

Лабораторная работа №2

Дисциплина: Информационная безопасность

Губина Ольга Вячеславовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	28
	Список литературы	29

Список иллюстраций

4.1	Создание учетной записи guest	9
4.2	Вход в систему под новым пользователем guest	10
4.3	Домашняя директория пользователя guest	10
4.4	Информация о пользователе guest	10
4.5	Файл /etc/passwd	12
4.6	Директории в /home/	12
4.7	Расширенные атрибуты поддиректории	13
4.8	Атрибуты директорий	13
4.9	Атрибуты директорий	14
4.10	Изменение прав доступа на dir1	14
4.11	Попытки взаимодействия с каталогом dir1	15
4.12	d(700), -(100)	16
4.13	d(700), -(200)	16
4.14	d(600), -(600)	17
4.15	d(500), -(100)	17
4.16	d(400), -(400)	18
4.17	d(300), -(600)	18
4.18	d(200), -(700)	19
4.19	d(100), -(100)	19

Список таблиц

4.1	Установленные права и разрешенные действия	20
4.2	Минимальные права для совершения операций	27

1 Цель работы

Получение практических навыков работы в консоли с атрибутами файлов, закрепление теоретических основ дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux.

2 Задание

- Произвести работу в консоли с атрибутами от имени пользователя *guest*;
- Составить опытным путем таблицы “Установленные права и разрешенные действия” и “Минимальные права для совершения операций”.

3 Теоретическое введение

В данной лабораторной работе нам предстоит поработать с правами доступа файлов и директорий. **Права доступа** определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами.

Есть 3 вида разрешений. Они определяют права пользователя на 3 действия: чтение, запись и выполнение. В Linux эти действия обозначаются вот так:

- **r** — read (чтение) — право просматривать содержимое файла;
- **w** — write (запись) — право изменять содержимое файла;
- **x** — execute (выполнение) — право запускать файл, если это программа или скрипт.

У каждого файла есть 3 группы пользователей, для которых можно устанавливать права доступа.

- **owner** (владелец) — отдельный человек, который владеет файлом. Обычно это тот, кто создал файл, но владельцем можно сделать и кого-то другого.
- **group** (группа) — пользователи с общими заданными правами.
- **others** (другие) — все остальные пользователи, не относящиеся к группе и не являющиеся владельцами.[1]

Чтобы увидеть текущие назначения владельца, вы можете использовать команду `ls -l`. Эта команда показывает пользователя и группу-владельца.

С помощью команды `ls` вы можете отобразить владельца файлов в данном каталоге. Иногда может оказаться полезным получить список всех файлов в системе, в которых в качестве владельца указан данный пользователь или группа. Для этого вы можете использовать `find`. Аргумент `find -user` может быть использован для этой цели.

Чтобы применить соответствующие разрешения, первое, что нужно учитывать, это владение. Для этого есть команда `chown`.^[2]

Для того, чтобы позволить обычным пользователям выполнять программы от имени суперпользователя без знания его пароля была придумана такая вещь, как SUID и SGID биты. Рассмотрим эти полномочия подробнее.

- **SUID** - если этот бит установлен, то при выполнении программы, `id` пользователя, от которого она запущена заменяется на `id` владельца файла. Фактически, это позволяет обычным пользователям запускать программы от имени суперпользователя;
- **SGID** - этот флаг работает аналогичным образом, только разница в том, что пользователь считается членом группы, с которой связан файл, а не групп, к которым он действительно принадлежит. Если SGID флаг установлен на каталог, все файлы, созданные в нем, будут связаны с группой каталога, а не пользователя. Такое поведение используется для организации общих папок;
- **Sticky-bit** - этот бит тоже используется для создания общих папок. Если он установлен, то пользователи могут только создавать, читать и выполнять файлы, но не могут удалять файлы, принадлежащие другим пользователям.^[3]

4 Выполнение лабораторной работы

1. В установленной при выполнении предыдущей лабораторной работы операционной системе создадим учётную запись пользователя guest (используя учётную запись администратора) (рис. 4.1): `useradd guest`.

```
[ovgubina@ovgubina ~]$ sudo -i
[sudo] password for ovgubina:
[root@ovgubina ~]# useradd guest
[root@ovgubina ~]# passwd guest
Changing password for user guest.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
[root@ovgubina ~]# passwd guest
Changing password for user guest.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ovgubina ~]#
```

Рис. 4.1: Создание учетной записи guest

2. Зададим пароль для пользователя guest (используя учётную запись администратора) (рис. 4.1): `passwd guest`.
3. Перезапустила машину и вошла в систему от имени пользователя guest (рис. 4.2).

