

# **Отчет по лабораторной работе №5**

**по предмету Информационная безопасность**

Алхимова Дарья Сергеевна

# Содержание

<b>Цель работы</b>	<b>4</b>
<b>Задание</b>	<b>5</b>
Часть 1 . . . . .	5
Часть 2 . . . . .	8
<b>Теоретическое введение</b>	<b>10</b>
<b>Выполнение лабораторной работы</b>	<b>12</b>
Часть 1 . . . . .	12
Часть 2 . . . . .	18
<b>Выводы</b>	<b>23</b>
<b>Список литературы</b>	<b>24</b>

## Список иллюстраций

1	Создание файла программы . . . . .	12
2	Проверка наличия файла . . . . .	13
3	Сравнения вывода программ simpleid и id . . . . .	13
4	Добавление в программу действительных идентификаторов . .	14
5	Компилирование и запуск программы simpleid2 . . . . .	14
6	Проверка изменений файла . . . . .	15
7	Сравнения вывода программ simpleid2 и id . . . . .	15
8	Повтор операций для SetGID-бита . . . . .	15
9	Проверка доступа guest к readfile . . . . .	17
10	Чтение readfile readfile.c . . . . .	18
11	Чтение readfile /etc/shadow . . . . .	18
12	Проверка атрибута Sticky . . . . .	19
13	Чтение file01.txt пользователем guest2 . . . . .	19
14	Дозапись file01.txt пользователем guest2 . . . . .	20
15	Дозапись с удалением file01.txt пользователем guest2 . . . . .	20
16	Удаление file01.txt пользователем guest2 . . . . .	20
17	Снятие атрибута Sticky . . . . .	21
18	Возвращение атрибута Sticky на директорию /tmp . . . . .	22

## Цель работы

Целью данной работы является 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.

# Задание

## Часть 1

1. Войдите в систему от имени пользователя guest.

2. Создайте программу simpleid.c:

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

3. Скомпилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid

4. Выполните программу simpleid: ./simpleid

5. Выполните системную программу id (id) и сравните полученный вами результат с данными предыдущего пункта задания.

6. Усложните программу, добавив вывод действительных идентификаторов:

```
#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;
}
```

Получившуюся программу назовите simpleid2.c.

7. Скомпилируйте и запустите simpleid2.c:
- ```
gcc simpleid2.c -o simpleid2
./simpleid2
```
8. От имени суперпользователя выполните команды: `chown root:guest /home/guest/simpleid2` `chmod u+s /home/guest/simpleid2`
9. Используйте `sudo` или повысьте временно свои права с помощью `su`. Поясните, что делают эти команды.
10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2`
11. Запустите simpleid2 и `id`: `./simpleid2` `id` Сравните результаты.
12. Прodelайте тоже самое относительно SetGID-бита.

13. Создайте программу 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]);
    }
    while (bytes_read == sizeof (buffer));
    close (fd);
    return 0;
}
```

14. Откомпилируйте её: gcc readfile.c -o readfile.

15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

16. Проверьте, что пользователь guest не может прочитать файл readfile.c.

17. Смените у программы readfile владельца и установите SetU'D-бит.

18. Проверьте, может ли программа readfile прочитать файл readfile.c?
19. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Отразите полученный результат и ваши объяснения в отчёте.

## Часть 2

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду `ls -l / | grep tmp`
2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt`
3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt`  
`chmod o+rw /tmp/file01.txt`  
`ls -l /tmp/file01.txt`
4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: `cat /tmp/file01.txt`
5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой `echo "test2" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
6. Проверьте содержимое файла командой `cat /tmp/file01.txt`
7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
8. Проверьте содержимое файла командой `cat /tmp/file01.txt`
9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt` Удалось ли вам удалить файл?



10. Повысьте свои права до суперпользователя командой `su -` и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`:  
`chmod -t /tmp`
11. Покиньте режим суперпользователя командой `exit`
12. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет:  
`ls -l / | grep tmp`
13. Повторите предыдущие шаги. Какие наблюдаются изменения?
14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.
15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`:

