем сделала команду chattr +i myfile и попробовала удалить этот файл. Ничего не вышло (рис. [3.8]).



```
pvbarabash@pvbarabash:/data/main
[root@pvbarabash main]# touch myfile
[root@pvbarabash main]# chattr +i myfile
[root@pvbarabash main]# rm myfile
rm: remove regular empty file 'myfile'? yes
rm: cannot remove 'myfile': Operation not permitted
[root@pvbarabash main]#
```

Рис. 3.8: Блокировка для удаления файла

## ***4 Выводы***

Я получила навыки настройки базовых и специальных прав доступа для групп пользователей в операционной системе типа Linux.