

Отчет по лабораторной работе №5

Основы информационной безопасности

Швецов Михаил, НКАбд-01-23

Содержание

1	Цель работы.....	1
2	Теоретическое введение.....	1
3	Выполнение лабораторной работы.....	2
4	Выводы	9
	Список литературы	10

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 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. 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 запретом с помощью setenforce 0 (рис. 1).

```
mrshvecov@mrshvecov:~$ whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
mrshvecov@mrshvecov:~$ 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 11.5.0 20240719 (Red Hat 11.5.0-5) (GCC)
mrshvecov@mrshvecov:~$ set
```

Подготовка к лабораторной работе

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

```
mrshvecov@mrshvecov:~ -- su guest
mrshvecov@mrshvecov:~ x mrshvecov@mrshvecov:~ -- su gu... x
[mrshvecov@mrshvecov ~]$ su guest
Пароль:
```

Вход от имени пользователя guest

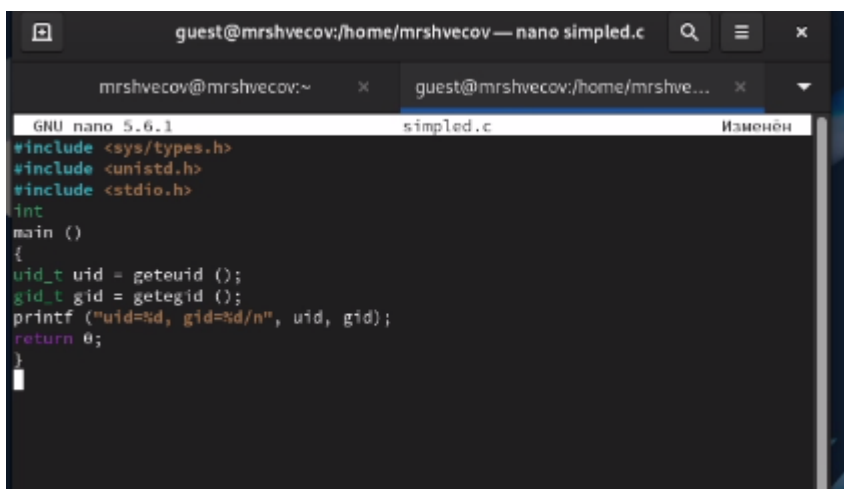
Создание файла `simplified.c` и запись в файл кода (рис. 3)

```
[guest@mrshvecov mrshvecov]$ touch simplified.c
touch: невозможно выполнить touch для 'simplified.c': Отказано в доступе
[guest@mrshvecov mrshvecov]$
```

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; }
```

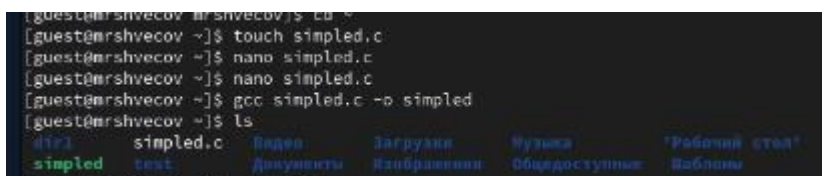
Содержимое файла выглядит следующим образом (рис. 4)



```
GNU nano 5.6.1 simplified.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
    uid_t uid = getuid ();
    gid_t gid = getgid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Содержимое файла

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



```
[guest@mrshvecov ~]$ touch simplified.c
[guest@mrshvecov ~]$ nano simplified.c
[guest@mrshvecov ~]$ gcc simplified.c -o simplified
[guest@mrshvecov ~]$ ls
dir/  simplified.c  Бадес  Загрузки  Музыка  'Рабочий стол'
simplified  test  Документы  Избранное  Общедоступные  Шаблоны
```

Компиляция файла

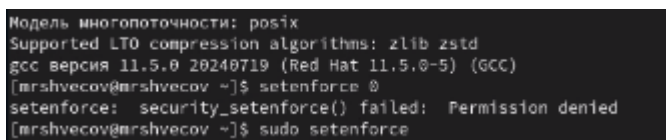
Запускаю исполняемый файл. В выводе файла выписаны номера пользователя и групп, от вывода при вводе if, они отличаются только тем, что информации меньше (рис. 6)



```
[guest@mrshvecov ~]$ ./simplified
uid=1001, gid=1001/n[guest@mrshvecov ~]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest),10(wheel) контекст=unconfined
_u:unconfined_r:unconfined_t:s0-s0:c0,c1023
[guest@mrshvecov ~]$ touch simplified2.c
[guest@mrshvecov ~]$ nano simplified2
```

Сравнение команд

Создание, запись в файл и компиляция файла simplified2.c. Запуск программы (рис. 7)

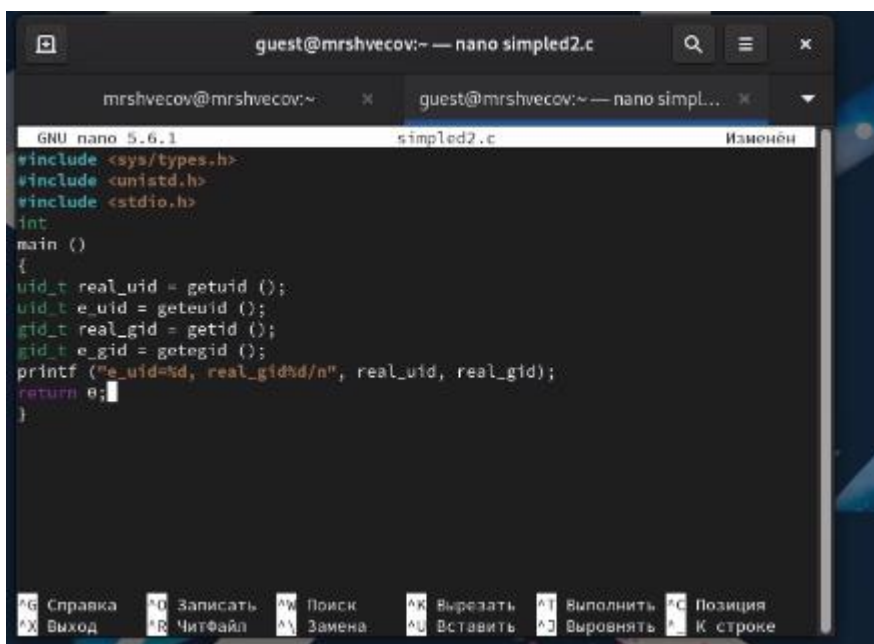


```
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.5.0 20240719 (Red Hat 11.5.0-5) (GCC)
[mrshvecov@mrshvecov ~]$ setenforce 0
setenforce: security_setenforce() failed: Permission denied
[mrshvecov@mrshvecov ~]$ sudo setenforce
```

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; }

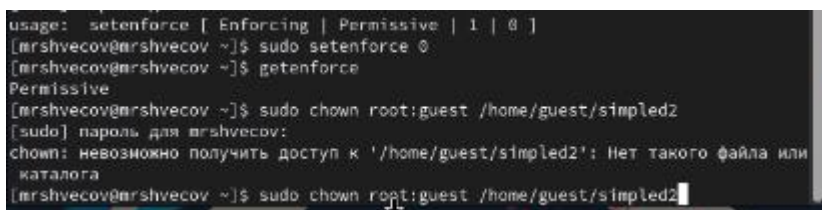
(рис. 8)



```
GNU nano 5.6.1 simplified2.c Изменён
#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 = getid ();
gid_t e_gid = getegid ();
printf ("e_uid=%d, real_gid=%d\n", real_uid, real_gid);
return 0;
}
```

Содержимое файла

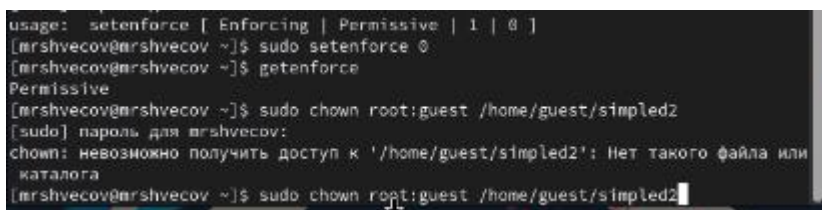
С помощью `chown` изменяю владельца файла на суперпользователя, с помощью `chmod` изменяю права доступа (рис. 9)



```
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
[mrshvecov@mrshvecov ~]$ sudo setenforce 0
[mrshvecov@mrshvecov ~]$ getenforce
Permissive
[mrshvecov@mrshvecov ~]$ sudo chown root:guest /home/guest/simplified2
[sudo] пароль для mrshvecov:
chown: невозможно получить доступ к '/home/guest/simplified2': Нет такого файла или каталога
[mrshvecov@mrshvecov ~]$ sudo chown root:guest /home/guest/simplified2
```

Смена владельца файла и прав доступа к файлу

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



```
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
[mrshvecov@mrshvecov ~]$ sudo setenforce 0
[mrshvecov@mrshvecov ~]$ getenforce
Permissive
[mrshvecov@mrshvecov ~]$ sudo chown root:guest /home/guest/simplified2
[sudo] пароль для mrshvecov:
chown: невозможно получить доступ к '/home/guest/simplified2': Нет такого файла или каталога
[mrshvecov@mrshvecov ~]$ sudo chown root:guest /home/guest/simplified2
```

Запуск файла

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

```
guest@mrshvecov:~  
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest),10(wheel) контекст=unconfined  
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@mrshvecov ~]$ touch simplified2.c  
[guest@mrshvecov ~]$ nano simplified2.c  
[guest@mrshvecov ~]$ gcc simplified2.c -o simplified  
simplified2.c: В функции «main»:  
simplified2.c:9:18: предупреждение: неявная декларация функции «getid»; имелось в в  
иду «getid»? [-Wimplicit-function-declaration]  
  9 | gid_t real_gid = getid ();  
    |                   ^~~~~  
    |                   getid  
/usr/bin/ld: /tmp/ccVa7kUo.o: в функции «main»:  
simplified2.c:(.text+0x1e): неопределённая ссылка на «getid»  
collect2: ошибка: выполнение ld завершилось с кодом возврата 1  
[guest@mrshvecov ~]$ touch nanofile.c  
[guest@mrshvecov ~]$ nano readfile.c  
[guest@mrshvecov ~]$ gcc readfile.c -o readfile  
[guest@mrshvecov ~]$ ls  
dir1  readfile.c  test  Загрузки  Общедоступные  
nanofile.c  simplified2.c  Видео  Изображения  'Рабочий стол'  
readfile  simplified.c  Документы  Музыка  Шаблоны  
[guest@mrshvecov ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@mrshvecov ~]$
```

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;  
}
```

Содержимое файла

```
[guest@mrshvecov mrshvecov]$ touch simplified.c
touch: невозможно выполнить touch для 'simplified.c': Отказано в доступе
[guest@mrshvecov mrshvecov]$
```

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

```
[guest@mrshvecov mrshvecov]$ touch simplified.c
touch: невозможно выполнить touch для 'simplified.c': Отказано в доступе
[guest@mrshvecov mrshvecov]$
```

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

[illegible]

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

[illegible]

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

```
root:$6$3reywnb0G.6fHL7$1td/ZD0qRQEdbaZehnNr0Kq7LhY9HS4Ip0CdU6H/hmKbVhfSqsO2
gd3/YkGPNmw5AD2t0THL7ZYUx14eD/rU0::0:99999:7::
bin::19469:0:99999:7::
daemon::19469:0:99999:7::
adm::19469:0:99999:7::
lp:19469:0:99999:7::
sync::19469:0:99999:7::
```

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


```
drwxrwxrwt. 18 root root 4096 апр 13 04:21 tmp
```

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 ~]$
```

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]$
```

Попытка чтения файла

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

```
ns --show-tilde --show-dot sq
Ошибка сегментирования (стек памяти сброшен на диск)
[guest@mrshvecov ~]$ echo "test" > /tmp/file01.txt
[guest@mrshvecov ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 сен 15 06:23 /tmp/file01.txt
```

Попытка записи в файл

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

```
rm: удалить защищенный от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Попытка удалить файл

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


```
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@mrshvecov mrshvecov]$ su -
Пароль:
[root@mrshvecov ~]# chmod -t /tmp
[root@mrshvecov ~]# exit
выход
[guest2@mrshvecov mrshvecov]$ ls -la /tmp
drwxrwxrwx. 1 root root 4096 сен 15 06:23 /tmp
```

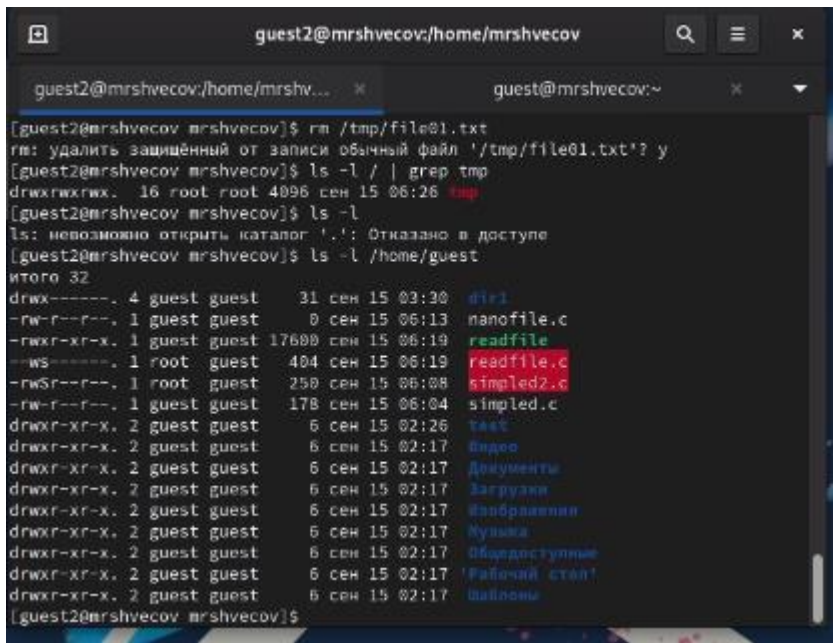
Смена атрибутов файла

Проверяем, что атрибут действительно снят (рис. 24)

```
ns --show-citde --show-dot $@
Ошибка сегментирования (стек памяти сброшен на диск)
[guest2@mrshvecov ~]$ echo "test" > /tmp/file01.txt
[guest2@mrshvecov ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 сен 15 06:23 /tmp/file01.txt
```

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@mrshvecov:/home/mrshvecov
[guest2@mrshvecov mrshvecov]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
[guest2@mrshvecov mrshvecov]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 сен 15 06:26 tmp
[guest2@mrshvecov mrshvecov]$ ls -l
ls: невозможно открыть каталог '.': Отказано в доступе
[guest2@mrshvecov mrshvecov]$ ls -l /home/guest
итого 32
drwx-----. 4 guest guest    31 сен 15 03:30 dir1
-rw-r--r--. 1 guest guest     0 сен 15 06:13 nanofile.c
-rwxr-xr-x. 1 guest guest 17680 сен 15 06:19 readfile
--ws-----. 1 root  guest   404 сен 15 06:19 readfile.c
-rwsr--r--. 1 root  guest   250 сен 15 06:08 simpled2.c
-rw-r--r--. 1 guest guest   178 сен 15 06:04 simpled.c
drwxr-xr-x. 2 guest guest     6 сен 15 02:26 test
drwxr-xr-x. 2 guest guest     6 сен 15 02:17 видео
drwxr-xr-x. 2 guest guest     6 сен 15 02:17 Документы
drwxr-xr-x. 2 guest guest     6 сен 15 02:17 Загрузки
drwxr-xr-x. 2 guest guest     6 сен 15 02:17 Изображения
drwxr-xr-x. 2 guest guest     6 сен 15 02:17 Музыка
drwxr-xr-x. 2 guest guest     6 сен 15 02:17 Общедоступные
drwxr-xr-x. 2 guest guest     6 сен 15 02:17 'Рабочий стол'
drwxr-xr-x. 2 guest guest     6 сен 15 02:17 Шаблоны
[guest2@mrshvecov mrshvecov]$
```

Повтор предыдущих действий

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

Изменение атрибутов

4 Выводы

Изучила механизм изменения идентификаторов, применила SetUID- и Sticky-биты. Получила практические навыки работы в кон- соли 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>.