

Лабораторная Работа №5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Операционные системы

Барсегян В.Л.

Российский университет дружбы народов им. Патриса Лумумбы, Москва, Россия

- Барсегян Вардан Левонович
- НПИбд-01-22
- Российский университет дружбы народов
- [1132222005@pfur.ru]
- <https://github.com/VARdamn/oib>

Вводная часть

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.

Выполнение лабораторной работы

Создаю файл simpleid.c

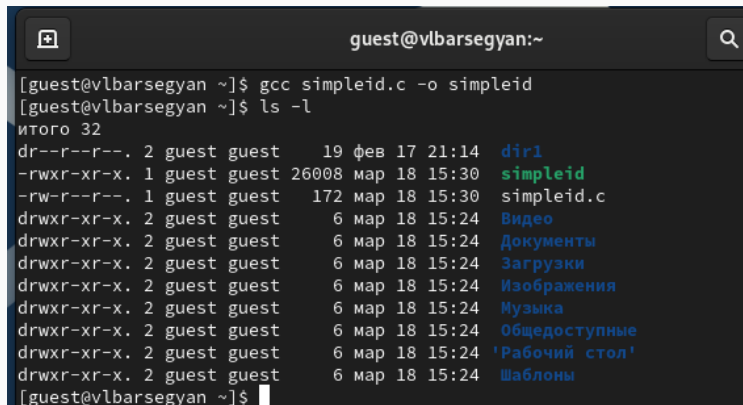


```
GNU nano 5.6.1 simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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

Figure 1: Программа simpleid.c

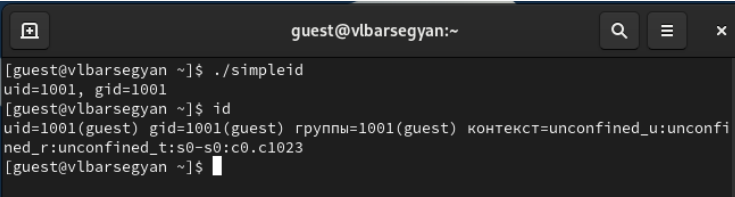
Компилирую программу командой `gcc simpleid.c -o simpleid` и проверяю, что файл создан



```
guest@vlbarsegyan:~  
[guest@vlbarsegyan ~]$ gcc simpleid.c -o simpleid  
[guest@vlbarsegyan ~]$ ls -l  
итого 32  
dr--r--r--. 2 guest guest    19 фев 17 21:14 dir1  
-rwxr-xr-x. 1 guest guest 26008 мар 18 15:30 simpleid  
-rw-r--r--. 1 guest guest   172 мар 18 15:30 simpleid.c  
drwxr-xr-x. 2 guest guest     6 мар 18 15:24 Видео  
drwxr-xr-x. 2 guest guest     6 мар 18 15:24 Документы  
drwxr-xr-x. 2 guest guest     6 мар 18 15:24 Загрузки  
drwxr-xr-x. 2 guest guest     6 мар 18 15:24 Изображения  
drwxr-xr-x. 2 guest guest     6 мар 18 15:24 Музыка  
drwxr-xr-x. 2 guest guest     6 мар 18 15:24 Общедоступные  
drwxr-xr-x. 2 guest guest     6 мар 18 15:24 'Рабочий стол'  
drwxr-xr-x. 2 guest guest     6 мар 18 15:24 Шаблоны  
[guest@vlbarsegyan ~]$
```

Figure 2: Компиляция программы

Выполняю программу `simpleid` командой `./simpleid`, а затем системную программу `id` - вывод одинаков



```
guest@vlbarsegyan:~  
[guest@vlbarsegyan ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@vlbarsegyan ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@vlbarsegyan ~]$
```

Figure 3: Выполнение программы

Усложняю программу и записываю ее в файл simpleid2.c



```
guest@vlbarsegyan:~ — nano simpleid.c
GNU nano 5.6.1                                simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

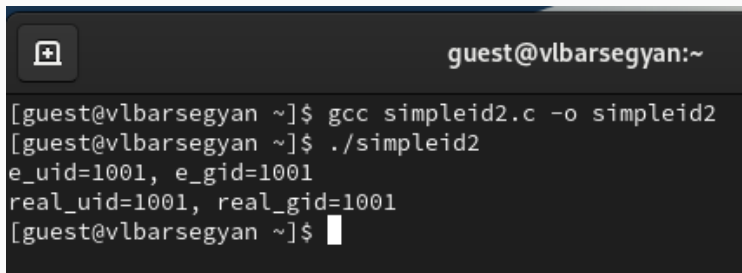
int main() {
    uid_t real_uid = getuid();
    uid_t e_uid = geteuid();

    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: Программа simpleid2.c

Компилирую и запускаю программу командами *gcc simpleid2.c -o simpleid2* и *./simpleid2*

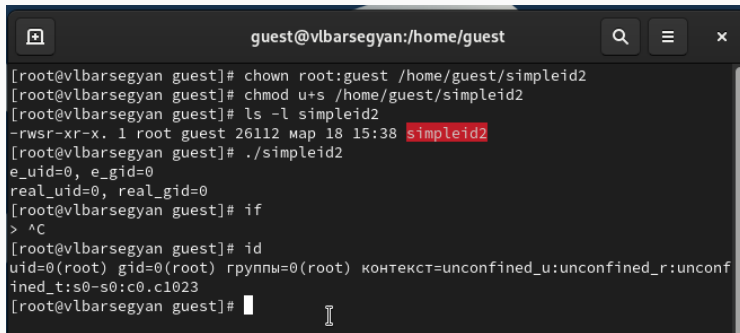


A terminal window with a dark background. The title bar shows a window icon and the text "guest@vlbarsegyan:~". The terminal content shows the following commands and output:

```
[guest@vlbarsegyan ~]$ gcc simpleid2.c -o simpleid2
[guest@vlbarsegyan ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@vlbarsegyan ~]$
```

Figure 5: Компиляция и запуск

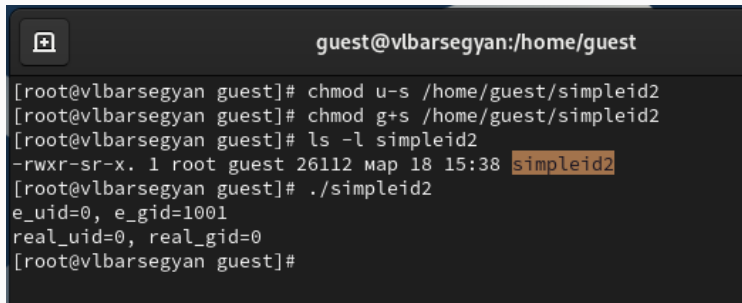
От суперпользователя выполняю команды `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`. Проверяю правильность новых атрибутов командой `ls -l simpleid2`. Запускаю `simpleid2` и `id: ./simpleid2`, `id`



```
guest@vlbarsegyan:/home/guest
[root@vlbarsegyan guest]# chown root:guest /home/guest/simpleid2
[root@vlbarsegyan guest]# chmod u+s /home/guest/simpleid2
[root@vlbarsegyan guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26112 map 18 15:38 simpleid2
[root@vlbarsegyan guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@vlbarsegyan guest]# if
> ^C
[root@vlbarsegyan guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconf
ined_t:s0-s0:c0.c1023
[root@vlbarsegyan guest]#
```

Figure 6: Изменение атрибутов, запуск

Делаю тоже самое относительно SetGID-бита: устанавливаю его командой `chmod g+s /home/guest/simpleid2`, проверяю установку нового атрибута и запускаю `simpleid2` и `id`

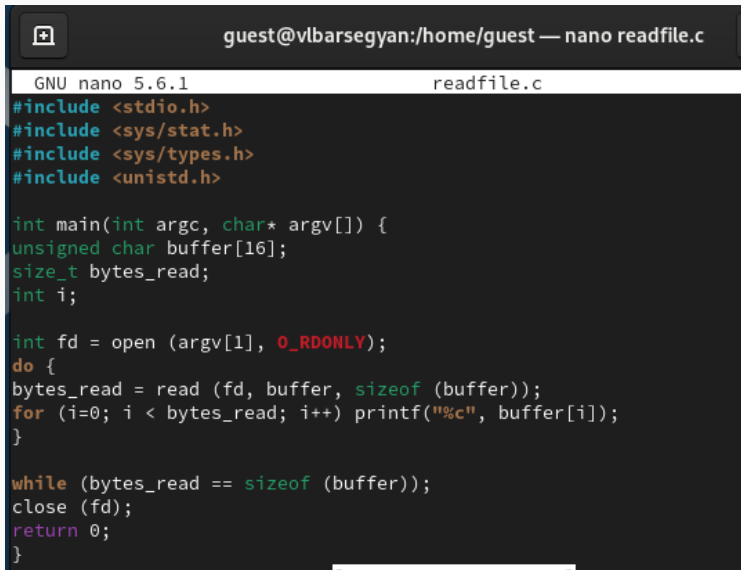


```
guest@vlbarsegyan:/home/guest

[root@vlbarsegyan guest]# chmod u-s /home/guest/simpleid2
[root@vlbarsegyan guest]# chmod g+s /home/guest/simpleid2
[root@vlbarsegyan guest]# ls -l simpleid2
-rwxr-sr-x. 1 root guest 26112 map 18 15:38 simpleid2
[root@vlbarsegyan guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@vlbarsegyan guest]#
```

Figure 7: Изменение SetGID-бита и проверка

Создаю программу readfile.c



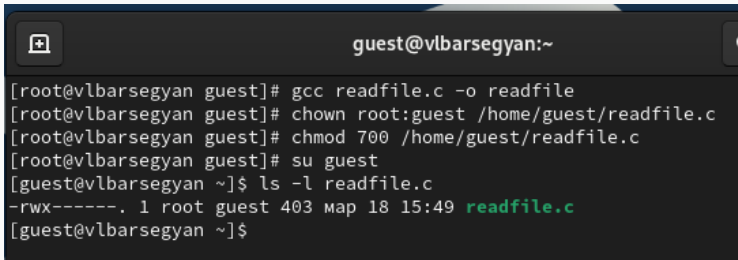
```
guest@vlbarsegyan:/home/guest — nano readfile.c
GNU nano 5.6.1 readfile.c
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

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;
}
```

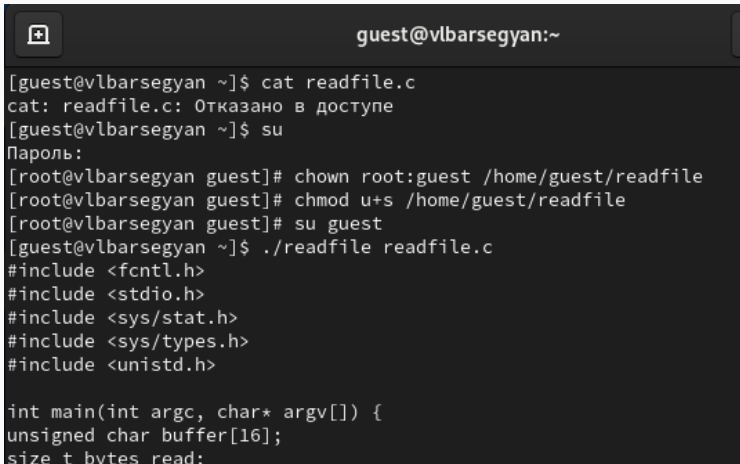
Компилирую ее командой `gcc readfile.c -o readfile` и изменяю права доступа так, чтобы только суперпользователь мог прочитать его, а `guest` не мог

A terminal window titled 'guest@vlbarsegyan:~' with a dark background. It shows a series of commands and their outputs. The commands are: 'gcc readfile.c -o readfile', 'chown root:guest /home/guest/readfile.c', 'chmod 700 /home/guest/readfile.c', and 'su guest'. The output of 'ls -l readfile.c' shows the file permissions as '-rwx-----', owner 'root', group 'guest', size '403', and modification time 'map 18 15:49'. The filename 'readfile.c' is highlighted in green in the output.

```
guest@vlbarsegyan:~  
[root@vlbarsegyan guest]# gcc readfile.c -o readfile  
[root@vlbarsegyan guest]# chown root:guest /home/guest/readfile.c  
[root@vlbarsegyan guest]# chmod 700 /home/guest/readfile.c  
[root@vlbarsegyan guest]# su guest  
[guest@vlbarsegyan ~]$ ls -l readfile.c  
-rwx-----. 1 root guest 403 map 18 15:49 readfile.c  
[guest@vlbarsegyan ~]$
```

Figure 9: Компиляция программы, смена прав доступа

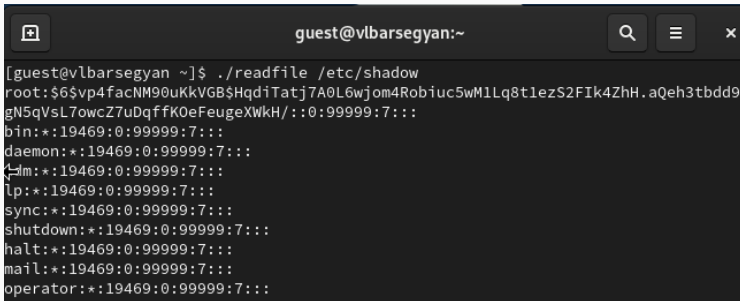
Командой `cat readfile.c` проверяю, что пользователь `guest` не может прочитать файл `readfile.c`. Устанавливаю SetU'D-бит и теперь от пользователя `guest` можно прочитать файл



```
guest@vlbarsegyan:~  
[guest@vlbarsegyan ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@vlbarsegyan ~]$ su  
Пароль:  
[root@vlbarsegyan guest]# chown root:guest /home/guest/readfile  
[root@vlbarsegyan guest]# chmod u+s /home/guest/readfile  
[root@vlbarsegyan guest]# su guest  
[guest@vlbarsegyan ~]$ ./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;
```

Figure 10: Установка SetU'D-бита. проверка

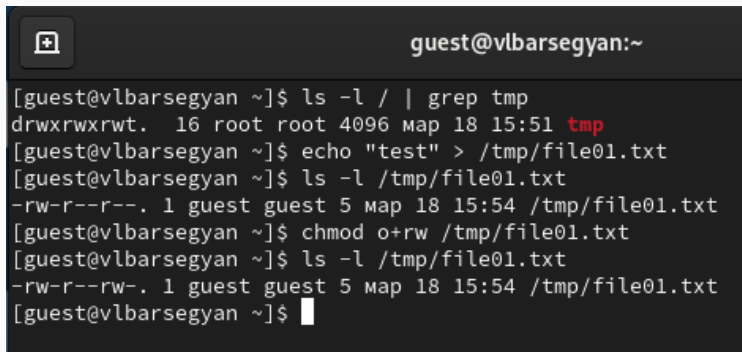
Проверяю, может ли программа readfile прочитать файл /etc/shadow - да, может

A terminal window titled 'guest@vlbarsegyan:~' with search, menu, and close icons in the top right. The command './readfile /etc/shadow' has been executed, displaying the contents of the /etc/shadow file. The output shows the root user's entry with a long hashed password, followed by system users like bin, daemon, lp, sync, shutdown, halt, mail, and operator, all with empty password fields.

```
[guest@vlbarsegyan ~]$ ./readfile /etc/shadow
root:$6$vp4facNM90uKkVGB$HqdiTatj7A0L6wjom4Robiuc5wM1Lq8tlezS2FIk4ZhH.aQeh3tbdd9
gN5qVsL7owcZ7uDqffK0eFeugeXWkH/::0:99999:7:::
bin:*:19469:0:99999:7:::
daemon:*:19469:0:99999:7:::
lp:*:19469:0:99999:7:::
sync:*:19469:0:99999:7:::
shutdown:*:19469:0:99999:7:::
halt:*:19469:0:99999:7:::
mail:*:19469:0:99999:7:::
operator:*:19469:0:99999:7:::
```

Figure 11: Файл /etc/shadow

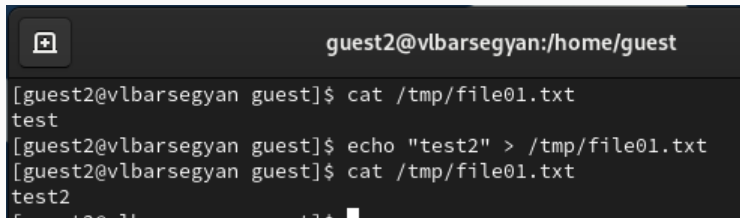
Проверяю, установлен ли атрибут Sticky на директории /tmp командой `ls -l | grep tmp`. От пользователя guest создаю файл со словом test командой `echo "test" > /tmp/file01.txt`. Просматриваю атрибуты у только что созданного файла и разрешаю чтение и запись для категории пользователей «все остальные»



```
guest@vlbarsegyan:~  
[guest@vlbarsegyan ~]$ ls -l / | grep tmp  
drwxrwxrwt. 16 root root 4096 map 18 15:51 tmp  
[guest@vlbarsegyan ~]$ echo "test" > /tmp/file01.txt  
[guest@vlbarsegyan ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 map 18 15:54 /tmp/file01.txt  
[guest@vlbarsegyan ~]$ chmod o+rw /tmp/file01.txt  
[guest@vlbarsegyan ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 map 18 15:54 /tmp/file01.txt  
[guest@vlbarsegyan ~]$
```

Figure 12: Проверка атрибута, работа с файлом

От пользователя `guest` пробую прочитать файл командой `cat /tmp/file01.txt`, далее записываю в файл слово `test2` и вновь читаю его - текст файла изменен

A terminal window with a dark background. The title bar shows a window icon and the text "guest2@vlbarsegyan:/home/guest". The terminal content shows three commands and their outputs: 1. Command: `cat /tmp/file01.txt`, Output: `test`. 2. Command: `echo "test2" > /tmp/file01.txt`. 3. Command: `cat /tmp/file01.txt`, Output: `test2`.

```
guest2@vlbarsegyan:/home/guest
[guest2@vlbarsegyan guest]$ cat /tmp/file01.txt
test
[guest2@vlbarsegyan guest]$ echo "test2" > /tmp/file01.txt
[guest2@vlbarsegyan guest]$ cat /tmp/file01.txt
test2
```

Figure 13: Действия с файлом от другого пользователя

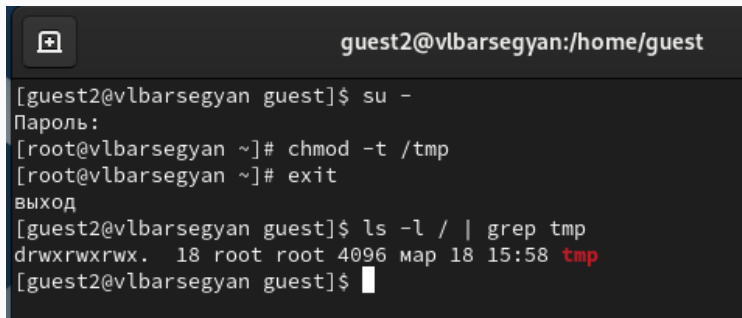
От пользователя `guest2` пробую записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt` - операцию выполнить удалось. Просматриваю содержимое файла и пробую удалить его - удалить не удалось

A terminal window with a dark background. The title bar shows a window icon and the text "guest2@vlbarsegyan:/home/guest". The terminal contains the following text:

```
[guest2@vlbarsegyan guest]$ echo "test3" > /tmp/file01.txt
[guest2@vlbarsegyan guest]$ cat /tmp/file01.txt
test3
[guest2@vlbarsegyan guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@vlbarsegyan guest]$
```

Figure 14: Действия с файлом от другого пользователя

От суперпользователя ввожу команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`. Проверяю от пользователя `guest2`, что атрибута `t` у директории `/tmp` нет командой `ls -l / | grep tmp`




```
guest2@vlbarsegyan:/home/guest

[guest2@vlbarsegyan guest]$ su -
Пароль:
[root@vlbarsegyan ~]# chmod -t /tmp
[root@vlbarsegyan ~]# exit
выход
[guest2@vlbarsegyan guest]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 map 18 15:58 tmp
[guest2@vlbarsegyan guest]$
```

Figure 15: Снятие Sticky-бита с директории

Снова пробуем записать, прочитать и удалить файл - все операции выполнены успешно

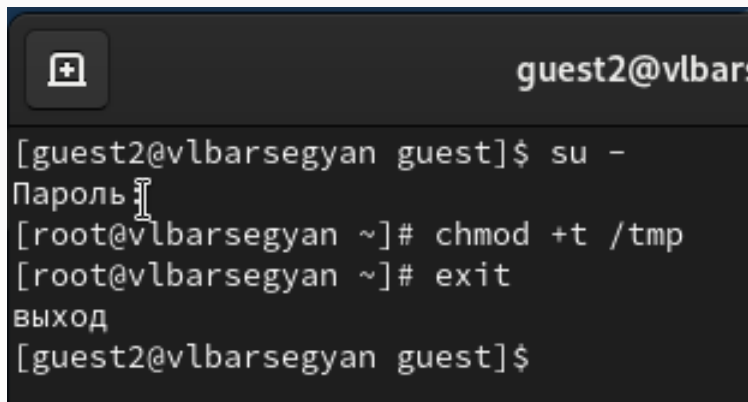


```
guest2@vlbarsegyan:/home/guest

[guest2@vlbarsegyan guest]$ echo "test2" > /tmp/file01.txt
[guest2@vlbarsegyan guest]$ cat /tmp/file01.txt
test2
[guest2@vlbarsegyan guest]$ echo "test3" > /tmp/file01.txt
[guest2@vlbarsegyan guest]$ cat /tmp/file01.txt
test3
[guest2@vlbarsegyan guest]$ rm /tmp/file01.txt
[guest2@vlbarsegyan guest]$
```

Figure 16: Запись, чтение и удаление

Возвращаюсь в суперпользователя и возвращаю атрибут `t` на директорию `/tmp` командой `chmod +t /tmp`



```
guest2@vlbarsegyan  
[guest2@vlbarsegyan guest]$ su -  
Пароль:  
[root@vlbarsegyan ~]# chmod +t /tmp  
[root@vlbarsegyan ~]# exit  
выход  
[guest2@vlbarsegyan guest]$
```

Figure 17: Возвращение атрибута `t`

Я научился применять SetUID- и Sticky-биты, поработал с дополнительными атрибутами в консоли, рассмотрел работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.