

# **Отчёт по лабораторной работе 5**

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

Максимова Ксения НБИбд-02-18

# **Содержание**

<b>1 Цель работы</b>	<b>4</b>
<b>2 Задание</b>	<b>5</b>
<b>3 Теоретическое введение</b>	<b>6</b>
<b>4 Выводы</b>	<b>29</b>
<b>Список литературы</b>	<b>30</b>

# List of Figures

3.1 Рис 1.Вход в систему . . . . .	8
3.2 Рис 2.Программа simpleid.c . . . . .	9
3.3 Рис 3.Компиляция программы . . . . .	9
3.4 Рис 4.Файл программы . . . . .	10
3.5 Рис 5.Выполнение программы . . . . .	10
3.6 Рис 6.Команда id . . . . .	11
3.7 Рис 7.Программа simpleid2.c . . . . .	12
3.8 Рис 8.Компилияция simpleid2.c . . . . .	12
3.9 Рис 9.Команды chown и chmod . . . . .	13
3.10 Рис 10.Проверка правильности установки новых атрибутов . . . . .	14
3.11 Рис 11.Результаты simpleid2 и id . . . . .	14
3.12 Рис 12.Программа readfile.c . . . . .	15
3.13 Рис 13.Компилияция readfile.c . . . . .	16
3.14 Рис 14.Изменение владельца файла readfile.c и изменение прав доступа . . . . .	16
3.15 Рис 15.Проверка доступа . . . . .	17
3.16 Рис 16.Проверка . . . . .	18
3.17 Рис 17.Проверка . . . . .	18
3.18 Рис 18.Проверка наличия атрибута . . . . .	19
3.19 Рис 19.Создание файла file01.txt . . . . .	20
3.20 Рис 20.Атрибуты файла file01.txt . . . . .	21
3.21 Рис 21.Атрибуты файла file01.txt . . . . .	21
3.22 Рис 22.Чтение файла file01.txt . . . . .	22
3.23 Рис 23.Запись в файл file01.txt . . . . .	23
3.24 Рис 24.Проверка содержимого . . . . .	23
3.25 Рис 25.Проверка возможности удаления . . . . .	24
3.26 Рис 26.Вход от имени суперпользователя . . . . .	25
3.27 Рис 27.Снимаем атрибут t . . . . .	25
3.28 Рис 28.Покидаем режим суперпользователя . . . . .	26
3.29 Рис 29.Проверяем наличие атрибута t . . . . .	26
3.30 Рис 30.Повторяем предыдущие шаги . . . . .	27
3.31 Рис 31.Возвращаем атрибут . . . . .	28

# **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 механизм изменения идентификаторов

### **3 Теоретическое введение**

Unix отслеживает не символьные имена владельцев и групп, а их идентификаторы (UID - для пользователей и GID для групп). Эти идентификаторы хранятся в файлах “/etc/passwd и”/etc/group соответственно. Символьные эквиваленты идентификаторов используются только для удобства, например, при использовании команды ls, идентификаторы заменяются соответствующими символьными обозначениями[[4]].

SUID (Set User ID, Бит смены владельца) - это разрешение файловой системы Linux, которое позволяет запустить исполняемый файл от имени его владельца[[1]]. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда sudo[[2]]. Не смотря на то, что функция setuid очень полезна во многих случаях, ее неправильное использование может представлять угрозу безопасности [[2]], если атрибут setuid назначен исполняемым программам, которые не были тщательно разработаны. Из-за потенциальных проблем с безопасностью многие операционные системы игнорируют атрибут setuid при применении к исполняемым сценариям оболочки[[3]].

SGID (Set Group ID) - файл будет запускаться пользователем от имени группы, которая владеет файлом[[2]].

Флаги setuid и setgid имеют различные эффекты в зависимости от того, применяются ли они к файлу, каталогу или двоичному исполняемому или недвоичному исполняемому файлу. Флаги setuid и setgid влияют только на двоичные исполня-

емые файлы, а не на скрипты (например, Bash, Perl, Python).[[3]]

Sticky bit - это флаг права доступа пользователя, который может быть назначен файлам и каталогам в Unix-подобных системах. Существует два определения: одно для файлов, другое для каталогов.

Для файлов, особенно исполняемых файлов, суперпользователь может пометить их как сохраняемые в основной памяти, даже когда их потребность заканчивается, чтобы свести к минимуму замену, которая может произойти, когда возникнет другая необходимость, и теперь файл необходимо перезагрузить из относительно медленной вторичной памяти. Эта функция устарела из-за оптимизации подкачки.

Для каталогов, когда установлен бит привязки каталога, файловая система обрабатывает файлы в таких каталогах особым образом, чтобы только владелец файла, владелец каталога или пользователь root могли переименовать или удалить файл. Без набора бит привязки любой пользователь с разрешениями на запись и выполнение для каталога может переименовывать или удалять содержащиеся файлы независимо от владельца файла. Обычно это устанавливается в каталоге “/tmp, чтобы обычные пользователи не могли удалять или перемещать файлы других пользователей[[3]]. # Выполнение лабораторной работы

### Создание программы

1. Войдите в систему от имени пользователя guest.

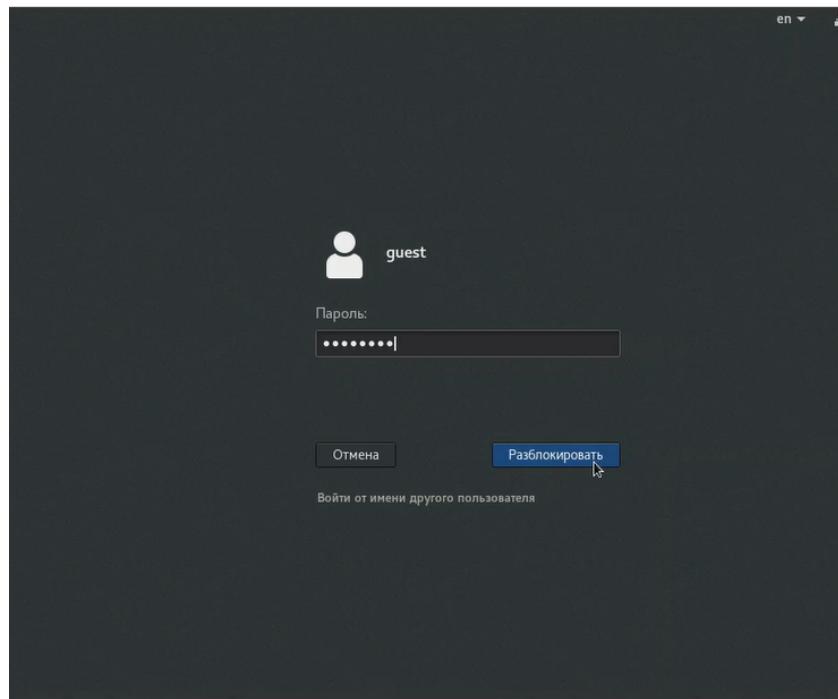
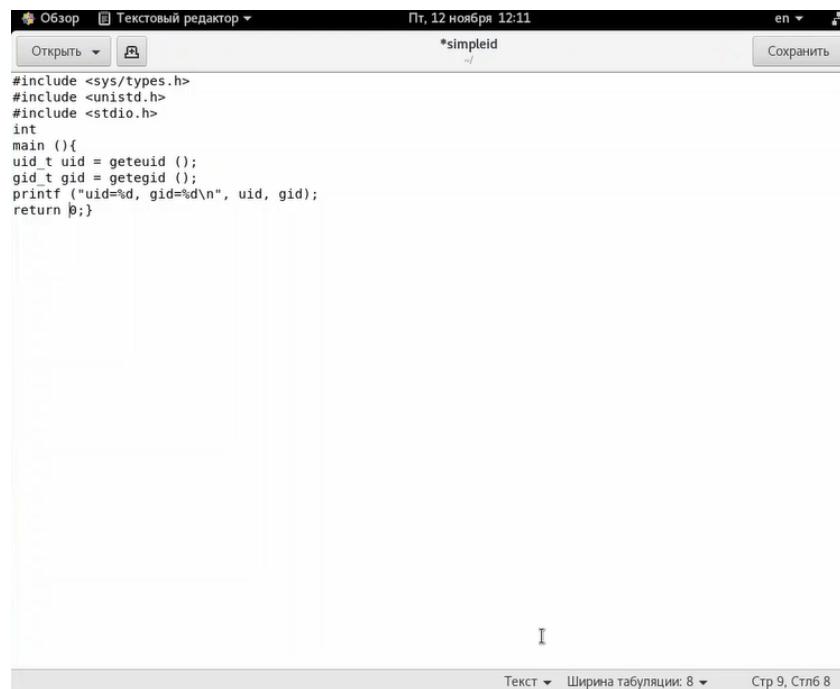


Figure 3.1: Рис 1.Вход в систему

Рисунок 1

2. Создайте программу simpleid.c:

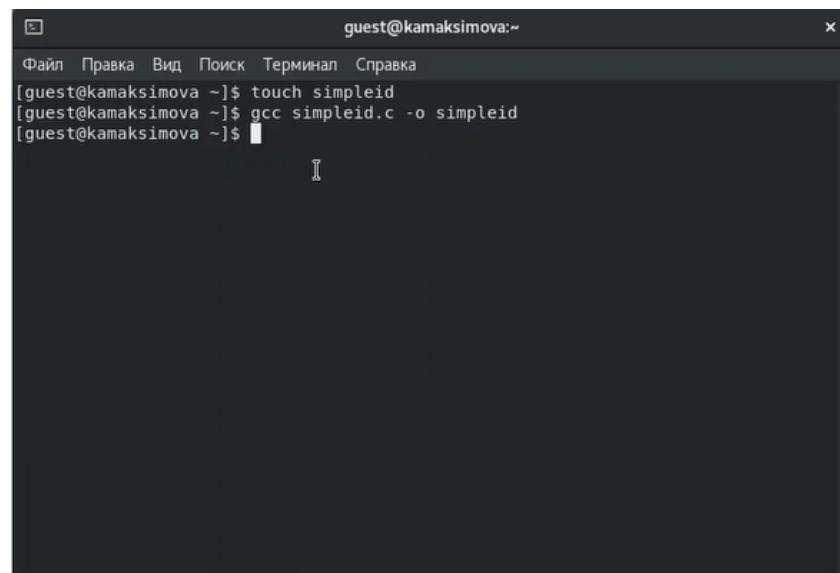


```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main (){
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Figure 3.2: Рис 2.Программа simpleid.c

## Рисунок 2

3. Скомпилируйте программу и убедитесь, что файл программы создан:



```
guest@kamaksimova:~$ touch simpleid
[guest@kamaksimova ~]$ gcc simpleid.c -o simpleid
[guest@kamaksimova ~]$
```

Figure 3.3: Рис 3.Компиляция программы

Рисунок 3

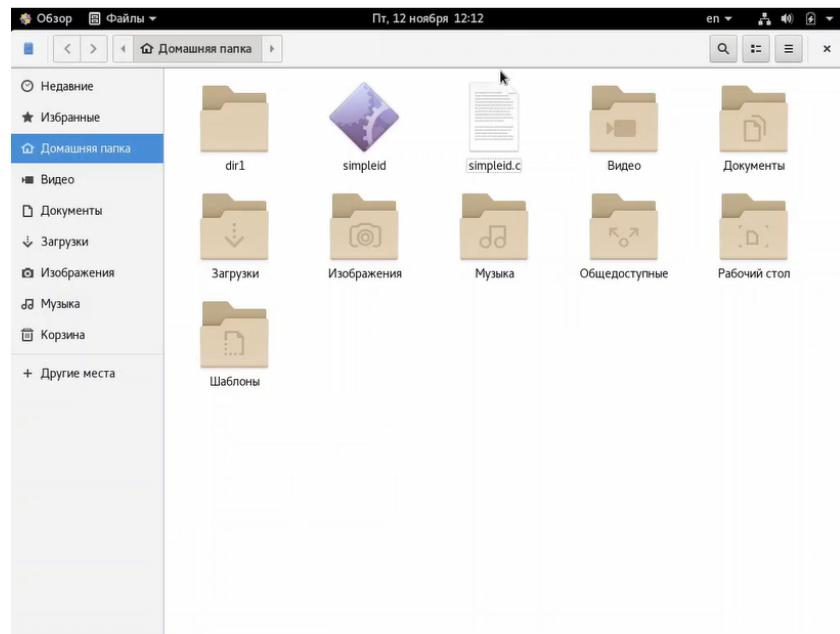


Figure 3.4: Рис 4.Файл программы

Рисунок 4

4. Выполните программу simpleid:

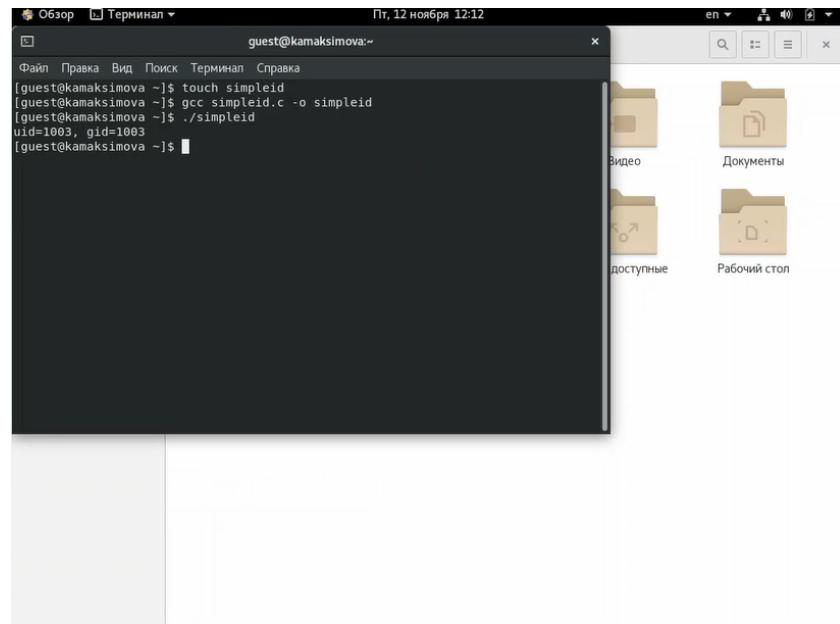


Figure 3.5: Рис 5.Выполнение программы

Рисунок 5

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

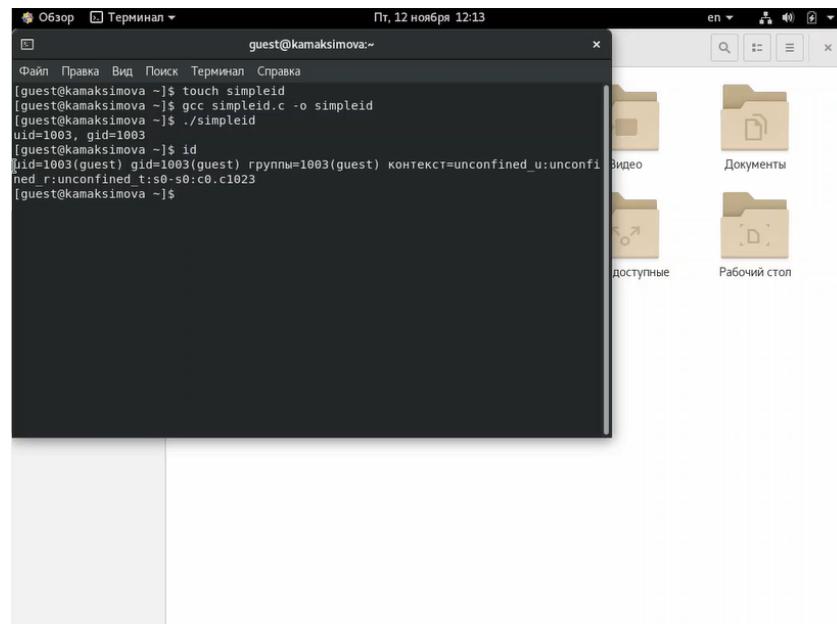


Figure 3.6: Рис 6.Команда id

Рисунок 6

Вывод программы simpleid и вывод системной программы id совпадает

6. Усложните программу, добавив вывод действительных идентификаторов:

Получившуюся программу назовите simpleid2.c.

The screenshot shows a dual-pane text editor window. The left pane contains the code for `simpleid.c`, and the right pane contains the code for `simpleid2.c`. Both files include headers for `sys/types.h`, `unistd.h`, and `stdio.h`. The `simpleid.c` file defines a `main` function that prints the effective user ID (`e_uid`) and group ID (`e_gid`). The `simpleid2.c` file also defines a `main` function that prints the real user ID (`real_uid`) and group ID (`real_gid`). The code is as follows:

```
#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 3.7: Рис 7.Программа simpleid2.c

Рисунок 7

7. Скомпилируйте и запустите `simpleid2.c`:

The screenshot shows a terminal window with a command-line interface. The user has navigated to the directory containing `simpleid2.c`. They run the command `gcc simpleid2.c -o simpleid2` to compile the program. After compilation, they execute the program with `./simpleid2`. The terminal output shows the real user ID (1003) and group ID (1003) being printed twice, once as effective and once as real.

```
Файл Правка Вид Поиск Терминал Справка
guest@guest@kamaksimova:~$ id
uid=1003, gid=1003
[guest@guest@kamaksimova ~]$ gcc simpleid2.c -o simpleid2
[guest@guest@kamaksimova ~]$ ./simpleid2
e_uid=1003, e_gid=1003
real_uid=1003, real_gid=1003
[guest@guest@kamaksimova ~]$
```

Figure 3.8: Рис 8.Компеляция simpleid2.c

Рисунок 8

8. От имени суперпользователя выполните команды:

```
chown root:guest "/home/guest/simpleid2 chmod u+s"/home/guest/simpleid2
```

The screenshot shows a terminal window titled "Терминал" (Terminal) with the command prompt "guest@kamaksimova:~". The terminal displays the following session:

```
guest@kamaksimova:~$ gcc simpleid2.c -o simpleid2
guest@kamaksimova:~$ ./simpleid2
e_uid=1003, e_gid=1003
[guest@kamaksimova ~]$ su -
Password:
[root@kamaksimova ~]# chown root:guest /home/guest/simpleid2
[root@kamaksimova ~]# chmod u+s /home/guest/simpleid2
[root@kamaksimova ~]# exit
выход
[guest@kamaksimova ~]$
```

Below the terminal window, a tooltip provides instructions for verification:

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. Проделайте тоже самое относительно SetGID-бита.
13. Создайте программу readfile.c:

```
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
```

Figure 3.9: Рис 9.Команды chown и chmod

Рисунок 9

9. Используйте sudo или повысьте временно свои права с помощью su. Поясните, что делают эти команды. Команда: chown root:guest "/home/guest/simpleid2" меняет владельца файла simpleid2 Команда: chmod u+s"/home/guest/simpleid2" меняет атрибуты на файле simpleid2

10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```

guest@kamaksimova:~$ gcc simpleid2.c -o simpleid2
[guest@kamaksimova ~]$ ./simpleid2
e_uid=1003, e_gid=1003
real uid=1003, real gid=1003
[guest@kamaksimova ~]$ su -
Пароль:
[root@kamaksimova ~]# chown root:guest /home/guest/simpleid2
[root@kamaksimova ~]# chmod u+s /home/guest/simpleid2
[root@kamaksimova ~]# exit
выход
[guest@kamaksimova ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 12 12:16 simpleid2
[guest@kamaksimova ~]$ id
10. Выполните проверку правильности установки новых атрибутов
владельца файла simpleid2:
ls -l simpleid2
11. Запустите simpleid2 и id:
./simpleid2
id
Сравните результаты.
12. Проверьте 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>

```

Figure 3.10: Рис 10.Проверка правильности установки новых атрибутов

Рисунок 10

11. Запустите simpleid2 и id: Сравните результаты.

```

guest@kamaksimova:~$ ./simpleid2
e_uid=1003, e_gid=1003
real_uid=1003, real_gid=1003
[guest@kamaksimova ~]$ su -
Пароль:
[root@kamaksimova ~]# chown root:guest /home/guest/simpleid2
[root@kamaksimova ~]# chmod u+s /home/guest/simpleid2
[root@kamaksimova ~]# exit
выход
[guest@kamaksimova ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 12 12:16 simpleid2
[guest@kamaksimova ~]$ ./simpleid2
e_uid=0, e_gid=1003
real_uid=1003, real_gid=1003
[guest@kamaksimova ~]$ id
#id=1003(guest) gid=1003(group) группы=1003(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s=0:s=0:c1023
[guest@kamaksimova ~]$ id
10. Выполните проверку правильности установки новых атрибутов
владельца файла simpleid2:
ls -l simpleid2
11. Запустите simpleid2 и id:
./simpleid2
id
Сравните результаты.
12. Проверьте 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>

```

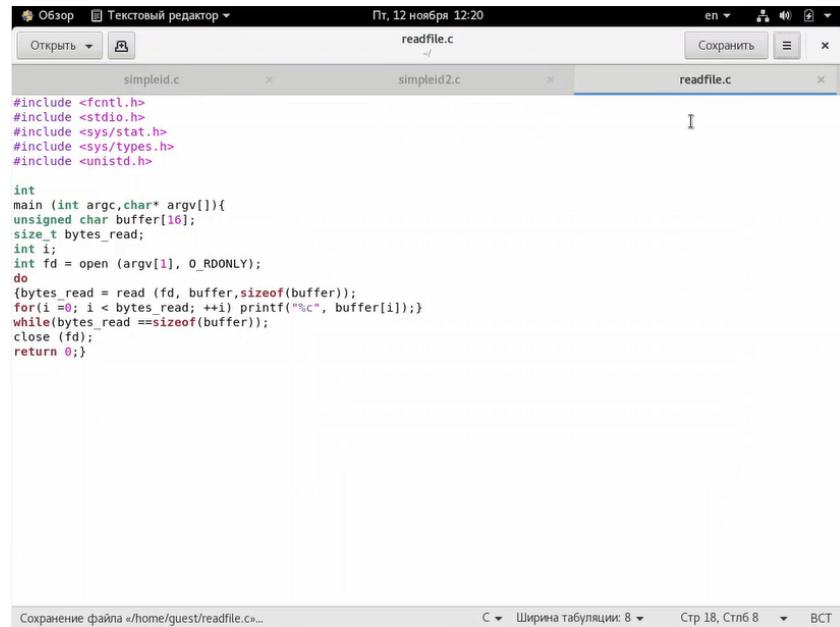
Figure 3.11: Рис 11.Результаты simpleid2 и id

Рисунок 11

Результаты вывода программ simpleid2 и id совпадают

12. Проделайте 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[10];
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;}  
Сохранение файла «/home/guest/readfile.c»... Сохранить
```

Figure 3.12: Рис 12.Программа readfile.c

Рисунок 12

14. Откомпилируйте её.

```

Пт, 12 ноября 12:21
005-lab_discret_sticky_pc | + https://esystem.rudn.ru/pluginfile.php/1198308/mod_resource/content/2 ...
5 из 7 - + Автоматически
{
    unsigned char buffer[16];
}

guest@kamaksimova:~$

Файл Правка Вид Поиск Терминал Справка
printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
e_uid
[guest@kamaksimova ~]$ gcc simpleid2.c -o simpleid2
[guest@kamaksimova ~]$ ./simpleid2
e_uid=1003, e_gid=1003
real_uid=1003, real_gid=1003
[guest@kamaksimova ~]$ su -
Пароль:
[root@kamaksimova ~]# chown root:guest /home/guest/simpleid2
[root@kamaksimova ~]# chmod u+s /home/guest/simpleid2
[root@kamaksimova ~]# exit
выход
[guest@kamaksimova ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 12 12:16 simpleid2
[guest@kamaksimova ~]$ ./simpleid2
e_uid=0, e_gid=1003
real_uid=1003, real_gid=1003
[guest@kamaksimova ~]$ id
uid=1003(guest) gid=1003(guest) группы=1003(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@kamaksimova ~]$ touch readfile
[guest@kamaksimova ~]$ gcc readfile.c -o readfile
[guest@kamaksimova ~]$ 

```

Figure 3.13: Рис 13.Компиляция readfile.c

### Рисунок 13

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

```

root@kamaksimova:~$ 
root@kamaksimova:~$ return 0;
root@kamaksimova:~$ 

Файл Правка Вид Поиск Терминал Справка
[root@kamaksimova ~]# exit
выход
[guest@kamaksimova ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 12 12:16 simpleid2
[guest@kamaksimova ~]$ ./simpleid2
e_uid=0, e_gid=1003
real_uid=1003, real_gid=1003
[guest@kamaksimova ~]$ id
uid=1003(guest) gid=1003(guest) группы=1003(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@kamaksimova ~]$ touch readfile
[guest@kamaksimova ~]$ gcc readfile.c -o readfile
[guest@kamaksimova ~]$ chown root:guest /home/guest/simpleid2
chown: изменение владельца '/home/guest/simpleid2': Операция не позволена
[guest@kamaksimova ~]$ su -
Пароль:
su: Свой при проверке подлинности
[guest@kamaksimova ~]$ su -
Пароль:
[root@kamaksimova ~]# chown root:guest /home/guest/readfile.c
[root@kamaksimova ~]# chmod 700 readfile.c
chmod: невозможно получить доступ к 'readfile.c': Нет такого файла или каталога
[root@kamaksimova ~]# chmod 700 /home/guest/readfile.c
[root@kamaksimova ~]# 

#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

```

Figure 3.14: Рис 14.Изменение владельца файла readfile.c и изменение прав доступа

Рисунок 14

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

The screenshot shows a terminal window titled 'root@kamaksimova:~'. The terminal displays the following command-line session:

```
return 0;
root@kamaksimova:~#
Файл Правка Вид Поиск Терминал Справка
e_uid=0, e_gid=1003
real_uid=1003, real_gid=1003
[guest@kamaksimova ~]$ id
uid=1003(guest) gid=1003(guest) группы=1003(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c023
[guest@kamaksimova ~]$ touch readfile
[guest@kamaksimova ~]$ gcc readfile.c -o readfile
[guest@kamaksimova ~]$ chown root:guest /home/guest/simpleid2
chown: изменение владельца '/home/guest/simpleid2': Операция не позволена
[guest@kamaksimova ~]$ su -
Пароль:
su: Свой при проверке подлинности
[guest@kamaksimova ~]$ su -
Пароль:
[root@kamaksimova ~]# chown root:guest /home/guest/readfile.c
[root@kamaksimova ~]# chmod 700 readfile.c
chmod: невозможно получить доступ к 'readfile.c': Нет такого файла или каталога
[root@kamaksimova ~]# chmod 700 /home/guest/readfile.c
[root@kamaksimova ~]# ls -l /home/guest/readfile.c
ls: невозможно получить доступ к '/home/guest/readfile.c': Нет такого файла или каталога
[root@kamaksimova ~]# ls -l /home/guest/readfile.c
-rwx-----. 1 root guest 392 ноя 12 12:20 /home/guest/readfile.c
[root@kamaksimova ~]#
```

At the bottom of the terminal window, there is a code snippet for a C program:

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

Figure 3.15: Рис 15.Проверка доступа

Рисунок 15

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

18. Проверьте, может ли программа readfile прочитать файл readfile.c

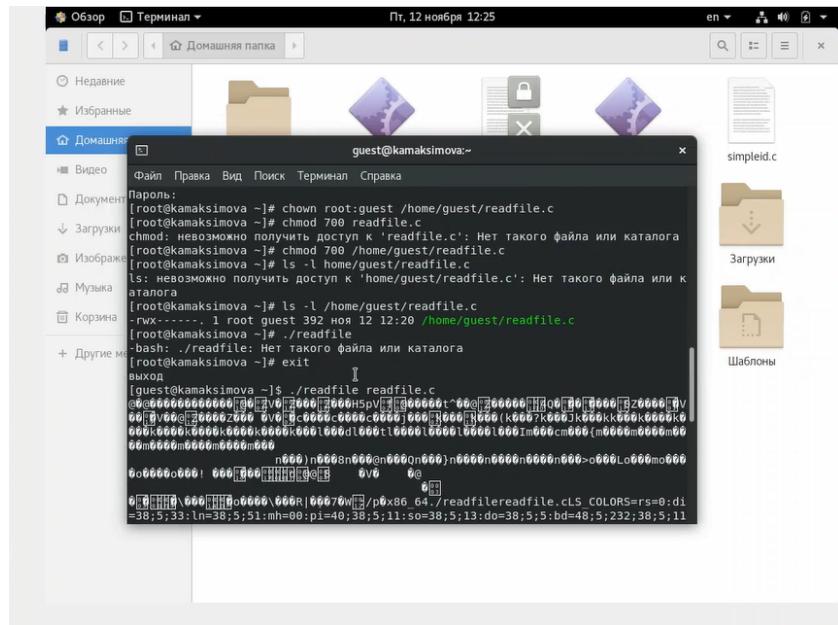


Figure 3.16: Рис 16.Проверка

Рисунок 16

19. Проверьте, может ли программа readfile прочитать файл “/etc/shadow”

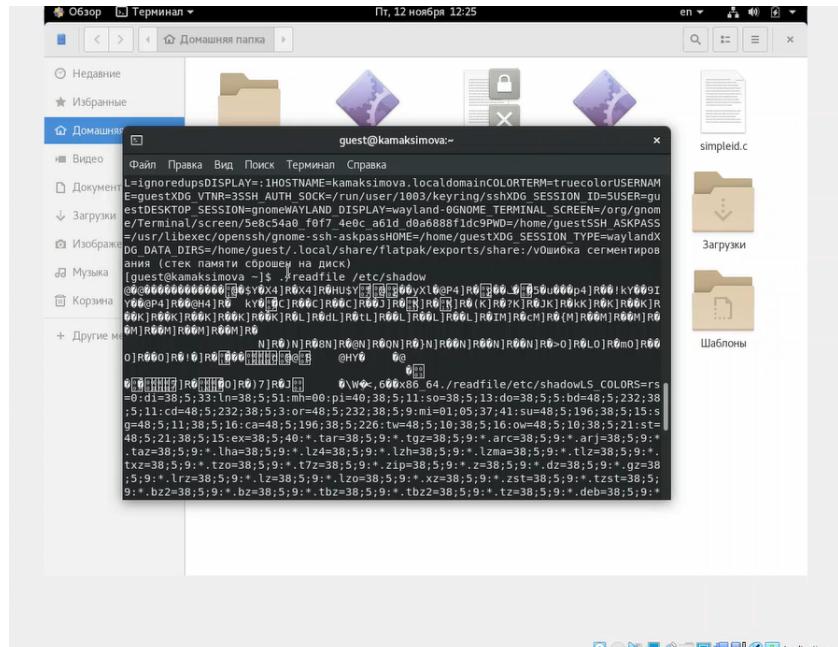


Figure 3.17: Рис 17.Проверка

Рисунок 17

### Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории “/tmp

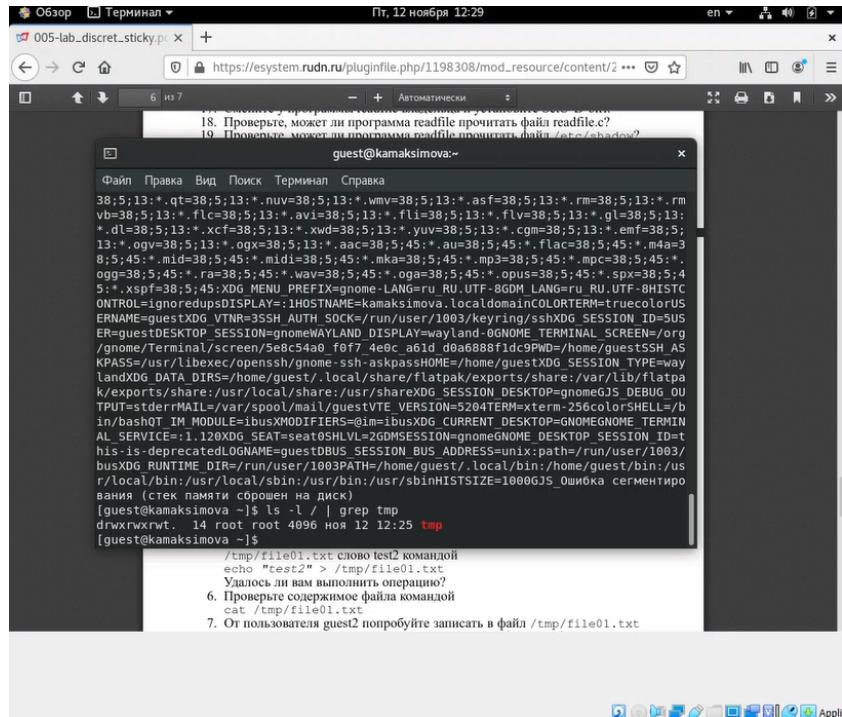


Figure 3.18: Рис 18.Проверка наличия атрибута

Рисунок 18

2. От имени пользователя guest создайте файл file01.txt в директории “/tmp со словом test:

```

18. Проверьте, может ли программа readfile прочитать файл readfile.c?
19. Проверьте, может ли программа readfile прочитать файл /etc/shadow?
20. Успешно ли был создан файл /tmp/file01.txt?
21. Правильны ли содержимое файла file01.txt?
22. У пользователя guest2 попробуйте записать в файл /tmp/file01.txt

```

Figure 3.19: Рис 19.Создание файла file01.txt

Рисунок 19

3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»:

Пт, 12 ноября 12:30

https://esystem.rudn.ru/pluginfile.php/1198308/mod\_resource/content/2

```

guest@kamaksimova:~>

Файл Правка Вид Поиск Терминал Справка
огр=38;5;45-*.. ga=38;5;45-*.. wa=38;5;45-*.. og=38;5;45-*.. opus=38;5;45-*.. spx=38;5;4
5-*.. xsp=38;5;45-*.. :DG MENU=PREFIX=gnome_LANG=ru_RU.UTF-8GDM LANG=ru_RU.UTF-8HISTC
ONTROL=ignoreupsDISPLAY=:1HOSTNAME=kamaksimova.localdomainCOLORTERM=truecolorUS
ERNAME=guestXDG_VTNR=3SSH_AUTH_SOCK=/run/user/1003/keyring/shXDG_SESSION_ID=5US
ER=guestDESKTOP_SESSION=gnomeWAYLAND DISPLAY=wayland-0GNOME_TERMINAL SCREEN=/org
/gnome/Terminal/screen/5e8c54a0_f0f7_4e0c_a61d_daa688ff1dc9PWD=/home/guestSSH_AS
KPASS=/usr/libexec/openssh/gnome-ssh-askpassHOME=/home/guestXDG_SESSION_TYPE=way
landXDG_DATA_DIRS=/home/guest/.local/share/flatpak(exports/share:/var/lib/flatpa
k(exports/share:/usr/local/share:/usr/share)SESSION_DESKTOP=gnomeGJS_DEBUG_OU
TPUT=stderMAIL=/var/spool/mail/guestVT VERSION=5204TERM=xterm-256colorSHELL=b
in/bashQT_IM_MODULE=ibusXMODIFIERS=@im=ibusXDG_CURRENT_DESKTOP=GNOME GNOME TERMIN
AL_SERVICE=@.1.120XDG_SEAT=seat0SHVL=2GMSSESSION=gnomeGNOME_DESKTOP SESSION_ID=t
his_is_deprecatedLOGNAME=guestDBUS SESSION_BUS_ADDRESS=unix:path=/run/user/1003/
busXDG_RUNTIME_DIR=/run/user/1003PATH=/home/guest/.local/bin:/home/guest/bin:/us
r/local/bin:/usr/local/sbin:/usr/bin:/usr/sbinHISTORYSIZE=1000GJS_Ошибка сегментиро
вания (стек памяти сорван на диск)
[guest@kamaksimova ~]$ ls -l / | grep tmp
drwxrwxrwt. 14 root root 4096 ноя 12 12:25 tmp
[guest@kamaksimova ~]$ echo "test" > /tmp/file01.txt
bash: echotest: команда не найдена...
[guest@kamaksimova ~]$ ls -l /tmp/file01.txt
[guest@kamaksimova ~]$ cat /tmp/file01.txt
7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt
[guest@kamaksimova ~]$
```

/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

Figure 3.20: Рис 20. Атрибуты файла file01.txt

Рисунок 20

Пт, 12 ноября 13:23

https://esystem.rudn.ru/pluginfile.php/1198308/mod\_resource/content/2

```

guest@kamaksimova:~>

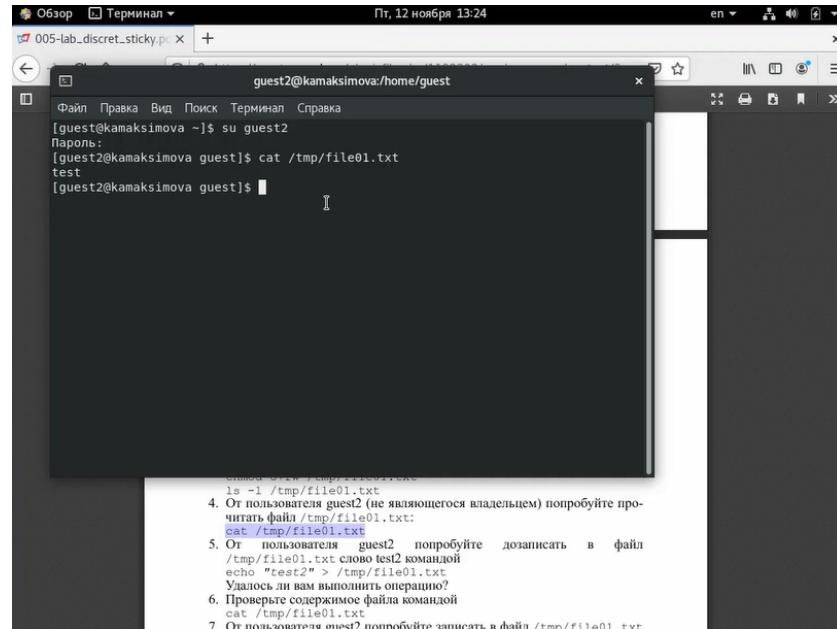
Файл Правка Вид Поиск Терминал Справка
огр=38;5;45-*.. ga=38;5;45-*.. wa=38;5;45-*.. og=38;5;45-*.. opus=38;5;45-*.. spx=38;5;4
5-*.. xsp=38;5;45-*.. :DG MENU=PREFIX=gnome_LANG=ru_RU.UTF-8GDM LANG=ru_RU.UTF-8HISTC
ONTROL=ignoreupsDISPLAY=:1HOSTNAME=kamaksimova.localdomainCOLORTERM=truecolorUS
ERNAME=guestXDG_VTNR=3SSH_AUTH_SOCK=/run/user/1003/keyring/shXDG_SESSION_ID=5US
ER=guestDESKTOP_SESSION=gnomeWAYLAND DISPLAY=wayland-0GNOME_TERMINAL SCREEN=/org
/gnome/Terminal/screen/5e8c54a0_f0f7_4e0c_a61d_daa688ff1dc9PWD=/home/guestSSH_AS
KPASS=/usr/libexec/openssh/gnome-ssh-askpassHOME=/home/guestXDG_SESSION_TYPE=way
landXDG_DATA_DIRS=/home/guest/.local/share/flatpak(exports/share:/var/lib/flatpa
k(exports/share:/usr/local/share:/usr/share)SESSION_DESKTOP=gnomeGJS_DEBUG_OU
TPUT=stderMAIL=/var/spool/mail/guestVT VERSION=5204TERM=xterm-256colorSHELL=b
in/bashQT_IM_MODULE=ibusXMODIFIERS=@im=ibusXDG_CURRENT_DESKTOP=GNOME GNOME TERMIN
AL_SERVICE=@.1.120XDG_SEAT=seat0SHVL=2GMSSESSION=gnomeGNOME_DESKTOP SESSION_ID=t
his_is_deprecatedLOGNAME=guestDBUS SESSION_BUS_ADDRESS=unix:path=/run/user/1003/
busXDG_RUNTIME_DIR=/run/user/1003PATH=/home/guest/.local/bin:/home/guest/bin:/us
r/local/bin:/usr/local/sbin:/usr/bin:/usr/sbinHISTORYSIZE=1000GJS_Ошибка сегментиро
вания (стек памяти сорван на диск)
[guest@kamaksimova ~]$ ls -l / | grep tmp
drwxrwxrwt. 14 root root 4096 ноя 12 12:25 tmp
[guest@kamaksimova ~]$ echo "test" > /tmp/file01.txt
bash: echotest: команда не найдена...
[guest@kamaksimova ~]$ ls -l /tmp/file01.txt
[guest@kamaksimova ~]$ cat /tmp/file01.txt
[guest@kamaksimova ~]$ chmod o+r /tmp/file01.txt
[guest@kamaksimova ~]$ ls -l /tmp/file01.txt
[guest@kamaksimova ~]$ echo "test2" > /tmp/file01.txt
[guest@kamaksimova ~]$ cat /tmp/file01.txt
7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt
[guest@kamaksimova ~]$
```

/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

Figure 3.21: Рис 21. Атрибуты файла file01.txt

Рисунок 21

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл “/tmp/file01.txt”:



```
guest2@kamaksimova:~$ ls -l /tmp/file01.txt
-rw-r--r-- 1 root root 4 2015-01-01 12:00 /tmp/file01.txt
guest2@kamaksimova:~$ cat /tmp/file01.txt
test
guest2@kamaksimova:~$
```

The terminal window shows the user 'guest2' at the prompt. The user runs the command 'cat /tmp/file01.txt'. The output of the command is 'test', which is the content of the file. The terminal window has a dark theme with white text.

Figure 3.22: Рис 22.Чтение файла file01.txt

Рисунок 22

5. От пользователя guest2 попробуйте дозаписать в файл “/tmp/file01.txt” слово test2

```

[guest2@kamaksimova ~]$ su guest2
Пароль:
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
bash: echotest2: команда не найдена...
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
[guest2@kamaksimova guest]$

```

Синтаксис: 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

Figure 3.23: Рис 23.Запись в файл file01.txt

Рисунок 23

## 6. Проверьте содержимое файла

```

[guest2@kamaksimova ~]$ su guest2
Пароль:
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
bash: echotest2: команда не найдена...
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test2
[guest2@kamaksimova guest]$

```

Синтаксис: 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

Figure 3.24: Рис 24.Проверка содержимого

Рисунок 24

9. От пользователя guest2 попробуйте удалить файл “/tmp/file01.txt”

The screenshot shows a terminal window titled "guest2@kamaksimova/home/guest". The terminal session is as follows:

```
[guest@kamaksimova ~]$ su guest2
Пароль:
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
bash: echotest2: команда не найдена...
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test2
[guest2@kamaksimova guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@kamaksimova guest]$ rm tmp/file01.txt
rm: невозможно удалить 'tmp/file01.txt': Отказано в доступе
```

Below the terminal window, there is a tooltip with the following text:

Удалось ли вам выполнить операцию?  
8. Проверьте содержимое файла командой  
cat /tmp/file01.txt  
9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt команной  
rm /tmp/file01.txt  
Удалось ли вам удалить файл?  
10. Повысьте свои права до суперпользователя следующей командой  
su -  
и выполните после этого команду, снимающую атрибут t (Sticky-bit) с директории /tmp:

Figure 3.25: Рис 25.Проверка возможности удаления

Рисунок 25

10. Повысьте свои права до суперпользователя и выполните после этого команду, снимающую атрибут t (Sticky-bit) с директории “/tmp”:

```

root@kamaksimova:~#
[guest@kamaksimova guest] $ su guest2
Пароль:
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
bash: echotest2: команда не найдена...
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test2
[guest2@kamaksimova guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@kamaksimova guest]$ rm tmp/file01.txt
rm: невозможно удалить 'tmp/file01.txt': Отказано в доступе
[guest2@kamaksimova guest]$ su -
Пароль:
[root@kamaksimova ~]# 

```

Если вышел > /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 следующей командой  
`su -`  
и выполните после этого команду, снимающую атрибут t (Sticky-bit) с директории /tmp:

Figure 3.26: Рис 26.Вход от имени суперпользователя

Рисунок 26

```

root@kamaksimova:~#
[guest@kamaksimova ~] $ su guest2
Пароль:
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
bash: echotest2: команда не найдена...
[guest2@kamaksimova guest]$ echo "test2">> /tmp/file01.txt
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
test2
[guest2@kamaksimova guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@kamaksimova guest]$ rm tmp/file01.txt
rm: невозможно удалить 'tmp/file01.txt': Отказано в доступе
[guest2@kamaksimova guest]$ su -
Пароль:
[root@kamaksimova ~]# chmod -t /tmp
[root@kamaksimova ~]# 

```

Если вышел > /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 следующей командой  
`su -`  
и выполните после этого команду, снимающую атрибут t (Sticky-bit) с директории /tmp:

Figure 3.27: Рис 27.Снимаем атрибут t

Рисунок 27

## 11. Покиньте режим суперпользователя

```

guest2@kamaksimova:/home/guest
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@kamaksimova guest]$ rm tmp/file01.txt
rm: невозможно удалить 'tmp/file01.txt': Отказано в доступе
[guest2@kamaksimova guest]$ su -
Пароль:
[root@kamaksimova ~]# chmod -t /tmp
[root@kamaksimova ~]# ls -l /tmp
итого 4
-rw-rw-rw-. 1 guest guest 6 ноя 12 13:24 file01.txt
drwx----- 3 root root 17 ноя 12 09:28 systemd-private-49958a3c6f
ce427197fc7a8647f72bc8-colord.service-zyl91
drwx----- 3 root root 17 ноя 12 09:28 systemd-private-49958a3c6f
ce427197fc7a8647f72bc8-ModemManager.service-b5XK01
drwx----- 3 root root 17 ноя 12 09:28 systemd-private-49958a3c6f
ce427197fc7a8647f72bc8-rtkit-daemon.service-4q2xTf
drwx----- 2 guest guest 6 ноя 12 09:37 Temp-38d31555-c564-42cf-ad
51-f6c6fa893f7f
drwx----- 3 guest guest 31 ноя 12 09:37 Temp-64eb632f-84b8-42c6-bf
f8-92739562094
drwx----- 2 kamaksimova kamaksimova 6 окт 12 19:37 tracker-extract-files.1000
drwx----- 2 guest guest 6 ноя 12 12:23 tracker-extract-files.1003
[root@kamaksimova ~]# exit
выход
[guest2@kamaksimova guest]$ 
```

Удалось ли вам выполнить операцию?

- Проверьте содержимое файла командой `cat /tmp/file01.txt`
- От пользователя `guest2` попробуйте удалить файл `/tmp/file01.txt` командой `rm /tmp/file01.txt`
- Удалось ли вам удалить файл?
- Повысьте свои права до суперпользователя следующей командой `su -` и выполните после этого команду, снимающую атрибут `t` (Sticky-bit) с директории `/tmp`:

Figure 3.28: Рис 28. Покидаем режим суперпользователя

Рисунок 28

12. От пользователя `guest2` проверьте, что атрибута `t` у директории `"/tmp"` нет:

```

guest2@kamaksimova:/home/guest
rm: невозможно удалить '/tmp/file01.txt': Отказано в доступе
[guest2@kamaksimova guest]$ su -
Пароль:
[root@kamaksimova ~]# chmod -t /tmp
[root@kamaksimova ~]# ls -l /tmp
итого 4
-rw-rw-rw-. 1 guest guest 6 ноя 12 13:24 file01.txt
drwx----- 3 root root 17 ноя 12 09:28 systemd-private-49958a3c6f
ce427197fc7a8647f72bc8-colord.service-zyl91
drwx----- 3 root root 17 ноя 12 09:28 systemd-private-49958a3c6f
ce427197fc7a8647f72bc8-ModemManager.service-b5XK01
drwx----- 3 root root 17 ноя 12 09:28 systemd-private-49958a3c6f
ce427197fc7a8647f72bc8-rtkit-daemon.service-4q2xTf
drwx----- 2 guest guest 6 ноя 12 09:37 Temp-38d31555-c564-42cf-ad
51-f6c6fa893f7f
drwx----- 3 guest guest 31 ноя 12 09:37 Temp-64eb632f-84b8-42c6-bf
f8-92739562094
drwx----- 2 kamaksimova kamaksimova 6 окт 12 19:37 tracker-extract-files.1000
drwx----- 2 guest guest 6 ноя 12 12:23 tracker-extract-files.1003
[root@kamaksimova ~]# exit
выход
[guest2@kamaksimova guest]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 ноя 12 13:26 tmp
[guest2@kamaksimova guest]$ 
```

11. Покиньте режим суперпользователя командой `exit`

- От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет:
- Повторите предыдущие шаги. Какие наблюдаются изменения?
- Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.

Figure 3.29: Рис 29. Проверяем наличие атрибута t

Рисунок 29

### 13. Повторите предыдущие шаги.

The screenshot shows a terminal window titled "guest2@kamaksimova:/home/guest". The terminal displays the following command-line session:

```
[guest2@kamaksimova guest]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 ноя 12 13:26 tmp
[guest2@kamaksimova guest]$ echo "2"> /tmp/file01.txt
[guest2@kamaksimova guest]$ echo "2"> /tmp/file01.txt
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
2
[guest2@kamaksimova guest]$ rm tmp/file01.txt
rm: невозможно удалить 'tmp/file01.txt': Отказано в доступе
[guest2@kamaksimova guest]$ su
[суперпользовательский режим]
    chmod -t /tmp
11. Покиньте режим суперпользователя командой
exit
12. От пользователя guest2 проверьте, что атрибут 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.
```

Figure 3.30: Рис 30.Повторяем предыдущие шаги

Рисунок 30

14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Нет, файл удалить не удалось
15. Повысьте свои права до суперпользователя и верните атрибут t на директорию “/tmp”:

```
guest2@kamaksimova:~$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 ноя 12 13:26 tmp
[guest2@kamaksimova guest]$ echo "2"> /tmp/file01.txt
[guest2@kamaksimova guest]$ cat /tmp/file01.txt
2
[guest2@kamaksimova guest]$ rm tmp/file01.txt
rm: невозможно удалить 'tmp/file01.txt': Отказано в доступе
[guest2@kamaksimova guest]$ su -
Пароль:
su: Сбой при проверке подлинности
[guest2@kamaksimova guest]$ su -
Пароль:
[root@kamaksimova ~]# chmod +t /tmp
[root@kamaksimova ~]# exit
выход
[guest2@kamaksimova guest]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 ноя 12 13:26 tmp
11. Покиньте режим суперпользователя командой
exit.
12. От пользователя guest2 проверьте, что атрибут 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 отчет.
```

Figure 3.31: Рис 31.Возвращаем атрибут

Рисунок 31

## **4 Выводы**

Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрены работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# **Список литературы**

1. Повышение привилегий в ОС Linux через SUID/SGID
2. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux
3. Setuid
4. Права доступа Unix, SUID, SGID, Sticky биты