

Отчёт по лабораторной работе №5

**Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов**

Ханина Ирина Владимировна, НБИбд-02-18

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	17
	Список литературы	18

List of Tables

List of Figures

4.1	Рис 1. Подготовка лабораторного стенда	9
4.2	Рис 2. Создание файла simpleid.c	10
4.3	Рис 3. Выполнение программ simpleid и id	10
4.4	Рис 4. Создание файла simpleid2.c	10
4.5	Рис 5. Выполнение программы simpleid2	11
4.6	Рис 6. Установление SetUID-бита	11
4.7	Рис 7. Установление SetGID-бита	11
4.8	Рис 8. Создание файла readfile.c	12
4.9	Рис 9. Компиляция readfile.c	12
4.10	Рис 10. Смена владельцев файлов readfile.c и readfile, установление SetU'D-бита	13
4.11	Рис 11. Проверка чтения файла readfile.c программой readfile . . .	13
4.12	Рис 12. Проверка чтения файла /etc/shadow программой readfile .	14
4.13	Рис 13. Создание файла file01.txt и выполнение различных команд от пользователя guest2	15
4.14	Рис 14. Попытка удаления файла файл от имени пользователя, не являющегося его владельцем	16
4.15	Рис 15. Возвращение атрибута t на директорию /tmp	16

1 Цель работы

Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Задание

Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

3 Теоретическое введение

Чтобы получить доступ к файлам и директориям в Linux, используются разрешения. Эти разрешения назначаются трем объектам: владельцу, группе и остальным пользователям. При создании файла или директории тот пользователь, от имени которого был создан файл или директория, становится его владельцем, а группой устанавливается основная группа владельца. Но владельца файла и группу можно менять, для этого используются команды `chown`.

Система разрешений Linux была изобретена в 1970-х годах. Поскольку вычислительные потребности были ограничены в те годы, базовая система разрешений была довольно ограничена. Эта система разрешений использует три разрешения, которые можно применять к файлам и каталогам:

- `r` - разрешение на чтение;
- `w` - разрешение на запись;
- `x` - разрешение на выполнение. [1]

Помимо основных разрешений в Linux также есть набор расширенных разрешений:

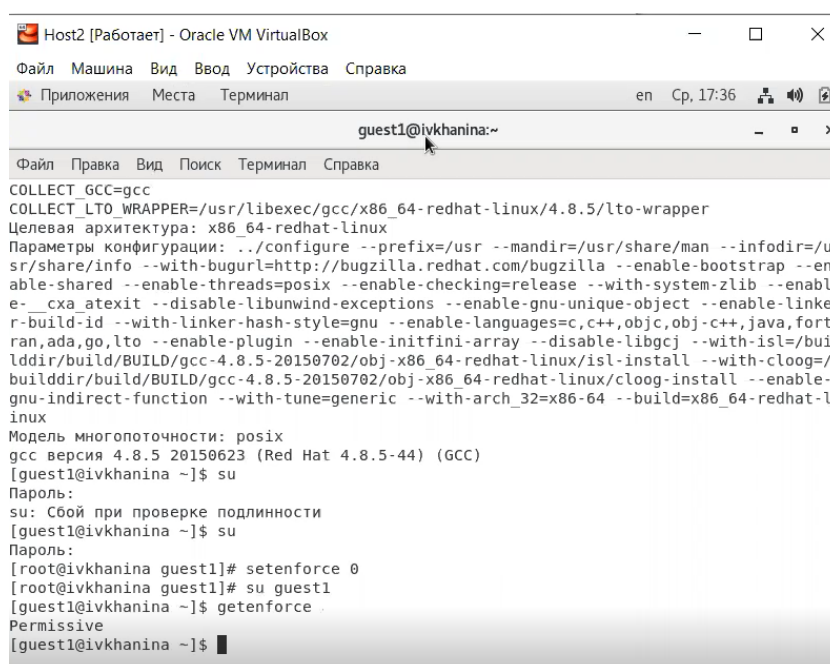
- SUID - разрешение на установку идентификатора пользователя. Это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо.

- SGID - разрешение на установку идентификатора группы. Принцип работы очень похож на SUID с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.
- Sticky bit - в случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Пример использования этого бита в операционной системе - это системная папка /tmp . Эта папка разрешена на запись любому пользователю, но удалять файлы в ней могут только пользователи, являющиеся владельцами этих файлов. Когда вы применяете sticky bit, пользователь может удалять файлы, только если выполняется одно из следующих условий: пользователь является владельцем файл или пользователь является владельцем каталога, в котором находится файл. [2]

Чтобы применить SUID, SGID и sticky bit, можно использовать команду `chmod`. Для SUID используйте `chmod u+s`. Для SGID используйте `chmod g+s`. Для sticky bit используйте `chmod +t`, а затем имя файла или каталога, для которого вы хотите установить разрешения. [1]

4 Выполнение лабораторной работы

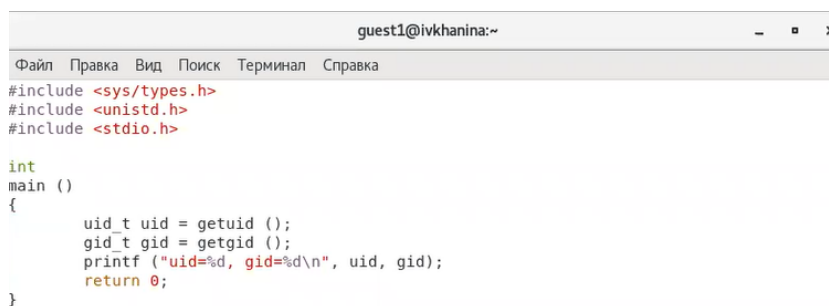
1. Выполнила подготовку лабораторного стенда. (рис. 1)



```
Host2 [Работаer] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Приложения  Места  Терминал  en  Cp, 17:36
guest1@ivkhanina:~
Файл  Правка  Вид  Поиск  Терминал  Справка
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/u
sr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --en
able-shared --enable-threads=posix --enable-checking=release --with-system-zlib --enabl
e_cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker
r-build-id --with-linker-hash-style=gnu --enable-languages=c,c++,objc,obj-c++,java,fort
ran,ada,go,lto --enable-plugin --enable-initfini-array --disable-libgck --with-isl=/bui
ldir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/bui
ldir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-
gnu-indirect-function --with-tune=generic --with-arch_32=x86_64 --build=x86_64-redhat-l
inux
Модель многопоточности: posix
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
[guest1@ivkhanina ~]$ su
Пароль:
su: Сбой при проверке подлинности
[guest1@ivkhanina ~]$ su
Пароль:
[root@ivkhanina guest1]# setenforce 0
[root@ivkhanina guest1]# su guest1
[guest1@ivkhanina ~]$ getenforce
Permissive
[guest1@ivkhanina ~]$
```


Figure 4.1: Рис 1. Подготовка лабораторного стенда

2. Я вошла в систему от имени пользователя guest1 и создала программу simpleid.c. (рис. 2). Скомпилировала программу с помощью команды gcc simpleid.c -o simpleid и убедилась, что файл программы был создан. Выполнила программу simpleid. Затем я выполнила системную программу id и сравнила полученный результат: вывод uid и gid одинаковый. (рис. 3)



```
guest1@ivkhanina:~  
Файл Правка Вид Поиск Терминал Справка  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main ()  
{  
    uid_t uid = getuid ();  
    gid_t gid = getgid ();  
    printf ("uid=%d, gid=%d\n", uid, gid);  
    return 0;  
}
```

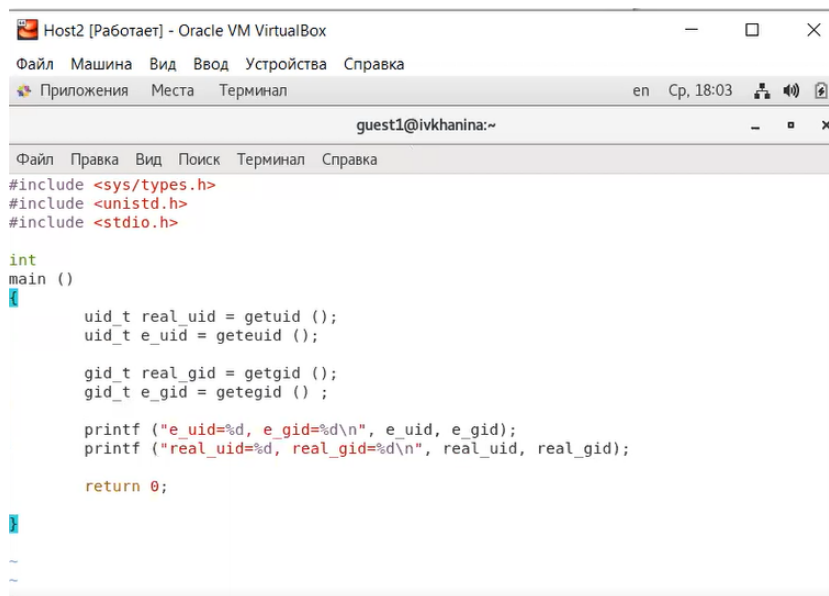
Figure 4.2: Рис 2. Создание файла simpleid.c



```
[guest1@ivkhanina ~]$ vi simpleid.c  
[guest1@ivkhanina ~]$ gcc simpleid.c -o simpleid  
[guest1@ivkhanina ~]$ ./simpleid  
uid=1002, gid=1002  
[guest1@ivkhanina ~]$ id  
uid=1002(guest1) gid=1002(guest1) группы=1002(guest1) контекст=unconfined_u:unconfined_  
r:unconfined_t:s0-s0:c0.c1023  
[guest1@ivkhanina ~]$
```