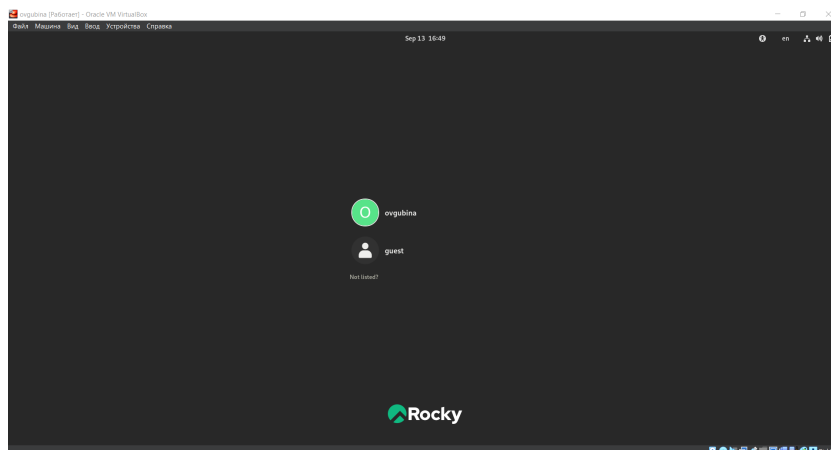


Рис. 4.2: Вход в систему под новым пользователем guest

4. Определим директорию, в которой находимся, командой `pwd` (рис. 4.3). Сравним её с приглашением командной строки. В командной строке видим символ `~`, что свидетельствует о том, что мы находимся в домашней директории. Определим, является ли она действительно домашней директорией, введя команду `cd`, которая позволяет перейти в домашнюю директорию. Видим, что ничего не меняется. **Мы находимся в своей домашней директории.**

```
[guest@ovgubina ~]$ pwd
/home/guest
[guest@ovgubina ~]$ cd
[guest@ovgubina ~]$ cd /home/guest
[guest@ovgubina ~]$
```

Рис. 4.3: Домашняя директория пользователя guest

5. Уточним имя пользователя командой `whoami` (рис. 4.4).

```
[guest@ovgubina ~]$ whoami
guest
[guest@ovgubina ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@ovgubina ~]$ groups
guest
[guest@ovgubina ~]$
```

Рис. 4.4: Информация о пользователе guest

Видим, что имя нашего пользователя - **guest**.

6. Уточним имя пользователя, его группу, а также группы, куда входит пользователь, командой `id` (рис. 4.4).

Видим следующие данные: `uid = 1001(guest), gid = 1001(guest), groups = 1001(guest)`.

Сравним вывод `id` с выводом команды `groups` (рис. 4.4).

Данная команда показывает группы текущего пользователя, аналогично выводу команды `id`, группы пользователя `guest` - это группа `guest`.

7. Сравните полученную информацию об имени пользователя с данными, выводимыми в приглашении командной строки.

Команда `whoami` дала нам понять, что имя пользователя - `guest`. В начале приглашения командной строки как раз указано имя нашего пользователя - все сходится.

8. Просмотрим файл `/etc/passwd` (рис. 4.5):

```
cat /etc/passwd
```

```
[guest@ovgubina ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
dbus:x:81:81:system message bus:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
sssd:x:997:993:User for sssd:/sbin/nologin
pipewire:x:996:992:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
libstoragemgmt:x:990:990:daemon account for libstoragemgmt:/usr/sbin/nologin
systemd-oom:x:989:989:systemd Userspace OOM Killer:/usr/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
geoclue:x:988:987:User for geoclue:/var/lib/geoclue:/sbin/nologin
cockpit-ws:x:987:986:User for cockpit web service:/nonexisting:/sbin/nologin
cockpit-wsInstance:x:986:985:User for cockpit-ws instances:/nonexisting:/sbin/nologin
flatpak:x:985:984:User for flatpak system helper:/sbin/nologin
colord:x:984:983:User for colord:/var/lib/colord:/sbin/nologin
clevis:x:983:982:clevis Decryption Framework unprivileged user:/var/cache/clevis:/usr/sbin/nologin
setroubleshoot:x:982:981:SELinux troubleshoot server:/var/lib/setroubleshoot:/sbin/nologin
gdm:x:42:42:/var/lib/gdm:/sbin/nologin
pesign:x:981:980:Group for the pesign signing daemon:/run/pesign:/sbin/nologin
gnome-initial-setup:x:980:979:/run/gnome-initial-setup:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
chrony:x:979:978:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:978:977:dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
tcpdump:x:72:72:/sbin/nologin
ovgubina:x:1000:1000:ovgubina:/home/ovgubina:/bin/bash
vboxadd:x:977:1:/var/run/vboxadd:/bin/false
guest:x:1001:1001:/home/guest:/bin/bash
[guest@ovgubina ~]$
```

Рис. 4.5: Файл /etc/passwd

Найдем в нем последнюю запись - запись о текущем пользователе (выделено на рис. 4.5). Данная строка показывает, что uid = 1001, gid = 1001, что соответствует результатам предыдущих команд.

9. Определим существующие в системе директории командой `ls -l /home/` (рис. 4.6).

```
[guest@ovgubina ~]$ ls -l /home/
total 8
drwx-----. 14 guest      guest    4096 Sep 13 16:49 guest
drwx-----. 14 ovgubina  ovgubina 4096 Sep 13 16:47 ovgubina
[guest@ovgubina ~]$
```

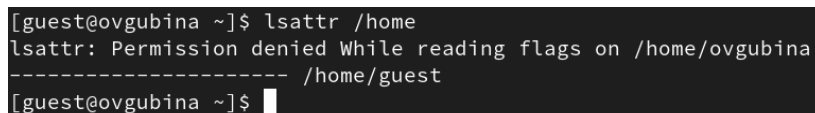
Рис. 4.6: Директории в /home/

Нам удалось получить список поддиректорий директории /home. Директории имеют следующие права: владельцы данных директорий обладают полными

правами (на чтение, запись и выполнение), в то время как группы и другие пользователи обладают нулевыми правами.

10. Проверим, какие расширенные атрибуты установлены на поддиректориях, находящихся в директории /home, командой (рис. 4.7):

```
lsattr /home
```



```
[guest@ovgubina ~]$ lsattr /home
lsattr: Permission denied While reading flags on /home/ovgubina
----- /home/guest
[guest@ovgubina ~]$
```

Рис. 4.7: Расширенные атрибуты поддиректории

Нам удалось просмотреть расширенные атрибуты своей домашней директории - оказалось, что никаких расширенных атрибутов нет. В то же время нам отказывают в доступе к просмотру расширенных атрибутов директории другого пользователя (рис. [fig?];007).

11. Создадим в домашней директории поддиректорию dir1 командой `mkdir dir1`.

Определим командами `ls -l` (рис. 4.8) и `lsattr` (рис. 4.9), какие права доступа и расширенные атрибуты были выставлены на директорию dir1.



```
[guest@ovgubina ~]$ ls -l
total 0
drwxr-xr-x. 2 guest guest  6 Sep 13 16:49 Desktop
drwxr-xr-x. 2 guest guest  6 Sep 13 17:01 dir1
drwxr-xr-x. 2 guest guest  6 Sep 13 16:49 Documents
drwxr-xr-x. 2 guest guest  6 Sep 13 16:49 Downloads
drwxr-xr-x. 2 guest guest  6 Sep 13 16:49 Music
drwxr-xr-x. 2 guest guest 147 Sep 13 17:00 Pictures
drwxr-xr-x. 2 guest guest  6 Sep 13 16:49 Public
drwxr-xr-x. 2 guest guest  6 Sep 13 16:49 Templates
drwxr-xr-x. 2 guest guest  6 Sep 13 16:49 Videos
```

Рис. 4.8: Атрибуты директорий

```
[guest@ovgubina ~]$ lsattr
----- ./Desktop
----- ./Downloads
----- ./Templates
----- ./Public
----- ./Documents
----- ./Music
----- ./Pictures
----- ./Videos
----- ./dir1
[guest@ovgubina ~]$
```

Рис. 4.9: Атрибуты директорий

Видим, что в директории `dir1` ее владелец обладает полными правами (`rwX`), а группы пользователей и другие пользователи имеют права только на чтение и выполнение (`r-x`). Расширенных атрибутов у каталога нет.

12. Снимим с директории `dir1` все атрибуты командой `chmod 000 dir1` и проверим правильность выполнения с помощью команды `ls -l` (рис. 4.10).

```
[guest@ovgubina ~]$ chmod 000 dir1
[guest@ovgubina ~]$ ls -l
total 4
drwxr-xr-x. 2 guest guest 6 Sep 13 16:49 Desktop
d----- . 2 guest guest 6 Sep 13 17:01 dir1
drwxr-xr-x. 2 guest guest 6 Sep 13 16:49 Documents
drwxr-xr-x. 2 guest guest 6 Sep 13 16:49 Downloads
drwxr-xr-x. 2 guest guest 6 Sep 13 16:49 Music
drwxr-xr-x. 2 guest guest 4096 Sep 13 17:06 Pictures
drwxr-xr-x. 2 guest guest 6 Sep 13 16:49 Public
drwxr-xr-x. 2 guest guest 6 Sep 13 16:49 Templates
drwxr-xr-x. 2 guest guest 6 Sep 13 16:49 Videos
[guest@ovgubina ~]$
```

Рис. 4.10: Изменение прав доступа на `dir1`

Видим, что теперь на директорию `dir1` нет никаких прав.

13. Попытаемся создать в директории `dir1` файл `file1` командой `echo "test" > /home/guest/dir1/file1`.

```
[guest@ovgubina ~]$ echo "test" > /home/guest/dir1/file1
bash: /home/guest/dir1/file1: Permission denied
[guest@ovgubina ~]$ ls -l /home/guest/dir1
ls: cannot open directory '/home/guest/dir1': Permission denied
[guest@ovgubina ~]$
```

Рис. 4.11: Попытки взаимодействия с каталогом dir1

Мы получили отказ в выполнении операции по созданию файла, потому что мы не обладаем правами на это, поскольку в предыдущих шагах мы обнулили все права данного каталога.

Из-за нулевых прав мы также не можем посмотреть содержимое каталога. Если пытаться взаимодействовать с файлом система ответит, что такого файла нет, значит файл не создался, что логично, поскольку у нас нет прав на создание файлов в данной директории.

14. Заполните таблицу «Установленные права и разрешённые действия», выполняя действия от имени владельца директории (файлов), определив опытным путём, какие операции разрешены, а какие нет. Если операция разрешена, занесите в таблицу знак «+», если не разрешена, знак «-».

Для определения опытным путем будем использовать следующие действия в соответствии со столбцами таблицы:

- `cd dir1` - смена директории;
- `touch <новый_файл>` - создание файла;
- `rm <новый_файл>` - удаление файла;
- `ls -l (dir1)` - просмотр файлов в директории;
- `echo "test" > <файл_с_установленными_правами>` - запись в файл;
- `cat <файл_с_установленными_правами>` - чтение файла;
- `mv <файл_с_установленными_правами> <переименование>` - переименование файла;
- `chattr <атрибуты> <файл_с_установленными_правами>` смена атрибутов файла.

В качестве примера приведу осуществление проверки для прав доступа drwx----- (700), ---x-----(100) (рис. 4.12), drwx----- (700), ---x----- (200) (рис. 4.13), drw----- (600), -rw----- (600) (рис. 4.14), dr-x----- (500), -r-x----- (100) (рис. 4.15), dr----- (400), -r----- (400) (рис. 4.16), d-wx----- (300), --wx----- (600) (рис. 4.17), d-w----- (200), --w----- (700) (рис. 4.18), d--x----- (100), ---x----- (100) (рис. 4.19).

```
[guest@ovgubina ~]$ chmod 700 dir1
[guest@ovgubina ~]$ cd dir1
[guest@ovgubina dir1]$ cd
[guest@ovgubina ~]$ chmod 100 dir1/file
[guest@ovgubina ~]$ cd dir1
[guest@ovgubina dir1]$ ls
ffff  file
[guest@ovgubina dir1]$ touch file700100
[guest@ovgubina dir1]$ rm file700100
[guest@ovgubina dir1]$ echo "test" > file
bash: file: Permission denied
[guest@ovgubina dir1]$ cat file
cat: file: Permission denied
[guest@ovgubina dir1]$ mv file fiil
[guest@ovgubina dir1]$ mv fiil file
[guest@ovgubina dir1]$ chattr +i file
chattr: Permission denied while reading flags on file
[guest@ovgubina dir1]$
```

