

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

---

Факультет физико-математических и естественных наук

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

---

Тема: Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

дисциплина: Информационная безопасность

Студент: Койфман Кирилл Дмитриевич

Группа: НПИбд-01-21

### Введение.

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.

Задачи.

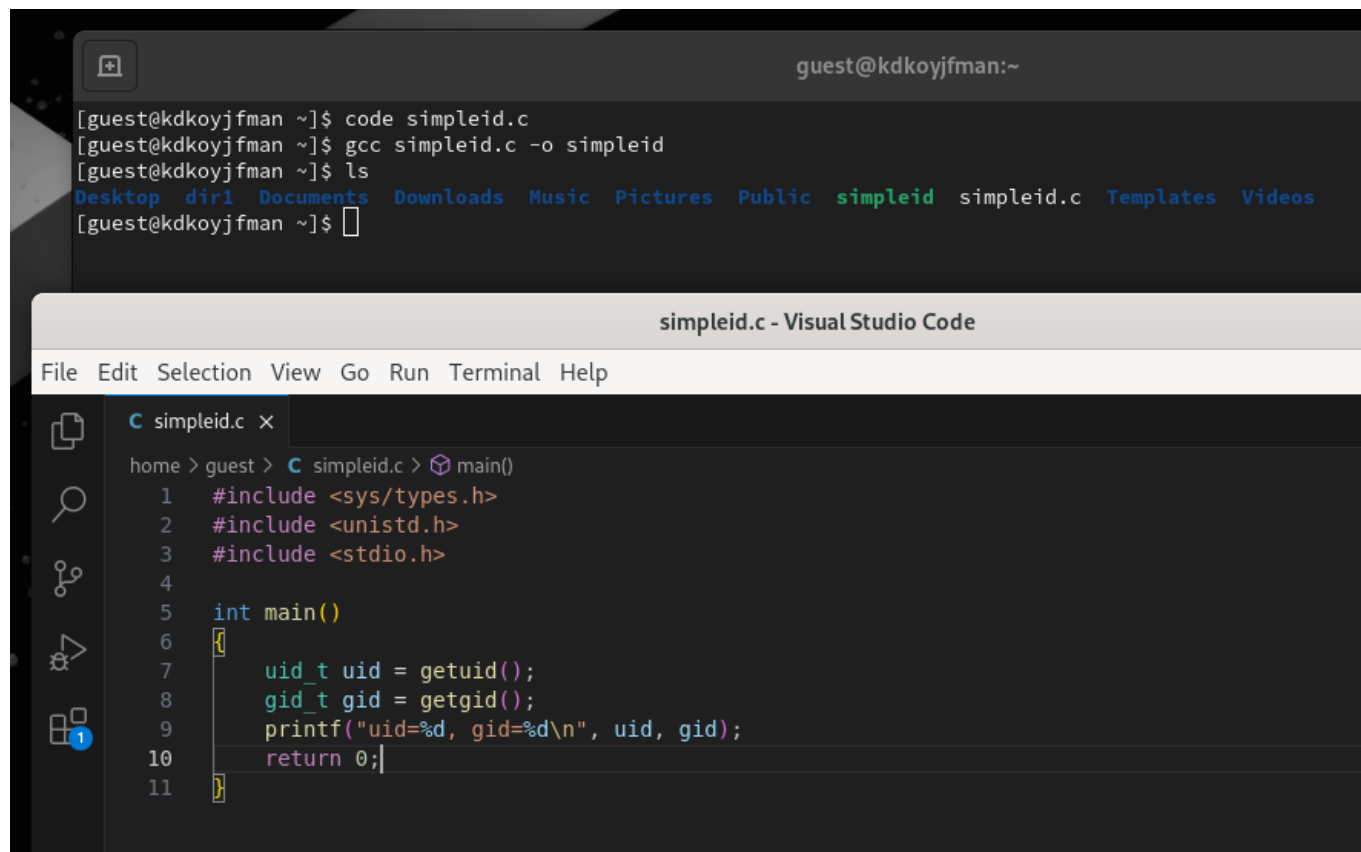
1. Создать несколько программ для проверки прав доступа к файлам с использованием SetUID-бита и SetGID-бита.

2. Провести последовательность тестов над файлами, использующих Sticky-бит.

## Ход работы

### 1 задание

Для начала от имени пользователя guest создадим программу simpleid.c, скомпилируем и выполним её, после чего сравним вывод программы с результатом команды `id`:



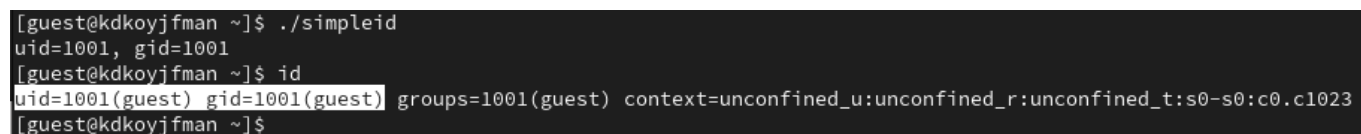
The image shows a terminal window and a Visual Studio Code editor. The terminal window, titled 'guest@kdkoyjman:~', shows the following commands and output:

```
[guest@kdkoyjman ~]$ code simpleid.c
[guest@kdkoyjman ~]$ gcc simpleid.c -o simpleid
[guest@kdkoyjman ~]$ ls
Desktop  dir1  Documents  Downloads  Music  Pictures  Public  simpleid  simpleid.c  Templates  Videos
[guest@kdkoyjman ~]$
```

The Visual Studio Code editor, titled 'simpleid.c - Visual Studio Code', shows the source code for 'simpleid.c':

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

РИС.1(программа успешно скомпилировалась и сохранилась)



The image shows a terminal window with the following commands and output:

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

РИС.2(вывод программы соответствует результату команды id)

Напишем другую программу, в которой будет производится вывод действительных идентификаторов, скомпилируем и запустим её (рис.3 - рис.4):

```
[guest@kdkoyjfm ~]$ cp simpleid.c simpleid2.c
[guest@kdkoyjfm ~]$ code simpleid2.c
[guest@kdkoyjfm ~]$ gcc simpleid2.c -o simpleid2
[guest@kdkoyjfm ~]$
```

simpleid2.c - Visual Studio Code

Edit Selection View Go Run Terminal Help

C simpleid.c C simpleid2.c X

home > guest > C simpleid2.c > main()

```

1  #include <sys/types.h>
2  #include <unistd.h>
3  #include <stdio.h>
4
5  int main()
6  {
7      uid_t real_uid = getuid();
8      uid_t e_uid = geteuid();
9
10     gid_t real_gid = getgid();
11     gid_t e_gid = getegid();
12
13     printf("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
14     printf("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
15     return 0;
16 }
```

РИС.3(программа успешно скомпилировалась)

```
[guest@kdkoyjfm ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@kdkoyjfm ~]$
```

РИС.4(вывод программы)

Далее от имени суперпользователя изменим владельца файла программы simpleid2 и установим SetUID-бит, потом выполним проверку правильности установки новых атрибутов и смены владельца файла, а также запустим simpleid2 и id (рис.5 - рис.8):

```
[root@kdkoyjfm /]# chown root:guest /home/guest/simpleid2
[root@kdkoyjfm /]# chmod u+s /home/guest/simpleid2
[root@kdkoyjfm /]#
```

РИС.5(изменение владельца файла программы simpleid2 и установка SetUID-бита)

```
[root@kdkoyjfm guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 17720 Oct  5 18:44 simpleid2
[root@kdkoyjfm guest]#
```

РИС.6(изменение владельца файла программы simpleid2 и установка SetUID-бита прошли успешно)

```
[root@kdkoyjfmman guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@kdkoyjfmman guest]#
```

РИС.7(результаты программы)

```
[root@kdkoyjfmman guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
```

РИС.8(результаты команды)

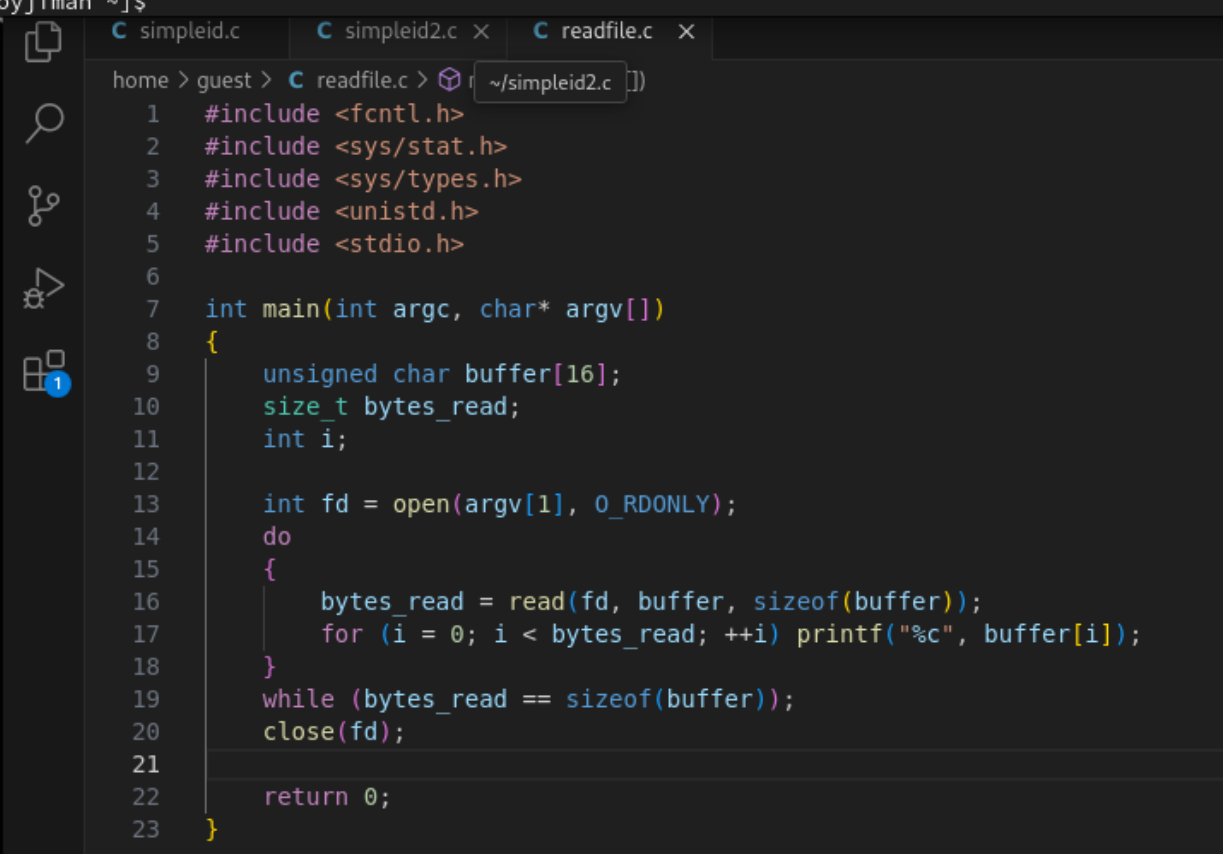
Прделаем ту же последовательность действий, но относительно SetGID-бита (рис.9):

```
[root@kdkoyjfmman /]# chmod g+s /home/guest/simpleid2
[root@kdkoyjfmman /]# ls -l /home/guest/simpleid2
-rwsr-sr-x. 1 root guest 17720 Oct  5 18:44 /home/guest/simpleid2
[root@kdkoyjfmman /]# /home/guest/.simpleid
bash: /home/guest/.simpleid: No such file or directory
[root@kdkoyjfmman /]# ./home/guest/simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@kdkoyjfmman /]#
```

РИС.9

А сейчас создадим программу readfile.c, скомпилируем её, сменим владельца файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис.10 - рис.11):

```
[guest@kdkoyjfmman ~]$ code readfile.c
[guest@kdkoyjfmman ~]$ gcc readfile.c -o readfile
[guest@kdkoyjfmman ~]$
```



```
1  #include <fcntl.h>
2  #include <sys/stat.h>
3  #include <sys/types.h>
4  #include <unistd.h>
5  #include <stdio.h>
6
7  int main(int argc, char* argv[])
8  {
9      unsigned char buffer[16];
10     size_t bytes_read;
11     int i;
12
13     int fd = open(argv[1], O_RDONLY);
14     do
15     {
16         bytes_read = read(fd, buffer, sizeof(buffer));
17         for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
18     }
19     while (bytes_read == sizeof(buffer));
20     close(fd);
21
22     return 0;
23 }
```

РИС.10(программа успешно скомпилировалась)

```
[root@kdkoyjfmam /]# chown root:guest /home/guest/readfile.c
[root@kdkoyjfmam /]# chmod 700 /home/guest/readfile.c
[root@kdkoyjfmam /]# chmod -r /home/guest/readfile.c
[root@kdkoyjfmam /]# chmod u+s /home/guest/readfile.c
[root@kdkoyjfmam /]# ls -l /home/guest/readfile.c
--ws-----. 1 root guest 464 Oct  5 19:05 /home/guest/readfile.c
[root@kdkoyjfmam /]# chmod u+r /home/guest/readfile.c
[root@kdkoyjfmam /]# ls -l /home/guest/readfile.c
-rws-----. 1 root guest 464 Oct  5 19:05 /home/guest/readfile.c
[root@kdkoyjfmam /]#
```

РИС.11(смена владельца и установка прав)

И попробуем считать файл от имени guest (рис.12):

```
[guest@kdkoyjfmam ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@kdkoyjfmam ~]$
```

РИС.12(в доступе отказано)

Сменим у программы readfile владельца и установим SetU'D-бит, после чего проверим может ли программа readfile прочитать файл readfile.c и файл /etc/shadow (рис.12 - рис.13):

```
[root@kdkoyjfmam /]# chmod 700 /home/guest/readfile
[root@kdkoyjfmam /]# ls -l /home/guest/readfile
-rwx-----. 1 root guest 17664 Oct  5 19:05 /home/guest/readfile
[root@kdkoyjfmam /]# chown root:guest /home/guest/readfile
[root@kdkoyjfmam /]# ls -l /home/guest/readfile
-rwx-----. 1 root guest 17664 Oct  5 19:05 /home/guest/readfile
[root@kdkoyjfmam /]# chmod u+s /home/guest/readfile
[root@kdkoyjfmam /]# ls -l /home/guest/readfile
-rws-----. 1 root guest 17664 Oct  5 19:05 /home/guest/readfile
[root@kdkoyjfmam /]#
```

РИС.12(смена владельца и установка прав)

```
[guest@kdkoyjfmam ~]$ ./readfile readfile.c
bash: ./readfile: Permission denied
[guest@kdkoyjfmam ~]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
[guest@kdkoyjfmam ~]$
```

РИС.13(файлы не считываются)

## 2 задание

---

Теперь проверим, установлен ли атрибут Sticky на директории /tmp, после чего от имени guest создадим файл file01.txt в этой директории со словом "test" и просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» (рис.14 - рис.15):

```
[guest@kdkoyjfm ~]$ ls -l / | grep tmp
drwxrwxrwt.  22 root root 4096 Oct  5 19:15 tmp
[guest@kdkoyjfm ~]$
```

РИС.14(атрибут Sticky на директории /tmp установлен)

```
[guest@kdkoyjfm ~]$ echo "test" > /tmp/file01.txt
[guest@kdkoyjfm ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  5 19:22 /tmp/file01.txt
[guest@kdkoyjfm ~]$ chmod o+rw /tmp/file01.txt
[guest@kdkoyjfm ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  5 19:22 /tmp/file01.txt
[guest@kdkoyjfm ~]$
```

РИС.15(атрибут Sticky на директории /tmp установлен)

А сейчас от имени пользователя guest2 попробуем прочитать файл file01.txt (рис.16):

```
[guest2@kdkoyjfm /]$ cat /tmp/file01.txt
test
[guest2@kdkoyjfm /]$
```

РИС.16

Попробуем дозаписать в этот файл (рис.17), стереть всё его содержимое (рис.18) и удалить его (рис.19):

```
[guest2@kdkoyjfm /]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@kdkoyjfm /]$ cat /tmp/file01.txt
test
[guest2@kdkoyjfm /]$
```

РИС.17(отредактировать файл нельзя)

```
[guest2@kdkoyjfm /]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@kdkoyjfm /]$ cat /tmp/file01.txt
test
[guest2@kdkoyjfm /]$
```

РИС.18(стереть всё его содержимое нельзя)

```
[guest2@kdkoyjfm /]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@kdkoyjfm /]$
```

РИС.19(удалить файл нельзя)

Повысим права до суперпользователя, после этого снимим атрибут t (Sticky-бит) с директории /tmp и покинем режим суперпользователя (рис.20):

```
[guest2@kdkoyjfmam /]$ su -  
Password:  
[root@kdkoyjfmam ~]# chmod -t /tmp  
[root@kdkoyjfmam ~]# exit  
logout  
[guest2@kdkoyjfmam /]$ █
```

Рис.20

Повторим предыдущие шаги (рис.21):

```
[guest2@kdkoyjfmam /]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 Oct  5 19:22 /tmp/file01.txt  
[guest2@kdkoyjfmam /]$ echo "test2" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@kdkoyjfmam /]$ cat /tmp/file01.txt  
test  
[guest2@kdkoyjfmam /]$ echo "test3" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@kdkoyjfmam /]$ cat /tmp/file01.txt  
test  
[guest2@kdkoyjfmam /]$ rm /tmp/file01.txt  
rm: remove write-protected regular file '/tmp/file01.txt'? y  
[guest2@kdkoyjfmam /]$ ls /tmp  
{35A10EB4-D72C-43C0-8138-5E260A8DEB5C}  
{70D626D9-2FA7-43B5-BA31-722AC30B3AF0}  
{7D5FC21F-1E7C-49C4-B730-53E8971FCF06}  
{AC828B4B-3188-4302-A41A-789A23D8EE44}  
{FCC10A19-621C-4047-8A40-E6DFEF579095}  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-chronyd.service-zF4CDQ  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-colord.service-6HWtQz  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-dbus-broker.service-KTRAS2  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-fwupd.service-pG876o  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-kdump.service-q54vPc  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-ModemManager.service-Cc0Fch  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-power-profiles-daemon.service-DUxwn8  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-rtkit-daemon.service-rgGiGu  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-switcheroo-control.service-8vIHGe  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-systemd-logind.service-ZL0py9  
systemd-private-dc338a9562ea4ec3bf45e0b1d2eda223-upower.service-jp0IyM  
vboxguest-Module.symvers  
[guest2@kdkoyjfmam /]$
```

Рис.21(удалось удалить файл)

Вновь повысим права до суперпользователя, после этого восстановим атрибут t (Sticky-бит) в директории /tmp (рис.22):

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

Рис.22

## Заключение

В ходе проделанной лабораторной работы мной были усвоены навыки работы в консоли с дополнительными атрибутами, а также изучены механизмы изменения идентификаторов с применением SetUID- и Sticky-битов.