

Дисcretionное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Основы информационной безопасности

Симонова Полина Игоревна

Содержание

1 Цель работы	5
2 Теоретическое введение	6
3 Выполнение лабораторной работы	8
4 Выводы	20
5 Список литературы	21

Список иллюстраций

3.1	Подготовка к лабораторной работе	8
3.2	Вход от имени пользователя guest	8
3.3	Создание файла	9
3.4	Содержимое файла	9
3.5	Компиляция файла	10
3.6	Сравнение команд	10
3.7	Создание и компиляция файла	10
3.8	Содержимое файла	11
3.9	Смена владельца файла и прав доступа к файлу	12
3.10	Запуск файла	12
3.11	Создание и компиляция файла	12
3.12	Содержимое файла	14
3.13	Смена владельца файла и прав доступа к файлу	14
3.14	Попытка прочесть содержимое файла	15
3.15	Попытка прочесть содержимое файла программой	15
3.16	Попытка прочесть содержимое файла программой	15
3.17	Чтение файла от имени суперпользователя	16
3.18	Проверка атрибутов директории tmp	16
3.19	Создание файла, изменение прав доступа	16
3.20	Попытка чтения файла	17
3.21	Попытка записи в файл	17
3.22	Попытка удалить файл	17
3.23	Смена атрибутов файла	17
3.24	Проверка атрибутов директории	18
3.25	Повтор предыдущих действий	18
3.26	Изменение атрибутов	19

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 Теоретическое введение

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута.

Sticky bit

Используется в основном для каталогов, чтобы защитить в них файлы. В такой каталог может писать любой пользователь. Но, из такой директории пользователь может удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

SUID (Set User ID)

Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ к важным системным файлам, которые принадлежат пользователю root.

SGID (Set Group ID)

Аналогичен `suid`, но относиться к группе. Если установить `sgid` для каталога, то все файлы созданные в нем, при запуске будут принимать идентификатор группы

каталога, а не группы владельца, который создал файл в этом каталоге.

Обозначение атрибутов sticky, suid, sgid

Специальные права используются довольно редко, поэтому при выводе программы ls -l символ, обозначающий указанные атрибуты, закрывает символ стандартных прав доступа.

Пример: rwsrwsrwt

где первая s — это suid, вторая s — это sgid, а последняя t — это sticky bit

В приведенном примере не понятно, rwt — это rw- или rwx? Определить это просто. Если t маленькое, значит x установлен. Если T большое, значит x не установлен. То же самое правило распространяется и на s.

В числовом эквиваленте данные атрибуты определяются первым символом при четырехзначном обозначении (который часто опускается при назначении прав), например в правах 1777 — символ 1 обозначает sticky bit.

2. Компилятор GCC

GCC - это свободно доступный оптимизирующий компилятор для языков C, C++. Собственно программа gcc это некоторая надстройка над группой компиляторов, которая способна анализировать имена файлов, передаваемые ей в качестве аргументов, и определять, какие действия необходимо выполнить. Файлы с расширением .cc или .C рассматриваются, как файлы на языке C++, файлы с расширением .c как программы на языке C, а файлы с расширением .o считаются объектными [gcc].

3 Выполнение лабораторной работы

Для лабораторной работы необходимо проверить, установлен ли компилятор gcc, команда `gcc -v` позволяет это сделать. Также осуществляется отключение системы запретом с помощью `setenforce 0` (рис. 1).

```
pisimonova@pisimonova:~$ gcc -v
bash: gcc: команда не найдена...
Установить пакет «gcc», предоставляющий команду «gcc»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов...
Следующие пакеты должны быть установлены:
cpp-14.3.1-2.1.el10.x86_64      The C Preprocessor
gcc-14.3.1-2.1.el10.x86_64      Various compilers (C, C++, Objective-C, ...)
glibc-devel-2.39-58.el10_1.2.x86_64   Object files for development using standard C libraries.
kernel-headers-6.12.0-124.31.1.el10_1.x86_64   Header files for the Linux kernel for use by glibc
libcrypt-devel-4.4.36-10.el10.x86_64   Development files for libcrypt
Подолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/14/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:amdgcn-amdhsa
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org
/
```

