

# Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

---

Садова Д. А.

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Садова Диана Алексеевна
- студент бакалавриата
- Российский университет дружбы народов
- [113229118@pfur.ru]
- <https://DianaSadova.github.io/ru/>

# Вводная часть

---

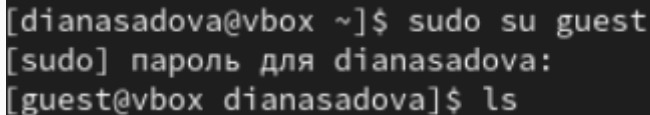
- Нам важно понимать как изменяются индикаторы, и SetUID- , Sticky-битов. Так же, нужно отрабатывать навыки работы в консоли с дополнительными атрибутами

- 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

- Текст лабораторной работы № 5

## Создание программы

1. Войдите в систему от имени пользователя guest.

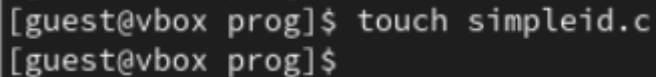


```
[dianasadova@vbox ~]$ sudo su guest  
[sudo] пароль для dianasadova:  
[guest@vbox dianasadova]$ ls
```

**Рис. 1:** Заходим в систему пользователя guest



2. Создайте программу simpleid.c:

A terminal window with a dark background and light gray text. The prompt is [guest@vbox prog]\$ and the command touch simpleid.c has been entered. The prompt is shown again on the next line.

```
[guest@vbox prog]$ touch simpleid.c  
[guest@vbox prog]$
```

**Рис. 2:** Создаем файл с именем simpleid

```
#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;
}
```

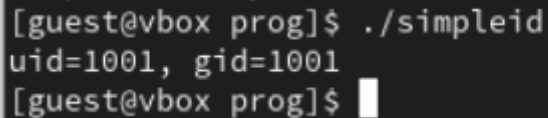
Рис. 3: Код

3. Скомпилируйте программу и убедитесь, что файл программы создан:

```
[guest@vbox prog]$ gcc simpleid.c -o simpleid  
[guest@vbox prog]$ ls  
simpleid  simpleid.c  
[guest@vbox prog]$
```

**Рис. 4:** Скомпилируем программу

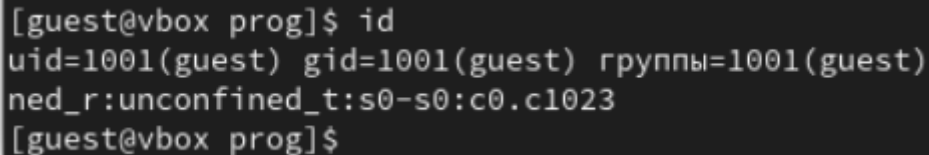
4. Выполните программу simpleid:

A terminal window with a dark background and light gray text. The prompt is [guest@vbox prog]\$. The first command is ./simpleid, which outputs uid=1001, gid=1001. The second command is a blank line, indicated by a white cursor block.

```
[guest@vbox prog]$ ./simpleid
uid=1001, gid=1001
[guest@vbox prog]$
```

**Рис. 5:** Выполняем код

5. Выполните системную программу id:

A terminal window with a dark background. The prompt is [guest@vbox prog]\$ and the command id has been entered. The output shows the user's identity and group information in both English and Russian. The prompt is repeated at the bottom.

```
[guest@vbox prog]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest)
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@vbox prog]$
```

**Рис. 6:** Выполняем системную программу id

и сравните полученный вами результат с данными предыдущего пункта задания.

Записи идентичны.

6. Усложните программу, добавив вывод действительных идентификаторов:

```
GNU nano 5.6.1 simpleid.c
#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;
}
```

7. Скомпилируйте и запустите simpleid2.c:

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

**Рис. 8:** Скомпилируем программу



8. От имени суперпользователя выполните команды:

```
[dianasadova@vbox ~]$ sudo -i  
[sudo] пароль для dianasadova:  
[root@vbox ~]# chown root:guest /home/guest/prog/simpleid2  
[root@vbox ~]# chmod u+s /home/guest/prog/simpleid2  
[root@vbox ~]#
```

**Рис. 9:** Выполняем команды

9. Используйте `sudo` или повысьте временно свои права с помощью `su`.

```
[guest@vbox prog]$ sudo ./simpleid2
[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
[guest@vbox prog]$ sudo id
[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
```

**Рис. 10:** Переходим в режим суперпользователя

Поясните, что делают эти команды.

Мы пытаемся запустить код в файле `simpleid2.c` от имени суперпользователя.

10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
[guest@vbox prog]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 17656 map 14 12:25 simpleid2
[guest@vbox prog]$
```

**Рис. 11:** Проверяем правильности новых атрибутов

11. Запустите simpleid2 и id:

```
[guest@vbox prog]$ ./simpleid2  
id  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@vbox prog]$
```

Рис. 12: Запускаем simpleid2 и id

Сравните результаты.

simpleid2 - выдает только информацию о id (номер).

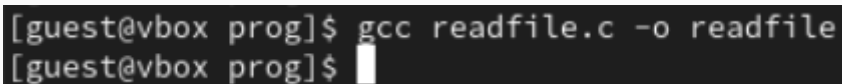
id - дает больше информации об пользователе и его группах.

12. Прodelайте тоже самое относительно SetGID-бита.

13. Создайте программу readfile.c:

```
GNU nano 5.6.1                                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[])
{
    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);
}
```

14. Откомпилируйте её.

A screenshot of a terminal window with a dark background. It shows two lines of text: the first line is a command to compile a C file named 'readfile.c' into an executable named 'readfile' using the 'gcc' compiler; the second line shows the prompt after the command has been executed.

```
[guest@vbox prog]$ gcc readfile.c -o readfile  
[guest@vbox prog]$
```

**Рис. 14:** Скомпилируем программу



15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```
[guest@vbox prog]$ su dianasadova
Пароль:
[dianasadova@vbox prog]$ sudo chown root:guest /home/guest/simpleid2
[sudo] пароль для dianasadova:
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла ил
и каталога
[dianasadova@vbox prog]$ sudo chown root:guest /home/guest/prog/simpleid2
[dianasadova@vbox prog]$ sudo chown root:guest /home/guest/prog/readfile
[dianasadova@vbox prog]$ sudo chmod u+s /home/guest/prog/readfile
[dianasadova@vbox prog]$
```

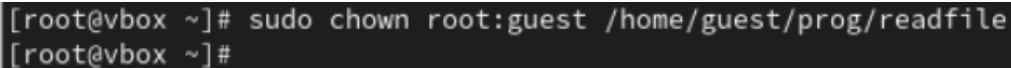
**Рис. 15:** Меняем прова доступа

16. Проверьте, что пользователь guest не может прочитать файл readfile.c.

```
[dianasadova@vbox prog]$ sudo chmod ga-rwx /home/guest/prog/readfile  
[dianasadova@vbox prog]$
```

**Рис. 16:** Проверяем, что guest не может прочитать файл

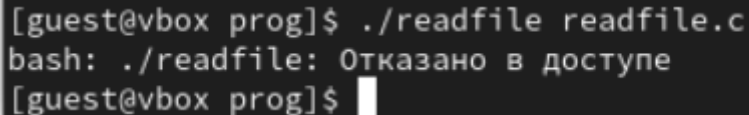
17. Смените у программы readfile владельца и установите SetU'D-бит..

A terminal window with a dark background and light gray text. The prompt is [root@vbox ~]#. The command entered is sudo chown root:guest /home/guest/prog/readfile. The prompt is repeated on the next line.

```
[root@vbox ~]# sudo chown root:guest /home/guest/prog/readfile  
[root@vbox ~]#
```

**Рис. 17:** Снимаем у readfile владельца

18. Проверьте, может ли программа readfile прочитать файл readfile.c?

A terminal window with a black background and white text. The prompt is [guest@vbox prog]\$ and the command entered is ./readfile readfile.c. The output is bash: ./readfile: Отказано в доступе. The prompt is repeated at the end of the line.

```
[guest@vbox prog]$ ./readfile readfile.c
bash: ./readfile: Отказано в доступе
[guest@vbox prog]$
```

**Рис. 18:** Проверяем может ли readfile прочитать файл readfile.c

19. Проверьте, может ли программа readfile прочитать файл /etc/shadow?

```
[guest@vbox prog]$ ./readfile /etc/shadow  
bash: ./readfile: Отказано в доступе  
[guest@vbox prog]$
```

**Рис. 19:** Проверяем может ли readfile прочитать файл /etc/shadow

## Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду

```
[guest@vbox prog]$ ls -l / | grep tmp  
drwxrwxrwt. 17 root root 4096 map 14 12:42 tmp  
[guest@vbox prog]$
```

**Рис. 20:** Определяем установлен ли атрибут Sticky

2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test:

```
[guest@vbox prog]$ echo "test" > /tmp/file01.txt  
[guest@vbox prog]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 map 14 12:45 /tmp/file01.txt  
[guest@vbox prog]$
```

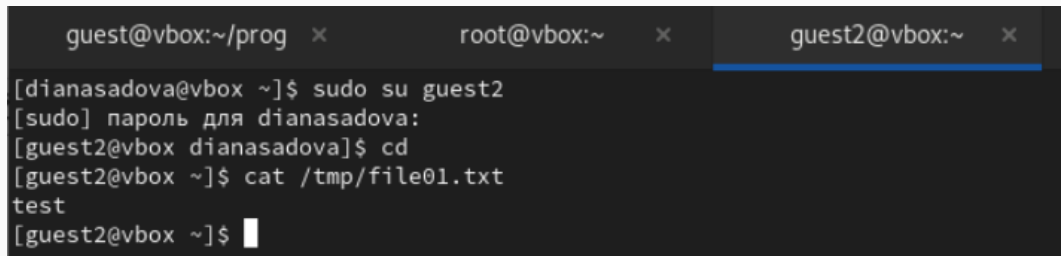
**Рис. 21:** Создаем файл file01.txt

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

```
[guest@vbox prog]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 map 14 12:45 /tmp/file01.txt
[guest@vbox prog]$ chmod o+rw /tmp/file01.txt
[guest@vbox prog]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 map 14 12:45 /tmp/file01.txt
[guest@vbox prog]$
```

Рис. 22: Просматриваем все атрибуты

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

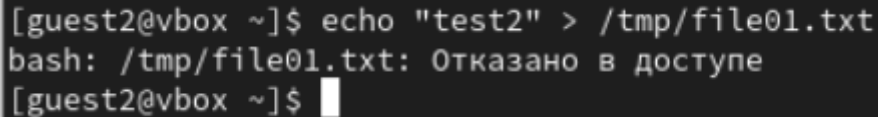
A terminal window with three tabs: 'guest@vbox:~/prog', 'root@vbox:~', and 'guest2@vbox:~'. The 'guest2@vbox:~' tab is active. The terminal shows the following commands and output:

```
[dianasadova@vbox ~]$ sudo su guest2
[sudo] пароль для dianasadova:
[guest2@vbox dianasadova]$ cd
[guest2@vbox ~]$ cat /tmp/file01.txt
test
[guest2@vbox ~]$
```

Рис. 23: Пробуем читать файл от guest2



5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой

A terminal window with a dark background and light gray text. The prompt is [guest2@vbox ~]\$. The user enters the command echo "test2" > /tmp/file01.txt. The terminal outputs bash: /tmp/file01.txt: Отказано в доступе. The prompt returns to [guest2@vbox ~]\$.

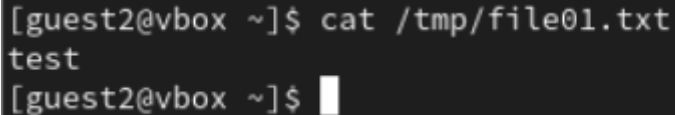
```
[guest2@vbox ~]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vbox ~]$
```

**Рис. 24:** Пробуем что-то дописать в файл

Удалось ли вам выполнить операцию?

Нет. Отказано в доступе.

6. Проверьте содержимое файла командой

A terminal window with a black background and white text. The prompt is [guest2@vbox ~]\$. The command cat /tmp/file01.txt is entered. The output test is displayed on the next line. The prompt [guest2@vbox ~]\$ is shown again with a white cursor block.

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

**Рис. 25:** Проверяем содержимое файла

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

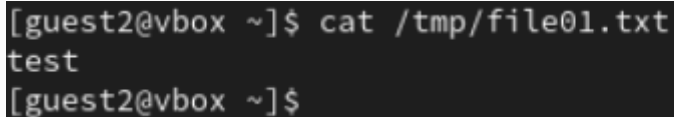
```
[guest2@vbox ~]$ echo "test3" > /tmp/file01.txt  
bash: /tmp/file01.txt: Отказано в доступе  
[guest2@vbox ~]$
```

**Рис. 26:** Пробуем что-то дописать в файл

Удалось ли вам выполнить операцию?

Нет. Отказано в доступе.

8. Проверьте содержимое файла командой



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

**Рис. 27:** Проверяем содержимое файла

9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой

```
[guest2@vbox ~]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@vbox ~]$
```

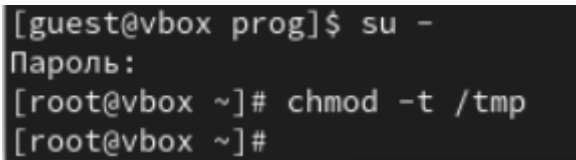
**Рис. 28:** Пробуем удалить файл

Удалось ли вам удалить файл?

Нет. Отказано в доступе.



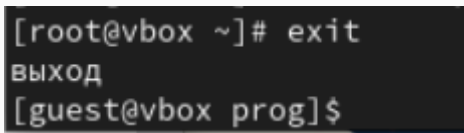
10. Повысьте свои права до суперпользователя следующей командой `su -` и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`:

A terminal window with a dark background and light gray text. The first line shows the prompt '[guest@vbox prog]\$' followed by the command 'su -'. The second line shows the prompt 'Пароль:' (Password:). The third line shows the prompt '[root@vbox ~]#' followed by the command 'chmod -t /tmp'. The fourth line shows the prompt '[root@vbox ~]#'.

```
[guest@vbox prog]$ su -  
Пароль:  
[root@vbox ~]# chmod -t /tmp  
[root@vbox ~]#
```

**Рис. 29:** Переходим в режим суперпользователя и снимаем атрибут

11. Покиньте режим суперпользователя командой

A terminal window with a dark background. The first line shows the root prompt '[root@vbox ~]#', followed by the command 'exit'. The second line shows the output 'выход' (exit in Russian). The third line shows the guest user prompt '[guest@vbox prog]\$'.

**Рис. 30:** Выходим из режима суперпользователя

12. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет:

```
[guest2@vbox ~]$ ls -l / | grep tmp  
drwxrwxrwx. 17 root root 4096 мар 14 12:50 tmp  
[guest2@vbox ~]$
```

**Рис. 31:** Проверяем наличие атрибута t

13. Повторите предыдущие шаги. Какие наблюдаются изменения?

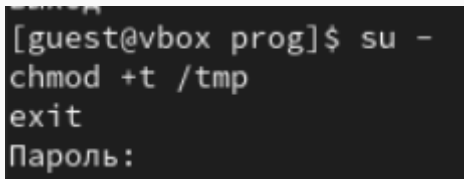
```
[guest2@vbox ~]$ cat /tmp/file01.txt
test
[guest2@vbox ~]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vbox ~]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vbox ~]$ cat /tmp/file01.txt
test
[guest2@vbox ~]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
[guest2@vbox ~]$
```

Рис. 32: Повторяем предыдущие шаги

14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.

Да, мы смогли удалить файл

15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`:

A terminal window with a black background and white text. The text shows a user at a prompt switching to root and restoring permissions to /tmp.

```
[guest@vbox prog]$ su -  
chmod +t /tmp  
exit  
Пароль:
```

**Рис. 33:** Возвращаем предыдущие шаги

- Изучили механизм изменения идентификаторов, применения SetUID- и Sticky-битов.
- Получили практические навыки работы в консоли с дополнительными атрибутами.
- Рассмотрели работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов