

# **Отчёт по лабораторной работе**

**Лабораторная №5**

Дерябина Мария Сергеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Исследование SetUID- и SetGID-битов . . . . .	6
2.2	Исследование Sticky-бита . . . . .	12
<b>3</b>	<b>Вывод</b>	<b>14</b>

# List of Tables

# List of Figures

2.1	Установка gss . . . . .	6
2.2	Снятие ограничений SELinux . . . . .	6
2.3	Код программы simpleid.c . . . . .	7
2.4	Компиляция и выполнение программы simpleid.c . . . . .	7
2.5	Код программы simpleid2.c . . . . .	8
2.6	Компиляция и выполнение программы simpleid2.c . . . . .	8
2.7	Изменение атрибутов программы simpleid2 . . . . .	8
2.8	Вывод программы simpleid2 с атрибутом SetUID . . . . .	9
2.9	Добавление атрибута SetGID к программе simpleid2 . . . . .	9
2.10	Вывод программы simpleid2 с атрибутом SetUID и SetGID . . . . .	9
2.11	Код программы readfile.c . . . . .	10
2.12	Выполнение программы readfile . . . . .	10
2.13	Смена атрибутов файла readfile.c . . . . .	11
2.14	Проверка атрибутов файла readfile.c . . . . .	11
2.15	Добавление SetUID-бита к программе readfile . . . . .	11
2.16	Чтение файла readfile.c с помощью readfile . . . . .	11
2.17	Чтение файла /etc/shadow с помощью readfile . . . . .	12
2.18	Проверка атрибута Sticky и создание файла в /tmp . . . . .	12
2.19	Выполнение операций над file01.txt от имени guest2 . . . . .	13
2.20	Снятие атрибута Sticky с /tmp . . . . .	13
2.21	Выполнение операций над file01.txt со снятым атрибутом Sticky . . . . .	13

# 1 Цель работы

Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-биты. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

## 2 Выполнение лабораторной работы

Для выполнения работы, установила компилятор gcc и отключила защиту SELinux (рис. 2.1, 2.2).

```
[root@msderyabina guest]# yum install gcc -y
CentOS Linux 8 - AppStream        6.6 kB/s | 4.3 kB      00:00
CentOS Linux 8 - AppStream        1.2 MB/s | 9.6 MB      00:08
CentOS Linux 8 - BaseOS           8.4 kB/s | 3.9 kB      00:00
CentOS Linux 8 - BaseOS           6.4 MB/s | 8.5 MB      00:01
CentOS Linux 8 - Extras           3.3 kB/s | 1.5 kB      00:00
CentOS Linux 8 - Extras           14 kB/s | 10 kB       00:00
Зависимости разрешены.
=====
Пакет                Архитектура  Версия                Репозиторий  Размер
=====
Установка:
gcc                  x86_64       8.4.1-1.el8           appstream    23 M
Установка зависимостей:
cpp                  x86_64       8.4.1-1.el8           appstream    10 M
glibc-devel          x86_64       2.28-151.el8          baseos       1.0 M
glibc-headers        x86_64       2.28-151.el8          baseos       478 k
isl                   x86_64       0.16.1-6.el8          appstream    841 k
kernel-headers       x86_64       4.18.0-305.25.1.el8_4 baseos       7.2 M
libxcrypt-devel       x86_64       4.1.1-4.el8           baseos       25 k
Результат транзакции
```

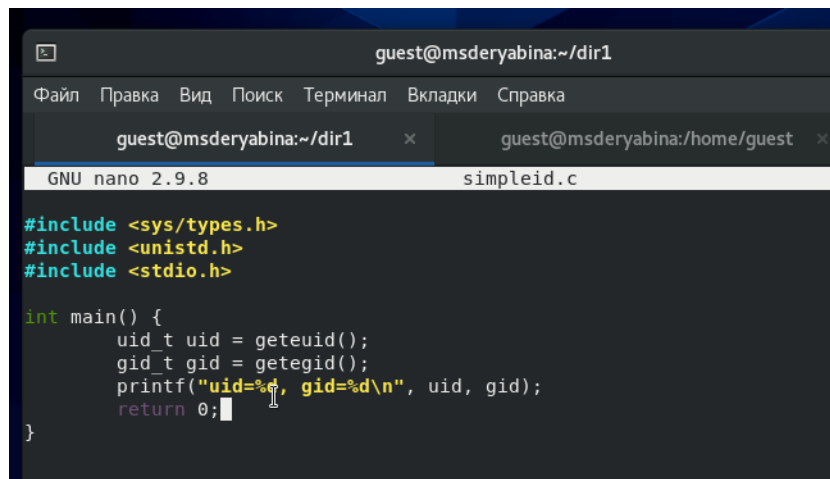
Figure 2.1: Установка gcc

```
[root@msderyabina guest]# getenforce
Enforcing
[root@msderyabina guest]# setenforce 0
[root@msderyabina guest]# getenforce
Permissive
[root@msderyabina guest]#
```

Figure 2.2: Снятие ограничений SELinux

### 2.1 Исследование SetUID- и SetGID-битов

Вошла в систему от пользователя guest и создала программу simpleid.c (рис. 2.3).



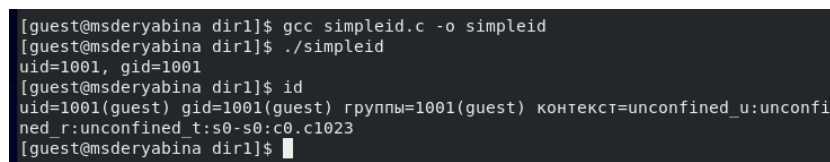
```
guest@msderyabina:~/dir1
Файл Правка Вид Поиск Терминал Вкладки Справка
guest@msderyabina:~/dir1 x guest@msderyabina:/home/guest x
GNU nano 2.9.8 simpleid.c

#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 2.3: Код программы simpleid.c

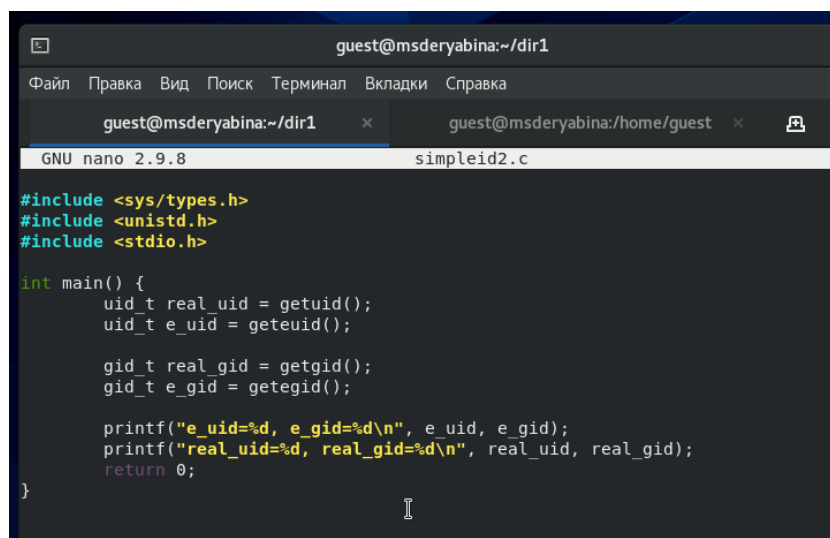
Скомпилировала и выполнила программу. Полученный результат совпал с выводом команды id (рис. 2.4)



```
[guest@msderyabina dir1]$ gcc simpleid.c -o simpleid
[guest@msderyabina dir1]$ ./simpleid
uid=1001, gid=1001
[guest@msderyabina dir1]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@msderyabina dir1]$
```

Figure 2.4: Компиляция и выполнение программы simpleid.c

Добавила в программу вывод действительных идентификаторов, назвала ее simpleid2.c (рис. 2.5).



```
guest@msderyabina:~/dir1
Файл Правка Вид Поиск Терминал Вкладки Справка
guest@msderyabina:~/dir1 x guest@msderyabina:/home/guest x
GNU nano 2.9.8 simpleid2.c

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

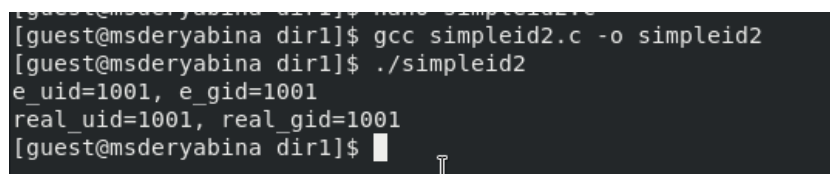
int main() {
    uid_t real_uid = getuid();
    uid_t e_uid = geteuid();

    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;
}
```

Figure 2.5: Код программы simpleid2.c

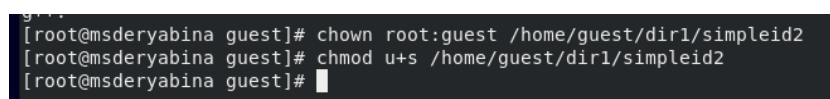
Скомпилировала и запустила программу simpleid2.c. Действительные идентификаторы совпали с эффективными (рис. 2.6)



```
[guest@msderyabina dir1]$ gcc simpleid2.c -o simpleid2
[guest@msderyabina dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@msderyabina dir1]$
```

Figure 2.6: Компиляция и выполнение программы simpleid2.c

От имени суперпользователя изменила владельца программы simpleid2 на root и добавила атрибут SetUID. (рис. 2.7)



```
[root@msderyabina guest]# chown root:guest /home/guest/dir1/simpleid2
[root@msderyabina guest]# chmod u+s /home/guest/dir1/simpleid2
[root@msderyabina guest]#
```

Figure 2.7: Изменение атрибутов программы simpleid2

Проверила правильность установки новых атрибутов и смены владельца файла simpleid2 и запустила simpleid2. Теперь вывод программы отличается от вывода команды id. Действительные идентификаторы остались прежними, а эффективный идентификатор пользователя теперь равен 0 - это идентификатор суперпользователя. Это значит, что



пользователь guest использует права суперпользователя во время выполнения программы (рис. 2.8)

```
[guest@msderyabina dir1]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 13 13:35 simpleid2
[guest@msderyabina dir1]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@msderyabina dir1]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@msderyabina dir1]$
[guest@msderyabina dir1]$
```

Figure 2.8: Вывод программы simpleid2 с атрибутом SetUID

Прodelала то же самое относительно SetGID-бита. Результат оказался аналогичным, теперь при выполнении simpleid2 от пользователя guest эффективный идентификатор группы равен идентификатору группы суперпользователя (рис. 2.9, 2.10)

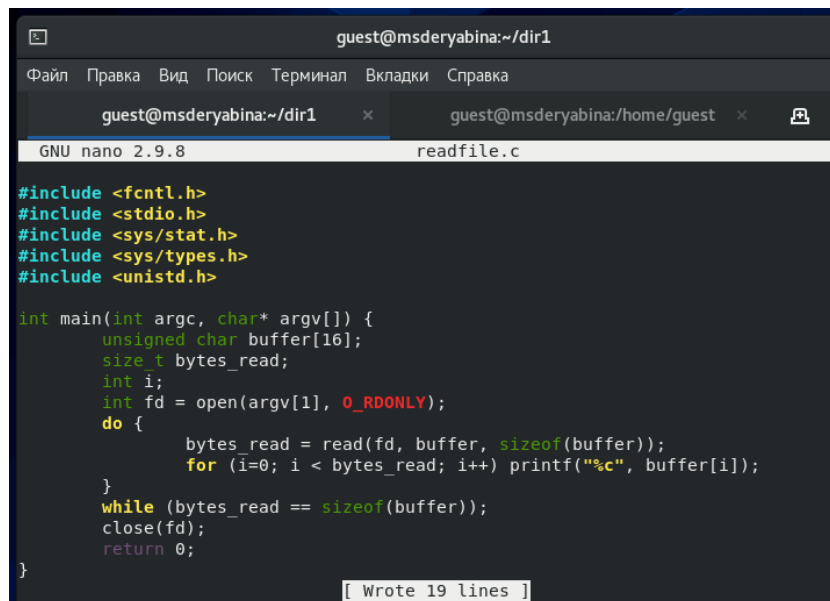
```
[root@msderyabina guest]# chown root:root /home/guest/dir1/simpleid2
[root@msderyabina guest]# chmod g+s /home/guest/dir1/simpleid2
[root@msderyabina guest]# chmod u+s /home/guest/dir1/simpleid2
[root@msderyabina guest]#
```

Figure 2.9: Добавление атрибута SetGID к программе simpleid2

```
[guest@msderyabina dir1]$ ls -l simpleid2
-rwsrwsr-x. 1 root root 17648 ноя 13 13:35 simpleid2
[guest@msderyabina dir1]$ ./simpleid2
e_uid=0, e_gid=0
real_uid=1001, real_gid=1001
[guest@msderyabina dir1]$
```

Figure 2.10: Вывод программы simpleid2 с атрибутом SetUID и SetGID

Создала программу readfile.c (рис. 2.11)



```
guest@msderyabina:~/dir1
Файл Правка Вид Поиск Терминал Вкладки Справка
guest@msderyabina:~/dir1 x guest@msderyabina:/home/guest x
GNU nano 2.9.8 readfile.c

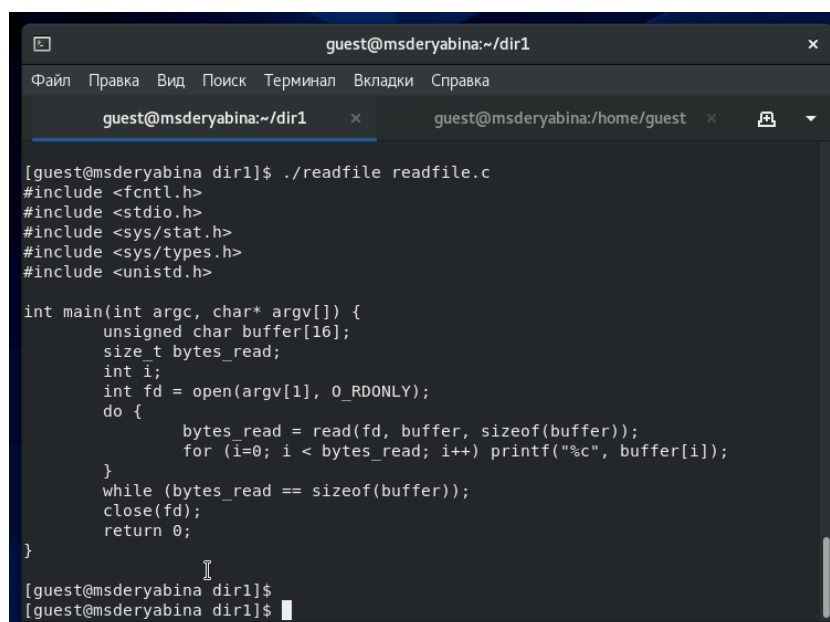
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

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;
}

[ Wrote 19 lines ]
```

Figure 2.11: Код программы readfile.c

Откомпилировала и проверила корректность выполнения программы (рис. 2.12)



```
guest@msderyabina:~/dir1
Файл Правка Вид Поиск Терминал Вкладки Справка
guest@msderyabina:~/dir1 x guest@msderyabina:/home/guest x
[guest@msderyabina dir1]$ ./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;
}
[guest@msderyabina dir1]$
[guest@msderyabina dir1]$
```

Figure 2.12: Выполнение программы readfile

Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь мог прочитать его, а guest не мог (рис. 2.13)

```
[root@msderyabina guest]# chown root:root /home/guest/dir1/readfile.c
[root@msderyabina guest]# chmod o-r /home/guest/dir1/readfile.c
[root@msderyabina guest]#
```

Figure 2.13: Смена атрибутов файла readfile.c

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

```
[guest@msderyabina dir1]$ ls -l readfile.c
-rw-rw----. 1 root root 409 ноя 13 15:18 readfile.c
[guest@msderyabina dir1]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Figure 2.14: Проверка атрибутов файла readfile.c

Сменила у программы readfile владельца на root и установила SetUID-бит (рис. 2.15)

```
[root@msderyabina guest]# chown root:root /home/guest/dir1/readfile
[root@msderyabina guest]# chmod u+s /home/guest/dir1/readfile
[root@msderyabina guest]#
```

Figure 2.15: Добавление SetUID-бита к программе readfile

Теперь с помощью программы readfile можно от имени пользователя guest прочитать файл readfile.c. Также можно прочитать файл /etc/shadow, хотя guest не имеет к нему доступа. (рис. 2.16, 2.17)

```
[guest@msderyabina dir1]$ ./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;
}
[guest@msderyabina dir1]$ ls -l readfile
-rwsrwxr-x. 1 root root 17592 ноя 13 15:19 readfile
[guest@msderyabina dir1]$
```

Figure 2.16: Чтение файла readfile.c с помощью readfile

```
[guest@msderyabina dir1]$ ls -l /etc/shadow
-----. 1 root root 1611 окт 16 15:21 /etc/shadow
[guest@msderyabina dir1]$ ./readfile /etc/shadow
root:$6$bWmmy8CWfa7L8gcR$q/RQDKAd2IoneGZSGE0yBaVPxtNz0ZsMSgHpXcVCe28oo0khVwmBPQd
WC13v/mJAldoKUjorH0dCaNER/1LFd.: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:!!:18888:!!!!:
systemd-coredump:!!:18888:!!!!:
systemd-resolve:!!:18888:!!!!:
tss:!!:18888:!!!!:
polkitd:!!:18888:!!!!:
geoclue:!!:18888:!!!!:
```

Figure 2.17: Чтение файла /etc/shadow с помощью readfile

## 2.2 Исследование Sticky-бита

Посмотрела, что на директории /tmp установлен атрибут Sticky. От имени пользователя guest создала файл file01.txt в директории /tmp со словом “test”. Посмотрела атрибуты у file01.txt и разрешила чтение и запись для категории пользователей “other” (рис. 2.18)

```
[guest@msderyabina dir1]$ ls -l / | grep tmp
drwxrwxrwt. 13 root root 4096 ноя 13 15:29 tmp
[guest@msderyabina dir1]$ echo "test" > /tmp/file01.txt
[guest@msderyabina dir1]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 15:41 /tmp/file01.txt

[guest@msderyabina dir1]$
[guest@msderyabina dir1]$ chmod o+w /tmp/file01.txt
[guest@msderyabina dir1]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 15:41 /tmp/file01.txt

[guest@msderyabina dir1]$
[guest@msderyabina dir1]$
```

Figure 2.18: Проверка атрибута Sticky и создание файла в /tmp

От пользователя guest2 попробовала выполнить различные действия - прочитать файл, дозаписать текст в файл, переписать текст в файле, удалить файл. Получилось сделать все, кроме удаления файла (рис. 2.19)

```

[guest2@msderyabina guest]$ cat /tmp/file01.txt
test

[guest2@msderyabina guest]$
[guest2@msderyabina guest]$ echo "test2" > /tmp/file01.txt
[guest2@msderyabina guest]$ cat /tmp/file01.txt
test2
[guest2@msderyabina guest]$ echo "test3" >> /tmp/file01.txt
[guest2@msderyabina guest]$ cat /tmp/file01.txt
test2
test3
[guest2@msderyabina guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@msderyabina guest]$
[guest2@msderyabina guest]$

```

Figure 2.19: Выполнение операций над file01.txt от имени guest2

От имени суперпользователя сняла Sticky-бит с директории /tmp (рис. 2.20)

```

[root@msderyabina guest]# chmod -t /tmp
[root@msderyabina guest]#

```

Figure 2.20: Снятие атрибута Sticky с /tmp

Повторила предыдущие шаги. В этот раз удалось удалить file01.txt.

Таким образом, со снятым атрибутом Sticky можно удалить из директории файл от имени пользователя, не являющегося его владельцем. Вернула атрибут t на директорию /tmp (рис. 2.21)

```

[guest2@msderyabina guest]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 ноя 13 15:48 tmp
[guest2@msderyabina guest]$ cat /tmp/file01.txt
test2
test3
[guest2@msderyabina guest]$ echo "test2" > /tmp/file01.txt
[guest2@msderyabina guest]$ cat /tmp/file01.txt
test2
[guest2@msderyabina guest]$ rm /tmp/file01.txt
[guest2@msderyabina guest]$ ls -l / | grep tmp
drwxrwxrwt. 13 root root 4096 ноя 13 15:56 tmp
[guest2@msderyabina guest]$

```

Figure 2.21: Выполнение операций над file01.txt со снятым атрибутом Sticky

## 3 Вывод

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