Figure 4.3: Рис 3. Выполнение программ simpleid и id

3. Усложнила программу, добавив вывод действительных идентификаторов. Получившуюся программу назвала simpleid2.c. (рис. 4). Скомпилировала и запустила simpleid2.c. (рис. 5).



```
Host2 [Работает] - Oracle VM VirtualBox  
Файл Машина Вид Ввод Устройства Справка  
Приложения Места Терминал en Cp, 18:03  
guest1@ivkhanina:~  
Файл Правка Вид Поиск Терминал Справка  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main ()  
{  
    uid_t real_uid = getuid ();  
    uid_t e_uid = geteuid ();  
  
    gid_t real_gid = getgid ();  
    gid_t e_gid = getegid ();  
  
    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);  
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);  
  
    return 0;  
}
```

Figure 4.4: Рис 4. Создание файла simpleid2.c

```
[guest1@ivkhanina ~]$ vi simpleid2.c
[guest1@ivkhanina ~]$ gcc simpleid2.c -o simpleid2
[guest1@ivkhanina ~]$ ./simpleid2
e_uid=1002, e_gid=1002
real_uid=1002, real_gid=1002
```

Figure 4.5: Рис 5. Выполнение программы simpleid2

4. От имени суперпользователя выполнила команды `chown root:guest1 /home/guest1/simpleid2` и `chmod u+s /home/guest1/simpleid2`. Использовала команду `su`, чтобы временно повысить свои права. Выполнила проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`, введя команду `ls -l simpleid2`. Запустила `simpleid2` и `id`, сравнила результаты. Программа `simpleid2` была запущена с правами суперпользователя - владельца файла, хотя действительный `uid` пользователя `guest1` другой. (рис. 6)

```
[guest1@ivkhanina ~]$ su
Пароль:
[root@ivkhanina guest1]# chown root:guest1 /home/guest1/simpleid2
[root@ivkhanina guest1]# chmod u+s /home/guest1/simpleid2
[root@ivkhanina guest1]# su guest1
[guest1@ivkhanina ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest1 8576 ноя  3 18:04 simpleid2
[guest1@ivkhanina ~]$ ./simpleid2
e_uid=0, e_gid=1002
real_uid=1002, real_gid=1002
[guest1@ivkhanina ~]$ id
uid=1002(guest1) gid=1002(guest1) группы=1002(guest1) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest1@ivkhanina ~]$ █
```

Figure 4.6: Рис 6. Установление SetUID-бита

5. Проделала те самые действия относительно SetGID-бита. (рис. 7)

```
[guest1@ivkhanina ~]$ su
Пароль:
[root@ivkhanina guest1]# chown root:guest1 /home/guest1/simpleid2
[root@ivkhanina guest1]# chmod g+s /home/guest1/simpleid2
[root@ivkhanina guest1]# su guest1
[guest1@ivkhanina ~]$ ls -l simpleid2
-rwxrwsr-x. 1 root guest1 8576 ноя  3 18:04 simpleid2
[guest1@ivkhanina ~]$ ./simpleid2
e_uid=1002, e_gid=1002
real_uid=1002, real_gid=1002
[guest1@ivkhanina ~]$ id
uid=1002(guest1) gid=1002(guest1) группы=1002(guest1) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

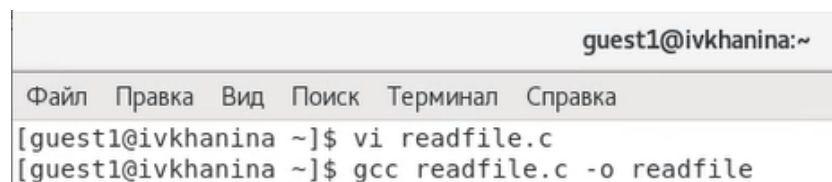
Figure 4.7: Рис 7. Установление SetGID-бита

6. Создала программу readfile.c. (рис. 8). Скомпилировала её при помощи команды gcc readfile.c -o readfile. (рис. 9)



```
guest1@ivkhanina:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
#include <fcntl.h>  
#include <stdio.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
int  
main (int argc, char* argv[])  
{  
    unsigned char buffer[16];  
    size_t bytes_read;  
    int i;  
  
    int fd = open (argv[1], O_RDONLY);  
    do  
    {  
        bytes_read = read (fd, buffer, sizeof (buffer));  
        for (i=0; i < bytes_read; ++i) printf ("%c", buffer[i]);  
    }  
  
    while (bytes_read == sizeof (buffer));  
    close (fd);  
    return 0;  
}  
-- ВСТАВКА --
```

Figure 4.8: Рис 8. Создание файла readfile.c



```
guest1@ivkhanina:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[guest1@ivkhanina ~]$ vi readfile.c  
[guest1@ivkhanina ~]$ gcc readfile.c -o readfile
```

Figure 4.9: Рис 9. Компиляция readfile.c

7. Далее я сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest1 не мог. После этого я проверила, что пользователь guest1 не может прочитать файл readfile.c. Сменила у программы readfile владельца и установила SetU'D-бит. (рис. 10)

```

[guest1@ivkhanina ~]$ su
Пароль:
[root@ivkhanina guest1]# chown root:guest1 /home/guest1/readfile.c
[root@ivkhanina guest1]# chmod 700 /home/guest1/readfile.c
[root@ivkhanina guest1]# su guest1
[guest1@ivkhanina ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest1@ivkhanina ~]$ su
Пароль:
[root@ivkhanina guest1]# chown root:guest1 /home/guest1/readfile
[root@ivkhanina guest1]# chmod u+s /home/guest1/readfile
[root@ivkhanina guest1]# chmod g+s /home/guest1/readfile

```

Figure 4.10: Рис 10. Смена владельцев файлов readfile.c и readfile, установление SetU'D-бита

8. Я проверила, может ли программа readfile прочитать файл readfile.c (рис. 11), а также может ли программа readfile прочитать файл /etc/shadow. (рис. 12) Программа смогла их прочитать.

```

[guest1@ivkhanina ~]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    int fd = open (argv[1], O_RDONLY);
    do
    {
        bytes_read = read (fd, buffer, sizeof (buffer));
        for (i=0; i < bytes_read; ++i) printf ("%c", buffer[i]);
    }
}

```

Figure 4.11: Рис 11. Проверка чтения файла readfile.c программой readfile

```
guest1@ivkhanina:~  
Файл Правка Вид Поиск Терминал Справка  
chorny:!!:18884:~::~:  
unbound:!!:18884:~::~:  
qemu:!!:18884:~::~:  
tss:!!:18884:~::~:  
usbmuxd:!!:18884:~::~:  
geoclue:!!:18884:~::~:  
gluster:!!:18884:~::~:  
gdm:!!:18884:~::~:  
rpcuser:!!:18884:~::~:  
nfsnobody:!!:18884:~::~:  
gnome-initial-setup:!!:18884:~::~:  
sshd:!!:18884:~::~:  
avahi:!!:18884:~::~:  
postfix:!!:18884:~::~:  
ntp:!!:18884:~::~:  
tcpdump:!!:18884:~::~:  
ivkhanina:$6$WZqDlnfPBEUi0v0u$JCil7.kyL1yzZHp0Fgdq.EAGeb0T5glwLgWn7aC1Pb6YEWswycJopkxsQ  
P.9mkZfBn.vcWxiZAKkdv0XZQ73c0:0:99999:7:::  
guest:$6$vpDWjLAN$j4tqTp3M55e.4AtX.MSjhjYC2PYhZ8l0/9FehrX23y0/8uNKNqVmYah.HkkGOQTawWaf  
f0m1lgrAUFO0xkYu1:18896:0:99999:7:::  
guest1:$6$ELo0MD4A$UJIpa2tV6WxpYAR9j8hpc1VIebcUYKiH7ZneBtBkIznZEMrAR02Yn5.0iU3eiEPqaxHL  
7xuk7ibVerabrgx0K.:18896:0:99999:7:::  
guest2:$6$bcdZ6nbw$JYd3B1.p/6ZAUEJaNT5tkz1iamQEICmIYLFUFiXxhWm4N7JPN4MhvexXrK3Zzg7ZilXV  
i2.uB9ViT48mbhRNd.:18905:0:99999:7:::  
[guest1@ivkhanina ~]$
```

Figure 4.12: Рис 12. Проверка чтения файла /etc/shadow программой readfile

9. Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду `ls -l / | grep tmp`. Да, данный атрибут установлен. От имени пользователя guest1 создала файл file01.txt в директории /tmp со словом test. Затем я просмотрела атрибуты у созданного файла и разрешила чтение и запись для категории пользователей «все остальные». От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt, а так же сделать дозапись слова test2 в файл /tmp/file01.txt командой `echo "test2" > /tmp/file01.txt`. Мне удалось выполнить операцию. Далее я распечатала содержимое файла командой `cat /tmp/file01.txt`. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стеревав при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt`. Я смогла это сделать. Затем я проверила содержимое файла командой `cat /tmp/file01.txt`. (рис. 13)

```

[guest1@ivkhanina ~]$ ls -l / | grep tmp
drwxrwxrwt. 28 root root 4096 ноя  3 18:25 tmp
[guest1@ivkhanina ~]$ echo "test" > /tmp/file01.txt
[guest1@ivkhanina ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest1 guest1 5 ноя  3 18:30 /tmp/file01.txt
[guest1@ivkhanina ~]$ chmod o+rw /tmp/file01.txt
[guest1@ivkhanina ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest1 guest1 5 ноя  3 18:30 /tmp/file01.txt
[guest1@ivkhanina ~]$ su guest2
Пароль:
[guest2@ivkhanina guest1]$ pwd
/home/guest1
[guest2@ivkhanina guest1]$ cd /home/guest2
[guest2@ivkhanina ~]$ pwd
/home/guest2
[guest2@ivkhanina ~]$ cat /tmp/file01.txt
test
[guest2@ivkhanina ~]$ echo "test2" > /tmp/file01.txt
[guest2@ivkhanina ~]$ cat /tmp/file01.txt
test2
[guest2@ivkhanina ~]$ echo "test3" > /tmp/file01.txt
[guest2@ivkhanina ~]$ cat /tmp/file01.txt
test3

```

Figure 4.13: Рис 13. Создание файла file01.txt и выполнение различных команд от пользователя guest2

10. От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt`. Я не смогла удалить файл. Повысила свои права до суперпользователя, введя команду `su -`, и выполните после этого команду `chmod -t /tmp`. Покинула режим суперпользователя командой `exit`, после чего от пользователя guest2 проверила, что атрибута `t` у директории `/tmp` нет. Повторила предыдущие шаги. Мне удалось удалить файл от имени пользователя, не являющегося его владельцем. (рис. 14).

```
root@ivkhanina:~  
Файл Правка Вид Поиск Терминал Справка  
[guest2@ivkhanina ~]$ rm /tmp/file01.txt  
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена  
[guest2@ivkhanina ~]$ su -  
Пароль:  
Последний вход в систему: Ср ноя  3 18:25:02 MSK 2021 на pts/0  
[root@ivkhanina ~]# chmod -t /tmp  
[root@ivkhanina ~]# exit  
logout  
[guest2@ivkhanina ~]$ pwd  
/home/guest2  
[guest2@ivkhanina ~]$ ls -l / | grep tmp  
drwxrwxrwx. 28 root root 4096 ноя  3 18:34 tmp  
[guest2@ivkhanina ~]$ cat /tmp/file01.txt  
test3  
[guest2@ivkhanina ~]$ echo "test2" > /tmp/file01.txt  
[guest2@ivkhanina ~]$ cat /tmp/file01.txt  
test2  
[guest2@ivkhanina ~]$ echo "test3" > /tmp/file01.txt  
[guest2@ivkhanina ~]$ cat /tmp/file01.txt  
test3  
[guest2@ivkhanina ~]$ rm /tmp/file01.txt  
[guest2@ivkhanina ~]$ su -  
Пароль:  
Последний вход в систему: Ср ноя  3 18:34:11 MSK 2021 на pts/0
```

Figure 4.14: Рис 14. Попытка удаления файла файл от имени пользователя, не являющегося его владельцем

12. Я вновь повысила свои права до суперпользователя и вернула атрибут `t` на директорию `/tmp`. (рис. 15)

```
[guest2@ivkhanina ~]$ su -  
Пароль:  
Последний вход в систему: Ср ноя  3 18:34:11 MSK 2021 на pts/0  
[root@ivkhanina ~]# chmod +t /tmp  
[root@ivkhanina ~]# exit  
logout  
[guest2@ivkhanina ~]$ █
```

Figure 4.15: Рис 15. Возвращение атрибута `t` на директорию `/tmp`

5 Выводы

В результате выполнения лабораторной работы я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, получила практические навыки работы в консоли с дополнительными атрибутами, рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

1. Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask)
2. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux