

Отчёт по лабораторной работе №3

Настройка прав доступа

Саенко Ангелина Андреевна

Содержание

1	Цель работы	5
2	Ход выполнения работы	6
2.1	Управление базовыми разрешениями	6
2.2	Управление специальными разрешениями	7
2.3	Управление расширенными разрешениями с использованием спис- ков ACL	10
3	Контрольные вопросы	13
4	Заключение	16

Список иллюстраций

2.1	Создание новых каталогов	6
2.2	Установка разрешений	7
2.3	Вход под пользователем bob и создание файла	7
2.4	Создание файлов под пользователем alice	8
2.5	Удаление файлов, принадлежащие пользователю alice и создание новых	8
2.6	Установка бит идентификатора	9
2.7	Попытка удалить файлы, принадлежащие другому пользователю	9
2.8	Добавление пользователей в группы и проверка членства	10
2.9	Создание файлов и проверка прав доступа	10
2.10	Установка ACL	11
2.11	Проверка назначенных полномочий	11
2.12	Проверка операций с файлами	12

Список таблиц

1 Цель работы

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

2 Ход выполнения работы

2.1 Управление базовыми разрешениями

Для создания структуры каталогов с разными разрешениями доступа для разных групп пользователей я открыла терминал с учётной записью root и в корневом каталоге создала каталоги /data/main и /data/third. Владелец данных каталогов является root. Это можно увидеть на скриншоте ниже.



```
root@aasaenko:~ -- -bash
aasaenko@aasaenko:~$ firefox &
[1] 3314
aasaenko@aasaenko:~$ su -
Пароль:
Последний вход в систему: Ср сен 10 19:55:59 MSK 2025 на pts/0
root@aasaenko:~# mkdir -p /data/main /data/third
root@aasaenko:~# ls -Al /data
итого 0
drwxr-xr-x. 2 root root 6 сен 12 12:22 main
drwxr-xr-x. 2 root root 6 сен 12 12:22 third
```

Рис. 2.1: Создание новых каталогов

Перед установкой разрешений я изменила владельцев этих каталогов на main и third. После чего сделала проверку и установила сами разрешения, позволяющие владельцам каталогов записывать файлы в эти каталоги и запрещающие доступ к содержимому каталогов всем другим пользователям и группам. После сделала проверку. Это можно увидеть на скриншоте.

```
drwxr-xr-x. 2 root root 6 сен 12 12:22 third
root@aasaenko:~# chgrp main /data/main
root@aasaenko:~# chgrp main /data/third
root@aasaenko:~# chgrp main /data/main
root@aasaenko:~# chgrp third /data/third
root@aasaenko:~# ls -Al /data
итого 0
drwxr-xr-x. 2 root main 6 сен 12 12:22 main
drwxr-xr-x. 2 root third 6 сен 12 12:22 third
root@aasaenko:~# chmod 770 /data/main
root@aasaenko:~# chmod 770 /data/third
root@aasaenko:~# ^C
root@aasaenko:~# ls -Al /data
итого 0
drwxrwx---. 2 root main 6 сен 12 12:22 main
drwxrwx---. 2 root third 6 сен 12 12:22 third
root@aasaenko:~#
```

Рис. 2.2: Установка разрешений

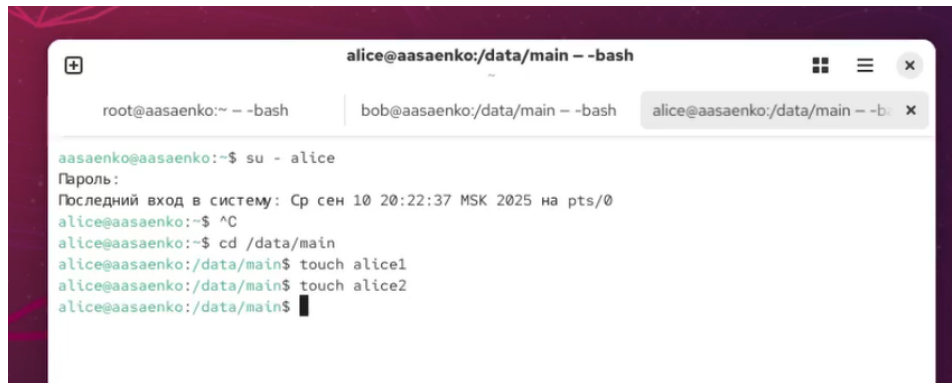
Затем я перешла под учётную запись пользователя **bob**, перешла в каталог `data/main` и создала файл `emptyfile` в этом каталоге. Имя владельца и группы совпадают и равны **bob**. Попробовав перейти под пользователем **bob** в каталог `/data/third`, я получила отказ в доступе. Это произошло, так как **bob** входит в другую группу. Это можно увидеть ниже.

```
bob@aasaenko:/data/main -- -bash
root@aasaenko:~ -- -bash
aasaenko@aasaenko:~$ su - bob
Пароль:
bob@aasaenko:~$ cd /data/main
bob@aasaenko:/data/main$ touch emptyfile
bob@aasaenko:/data/main$ ls -Al
итого 0
-rw-r--r--. 1 bob bob 0 сен 12 12:24 emptyfile
bob@aasaenko:/data/main$ ^C
bob@aasaenko:/data/main$ cd /data/third
-bash: cd: /data/third: Отказано в доступе
bob@aasaenko:/data/main$
```

Рис. 2.3: Вход под пользователем bob и создание файла

2.2 Управление специальными разрешениями

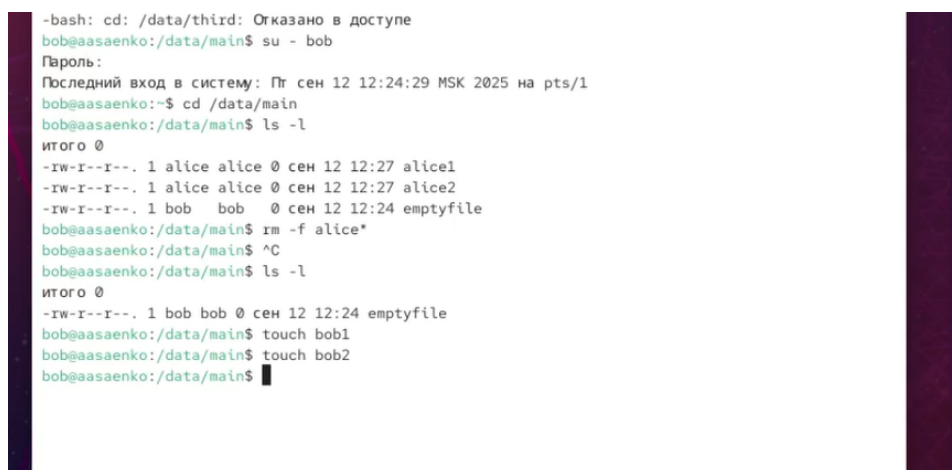
Открою новый терминал под пользователем **alice** и в каталоге `/data/main` создам два файла `alice1` и `alice2`. Выполнение можно увидеть ниже.



```
alice@aasaenko:/data/main -- -bash
root@aasaenko:~ -- -bash  bob@aasaenko:/data/main -- -bash  alice@aasaenko:/data/main -- -b
aasaenko@aasaenko:~$ su - alice
Пароль:
Последний вход в систему: Ср сен 10 20:22:37 MSK 2025 на pts/0
alice@aasaenko:~$ ^C
alice@aasaenko:~$ cd /data/main
alice@aasaenko:/data/main$ touch alice1
alice@aasaenko:/data/main$ touch alice2
alice@aasaenko:/data/main$
```

Рис. 2.4: Создание файлов под пользователем **alice**

В другом терминале перейду под учётную запись пользователя **bob**. В каталоге `/data/main` посмотрю файлы, созданные пользователем **alice**. После чего удалю файлы, принадлежащие пользователю **alice** и создам новые файлы, которые принадлежат пользователю **bob**. Это можете увидеть ниже на скриншоте.



```
-bash: cd: /data/third: Отказано в доступе
bob@aasaenko:/data/main$ su - bob
Пароль:
Последний вход в систему: Пт сен 12 12:24:29 MSK 2025 на pts/1
bob@aasaenko:~$ cd /data/main
bob@aasaenko:/data/main$ ls -l
итого 0
-rw-r--r--. 1 alice alice 0 сен 12 12:27 alice1
-rw-r--r--. 1 alice alice 0 сен 12 12:27 alice2
-rw-r--r--. 1 bob  bob  0 сен 12 12:24 emptyfile
bob@aasaenko:/data/main$ rm -f alice*
bob@aasaenko:/data/main$ ^C
bob@aasaenko:/data/main$ ls -l
итого 0
-rw-r--r--. 1 bob bob 0 сен 12 12:24 emptyfile
bob@aasaenko:/data/main$ touch bob1
bob@aasaenko:/data/main$ touch bob2
bob@aasaenko:/data/main$
```

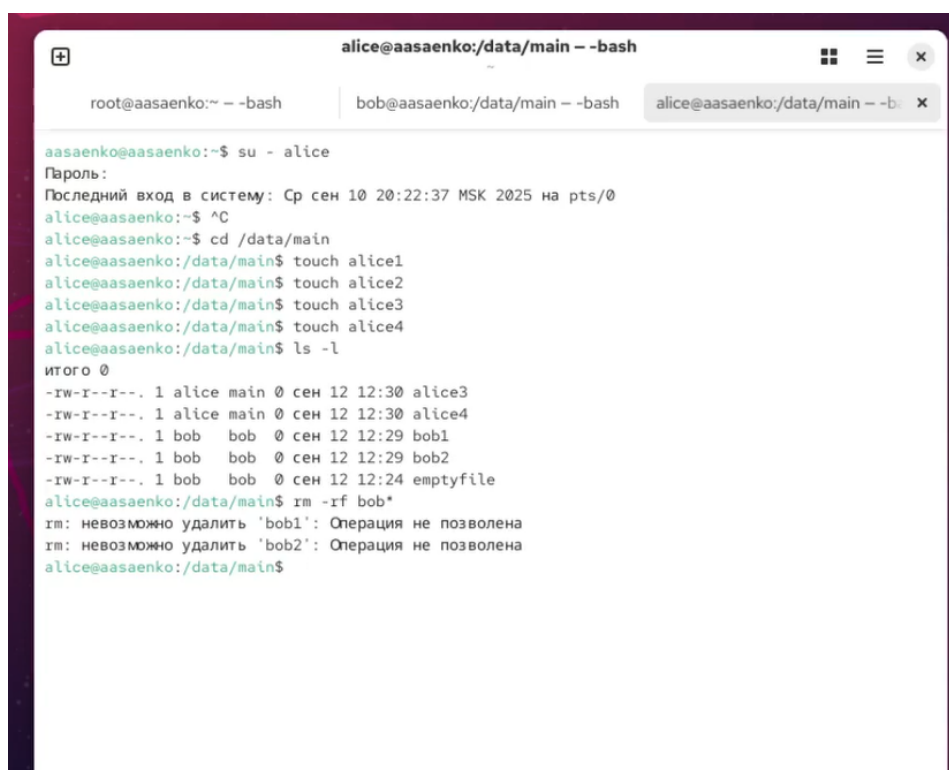
Рис. 2.5: Удаление файлов, принадлежащие пользователю **alice** и создание новых

После этого под пользователем **root** для каталога `/data/main` установлю бит идентификатор группы, а также `sticky`-бит для разделяемого (общего) каталога группы.


```
drwxrwx---. 2 root third 6 сен 12 12:22 third
root@aasaenko:~# chmod g+s,o+t /data/main
root@aasaenko:~#
```

