

Лабораторная работа № 2

Дискреционное разграничение прав в Linux.
Основные атрибуты

Миленин Иван Витальевич

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.

Задание

Создать нового пользователя с нужными правами, провести эксперимент по выявлению минимально необходимых прав для действий над файловой структурой.

Теоретическое описание

В операционной системе Linux есть много отличных функций безопасности, но она из самых важных - это система прав доступа к файлам. Linux, как последователь идеологии ядра Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в linux продуманы очень хорошо.

Изначально каждый файл имел три параметра доступа [1]:

- Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;

- Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;

- Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

- Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение.

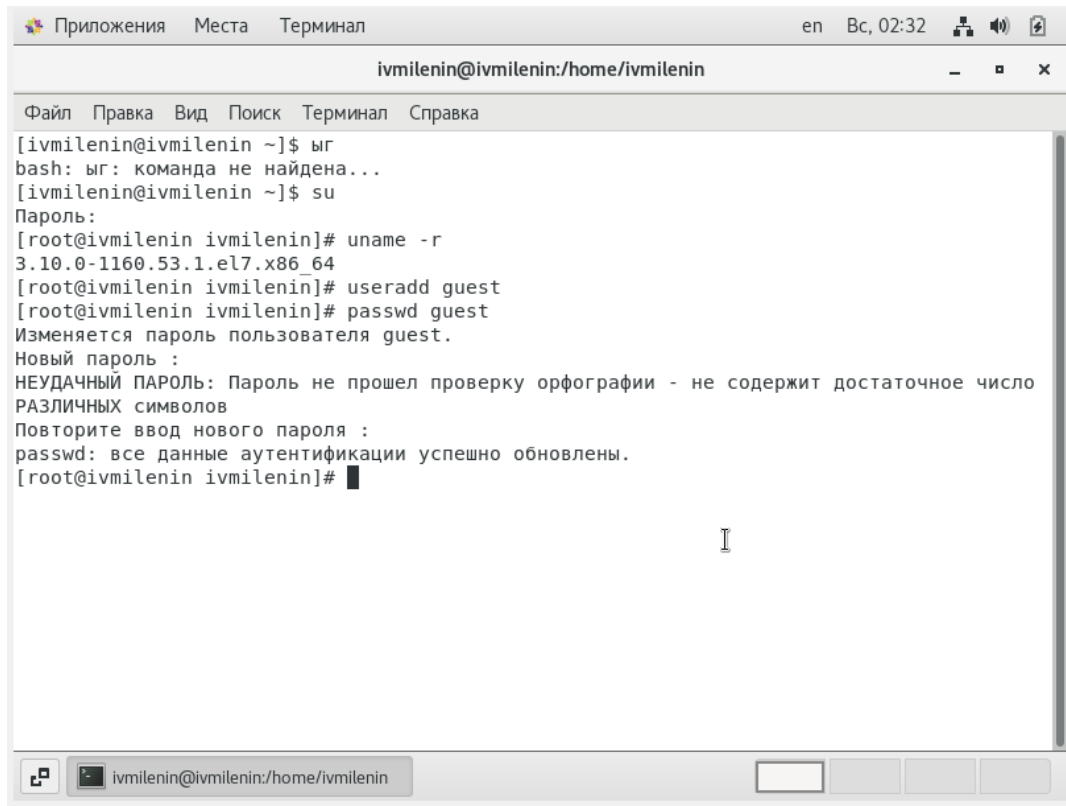
- Группа - любая группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу.

- Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла.

Для управления правами используется команда `chmod`. При использовании `chmod` вы можете устанавливать разрешения для пользователя (user), группы (group) и других (other). Вы можете использовать эту команду в двух режимах: относительный режим и абсолютный режим. В абсолютном режиме три цифры используются для установки основных разрешений [2].

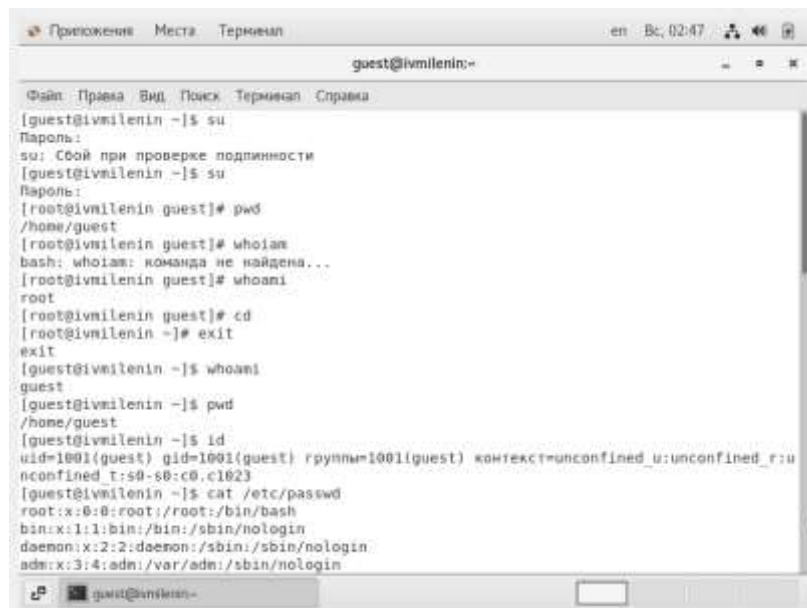
Ход работы

1. Создаем нового пользователя `guest` и задаем для него пароль (иллюстр. [-@fig:001]). Входим на новую учетную запись, вводим `pwd`. Мы находимся в домашней директории, о чем говорит значок “тильда” (~) в приглашении командной строки, вывод команды `pwd` и тот факт, что мы еще никуда не переходили, а по умолчанию пользователь стартует в домашней директории. Далее проверяем пользователя командой `whoami` - мы `guest`. Командой `id` проверяем группы и имя пользователя. Тут тоже имя пользователя `guest`, группа `guest`, `uid` и `gid` равны 1001. Команда `groups` выводит единственную группу `guest`, куда мы входим. Приглашение командной строки так же указывает на то, что мы работаем под пользователем `guest` (иллюстр. [-@fig:002]).



```
ivmilenin@ivmilenin:~/home/ivmilenin
Файл  Правка  Вид  Поиск  Терминал  Справка
[ivmilenin@ivmilenin ~]$ ыг
bash: ыг: команда не найдена...
[ivmilenin@ivmilenin ~]$ su
Пароль:
[root@ivmilenin ivmilenin]# uname -r
3.10.0-1160.53.1.el7.x86_64
[root@ivmilenin ivmilenin]# useradd guest
[root@ivmilenin ivmilenin]# passwd guest
Изменяется пароль пользователя guest.
Новый пароль :
НЕУДАЧНЫЙ ПАРОЛЬ: Пароль не прошел проверку орфографии - не содержит достаточное число
РАЗЛИЧНЫХ символов
Повторите ввод нового пароля :
passwd: все данные аутентификации успешно обновлены.
[root@ivmilenin ivmilenin]#
```

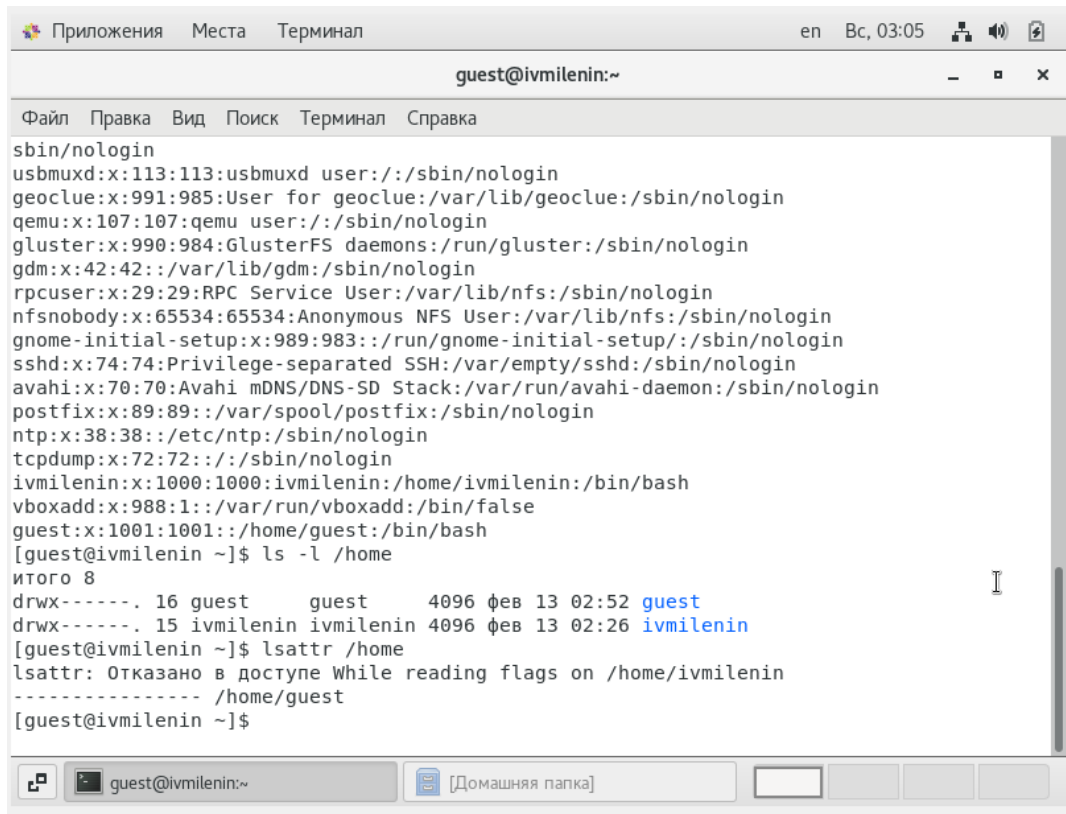
Создание пользователя



```
guest@ivmlenin:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@ivmlenin ~]$ su  
Пароль:  
su: Сбой при проверке подлинности  
[guest@ivmlenin ~]$ su  
Пароль:  
[root@ivmlenin guest]# pwd  
/home/guest  
[root@ivmlenin guest]# whoiam  
bash: whoiam: команда не найдена...  
[root@ivmlenin guest]# whoami  
root  
[root@ivmlenin guest]# cd  
[root@ivmlenin ~]# exit  
exit  
[guest@ivmlenin ~]$ whoami  
guest  
[guest@ivmlenin ~]$ pwd  
/home/guest  
[guest@ivmlenin ~]$ id  
uid=1001(guest) gid=1001(guest) rpyynn=1001(guest) контекст=unconfined_u:unconfined_r:u  
nconfined_t:s0-s0:c0.c1023  
[guest@ivmlenin ~]$ 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
```

Лог консоли guest

2. Теперь проверим `/etc/passwd` командой `cat`. Найдем там себя в последней строчке. Наш `uid` равен `gid` и равен 1001, что совпадает с выводом `id`. Проверим существующие домашние директории командой `ls -l /home`. Успешно. Тут увидим две папки (по количеству пользователей), обе с правами 700. Проверим вывод команды `lsattr` на те же папки, тут увидим, что расширенных атрибутов нашей домашней директории нет, а просмотреть атрибуты папки другого пользователя нам не дает система (иллюстр. [-@fig:003]).



