

Отчет по лабораторной работе №5

Основы информационной безопасности

Дворкина Ева, НКАбд-01-22

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	18
	Список литературы	19

Список иллюстраций

3.1	Подготовка к лабораторной работе	8
3.2	Вход от имени пользователя guest	9
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	11
3.10	Запуск файла	12
3.11	Создание и компиляция файла	12
3.12	Содержимое файла	13
3.13	Смена владельца файла и прав доступа к файлу	13
3.14	Попытка прочесть содержимое файла	14
3.15	Попытка прочесть содержимое файла программой	14
3.16	Попытка прочесть содержимое файла программой	14
3.17	Чтение файла от имени суперпользователя	14
3.18	Проверка атрибутов директории tmp	15
3.19	Создание файла, изменение прав доступа	15
3.20	Попытка чтения файла	15
3.21	Попытка записи в файл	16
3.22	Попытка удалить файл	16
3.23	Смена атрибутов файла	16
3.24	Проверка атрибутов директории	16
3.25	Повтор предыдущих действий	17
3.26	Изменение атрибутов	17

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. [1]

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 относится 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 `gwx`? 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`. Остальные атрибуты имеют следующие числовое соответствие:

1 — установлен `sticky bit`

2 — установлен `sgid`

4 — установлен `suid`

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 [2].

3 Выполнение лабораторной работы

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

```
[evdvorkina@evdvorkina ~]$ whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz /usr/share/info/gcc.info.gz
[evdvorkina@evdvorkina ~]$ whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
[evdvorkina@evdvorkina ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --enable-bootstrap --enable-host-pie --enable-host-bind-now --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --enable-plugin --enable-initfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x86-64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lto --enable-link-serialization=1
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.4.1 20230605 (Red Hat 11.4.1-2) (GCC)
[evdvorkina@evdvorkina ~]$ setenforce 0
setenforce: security_setenforce() failed: Permission denied
[evdvorkina@evdvorkina ~]$ sudo setenforce 0
[sudo] пароль для evdvorkina:
[evdvorkina@evdvorkina ~]$ getenforce
Permissive
[evdvorkina@evdvorkina ~]$
```

Рис. 3.1: Подготовка к лабораторной работе

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

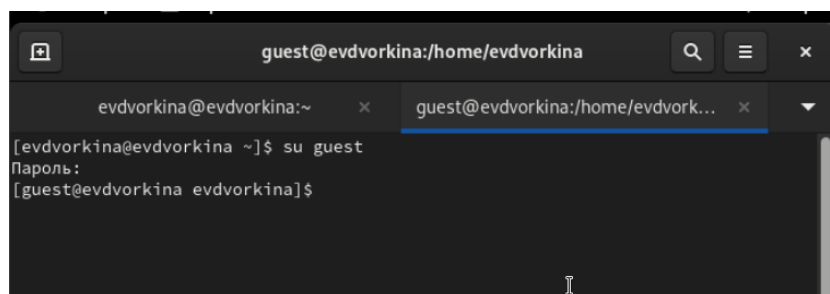


Рис. 3.2: Вход от имени пользователя guest

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

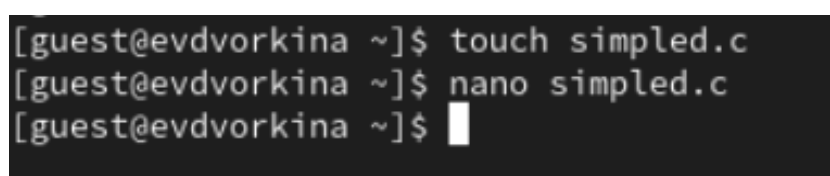


Рис. 3.3: Создание файла

C++ Листинг 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; }`

Содержимое файла выглядит следующи образом (рис. 4)

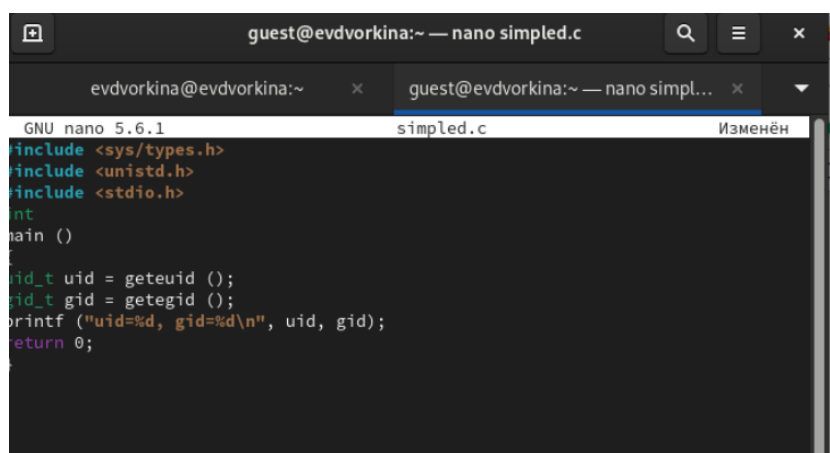


Рис. 3.4: Содержимое файла

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

```
[guest@evdvorkina ~]$ gcc simpled.c -o simpled
[guest@evdvorkina ~]$ ls
dir1      test      Видео      Изображения  'Рабочий стол'
simpled    test10    Документы  Музыка       Шаблоны
simpled.c  test2     Загрузки   Общедоступные
[guest@evdvorkina ~]$ ./simpled
```

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

Рис. 3.6: Сравнение команд

Создание, запись в файл и компиляция файла `simpled2.c`. Запуск программы (рис. 7)

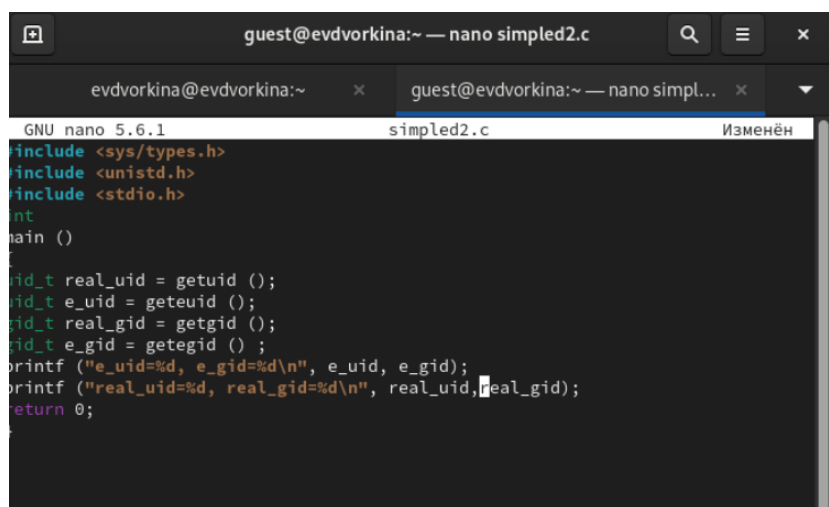
```
[guest@evdvorkina ~]$ touch simpled2.c
[guest@evdvorkina ~]$ nano simpled2.c
[guest@evdvorkina ~]$ gcc simpled2.c -o simpled2
[guest@evdvorkina ~]$ ./simpled2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@evdvorkina ~]$
```

Рис. 3.7: Создание и компиляция файла

C++ Листинг 2 `#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; }
```

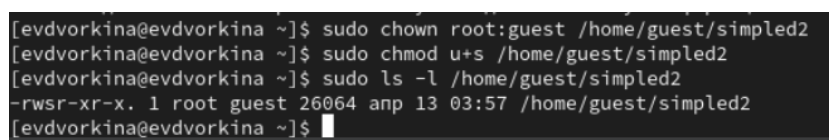
(рис. 8)



```
GNU nano 5.6.1 simplified2.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. 3.8: Содержимое файла

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



```
[evdvorkina@evdvorkina ~]$ sudo chown root:guest /home/guest/simplified2
[evdvorkina@evdvorkina ~]$ sudo chmod u+s /home/guest/simplified2
[evdvorkina@evdvorkina ~]$ sudo ls -l /home/guest/simplified2
-rwsr-xr-x. 1 root guest 26064 anp 13 03:57 /home/guest/simplified2
[evdvorkina@evdvorkina ~]$
```

Рис. 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)

```
[evdvorkina@evdvorkina ~]$ sudo /home/guest/simplified2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[evdvorkina@evdvorkina ~]$ id
uid=1000(evdvorkina) gid=1000(evdvorkina) группы=1000(evdvorkina),10(wheel) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[evdvorkina@evdvorkina ~]$ sudo id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[evdvorkina@evdvorkina ~]$
```

Рис. 3.10: Запуск файла

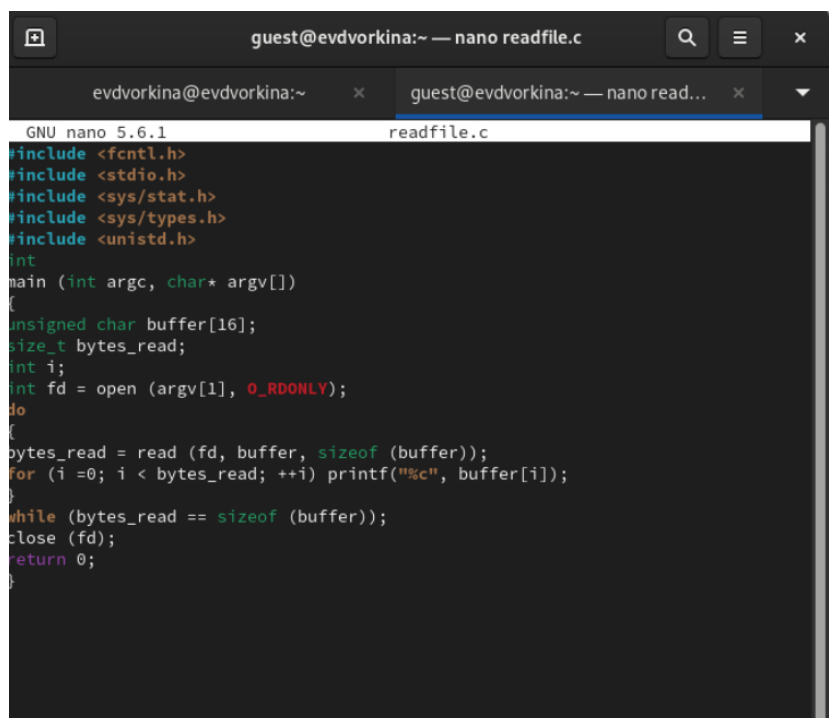
Создание и компиляция файла readfile.c (рис. 11)

```
[guest@evdvorkina ~]$ touch readfile.c
[guest@evdvorkina ~]$ nano readfile.c
[guest@evdvorkina ~]$ nano readfile.c
[guest@evdvorkina ~]$ gcc readfile.c -o readfile
[guest@evdvorkina ~]$ ls
dir1      simplified      simplified.c    test2          Загрузки      Общедоступные
readfile  readfile       test           Видео          Изображения  'Рабочий стол'
readfile.c  simplified2.c  test10        Документы     Музыка        Шаблоны
```

Рис. 3.11: Создание и компиляция файла

```
C++ Листинг 3 #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; }
```

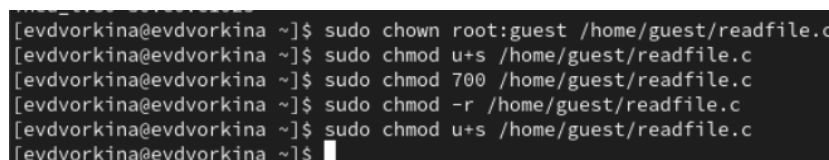
(рис. 12)



```
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;
}
```

Рис. 3.12: Содержимое файла

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



```
[evdvorkina@evdvorkina ~]$ sudo chown root:guest /home/guest/readfile.c
[evdvorkina@evdvorkina ~]$ sudo chmod u+s /home/guest/readfile.c
[evdvorkina@evdvorkina ~]$ sudo chmod 700 /home/guest/readfile.c
[evdvorkina@evdvorkina ~]$ sudo chmod -r /home/guest/readfile.c
[evdvorkina@evdvorkina ~]$
```

Рис. 3.13: Смена владельца файла и прав доступа к файлу

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

```
[guest@evdvorkina ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@evdvorkina ~]$
```

Рис. 3.14: Попытка прочесть содержимое файла

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

[illegible]

Рис. 3.15: Попытка прочесть содержимое файла программой

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

```
[guest@evdworkina ~]$ ./readfile /etc/shadow  
*****t-0MTK6y>p0T0ZCMT0t0z[-*****  
S-  
t-t-t-t-t-t-x-x-x-x-x-x-x-u-  
|-|-|_|_-||_||_||_||_||_}|_} }_  
pe
```

Рис. 3.16: Попытка прочесть содержимое файла программой

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

```
[evdvorkina@evdvorkina ~]$ sudo /home/guest/readfile /etc/shadow
[sudo] пароль для evdvorkina:
root:$6$3reywnb0G.0EFHL7$1td/ZD0qRRQEdBaZehnNr0Kq7lhY9HS4Ip0CdU6M/hmKBvFhSqs02
gd3/YkGPNmw5AD2t0ThLFZYuXi4eD/rU0::0:09999:7:::
bin::19469:0:99999:7:::
daemon::19469:0:99999:7:::
adm::19469:0:99999:7:::
lp::19469:0:99999:7:::
nlp::19469:0:99999:7:::
```

Рис. 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)

```
[evdvorkina@evdvorkina ~]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 апр 13 04:21 tmp
```

Рис. 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@evdvorkina ~]$ echo "test" > /tmp/file01.txt
[guest@evdvorkina ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 апр 13 04:26 /tmp/file01.txt
[guest@evdvorkina ~]$ chmod o+rw /tmp/file01.txt
[guest@evdvorkina ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 апр 13 04:26 /tmp/file01.txt
[guest@evdvorkina ~]$
```

Рис. 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)

```
[evdvorkina@evdvorkina ~]$ su guest2
Пароль:
su: Сбой при проверке подлинности
[evdvorkina@evdvorkina ~]$ su guest2
Пароль:
[guest2@evdvorkina evdvorkina]$ cat /tmp/file01.txt
test
[guest2@evdvorkina evdvorkina]$ echo 'test2' >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@evdvorkina evdvorkina]$ cat /tmp/file01.txt
test
[guest2@evdvorkina evdvorkina]$
```

Рис. 3.20: Попытка чтения файла

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

```
[guest2@evdvorkina evdvorkina]$ echo 'test3' > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@evdvorkina evdvorkina]$ cat /tmp/file01.txt
test
[guest2@evdvorkina evdvorkina]$
```

Рис. 3.21: Попытка записи в файл

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

```
[guest2@evdvorkina evdvorkina]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Рис. 3.22: Попытка удалить файл

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

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

Рис. 3.23: Смена атрибутов файла

Проверяем, что атрибут действительно снят (рис. 24)

```
выход
[guest2@evdvorkina evdvorkina]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 апр 13 04:32 tmp
```

Рис. 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@evdvorkina evdvorkina]$ cat /tmp/file01.txt
test
[guest2@evdvorkina evdvorkina]$ echo 'test2' >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@evdvorkina evdvorkina]$ cat /tmp/file01.txt
test
[guest2@evdvorkina evdvorkina]$ echo 'test3' > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@evdvorkina evdvorkina]$ cat /tmp/file01.txt
test
[guest2@evdvorkina evdvorkina]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
[guest2@evdvorkina evdvorkina]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 апр 13 04:35 tmp
[guest2@evdvorkina evdvorkina]$ ls -l
ls: невозможно открыть каталог '.': Отказано в доступе
[guest2@evdvorkina evdvorkina]$ ls -l /home/guest
итого 108
drwx-----. 3 guest guest   38 мар 3 01:55 dir1
-rwxr-xr-x. 1 guest guest 26008 апр 13 04:19 readfile
-rw-r--r--. 1 guest guest   402 апр 13 04:19 readfile1.c
--ws-----. 1 root guest   402 апр 13 04:08 readfile.c
-rwxr-xr-x. 1 guest guest 25960 апр 13 03:53 simplified
-rwsr-xr-x. 1 root guest 26064 апр 13 03:57 simplified2
-rw-r--r--. 1 guest guest   302 апр 13 03:56 simplified2.c
-rw-r--r--. 1 guest guest   175 апр 13 03:53 simplified.c
-rw-r--r--. 1 guest guest    5 фев 18 20:39 test
-----. 1 guest guest    5 фев 18 20:27 test10
-----. 1 guest guest    0 фев 18 21:05 test2
drwxr-xr-x. 2 guest guest    6 фев 18 18:49 Видео
drwxr-xr-x. 2 guest guest    6 фев 18 18:49 Документы
drwxr-xr-x. 2 guest guest    6 фев 18 18:49 Загрузки
drwxr-xr-x. 2 guest guest    6 фев 18 18:49 Изображения
drwxr-xr-x. 2 guest guest    6 фев 18 18:49 Музыка
drwxr-xr-x. 2 guest guest    6 фев 18 18:49 Общедоступные
drwxr-xr-x. 2 guest guest    6 фев 18 18:49 'Рабочий стол'
drwxr-xr-x. 2 guest guest    6 фев 18 18:49 Шаблоны
[guest2@evdvorkina evdvorkina]$

```

Рис. 3.25: Повтор предыдущих действий

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

```

[guest2@evdvorkina evdvorkina]$ su -
Пароль:
[root@evdvorkina ~]# chmod +t /tmp
[root@evdvorkina ~]# exit
ВЫХОД
[guest2@evdvorkina evdvorkina]$

```

Рис. 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.

Список литературы

1. Дополнительные атрибуты файлов: sticky bit, suid, sgid [Электронный ресурс]. 2018. URL: <https://tokmakov.msk.ru/blog/item/141>.
2. Инструментарий программиста в Linux: Компилятор GCC [Электронный ресурс]. URL: <http://parallel.imm.uran.ru/freesoft/make/instrum.html>.