

# Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

---

Чванова Ангелина Дмитриевна

2024 год

Российский университет дружбы народов, Москва, Россия

## Докладчик

- Чванова Ангелина Дмитриевна
- студент
- Российский университет дружбы народов
- angelinachdm@gmail.com
- <https://adchvanova-new.github.io/ru/>



# 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

# Теоретическое введение

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.

# Выполнение лабораторной работы

## 5.2.1. Подготовка лабораторного стенда

```
[root@adchvanova ~]# yum install gcc
Rocky Linux 9 - BaseOS                    387 B/s | 4.1 kB    00:10
Rocky Linux 9 - BaseOS                    139 kB/s | 2.3 MB   00:16
Rocky Linux 9 - AppStream                  437 B/s | 4.5 kB    00:10
Rocky Linux 9 - AppStream                  451 kB/s | 8.0 MB   00:18
Rocky Linux 9 - Extras                     278 B/s | 2.9 kB    00:10
Package gcc-11.4.1-3.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@adchvanova ~]# setenforce 0
[root@adchvanova ~]# getenforce
Permissive
[root@adchvanova ~]#
```

Рис. 1: (рис. 1. Установка gss)

# Выполнение лабораторной работы

## 5.3.1 Создание программы

Войдите в систему от имени пользователя guest. Создайте программу simpleid.c.

```
[guest@adchvanova ~]$ mkdir lab5
[guest@adchvanova ~]$ cd lab5/
[guest@adchvanova lab5]$ touch simpleid.c
```



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t uid = geteuid ();
8     gid_t gid = getegid ();
9     printf ("uid=%d, gid=%d\n", uid, gid);
10    return 0;
11 }
```

# Выполнение лабораторной работы

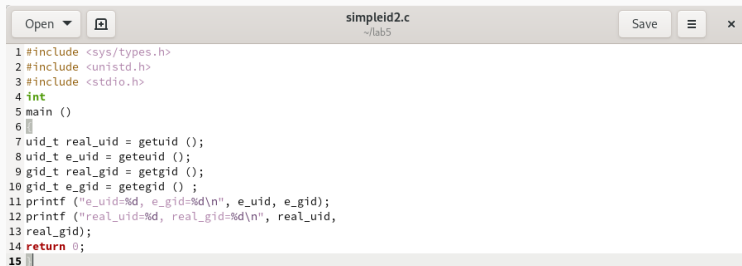
Скомпилируйте программу и убедитесь, что файл программы создан. Выполните программу simpleid. Выполните системную программу id

```
[guest@adchvanova lab5]$ gcc simpleid.c -o simpleid
[guest@adchvanova lab5]$ ./simpleid
uid=1001, gid=1001
[guest@adchvanova lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@adchvanova lab5]$
```

**Рис. 3:** (рис. 3. 3-5 пункты задания лабораторной)

# Выполнение лабораторной работы

Усложните программу, добавив вывод действительных идентификаторов.



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t real_uid = getuid ();
8     uid_t e_uid = geteuid ();
9     gid_t real_gid = getgid ();
10    gid_t e_gid = getegid () ;
11    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
12    printf ("real_uid=%d, real_gid=%d\n", real_uid,
13    real_gid);
14    return 0;
15 }
```

**Рис. 4:** (рис. 4. simpleid2.c)



# Выполнение лабораторной работы

Скомпилируйте и запустите simpleid2.c: `gcc simpleid2.c -o simpleid2`  
`./simpleid2`

```
[guest@adchvanova lab5]$ touch simpleid2.c
[guest@adchvanova lab5]$ gcc simpleid2.c -o simpleid2
[guest@adchvanova lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@adchvanova lab5]$
```

**Рис. 5:** (рис. 5. 7 пункт задания лабораторной)

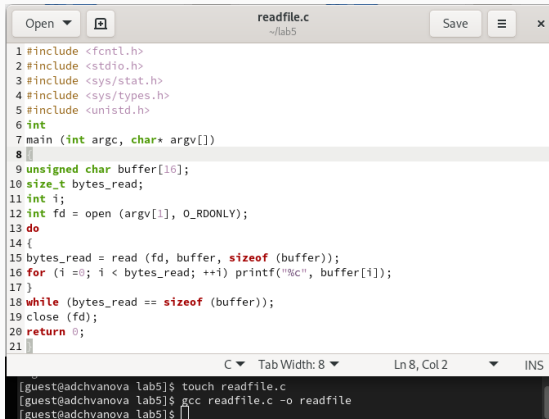
# Выполнение лабораторной работы

От имени суперпользователя выполнила команды “sudo chown root:guest /home/guest/simpleid2” и “sudo chmod u+s /home/guest/simpleid2”, затем выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/simpleid2”. Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2

```
[root@adchvanova /]# cd /home/guest/lab5/
[root@adchvanova lab5]# chown root:guest simpleid2
[root@adchvanova lab5]# chmod u+s simpleid2
[root@adchvanova lab5]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 24488 Oct  3 23:08 simpleid2
[root@adchvanova lab5]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@adchvanova lab5]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
```

# Выполнение лабораторной работы

Создайте программу readfile.c Откомпилируйте её.



The screenshot shows a code editor window titled 'readfile.c' with the file path '~ /lab5'. The editor contains the following C code:

```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6 int
7 main (int argc, char* argv[])
8 {
9     unsigned char buffer[16];
10    size_t bytes_read;
11    int i;
12    int fd = open (argv[1], O_RDONLY);
13    do
14    {
15        bytes_read = read (fd, buffer, sizeof (buffer));
16        for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
17    }
18    while (bytes_read == sizeof (buffer));
19    close (fd);
20    return 0;
21 }
```

Below the code editor is a terminal window showing the following commands and output:

```
[guest@adchvanova lab5]$ touch readfile.c
[guest@adchvanova lab5]$ gcc readfile.c -o readfile
[guest@adchvanova lab5]$
```

Рис. 7: (рис. 7. readfile.c)

# Выполнение лабораторной работы

Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```
[guest@adchvanova lab5]$ su
Password:
[root@adchvanova lab5]# chown root:guest readfile
[root@adchvanova lab5]# chmod 700 readfile
[root@adchvanova lab5]# chown root:guest readfile
[root@adchvanova lab5]# chmod -r readfile.c
[root@adchvanova lab5]# chmod u+s readfile
[root@adchvanova lab5]#
```

**Рис. 8:** (рис. 8. chmod)

## Выполнение лабораторной работы

Проверьте, что пользователь guest не может прочитать файл readfile.c. Смените у программы readfile владельца и установите SetU'D-бит. Проверьте, может ли программа readfile прочитать файл readfile.c? Проверьте, может ли программа readfile прочитать файл /etc/shadow? Отрадите полученный результат и ваши объяснения в отчёте.

```
[guest@adchvanova lab5]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@adchvanova lab5]$ ./readfile readfile.c
bash: ./readfile: Permission denied
[guest@adchvanova lab5]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
[guest@adchvanova lab5]$
```

**Рис. 9:** (рис. 9. 16-19 пункты Guest)

# Выполнение лабораторной работы

От имени суперпользователя все команды удастся выполнить.

```
[root@adchvanova lab5]# cat 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;
}
[root@adchvanova lab5]# ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int
```

**Рис. 10:** (рис. 10. 16-18 пункты суперпользователь)

# Выполнение лабораторной работы

```
[root@adchvanova lab5]# ./readfile /etc/shadow
root:$6$FhSAvo7hjKTLM1RQ$It68woyqTnnbUrelK29VcARNL2nu3VqzRXHukEpIvb/u5S8uQHJVSK
f12TxhYlenqW/q0lw20DSpl0bfqFKP.:0:99999:7:::
bin:!:19820:0:99999:7:::
daemon:!:19820:0:99999:7:::
adm:!:19820:0:99999:7:::
lp:!:19820:0:99999:7:::
sync:!:19820:0:99999:7:::
shutdown:!:19820:0:99999:7:::
halt:!:19820:0:99999:7:::
mail:!:19820:0:99999:7:::
operator:!:19820:0:99999:7:::
```

**Рис. 11:** (рис. 11. 19 пункт суперпользователь)

# Выполнение лабораторной работы

## 5.3.2. Исследование Sticky-бита

Выясните, установлен ли атрибут Sticky на директории /tmp. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»

```
[guest@adchvanova lab5]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 Oct  3 23:21 tmp
[guest@adchvanova lab5]$ echo "test" > /tmp/file01.txt
[guest@adchvanova lab5]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  3 23:23 /tmp/file01.txt
[guest@adchvanova lab5]$ chmod o+rw /tmp/file01.txt
[guest@adchvanova lab5]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  3 23:23 /tmp/file01.txt
[guest@adchvanova lab5]$
```

Рис. 12: (рис. 12. 1-3 пункты)



# Выполнение лабораторной работы

От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt. От пользователя guest2 попробуйте дозаписать в файл. Проверьте содержимое файла командой. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm /tmp/file01.txt. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет:

```
[guest@adchvanova lab5]$ su guest2
Password:
[guest2@adchvanova lab5]$ cat /tmp/file01.txt
test
[guest2@adchvanova lab5]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@adchvanova lab5]$ cat /tmp/file01.txt
test
[guest2@adchvanova lab5]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@adchvanova lab5]$ cat /tmp/file01.txt
```

## Выполнение лабораторной работы

Повторите предыдущие шаги. Какие наблюдаются изменения. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`

```
[guest2@adchvanova lab5]$ su -  
Password:  
[root@adchvanova ~]# chmod +t /tmp  
[root@adchvanova ~]# exit  
logout  
[guest2@adchvanova lab5]$
```

Рис. 14: (рис. 14. Возвращение атрибута)

Были изучены механизмы изменения идентификаторов и применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Были рассмотрены работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

# Список литературы. Библиография

[0] Методические материалы курса

[1] Дополнительные атрибуты: <https://tokmakov.msk.ru/blog/item/141>

[2] Компилятор GSS: <http://parallel.imm.uran.ru/freesoft/make/instrum.html>