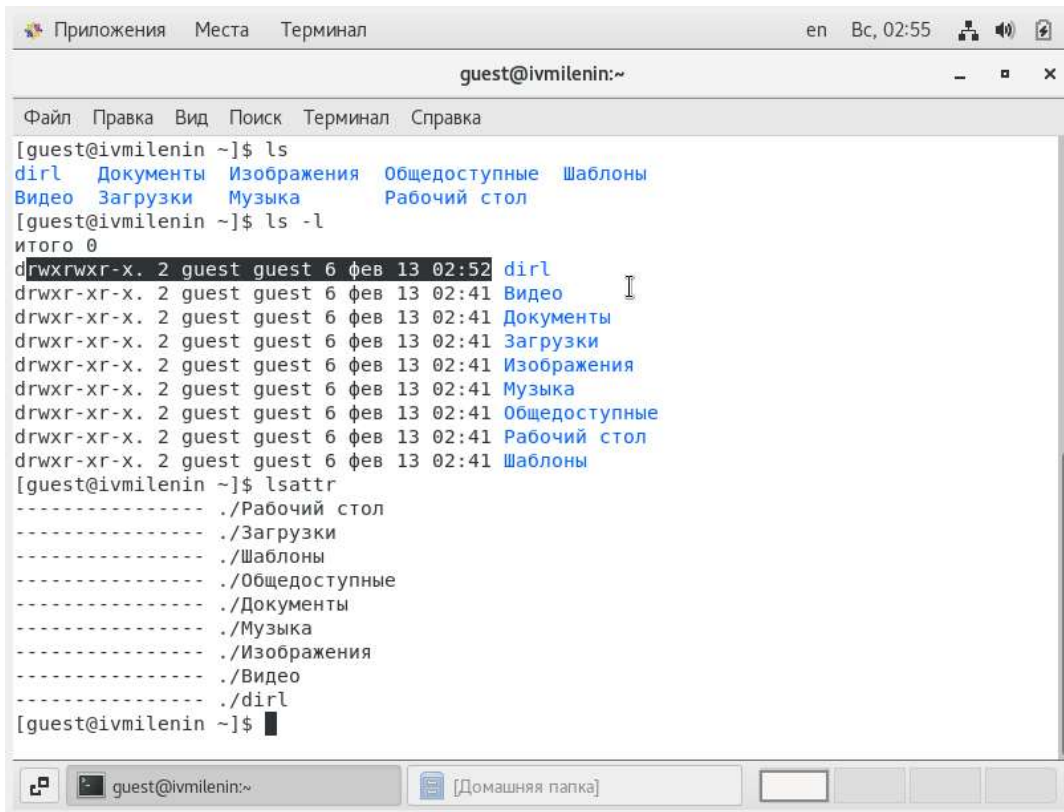
```
Приложения  Места  Терминал  en  Вс, 03:05  [иконки]
guest@ivmilenin:~

Файл  Правка  Вид  Поиск  Терминал  Справка

sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
geoclue:x:991:985:User for geoclue:/var/lib/geoclue:/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
gluster:x:990:984:GlusterFS daemons:/run/gluster:/sbin/nologin
gdm:x:42:42:/var/lib/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
gnome-initial-setup:x:989:983:/run/gnome-initial-setup:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
postfix:x:89:89:/var/spool/postfix:/sbin/nologin
ntp:x:38:38:/etc/ntp:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
ivmilenin:x:1000:1000:ivmilenin:/home/ivmilenin:/bin/bash
vboxadd:x:988:1:/var/run/vboxadd:/bin/false
guest:x:1001:1001:/home/guest:/bin/bash
[guest@ivmilenin ~]$ ls -l /home
итого 8
drwx-----. 16 guest      guest      4096 фев 13 02:52 guest
drwx-----. 15 ivmilenin ivmilenin 4096 фев 13 02:26 ivmilenin
[guest@ivmilenin ~]$ lsattr /home
lsattr: Отказано в доступе While reading flags on /home/ivmilenin
----- /home/guest
[guest@ivmilenin ~]$
```

Лог консоли по проверке домашних директорий

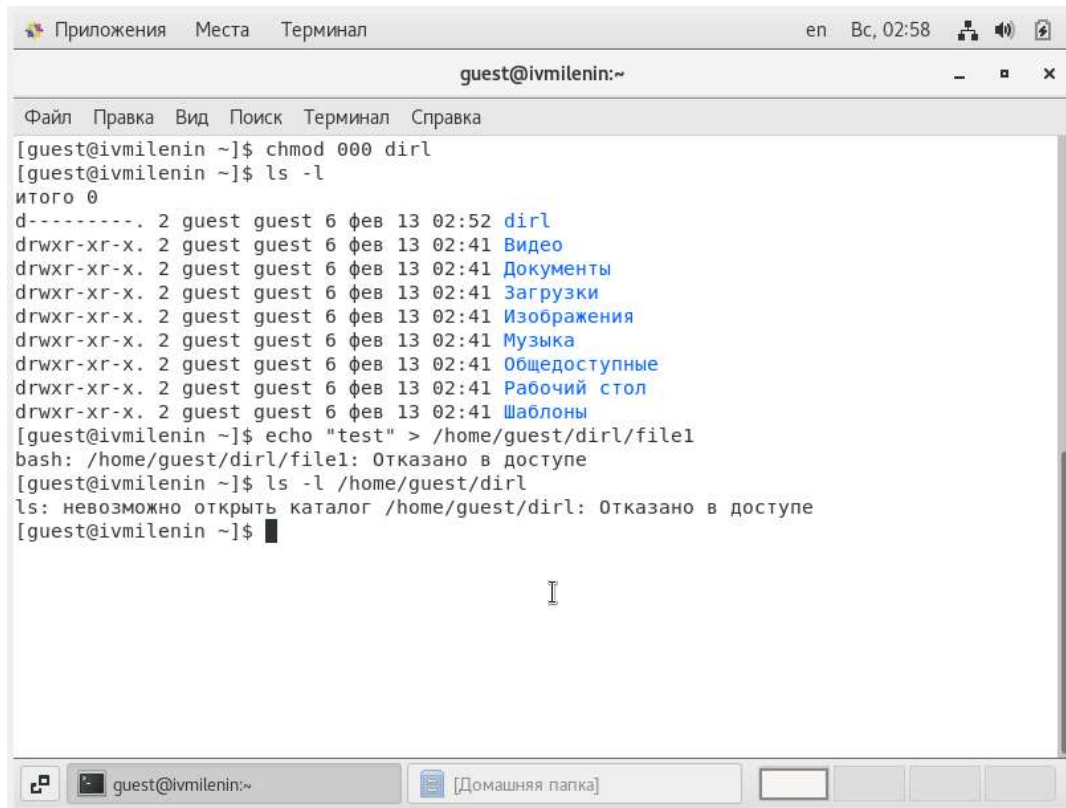
3. Создадим в домашней директории guest папку dir1. Папка получила права 775 и не получила никаких расширенных атрибутов (иллюстр. [-@fig:004]).



```
guest@ivmilenin:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@ivmilenin ~]$ ls  
dirl  Документы  Изображения  Общедоступные  Шаблоны  
Видео  Загрузки  Музыка  Рабочий стол  
[guest@ivmilenin ~]$ ls -l  
итого 0  
drwxrwxr-x. 2 guest guest 6 фев 13 02:52 dirl  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Видео  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Документы  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Загрузки  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Изображения  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Музыка  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Общедоступные  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Рабочий стол  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Шаблоны  
[guest@ivmilenin ~]$ lsattr  
----- ./Рабочий стол  
----- ./Загрузки  
----- ./Шаблоны  
----- ./Общедоступные  
----- ./Документы  
----- ./Музыка  
----- ./Изображения  
----- ./Видео  
----- ./dirl  
[guest@ivmilenin ~]$
```

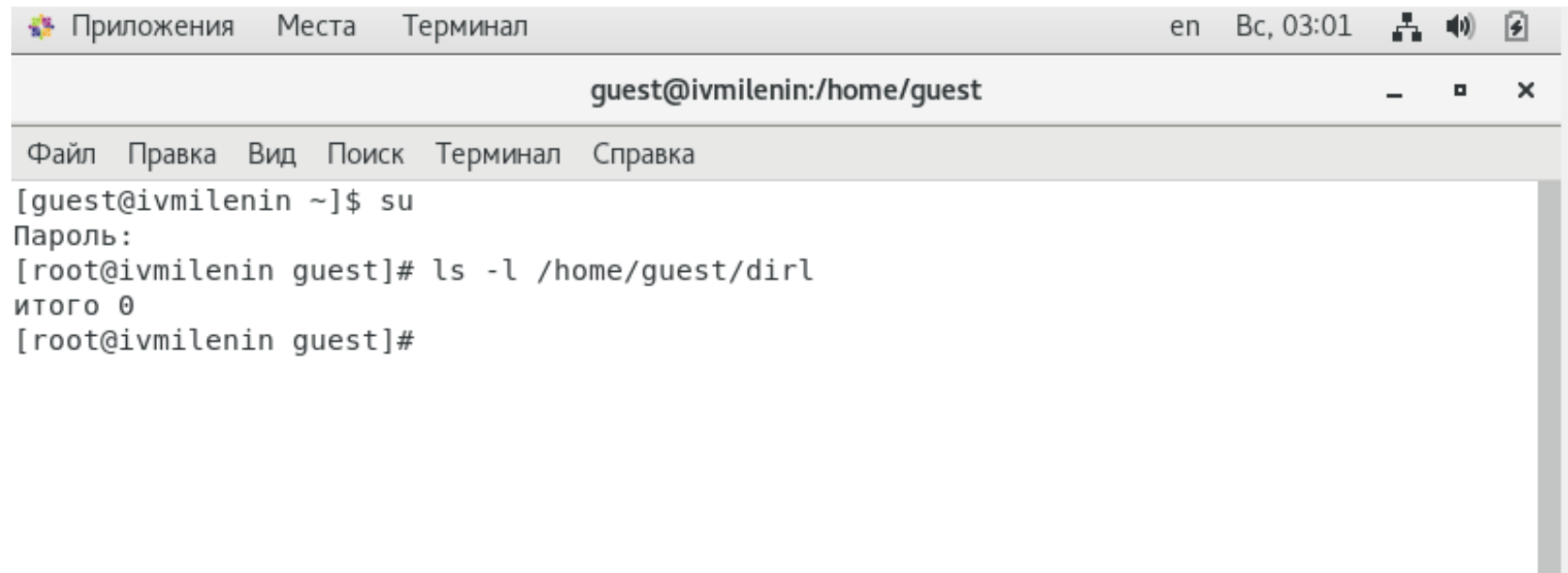
Лог консоли по созданию dir1

4. Командой `chmod` обнуляем права на `dir1` и проверяем это. Далее пытаемся создать в папке файл `file1`, что у нас не выходит, так как система блокирует действие из-за недостатка прав (иллюстр. [-@fig:005]). Соответственно, и сам файл создан не был (иллюстр. [-@fig:006]).



```
guest@ivmilenin:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[guest@ivmilenin ~]$ chmod 000 dir1  
[guest@ivmilenin ~]$ ls -l  
итого 0  
d----- . 2 guest guest 6 фев 13 02:52 dir1  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Видео  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Документы  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Загрузки  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Изображения  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Музыка  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Общедоступные  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Рабочий стол  
drwxr-xr-x. 2 guest guest 6 фев 13 02:41 Шаблоны  
[guest@ivmilenin ~]$ echo "test" > /home/guest/dirl/file1  
bash: /home/guest/dirl/file1: Отказано в доступе  
[guest@ivmilenin ~]$ ls -l /home/guest/dirl  
ls: невозможно открыть каталог /home/guest/dirl: Отказано в доступе  
[guest@ivmilenin ~]$
```

Обнуление прав и попытка создания файла

A terminal window titled "guest@ivmilenin:/home/guest" with a menu bar containing "Файл", "Правка", "Вид", "Поиск", "Терминал", and "Справка". The terminal shows a user switching to root with 'su', then running 'ls -l /home/guest/dir1' which returns 'итого 0'.

```
guest@ivmilenin:/home/guest  
[guest@ivmilenin ~]$ su  
Пароль:  
[root@ivmilenin guest]# ls -l /home/guest/dir1  
итого 0  
[root@ivmilenin guest]#
```

Пустая папка dir1 после попытки создать файл

5. Следующим шагом проведем эксперимент по выявлению минимально необходимых прав для действий над файловой структурой. Для этого используем нашу папку dir1, файлы внутри неё и функционал прав доступа ОС Linux. Для каждой комбинации атрибутов доступа (r, w, x) на папку и на файл попробуем осуществить ряд действий и таким образом выявим минимально необходимые права для каждого действия. Атрибуты используем только для владельца, поэтому комбинаций будет $2^3 \cdot 2^3 = 2^6 = 64$. В каждой строчке будет по 8 действий. Проверять осуществимость функции будем следующими командами:

touch для создания файла в директории;

rm для удаления файла в директории;

echo для записи в файл;

cat для чтения из файла;

mv для переименования файла;

chattr для изменения атрибутов файла;

cd для смены директории;

ls для просмотра файлов в директории.

```
guest@ivmilenin:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@ivmilenin ~]$ touch dirl/file2  
touch: невозможно выполнить touch для «dirl/file2»: Отказано в доступе  
[guest@ivmilenin ~]$ rm dirl/file1  
rm: невозможно удалить «dirl/file1»: Отказано в доступе  
[guest@ivmilenin ~]$ echo "1" > dirl/file1  
bash: dirl/file1: Отказано в доступе  
[guest@ivmilenin ~]$ cat dirl/file1  
cat: dirl/file1: Отказано в доступе  
[guest@ivmilenin ~]$ cd dirl/file1  
bash: cd: dirl/file1: Отказано в доступе  
[guest@ivmilenin ~]$ mv dirl/file1 dirl/file2  
mv: не удалось получить доступ к «dirl/file2»: Отказано в доступе  
[guest@ivmilenin ~]$ chattr -u dirl/file1  
bash: chattr: команда не найдена...  
Аналогичная команда: 'chat'  
[guest@ivmilenin ~]$ chattr -u dirl/file1  
chattr: Отказано в доступе while trying to stat dirl/file1  
[guest@ivmilenin ~]$
```

Пример ввода команд для проверки прав

Полученные результаты представлены в виде таблицы (иллюстр. [-@fig:008]).

[illegible]

Установленные права и разрешённые действия

На основе данных полученной выше таблицы построим вторую таблицу, иллюстрирующую минимально необходимые права для совершения определенных операций.

Операция	Мин. права на директорию	Мин. права на файл
Создание файла	300	000
Удаление файла	300	000
Чтение файла	100	400
Запись в файл	100	200
Переименование файла	300	000
Создание поддиректории	300	100
Удаление поддиректории	300	100

Выводы

В ходе работы мы успешно провели эксперимент по выявлению минимально необходимых прав для действий над файловой структурой, получили ряд практических навыков работы в консоли с атрибутами файлов, закрепили теоретические основы дискреционного разграничения доступа в ОС Linux.

Список литературы

1. Права доступа к файлам в linux. // Losst. 2020. URL: <https://losst.ru/prava-dostupa-k-fajlam-v-linux> (дата обращения 01.10.2021).
2. Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask). // habr.com. 2019. URL: <https://habr.com/ru/post/469667/> (дата обращения 01.10.2021).
3. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян.
Информационная безопасность компьютерных сетей:
лабораторные работы. // Факультет физико-