```
su -  
chmod +t /tmp  
exit
```

# Теоретическое введение

Компиляторы, доступные в Linux-системах, являются частью коллекции GNU-компиляторов, известной как GCC (GNU Compiler Collection, подробнее см. <http://gcc.gnu.org>). В неё входят компиляторы языков C, C++, Java, Objective-C, Fortran и Chill. Будем использовать лишь первые два. Компилятор языка C называется gcc. Компилятор языка C++ называется g++ и запускается с параметрами почти так же, как gcc. Проверить это можно следующими командами:

```
whereis gcc
```

```
whereis g++
```

Первый шаг заключается в превращении исходных файлов в объектный код: `gcc -c file.c` В случае успешного выполнения команды (отсутствие ошибок в коде) полученный объектный файл будет называться `file.o`. Объектные файлы невозможно запускать и использовать, поэтому после компиляции для получения готовой программы объектные файлы необходимо скомпоновать. Компоновать можно один или несколько файлов. В случае использования хотя бы одного из файлов, написанных на C++, компоновка производится с помощью компилятора g++. Строго говоря, это тоже не вполне верно. Компоновка объектного кода, сгенерированного чем бы то ни было (хоть вручную), производится линкером ld, g++ его просто вызывает изнутри. Если же все файлы написаны на языке C, нужно использовать компилятор gcc. Например, так: `gcc -o program file.o` В

случае успешного выполнения команды будет создана программа `program` (исполняемый файл формата ELF с установленным атрибутом `+x`).

# Выполнение лабораторной работы

## Часть 1

1. Вошла в систему от имени пользователя guest и создала программу simpleid.c. (рис. 1). С помощью команды nano открыла файл для редактирования и внесла туда код из задания:

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

```
[guest1@dsalkhimova ~]$ touch simpleid.c
[guest1@dsalkhimova ~]$ ls
simpleid.c
[guest1@dsalkhimova ~]$ nano simpleid.c
```

Рис. 1: Создание файла программы

2. Скомпилировала программу и убедилась, что файл программы создан:

gcc simpleid.c -o simpleid (рис. 2).

```
[guest1@dsalkhimova ~]$ gcc simpleid.c -o simpleid
[guest1@dsalkhimova ~]$ ls
simpleid  simpleid.c
```

Рис. 2: Проверка наличия файла

3. Выполнила программу simpleid (./simpleid) и системную программу id (id). Вывод программ сходится - они передают одинаковые параметры пользователя и его группы. (рис. 3).

```
[guest1@dsalkhimova ~]$ ./simpleid
uid=1001, gid=1001
[guest1@dsalkhimova ~]$ id
uid=1001(guest1) gid=1001(guest1) groups=1001(guest1) context=unconfined u:unconfined r:unconfined t:s0-s0:c0.c1023
```

Рис. 3: Сравнения вывода программ simpleid и id

4. Усложнила программу, добавив вывод действительных идентификаторов:

```
#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;
} Получившуюся программу назвала simpleid2.c. (рис. 4).
```

```
[guest1@dsalkhimova ~]$ touch simpleid2.c
[guest1@dsalkhimova ~]$ ls
simpleid  simpleid2.c  simpleid.c
```

Рис. 4: Добавление в программу действительных идентификаторов

5. Скомпилировала и запустила simpleid2.c (рис. 5): `gcc simpleid2.c -o simpleid2 ./simpleid2`

```
[guest1@dsalkhimova ~]$ gcc simpleid2.c -o simpleid2
[guest1@dsalkhimova ~]$ ls
simpleid  simpleid2  simpleid2.c  simpleid.c
[guest1@dsalkhimova ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 5: Компилирование и запуск программы simpleid2

6. От имени суперпользователя выполнила команды:

`chown root:guest /home/guest/simpleid2` (рис. 6) и `chmod u+s /home/guest/simpleid2` (рис. 7)

```
[root@dsalkhimova guest1]# chown root:guest1 /home/guest1/simpleid2
[root@dsalkhimova guest1]# ls
simpleid  simpleid2  simpleid2.c  simpleid.c
[root@dsalkhimova guest1]# ls -l
total 32
-rwxrwxr-x. 1 guest1 guest1 8512 Oct  4 14:22 simpleid
-rwxrwxr-x. 1 root   guest1 8616 Oct  4 14:28 simpleid2
-rw-rw-r--. 1 guest1 guest1  335 Oct  4 14:28 simpleid2.c
-rw-rw-r--. 1 guest1 guest1  177 Oct  4 14:22 simpleid.c
[root@dsalkhimova guest1]# chmod u+s /home/guest1/simpleid2
[root@dsalkhimova guest1]# ls -l
total 32
-rwxrwxr-x. 1 guest1 guest1 8512 Oct  4 14:22 simpleid
-rwsrwxr-x. 1 root   guest1 8616 Oct  4 14:28 simpleid2
-rw-rw-r--. 1 guest1 guest1  335 Oct  4 14:28 simpleid2.c
-rw-rw-r--. 1 guest1 guest1  177 Oct  4 14:22 simpleid.c
```

7. Повысила временно свои права с помощью `su`.

- Самый частый пример использования `sudo` - выполнение программы от имени суперпользователя. Для этого достаточно написать `sudo` перед именем программы.

- Если вызов команды su происходит без аргументов, то происходит смена пользователя оболочки shell на суперпользователя root. Программа выдаст приглашение ввода пароля, если пароль будет верным, то текущим пользователем станет root.

8. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2` (рис. 8)

```
[root@dsalkhimova guest1]# ls -l simpleid2
-rwsrwxr-x. 1 root guest1 8616 Oct  4 14:28 simpleid2
```

Рис. 6: Проверка изменений файла

9. Запустила simpleid2 (`./simpleid2`) и id (`id`) (рис. 9).

```
[root@dsalkhimova guest1]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@dsalkhimova guest1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 7: Сравнения вывода программ simpleid2 и id

Результаты вывода сходятся.

10. Проделала то же самое относительно SetGID-бита.(рис. 10).

```
[root@dsalkhimova guest1]# chmod g+s /home/guest1/simpleid2
[root@dsalkhimova guest1]# ls -l
total 32
-rwxrwxr-x. 1 guest1 guest1 8512 Oct  4 14:22 simpleid
-rwsrwsr-x. 1 root    guest1 8616 Oct  4 14:28 simpleid2
-rw-rw-r--. 1 guest1 guest1  335 Oct  4 14:28 simpleid2.c
-rw-rw-r--. 1 guest1 guest1  177 Oct  4 14:22 simpleid.c
[root@dsalkhimova guest1]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@dsalkhimova guest1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 8: Повтор операций для SetGID-бита

11. Создала и скомпилировала программу readfile.c (рис. 11): #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;

    [root@dsalkhimova guest1]# touch readfile.c
    [root@dsalkhimova guest1]# nano readfile.c
    [root@dsalkhimova guest1]# gcc readfile.c -o readfile
    [root@dsalkhimova guest1]# ls
    readfile  simpleid  simpleid2.c
    readfile.c simpleid2 simpleid.c
}
```

12. Сменила владельца у файла readfile.c на dsalkhimova (рис. 12) и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. 13).



```

[root@dsalkhimova guest1]# chown dsalkhimova readfile.c
[root@dsalkhimova guest1]# ls -l
total 48
-rwxr-xr-x. 1 root      root      8552 Oct  4 14:50 readfile
-rw-r--r--. 1 dsalkhimova root      395 Oct  4 14:49 readfile.c
-rwxrwxr-x. 1 guest1    guest1    8512 Oct  4 14:22 simpleid
-rwsrwsr-x. 1 root      guest1    8616 Oct  4 14:28 simpleid2
-rw-rw-r--. 1 guest1    guest1     335 Oct  4 14:28 simpleid2.c
-rw-rw-r--. 1 guest1    guest1     177 Oct  4 14:22 simpleid.c
[root@dsalkhimova guest1]# chmod ugo-r readfile.c
[root@dsalkhimova guest1]# ls -l
total 48
-rwxr-xr-x. 1 root      root      8552 Oct  4 14:50 readfile
--w-----. 1 dsalkhimova root      395 Oct  4 14:49 readfile.c
-rwxrwxr-x. 1 guest1    guest1    8512 Oct  4 14:22 simpleid
-rwsrwsr-x. 1 root      guest1    8616 Oct  4 14:28 simpleid2
-rw-rw-r--. 1 guest1    guest1     335 Oct  4 14:28 simpleid2.c
-rw-rw-r--. 1 guest1    guest1     177 Oct  4 14:22 simpleid.c

```

13. Проверила, что пользователь guest не может прочитать файл readfile.c (рис. 14).