Рисунок 3.1: Подготовка к лабораторной работе

Осуществляется вход от имени пользователя guest (рис. 2).

```
pisimonova@pisimonova:~$ su guest
Пароль:
```

Рисунок 3.2: Вход от имени пользователя guest

Создание файла simpleid.c и запись в файл кода (рис. 3)

```
guest@pisimonova:~$ touch simpleid.c
guest@pisimonova:~$ nano simpleid.c
guest@pisimonova:~$ gcc simpleid.c -o simpleid
```

Рисунок 3.3: Создание файла

```
#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;
}
```

Листинг 1

Содержимое файла выглядит следующим образом (рис. 4)



The screenshot shows a terminal window titled "guest@pisimonova:~ – nano simpleid.c /home/guest". The window displays the C code from Listing 1. The code includes standard library headers, a main function that prints the user's effective UID and GID, and a return statement. The code is highlighted with color-coded syntax.

```
GNU nano 8.1
#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;
}
```

Рисунок 3.4: Содержимое файла

Компилирую файл, проверяю, что он скомпилировался (рис. 5)

```
guest@pisimonova:~$ gcc simpleid.c -o simpleid
guest@pisimonova:~$ ls
dir1      simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
simpleid  Видео       Загрузки   Музыка        'Рабочий стол'
```

Рисунок 3.5: Компиляция файла

Запускаю исполняемый файл. В выводе файла выписаны номера пользователя и групп, от вывода при вводе if, они отличаются только тем, что информации меньше (рис. 6)

```
guest@pisimonova:~$ ./simpleid
uid=1001, gid=1001
guest@pisimonova:~$ id
uid=1001(guest) gid=1001(guest) группа=1001(guest) контекст=unconfined_u:un
confined_r:unconfined_t:s0-s0:c0.c1023
```

Рисунок 3.6: Сравнение команд

Создание, запись в файл и компиляция файла simpleid2.c. Запуск программы (рис. 7)

```
guest@pisimonova:~$ touch simpleid2.c
guest@pisimonova:~$ nano simpleid2.c
guest@pisimonova:~$ gcc simpleid2.c -o simpleid2
simpleid2.c: В функции «main»:
simpleid2.c:13:11: ошибка: в программе обнаружен некорректный символ «\342»
  13 | real_gid);<U+21AA><U+2192>
      ^~~~~~
simpleid2.c:13:12: ошибка: в программе обнаружен некорректный символ «\342»
  13 | real_gid);<U+21AA><U+2192>
      ^~~~~~
guest@pisimonova:~$ nano simpleid2.c
guest@pisimonova:~$ gcc simpleid2.c -o simpleid2
guest@pisimonova:~$ ls
dir1      simpleid2.c  Документы  Изображения  Общедоступные  Шаблоны
simpleid  simpleid.c  Загрузки   Музыка        'Рабочий стол'
simpleid2  Видео       Иконки      'Рабочий стол'
guest@pisimonova:~$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Рисунок 3.7: Создание и компиляция файла

```
#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;
}

```

Листинг 2

(рис. 8)

The screenshot shows a terminal window titled "guest@pisimonova:~ - nano simpleid2.c" with the path "/home/guest". The window displays the following C code:

```

GNU nano 8.1           simpleid2.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;
}

```

Рисунок 3.8: Содержимое файла

С помощью chown изменяю владельца файла на суперпользователя, с помощью chmod изменяю права доступа (рис. 9)

```
pisimonova@pisimonova:~$ sudo chown root:guest /home/guest/simpleid2
pisimonova@pisimonova:~$ sudo chmod u+s /home/guest/simpleid2
```

```
pisimonova@pisimonova:~$ sudo ls -l simpleid2
ls: невозможно получить доступ к 'simpleid2': Нет такого файла или каталога
pisimonova@pisimonova:~$ sudo ls -l /home/guest/simpleid2
-rwsr-xr-x. 1 root guest 16920 фев 14 14:38 /home/guest/simpleid2
```

Рисунок 3.9: Смена владельца файла и прав доступа к файлу

Сравнение вывода программы и команды id, наша команда снова вывела только ограниченное количество информации(рис. 10)

```
pisimonova@pisimonova:~$ sudo /home/guest/simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
pisimonova@pisimonova:~$ sudo id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:u
nconfined_t:s0-s0:c0.c1023
pisimonova@pisimonova:~$ exit
```

Рисунок 3.10: Запуск файла

Создание и компиляция файла readfile.c (рис. 11)

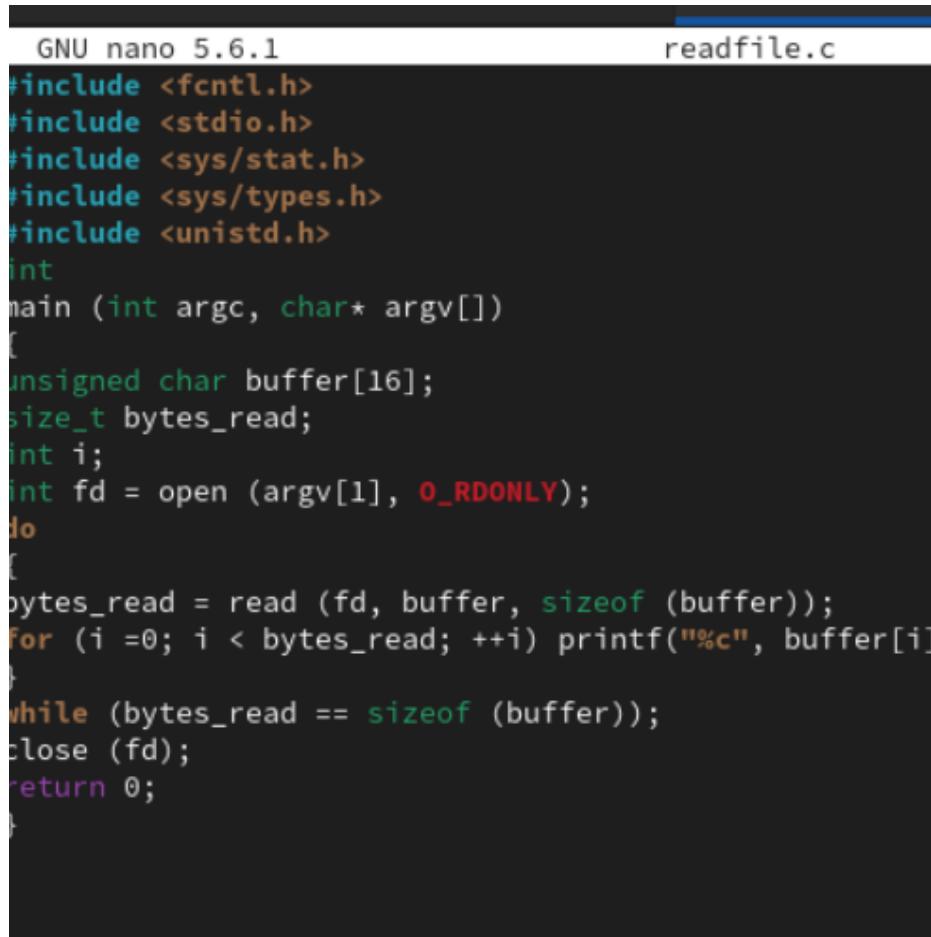
```
guest@pisimonova:~$ touch readfile.c
guest@pisimonova:~$ nano readfile.c
guest@pisimonova:~$ gcc readfile.c -o readfile
guest@pisimonova:~$ ls
dir1      simpleid      simpleid.c    Загрузки    Общедоступные
readfile  simpleid2     Видео        Изображения 'Рабочий стол'
readfile.c simpleid2.c  Документы   Музыка      Шаблоны
... . . .
```

Рисунок 3.11: Создание и компиляция файла

```
#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;
}
```

Листинг 3



```
GNU nano 5.6.1                                     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;
}
```

Рисунок 3.12: Содержимое файла

Снова от имени суперпользователи меняю владельца файла readfile. Далее меняю права доступа так, чтобы пользователь guest не смог прочесть содержимое файла (рис. 13)

```
pisimonova@pisimonova:~$ sudo chown root:guest /home/guest/readfile.c
pisimonova@pisimonova:~$ sudo chmod u+s /home/guest/readfile.c
pisimonova@pisimonova:~$ sudo chmod 700 /home/guest/readfile.c
pisimonova@pisimonova:~$ sudo chmod -r /home/guest/readfile.c
pisimonova@pisimonova:~$ sudo chmod u+s /home/guest/readfile.c
nisimonova@nisimonova:~$ su guest
```

Рисунок 3.13: Смена владельца файла и прав доступа к файлу

Проверка прочесть файл от имени пользователя guest.Прочесть файл не удается (рис. 14)

```
guest@pisimonova:~$ cat readfile.c  
cat: readfile.c: Отказано в доступе
```

Рисунок 3.14: Попытка прочесть содержимое файла

Попытка прочесть тот же файл с помощью программы `readfile`, в ответ получаем «отказано в доступе» (рис. 15)

Рисунок 3.15: Попытка прочесть содержимое файла программой

Попытка прочесть файл \etc\shadow с помощью программы, все еще получаем отказ в доступе (рис. 16)

```
guest@pisimonova:~$ ./readfile /etc/shadow
|Q|@|D#|A|rYt|j|b|\IV<px      GV=@|[Yt||rYtGp@|rYt||@xrYt|8||Yt||Yt|
|Yt||Yt||Yt||Yt|(|Yt|?|Yt|o|Yt||Yt||Yt||Yt||FYt||Yt||Yt||Yt||Yt|/|Yt|
D|Yt|Z|Yt||Yt||Yt||Yt|?|Yt|.|Yt|R|Yt|l|Yt|}||Yt||Yt|U|Yt|Yt||Yt||Yt||Yt|;|Y
t|F|Yt|Q|Yt|Y|Yt|k|Yt|}||Yt||Yt|HYt||Yt||Yt||Yt|L|Yt|]||Yt||Yt||Yt|h|Yt|
|Yt||PX      G3||||d@pxX      G      p@

|
||||uYt|?|Yt||uYt|j|b|\1|1| x86_64 ./readfile/etc/shadowSHELL=/bin/bashSESS
ION MANAGER=local/unix:@/tmp/.ICE-unix/2393.unix/unix:@/tmp/.ICE-unix/2393CO
```

Рисунок 3.16: Попытка прочесть содержимое файла программой

Пробуем прочесть эти же файлы от имени суперпользователя и чтение файлов проходит успешно (рис. 17)

```
pisimonova@pisimonova:~$ sudo /home/guest/readfile /etc/shadow
root:$y$j9T$szYenPDDQgj0aE34FNj70aUF$gv8u9dZtGzaXbu1lf9KN3eK75PsAKe2BMP7xjB
BGibD::0:99999:7:::
bin:*:20186:0:99999:7:::
daemon:*:20186:0:99999:7:::
adm:*:20186:0:99999:7:::
lp:*:20186:0:99999:7:::
sync:*:20186:0:99999:7:::
shutdown:*:20186:0:99999:7:::
```

Рисунок 3.17: Чтение файла от имени суперпользователя

Проверяем папку tmp на наличие атрибута Sticky, т.к. в выводе есть буква t, то атрибут установлен (рис. 18)

```
pisimonova@pisimonova:~$ sudo ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 фев 14 14:54 tmp
```

Рисунок 3.18: Проверка атрибутов директории tmp

От имени пользователя guest создаю файл с текстом, добавляю права на чтение и запись для других пользователей (рис. 19)

```
guest@pisimonova:~$ echo "test" > /tmp/file01.txt
guest@pisimonova:~$ ls
dir1      simpleid      simpleid.c   Загрузки      Общедоступные
readfile  simpleid2     Видео        Изображения  'Рабочий стол'
readfile.c simpleid2.c  Документы    Музыка       Шаблоны
guest@pisimonova:~$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 фев 14 14:58 /tmp/file01.txt
guest@pisimonova:~$ chmod o+rw /tmp/file01.txt
guest@pisimonova:~$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 фев 14 14:58 /tmp/file01.txt
```

Рисунок 3.19: Создание файла, изменение прав доступа

Вхожу в систему от имени пользователя guest2, от его имени могу прочитать файл file01.txt, но перезаписать информацию в нем не могу (рис. 20)

```
guest@pisimonova:~$ su guest2
Пароль:
guest2@pisimonova:/home/guest$ cd
guest2@pisimonova:~$ cat /tmp/file01.txt
test
guest2@pisimonova:~$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@pisimonova:~$ cat /tmp/file01.txt
test
```

Рисунок 3.20: Попытка чтения файла

Также невозможно добавить в файл file01.txt новую информацию от имени пользователя guest2 (рис. 21)

```
guest2@pisimonova:~$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@pisimonova:~$ cat /tmp/file01.txt
test
```

Рисунок 3.21: Попытка записи в файл

Далее пробуем удалить файл, снова получаем отказ (рис. 22)

```
guest2@pisimonova:~$ rm /tmp/file01.txt
rm: удалить защищенный от записи обычный файл '/tmp/file01.txt'? у
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Рисунок 3.22: Попытка удалить файл

От имени суперпользователя снимаем с директории атрибут Sticky (рис. 23)

```
guest2@pisimonova:~$ su -
Пароль:
root@pisimonova:~# chmod -t /tmp
root@pisimonova:~# exit
ВЫХОД
guest2@pisimonova:~$ ls -l / | grep tmp
```

Рисунок 3.23: Смена атрибутов файла

Проверяем, что атрибут действительно снят (рис. 24)

```
guest2@pisimonova:~$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 фев 14 15:04 tmp
```

Рисунок 3.24: Проверка атрибутов директории

Далее был выполнен повтор предыдущих действий. По результатам без Sticky-бита запись в файл и дозапись в файл осталась невозможной, зато удаление файла прошло успешно (рис. 25)

```
guest2@pisimonova:~$ cat /tmp/file01.txt
test
guest2@pisimonova:~$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@pisimonova:~$ cat /tmp/file01.txt
test
guest2@pisimonova:~$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@pisimonova:~$ cat /tmp/file01.txt
test
guest2@pisimonova:~$ rm /tmp/file01.txt
rm: удалить защищенный от записи обычный файл '/tmp/file01.txt'? y
guest2@pisimonova:~$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 фев 14 15:05 tmp
guest2@pisimonova:~$ ls -l
итого 0
guest2@pisimonova:~$ ls -l /home/guest
итого 72
drwx----- 2 guest guest 19 фев 14 13:59 dir1
-rw-r--r-- 1 guest guest 16864 фев 14 14:48 readfile
--ws----- 1 root guest 402 фев 14 14:47 readfile.c
-rw-r--r-- 1 guest guest 16816 фев 14 14:35 simpleid
-rwsr--r-- 1 root guest 16920 фев 14 14:38 simpleid2
-rw-r--r-- 1 guest guest 303 фев 14 14:38 simpleid2.c
-rw-r--r-- 1 guest guest 175 фев 14 14:35 simpleid.c
drwxr-xr-x. 2 guest guest 6 фев 13 21:34 Видео
drwxr-xr-x. 2 guest guest 6 фев 13 21:34 Документы
drwxr-xr-x. 2 guest guest 6 фев 13 21:34 Загрузки
drwxr-xr-x. 2 guest guest 6 фев 13 21:34 Изображения
drwxr-xr-x. 2 guest guest 6 фев 13 21:34 Музыка
drwxr-xr-x. 2 guest guest 6 фев 13 21:34 Общедоступные
drwxr-xr-x. 2 guest guest 6 фев 13 21:34 'Рабочий стол'
drwxr-xr-x. 2 guest guest 6 фев 13 21:34 Шаблоны
```

Рисунок 3.25: Повтор предыдущих действий

Возвращение директории tmp атрибута t от имени суперпользователя (рис. 26)

```
root@pisimonova:~# chmod +t /tmp
root@pisimonova:~# exut
bash: exut: команда не найдена...
exit
root@pisimonova:~# exit
ВЫХОД
```

Рисунок 3.26: Изменение атрибутов

4 Выводы

Изучила механизм изменения идентификаторов, применила SetUID- и Sticky-биты. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

5 Список литературы

- [0] Методические материалы курса
- [1] Права доступа: <https://codechick.io/tutorials/unix-linux/unix-linux-permissions>