

# Отчёт по лабораторной работе №5

---

Аветисян Давид Артурович

7 октября 2023

РУДН, Москва, Россия

## Отчет по лабораторной работе №5

---

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

SetUID, SetGID и Sticky - это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги. • SetUID (set user ID upon execution — «установка ID пользователя во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца исполняемого файла. • SetGID (set group ID upon execution — «установка ID группы во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами группы исполняемого файла. • Sticky bit в основном используется в общих каталогах, таких как /var или /tmp, поскольку пользователи могут создавать файлы, читать и выполнять их, принадлежащие другим пользователям, но не могут удалять файлы, принадлежащие другим пользователям.

## 1 часть: Создание программы

Для начала мы убеждаемся, что компилятор gcc установлен, используя команду “gcc -v”. Затем отключаем систему запретов до очередной перезагрузки системы командой “sudo setenforce 0”, после чего команда “getenforce” выводит “Permissive”.

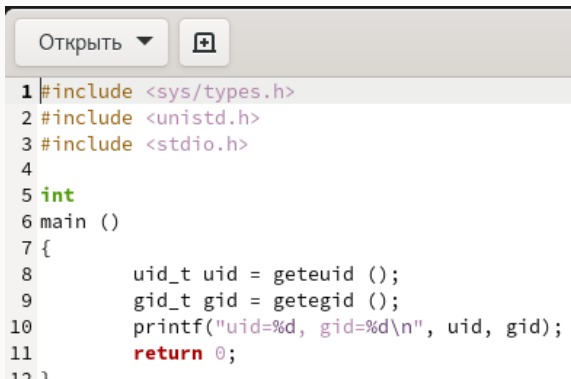
```
[daavetisyan@daavetisyan ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --enable-bootstrap --enable-host-p
g/ --enable-shared --enable-threads=posix --enable-checking=release --w
--enable-initfini-array --without-isl --enable-multilib --with-linker-h
--arch_32=x86_64 --build=x86_64-redhat-linux --with-build-config=bootstr
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.3.1 20211121 (Red Hat 11.3.1-4) (GCC)
[daavetisyan@daavetisyan ~]$ sudo setenforce 0
```

Мы полагаем, что ваш системный администратор изложил вам основы безопасности. Как правило, всё сводится к трём следующим правилам:

- №1) Уважайте частную жизнь других.
- №2) Думайте, прежде что-то вводить.
- №3) С большой властью приходит большая ответственность.

```
[sudo] пароль для daavetisyan:
[daavetisyan@daavetisyan ~]$ getenforce
Permissive
[daavetisyan@daavetisyan ~]$
```

Код программы выглядит следующим образом.



The image shows a screenshot of a code editor window. At the top, there is a toolbar with a button labeled "Открыть" (Open) and a plus icon. Below the toolbar, the code is displayed with line numbers on the left. The code is in C and includes headers for system types, unistd, and stdio. It defines a main function that calls geteuid and getegid to retrieve the effective user and group IDs, prints them using printf, and then returns 0.

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

Figure 2: Рисунок 2

Скомпилируем программу и убедимся, что файл программы был создан командой “gcc simpleid.c -o simpleid”. Выполняем программу simpleid командой “./simpleid”, а затем системную программу id командой “id”. Результаты, полученные в результате выполнения обеих команд, совпадают(uid=1001 и gid=1001).

```
[guest@daavetisyan lab05]$ gcc simpleid.c -o simpleid
[guest@daavetisyan lab05]$ ./simpleid
uid=1001, gid=1001
[guest@daavetisyan lab05]$ id
uid=1001(guest) gid=1001(guest) rpynmw=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@daavetisyan lab05]$
```

Figure 3: Рисунок 3

От имени суперпользователя выполняем команды “sudo chown root:guest /home/guest/lab05/simpleid2” и “sudo chmod u+s /home/guest/lab05/simpleid2”, затем выполняем проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/lab05/simpleid2”. Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.

```
[daavetisyan@daavetisyan ~]$ sudo chown root: /home/guest/lab05/simpleid2
[sudo] пароль для daavetisyan:
[daavetisyan@daavetisyan ~]$ sudo chmod u+s /home/guest/lab05/simpleid2
[daavetisyan@daavetisyan ~]$ sudo ls -l /home/guest/lab05/simpleid2
-rwsr-xr-x. 1 root root 26064 окт  9 19:08 /home/guest/lab05/simpleid2
[daavetisyan@daavetisyan ~]$
```

Figure 4: Рисунок 4

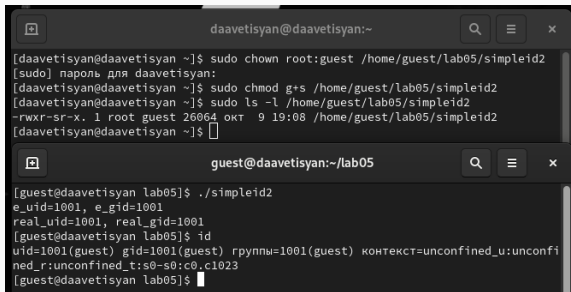


Запускаем программы simpleid2 и id. Теперь появились различия в uid.

```
[guest@daavetisyan lab05]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@daavetisyan lab05]$ id  
uid=1001(guest) gid=1001(guest) rpyнu=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@daavetisyan lab05]$
```

Figure 5: Рисунок 5

Прделаем тоже самое относительно SetGID-бита. Также можем заметить различия с предыдущим пунктом.

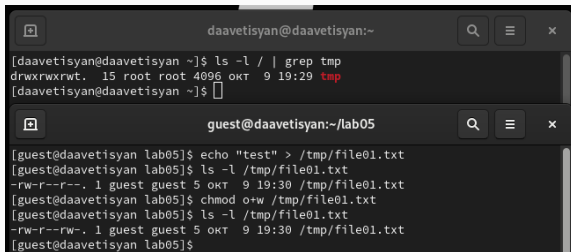


```
daavetisyan@daavetisyan:~  
[daavetisyan@daavetisyan ~]$ sudo chown root:guest /home/guest/lab05/simpleid2  
[sudo] пароль для daavetisyan:  
[daavetisyan@daavetisyan ~]$ sudo chmod g+s /home/guest/lab05/simpleid2  
[daavetisyan@daavetisyan ~]$ sudo ls -l /home/guest/lab05/simpleid2  
-rwxr-sr-x. 1 root guest 26064 окт 9 19:08 /home/guest/lab05/simpleid2  
[daavetisyan@daavetisyan ~]$  
  
guest@daavetisyan:~/lab05  
[guest@daavetisyan lab05]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@daavetisyan lab05]$ id  
uid=1001(guest) gid=1001(guest) rпппы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@daavetisyan lab05]$
```

Figure 6: Рисунок 6

## 2 часть: Исследование Sticky-бита

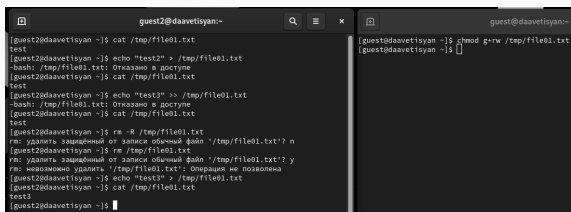
Командой “ls -l / | grep tmp” убеждаемся, что атрибут Sticky на директории /tmp установлен. От имени пользователя guest создаём файл file01.txt в директории /tmp со словом test командой “echo”test” > /tmp/file01.txt”. Просматриваем атрибуты у только что созданного файла и разрешаем чтение и запись для категории пользователей “все остальные” командами “ls -l /tmp/file01.txt” и “chmod o+rw /tmp/file01.txt”.



```
daavetisyan@daavetisyan:~  
[daavetisyan@daavetisyan ~]$ ls -l / | grep tmp  
drwxrwxrwt. 15 root root 4096 окт 9 19:29 tmp  
[daavetisyan@daavetisyan ~]$  
  
guest@daavetisyan:~/lab05  
[guest@daavetisyan lab05]$ echo "test" > /tmp/file01.txt  
[guest@daavetisyan lab05]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 окт 9 19:30 /tmp/file01.txt  
[guest@daavetisyan lab05]$ chmod o+w /tmp/file01.txt  
[guest@daavetisyan lab05]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 окт 9 19:30 /tmp/file01.txt  
[guest@daavetisyan lab05]$
```

Figure 7: Рисунок 7

От имени пользователя guest2 пробуем прочитать файл командой “cat /tmp/file01.txt” - это удалось. Далее пытаемся дозаписать в файл слово test2, проверить содержимое файла и записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию - эти операции удалось выполнить только в случае, если еще дополнительно разрешить чтение и запись для группы пользователей командой “chmod g+rw /tmp/file01.txt”. От имени пользователя guest2 пробуем удалить файл - это не удастся ни в каком из случаев, возникает ошибка.



```
guest2@daavetisyan:~  
[guest2@daavetisyan ~]$ cat /tmp/file01.txt  
test  
[guest2@daavetisyan ~]$ echo "test2" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Отказано в доступе  
[guest2@daavetisyan ~]$ cat /tmp/file01.txt  
test  
[guest2@daavetisyan ~]$ echo "test3" >> /tmp/file01.txt  
-bash: /tmp/file01.txt: Отказано в доступе  
[guest2@daavetisyan ~]$ cat /tmp/file01.txt  
test  
[guest2@daavetisyan ~]$ rm -f /tmp/file01.txt  
rm: удалить защищенный от записи обычный файл '/tmp/file01.txt'? n  
[guest2@daavetisyan ~]$ rm /tmp/file01.txt  
rm: удалить защищенный от записи обычный файл '/tmp/file01.txt'? y  
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена  
[guest2@daavetisyan ~]$ echo "test3" > /tmp/file01.txt  
[guest2@daavetisyan ~]$ cat /tmp/file01.txt  
test3  
[guest2@daavetisyan ~]$
```

```
guest2@daavetisyan:~  
[guest2@daavetisyan ~]$ chmod g+rw /tmp/file01.txt  
[guest2@daavetisyan ~]$
```

Figure 8: Рисунок 8

Повышаем права до суперпользователя командой “su -” и выполняем команду, снимающую атрибут t с директории /tmp “chmod -t /tmp”. После чего покидаем режим суперпользователя командой “exit”. Повторяем предыдущие шаги. Теперь нам удаётся удалить файл file01.txt от имени пользователя, не являющегося его владельцем.

```
[guest2@daavetisyan ~]$ su -
Пароль:
[root@daavetisyan ~]# chmod -t /tmp/
[root@daavetisyan ~]# exit
выход
[guest2@daavetisyan ~]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 окт 9 19:37 tmp
[guest2@daavetisyan ~]$ cat /tmp/file01.txt
test3
[guest2@daavetisyan ~]$ echo "test2" >> /tmp/file01.txt
[guest2@daavetisyan ~]$ cat /tmp/file01.txt
test3
test2
[guest2@daavetisyan ~]$ echo "test3" > /tmp/file01.txt
[guest2@daavetisyan ~]$ cat /tmp/file01.txt
test3
[guest2@daavetisyan ~]$ rm /tmp/file01.txt
[guest2@daavetisyan ~]$ ls -l /tmp/
total 0
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-chronyd.service-5fiQz0
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-color.service-v22a3
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-dbus-broker.service-gruWJF
drwx----- 3 root root 17 окт 9 18:48 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-fuupd.service-HVUTWA
drwx----- 3 root root 17 окт 9 18:52 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-geoclue.service-Pu0Gnk
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-ModemManager.service-q2H5dv
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-power-profiles-daemon.service-1B0R5z
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-rtkit-daemon.service-9bhzc
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-switcheroo-control.service-6L4sd
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-systemd-logind.service-Cukjg
drwx----- 3 root root 17 окт 9 18:47 systemd-private-79bc9477a8874b5b8c1cd2112a9c82a-upower.service-GPTbtr
[guest2@daavetisyan ~]$
```

Figure 9: Рисунок 9

- В ходе выполнения данной лабораторной работы я изучил механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.