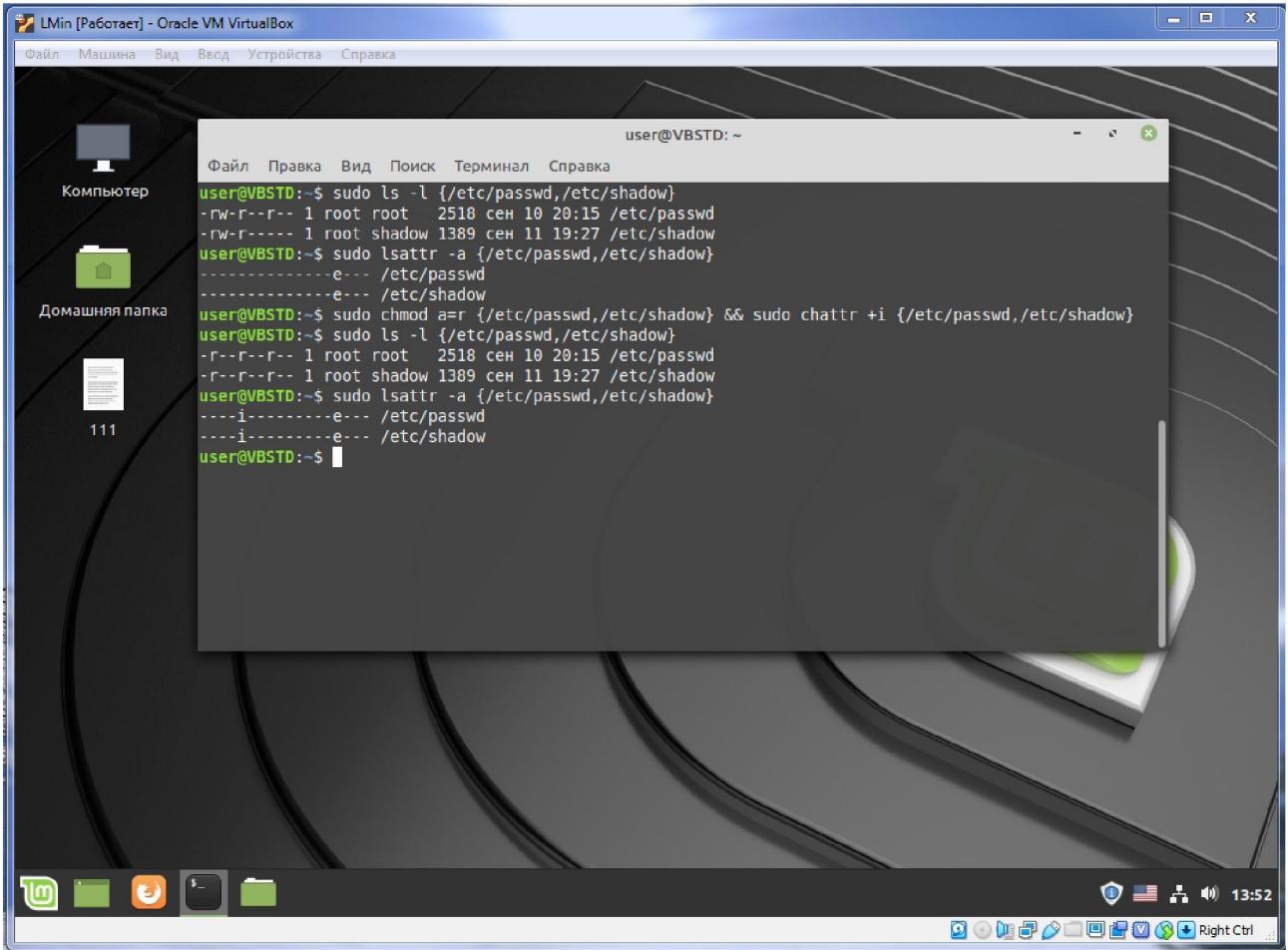


3. Защитить нашу систему Linux от создания или модификаций базы данных пользователей. Опишите возможные шаги для этого.

Т. к. все учетные записи, используемые в системе хранят свои наименования в /etc/passwd, а хэши паролей в /etc/shadow, то для защиты от модификации базы данных пользователей нашей ОС, рациональнее всего для защиты от создания или модификаций базы данных пользователей было бы защитить от изменений именно эти файлы с данными. Поступим следующим образом:

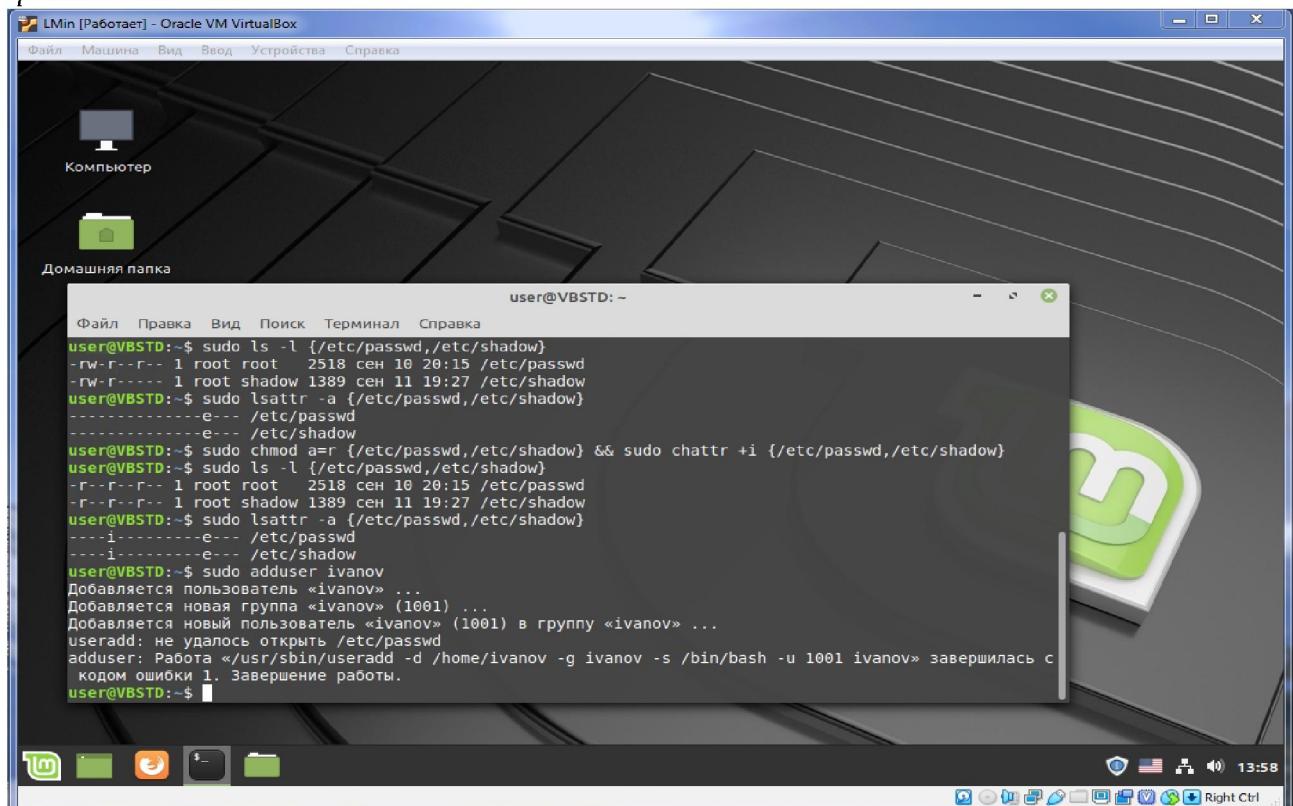
Заменим для всех пользователей права на базу данных пользователей, представленную файлами {/etc/passwd, /etc/shadow} на только чтение и защитим флагом, который нельзя удалить (может только root) неизменяемый атрибут (chattr +i).

```
sudo chattr +i {/etc/passwd, /etc/shadow} && sudo chmod a=r {/etc/passwd, /etc/shadow}
```



И проверим себя:

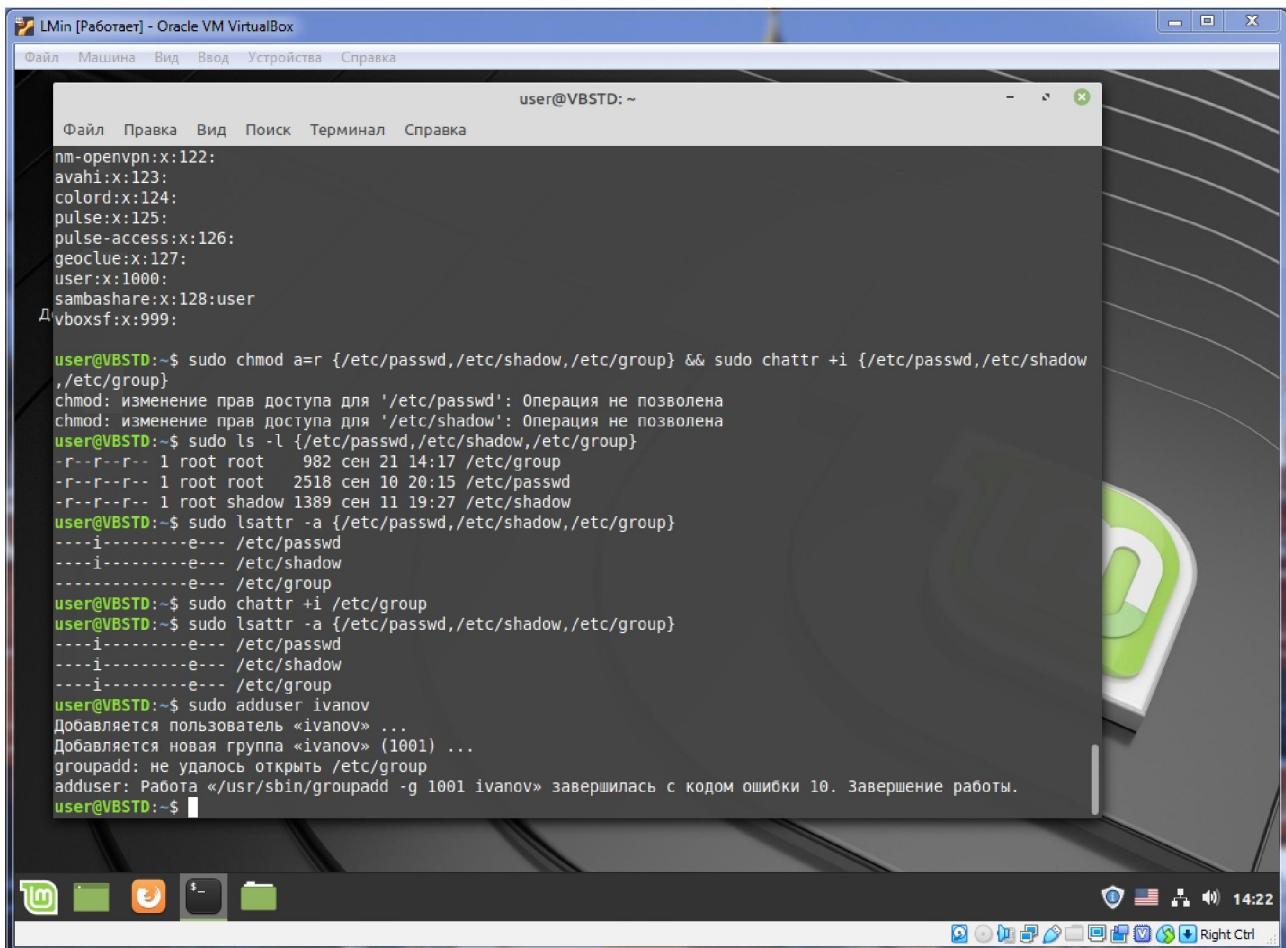
Для проверки предлагаю попытку добавления в систему некого пользователя по фамилии Иванов:



И так, мы видим, что процедура добавления пользователя вернула **код ошибки 1: не удалось открыть на изменение** `/etc/passwd`, однако у нас в системе появилась новая группа это группа «ivanov» с GUID (1001), надо полагать, что эта группа в нашей ОС при защите базы данных пользователей является лишней → может быть удалена через редактор `vi`, а файл `/etc/group`, тоже подлежит защите, значит конечный вид команды по защите базы данных пользователей в ОС семейства Linux должен быть таким:

```
sudo chattr +i {/etc/passwd,/etc/shadow,/etc/group} &&
sudo chmod a=r {/etc/passwd,/etc/shadow,/etc/group}
```

Да, ещё 1 проверка показала, что 2 файла уже защищены, поэтому пришлось вместо приведённого правильного вида команды защитить 3 файл «вручную», но на чистой системе данная команда пройдёт без ошибок.



```
ЛМин [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
user@VBSTD: ~
Файл Правка Вид Поиск Терминал Справка
nm-openvpn:x:122:
avahi:x:123:
colord:x:124:
pulse:x:125:
pulse-access:x:126:
geoclue:x:127:
user:x:1000:
sambashare:x:128:user
 vboxsf:x:999:

user@VBSTD:~$ sudo chmod a=r {/etc/passwd,/etc/shadow,/etc/group} && sudo chattr +i {/etc/passwd,/etc/shadow,/etc/group}
chmod: изменение прав доступа для '/etc/passwd': Операция не позволена
chmod: изменение прав доступа для '/etc/shadow': Операция не позволена
user@VBSTD:~$ sudo ls -l {/etc/passwd,/etc/shadow,/etc/group}
-r--r--r-- 1 root root  982 сен 21 14:17 /etc/group
-r--r--r-- 1 root root 2518 сен 10 20:15 /etc/passwd
-r--r--r-- 1 root shadow 1389 сен 11 19:27 /etc/shadow
user@VBSTD:~$ sudo lsattr -a {/etc/passwd,/etc/shadow,/etc/group}
----i-----e--- /etc/passwd
----i-----e--- /etc/shadow
----i-----e--- /etc/group
user@VBSTD:~$ sudo chattr +i /etc/group
user@VBSTD:~$ sudo lsattr -a {/etc/passwd,/etc/shadow,/etc/group}
----i-----e--- /etc/passwd
----i-----e--- /etc/shadow
----i-----e--- /etc/group
user@VBSTD:~$ sudo adduser ivanov
Добавляется пользователь «ivanov» ...
Добавляется новая группа «ivanov» (1001) ...
groupadd: не удалось открыть /etc/group
adduser: Работа «/usr/sbin/groupadd -g 1001 ivanov» завершилась с кодом ошибки 10. Завершение работы.
user@VBSTD:~$
```

4. Создайте директорию, которая будет принадлежать группе пользователей, причем каждый пользователь из этой группы должен иметь возможность читать данные из файлов, записывать данные в файлы и создавать новые файлы. Сделайте так, чтобы пользователи могли удалять только собственноручно созданные файлы.

Действия:

- 1) `usr@PCLin:~$ sudo groupadd testusrs #Заведём новую группу пользователей`
- 2) Добавим в нашу тестовую систему для примера 3-х пользователей: `otus`, `otus2`, `otus3` и добавим их в группу `testusrs`:

```
sudo useradd -g testusrs otus
```

```
sudo useradd -g testusr otus2
sudo useradd -g testusr otus3
```

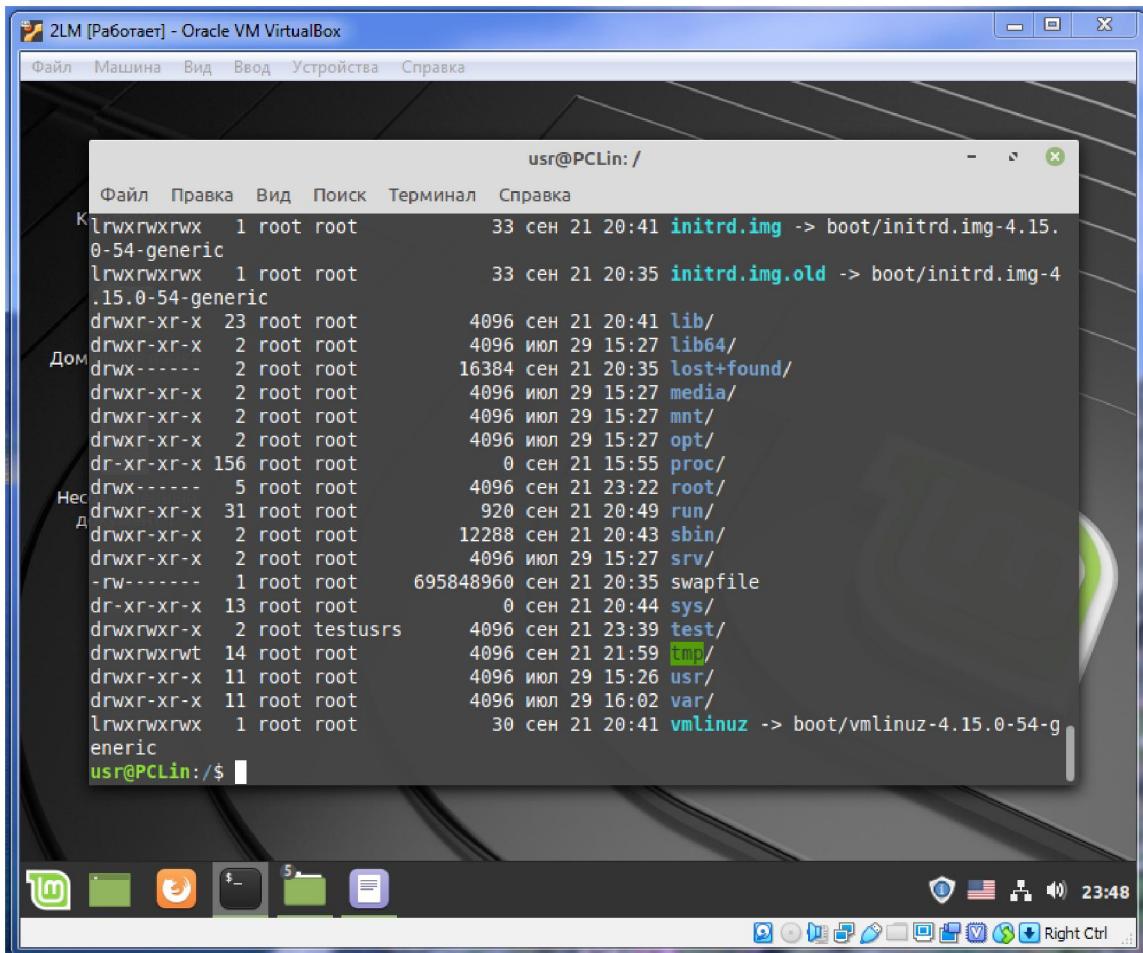
```
usr@PCLin:~$ sudo -u otus groups
testusr
usr@PCLin:~$ sudo -u otus2 groups
testusr
usr@PCLin:~$ sudo -u otus3 groups
testusr
```

3) Создадим в корне нашего системного диска некую директорию **test** и рекурсивно назначим ей и всему её содержимому владельцу группу пользователей **testusr**:

```
usr@PCLin:~$ cd / #Переходим в корень
usr@PCLin:$ sudo mkdir -p test #создаём директорию
```

```
usr@PCLin:~$ cd /
usr@PCLin:/$ sudo mkdir -p test
usr@PCLin:/$ ls
bin dev initrd.img lib64 mnt root srv test var
boot etc initrd.img.old lost+found opt run swapfile tmp vmlinuz
cdrom home lib media proc sbin sys usr
usr@PCLin:/$ sudo chown -R :testusr /test
usr@PCLin:/$ ls -l /test
итого 0
```

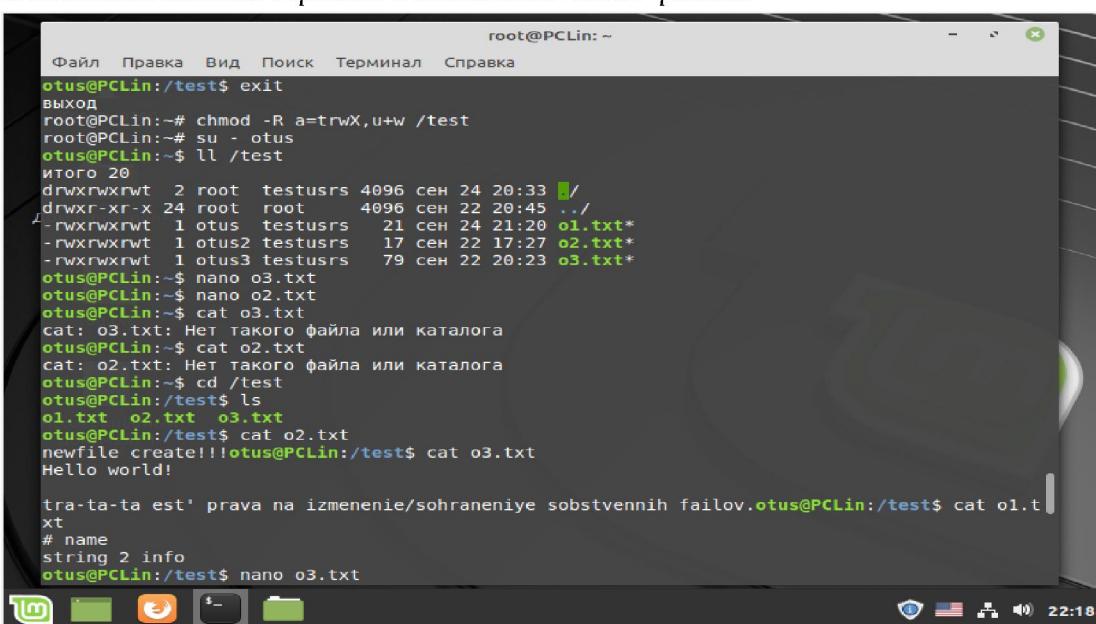
```
usr@PCLin:$ ls #Проверяем наличие
bin dev initrd.img lib64 mnt root srv test var
boot etc initrd.img.old lost+found opt run swapfile tmp vmlinuz
cdrom home lib media proc sbin sys usr
usr@PCLin:/$ sudo chown -R :testusr /test #Меняем владельца
usr@PCLin:/$ ls -l /test #Смотрим текущие права
```



...

usr@PCLin:~\$ sudo chmod -R a=trwX,u+w /test #Заменяем текущие права директории на необходимые нам для выполнения одного из пунктов условия задания, а именно:

Создать директорию, которая будет принадлежать группе пользователей, причем каждый пользователь из этой группы должен иметь возможность читать данные из файлов, записывать данные в файлы и создавать новые файлы.



За одно проверяем себя, изменяя содержимое файлов, перезаписывая их и читая.

Теперь по 2 пункту задания:

Сделайте так, чтобы пользователи могли удалять только собственоручно созданные файлы.

Для реализации этого пункта, заменим права нашей директории следующим образом:

```
usr@PCLin:~$ sudo chmod -R a=trX,u+w,g+w /test
```

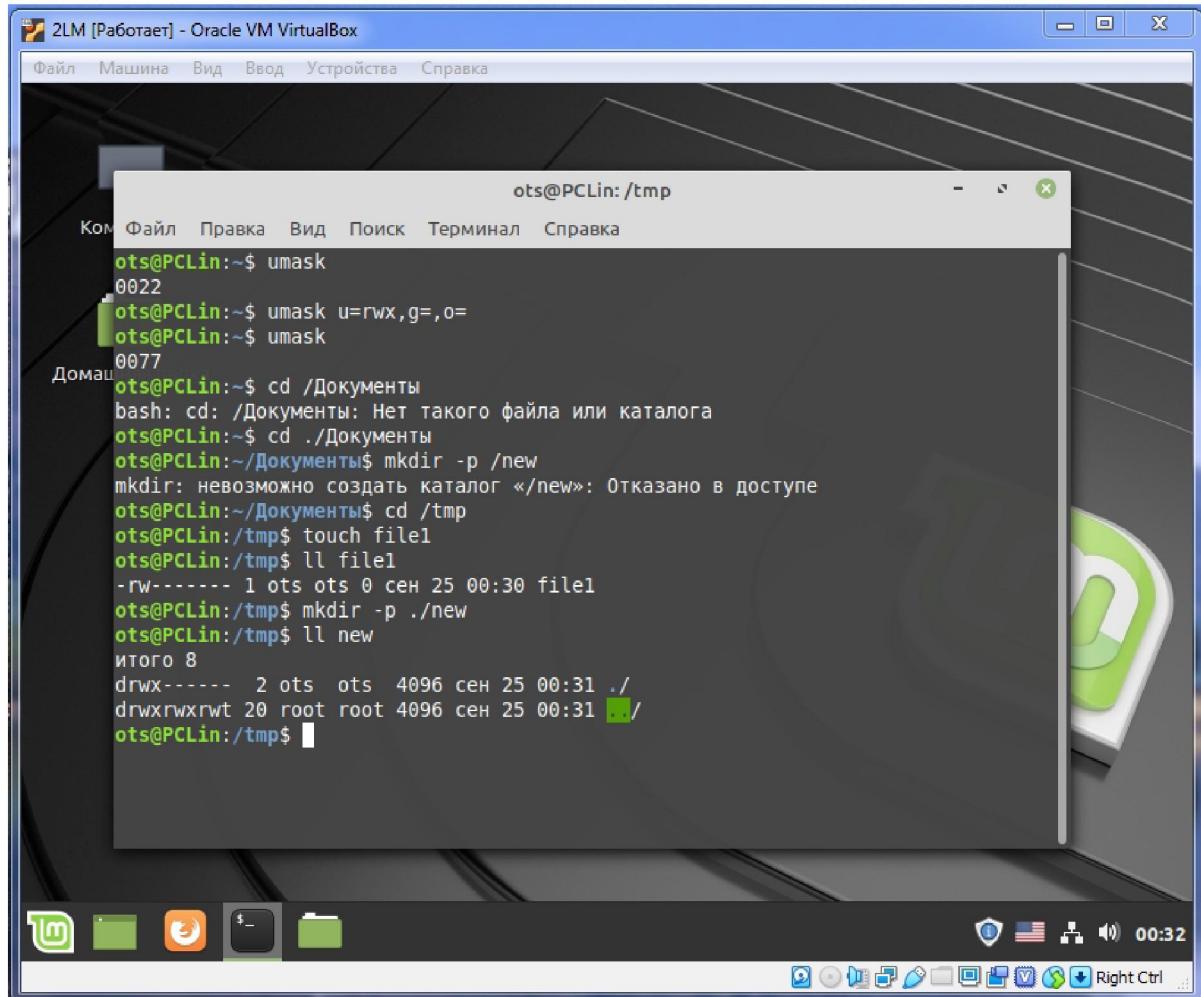
```
otus@PCLin: /test
Файл Правка Вид Поиск Терминал Справка
ВЫХОД
root@PCLin:~# chmod -R a=trX,u+w,g+w /test
root@PCLin:~# su - otus
otus@PCLin:~$ cd /test
otus@PCLin:/test$ cat ot.txt
ototot
45345
otus@PCLin:/test$ rm ot.txt
otus@PCLin:/test$ cat ot3.txt
cat: ot3.txt: Нет такого файла или каталога
otus@PCLin:/test$ ls
o1.txt o2.txt o3.txt
otus@PCLin:/test$ cat o3.txt
Hello world!

tra-ta-ta est' prava na izmenenie/sohraneniye sobstvennih failov.

otus@PCLin:/test$ rm o3.txt
rm: невозможно удалить 'o3.txt': Операция не позволена
otus@PCLin:/test$ cat o2.txt
newfile create!!!
otus@PCLin:/test$ rm o2.txt
rm: невозможно удалить 'o2.txt': Операция не позволена
otus@PCLin:/test$ touch file.txt
otus@PCLin:/test$ nano file.txt
otus@PCLin:/test$ cat file.txt
12345
54321
otus@PCLin:/test$
```

5. Установите значение **umask 077**, используя **символьный формат**. Проверьте работоспособность использованной команды. Затем выведите значение umask в восьмеричной и символьной форме.

Действия:



Вот такие команды:

Т.к umask есть функция XOR, от базовых прав, используемых в символьном формате, значит umask 077 можно представить следующим способом:

```
$ umask u=rwx,g=,o=
$ touch file1
$ ls -l file1
```

приведут к такому результату:

```
-rw----- 1 ots ots 0 сен 25 00:30 file1
```

Что же касается директории, то там права выставляются таким образом:

```
drwx----- 2 ots ots 4096 сен 25 00:31 ./
```

6. В данном уроке была продемонстрирована методика использования утилит **chattr** и **lsattr** для управления дополнительными атрибутами файлов, позволяющими предотвратить случайную или преднамеренную модификацию последних. *Помните о том, что вы не можете полагаться на утилиту chattr как на инструмент дополнительной защиты системы, так как снятие соответствующих атрибутов с файлов не будет представлять каких-либо сложностей для злоумышленников. Однако есть один надежный способ защититься от этой уязвимости, укажите его.*

После назначения атрибута защиты файла или директории **chattr -i <имя>**, один из способов решения этой проблемы – ограничение доступа к самой утилите **chattr**.

Некоторые источники в интернете (как, к примеру, этот: <https://losst.ru/neizmenyaemye-fajly-v-linux>) так же рекомендуют как другой вариант решения – **отключение функции ядра CAP_LINUX_IMMUTABLE, но в моём понимании, это влечёт серьёзные изменения базовых механизмов, предусмотренных разработчиком «из коробки», поэтому детальнее остановимся на первом из вариантов:**

```
usr@PCLin:~$ whereis chattr
chattr: /usr/bin/chattr /usr/share/man/man1/chattr.1.gz
usr@PCLin:~$ ll /usr/bin/chattr /usr/share/man/man1/chattr.1.gz
-rwxr-xr-x 1 root root 14336 янв 25 2019 /usr/bin/chattr*
-rw-r--r-- 1 root root 2825 янв 25 2019 /usr/share/man/man1/chattr.1.gz
usr@PCLin:~$ sudo chmod ugo=,a= /usr/bin/chattr
usr@PCLin:~$ ll /usr/bin/chattr
----- 1 root root 14336 янв 25 2019 /usr/bin/chattr
```

7. Работая с администраторами, которые недавно познакомились с атрибутом SUID, вы заметили сценарий ниже - объясните что он делает и в чем здесь могут быть опасности:

```
% ls change-pass
-rwsr-x--- 1 root helpdesk
37 Feb 26 16:35 change-pass
```

```
% cat change-pass
#!/bin/csh -b
set user = $1
passwd $user
```

Опасность, как мне видится, состоит в том, что на helpdesk, имеющий в своей группе бит исполнения, будут делегированы полномочия владельца файла сценария — root, любой локальный или удалённый пользователь сможет использовать такой файл и при выполнении данного скрипта с подменой переменной \$user на произвольное выражение, можно лишить себя возможности корректного прохождения процедуры аутентификации в ОС.

```
% cat change-pass
#!/bin/csh -b
set user = $1
passwd $user
```

8. Перепишите данный скрипт для безопасного использования. Какие рекомендации вы могли вы дать?

Думаю, наиболее рациональный способ переписать данный скрипт – это изменить набор ключей csh таким образом, чтобы наш скрипт не выполнился.

```
% cat change-pass
#!/bin/csh -d
set user = $1
passwd $user
```