Рис. 4.12: d(700), -(100)

```
[guest@ovgubina ~]$ chmod 700 dir1
[guest@ovgubina ~]$ chmod 200 dir1/file
[guest@ovgubina ~]$ cd dir1
[guest@ovgubina dir1]$ touch file700200
[guest@ovgubina dir1]$ rm file700200
[guest@ovgubina dir1]$ echo "test" > file
[guest@ovgubina dir1]$ cat file
cat: file: Permission denied
[guest@ovgubina dir1]$ mv file ffile
[guest@ovgubina dir1]$ mv ffile file
[guest@ovgubina dir1]$ chattr +i file
chattr: Permission denied while reading flags on file
[guest@ovgubina dir1]$
```

Рис. 4.13: d(700), -(200)


```

[guest@ovgubina ~]$ chmod 700 dir1
[guest@ovgubina ~]$ chmod 600 dir1/file
[guest@ovgubina ~]$ chmod 600 dir1
[guest@ovgubina ~]$ cd dir1
bash: cd: dir1: Permission denied
[guest@ovgubina ~]$ touch dir1/file600600
touch: cannot touch 'dir1/file600600': Permission denied
[guest@ovgubina ~]$ echo "test" > dir1/file
bash: dir1/file: Permission denied
[guest@ovgubina ~]$ cat dir1/file
cat: dir1/file: Permission denied
[guest@ovgubina ~]$ mv dir1/file dir1/ffile
mv: failed to access 'dir1/ffile': Permission denied
[guest@ovgubina ~]$ chattr +i dir1/file
chattr: Permission denied while trying to stat dir1/file
[guest@ovgubina ~]$ rm dir1/file
rm: cannot remove 'dir1/file': Permission denied
[guest@ovgubina ~]$

```

Рис. 4.14: d(600), -(600)

```

[guest@ovgubina ~]$ chmod 700 dir1
[guest@ovgubina ~]$ chmod 100 dir1/file
[guest@ovgubina ~]$ chmod 500 dir1
[guest@ovgubina ~]$ cd dir1
[guest@ovgubina dir1]$ touch file500100
touch: cannot touch 'file500100': Permission denied
[guest@ovgubina dir1]$ echo "test" > file
bash: file: Permission denied
[guest@ovgubina dir1]$ cat file
cat: file: Permission denied
[guest@ovgubina dir1]$ ls
ffff  file
[guest@ovgubina dir1]$ mv file ffile
mv: cannot move 'file' to 'ffile': Permission denied
[guest@ovgubina dir1]$ chattr +i file
chattr: Permission denied while reading flags on file
[guest@ovgubina dir1]$

```

Рис. 4.15: d(500), -(100)

```

[guest@ovgubina ~]$ chmod 700 dir1
[guest@ovgubina ~]$ chmod 400 dir1/file
[guest@ovgubina ~]$ chmod 400 dir1
[guest@ovgubina ~]$ cd dir1
bash: cd: dir1: Permission denied
[guest@ovgubina ~]$ touch dir1/file400400
touch: cannot touch 'dir1/file400400': Permission denied
[guest@ovgubina ~]$ echo "test" > dir1/file
bash: dir1/file: Permission denied
[guest@ovgubina ~]$ cat dir1/file
cat: dir1/file: Permission denied
[guest@ovgubina ~]$ ls dir1
ls: cannot access 'dir1/ffff': Permission denied
ls: cannot access 'dir1/file': Permission denied
ffff  file
[guest@ovgubina ~]$ mv dir1/file dir1/ffile
mv: failed to access 'dir1/ffile': Permission denied
[guest@ovgubina ~]$ chattr +i dir1/file
chattr: Permission denied while trying to stat dir1/file
[guest@ovgubina ~]$ rm dir1/file
rm: cannot remove 'dir1/file': Permission denied
[guest@ovgubina ~]$