Рис. 2.6: Установка бит идентификатора

Далее в терминале под пользователем **alice** создам в каталоге /data/main файлы alice3 и alice4. После чего в терминале под пользователем **alice** попробую удалить файлы, принадлежащие пользователю **bob**. Замечу, что sticky-bit предотвратит удаление этих файлов пользователем **alice**, поскольку этот пользователь не является владельцем этих файлов. Это вы можете увидеть на скриншоте.



```
alice@aasaenko:/data/main ~ -bash
root@aasaenko:~ - -bash
bob@aasaenko:/data/main ~ -bash
alice@aasaenko:/data/main ~ -b

aasaenko@aasaenko:~$ su - alice
Пароль:
Последний вход в систему: Ср сен 10 20:22:37 MSK 2025 на pts/0
alice@aasaenko:~$ ^C
alice@aasaenko:~$ cd /data/main
alice@aasaenko:/data/main$ touch alice1
alice@aasaenko:/data/main$ touch alice2
alice@aasaenko:/data/main$ touch alice3
alice@aasaenko:/data/main$ touch alice4
alice@aasaenko:/data/main$ ls -l
итого 0
-rw-r--r--. 1 alice main 0 сен 12 12:30 alice3
-rw-r--r--. 1 alice main 0 сен 12 12:30 alice4
-rw-r--r--. 1 bob   bob   0 сен 12 12:29 bob1
-rw-r--r--. 1 bob   bob   0 сен 12 12:29 bob2
-rw-r--r--. 1 bob   bob   0 сен 12 12:24 emptyfile
alice@aasaenko:/data/main$ rm -rf bob*
rm: невозможно удалить 'bob1': Операция не позволена
rm: невозможно удалить 'bob2': Операция не позволена
alice@aasaenko:/data/main$
```

Рис. 2.7: Попытка удалить файлы, принадлежащие другому пользователю

2.3 Управление расширенными разрешениями с использованием списков ACL

Под пользователем root устанавливаю 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. Далее проверю правильность установки разрешений.

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

Рис. 2.8: Добавление пользователей в группы и проверка членства

Создам новый файл с именем newfile1 в каталоге /data/main. Также проверю текущие значения полномочий. У данного файла такие права доступа : пользователь может как читать, так и писать, группа может только читать и другие могут только читать данный файл. Аналогичные действия выполню и для каталога /data/third. Права доступа в нём будут такие же, как и у первого каталога. Это можно увидеть ниже.

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

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

Рис. 2.9: Создание файлов и проверка прав доступа

Далее я устанавливаю ACL по умолчанию для каталога /data/main, после чего добавлю ACL по умолчанию для каталога /data/third.

```
root@aasaenko:~# setfacl -m d:g:third:rxw /data/main
root@aasaenko:~# setfacl -m d:g:main:rxw /data/third
```

Рис. 2.10: Установка ACL

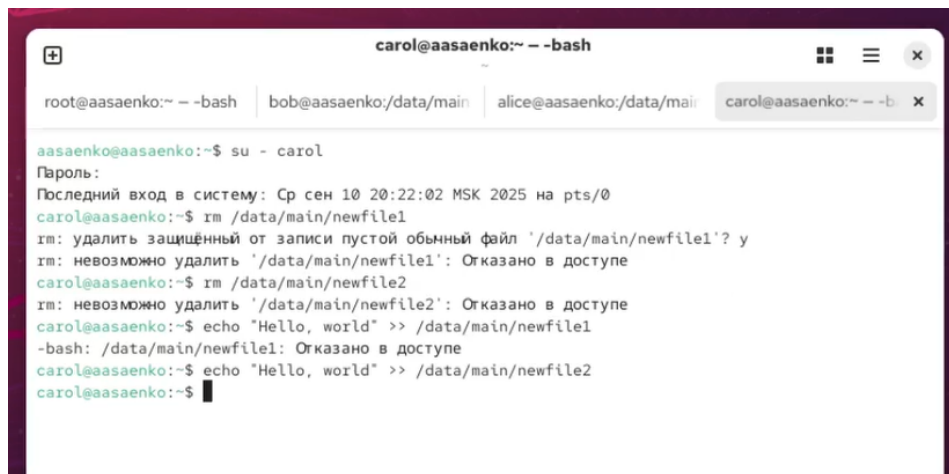
Затем я проверю, что всё работает, добавив новый файл и проверю текущие значения полномочий. После чего выполню аналогичные действия для второго каталога.

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

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

Рис. 2.11: Проверка назначенных полномочий

Войду под учётной записью пользователя **carol**. Операции с файлами ниже мне не удастся выполнить, так как пользователь принадлежит другой группе. При этом запись в файл `newfile2` я смогу сделать. Это можно увидеть ниже на скриншоте.



The image shows a terminal window with a dark purple border. At the top, the title bar reads 'carol@aasaenko:~ -- bash'. Below the title bar, there are four tabs: 'root@aasaenko:~ -- bash', 'bob@aasaenko:/data/main', 'alice@aasaenko:/data/mai', and 'carol@aasaenko:~ -- bash'. The main terminal area shows the following commands and output:

```
aasaenko@aasaenko:~$ su - carol
Пароль:
Последний вход в систему: Ср сен 10 20:22:02 MSK 2025 на pts/0
carol@aasaenko:~$ rm /data/main/newfile1
rm: удалить защищенный от записи пустой обычный файл '/data/main/newfile1'? y
rm: невозможно удалить '/data/main/newfile1': Отказано в доступе
carol@aasaenko:~$ rm /data/main/newfile2
rm: невозможно удалить '/data/main/newfile2': Отказано в доступе
carol@aasaenko:~$ echo "Hello, world" >> /data/main/newfile1
-bash: /data/main/newfile1: Отказано в доступе
carol@aasaenko:~$ echo "Hello, world" >> /data/main/newfile2
carol@aasaenko:~$
```

Рис. 2.12: Проверка операций с файлами

3 Контрольные вопросы

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

Для этого можно использовать несколько команд:

- `chown :группа файл` - устанавливает владельца группы для файла;
- `chown пользователь:группа файл` - одновременно устанавливает владельца и группу;
- `chown :группа каталог/*` - устанавливает группу для всех файлов в каталоге.

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

- `find / -user имя_пользователя` - поиск файлов по владельцу;
- `find / -uid UID_пользователя` - поиск по UID пользователя.

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?

- `chmod -R ug+rwX /data` - устанавливает права `rwX` для пользователя и группы;

- `chmod -R o-rwx /data` - убирает все права для других.
4. **Какая команда позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым?**
- `chmod +x файл` - добавляет право на выполнение для всех;
 - `chmod u+x файл` - добавляет право на выполнение только для владельца.
5. **Какая команда позволяет убедиться, что групповые разрешения для всех новых файлов, создаваемых в каталоге, будут присвоены владельцу группы этого каталога?**
- `chmod g+s каталог` - устанавливает бит `setgid` для каталога;
 - `chmod 2775 каталог` - устанавливает права `rwxrwxr-x` с `setgid` битом.
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?**
- `chmod +t каталог` - устанавливает `sticky bit` на каталог;
 - `chmod 1777 каталог` - устанавливает права `gwx` для всех со `sticky bit`
7. **Какая команда добавляет ACL, который предоставляет членам группы права доступа на чтение для всех существующих файлов в текущем каталоге?**
- `setfacl -R -m g:группа:r *` - рекурсивно добавляет права чтения для группы;

- `setfacl -m g:группа:r файл` - добавляет права чтения для конкретного файла.

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?

- `setfacl -R -m g:группа:r каталог` - рекурсивно устанавливает права чтения;
- `setfacl -d -m g:группа:r каталог` - устанавливает права по умолчанию для новых файлов.

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

- `umask 007` - права только для владельца и группы (`rw-rw-r--`);
- `umask 077` - права только для владельца (`rw-r--r--`).

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

Напрямую вносить изменения в `/etc/group` не рекомендуется, так как это может привести к ошибкам. Корректнее использовать утилиты:

- `chattr +i myfile` - устанавливает атрибут “immutable” (неизменяемый);
- `chattr +a myfile` - устанавливает атрибут “append-only” (только дополнение).

4 Заключение

В ходе выполнения лабораторной работы я приобрела практические навыки настройки базовых, специальных и расширенных прав доступа для групп пользователей в операционной системе Linux.

Были выполнены следующие действия:

- изучение команд управления правами доступа: `chgrp`, `chmod`, `getfacl`, `setfacl`;
- создание иерархии каталогов с различными уровнями доступа для разных групп пользователей;
- изменение системных параметров для автоматического формирования домашнего каталога;
- настройка базовых разрешений с использованием традиционной системы прав Linux;
- применение специальных разрешений (`setgid` и `sticky bit`) для обеспечения безопасного обмена файлами;
- работа с расширенными списками контроля доступа (ACL) для гибкого управления правами;
- установка наследуемых ACL-правил для автоматического применения прав к новым файлам.

В процессе работы я закрепила понимание механизмов управления доступом в Linux, научилась настраивать права для совместной работы пользователей в группах, а также обеспечивать безопасность данных с помощью специальных атрибутов файловой системы. Полученный опыт показал важность корректной настройки прав доступа для многопользовательских сред.