```

[root@dsalkhimova guest1]# su guest1
[guest1@dsalkhimova ~]$ cat readfile.c
cat: readfile.c: Permission denied

```

Рис. 9: Проверка доступа guest к readfile

14. Сменила у программы readfile владельца (рис. 15) и установила SetU'D-бит (рис. 16).

```

[guest1@dsalkhimova ~]$ su
Password:
[root@dsalkhimova guest1]# chown dsalkhimova readfile
[root@dsalkhimova guest1]# ls -l
total 48
-rwxr-xr-x. 1 dsalkhimova root      8552 Oct  4 14:50 readfile
--w-----. 1 dsalkhimova root      395 Oct  4 14:49 readfile.c
-rwxrwxr-x. 1 guest1    guest1    8512 Oct  4 14:22 simpleid
-rwsrwsr-x. 1 root      guest1    8616 Oct  4 14:28 simpleid2
-rw-rw-r--. 1 guest1    guest1     335 Oct  4 14:28 simpleid2.c
-rw-rw-r--. 1 guest1    guest1     177 Oct  4 14:22 simpleid.c
[root@dsalkhimova guest1]# chmod u+s readfile
[root@dsalkhimova guest1]# ls -l
total 48
-rwsr-xr-x. 1 dsalkhimova root      8552 Oct  4 14:50 readfile
--w-----. 1 dsalkhimova root      395 Oct  4 14:49 readfile.c
-rwxrwxr-x. 1 guest1    guest1    8512 Oct  4 14:22 simpleid
-rwsrwsr-x. 1 root      guest1    8616 Oct  4 14:28 simpleid2
-rw-rw-r--. 1 guest1    guest1     335 Oct  4 14:28 simpleid2.c
-rw-rw-r--. 1 guest1    guest1     177 Oct  4 14:22 simpleid.c

```

15. Проверила, что программа readfile может прочитать файл readfile.c (рис.

17):

[illegible]

Рис. 10: Чтение readfile readfile.c

16. Проверила, что программа readfile может прочитать файл /etc/shadow (рис.

18):

[illegible]

Рис. 11: Чтение readfile /etc/shadow

## Часть 2

1. Выяснила, установлен ли атрибут Sticky на директории /tmp: `ls -l / | grep tmp` (рис. 19). Атрибут `t` установлен.

```
[root@dsalkhimova guest1]# ls -l / | grep tmp
drwxrwxrwt.  25 root root 4096 Oct  4 14:55 tmp
```

Рис. 12: Проверка атрибута Sticky

- От имени пользователя guest создала файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt` (рис. 20, рис. 21)

```
[guest1@dsalkhimova ~]$ echo "test" > /tmp/file01.txt
[guest1@dsalkhimova ~]$ ls
readfile      simpleid      simpleid2.c
readfile.c    simpleid2     simpleid.c
[guest1@dsalkhimova ~]$ ls /tmp
file01.txt
ssh-d2U5j0i1Kzws
ssh-DvB4KFnNc8kz
[guest1@dsalkhimova ~]$ cat /tmp/file01.txt
test
```

- Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» (рис. 22):

```
ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt
```

! [Просмотр и изменение атрибутов file01.txt](images5/22.png){ #fig:022 width=70% }

- От пользователя guest2 (не являющегося владельцем) успешно прочитала файл /tmp/file01.txt: `cat /tmp/file01.txt` (рис. 23):

```
[root@dsalkhimova guest1]# su guest2
[guest2@dsalkhimova guest1]$ cd
[guest2@dsalkhimova ~]$ cat /tmp/file01.txt
test
```

Рис. 13: Чтение file01.txt пользователем guest2

- От пользователя guest2 успешно дозаписала в файл /tmp/file01.txt слово test2 командой `echo "test2" >> /tmp/file01.txt` и проверила содержимое файла командой `cat /tmp/file01.txt` (рис. 24):