```

Рис. 4.16: d(400), -(400)

```

[guest@ovgubina ~]$ chmod 700 dir1
[guest@ovgubina ~]$ chmod 600 dir1/file
[guest@ovgubina ~]$ chmod 300 dir1
[guest@ovgubina ~]$ cd dir1
[guest@ovgubina dir1]$ touch dir1/file300600
touch: cannot touch 'dir1/file300600': No such file or directory
[guest@ovgubina dir1]$ touch file300600
[guest@ovgubina dir1]$ ls
ls: cannot open directory '.': Permission denied
[guest@ovgubina dir1]$ rm file300600
[guest@ovgubina dir1]$ echo "test" > file
[guest@ovgubina dir1]$ cat file
test
[guest@ovgubina dir1]$ mv file ffile
[guest@ovgubina dir1]$ mv ffile file
[guest@ovgubina dir1]$ chattr +i file
chattr: Operation not permitted while setting flags on file
[guest@ovgubina dir1]$

```

Рис. 4.17: d(300), -(600)

```

[guest@ovgubina ~]$ chmod 700 dir1
[guest@ovgubina ~]$ chmod 700 dir1/file
[guest@ovgubina ~]$ chmod 200 dir1
[guest@ovgubina ~]$ cd dir1
bash: cd: dir1: Permission denied
[guest@ovgubina ~]$ touch dir1/file200700
touch: cannot touch 'dir1/file200700': Permission denied
[guest@ovgubina ~]$ echo "test" > dir1/file
bash: dir1/file: Permission denied
[guest@ovgubina ~]$ cat dir1/file
cat: dir1/file: Permission denied
[guest@ovgubina ~]$ ls dir1
ls: cannot open directory 'dir1': Permission denied
[guest@ovgubina ~]$ mv dir1/file dir1/ffilq
mv: failed to access 'dir1/ffilq': Permission denied
[guest@ovgubina ~]$ chatter +i dir1/file
chattr: Permission denied while trying to stat dir1/file
[guest@ovgubina ~]$ rm dir1/file
rm: cannot remove 'dir1/file': Permission denied
[guest@ovgubina ~]$ █

```

Рис. 4.18: d(200), -(700)

```

[guest@ovgubina ~]$ chmod 100 dir1/file
[guest@ovgubina ~]$ chmod 100 dir1
[guest@ovgubina ~]$ cd dir1
[guest@ovgubina dir1]$ touch file100100
touch: cannot touch 'file100100': Permission denied
[guest@ovgubina dir1]$ echo "test100100" > file
bash: file: Permission denied
[guest@ovgubina dir1]$ cat file
cat: file: Permission denied
[guest@ovgubina dir1]$ ls
ls: cannot open directory '.': Permission denied
[guest@ovgubina dir1]$ mv file ffile
mv: cannot move 'file' to 'ffile': Permission denied
[guest@ovgubina dir1]$ chatter +i file
chattr: Permission denied while reading flags on file
[guest@ovgubina dir1]$ rm file
rm: remove write-protected regular file 'file'? y
rm: cannot remove 'file': Permission denied
[guest@ovgubina dir1]$ █

```

Рис. 4.19: d(100), -(100)

Заполненная табл. 4.1 краткого описания стандартных каталогов Unix.

