

Отчёт по лабораторной работе 5

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

Румянцева Александра Сергеевна

Содержание

Цель работы	5
Задание	6
Теория	7
Текстовый (символический) метод CHMOD	7
Расширенные права доступа к файлам.	8
Бит sticky для дирктории.	8
Биты setgid и setuid для обычных файлов	8
Выполнение лабораторной работы	9
Подготовка к выполнению	9
Выполнение основной части лабораторной работы	9
Исследование Sticky-бита	15
Библиография	19
Выводы	20

Список иллюстраций

0.1	рис.1. Установка компилятора gcc.	9
0.2	рис.2. Программа simpleid.c	10
0.3	рис.3. Компиляция программы simpleid, её выполнение и сравнение с командой id.	10
0.4	рис.4. Программа simpleid2.c	11
0.5	рис.5. Компиляция программы simpleid2, её выполнение.	11
0.6	рис.6. Изменение владельца программы и установка SetUID-бита, проверка установки и изменения, запуск программы и команды id.	12
0.7	рис.7. Установка SetGID-бита, проверка установки, запуск программы и команды id.	12
0.8	рис.8. Программа readfile.c	13
0.9	рис.9. Компиляция readfile и другие действия в соответствии с 14-17 пунктами.	14
0.10	рис.10. Выполнение программы readfile с файлом readfile.c.	14
0.11	рис.11. Выполнение программы readfile с файлом /etc/shadow.	15
0.12	рис.12. Выполнение пунктов 1-4 исследования Sticky-бита	16
0.13	рис.13. Выполнение пунктов 5-9 исследования Sticky-бита	17
0.14	рис.14. Выполнение пунктов 10-13 исследования Sticky-бита	18
0.15	рис.15. Возвращение атрибута t на директорию.	18

Список таблиц

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.

Задание

Лабораторная работа подразумевает изучение влияния дополнительных атрибутов на файлы пользователя и изучение механизмов изменения идентификаторов.

Теория

Текстовый (символический) метод CHMOD

Команда `chmod` позволяет вам изменять права доступа к файлу, используя либо символьный, либо числовой режим, либо ссылочный файл. Мы объясним режимы более подробно позже в этой статье. Команда может принимать один или несколько файлов и / или каталогов, разделенных пробелом, в качестве аргументов. Только `root`, владелец файла или пользователь с привилегиями `sudo` могут изменять права доступа к файлу. Будьте особенно осторожны при использовании `chmod`, особенно при рекурсивном изменении разрешений.

С численным методом мы знакомы с прошлых лабораторных, изучим текстовый метод.

Синтаксис `chmod` команды при использовании символьного режима имеет следующий формат: `chmod [OPTIONS] [ugoa...][-+=]perms...[,...] FILE...`

Первый набор флагов (`[ugoa...]`), флаги пользователей, определяет, для каких классов пользователей изменяются права доступа к файлу.

u - Владелец файла. g - Пользователи, которые являются членами группы. o - Все остальные пользователи. a - Все пользователи, идентичные ugo.

Второй набор флагов (`[-+=]`), флагов операций, определяет, следует ли удалять, добавлять или устанавливать разрешения:

- Удаляет указанные разрешения.
- Добавляет указанные разрешения. = Изменяет текущие разрешения на указанные разрешения. Если после =символа не указано никаких разрешений ,

все разрешения из указанного пользовательского класса будут удалены.

Разрешения (perms...) можно явно задать с помощью либо ноль , либо один или несколько из следующих букв: r, w, x, X, s, и t. Используйте одну букву из набора u, g и o при копировании разрешений от одного к другому классу пользователей.

При настройке разрешений для нескольких пользовательских классов ([,...]) используйте запятые (без пробелов) для разделения символьных режимов.

Расширенные права доступа к файлам.

Бит sticky для директории.

Можно установить бит sticky для директории с целью предотвращения удаления файлов пользователями, которые не являются их непосредственными владельцами. Бит sticky отображается в той же позиции, что и символ права исполнения x для пользователей, не являющихся владельцами директории и не состоящих в группе, владеющей директорией. Сам бит sticky отображается с помощью символа t (в том случае, если подразумевается наличие символа x) или символа T (в том случае, если символ x не должен выводиться ввиду отсутствия соответствующих прав доступа к директории).

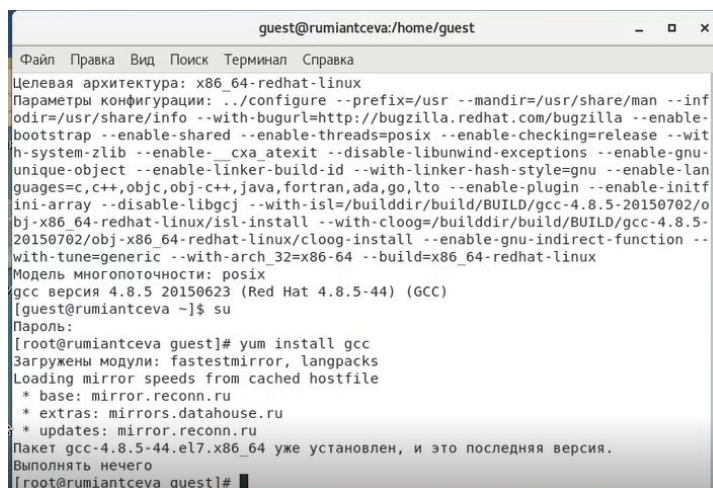
Биты setgid и setuid для обычных файлов

Эти биты позволяют запускать исполняемый файл с правами пользователя, владеющего файлом, а не с правами пользователя, инициировавшего запуск файла. Это означает, что в том случае, если какой-либо пользователь запустит программу, принадлежащую пользователю root, причем для исполняемого файла программы будет установлен бит setuid, то программа будет запущена от лица пользователя root. Такое поведение системы может оказаться опасным, но иногда оказывается и полезным для обхода ограничений безопасности.

Выполнение лабораторной работы

Подготовка к выполнению

1. Проверила установлен ли компилятор gcc (рис. 1). В моём случае он уже установлен.

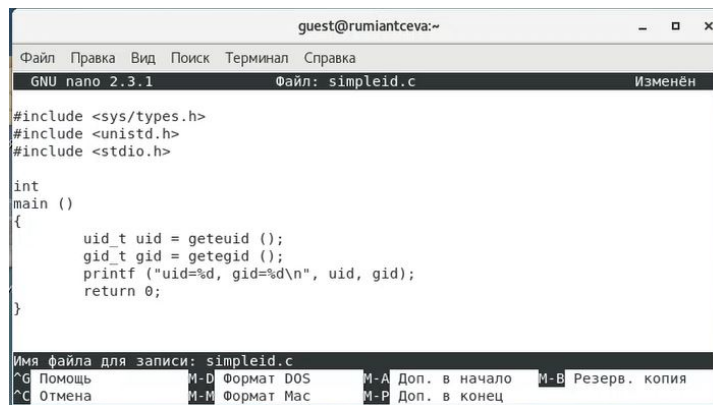


```
guest@rumiantceva:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --inf
odir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-
bootstrap --enable-shared --enable-threads=posix --enable-checking=release --wit
h-system-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-
unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-lan
guages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initf
ini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/o
bj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-
20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --
with-tune=generic --with-arch=32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
[guest@rumiantceva ~]$ su
Пароль:
[root@rumiantceva guest]# yum install gcc
Загружены модули: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.reconn.ru
 * extras: mirrors.datahouse.ru
 * updates: mirror.reconn.ru
Пакет gcc-4.8.5-44.el7.x86_64 уже установлен, и это последняя версия.
Выполнять нечего
[root@rumiantceva guest]#
```

Рис. 0.1: рис.1. Установка компилятора gcc.

Выполнение основной части лабораторной работы

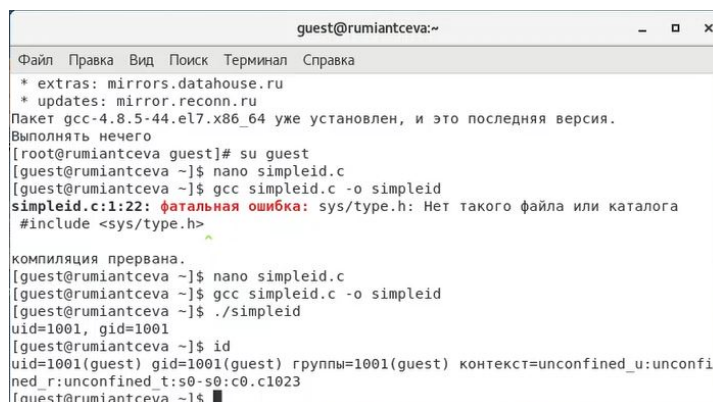
1. Вошла в систему от имени пользователя guest.
2. Создала программу simpleid.c (рис. 2).



```
guest@rumiantceva:~  
Файл Правка Вид Поиск Терминал Справка  
GNU nano 2.3.1 Файл: simpleid.c Изменён  
#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;  
}
```

Рис. 0.2: рис.2. Программа simpleid.c

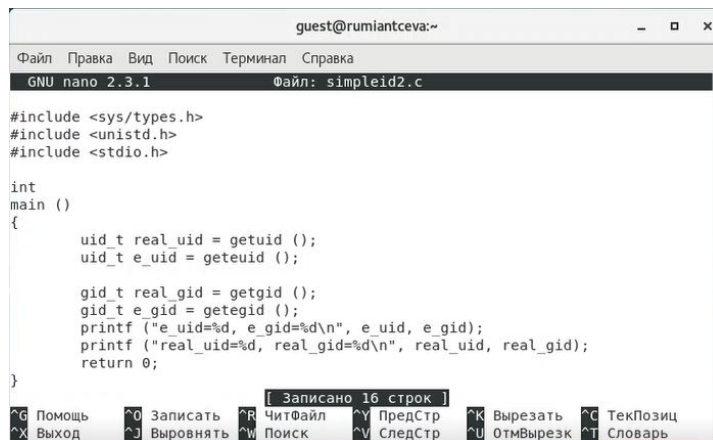
3. Скомпилировала программу и убедилась, что файл программы создан командой `gcc simpleid.c -o simpleid` (рис. 3).
4. Выполнила программу `simpleid` командой `./simpleid` (рис. 3).
5. Выполнила системную программу `id` и сравнила полученный результат с данными предыдущего пункта задания (рис. 3). Видим, что пользователи и группы совпадают. При этом команда `id` вывела действительные идентификаторы, а программа вывела эффективные, но при этом они совпадают и выводят 1001, то есть пользователя `guest`.



```
guest@rumiantceva:~  
Файл Правка Вид Поиск Терминал Справка  
* extras: mirrors.datahouse.ru  
* updates: mirror.reconn.ru  
Пакет gcc-4.8.5-44.el7.x86_64 уже установлен, и это последняя версия.  
Выполнять нечего  
[root@rumiantceva guest]# su guest  
[guest@rumiantceva ~]$ nano simpleid.c  
[guest@rumiantceva ~]$ gcc simpleid.c -o simpleid  
simpleid.c:1:22: фатальная ошибка: sys/types.h: Нет такого файла или каталога  
#include <sys/type.h>  
компиляция прервана.  
[guest@rumiantceva ~]$ nano simpleid.c  
[guest@rumiantceva ~]$ gcc simpleid.c -o simpleid  
[guest@rumiantceva ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@rumiantceva ~]$ id  
uid=1001(guest) gid=1001(guest) rpyнны=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@rumiantceva ~]$
```

Рис. 0.3: рис.3. Компиляция программы `simpleid`, её выполнение и сравнение с командой `id`.

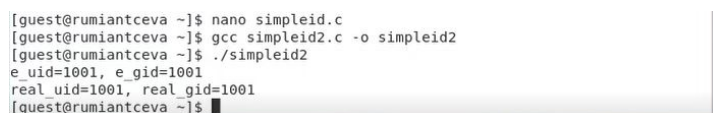
6. Усложнила программу, добавив вывод действительных идентификаторов, получившуюся программу назвала simpleid2.c (рис. 4).



```
guest@rumiantceva:~  
Файл Правка Вид Поиск Терминал Справка  
GNU nano 2.3.1 Файл: 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;  
}
```

Рис. 0.4: рис.4. Программа simpleid2.c

7. Скомпилировала и запустила simpleid2.c командами gcc simpleid2.c -o simpleid2 и ./simpleid2 (рис. 5). Видим, что программа выводит эффективные и действительные идентификаторы пользователя и группы для файла. Видим, что везде это 1001, то есть пользователь guest.



```
[guest@rumiantceva ~]$ nano simpleid.c  
[guest@rumiantceva ~]$ gcc simpleid2.c -o simpleid2  
[guest@rumiantceva ~]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@rumiantceva ~]$
```

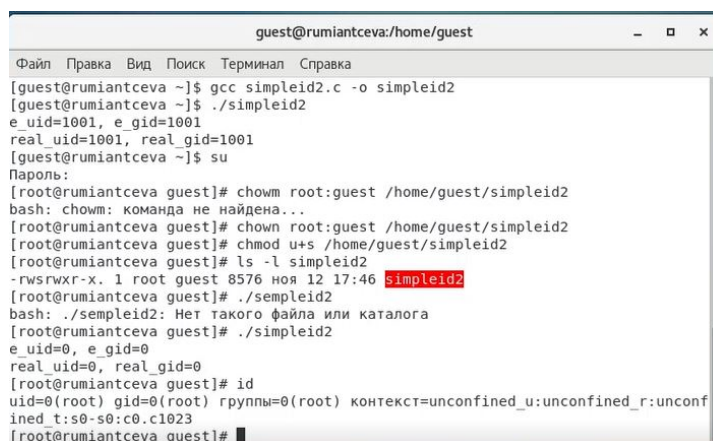
Рис. 0.5: рис.5. Компиляция программы simpleid2, её выполнение.

8-9. От имени суперпользователя выполнила команды: chown root:guest /home/guest/simpleid2 и chmod u+s /home/guest/simpleid2. Временно поменяв свои права с помощью su (рис. 6).

С помощью этих команд файлу simpleid2 изменила владельца и группу на root и guest соответственно (chown), а также установила на файл SetUID-бит (chmod).

10. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой ls -l simpleid2 (рис. 6).

11. Запустила simpleid2 и id командами ./simpleid2 и id (рис. 6). Сравнила результаты: действительные идентификаторы совпадают с выводом команды id - везде 0, то есть рут-пользователь. Так же важно заметить, что эффективные идентификаторы совпадают с действительными.



```
guest@rumiantceva:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@rumiantceva ~]$ gcc simpleid2.c -o simpleid2
[guest@rumiantceva ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@rumiantceva ~]$ su
Пароль:
[root@rumiantceva guest]# chown root:guest /home/guest/simpleid2
bash: chown: команда не найдена...
[root@rumiantceva guest]# chown root:guest /home/guest/simpleid2
[root@rumiantceva guest]# chmod u+s /home/guest/simpleid2
[root@rumiantceva guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя 12 17:46 simpleid2
[root@rumiantceva guest]# ./simpleid2
bash: ./simpleid2: Нет такого файла или каталога
[root@rumiantceva guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@rumiantceva guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@rumiantceva guest]#
```

Рис. 0.6: рис.6. Изменение владельца программы и установка SetUID-бита, проверка установки и изменения, запуск программы и команды id.

12. Проделала тоже самое относительно SetGID-бита (рис. 7)

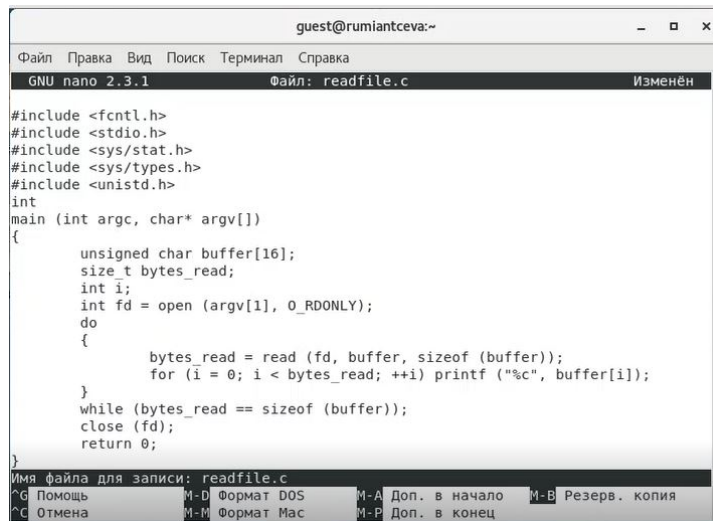
Установка SetGID-бита отражается к команде ls, а сравнение выполнения программы и команды id дало следующие результаты: действительные идентификаторы совпадают с выводом команды id - везде 0, то есть рут-пользователь. Но так же важно заметить, что эффективные идентификаторы отличны от действительных: пользователь - 0, группа - 1001.



```
[root@rumiantceva guest]# chmod g+s /home/guest/simpleid2
[root@rumiantceva guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 8576 ноя 12 17:46 simpleid2
[root@rumiantceva guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@rumiantceva guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@rumiantceva guest]#
```

Рис. 0.7: рис.7. Установка SetGID-бита, проверка установки, запуск программы и команды id.

13. Создала программу readfile.c (рис. 8).



```
guest@rumiantceva:~$ nano readfile.c
GNU nano 2.3.1 Файл: readfile.c Изменён

#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;
}
```

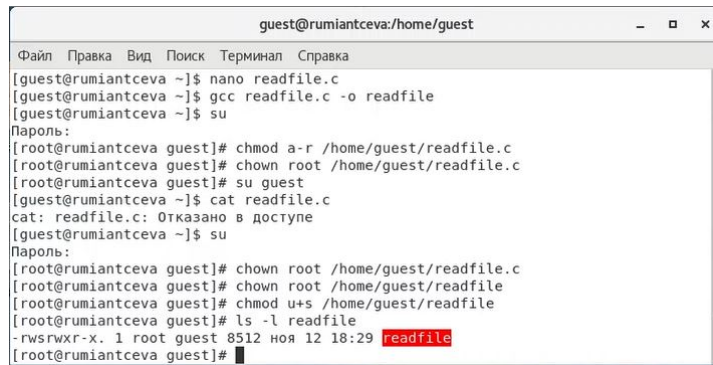
Рис. 0.8: рис.8. Программа readfile.c

14. Откомпилировала её командой `gcc readfile.c -o readfile` (рис. 9).

15. Сменила владельца у файла `readfile.c` (`chown`) и изменила права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог. Использовала `chmod a-r` (рис. 9).

16. Проверила, что пользователь `guest` не может прочитать файл `readfile.c` командой `cat` (рис. 9).

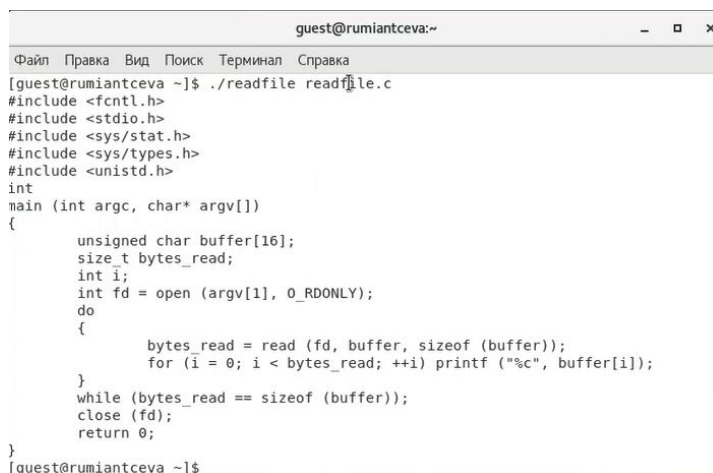
17. Сменила у программы `readfile` владельца и установила `SetUID`-бит (рис. 9).



```
guest@rumiantceva:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@rumiantceva ~]$ nano readfile.c
[guest@rumiantceva ~]$ gcc readfile.c -o readfile
[guest@rumiantceva ~]$ su
Пароль:
[root@rumiantceva guest]# chmod a-r /home/guest/readfile.c
[root@rumiantceva guest]# chown root /home/guest/readfile.c
[root@rumiantceva guest]# su guest
[guest@rumiantceva ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@rumiantceva ~]$ su
Пароль:
[root@rumiantceva guest]# chown root /home/guest/readfile.c
[root@rumiantceva guest]# chown root /home/guest/readfile
[root@rumiantceva guest]# chmod u+s /home/guest/readfile
[root@rumiantceva guest]# ls -l readfile
-rwsrwxr-x. 1 root guest 8512 ноя 12 18:29 readfile
[root@rumiantceva guest]#
```

Рис. 0.9: рис.9. Компиляция readfile и другие действия в соответствии с 14-17 пунктами.

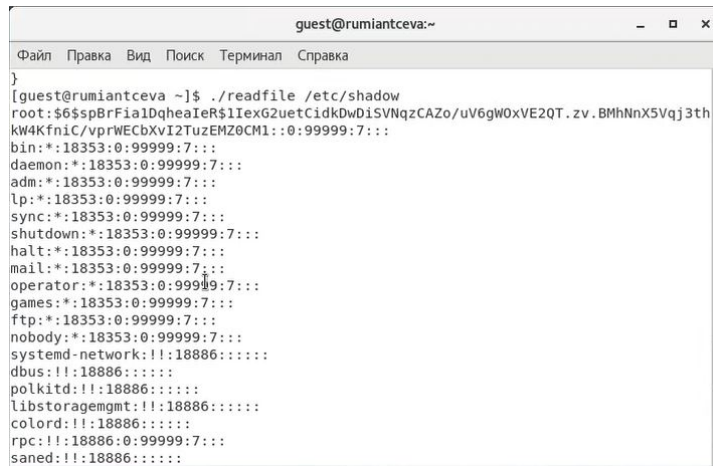
18. Проверила, может ли программа readfile прочитать файл readfile.c. Да, может (рис. 10).



```
guest@rumiantceva:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@rumiantceva ~]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;
    int fd = open (argv[1], O_RDONLY);
    do
    {
        bytes_read = read (fd, buffer, sizeof (buffer));
        for (i = 0; i < bytes_read; ++i) printf ("%c", buffer[i]);
    }
    while (bytes_read == sizeof (buffer));
    close (fd);
    return 0;
}
[guest@rumiantceva ~]$
```

Рис. 0.10: рис.10. Выполнение программы readfile с файлом readfile.c.

19. Проверила, может ли программа readfile прочитать файл /etc/shadow. Её выполнение возможно в том числе, так как владельцем файла являются root-пользователь (рис. 11).

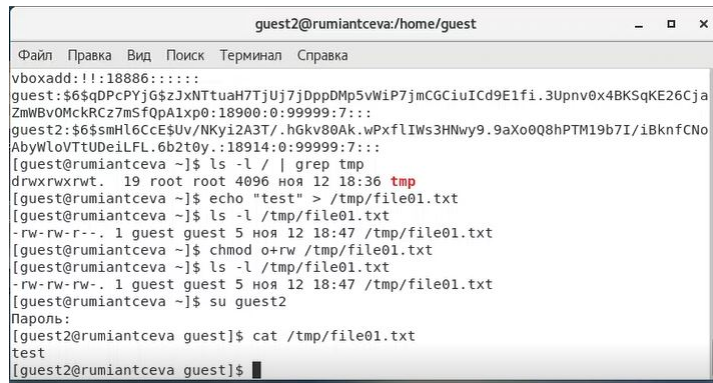


```
guest@rumiantceva:~$ ./readfile /etc/shadow
root:$6$spBrFialDqheaIeR$1IexG2uetCidkDwDiSVNqzCAZo/uV6gW0xVE2QT.zv.BMhNnX5Vqj3th
kW4KfniC/vprWEcbXvI2TuzEMZ0CM1::0:99999:7:::
bin:!:18353:0:99999:7:::
daemon:!:18353:0:99999:7:::
adm:!:18353:0:99999:7:::
lp:!:18353:0:99999:7:::
sync:!:18353:0:99999:7:::
shutdown:!:18353:0:99999:7:::
halt:!:18353:0:99999:7:::
mail:!:18353:0:99999:7:::
operator:!:18353:0:99999:7:::
games:!:18353:0:99999:7:::
ftp:!:18353:0:99999:7:::
nobody:!:18353:0:99999:7:::
systemd-network:!!:18886:::
dbus:!!:18886:::
polkitd:!!:18886:::
libstoragemgmt:!!:18886:::
colord:!!:18886:::
rpc:!!:18886:0:99999:7:::
saned:!!:18886:::
```

Рис. 0.11: рис.11. Выполнение программы readfile с файлом /etc/shadow.

Исследование Sticky-бита

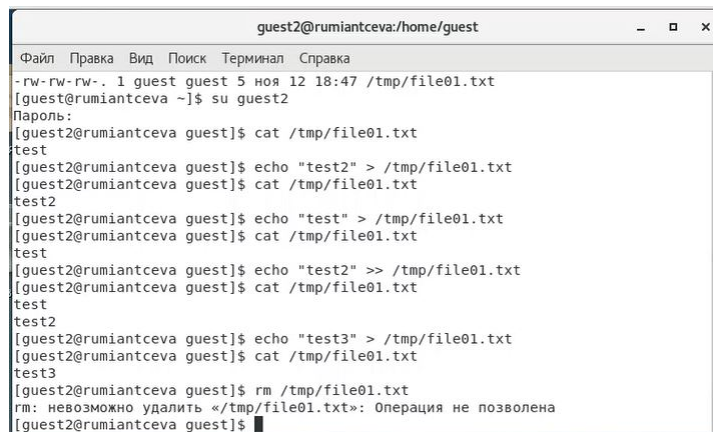
1. Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду `ls -l / | grep tmp` (рис. 12). Видим, что установлен, так как есть буква `t`.
2. От имени пользователя `guest` создала файл `file01.txt` в директории `/tmp` со словом `test` командой `echo "test" > /tmp/file01.txt` (рис. 12).
3. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные». Выполнила команды `ls -l /tmp/file01.txt` и `chmod o+rw /tmp/file01.txt` и `ls -l /tmp/file01.txt` (рис. 12).
4. От пользователя `guest2` (не являющегося владельцем) попробовала прочесть файл `/tmp/file01.txt` командой `cat /tmp/file01.txt` (рис. 12).



```
guest2@rumiantceva:/home/guest
vboxadd:!!!:18886:::
guest:$6$QDPCPYjG$zJxNTtuaH7TjUj7jDppDMp5vWiP7jmCGCiuICd9E1fi.3Upnv0x4BK5qKE26Cja
ZmWBv0MckRCz7mSfQpA1xp0:18900:0:99999:7:::
guest2:$6$smHl6CcE$Uv/NKy12A3T/.hGkv80Ak.wPxfLIWs3HNWY9.9aXo0Q8hPTM19b7I/iBknfCNO
AbyWloVTtUDeiLFL.6b2t0y.:18914:0:99999:7:::
[guest@rumiantceva ~]$ ls -l / | grep tmp
drwxrwxrwt. 19 root root 4096 ноя 12 18:36 tmp
[guest@rumiantceva ~]$ echo "test" > /tmp/file01.txt
[guest@rumiantceva ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 12 18:47 /tmp/file01.txt
[guest@rumiantceva ~]$ chmod o+rw /tmp/file01.txt
[guest@rumiantceva ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 12 18:47 /tmp/file01.txt
[guest@rumiantceva ~]$ su guest2
Пароль:
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test
[guest2@rumiantceva guest]$
```

Рис. 0.12: рис.12. Выполнение пунктов 1-4 исследования Sticky-бита .

5. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2 командой echo “test2” » /tmp/file01.txt. Мне удалось выполнить операцию (рис. 13).
6. Проверила содержимое файла командой cat /tmp/file01.txt (рис. 13).
7. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo “test3” > /tmp/file01.txt. Мне удалось выполнить операцию (рис. 13).
8. Проверила содержимое файла командой cat /tmp/file01.txt (рис. 13).
9. От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой rm /tmp/file01.txt. Мне нее удалось удалить файл (рис. 13).



```
guest2@rumiantceva:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
-rw-rw-rw-. 1 guest guest 5 ноя 12 18:47 /tmp/file01.txt
[guest@rumiantceva ~]$ su guest2
Пароль:
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test
[guest2@rumiantceva guest]$ echo "test2" > /tmp/file01.txt
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test2
[guest2@rumiantceva guest]$ echo "test" > /tmp/file01.txt
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test
[guest2@rumiantceva guest]$ echo "test2" >> /tmp/file01.txt
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test
test2
[guest2@rumiantceva guest]$ echo "test3" > /tmp/file01.txt
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test3
[guest2@rumiantceva guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@rumiantceva guest]$
```

Рис. 0.13: рис.13. Выполнение пунктов 5-9 исследования Sticky-бита .

Можем сделать вывод, что разрешена дозапись в файл, запись в файл, но мы не можем удалять файл из директории, на которую установлен атрибут Sticky.

10. Повысила свои права до суперпользователя следующей командой `su -` и выполнила после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp` (рис. 14).
11. Покинула режим суперпользователя командой `exit` (рис. 14).
12. От пользователя `guest2` проверила, что атрибута `t` у директории `/tmp` нет: `ls -l / | grep tmp` (рис. 14).
13. Повторила предыдущие шаги (рис. 14). Видим, что дозапись и запись также разрешены, но при этом удалось и удалить файл.

```
guest2@rumiantceva:/home/guest
Файл Правка Вид Поиск Терминал Справка
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test3
[guest2@rumiantceva guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@rumiantceva guest]$ su -
Пароль:
Последний вход в систему:Пт ноя 12 18:35:47 MSK 2021на pts/0
[root@rumiantceva ~]# chmod -t /tmp
[root@rumiantceva ~]# exit
logout
[guest2@rumiantceva guest]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 ноя 12 18:58 tmp
[guest2@rumiantceva guest]$ echo "test2" >> /tmp/file01.txt
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test3
test2
[guest2@rumiantceva guest]$ echo "test" > /tmp/file01.txt
[guest2@rumiantceva guest]$ cat /tmp/file01.txt
test
[guest2@rumiantceva guest]$ rm /tmp/file01.txt
[guest2@rumiantceva guest]$
```

Рис. 0.14: рис.14. Выполнение пунктов 10-13 исследования Sticky-бита .

14. Мне удалось удалить файл от имени пользователя, не являющегося его владельцем. Именно об этом и была часть теории выше.
15. Повысила свои права до суперпользователя и верните атрибут t на директорию /tmp: su -, chmod +t /tmp, exit (рис. 15).

```
guest2@rumiantceva:/home/guest
Файл Правка Вид Поиск Терминал Справка
systemd-private-92209102d52e4c95b1bb0c6606bcfae9-fprintd.service-xWmy43
systemd-private-92209102d52e4c95b1bb0c6606bcfae9-systemd-hostnamed.service-fRkV4p
systemd-private-ddfb57ee84da41f58f2746efa61589a1-bolt.service-20ZwUR
systemd-private-ddfb57ee84da41f58f2746efa61589a1-colord.service-4c7EQC
systemd-private-ddfb57ee84da41f58f2746efa61589a1-cups.service-5Ftlis
systemd-private-ddfb57ee84da41f58f2746efa61589a1-fwupd.service-onXrpy
systemd-private-ddfb57ee84da41f58f2746efa61589a1-rtkit-daemon.service-PmhY77
tracker-extract-files.1000
tracker-extract-files.1001
yum_save_tx.2021-10-28.19-19.06cIT3.yumtx
yum_save_tx.2021-11-12.16-06.2tTpe.yumtx
yum_save_tx.2021-11-12.16-12.1vbeWl.yumtx
[guest2@rumiantceva guest]$ su -
Пароль:
Последний вход в систему:Пт ноя 12 18:58:24 MSK 2021на pts/0
[root@rumiantceva ~]# chmod +t /tmp
[root@rumiantceva ~]# exit
logout
[guest2@rumiantceva guest]$ ls -l / | grep tmp
drwxrwxrwt. 20 root root 4096 ноя 12 19:02 tmp
[guest2@rumiantceva guest]$
```

Рис. 0.15: рис.15. Возвращение атрибута t на директорию.

Библиография

1. ТУИС РУДН
2. Статья “КОМАНДА CHMOD В LINUX (ПРАВА ДОСТУПА К ФАЙЛАМ)” на сайте baks.ru <https://baks.dev/article/terminal/chmod-command-in-linux-file-permissions>
3. Статья “Фундаментальные основы Linux” на сайте zalinux.ru http://rus-linux.net/MyLDP/BOOKS/Linux_Foundations/31/ch31.html

Выводы

Я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.