

# **Отчёт по лабораторной работе №5**

**Дисциплина: Основы информационной безопасности**

**Барсегян Вардан Левонович НПИбд-01-22**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Создание программы . . . . .	6
2.2	Исследование Sticky-бита . . . . .	11
<b>3</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

# Список иллюстраций

2.1	Программа simpleid.c . . . . .	6
2.2	Компиляция программы . . . . .	7
2.3	Выполнение программы . . . . .	7
2.4	Программа simpleid2.c . . . . .	8
2.5	Компиляция и запуск . . . . .	8
2.6	Изменение атрибутов, запуск . . . . .	9
2.7	Изменение SetGID-бита и проверка . . . . .	9
2.8	Программа readfile.c . . . . .	10
2.9	Компиляция программы, смена прав доступа . . . . .	10
2.10	Установка SetU'D-бита, проверка . . . . .	11
2.11	Файл /etc/shadow . . . . .	11
2.12	Проверка атрибута, работа с файлом . . . . .	12
2.13	Действия с файлом от другого пользователя . . . . .	12
2.14	Действия с файлом от другого пользователя . . . . .	12
2.15	Снятие Sticky-бита с директории . . . . .	13
2.16	Запись, чтение и удаление . . . . .	13
2.17	Возвращение атрибута t . . . . .	14

## Список таблиц

# 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 Выполнение лабораторной работы

### 2.1 Создание программы

1. Создаю файл `simpleid.c` (рис. 2.1)

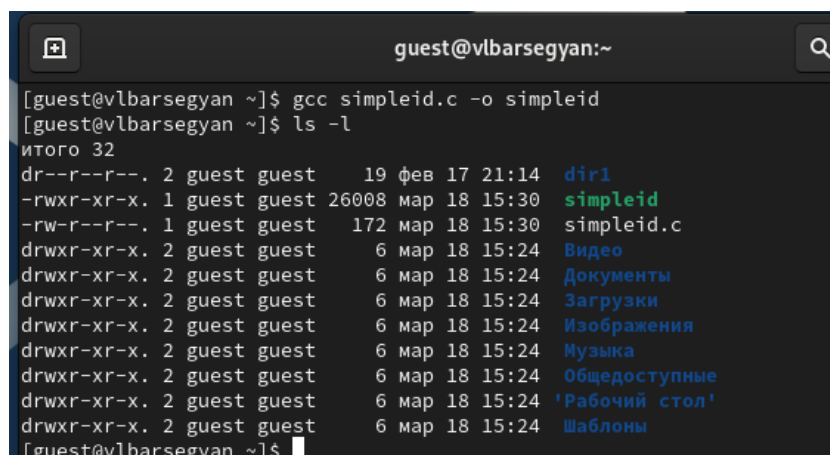


```
guest@vlbarsegyan:~ — nano simpleid.c
GNU nano 5.6.1 simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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

Рис. 2.1: Программа `simpleid.c`

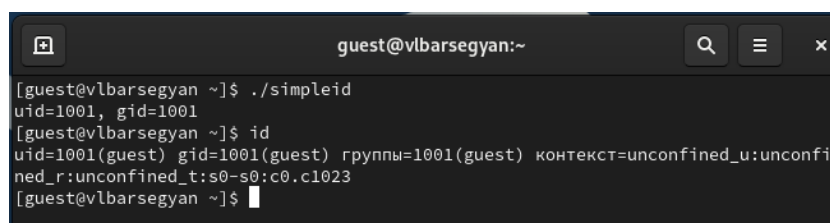
2. Компилирую программу командой `gcc simpleid.c -o simpleid` и проверяю, что файл создан (рис. 2.2)



```
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 ~]$
```

Рис. 2.2: Компиляция программы

3. Выполняю программу simpleid командой `./simpleid`, а затем системную программу `id` - вывод одинаков (рис. 2.3)



```
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 ~]$
```

Рис. 2.3: Выполнение программы

4. Усложняю программу и записываю ее в файл simpleid2.c (рис. 2.4)



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

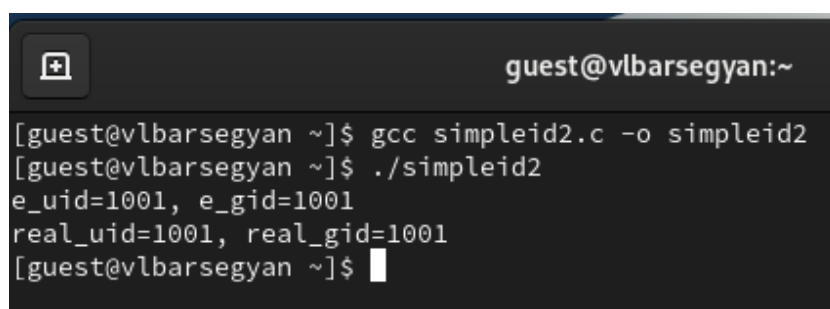
int main() {
    uid_t real_uid = getuid();
    uid_t e_uid = geteuid();

    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;
}
```

Рис. 2.4: Программа simpleid2.c

5. Компилирую и запускаю программу командами `gcc simpleid2.c -o simpleid2` и `./simpleid2` (рис. 2.5)

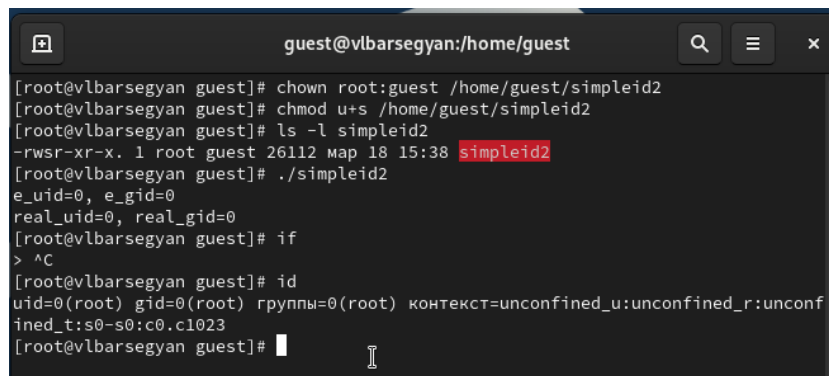


```
guest@vlbarsegyan:~
[guest@vlbarsegyan ~]$ gcc simpleid2.c -o simpleid2
[guest@vlbarsegyan ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@vlbarsegyan ~]$
```

Рис. 2.5: Компиляция и запуск

6. 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` (рис. 2.6)

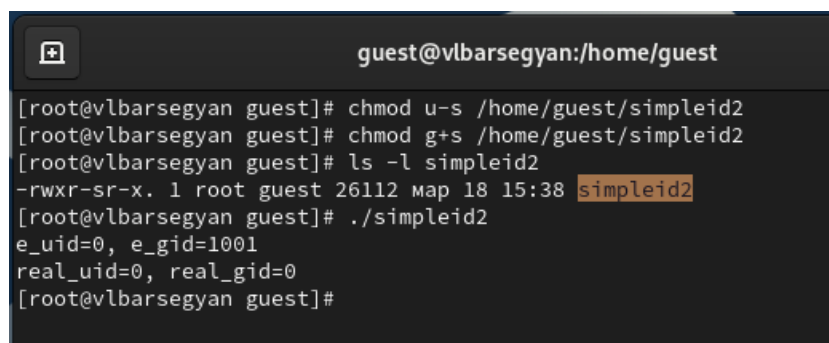




```
guest@vlbarsegyan:/home/guest
[root@vlbarsegyan guest]# chown root:guest /home/guest/simpleid2
[root@vlbarsegyan guest]# chmod u+s /home/guest/simpleid2
[root@vlbarsegyan guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26112 map 18 15:38 simpleid2
[root@vlbarsegyan guest]# ./simpleid2
e_uid=0, e_gid=0
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:unconfined_t:s0-s0:c0.c1023
[root@vlbarsegyan guest]#
```

Рис. 2.6: Изменение атрибутов, запуск

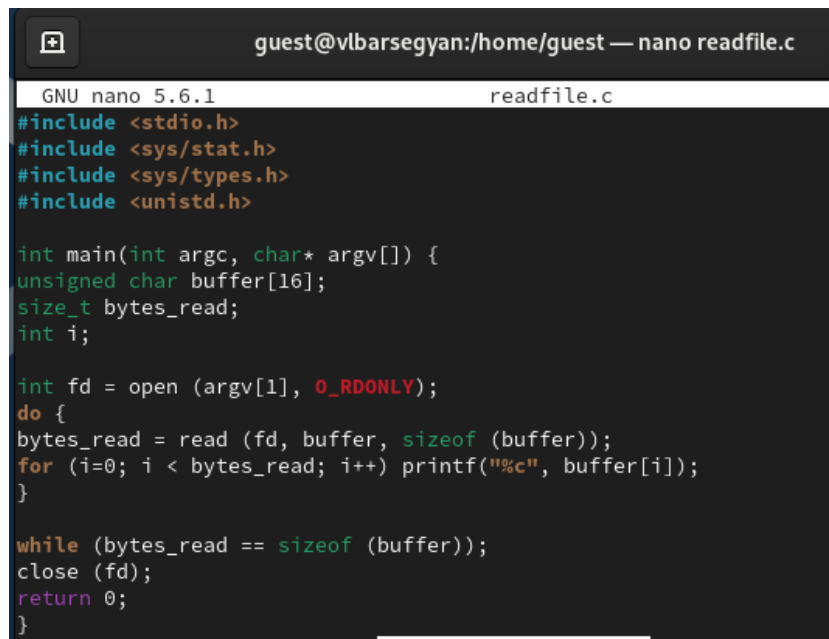
7. Делаю тоже самое относительно SetGID-бита: устанавливаю его командой `chmod g+s /home/guest/simpleid2`, проверяю установку нового атрибута и запускаю `simpleid2` и `id` (рис. 2.7)



```
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]#
```

Рис. 2.7: Изменение SetGID-бита и проверка

8. Создаю программу `readfile.c` (рис. 2.8)



```
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>

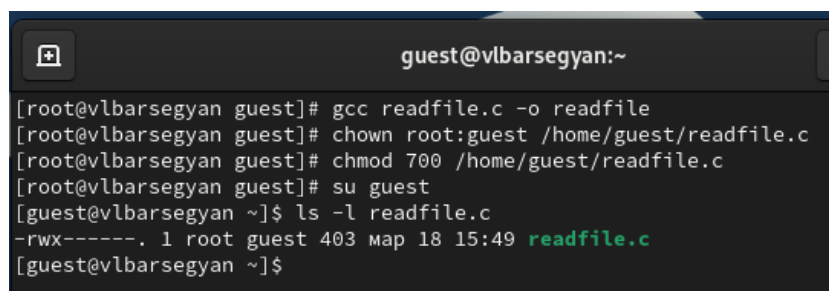
int main(int argc, char* argv[]) {
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    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;
}
```

Рис. 2.8: Программа readfile.c

9. Компилирую ее командой `gcc readfile.c -o readfile` и изменяю права доступа так, чтобы только суперпользователь мог прочитать его, а guest не мог (рис. 2.9)



```
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 ~]$
```

Рис. 2.9: Компиляция программы, смена прав доступа

10. 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 (рис. 2.10)

```
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;
```

Рис. 2.10: Установка SetU'D-бита, проверка

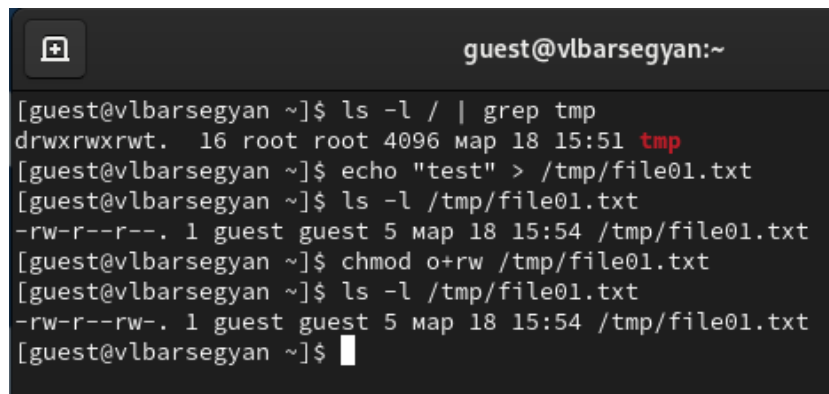
11. Проверяю, может ли программа readfile прочитать файл /etc/shadow - да, может (рис. 2.11)

```
guest@vlbarsegyan:~  
[guest@vlbarsegyan ~]$ ./readfile /etc/shadow  
root:$6$vp4facNM90uKkVGB$HqdiTatj7A0L6wjom4R0biuc5wM1Lq8t1ezS2FIk4ZhH.aQeh3tbdd9  
gN5qVsl7owcZ7uDqffK0eFeugeXWkH/:0:99999:7:::  
bin:!:19469:0:99999:7:::  
daemon:!:19469:0:99999:7:::  
adm:!: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:::
```

Рис. 2.11: Файл /etc/shadow

## 2.2 Исследование Sticky-бита

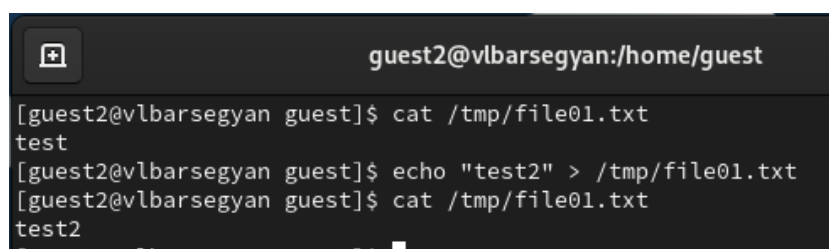
12. Проверяю, установлен ли атрибут Sticky на директории /tmp командой `ls -l | grep tmp`. От пользователя guest создаю файл со словом test командой `echo "test" > /tmp/file01.txt`. Просматриваю атрибуты у только что созданного файла и разрешаю чтение и запись для категории пользователей «все остальные» (рис. 2.12)



```
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 ~]$
```

Рис. 2.12: Проверка атрибута, работа с файлом


13. От пользователя guest пробую прочитать файл командой `cat /tmp/file01.txt`, далее записываю в файл слово `test2` и вновь читаю его - текст файла изменен (рис. 2.13)



```
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  
[guest2@vlbarsegyan guest]$
```

Рис. 2.13: Действия с файлом от другого пользователя

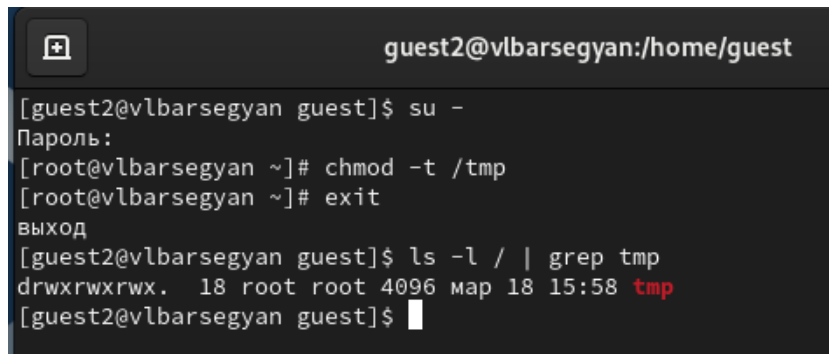
14. От пользователя `guest2` пробую записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt` - операцию выполнить удалось. Просматриваю содержимое файла и пробую удалить его - удалить не удалось (рис. 2.14)



```
guest2@vlbarsegyan:/home/guest  
[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]$
```

Рис. 2.14: Действия с файлом от другого пользователя

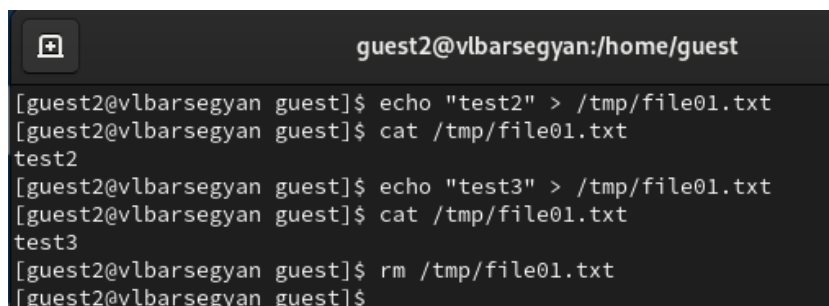
15. От суперпользователя ввожу команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`. Проверяю от пользователя `guest2`, что атрибута `t` у директории `/tmp` нет командой `ls -l / | grep tmp` (рис. 2.15)



```
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 мар 18 15:58 tmp
[guest2@vlbarsegyan guest]$
```

Рис. 2.15: Снятие Sticky-бита с директории

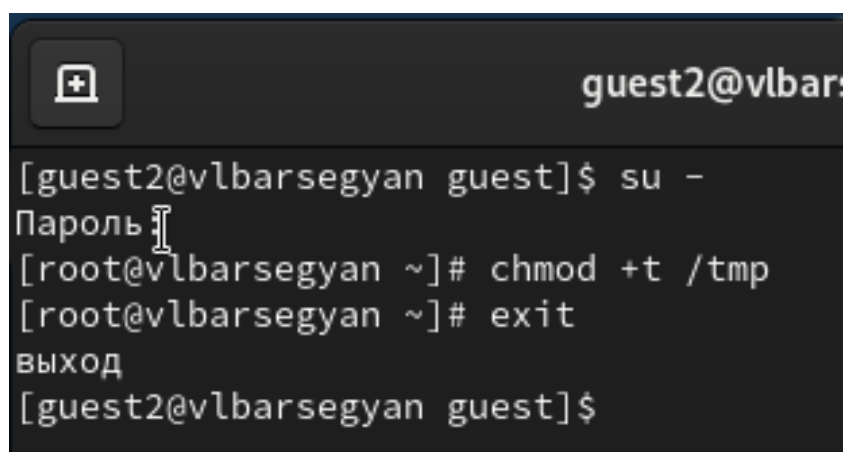
16. Снова пробую записать, прочитать и удалить файл - все операции выполнены успешно (рис. 2.16)



```
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]$
```

Рис. 2.16: Запись, чтение и удаление

17. Возвращаюсь в суперпользователя и возвращаю атрибут `t` на директорию `/tmp` командой `chmod +t /tmp` (рис. 2.17)



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

Рис. 2.17: Возвращение атрибута t

## 3 Выводы

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.

## **Список литературы**