Таблица 4.1: Установленные права и разрешенные действия

Права ди- ректории	Права файла	Со- зда- ние фай- ла	Уда- ле- ние фай- ла	За- пись в файл	Чте- ние фай- ла	Смена дирек- тории	Про- смотр фай- лов дирек- тории	Пере- име- нова- ние фай- ла	Смена аттри- бутов файла
----- (000)	----- (000)	-	-	-	-	-	-	-	-
----- (000)	-- x----- (100)	-	-	-	-	-	-	-	-
----- (000)	- w----- (200)	-	-	-	-	-	-	-	-
----- (000)	- wx----- (300)	-	-	-	-	-	-	-	-
----- (000)	r----- (400)	-	-	-	-	-	-	-	-
----- (000)	r- x----- (500)	-	-	-	-	-	-	-	-
----- (000)	rw----- (600)	-	-	-	-	-	-	-	-
----- (000)	rw- x----- (700)	-	-	-	-	-	-	-	-

Права ди- ректории	Права файла	Со- зда- ние фай- ла	Уда- ле- ние фай- ла	За- пись в файл	Чте- ние фай- ла	Смена дирек- тории	Про- смотр фай- лов дирек- тории	Пере- име- нова- ние фай- ла	Смена аттри- бутов файла
--x----- (100)	----- (000)	-	-	-	-	+	-	-	-
--x----- (100)	-- x----- (100)	-	-	-	-	+	-	-	-
--x----- (100)	- w----- (200)	-	-	+	-	+	-	-	-
--x----- (100)	- wx----- (300)	-	-	+	-	+	-	-	-
--x----- (100)	r----- (400)	-	-	+	+	+	-	-	+
--x----- (100)	r- x----- (500)	-	-	-	+	+	-	-	+
--x----- (100)	rw----- (600)	-	+	+	+	+	-	-	+
--x----- (100)	rwX----- (700)	-	+	+	+	+	-	-	+
-w----- (200)	----- (000)	-	-	-	-	-	-	-	-

Права ди- ректории	Права файла	Со- зда- ние фай- ла	Уда- ле- ние фай- ла	За- пись в файл	Чте- ние фай- ла	Смена дирек- тории	Про- смотр фай- лов дирек- тории	Пере- име- нова- ние фай- ла	Смена аттри- бутов файла
-w----- (200)	-- x----- (100)	-	-	-	-	-	-	-	-
-w----- (200)	- w----- (200)	-	-	-	-	-	-	-	-
-w----- (200)	- wx----- (300)	-	-	-	-	-	-	-	-
-w----- (200)	r----- (400)	-	-	-	-	-	-	-	-
-w----- (200)	r- x----- (500)	-	-	-	-	-	-	-	-
-w----- (200)	rw----- (600)	-	-	-	-	-	-	-	-
-w----- (200)	rwX----- (700)	-	-	-	-	-	-	-	-
-wx----- (300)	-----+ (000)	+	+	-	-	+	-	+	-
-wx----- (300)	-- x----- (100)	+	+	-	-	+	-	+	-

Права ди- ректории	Права файла	Со- зда- ние фай- ла	Уда- ле- ние фай- ла	За- пись в файл	Чте- ние фай- ла	Смена дирек- тории	Про- смотр фай- лов дирек- тории	Пере- име- нова- ние фай- ла	Смена аттри- бутов файла
-wx----- (300)	- w----- (200)	+	+	+	-	+	-	+	-
-wx----- (300)	- wx----- (300)	+	+	+	-	+	-	+	-
-wx----- (300)	r----- (400)	+	+	-	+	+	-	+	+
-wx----- (300)	r- x----- (500)	+	+	-	+	+	-	+	+
-wx----- (300)	rw----- (600)	+	+	+	+	+	-	+	+
-wx----- (300)	rwX----- (700)	+	+	+	+	+	-	+	+
r----- (400)	----- (000)	-	-	-	-	-	+	-	-
r----- (400)	-- x----- (100)	-	-	-	-	-	+	-	-
r----- (400)	- w----- (200)	-	-	-	-	-	+	-	-

Права ди- ректории	Права файла	Со- зда- ние фай- ла	Уда- ле- ние фай- ла	За- пись в файл	Чте- ние фай- ла	Смена дирек- тории	Про- смотр фай- лов дирек- тории	Пере- име- нова- ние фай- ла	Смена аттри- бутов файла
r----- (400)	- wx----- (300)	-	-	-	-	-	+	-	-
r----- (400)	r----- (400)	-	-	-	-	-	+	-	-
r----- (400)	r- x----- (500)	-	-	-	-	-	+	-	-
r----- (400)	rw----- (600)	-	-	-	-	-	+	-	-
r----- (400)	rwX----- (700)	-	-	-	-	-	+	-	-
r-x----- (500)	----- (000)	-	-	-	-	+	+	-	-
r-x----- (500)	-- x----- (100)	-	-	-	-	+	+	-	-
r-x----- (500)	- w----- (200)	-	-	+	-	+	+	-	-
r-x----- (500)	- wx----- (300)	-	-	+	-	+	+	-	-

Права ди- ректории	Права файла	Со- зда- ние фай- ла	Уда- ле- ние фай- ла	За- пись в файл	Чте- ние фай- ла	Смена дирек- тории	Про- смотр фай- лов дирек- тории	Пере- име- нова- ние фай- ла	Смена аттри- бутов файла
r-x----- (500)	r----- (400)	-	-	+	+	+	+	-	+
r-x----- (500)	r- x----- (500)	-	-	-	+	+	+	-	+
r-x----- (500)	rw----- (600)	-	+	+	+	+	+	-	+
r-x----- (500)	rwX----- (700)	-	+	+	+	+	+	-	+
rw----- (600)	----- (000)	-	-	-	-	-	+	-	-
rw----- (600)	-- x----- (100)	-	-	-	-	-	+	-	-
rw----- (600)	- w----- (200)	-	-	-	-	-	+	-	-
rw----- (600)	- wx----- (300)	-	-	-	-	-	+	-	-
rw----- (600)	r----- (400)	-	-	-	-	-	+	-	-

Права ди- ректории	Права файла	Со- зда- ние фай- ла	Уда- ле- ние фай- ла	За- пись в файл	Чте- ние фай- ла	Смена дирек- тории	Про- смотр фай- лов дирек- тории	Пере- име- нова- ние фай- ла	Смена аттри- бутов файла
rw----- (600)	r- x----- (500)	-	-	-	-	-	+	-	-
rw----- (600)	rw----- (600)	-	-	-	-	-	+	-	-
rw----- (600)	rwX----- (700)	-	-	-	-	-	+	-	-
rwX----- (700)	-----+ (000)	+	+	-	-	+	+	+	-
rwX----- (700)	-- x----- (100)	+	+	-	-	+	+	+	-
rwX----- (700)	- w----- (200)	+	+	+	-	+	+	+	-
rwX----- (700)	- wx----- (300)	+	+	+	-	+	+	+	-
rwX----- (700)	r-----+ (400)	+	+	-	+	+	+	+	+
rwX----- (700)	r- x----- (500)	+	+	-	+	+	+	+	+

Права ди- ректории	Права файла	Со- зда- ние фай- ла	Уда- ле- ние фай- ла	За- пись в файл	Чте- ние фай- ла	Смена дирек- тории	Про- смотр фай- лов дирек- тории	Пере- име- нова- ние фай- ла	Смена аттри- бутов файла
rwX----- (700)	rw-----+	+	+	+	+	+	+	+	+
rwX----- (700)	rwX-----+	+	+	+	+	+	+	+	+

15. На основании заполненной таблицы 4.1 определите те или иные минимально необходимые права для выполнения операций внутри директории dir1, заполните табл. 4.2.

Таблица 4.2: Минимальные права для совершения операций

Операция	Минимальные права на директорию	Минимальные права на файл
Создание файла	d-wx---(300)	-----(000)
Удаление файла	d-wx---(300)	-----(000)
Чтение файла	d-x---(100)	-r----(400)
Запись в файл	d-x---(100)	-w----(200)
Переименование файла	d-wx---(300)	-----(000)
Создание поддиректории	d-wx---(300)	-----(000)
Удаление поддиректории	d-wx---(300)	-----(000)

5 Выводы

Получила практические навыки работы в консоли с атрибутами файлов, закрепила теоретические основы дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux. Заполнила опытным путем таблицы “Установленные права и разрешенные действия” и “Минимальные права для совершения операций”.

Список литературы

1. Права доступа в Linux [Электронный ресурс]. 2023. URL: <https://codechick.io/tutorials/unix-linux/unix-linux-permissions>.
2. Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask) [Электронный ресурс]. 2023. URL: <https://habr.com/ru/articles/469667/>.
3. Права доступа к файлам в Linux [Электронный ресурс]. 2023. URL: <https://losst.pro/prava-dostupa-k-fajlam-v-linux>.