

# Лабораторная работа № 5

---

Тарусов Артём Сергеевич

2023, Москва

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.

1. Исследовать SetUID- и SetGID-биты.
2. Исследовать Sticky-бит.

От имени пользователя `guest` создадим программу `simpleid.c`, скомпилируем ее и убедимся, что файл создан.

```
[guest@user progs]$ touch simpleid.c  
[guest@user progs]$ gcc simpleid.c -o simpleid  
[guest@user progs]$ ls  
simpleid  simpleid.c
```

**Рис. 1:** Создание файла `simpleid.c`

Выполним команды `./simpleid` и `id` и убедимся, что полученные данные совпадают.

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

**Рис. 2:** Использование команд `./simpleid` и `id`

Усложним программу и запишем ее в файл `simpleid2.c`. Запустим получившуюся программу.

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

**Рис. 3:** Создание и запуск программы `simpleid2`

От имени суперпользователя установим новые атрибуты и сменим владельца файла simpleid2.

```
[guest@user progs]$ su
Password:
[root@user progs]# chown root:guest /home/guest/simpleid2
chown: cannot access '/home/guest/simpleid2': No such file or directory
[root@user progs]# chown root:guest simpleid2
[root@user progs]# chmod u+s simpleid2
[root@user progs]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 Oct  5 14:23 simpleid2
```

**Рис. 4:** Установки новых атрибутов и смена владельца файла simpleid2

Выполним команды `./simpleid2` и `id` и убедимся, что полученные данные совпадают.

```
[root@user progs]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@user progs]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

**Рис. 5:** Использование команд `./simpleid2` и `id`



Проделаем то же самое относительно SetGID-бита.

```
[root@user progs]# chmod g+s simpleid2
[root@user progs]# ls -l simpleid2
-rwsr-sr-x. 1 root guest 26064 Oct  5 14:23 simpleid2
[root@user progs]# exit
exit
[guest@user progs]$ ^M
: command not found...
[guest@user progs]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@user progs]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

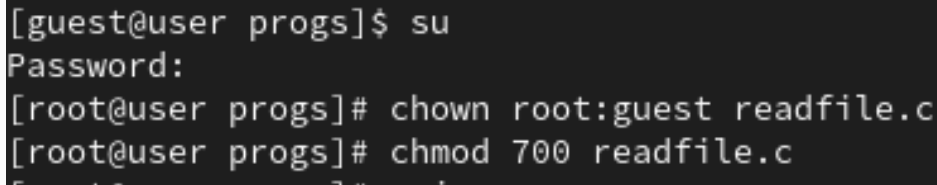
**Рис. 6:** Операции с SetGID-битом

Создадим и скомпилируем программу readfile.c.

```
[guest@user progs]$ touch readfile.c  
[guest@user progs]$ gcc readfile.c -o readfile
```

**Рис. 7:** Создание и компиляция программы readfile.c

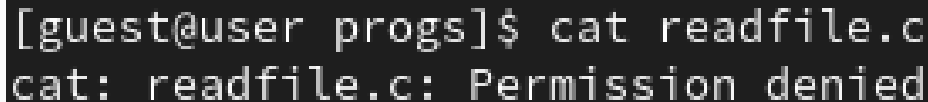
Сменим владельца у файла `readfile.c` и изменим права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог.

A terminal window with a dark background and light gray text. The prompt is [guest@user progs]\$ and the command is su. The prompt changes to [root@user progs]# after the password is entered. The command chown root:guest readfile.c is entered, followed by chmod 700 readfile.c.

```
[guest@user progs]$ su
Password:
[root@user progs]# chown root:guest readfile.c
[root@user progs]# chmod 700 readfile.c
```

**Рис. 8:** Изменение владельца и прав файла `readfile.c`

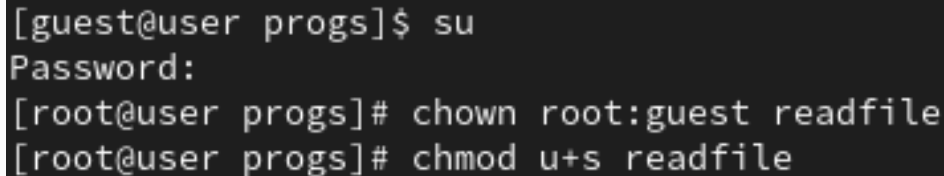
Проверим, что пользователь `guest` не может прочитать файл `readfile.c`.

A terminal window with a black background and white text. The prompt is `[guest@user progs]$`. The user has entered the command `cat readfile.c`. The output of the command is `cat: readfile.c: Permission denied`.

```
[guest@user progs]$ cat readfile.c
cat: readfile.c: Permission denied
```

**Рис. 9:** Проверка, что пользователь `guest` не может прочитать файл `readfile.c`.

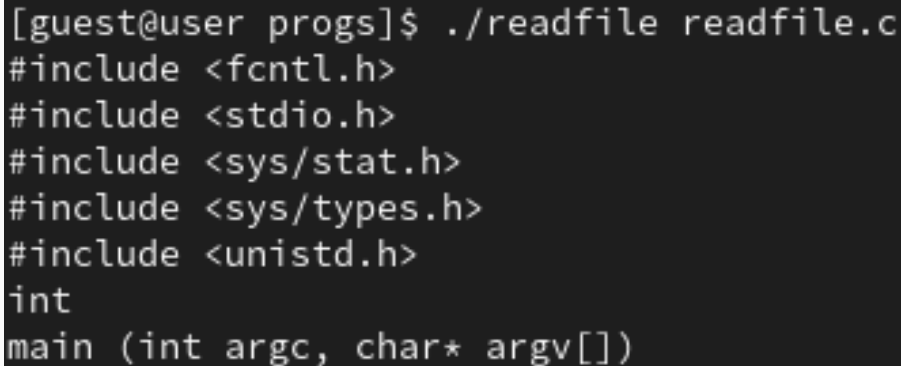
Сменим у программы readfile владельца и установим SetUID-бит.



```
[guest@user progs]$ su
Password:
[root@user progs]# chown root:guest readfile
[root@user progs]# chmod u+s readfile
```

**Рис. 10:** Работа с параметрами readfile

Проверим, может ли программа readfile прочитать файл readfile.c.



```
[guest@user progs]$ ./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[])
{
```

**Рис. 11:** Попытка прочитать файл readfile.c программой readfile

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

```
[guest@user progs]$ ./readfile /etc/shadow
root:$6$uzGylv1w8gk3IDI8$r0qN7B6vNyXUlgVDry0j906qXjjd2LsgS70tU.IBTWvfKD/IQ0f4G5v0GzUKnIPS030pVV7s/
999:7:::
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:::
shutdown:!:19469:0:99999:7:::
halt:!:19469:0:99999:7:::
mail:!:19469:0:99999:7:::
operator:!:19469:0:99999:7:::
games:!:19469:0:99999:7:::
ftp:!:19469:0:99999:7:::
nobody:!:19469:0:99999:7:::
systemd-coredump:!!!:19608:!!!!:
```

**Рис. 12:** Попытка прочитать файл /etc/shadow программой readfile

Выясним, установлен ли атрибут Sticky на директории /tmp.

```
[guest@user progs]$ ls -l / | grep tmp  
drwxrwxrwt. 16 root root 4096 Oct  5 14:40 tmp
```

**Рис. 13:** Чтение атрибутов директории /tmp

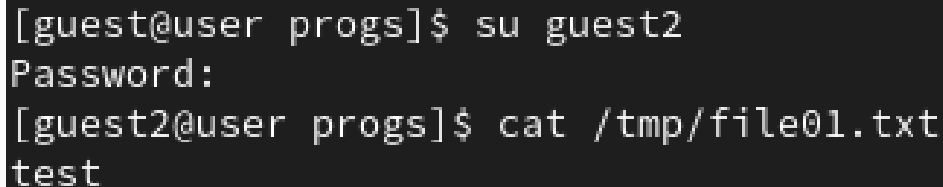


От имени пользователя `guest` создадим файл `file01.txt` в директории `/tmp` со словом `test`. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные».

```
[guest@user progs]$ echo "test" > /tmp/file01.txt
[guest@user progs]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  5 14:44 /tmp/file01.txt
[guest@user progs]$ chmod o+rw /tmp/file01.txt
[guest@user progs]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  5 14:44 /tmp/file01.txt
```

**Рис. 14:** Чтение атрибутов директории `/tmp`

От пользователя guest2 попробуем прочитать файл /tmp/file01.txt.

A terminal window with a black background and white text. The first line shows the prompt [guest@user progs]\$ followed by the command su guest2. The second line shows the prompt Password: followed by a blank line. The third line shows the prompt [guest2@user progs]\$ followed by the command cat /tmp/file01.txt. The fourth line shows the output test.

```
[guest@user progs]$ su guest2
Password:
[guest2@user progs]$ cat /tmp/file01.txt
test
```

**Рис. 15:** Попытка прочтения файла /tmp/file01.txt

От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2.

```
[guest2@user progs]$ echo "test2" >> /tmp/file01.txt  
[guest2@user progs]$ cat /tmp/file01.txt  
test2  
test2
```

**Рис. 16:** Попытка дозаписи в файл /tmp/file01.txt

От пользователя `guest2` попробуем записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию).

```
[guest2@user progs]$ echo "test3" > /tmp/file01.txt  
[guest2@user progs]$ cat /tmp/file01.txt  
test3
```

**Рис. 17:** Попытка записи в файл `/tmp/file01.txt`

От пользователя guest2 попробуем удалить файл /tmp/file01.txt.

```
[guest2@user progs]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

**Рис. 18:** Попытка удаления файла /tmp/file01.txt

От имени суперпользователя снимем атрибут `t` с директории `/tmp`. От пользователя `guest2` проверим, что атрибута `t` у директории `/tmp` нет.

```
[guest2@user progs]$ su
Password:
[root@user progs]# chmod -t /tmp
[root@user progs]# exit
exit
[guest2@user progs]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Oct  5 15:00 tmp
```

Рис. 19: Удаление атрибута `t` директории `/tmp`

Повторим предыдущие шаги. Теперь мы можем удалить файл.

```
[guest2@user progs]$ echo "test2" >> /tmp/file01.txt
[guest2@user progs]$ cat /tmp/file01.txt
test3
test2
[guest2@user progs]$ echo "test3" > /tmp/file01.txt
[guest2@user progs]$ cat /tmp/file01.txt
test3
[guest2@user progs]$ rm /tmp/file0l.txt
rm: cannot remove '/tmp/file0l.txt': No such file or directory
[guest2@user progs]$ rm /tmp/file01.txt
```

**Рис. 20:** Повторение предыдущих шагов

Повысим свои права до суперпользователя и вернем атрибут `t` на директорию `/tmp`.

```
[guest2@user progs]$ su
Password:
[root@user progs]# chmod +t /tmp
[root@user progs]# exit
exit
```

**Рис. 21:** Возвращение атрибута `t` директории `/tmp`



В рамках данной лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрены принципы работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.