```
[guest2@dsalkhimova ~]$ echo "test2" >> /tmp/file01.txt
[guest2@dsalkhimova ~]$ cat /tmp/file01.txt
test
test2
```

Рис. 14: Дозапись file01.txt пользователем guest2

6. От пользователя guest2 успешно дозаписала в файл /tmp/file01.txt слово test3 командой `echo "test3" > /tmp/file01.txt`, стерев при этом все предыдущее содержимое файла, и проверила содержимое файла командой `cat /tmp/file01.txt` (рис. 25):

```
[guest2@dsalkhimova ~]$ echo "test3" > /tmp/file01.txt
[guest2@dsalkhimova ~]$ cat /tmp/file01.txt
test3
```

Рис. 15: Дозапись с удалением file01.txt пользователем guest2

7. От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt` (рис. 26). Операция запрещена.

```
[guest2@dsalkhimova ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Рис. 16: Удаление file01.txt пользователем guest2

8. Повысила свои права до суперпользователя командой `su -` и выполнила после этого команду, снимающую атрибут `t` (Sticky-бит) с директории /tmp: `chmod -t /tmp`. Затем покинула режим суперпользователя командой `exit`. От пользователя guest2 проверила, что атрибута `t` у директории /tmp нет: `ls -l / | grep tmp` (рис. 27):

```

[guest2@dsalkhimova ~]$ su -
Password:
Last login: Tue Oct  4 15:07:38 MSK 2022 on pts/0
[root@dsalkhimova ~]# chmod -t /tmp
[root@dsalkhimova ~]# exit
logout
[guest2@dsalkhimova ~]$ ls -l / | grep tmp
drwxrwxrwx. 25 root root 4096 Oct  4 15:11 tmp

```

Рис. 17: Снятие атрибута Sticky

9. Повторила предыдущие шаги (рис. 28, рис. 29). Изменений нет, кроме того, что теперь удалось удалить файл file01.txt от имени guest2.

```

[guest2@dsalkhimova ~]$ cat /tmp/file01.txt
test3
[guest2@dsalkhimova ~]$ echo "test2" >> /tmp/file01.txt
[guest2@dsalkhimova ~]$ cat /tmp/file01.txt
test3
test2
[guest2@dsalkhimova ~]$ echo "test" > /tmp/file01.txt
[guest2@dsalkhimova ~]$ cat /tmp/file01.txt
test
[guest2@dsalkhimova ~]$ rm /tmp/file01.txt
[guest2@dsalkhimova ~]$ ls -l
total 0
[guest2@dsalkhimova ~]$ ls -l /tmp
total 564
drwx-----. 2 guest2      guest2      25 Oct  4 12:25 ssh-
5j0i1Kzws
drwx-----. 2 dsalkhimova dsalkhimova 24 Oct  4 13:34 ssh-
4KFnNc8kz
drwx-----. 2 guest1      guest1      25 Oct  4 12:37 ssh-
FUKo0eJoL
drwx-----. 3 root        root        17 Oct  4 13:32 syst

```

10. Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp (рис. 30):

```
[guest2@dsalkhimova ~]$ su -  
Password:  
Last login: Tue Oct  4 15:17:21 MSK 2022 on pts/0  
[root@dsalkhimova ~]# chmod +t /tmp  
[root@dsalkhimova ~]# ls -l / | grep tmp  
drwxrwxrwt. 25 root root 4096 Oct  4 15:20 tmp  
[root@dsalkhimova ~]# exit  
logout  
[guest2@dsalkhimova ~]$ █
```

Рис. 18: Возвращение атрибута Sticky на директорию /tmp

## Выводы

В процессе выполнения данной лабораторной работы я научилась работать с механизмами изменения идентификаторов, применением SetUID- и Sticky-битов. Получила практических навыков 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. Описание лабораторной работы 5 - URL: [https://esystem.rudn.ru/pluginfile.php/1652171/mod\\_resource/content/2/005-lab\\_discret\\_sticky.pdf](https://esystem.rudn.ru/pluginfile.php/1652171/mod_resource/content/2/005-lab_discret_sticky.pdf)
2. Команда SU в LINUX — URL: <https://losst.ru/komanda-su-v-linux?ysclid=l8u8lr4lj5312561951>
3. Команда SUDO в LINUX — URL: <https://losst.ru/komanda-sudo-v-linux?ysclid=l8uachqd98171389830>