

# Лабораторная работа №5

---

“Низамова Альфия Айдаровна. НФИбд-01-20”<sup>1</sup>

23 сентября, 2023, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цель работы

---

## 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.

**Ход лабораторной работы.**  
**Создание программы.**

---

Войдем в систему от имени пользователя guest

```
[guest@aanizamovalocaldomain ~]$ touch simpleid.c  
[guest@aanizamovalocaldomain ~]$ █
```

**Рис. 1:** Рис. 1

Создадим программу simpleid.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;
}
```

Рис. 2: Рис. 2

Скомпилируем программу и убедимся, что файл программы создан

Выполним программу simpleid

Выполним системную программу id и сравним полученный результат с данными предыдущего пункта задания

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

**Рис. 3:** Рис. 3

Усложним программу, добавив вывод действительных идентификаторов. Получившуюся программу назовем 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;
}
```



Скомпилируем и запустим simpleid2.c

От имени суперпользователя выполним заданные команды

Используем sudo или повысим временно свои права с помощью su

```
[guest@aanizamovalocaldomain ~]$ gcc simpleid2.c -o simpleid2
[guest@aanizamovalocaldomain ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aanizamovalocaldomain ~]$ su root
Password:
[root@aanizamovalocaldomain guest]# chown root:guest /home/guest/simpleid2
[root@aanizamovalocaldomain guest]# chmod u+s /home/guest/simpleid2
```

Рис. 5: Рис. 5

Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2

Запустим simpleid2 и id. Сравним результаты.

```
[guest@aanizamovalocaldomain ~]$ su root
Password:
[root@aanizamovalocaldomain guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 Oct  5 02:14 simpleid2
[root@aanizamovalocaldomain guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@aanizamovalocaldomain guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
```

Рис. 6: Рис. 6

## Проделаем то же самое относительно SetGID-бита

```
[root@aanizamovalocaldomain guest]# su guest
[guest@aanizamovalocaldomain ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aanizamovalocaldomain ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@aanizamovalocaldomain ~]$ su root
Password:
[root@aanizamovalocaldomain guest]# chmod g+s /home/guest/simpleid2
[root@aanizamovalocaldomain guest]# su guest
[guest@aanizamovalocaldomain ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 7: Рис. 7

## Создадим программу 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[i], 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));
```

Откомпилируем её

Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог

Проверим, что пользователь guest не может прочитать файл readfile.c

```
[guest@aanizamovalocaldomain ~]$ gcc readfile.c -o readfile
[guest@aanizamovalocaldomain ~]$ su root
Password:
[root@aanizamovalocaldomain guest]# chown root /home/guest/readfile.c
[root@aanizamovalocaldomain guest]# chmod 700 /home/guest/readfile.c
[root@aanizamovalocaldomain guest]# su guest
[guest@aanizamovalocaldomain ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

**Рис. 9:** Рис. 9



Проверим, может ли программа readfile прочитать файл /etc/shadow

```
[guest@aanizamovalocaldomain ~]$ ./readfile /etc/shadow
#####00000000 @ @>>@>@<@.>@<@8@8@ XX@X@D$@cd88@8@ P@cd @ @,,Q@cdR@cd.
>@>/lib64/ld-linux-x86-64.so.2GNU@GNU@GNU@00P-@#####NU#L h w read_libc_sta
rt_maincloseopenputcharlibc.so.6GLIBC_2.34GLIBC_2.2.5_ITM_deregisterTMCLoneTable
_gmon_start _ITM_registerTMCLoneTable+####ou@ @@@@&@&@&@ @@@(##00@&@H@H@H
```

Рис. 11: Рис. 11

## **Ход лабораторной работы. Исследование Sticky-бита.**

---



Выясним, установлен ли атрибут Sticky на директории /tmp

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test

Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «ВСЕ ОСТАЛЬНЫЕ»

```
[guest@aanizamovalocaldomain ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  5 02:48 tmp
[guest@aanizamovalocaldomain ~]$ echo "test" > /tmp/file01.txt
[guest@aanizamovalocaldomain ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  5 02:53 /tmp/file01.txt
[guest@aanizamovalocaldomain ~]$ chmod o+rw /tmp/file01.txt
[guest@aanizamovalocaldomain ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  5 02:53 /tmp/file01.txt
[guest@aanizamovalocaldomain ~]$ su guest2
```

## 4-6 От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt