

# **Отчёт по лабораторной работе №5**

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

Волков Тимофей Евгеньевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>13</b>

# List of Tables

# List of Figures

2.1	Программа simpleid.c . . . . .	6
2.2	Сравнение программы simpleid и команды id . . . . .	7
2.3	Усложнение программы . . . . .	7
2.4	Запуск программы simpleid2.c . . . . .	7
2.5	Установка SetUID-бита . . . . .	8
2.6	Установка SetGID-бита . . . . .	8
2.7	Программа readfile.c . . . . .	9
2.8	Проверка работы программы readfile . . . . .	10
2.9	Исследование Sticky-бита . . . . .	12

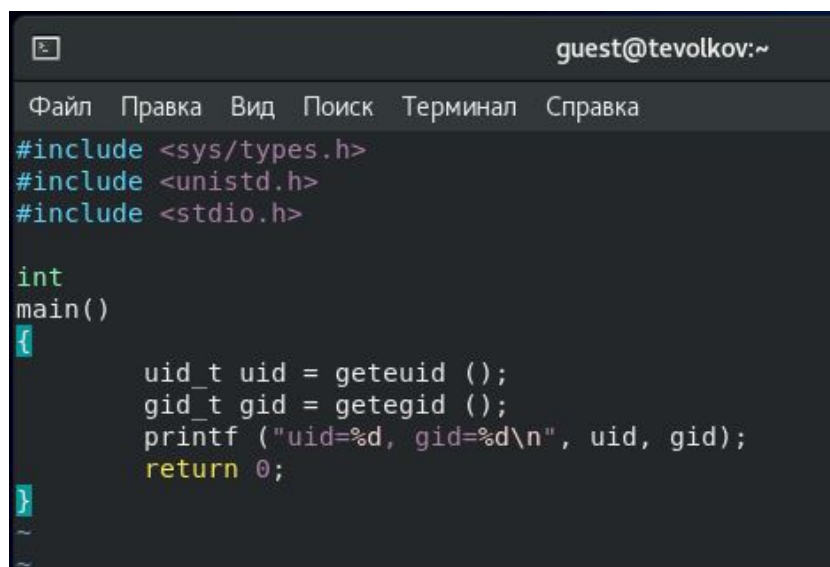
# 1 Цель работы

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.

## 2 Выполнение лабораторной работы

Войдите в систему от имени пользователя guest.

Создайте программу simpleid.c (fig. 2.1).

A screenshot of a terminal window with a dark background. The title bar at the top shows 'guest@tevolkov:~'. Below the title bar is a menu bar with options: 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The main area of the terminal displays the C code for 'simpleid.c'. The code includes three header files: <sys/types.h>, <unistd.h>, and <stdio.h>. It defines a 'main()' function that declares 'uid\_t uid' and 'gid\_t gid', calls 'geteuid()' and 'getegid()' to get the effective user and group IDs, prints them using 'printf' with the format 'uid=%d, gid=%d\n', and returns 0. The code is color-coded: keywords are green, comments are blue, and identifiers are purple. The cursor is at the end of the 'return 0;' line.

```
guest@tevolkov:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main()  
{  
    uid_t uid = geteuid ();  
    gid_t gid = getegid ();  
    printf ("uid=%d, gid=%d\n", uid, gid);  
    return 0;  
}
```

Figure 2.1: Программа simpleid.c

Скомпилируйте программу (fig. 2.2):

```
gcc simpleid.c -o simpleid
```

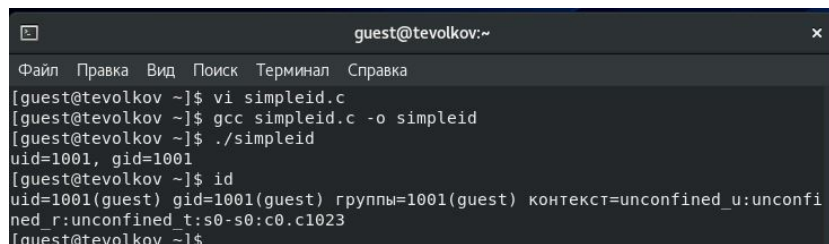
Выполните программу simpleid (fig. 2.2):

```
./simpleid
```

Выполните системную программу id (fig. 2.2):

```
id
```

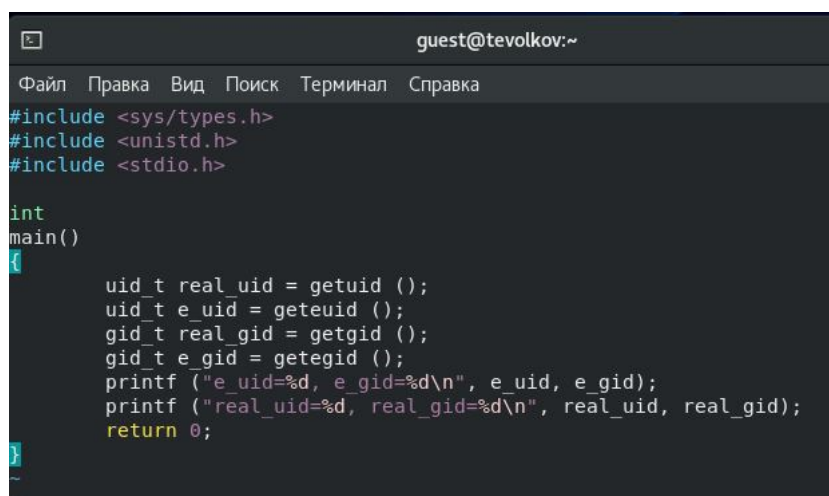
Программа и команда id выводят одинаковый uid и gid.



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

Figure 2.2: Сравнение программы simpleid и команды id

Усложните программу, добавив вывод действительных идентификаторов (fig. 2.3). Получившуюся программу назовите simpleid2.c.



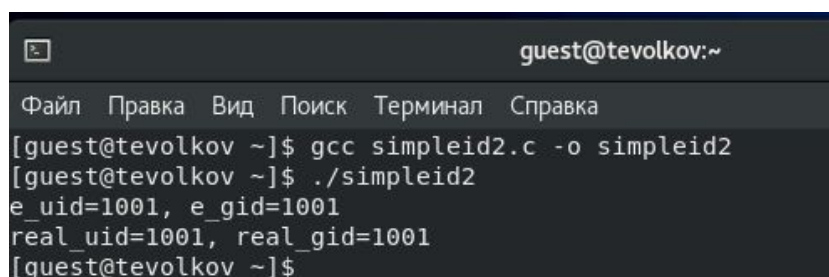
```
guest@tevolkov:~  
Файл Правка Вид Поиск Терминал Справка  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main()  
{  
    uid_t real_uid = getuid ();  
    uid_t e_uid = geteuid ();  
    gid_t real_gid = getgid ();  
    gid_t e_gid = getegid ();  
    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);  
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);  
    return 0;  
}
```

Figure 2.3: Усложнение программы

Скомпилируйте и запустите simpleid2.c (fig. 2.4):

```
gcc simpleid2.c -o simpleid2
```

```
./simpleid2
```



```
guest@tevolkov:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@tevolkov ~]$ gcc simpleid2.c -o simpleid2  
[guest@tevolkov ~]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@tevolkov ~]$
```

Figure 2.4: Запуск программы simpleid2.c

От имени суперпользователя выполните команды (fig. 2.5):

```
chown root:guest /home/guest/simpleid2
```

```
chmod u+s /home/guest/simpleid2
```

Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2(fig. 2.5):

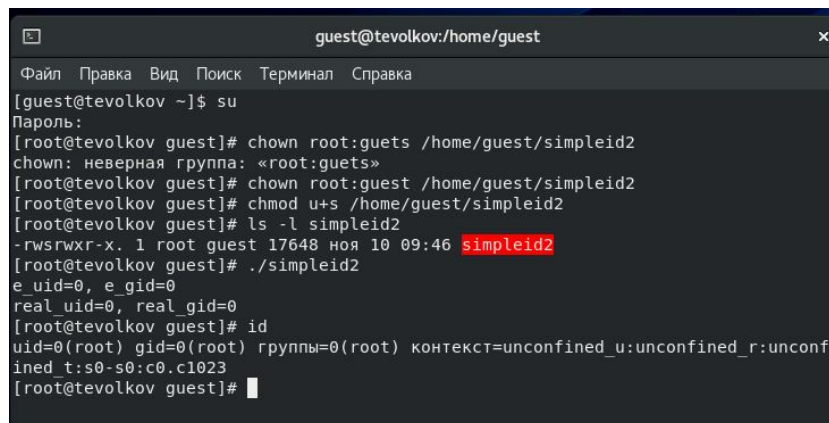
```
ls -l simpleid2
```

Запустите simpleid2 и id (fig. 2.5):

```
./simpleid2
```

```
id
```

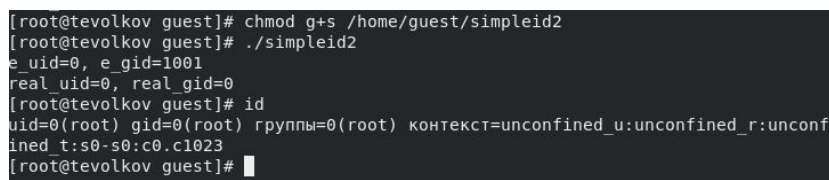
simpleid2 и id выводят одинаковые uid и gid, но отличающиеся от результатов предыдущих пунктов.



```
guest@tevolkov:/home/guest
Файл Правка Вид Поиск Терминал Справка
[guest@tevolkov ~]$ su
Пароль:
[root@tevolkov guest]# chown root:guets /home/guest/simpleid2
chown: неверная группа: «root:guets»
[root@tevolkov guest]# chown root:guest /home/guest/simpleid2
[root@tevolkov guest]# chmod u+s /home/guest/simpleid2
[root@tevolkov guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 10 09:46 simpleid2
[root@tevolkov guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@tevolkov guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@tevolkov guest]#
```

Figure 2.5: Установка SetUID-бита

Проделайте тоже самое относительно SetGID-бита (fig. 2.6).



```
[root@tevolkov guest]# chmod g+s /home/guest/simpleid2
[root@tevolkov guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@tevolkov guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@tevolkov guest]#
```

Figure 2.6: Установка SetGID-бита

Создайте программу readfile.c (fig. 2.7).



```
guest@tevolkov:~
Файл Правка Вид Поиск Терминал Справка
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <unistd.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 == sizeof (buffer));
    close (fd);
    return 0;
}
```

Figure 2.7: Программа readfile.c

Смените владельца у файла readfile.c и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (fig. 2.8).

Проверьте, что пользователь guest не может прочитать файл readfile.c (fig. 2.8).

Смените у программы readfile владельца и установите SetUID-бит (fig. 2.8).

Программа readfile может прочитать файл readfile.c и файл /etc/shadow.

```
guest@tevolkov:~  
[guest@tevolkov ~]$ su  
Пароль:  
[root@tevolkov guest]# chown root:guest /home/guest/readfile.c  
[root@tevolkov guest]# chmod 700 /home/guest/readfile.c  
[root@tevolkov guest]# ls -l /home/guest/readfile.c  
-rwx----- 1 root guest 397 ноя 10 10:15 /home/guest/readfile.c  
[root@tevolkov guest]# su guest  
[guest@tevolkov ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@tevolkov ~]$ su  
Пароль:  
[root@tevolkov guest]# chown root:guest /home/guest/readfile  
[root@tevolkov guest]# chmod u+s /home/guest/readfile  
[root@tevolkov guest]# su guest  
[guest@tevolkov ~]$ ./readfile readfile.c  
#include <fcntl.h>  
#include <stdio.h>  
#include <sys/stat.h>  
#include <unistd.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 == sizeof (buffer));  
    close (fd);  
    return 0;  
}  
[guest@tevolkov ~]$ ./readfile /etc/shadow  
root:$6$8FxbNnpnC03A1Wnp$okZWITwaESNkl3486nDfXjFo7x/hdD6job.b9TsI7EfW/j8ieVvV0IXA8DtRdugfGoV9cT3tGaaggJ7ck05Tvc  
l:0:99999:7:::  
bin:18397:0:99999:7:::  
daemon:18397:0:99999:7:::  
adm:18397:0:99999:7:::  
lp:18397:0:99999:7:::  
sync:18397:0:99999:7:::
```

Figure 2.8: Проверка работы программы readfile

Выясните, установлен ли атрибут Sticky на директории /tmp (fig. 2.9), для чего выполните команду

```
ls -l / | grep tmp
```

От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test (fig. 2.9):

```
echo "test" > /tmp/file01.txt
```

Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные» (fig. 2.9):

```
ls -l /tmp/file01.txt
```

```
chmod o+rw /tmp/file01.txt
```

```
ls -l /tmp/file01.txt
```

От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt (fig. 2.9):

```
cat /tmp/file01.txt
```

От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 (fig. 2.9) командой

```
echo "test2" > /tmp/file01.txt
```

Дозапись прошла успешно.

Проверьте содержимое файла командой

```
cat /tmp/file01.txt
```

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

```
echo "test3" > /tmp/file01.txt
```

Запись прошла успешно.

Проверьте содержимое файла (fig. 2.9) командой

```
cat /tmp/file01.txt
```

От пользователя guest2 попробуйте удалить файл /tmp/file01.txt (fig. 2.9) командой

```
rm /tmp/file01.txt
```

Удалить файл не удалось.

Повысьте свои права до суперпользователя (fig. 2.9) следующей командой  
su -

и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp (fig. 2.9):

```
chmod -t /tmp
```

Покиньте режим суперпользователя (fig. 2.9) командой  
exit

От пользователя guest2 проверьте, что атрибута t у директории /tmp нет (fig. 2.9) :

```
ls -l / | grep tmp
```

Повторите предыдущие шаги (fig. 2.9).

Возможно выполнить все команды из предыдущих шагов.

Повысьте свои права до суперпользователя и верните атрибут t на директорию

/tmp (fig. 2.9):

su -

chmod +t /tmp

exit

```
guest2@tevolkov:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@tevolkov ~]$ ls -l |grep tmp  
[guest@tevolkov ~]$ ls -l | grep tmp  
[guest@tevolkov ~]$ ls -l / | grep tmp  
drwxrwxrwt. 14 root root 4096 ноя 10 10:20 tmp  
[guest@tevolkov ~]$ echo "test" > /tmp/file01.txt  
[guest@tevolkov ~]$ ls -l /tmp/file01.txt  
-rw-rw-r--. 1 guest guest 5 ноя 10 10:29 /tmp/file01.txt  
[guest@tevolkov ~]$ chmod o+rw /tmp/file01.txt  
[guest@tevolkov ~]$ ls -l /tmp/file01.txt  
-rw-rw-rw-. 1 guest guest 5 ноя 10 10:29 /tmp/file01.txt  
[guest@tevolkov ~]$ su guest2  
Пароль:  
[guest2@tevolkov guest]$ exit  
exit  
[guest@tevolkov ~]$ su - guest2  
Пароль:  
[guest2@tevolkov ~]$ cat /tmp/file01.txt  
test  
[guest2@tevolkov ~]$ echo "test2" > /tmp/file01.txt  
[guest2@tevolkov ~]$ cat /tmp/file01.txt  
test2  
[guest2@tevolkov ~]$ echo "test2" /tmp/file01.txt  
test2 /tmp/file01.txt  
[guest2@tevolkov ~]$ cat /tmp/file01.txt  
test2  
[guest2@tevolkov ~]$ echo "test3" > /tmp/file01.txt  
[guest2@tevolkov ~]$ cat /tmp/file01.txt  
test3  
[guest2@tevolkov ~]$ rm /tmp/file01.txt  
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена  
[guest2@tevolkov ~]$ su -  
Пароль:  
[root@tevolkov ~]# chmod -t /tmp  
[root@tevolkov ~]# exit  
выход  
[guest2@tevolkov ~]$ ls -l / | grep tmp  
drwxrwxrwx. 14 root root 4096 ноя 10 10:35 tmp  
[guest2@tevolkov ~]$ echo "test2" > /tmp/file01.txt  
[guest2@tevolkov ~]$ cat /tmp/file01.txt  
test2  
[guest2@tevolkov ~]$ rm /tmp/file01.txt  
[guest2@tevolkov ~]$ su -  
Пароль:  
[root@tevolkov ~]# chmod +t /tmp  
[root@tevolkov ~]# exit  
выход  
[guest2@tevolkov ~]$  
[guest2@tevolkov ~]$
```

Figure 2.9: Исследование Sticky-бита

## 3 Выводы

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.