

# **Отчёт по лабораторной работе 5**

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

Гебриал Ибрам Есам Зекри НПИ-01-18

# Содержание

1	Цель работы	5
2	Теоретические сведения	6
3	Выполнение лабораторной работы	8
4	Выводы	19
5	Список литературы	20

## List of Tables

# List of Figures

3.1	Отключение системы SELINUX . . . . .	8
3.2	Создание программы simpleid.c . . . . .	8
3.3	Компиляция программы . . . . .	9
3.4	Выполнение программы . . . . .	9
3.5	Создание программы simpleid2.c . . . . .	10
3.6	Компиляция и выполнение программы . . . . .	10
3.7	Выполнение команды . . . . .	11
3.8	Проверка правильности установки новых атрибутов и смены владельца файла simpleid2 . . . . .	11
3.9	Выполнение программы . . . . .	11
3.10	Выполнение программы . . . . .	12
3.11	Создание программы readfile.c . . . . .	12
3.12	Компиляция программы . . . . .	13
3.13	Изменение владельца прав у файла readfile.c . . . . .	13
3.14	Проверка, что пользователь guest не может прочитать файл readfile.c	13
3.15	Изменение прав для программы readfile . . . . .	14
3.16	Проверка чтения файла readfile.c с помощью readfile . . . . .	14
3.17	Проверка чтения файла /etc/shadow с помощью readfile . . . . .	15
3.18	выяснение, установлен ли атрибут Sticky на директории /tmp . . . . .	15
3.19	Создание файла file01.txt в директории /tmp со словом test . . . . .	15
3.20	Просмотр атрибуты и разрешил чтение и запись для категории пользователей «все остальные». . . . .	16
3.21	Просмотр атрибуты и разрешил чтение и запись для категории пользователей «все остальные». . . . .	16
3.22	попытка дозаписи в файла, записи, удаления файла и проверка. . . . .	16
3.23	Снятие атрибута “t” с директории /tmp. . . . .	17
3.24	Повторение предыдущих шагов. . . . .	17
3.25	Добавление атрибута “t” на директорию /tmp . . . . .	18

# 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 Теоретические сведения

Linux — многопользовательская система, поэтому она должна предоставлять систему разрешений, чтобы контролировать авторизованные операции с файлами и каталогами, к которым относятся все системные ресурсы и устройства (в Unix-системах любое устройство представляется в виде файла или каталога). Этот принцип является общим для всех Unix-систем, но напомнить об этом ещё раз будет не лишним, тем более что существуют некоторые интересные и сравнительно малоизвестные способы применения.

У каждого файла и каталога имеются специальные разрешения для трёх категорий пользователей:

- его владельца (обозначается `u`, от «user»);
- его группы-владельца (обозначается `g`, от «group»), представленная всеми членами группы;
- остальных (обозначается `o`, от «other»).

Три типа прав могут использоваться совместно:

- чтение (обозначается `r`, от «read»);
- запись (или изменение, обозначается `w`, от «write»);
- исполнение (обозначается `x`, от «execute»).

Два своеобразных права имеют смысл для исполняемых файлов: `setuid` и `setgid` (обозначаются буквой «`s`»). Обратите внимание, что они часто называются «битами», поскольку каждое из этих логических значений может быть представлено

как 0 или 1. Эти два права позволяют любому пользователю выполнять программу на правах её владельца или группы соответственно. Данный механизм предоставляет доступ к функциям, требующим разрешений более высокого уровня, чем обычно есть у пользователя.

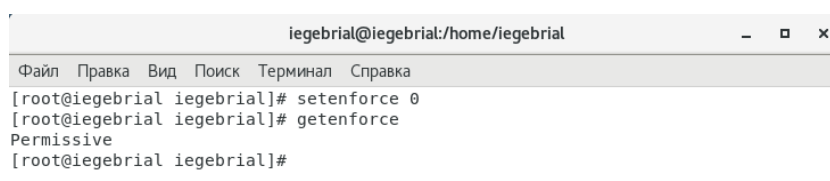
Поскольку `setuid`-программа, принадлежащая `root`, систематически запускается с правами суперпользователя, крайне важно убедиться в её безопасности и надёжности. Действительно, пользователь, которому удастся заставить её вызвать другую произвольную программу, сможет представиться как `root` и получить все права в системе. [1]

Чтобы применить соответствующие разрешения, первое, что нужно учитывать, это владение. Для этого есть команда `chown`.

Для управления правами используется команда `chmod`. При использовании `chmod` мы можем устанавливать разрешения для пользователя (`user`), группы (`group`) и других (`other`). [2]

### 3 Выполнение лабораторной работы

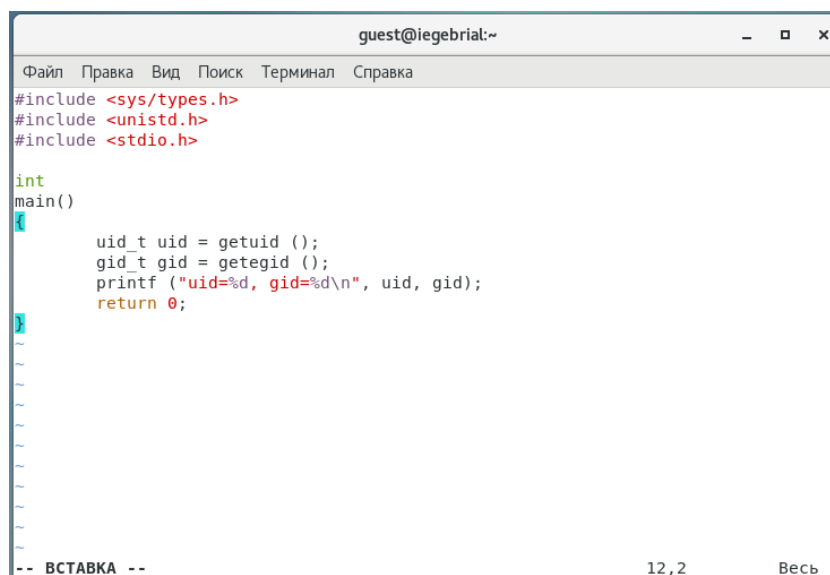
1. Отключил систему SELinux. (рис. 3.1)



```
iegebrial@iegebrial:/home/iegebrial
[root@iegebrial iegebrial]# setenforce 0
[root@iegebrial iegebrial]# getenforce
Permissive
[root@iegebrial iegebrial]#
```

Figure 3.1: Отключение системы SELINUX

2. Вошёл в систему от имени пользователя guest и создал программу simpleid.c. (рис. 3.2)



```
guest@iegebrial:~
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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

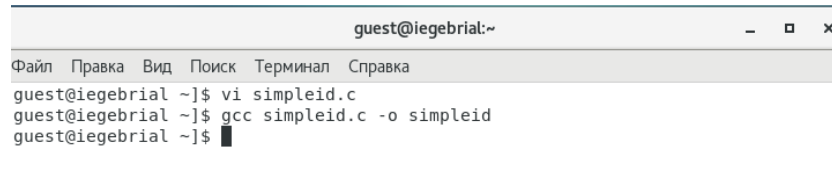
Figure 3.2: Создание программы simpleid.c

3. Скомпилировал программу. (рис. 3.3)



С помощью команды:

`gcc simpleid.c -o simpleid`

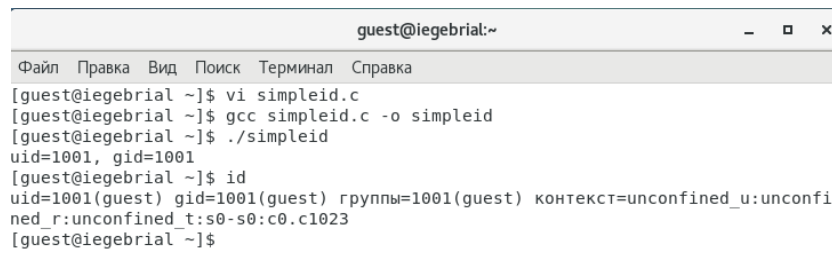


```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
guest@iegebrial ~]$ vi simpleid.c  
guest@iegebrial ~]$ gcc simpleid.c -o simpleid  
guest@iegebrial ~]$
```

Figure 3.3: Компиляция программы

4. Выполнил программу `simpleid` и выполнил системную программу `id`. (рис. 3.4)

При сравнении результата `id` и `simpleid` получил одинаковые данные.



```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ vi simpleid.c  
[guest@iegebrial ~]$ gcc simpleid.c -o simpleid  
[guest@iegebrial ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@iegebrial ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@iegebrial ~]$
```

Figure 3.4: Выполнение программы

5. Усложнил программу, добавив вывод действительных идентификаторов. (рис. 3.5)

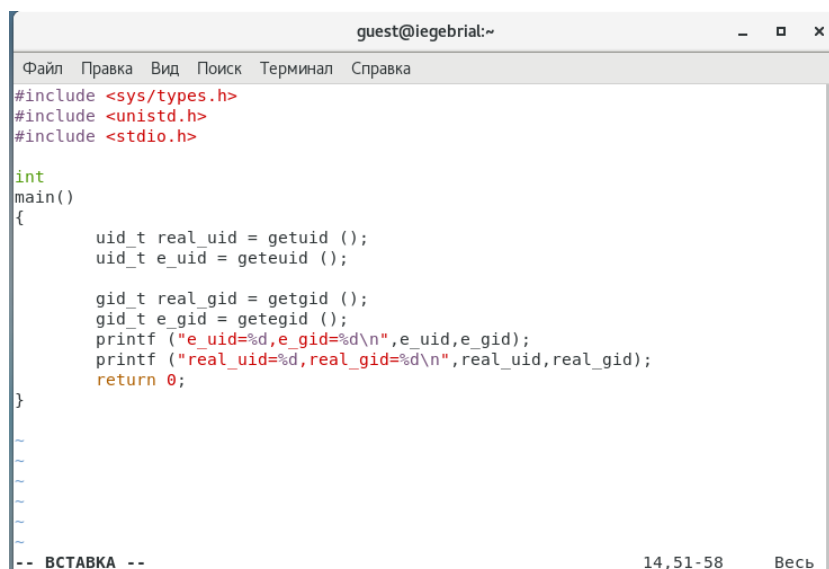
A terminal window titled 'guest@iegebrial:~' with a menu bar (Файл, Правка, Вид, Поиск, Терминал, Справка). The terminal shows the creation of a C program named 'simpleid2.c'. The code includes headers for <sys/types.h>, <unistd.h>, and <stdio.h>. The main function calls getuid, geteuid, getgid, and getegid to retrieve user and group IDs, then prints them in two lines. The status bar at the bottom shows '-- ВСТАВКА --', '14,51-58', and 'Весь'.

Figure 3.5: Создание программы simpleid2.c

6. Скомпилировал и запустил simpleid2.c. (рис. 3.6)

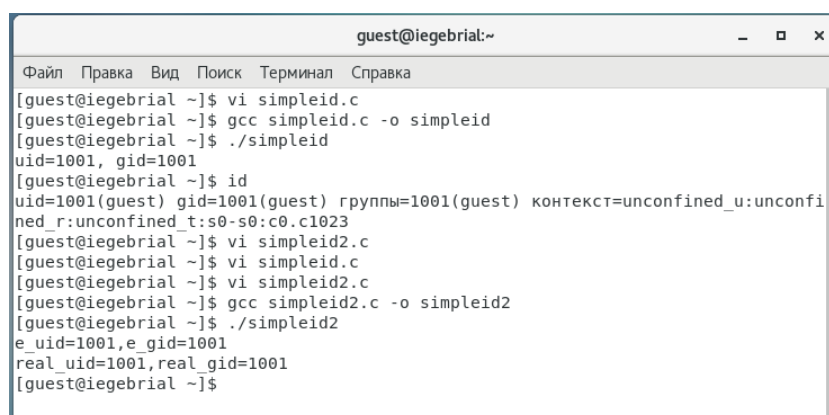
A terminal window titled 'guest@iegebrial:~' showing the compilation and execution of the program. The user runs 'vi simpleid.c', 'gcc simpleid.c -o simpleid', and './simpleid'. The output shows 'uid=1001, gid=1001'. Then the user runs 'id', showing 'uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023'. Next, the user runs 'vi simpleid2.c', 'gcc simpleid2.c -o simpleid2', and './simpleid2'. The output shows 'e\_uid=1001,e\_gid=1001' and 'real\_uid=1001,real\_gid=1001'.

Figure 3.6: Компиляция и выполнение программы

7. От имени суперпользователя выполнил команды. (рис. 3.7)

chown root:guest /home/guest/simpleid2 - поменял пользователь на root и группа на guest.

chmod u+s /home/guest/simpleid2 - разрешение для владельца на выполнение с разрешением суперпользователя.

```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[root@iegebrial ~]# chown root:guest /home/guest/simpleid2  
[root@iegebrial ~]# chmod u+s /home/guest/simpleid2  
[root@iegebrial ~]#
```

Figure 3.7: Выполнение команды

8. Выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2. (рис. 3.8)

```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8616 ноя 10 14:10 simpleid2  
[guest@iegebrial ~]$
```

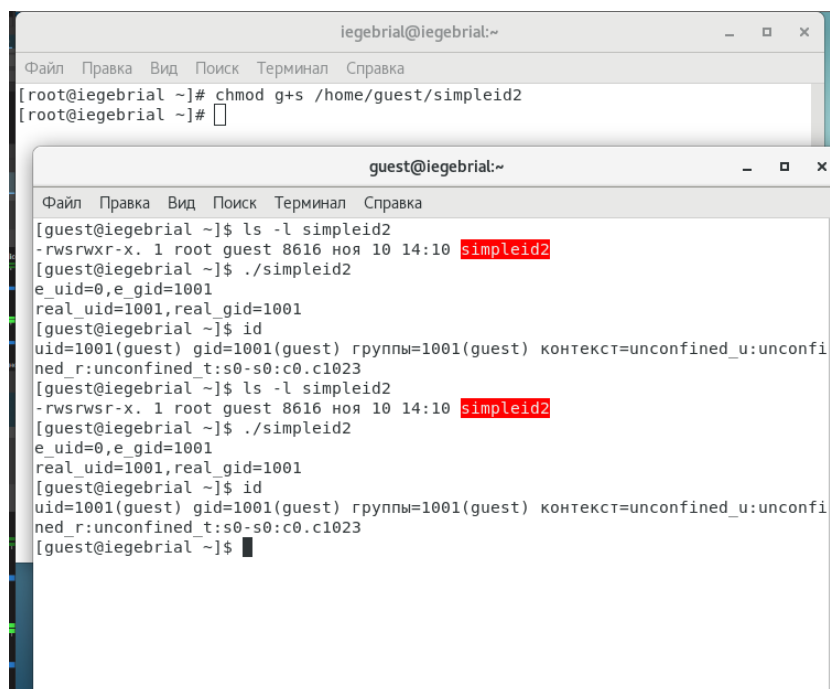
Figure 3.8: Проверка правильности установки новых атрибутов и смены владельца файла simpleid2

9. Запустил simpleid2 и id. (рис. 3.9)

```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8616 ноя 10 14:10 simpleid2  
[guest@iegebrial ~]$ ./simpleid2  
e_uid=0,e_gid=1001  
real_uid=1001,real_gid=1001  
[guest@iegebrial ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi  
ned r:unconfined t:s0-s0:c0.c1023
```

Figure 3.9: Выполнение программы

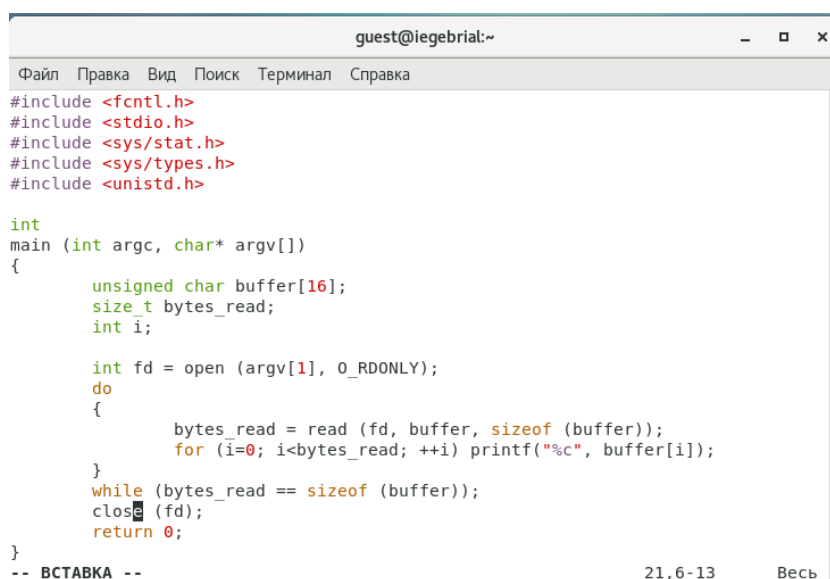
10. Проделал тоже самое относительно SetGID-бита. (рис. 3.10)



```
iegebrial@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[root@iegebrial ~]# chmod g+s /home/guest/simpleid2  
[root@iegebrial ~]#  
  
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8616 ноя 10 14:10 simpleid2  
[guest@iegebrial ~]$ ./simpleid2  
e_uid=0,e_gid=1001  
real_uid=1001,real_gid=1001  
[guest@iegebrial ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@iegebrial ~]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8616 ноя 10 14:10 simpleid2  
[guest@iegebrial ~]$ ./simpleid2  
e_uid=0,e_gid=1001  
real_uid=1001,real_gid=1001  
[guest@iegebrial ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@iegebrial ~]$
```

Figure 3.10: Выполнение программы

11. Создал программу readfile.c. (рис. 3.11)



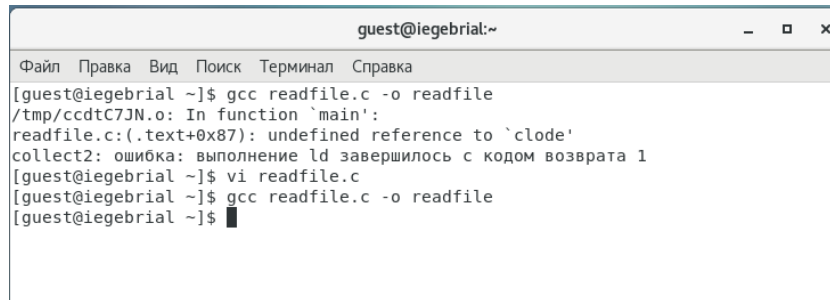
```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
#include <fcntl.h>  
#include <stdio.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
int  
main (int argc, char* argv[])  
{  
    unsigned char buffer[16];  
    size_t bytes_read;  
    int i;  
  
    int 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 -- 21,6-13 Весь
```

Figure 3.11: Создание программы readfile.c

12. Откомпилировал её. (рис. 3.12)

с помощью команды:

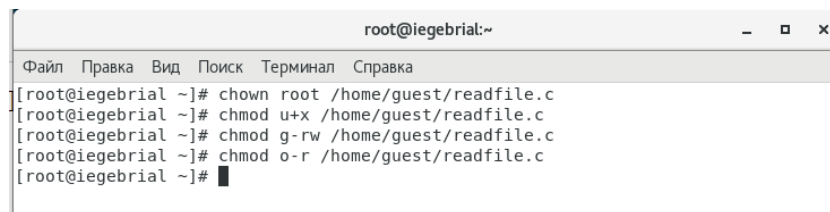
`gcc readfile.c -o readfile`



```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ gcc readfile.c -o readfile  
/tmp/ccdtC7JN.o: In function `main':  
readfile.c:(.text+0x87): undefined reference to `cldoe'  
collect2: ошибка: выполнение ld завершилось с кодом возврата 1  
[guest@iegebrial ~]$ vi readfile.c  
[guest@iegebrial ~]$ gcc readfile.c -o readfile  
[guest@iegebrial ~]$
```

Figure 3.12: Компиляция программы

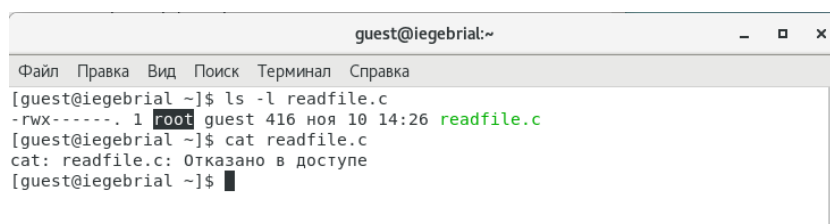
13. Сменил владельца у файла `readfile.c` (или любого другого текстового файла в системе) и изменил права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог.. (рис. 3.13)



```
root@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[root@iegebrial ~]# chown root /home/guest/readfile.c  
[root@iegebrial ~]# chmod u+x /home/guest/readfile.c  
[root@iegebrial ~]# chmod g-rw /home/guest/readfile.c  
[root@iegebrial ~]# chmod o-r /home/guest/readfile.c  
[root@iegebrial ~]#
```

Figure 3.13: Изменение владельца прав у файла `readfile.c`

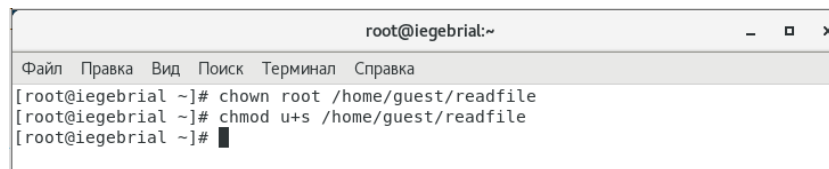
14. Проверил, что пользователь `guest` не может прочитать файл `readfile.c`. (рис. 3.14)



```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ ls -l readfile.c  
-rwx-----, 1 root guest 416 ноя 10 14:26 readfile.c  
[guest@iegebrial ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@iegebrial ~]$
```

Figure 3.14: Проверка, что пользователь `guest` не может прочитать файл `readfile.c`

15. Сменил у программы `readfile` владельца и установил SetU'D-бит. (рис. 3.15)

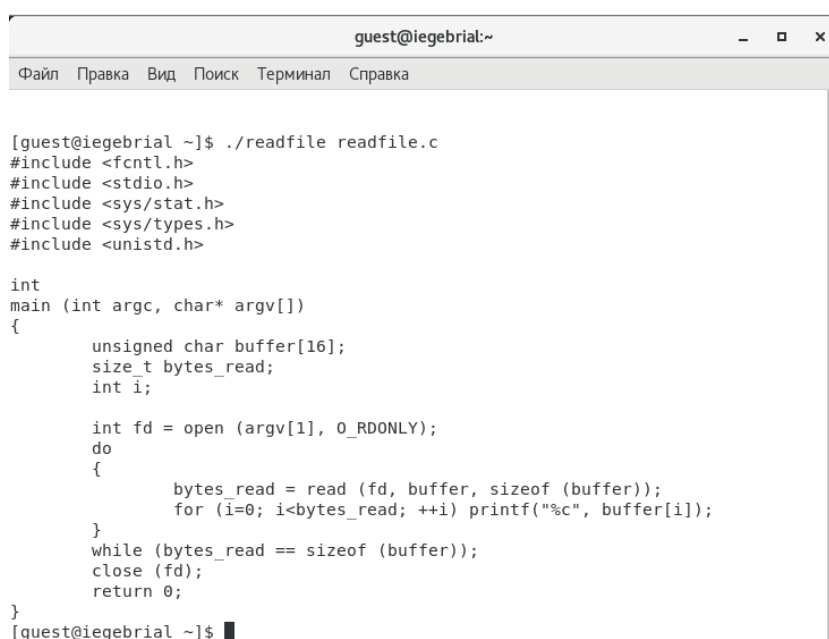


```
root@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[root@iegebrial ~]# chown root /home/guest/readfile  
[root@iegebrial ~]# chmod u+s /home/guest/readfile  
[root@iegebrial ~]#
```

Figure 3.15: Изменение прав для программы readfile

16. Проверил, что программа readfile может прочитать файл readfile.c. (рис. 3.16)

Да получилось для программы readfile читать файл readfile.c.

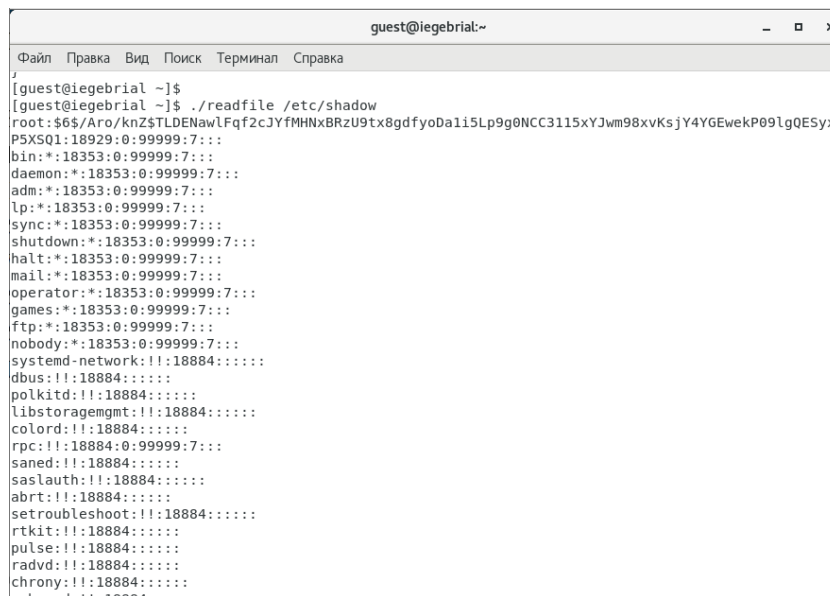


```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ ./readfile readfile.c  
#include <fcntl.h>  
#include <stdio.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
int  
main (int argc, char* argv[])  
{  
    unsigned char buffer[16];  
    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]);  
    }  
    while (bytes_read == sizeof (buffer));  
    close (fd);  
    return 0;  
}  
[guest@iegebrial ~]$
```

Figure 3.16: Проверка чтения файла readfile.c с помощью readfile

17. Проверил, что программа readfile может прочитать файл /etc/shadow. (рис. 3.17)

Да получилось для программы readfile читать файл /etc/shadow.



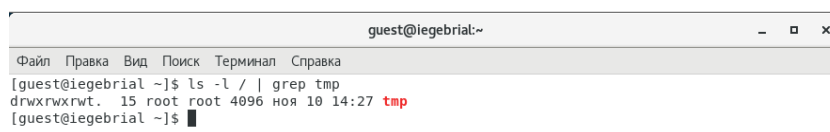
```
guest@iegebril:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebril ~]$  
[guest@iegebril ~]$ ./readfile /etc/shadow  
root:$6$/Aro/knZ$TLDENawlfqf2cJYfMHNxBRzU9tx8gdfyoDali5Lp9g0NCC3115xYJwm98xvKsjY4YGEwekP09lgQESyx  
P5XSQ1:18929:0:99999:7:::  
bin:!:18353:0:99999:7:::  
daemon:!:18353:0:99999:7:::  
adm:!:18353:0:99999:7:::  
lp:!:18353:0:99999:7:::  
sync:!:18353:0:99999:7:::  
shutdown:!:18353:0:99999:7:::  
halt:!:18353:0:99999:7:::  
mail:!:18353:0:99999:7:::  
operator:!:18353:0:99999:7:::  
games:!:18353:0:99999:7:::  
ftp:!:18353:0:99999:7:::  
nobody:!:18353:0:99999:7:::  
systemd-network:!!:18884::::::  
dbus:!!:18884::::::  
polkitd:!!:18884::::::  
libstoragemgmt:!!:18884::::::  
colord:!!:18884::::::  
rpc:!!:18884:0:99999:7:::  
sane:!!:18884::::::  
saslauth:!!:18884::::::  
abrt:!!:18884::::::  
setroubleshoot:!!:18884::::::  
rtkit:!!:18884::::::  
pulse:!!:18884::::::  
radvd:!!:18884::::::  
chrony:!!:18884::::::  
unbound:!!:18884::::::
```

Figure 3.17: Проверка чтения файла /etc/shadow с помощью readfile

18. Выяснил, установлен ли атрибут Sticky на директории /tmp (рис. 3.18)

С помощью команды

`ls -l / | grep tmp`



```
guest@iegebril:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebril ~]$ ls -l / | grep tmp  
drwxrwxrwt. 15 root root 4096 ноя 10 14:27 tmp  
[guest@iegebril ~]$
```

Figure 3.18: выяснение, установлен ли атрибут Sticky на директории /tmp

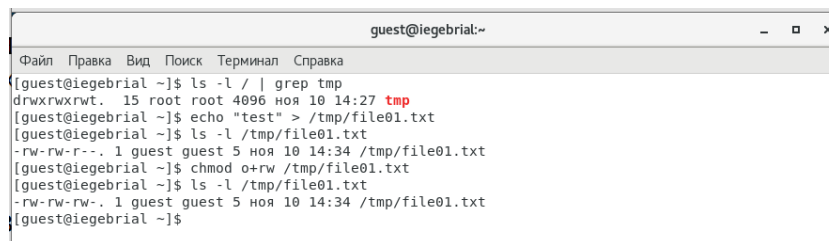
19. От имени пользователя guest создал файл file01.txt в директории /tmp со словом test. (рис. 3.19)



```
guest@iegebril:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebril ~]$ ls -l / | grep tmp  
drwxrwxrwt. 15 root root 4096 ноя 10 14:27 tmp  
[guest@iegebril ~]$ echo "test" > /tmp/file01.txt  
[guest@iegebril ~]$
```

Figure 3.19: Создание файла file01.txt в директории /tmp со словом test

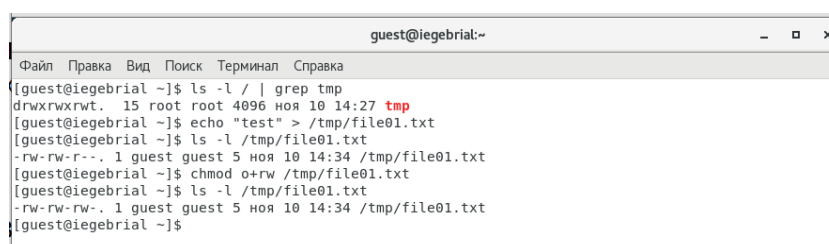
20. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные». (рис. 3.20)



```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ ls -l / | grep tmp  
drwxrwxrwt. 15 root root 4096 ноя 10 14:27 tmp  
[guest@iegebrial ~]$ echo "test" > /tmp/file01.txt  
[guest@iegebrial ~]$ ls -l /tmp/file01.txt  
-rw-rw-r--. 1 guest guest 5 ноя 10 14:34 /tmp/file01.txt  
[guest@iegebrial ~]$ chmod o+rw /tmp/file01.txt  
[guest@iegebrial ~]$ ls -l /tmp/file01.txt  
-rw-rw-rw-. 1 guest guest 5 ноя 10 14:34 /tmp/file01.txt  
[guest@iegebrial ~]$
```

Figure 3.20: Просмотр атрибуты и разрешил чтение и запись для категории пользователей «все остальные».

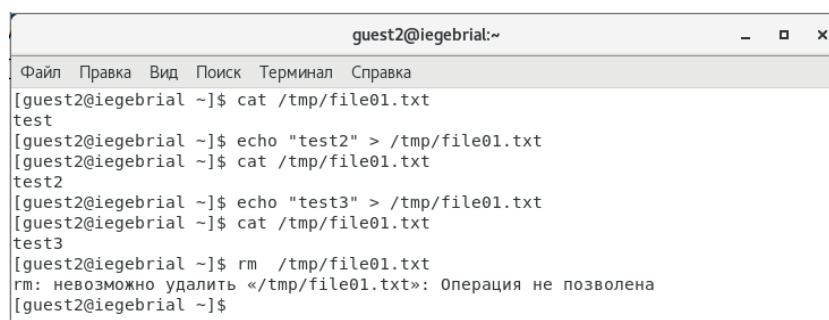
21. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные». (рис. 3.21)



```
guest@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@iegebrial ~]$ ls -l / | grep tmp  
drwxrwxrwt. 15 root root 4096 ноя 10 14:27 tmp  
[guest@iegebrial ~]$ echo "test" > /tmp/file01.txt  
[guest@iegebrial ~]$ ls -l /tmp/file01.txt  
-rw-rw-r--. 1 guest guest 5 ноя 10 14:34 /tmp/file01.txt  
[guest@iegebrial ~]$ chmod o+rw /tmp/file01.txt  
[guest@iegebrial ~]$ ls -l /tmp/file01.txt  
-rw-rw-rw-. 1 guest guest 5 ноя 10 14:34 /tmp/file01.txt  
[guest@iegebrial ~]$
```

Figure 3.21: Просмотр атрибуты и разрешил чтение и запись для категории пользователей «все остальные».

22. От пользователя guest2 попробовал дозаписать в файл /tmp/file01.txt слово test2 командой и проверял. Попробовал записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой. Проверил содержимое файла командой и потом постарался удалить файл но не получилось удалить файл (рис. 3.22)

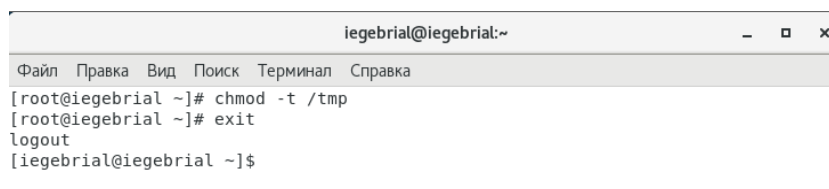


```
guest2@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
test  
[guest2@iegebrial ~]$ echo "test2" > /tmp/file01.txt  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
test2  
[guest2@iegebrial ~]$ echo "test3" > /tmp/file01.txt  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
test3  
[guest2@iegebrial ~]$ rm /tmp/file01.txt  
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена  
[guest2@iegebrial ~]$
```

Figure 3.22: попытка дозаписи в файла, записи, удаления файла и проверка.



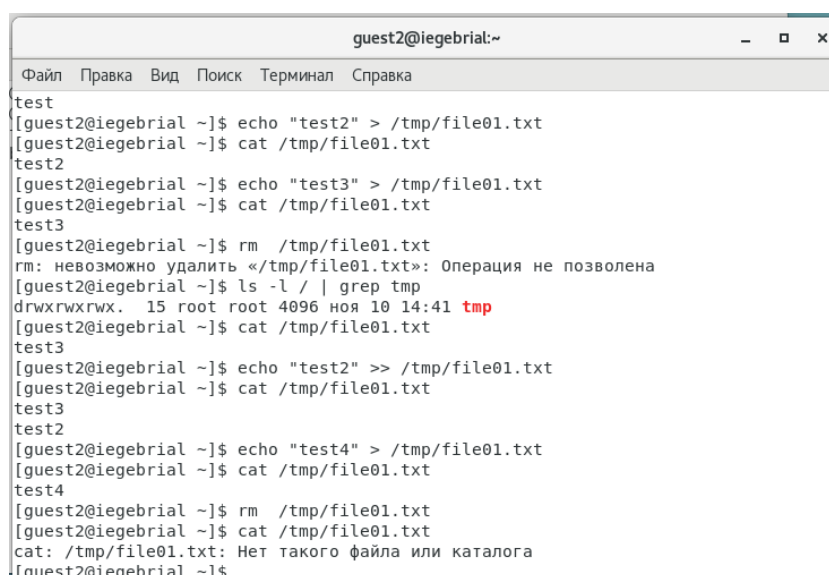
23. Повысил свои права до суперпользователя и выполнил после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp` покинул режим суперпользователя командой. (рис. 3.23)



```
iegebrial@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[root@iegebrial ~]# chmod -t /tmp  
[root@iegebrial ~]# exit  
logout  
[iegebrial@iegebrial ~]$
```

Figure 3.23: Снятие атрибута “t” с директории `/tmp`.

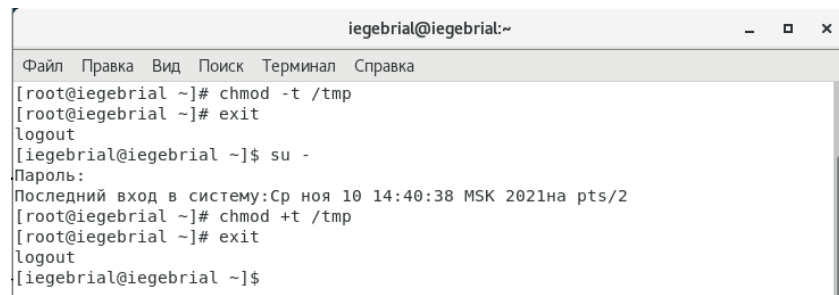
24. Повторил предыдущие шаги. Получилось удалить файл . (рис. 3.24)



```
guest2@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
test  
[guest2@iegebrial ~]$ echo "test2" > /tmp/file01.txt  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
test2  
[guest2@iegebrial ~]$ echo "test3" > /tmp/file01.txt  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
test3  
[guest2@iegebrial ~]$ rm /tmp/file01.txt  
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена  
[guest2@iegebrial ~]$ ls -l / | grep tmp  
drwxrwxrwx. 15 root root 4096 ноя 10 14:41 tmp  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
test3  
[guest2@iegebrial ~]$ echo "test2" >> /tmp/file01.txt  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
test3  
test2  
[guest2@iegebrial ~]$ echo "test4" > /tmp/file01.txt  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
test4  
[guest2@iegebrial ~]$ rm /tmp/file01.txt  
[guest2@iegebrial ~]$ cat /tmp/file01.txt  
cat: /tmp/file01.txt: Нет такого файла или каталога  
[guest2@iegebrial ~]$
```

Figure 3.24: Повторение предыдущих шагов.

25. Повысил свои права до суперпользователя и вернул атрибут `t` на директорию `/tmp` . (рис. 3.25)

A terminal window titled 'iegebrial@iegebrial:~' with standard window controls. The terminal shows a sequence of commands: a user logs out, switches to root with 'su -', runs 'chmod -t /tmp', exits root, logs out again, switches back to root with 'su -', runs 'chmod +t /tmp', and finally exits root. The prompt returns to the regular user.

```
iegebrial@iegebrial:~  
Файл Правка Вид Поиск Терминал Справка  
[root@iegebrial ~]# chmod -t /tmp  
[root@iegebrial ~]# exit  
logout  
[iegebrial@iegebrial ~]$ su -  
Пароль:  
Последний вход в систему: Ср ноя 10 14:40:38 MSK 2021 на pts/2  
[root@iegebrial ~]# chmod +t /tmp  
[root@iegebrial ~]# exit  
logout  
[iegebrial@iegebrial ~]$
```

Figure 3.25: Добавление атрибута “t” на директорию /tmp

## 4 Выводы

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

## 5 Список литературы

1. Управление правами. <https://debian-handbook.info/browse/ru-RU/stable/sect.rights-management.html>.
2. Права в Linux. //Losst.2020. URL:<https://losst.ru/neizmenyaemye-fajly-v-linux> (дата обращения 1.10.2019).