

Отчёт по лабораторной работе №5

Дисциплина: Информационная безопасность

Шапошникова Айталипа Степановна, НПИбд-02-18

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15

List of Tables

List of Figures

2.1	Программа simpleid.c	6
2.2	Компиляция программы	6
2.3	Выполнение программы simpleid	7
2.4	Программа id	7
2.5	Усложнение программы	7
2.6	Программа simpleid2.c	7
2.7	Изменение прав программы	8
2.8	Проверка	8
2.9	Запуск simpleid2 и id	8
2.10	SetGID-бита	8
2.11	Программа readfile.c	9
2.12	Компиляция readfile.c	9
2.13	Права файла readfile.c	9
2.14	guest не может прочитать файл readfile.c	10
2.15	Права файла readfile.c	10
2.16	Права файла readfile.c	11
2.17	Права файла readfile.c	11
2.18	Атрибут Sticky и создание file01.txt	12
2.19	Атрибуты file01.txt	12
2.20	Проверка	13
2.21	Проверка	13
2.22	Возвращаем атрибут t	14

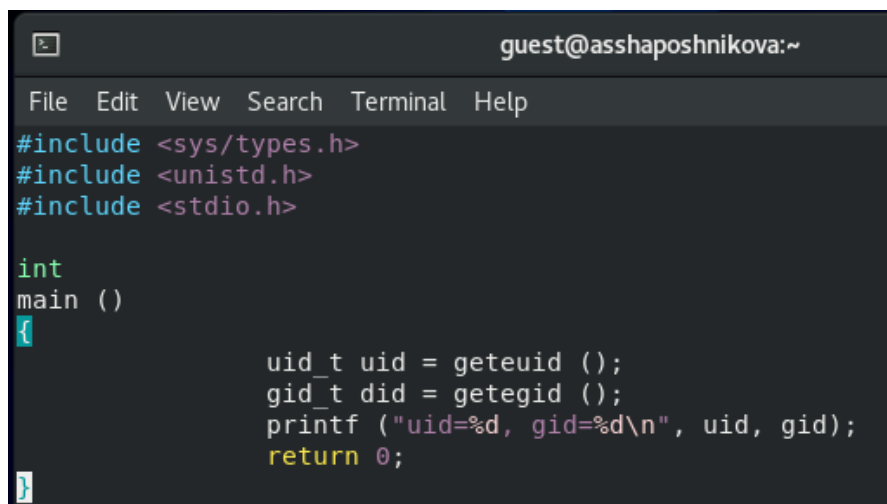
1 Цель работы

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

5.3.1. Создание программы

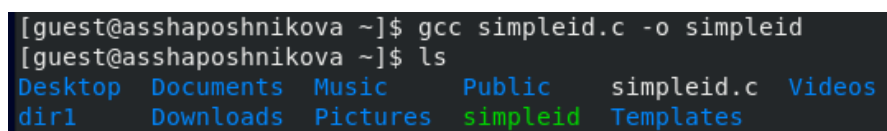
1. Вошли в систему от имени пользователя guest.
2. Создали программу simpleid.c (рис. 2.1).



```
guest@asshaposhnikova:~  
File Edit View Search Terminal Help  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main ()  
{  
    uid_t uid = geteuid ();  
    gid_t did = getegid ();  
    printf ("uid=%d, gid=%d\n", uid, gid);  
    return 0;  
}
```

Figure 2.1: Программа simpleid.c

3. Скомпилировали программу и убедились, что файл программы создан (рис. 2.2).



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

Figure 2.2: Компиляция программы

4. Выполнили программу simpleid (рис. 2.3).

```
[guest@asshaposnikova ~]$ ./simpleid
uid=1001, gid=1001
```

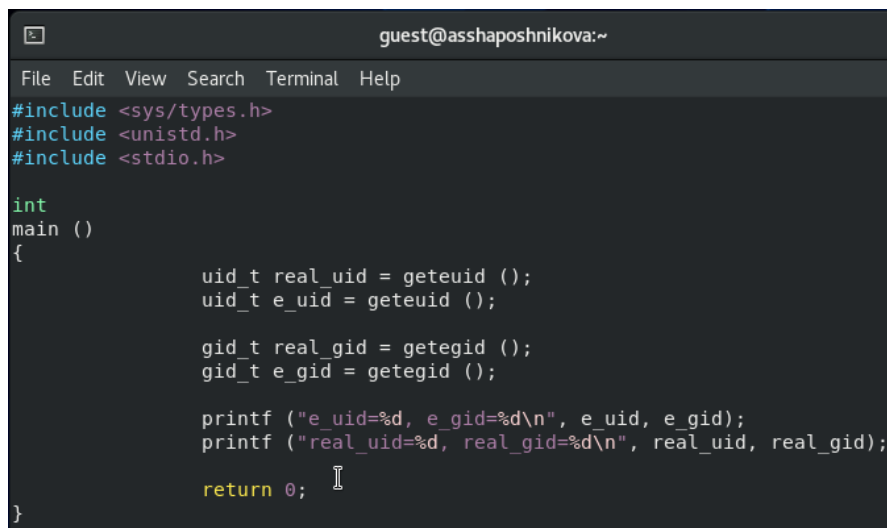
Figure 2.3: Выполнение программы simpleid

5. Выполнили системную программу id (рис. 2.4).

```
[guest@asshaposnikova ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined r:unconfined t:s0-s0:c0.c1023
```

Figure 2.4: Программа id

6. Усложнили программу, добавив вывод действительных идентификаторов (рис. 2.5). Получившуюся программу назвали simpleid2.c.



```
guest@asshaposnikova:~
File Edit View Search Terminal Help
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = getuid ();

    gid_t real_gid = getgid ();
    gid_t e_gid = getgid ();

    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 2.5: Усложнение программы

7. Скомпилировали и запустили simpleid2.c (рис. 2.6).

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

Figure 2.6: Программа simpleid2.c

8. От имени суперпользователя выполнили команды (рис. 2.7).

```
[guest@asshaposhnikova ~]$ su
Password:
[root@asshaposhnikova guest]# chown root:guest /home/guest/simpleid2
[root@asshaposhnikova guest]# chmod u+s /home/guest/simpleid2
```

Figure 2.7: Изменение прав программы

9. Повысили временно свои права с помощью su.
10. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (рис. 2.8).

```
[root@asshaposhnikova guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 17544 Nov 12 14:29 simpleid2
```

Figure 2.8: Проверка

11. Запустили simpleid2 и id (рис. 2.9).

```
[root@asshaposhnikova guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@asshaposhnikova guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@asshaposhnikova guest]#
```

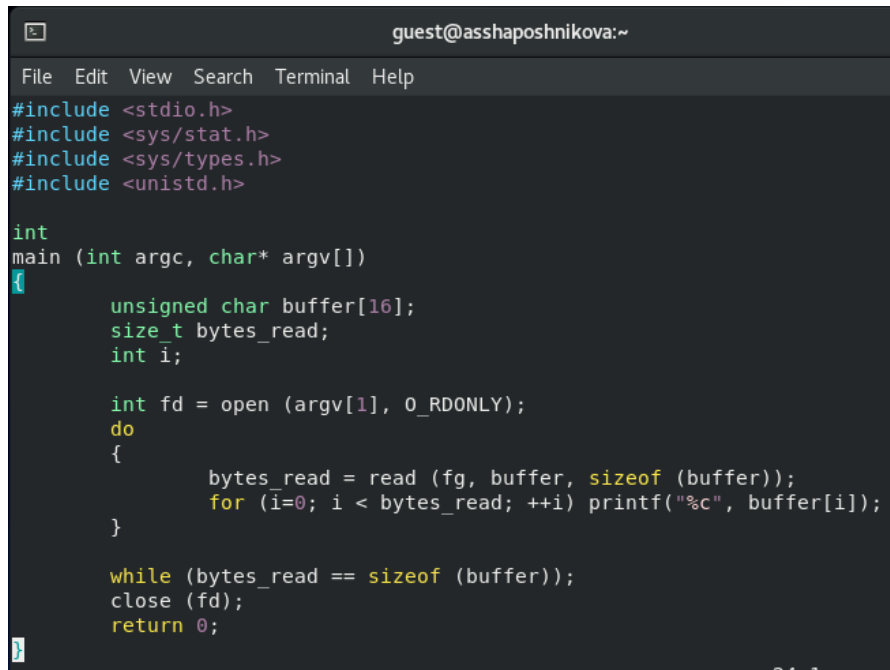
Figure 2.9: Запуск simpleid2 и id

12. Проделали тоже самое относительно SetGID-бита (рис. 2.10).

```
[root@asshaposhnikova guest]# chmod g+s /home/guest/simpleid2
[root@asshaposhnikova guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 17544 Nov 12 14:29 simpleid2
[root@asshaposhnikova guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=1001
[root@asshaposhnikova guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
```

Figure 2.10: SetGID-бита

13. Создали программу readfile.c (рис. 2.11).

A terminal window titled 'guest@asshaposhnikova:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The code is as follows:

```
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

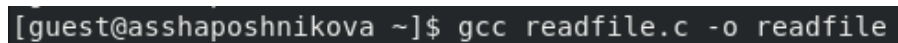
int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    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;
}
```

Figure 2.11: Программа readfile.c

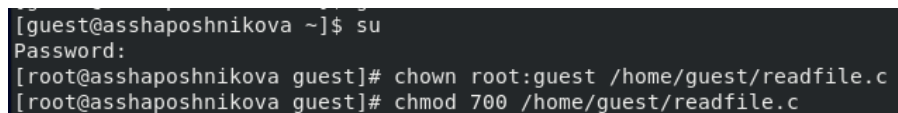
14. Откомпилировали её (рис. 2.12).

A terminal window showing the command to compile the program:

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

Figure 2.12: Компиляция readfile.c

15. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. 2.13).

A terminal window showing the commands to change ownership and permissions:

```
[guest@asshaposhnikova ~]$ su
Password:
[root@asshaposhnikova guest]# chown root:guest /home/guest/readfile.c
[root@asshaposhnikova guest]# chmod 700 /home/guest/readfile.c
```

Figure 2.13: Права файла readfile.c

16. Проверили, что пользователь guest не может прочитать файл readfile.c (рис. 2.14).

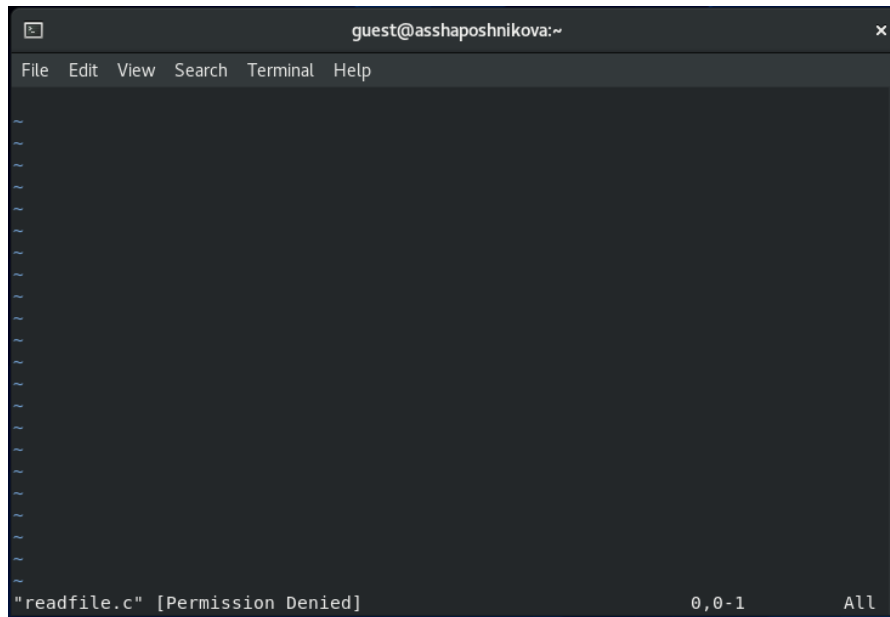


Figure 2.14: guest не может прочитать файл readfile.c

17. Сменили у программы readfile владельца и установили SetUID-бит (рис. 2.15).

```
[guest@asshaposhnikova ~]$ su
Password:
[root@asshaposhnikova guest]# chown root:guest /home/guest/readfile
[root@asshaposhnikova guest]# chmod u+s /home/guest/readfile
```

Figure 2.15: Права файла readfile.c

18. Проверили, может ли программа readfile прочитать файл readfile.c (рис. 2.16).

```
[guest@asshaposhnikova ~]$ ./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;
    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 2.16: Права файла readfile.c

19. Проверили, может ли программа readfile прочитать файл /etc/shadow (рис. 2.17).

```
[guest@asshaposhnikova ~]$ ./readfile /etc/shadow
root:$6$0BD.oEG220I03ocI$RfLLZpL5pU.7GUEeeg93gkDPTBjhY.ezTan03UFQmZc0YuCf30V30dS
fUJxJjcBgJ9HBsif2Qqtr9lWGbqqeE.: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:::
shutdown*:18397:0:99999:7:::
halt*:18397:0:99999:7:::
mail*:18397:0:99999:7:::
operator*:18397:0:99999:7:::
games*:18397:0:99999:7:::
ftp*:18397:0:99999:7:::
nobody*:18397:0:99999:7:::
dbus:!!:18879::::::
systemd-coredump:!!:18879::::::
systemd-resolve:!!:18879::::::
tss:!!:18879::::::
polkitd:!!:18879::::::
geoclue:!!:18879::::::
rtkit:!!:18879::::::
pipewire:!!:18879::::::
pulse:!!:18879::::::
```

Figure 2.17: Права файла readfile.c

5.3.2. Исследование Sticky-бита

1. Выяснили, установлен ли атрибут Sticky на директории /tmp (рис. 2.18). Да, установлен.

2. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test (рис. 2.18).

```
[guest@asshaposhnikova ~]$ ls -l / | grep tmp
drwxrwxrwt. 12 root root 4096 Nov 12 15:08 tmp
[guest@asshaposhnikova ~]$ echo "test" > /tmp/file01.txt
```

Figure 2.18: Атрибут Sticky и создание file01.txt

3. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные» (рис. 2.19).

```
[guest@asshaposhnikova ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Nov 12 15:11 /tmp/file01.txt
[guest@asshaposhnikova ~]$ chmod o+rw /tmp/file01.txt
[guest@asshaposhnikova ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Nov 12 15:11 /tmp/file01.txt
```

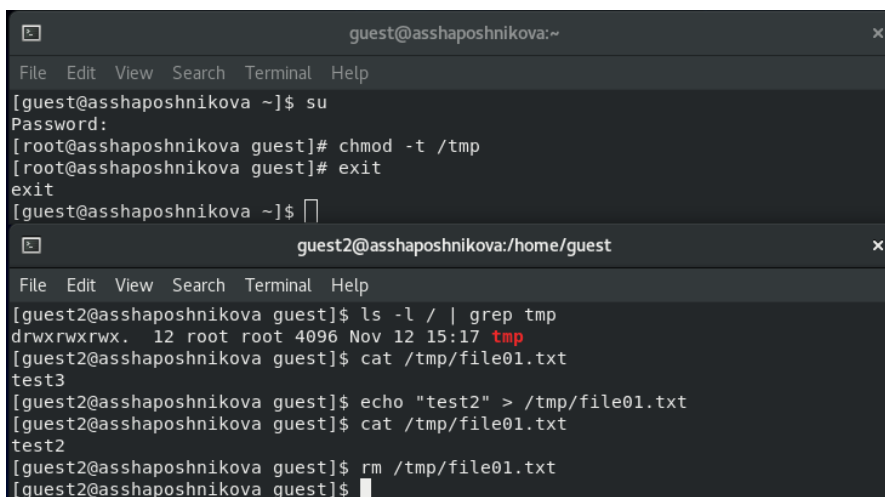
Figure 2.19: Атрибуты file01.txt

4. От пользователя guest2 (не являющегося владельцем) попробовали прочесть файл /tmp/file01.txt (рис. 2.20). Получилось.
5. От пользователя guest2 попробовали дозаписать в файл /tmp/file01.txt слово test2 (рис. 2.20). Получилось.
6. Проверили содержимое файла (рис. 2.20). Получилось.
7. От пользователя guest2 попробовали записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию (рис. 2.20). Получилось.
8. Проверили содержимое файла (рис. 2.20). Получилось.
9. От пользователя guest2 попробовали удалить файл /tmp/file01.txt (рис. 2.20). Не получилось.

```
[guest@asshaposhnikova ~]$ su guest2
Password:
[guest2@asshaposhnikova guest]$ cat /tmp/file01.txt
test
[guest2@asshaposhnikova guest]$ echo "test2" > /tmp/file01.txt
[guest2@asshaposhnikova guest]$ cat /tmp/file01.txt
test2
[guest2@asshaposhnikova guest]$ echo "test3" > /tmp/file01.txt
[guest2@asshaposhnikova guest]$ cat /tmp/file01.txt
test3
[guest2@asshaposhnikova guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@asshaposhnikova guest]$
```

Figure 2.20: Проверка

10. Повысили свои права до суперпользователя и выполнили после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp (рис. 2.21).
11. Покинули режим суперпользователя (рис. 2.21).
12. От пользователя guest2 проверили, что атрибута t у директории /tmp нет (рис. 2.21).
13. Повторили предыдущие шаги (рис. 2.21).
14. Удалось удалить файл от имени пользователя, не являющегося его владельцем (рис. 2.21).



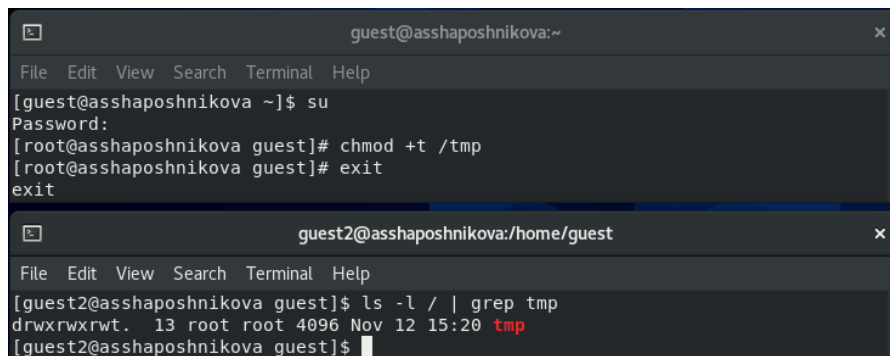
```

guest@asshaposhnikova:~
File Edit View Search Terminal Help
[guest@asshaposhnikova ~]$ su
Password:
[root@asshaposhnikova guest]# chmod -t /tmp
[root@asshaposhnikova guest]# exit
exit
[guest@asshaposhnikova ~]$

guest2@asshaposhnikova:/home/guest
File Edit View Search Terminal Help
[guest2@asshaposhnikova guest]$ ls -l / | grep tmp
drwxrwxrwx. 12 root root 4096 Nov 12 15:17 tmp
[guest2@asshaposhnikova guest]$ cat /tmp/file01.txt
test3
[guest2@asshaposhnikova guest]$ echo "test2" > /tmp/file01.txt
[guest2@asshaposhnikova guest]$ cat /tmp/file01.txt
test2
[guest2@asshaposhnikova guest]$ rm /tmp/file01.txt
[guest2@asshaposhnikova guest]$
```

Figure 2.21: Проверка

15. Повысили свои права до суперпользователя и вернули атрибут `t` на директорию `/tmp` (рис. 2.22).



The image shows two terminal windows. The top window is titled 'guest@asshaposhnikova:~' and shows the user switching to root with 'su', then running 'chmod +t /tmp' and exiting. The bottom window is titled 'guest2@asshaposhnikova:/home/guest' and shows the user running 'ls -l / | grep tmp', which displays the permissions 'drwxrwxrwt.' for the /tmp directory, with the 't' in red.

```
guest@asshaposhnikova:~  
File Edit View Search Terminal Help  
[guest@asshaposhnikova ~]$ su  
Password:  
[root@asshaposhnikova guest]# chmod +t /tmp  
[root@asshaposhnikova guest]# exit  
exit  
  
guest2@asshaposhnikova:/home/guest  
File Edit View Search Terminal Help  
[guest2@asshaposhnikova guest]$ ls -l / | grep tmp  
drwxrwxrwt. 13 root root 4096 Nov 12 15:20 tmp  
[guest2@asshaposhnikova guest]$
```

Figure 2.22: Возвращаем атрибут `t`

3 Выводы

После выполнения лабораторной работы №5 мы изучили механизмы 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.