

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**Презентация**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

**дисциплина: Информационная безопасность**

**Студент: Пиняева Анна Андреевна**

**Группа: НФИбд-02-20**

**МОСКВА**

**2023**

## Цель работы

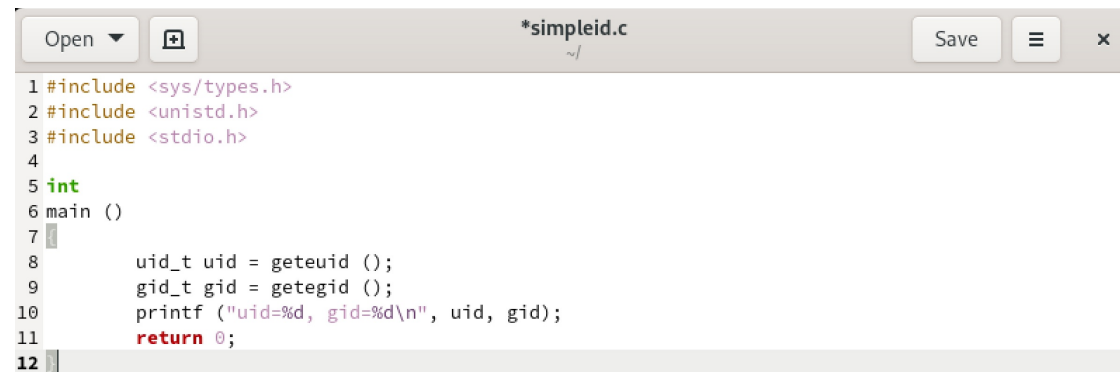
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

---

## Ход работы

1. Вошли в ситсему от имени пользователя guest. Создали программу simplified.c (рис. 1).

Рис. 1 Программа simplified.c:



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

*N/Solid*

2. Скомпилировали программу и убедились, что файл программы создан (рис. 2).

*Рис. 2 Компиляция и запуск программы:*

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

*N/Solid*

---

3. Выполнили программу (рис. 2).

4. Выполнили системную программу id (рис. 3).Получили результат аналогичный результату выполнения программы simplified.c.

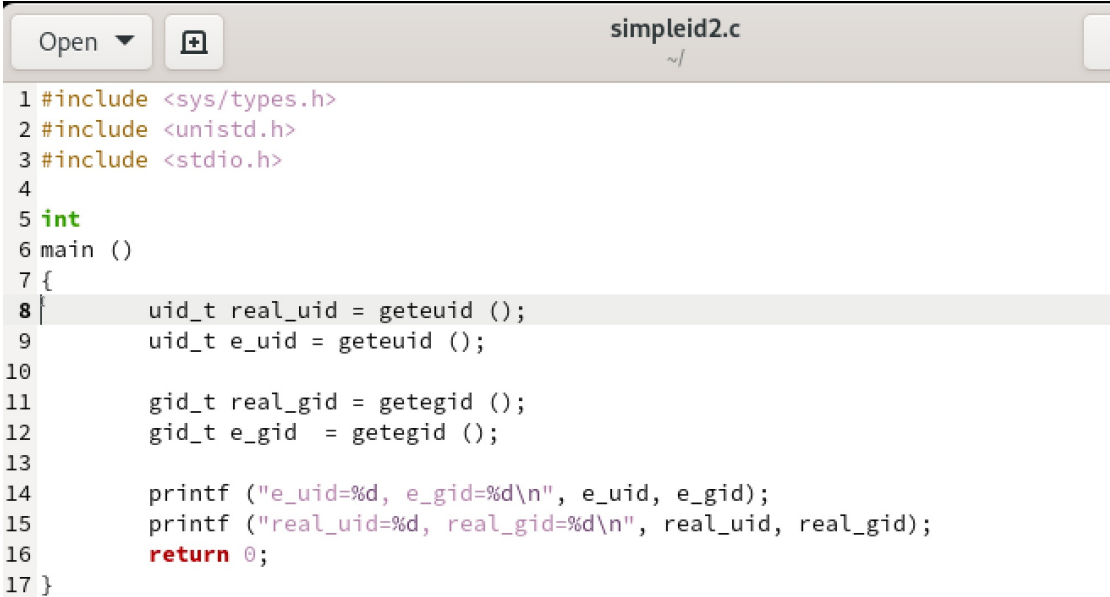
*Рис. 3 Выполнение программы id:*

```
[guest@user ~]$ gcc simpleid.c -o simpleid
[guest@user ~]$ ./simpleid
uid=1001, gid=1001
```

*N/Solid*

5. Создали новый файл с усложненной программой `simplified.c` `simplified2.c` (рис. 4).

Рис. 4 Программа `simplified2.c`:



The image shows a code editor window with a title bar containing 'Open', a '+' icon, the filename 'simplified2.c', and a 'S' icon. The code is written in C and includes standard headers for types, unistd, and stdio. It defines a 'main' function that retrieves real and effective user and group IDs using 'geteuid' and 'getegid', and prints them using 'printf'. The code is as follows:

```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main ()
7 {
8     uid_t real_uid = geteuid ();
9     uid_t e_uid = geteuid ();
10
11     gid_t real_gid = getegid ();
12     gid_t e_gid = getegid ();
13
14     printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
15     printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16     return 0;
17 }
```

*N/Solid*

## 6. Скомпилировали и запустили программу (рис. 5).

Рис. 5 Компиляция и запуск:

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

N/Solid

---

## 7. От имени суперпользователя выполнили команды: `chown root:guest /home/guest/simpleid2`, `chmod u+s /home/guest/simpleid2` (рис. 6).

Рис. 6 Выполнение команд:

```
[root@user user]# chown root:guest /home/guest/simpleid2
[root@user user]# chmod u+s /home/guest/simpleid2
[root@user user]# ls -l simpleid2
ls: cannot access 'simpleid2': No such file or directory
[root@user user]# ls -l /home/guest/simpleid2
-rwsr-xr-x. 1 root guest 25904 Oct  7 13:28 /home/guest/simpleid2
[root@user user]#
```

N/Solid

## 8. Выполнили проверку правильности установки новых атрибутов и смены владельца файла `simpleid2` (рис. 6).

## 9. Запустили simpleid2 и id (рис. 7).

Рис. 7 Запуск программ:

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

N/Solid

## 10. Сделали то же самое относительно SetGID-бита (рис. 8).

Рис. 8 Запуск программ:

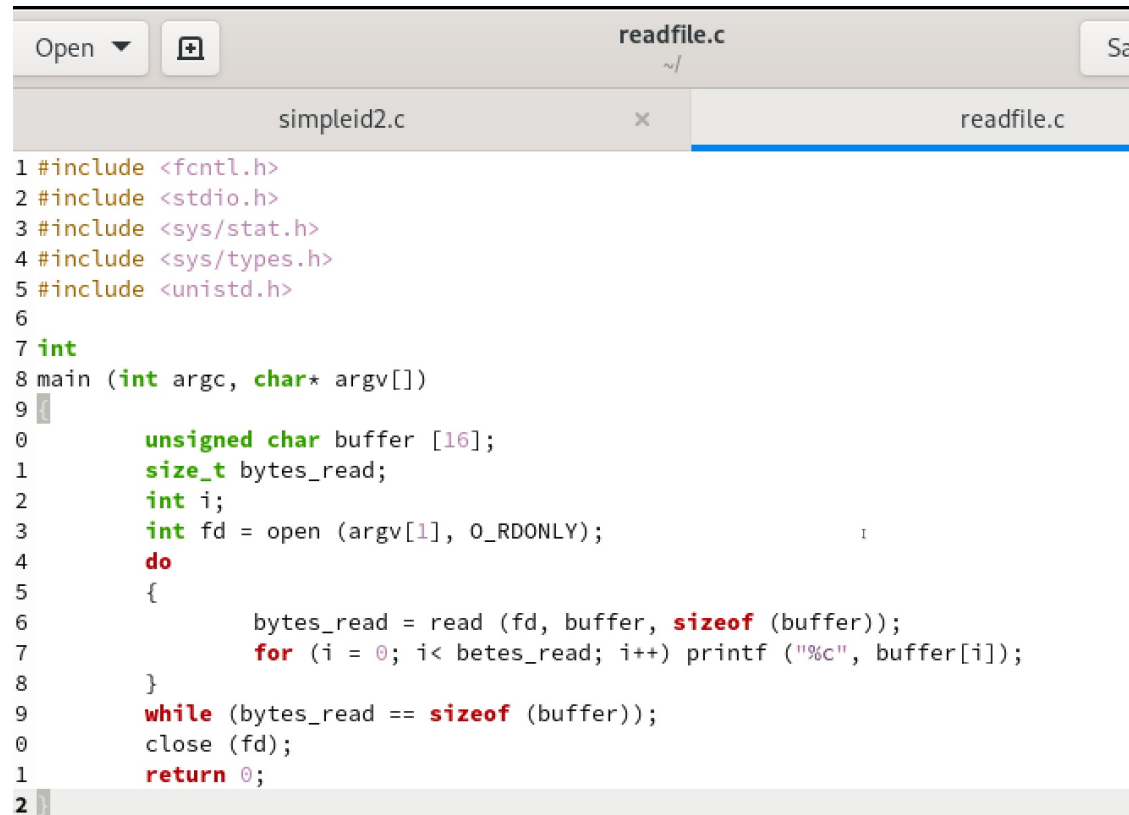
```
[root@user guest]# chmod g+s simpleid2
[root@user guest]# ls -l simpleid2
-rwsr-sr-x. 1 root guest 25904 Oct  7 13:28 simpleid2
[root@user guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=1001
[root@user guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@user guest]#
```

N/Solid

---

## 11. Создали программу readfile.c (рис. 9) и откомпилировали ее.

Рис. 9 Программа 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
7 int
8 main (int argc, char* argv[])
9 {
10     unsigned char buffer [16];
11     size_t bytes_read;
12     int i;
13     int fd = open (argv[1], O_RDONLY);
14     do
15     {
16         bytes_read = read (fd, buffer, sizeof (buffer));
17         for (i = 0; i < bytes_read; i++) printf ("%c", buffer[i]);
18     }
19     while (bytes_read == sizeof (buffer));
20     close (fd);
21     return 0;
22 }
```

*N/Solid*



**12. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. 9).**

*Рис. 10 Смена владельца и установка прав:*

```
[root@user guest]# chown root:guest readfile.c  
[root@user guest]# chmod 700 readfile.c  
[root@user guest]#
```

*N/Solid*

**13. Проверили, что пользователь guest не может прочитать файл readfile.c (рис. 11).**

*Рис. 11 Проверка:*

```
[guest@user user]$ cat readfile.c  
cat: readfile.c: Permission denied  
[guest@user user]$
```

*N/Solid*

---

14. Сменили у программы readfile владельца и установили SetU'D-бит. Проверили может ли программа прочитать файл readfile.c (рис. 12).

Рис. 12 Запуск программы:

```
[root@user guest]# ./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;
}
[root@user guest]#
```

## 15. Проверили может ли программа прочитать файл /etc/shadow (рис. 13).

Рис. 13 Запуск программ:

```
[root@user guest]# ./readfile /etc/shadow
root:$6$sYM56xKBvr7m040c$u41ggUld38htUK8ipPn/SURC5eU0crE0B0qHQBWIexoG6nNCST
0J6yn0823KoUSSCYXr2HDQbgfM1Bhz1::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:!!:19608:~~~~:
dbus:!!:19608:~~~~:
polkitd:!!:19608:~~~~:
avahi:!!:19608:~~~~:
rtkit:!!:19608:~~~~:
sssd:!!:19608:~~~~:
pipewire:!!:19608:~~~~:
```

*N/Solid*

---

16. Выяснили, установлен ли атрибут Sticky на директории /tmp (рис. 14).

Рис. 14 Команды:

```
[guest@user user]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 Oct  7 14:03 tmp
[guest@user user]$ echo "test" > /tmp/file01.txt
[guest@user user]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  7 14:09 /tmp/file01.txt
[guest@user user]$ chmod o+rw /tmp/file01.txt
[guest@user user]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  7 14:09 /tmp/file01.txt
```

N/Solid

17. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test (рис. 14).

18. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные» (рис. 14).

---

19. От пользователя guest2 попробовали прочитать файл /tmp/file01.txt (рис. 15).

Рис. 15 Чтение файла:

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

N/Solid

20. От пользователя `guest2` дозаписали в файл `/tmp/file01.txt` слово `test2` (рис. 16).

Рис. 16 Дозапись в файл:

```
[guest2@user user]$ echo "test2" >> /tmp/file01.txt
[guest2@user user]$ cat /tmp/file01.txt
test
test2
```

*N/Solid*

21. От пользователя `guest2` попробуйте записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию (рис. 17).

Рис. 17 Изменение файла:

```
[guest2@user user]$ echo "test3" > /tmp/file01.txt
[guest2@user user]$ cat /tmp/file01.txt
test3
[guest2@user user]$
```

*N/Solid*

---

22. От пользователя `guest2` попробовали удалить файл `/tmp/file01.txt` (рис. 18).

Рис. 18 Удаление файла:

```
[guest2@user user]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@user user]$
```

*N/Solid*

### 23. От имени суперпользователя сняли атрибут t с файла (рис. 19).

Рис. 19 Снятие атрибута:

```
[guest2@user user]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 Oct  7 14:11 tmp
[guest2@user user]$
```

N/Solid

### 24. Попробовали выполнить все предыдущие действия заново (рис. 20).

Рис. 20 Выполнение команд:

```
[guest2@user user]$ echo "test2" >> /tmp/file01.txt
[guest2@user user]$ cat /tmp/file01.txt
test3
test2
[guest2@user user]$ echo "test3" > /tmp/file01.txt
[guest2@user user]$ cat /tmp/file01.txt
test3
[guest2@user user]$ rm /tmp/file01.txt
[guest2@user user]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@user user]$
```

N/Solid

## 25. Вернули атрибут t (рис. 21).

*Рис. 21 Установка атрибута:*

```
[root@user guest]# chmod +t /tmp  
[root@user guest]# exit  
exit  
[user@user ~]$
```

*N/Solid*

---

## Вывод