

Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов

Хамдамова Айжана

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	18

Список иллюстраций

Список таблиц

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 Теоретическое введение

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута. [1]

Sticky bit Используется в основном для каталогов, чтобы защитить в них файлы. В такой каталог может писать любой пользователь. Но, из такой директории пользователь может удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

SUID (Set User ID) Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ к важным системным файлам, которые принадлежат пользователю root.

SGID (Set Group ID) Аналогичен suid, но относится к группе. Если установить sgid для каталога, то все файлы созданные в нем, при запуске будут принимать идентификатор группы каталога, а не группы владельца, который создал файл в этом каталоге.

Обозначение атрибутов sticky, suid, sgid Специальные права используются довольно редко, поэтому при выводе программы ls -l символ, обозначающий

указанные атрибуты, закрывает символ стандартных прав доступа.

Пример:

rwsrwsrwt

где первая s — это suid, вторая s — это sgid, а последняя t — это sticky bit

В приведенном примере не понятно, rwt — это rw- или rwx? Определить это просто. Если t маленькое, значит x установлен. Если T большое, значит x не установлен. То же самое правило распространяется и на s.

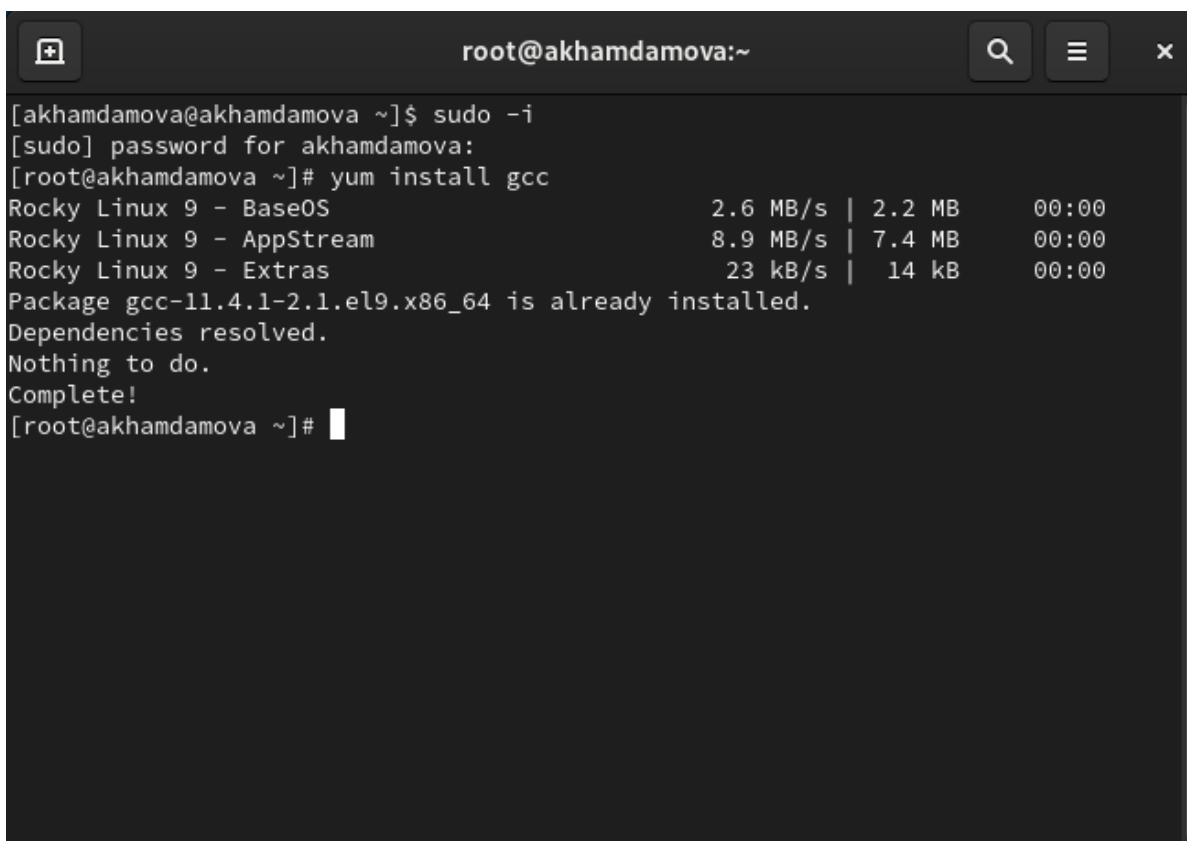
В числовом эквиваленте данные атрибуты определяются первым символом при четырехзначном обозначении (который часто опускается при назначении прав), например в правах 1777 — символ 1 обозначает sticky bit. Остальные атрибуты имеют следующие числовое соответствие:

1 — установлен sticky bit 2 — установлен sgid 4 — установлен suid

Компилятор GCC

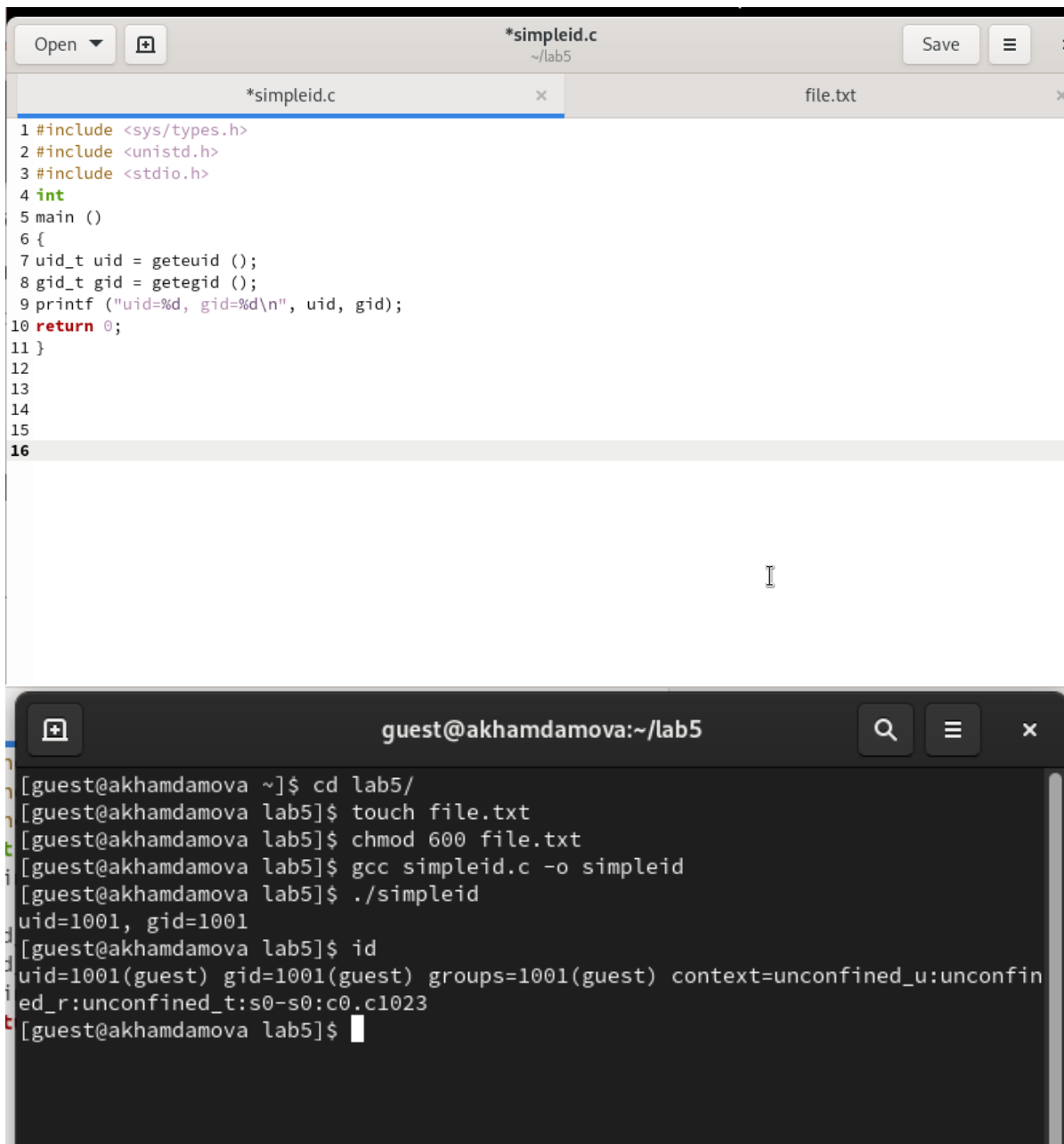
GCC - это свободно доступный оптимизирующий компилятор для языков C, C++. Собственно программа gcc это некоторая надстройка над группой компиляторов, которая способна анализировать имена файлов, передаваемые ей в качестве аргументов, и определять, какие действия необходимо выполнить. Файлы с расширением .cc или .C рассматриваются, как файлы на языке C++, файлы с расширением .c как программы на языке C, а файлы с расширением .o считаются объектными

3 Выполнение лабораторной работы



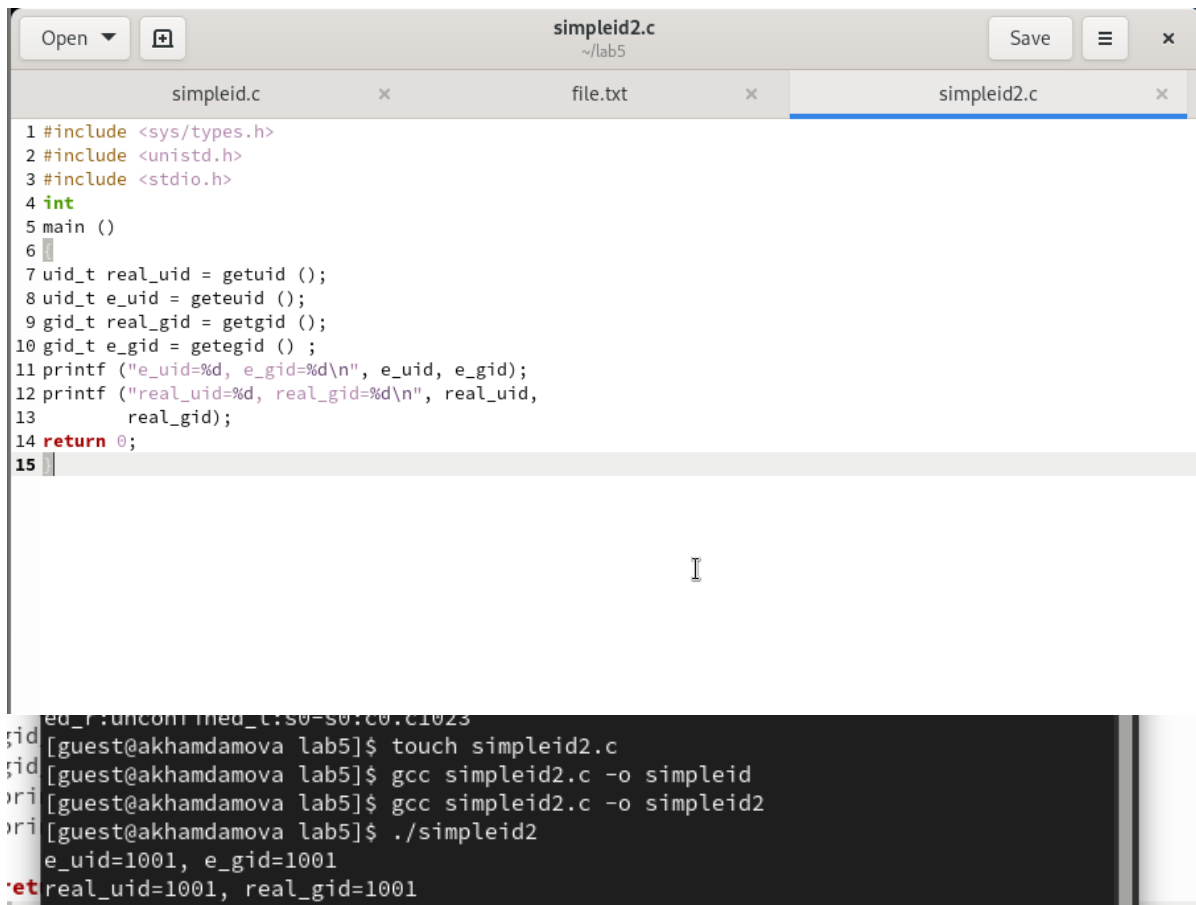
```
root@akhamdamova:~  
[akhamdamova@akhamdamova ~]$ sudo -i  
[sudo] password for akhamdamova:  
[root@akhamdamova ~]# yum install gcc  
Rocky Linux 9 - BaseOS                2.6 MB/s | 2.2 MB    00:00  
Rocky Linux 9 - AppStream             8.9 MB/s | 7.4 MB    00:00  
Rocky Linux 9 - Extras                 23 kB/s | 14 kB     00:00  
Package gcc-11.4.1-2.1.el9.x86_64 is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[root@akhamdamova ~]#
```

Создание программы 1. Войдите в систему от имени пользователя guest. 2. Создайте программу simpleid.c: 3. Скомпилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid 4. Выполните программу simpleid: ./simpleid 5. Выполните системную программу id:



6. Усложните программу, добавив вывод действительных идентификаторов
7. Скомпилируйте и запустите `simpleid2.c`: `gcc simpleid2.c -o simpleid2`
`./simpleid2`
8. От имени суперпользователя выполните команды: Информационная
безопасность компьютерных сетей 37 `chown root:guest /home/guest/simpleid2`
`chmod u+s /home/guest/simpleid2`

9. Используйте `sudo` или повысьте временно свои права с помощью `su`.
Поясните, что делают эти команды.
10. Выполните проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`: `ls -l simpleid2`
11. Запустите `simpleid2` и `id`: Сравните результаты.
12. Прodelайте тоже самое относительно SetGID-бита.



The screenshot shows a code editor window titled "simpleid2.c" with the following C code:

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

Below the code editor, a terminal window shows the following commands and output:

```
[guest@akhamdamova lab5]$ touch simpleid2.c
[guest@akhamdamova lab5]$ gcc simpleid2.c -o simpleid
[guest@akhamdamova lab5]$ gcc simpleid2.c -o simpleid2
[guest@akhamdamova lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

```
root@akhamdamova:~  
[root@akhamdamova ~]# ls -l /home/guest/lab5/simpleid2  
-rwsr-xr-x. 1 root guest 26064 Apr 12 10:07 /home/guest/lab5/simpleid2  
[root@akhamdamova ~]# cd /home/guest/lab5/simpleid2  
-bash: cd: /home/guest/lab5/simpleid2: Not a directory  
[root@akhamdamova ~]# /home/guest/lab5/simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@akhamdamova ~]# /home/guest/lab5/./simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@akhamdamova ~]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi  
ned_t:s0-s0:c0.c1023  
[root@akhamdamova ~]# chown root:guest /home/guest/lab5/simpleid2  
[root@akhamdamova ~]# chmod g+s /home/guest/lab5/simpleid2  
[root@akhamdamova ~]# ls -l /home/guest/lab5/simpleid2  
-rwxr-sr-x. 1 root guest 26064 Apr 12 10:07 /home/guest/lab5/simpleid2  
[root@akhamdamova ~]# /home/guest/lab5/./simpleid2  
e_uid=0, e_gid=1001  
real_uid=0, real_gid=0  
[root@akhamdamova ~]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi  
ned_t:s0-s0:c0.c1023  
[root@akhamdamova ~]#
```

13. Создайте программу readfile.c

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

14. Откомпилируйте её. `gcc readfile.c -o readfile`
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.
16. Проверьте, что пользователь `guest` не может прочитать файл `readfile.c`.
17. Смените у программы `readfile` владельца и установите SetU'D-бит.
18. Проверьте, может ли программа `readfile` прочитать файл `readfile.c`?
19. Проверьте, может ли программа `readfile` прочитать файл `/etc/shadow`? суперпользователь может

```
logout
[akhamdamova@akhamdamova lab5]$ touch readfile.c
touch: cannot touch 'readfile.c': Permission denied
[akhamdamova@akhamdamova lab5]$ su guest
Password:
[guest@akhamdamova lab5]$ touch readfile.c
[guest@akhamdamova lab5]$ gcc readfile.c -o readfile
```

```
su: Authentication failure
[guest@akhamdamova lab5]$ su
Password:
[root@akhamdamova lab5]# chown root:guest readfile
[root@akhamdamova lab5]# chmod 700 readfile
[root@akhamdamova lab5]# chown root:guest readfile
[root@akhamdamova lab5]# chmod -r readfile.c
[root@akhamdamova lab5]# chmod u+c readfile
chmod: invalid mode: 'u+c'
Try 'chmod --help' for more information.
[root@akhamdamova lab5]# chmod u+s readfile
[root@akhamdamova lab5]#

[root@akhamdamova lab5]# exit
exit
[guest@akhamdamova lab5]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@akhamdamova lab5]$ ./readfile readfile.c
bash: ./readfile: Permission denied
[guest@akhamdamova lab5]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
[guest@akhamdamova lab5]$
```

```

[guest@akhamdamova lab5]$ su
Password:
[root@akhamdamova lab5]# cat 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);
    return 0;
}
[root@akhamdamova lab5]# ./readfile 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;

```

```

[guest@akhamdamova /home/guest/lab5]
[root@akhamdamova lab5]# ./readfile /etc/shadow
root:$6$yH3ysfK9phc/5wvy$edd7FZBybWsjZ8fArD7X3FTnfX0.7jMoHzXdUtAvL.lcI0oYx0Hnkk4jVCz9.mpec5oZj.lXm7DH/cfX.vru1::0:99999:7:::
bin:!:19469:0:99999:7:::
daemon:!:19469:0:99999:7:::
adm:!:19469:0:99999:7:::
lp:!:19469:0:99999:7:::
sync:!:19469:0:99999:7:::
shutdown:!:19469:0:99999:7:::
halt:!:19469:0:99999:7:::
mail:!:19469:0:99999:7:::
operator:!:19469:0:99999:7:::
games:!:19469:0:99999:7:::
ftp:!:19469:0:99999:7:::
nobody:!:19469:0:99999:7:::
systemd-coredump:!:19769:!!!!:
dbus:!:19769:!!!!:
polkitd:!:19769:!!!!:
avahi:!:19769:!!!!:
rtkit:!:19769:!!!!:
pipewire:!:19769:!!!!:
sssd:!:19769:!!!!:
libstoragemgmt:!:19769:!!!!:
systemd-oom:!:19769:!!!!:
tss:!:19769:!!!!:
geoclue:!:19769:!!!!:
cockpit-ws:!:19769:!!!!:
cockpit-wsinstance:!:19769:!!!!:
flatpak:!:19769:!!!!:
colord:!:19769:!!!!:
clevi:!:19769:!!!!:
setroubleshoot:!:19769:!!!!:
gdm:!:19769:!!!!:
design:!:19769:!!!!:
gnome-initial-setup:!:19769:!!!!:
sshd:!:19769:!!!!:
chrony:!:19769:!!!!:
dnsmasq:!:19769:!!!!:
tcpdump:!:19769:!!!!:
akhamdamova:$6$dXz1PCLnWuUWnqJf$KtkqazhiREXaMEEj/rzmYIPWn/cE7/qWmnIHf33CVBcyTj/orL78GHyUF3MERMvt1kZMOB9ss7k8eDXfX60::0:99999:7:::
guest:$6$QXL7ANcrMRAPm2w$cxcp3UuI80nb9prRtn2XrggymhcmD4aGxbIbagCepoXI3refZAH7dzeu0BtpvgQAChWmMsR0jH50ASALrnQIO.:19797:0:99999:7:::
guest2:$6$0tnQzq3wAw7QIX3r$CRL9lPPeF6n3voG5q50zGkEf5hLqjmElGgX6cVNF/10lI6GdEkaMiHlBPBVAYiAwUUYrYVnwjrpZ1F/LT90:19797:0:99999:7:::
[root@akhamdamova lab5]#

```

Исследование Sticky-бита 1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду `ls -l / | grep tmp` 2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt` 3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt`

```
[guest@akhamdamova lab5]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Apr 12 10:37 tmp
[guest@akhamdamova lab5]$ echo "test" > /tmp/file01.txt
[guest@akhamdamova lab5]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Apr 12 10:40 /tmp/file01.txt
[guest@akhamdamova lab5]$ chmod o+rw /tmp/file01.txt
[guest@akhamdamova lab5]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Apr 12 10:40 /tmp/file01.txt
[guest@akhamdamova lab5]$
```

- От пользователя `guest2` (не являющегося владельцем) попробуйте прочитать файл `/tmp/file01.txt`: `cat /tmp/file01.txt`
- От пользователя `guest2` попробуйте дозаписать в файл `/tmp/file01.txt` слово

test2 командой `echo "test2" > /tmp/file01.txt` Удалось ли вам выполнить операцию?

6. Проверьте содержимое файла командой `cat /tmp/file01.txt`
7. 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` Удалось ли вам выполнить операцию? Не удалось

```
[guest@akhamdamova lab5]$ su guest2
Password:
[guest2@akhamdamova lab5]$ cat /tmp/file01.txt
test
[guest2@akhamdamova lab5]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@akhamdamova lab5]$ cat /tmp/file01.txt
test
[guest2@akhamdamova lab5]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@akhamdamova lab5]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@akhamdamova lab5]$
```

8. Проверьте содержимое файла командой `cat /tmp/file01.txt`
9. От пользователя `guest2` попробуйте удалить файл `/tmp/file01.txt` Удалось ли вам удалить файл?
10. Повысьте свои права до суперпользователя следующей командой и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`:
11. Покиньте режим суперпользователя командой
12. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет:
13. Повторите предыдущие шаги. Какие наблюдаются изменения?
14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Не удалось
15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`:


```
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@akhamdamova lab5]$ su -
Password:
su: Authentication failure
[guest2@akhamdamova lab5]$ su -
Password:
[root@akhamdamova ~]# exit
logout
[guest2@akhamdamova lab5]$ su
Password:
[root@akhamdamova lab5]# chmod -t /tmp
[root@akhamdamova lab5]# exit
exit
[guest2@akhamdamova lab5]$
```

```
[guest2@akhamdamova lab5]$ su
Password:
[root@akhamdamova lab5]# chmod -t /tmp
[root@akhamdamova lab5]# exit
exit
[guest2@akhamdamova lab5]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Apr 12 10:47 tmp
[guest2@akhamdamova lab5]$
```

```
password:
[root@akhamdamova lab5]# chmod +t /tmp
[root@akhamdamova lab5]# exit
exit
[guest2@akhamdamova lab5]$
```

4 Выводы

Изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.