

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

---

Мегегхо Меконтчу Надэж НПИбд-02-19

3 октября, 2022, Москва, Россия

Российский Университет Дружбы Народов

## Цели и задачи

---

- SUID - разрешение на установку идентификатора пользователя. Это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла.
- SGID - разрешение на установку идентификатора группы. Принцип работы очень похож на SUID с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.

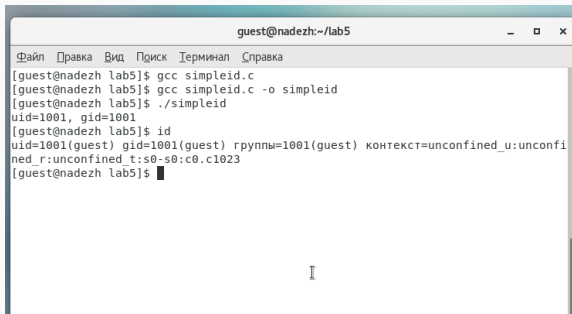
## Цель лабораторной работы

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.

# **Выполнение лабораторной работы**

---

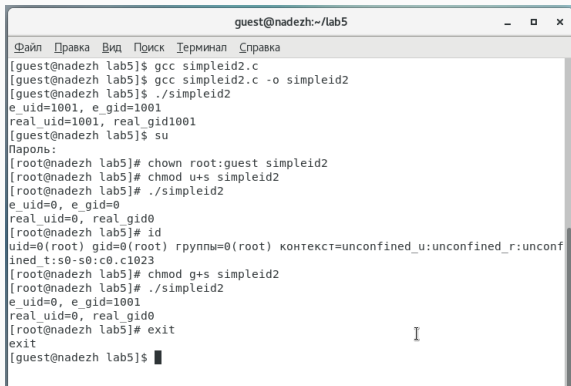
# Программа simpleid



```
guest@nadezh: ~/lab5
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@nadezh lab5]$ gcc simpleid.c
[guest@nadezh lab5]$ gcc simpleid.c -o simpleid
[guest@nadezh lab5]$ ./simpleid
uid=1001, gid=1001
[guest@nadezh lab5]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@nadezh lab5]$
```

**Figure 1:** результат программы simpleid

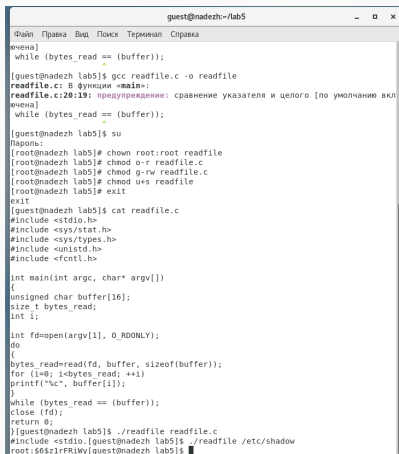
# Программа simpleid2



```
guest@nadezh:~/lab5
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@nadezh lab5]$ gcc simpleid2.c
[guest@nadezh lab5]$ gcc simpleid2.c -o simpleid2
[guest@nadezh lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@nadezh lab5]$ su
Пароль:
[root@nadezh lab5]# chown root:guest simpleid2
[root@nadezh lab5]# chmod u+s simpleid2
[root@nadezh lab5]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@nadezh lab5]# id
uid=0(root) gid=0(root) rгруппы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@nadezh lab5]# chmod g+s simpleid2
[root@nadezh lab5]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@nadezh lab5]# exit
exit
[guest@nadezh lab5]$
```

Figure 2: результат программы simpleid2

# Программа readfile



```
guest@nadezh:~/lab5
Файл Правка Вид Поиск Терминал Справка
очена]
while (bytes_read == (buffer));

[guest@nadezh lab5]$ gcc readfile.c -o readfile
readfile.c: 8 функции «main»:
readfile.c:20:19: предупреждение: сравнение указателя и целого [по умолчанию вкл
очена]
while (bytes_read == (buffer));

[guest@nadezh lab5]$ su
Напонь:
[root@nadezh lab5]# chown root:root readfile
[root@nadezh lab5]# chmod o-r readfile.c
[root@nadezh lab5]# chmod g-rw readfile.c
[root@nadezh lab5]# chmod u+s readfile
[root@nadezh lab5]# exit
exit
[guest@nadezh lab5]$ cat readfile.c
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>

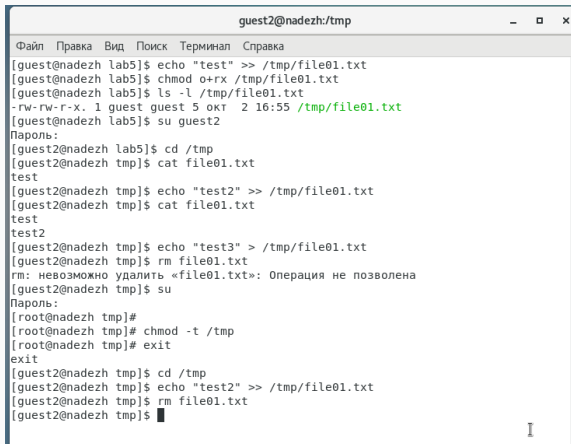
int main(int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    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 == (buffer));
    close (fd);
    return 0;
}[guest@nadezh lab5]$ ./readfile readfile.c
#include <stdio.h>[guest@nadezh lab5]$ ./readfile /etc/shadow
root:$6$zlrFRiWy[guest@nadezh lab5]$
```

Figure 3: результат программы readfile



# Исследование Sticky-бита

A screenshot of a terminal window titled 'guest2@nadezh:/tmp'. The window has a menu bar with 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows a series of commands and their outputs. First, a user 'guest' creates a file 'test' in '/tmp' and sets permissions 'o+rx'. Then, 'guest2' logs in as 'guest' and runs 'ls -l /tmp/file01.txt', showing permissions '-rw-rw-r-x' and owner 'guest'. 'guest2' then switches to 'root' using 'su' and runs 'rm file01.txt', which fails with the message 'rm: невозможно удалить «file01.txt»: Операция не позволена'. Finally, 'guest2' switches back to 'guest' and runs 'rm file01.txt' successfully. The terminal output is as follows:

```
guest2@nadezh:/tmp
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@nadezh lab5]$ echo "test" >> /tmp/file01.txt
[guest@nadezh lab5]$ chmod o+rx /tmp/file01.txt
[guest@nadezh lab5]$ ls -l /tmp/file01.txt
-rw-rw-r-x. 1 guest guest 5 окт  2 16:55 /tmp/file01.txt
[guest@nadezh lab5]$ su guest2
Пароль:
[guest2@nadezh lab5]$ cd /tmp
[guest2@nadezh tmp]$ cat file01.txt
test
[guest2@nadezh tmp]$ echo "test2" >> /tmp/file01.txt
[guest2@nadezh tmp]$ cat file01.txt
test
test2
[guest2@nadezh tmp]$ echo "test3" > /tmp/file01.txt
[guest2@nadezh tmp]$ rm file01.txt
rm: невозможно удалить «file01.txt»: Операция не позволена
[guest2@nadezh tmp]$ su
Пароль:
[root@nadezh tmp]#
[root@nadezh tmp]# chmod -t /tmp
[root@nadezh tmp]# exit
exit
[guest2@nadezh tmp]$ cd /tmp
[guest2@nadezh tmp]$ echo "test2" >> /tmp/file01.txt
[guest2@nadezh tmp]$ rm file01.txt
[guest2@nadezh tmp]$
```

Figure 4: исследование Sticky-бита

## **Выводы**

---

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.