

Лабораторная работа №5

**Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов**

Савурская Полина

Содержание

1	Цель работы	3
2	Задачи	4
3	Выполнение лабораторной работы	5
3.1	Создание программы	5
3.2	Исследование Sticky-бита	10
4	Вывод	13

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.

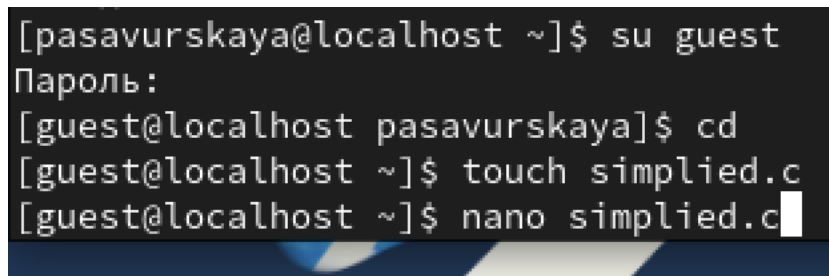
2 Задачи

- Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов.
- 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.

3 Выполнение лабораторной работы

3.1 Создание программы

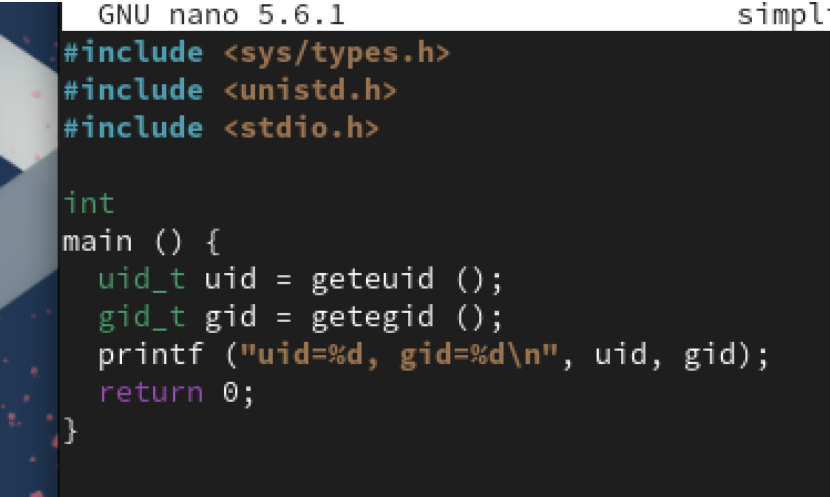
- 1) Войдем в систему от имени пользователя guest (рис. [3.1]).



```
[pasavurskaya@localhost ~]$ su guest
Пароль:
[guest@localhost pasavurskaya]$ cd
[guest@localhost ~]$ touch simplified.c
[guest@localhost ~]$ nano simplified.c
```

Рис. 3.1: Вход

- 2) Создадим программу simpleid.c (рис. [3.2] - [3.3]).



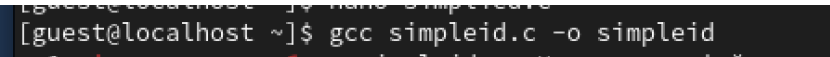
```

GNU nano 5.6.1                                simpl-
#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. 3.2: Программа




```

[guest@localhost ~]$ gcc simpleid.c -o simpleid

```

Рис. 3.3: Программа

3) Скомпилируем программу (рис. [3.4]).



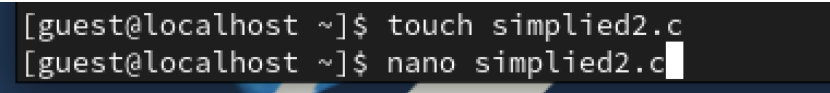
```

[guest@localhost ~]$ ./simpleid

```

Рис. 3.4: Компиляция

4) Выполним команду ./simpleid. После выполним команду id и сравним их (рис. [3.5]).



```

[guest@localhost ~]$ touch simplified2.c
[guest@localhost ~]$ nano simplified2.c

```

Рис. 3.5: Команда

5) Создадим программу simpleid2.c (рис. [3.6] - [3.7]).

```

GNU nano 5.6.1                                simplified2.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;
}

```

Рис. 3.6: Программа

```

[guest@localhost ~]$ gcc simpleid2.c -o simpleid2

```

Рис. 3.7: Программа

6) Скомпилируем программу (рис. [3.8]).

```

[guest@localhost ~]$ ./simpleid2

```

Рис. 3.8: Компиляция

7) Запустим программу (рис. [3.9]).

```

[guest@localhost ~]$ su -
Пароль:
Последний вход в систему: Сб апр 13 14:35:30 MSK 2024 на pts/0
[root@localhost ~]# chown root:guest /home/guest/simpleid2
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет
и каталога
[root@localhost ~]# chmod u+s /home/guest/simpleid2
chmod: невозможно получить доступ к '/home/guest/simpleid2': Нет
и каталога
[root@localhost ~]# exit
Выход

```

Рис. 3.9: Программа

8) От имени суперпользователя выполним следующие команды (рис. [3.10]).

```
[root@localhost ~]# chown root:guest /home/guest/simpleid2
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла или каталога
[root@localhost ~]# chmod u+s /home/guest/simpleid2
chmod: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла или каталога
[root@localhost ~]# exit
Выход
[guest@localhost ~]$ ls -l simpleid2
ls: невозможно получить доступ к 'simpleid2': Нет такого файла или каталога
[guest@localhost ~]$ ./simpleid2
bash: ./simpleid2: Нет такого файла или каталога
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_r:unconfined_t:s0-s0:c0.c1023
```

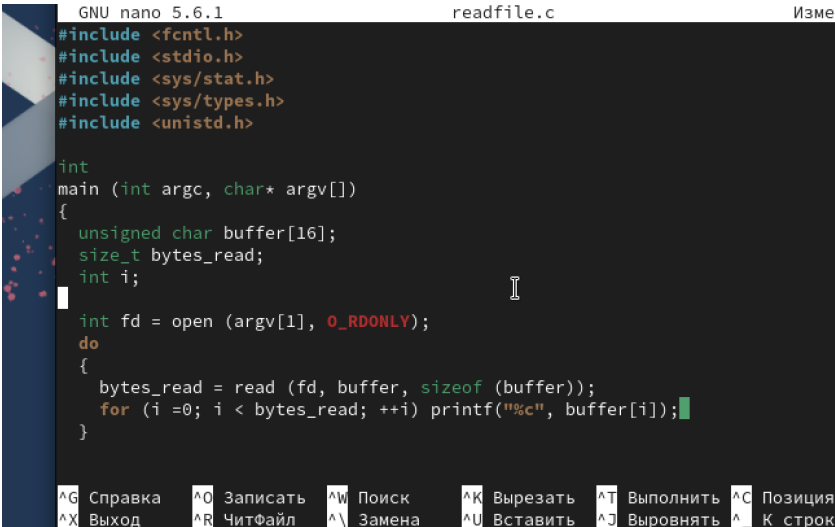
Рис. 3.10: Команды

9) Выполним проверку правильности установки новых атрибутов (рис. [3.11]).

```
[guest@localhost ~]$ touch readfile.c
[guest@localhost ~]$ nano readfile.c
```

Рис. 3.11: Проверка

10) Запустим simpleid2 и id, сравним результаты (рис. [3.12]).



```
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]);
    }
}
```

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^_ Замена ^U Вставить ^J Выровнять ^_ К строку

Рис. 3.12: Команды

11) Создадим программу readfile.c (рис. [3.13] - [3.14]).

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

Рис. 3.13: Программа

```
[guest@localhost ~]$ ./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
```

Рис. 3.14: Программа

12) Откомпилируем её (рис. [3.15]).

```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 apr 13 14:50 tmp
[guest@localhost ~]$
```

Рис. 3.15: Компиляция

13) Выполним команду (рис. [3.16]).

```
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$
```

Рис. 3.16: Команда

3.2 Исследование Sticky-бита

- 1) Выясним, установлен ли атрибут Sticky на директории /tmp (рис. [3.17]).

```
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 апр 13 14:52 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 апр 13 14:52 /tmp/file01.txt
[guest@localhost ~]$
```

Рис. 3.17: Sticky

- 2) От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test (рис. [3.18]).

```
[guest@localhost ~]$ cat /tmp/file01.txt
test
```

Рис. 3.18: Файл

- 3) Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» (рис. [3.19]).

```
[guest@localhost ~]$ echo "test2" /tmp/file01.txt
test2 /tmp/file01.txt
[guest@localhost ~]$ cat /tmp/file01.txt
test
[guest@localhost ~]$
```

Рис. 3.19: Изменение атрибутов

- 4) От пользователя guest2 попробуем прочесть файл (рис. [3.20]).

```
[guest@localhost ~]$ rm /tmp/file01.txt
```

Рис. 3.20: Чтение файла

- 5) От пользователя guest2 попробуем дозаписать в файл слово test2 (рис. [3.21]).
Операцию выполнить не удалось.

```
[guest@localhost ~]$ su -  
Пароль:  
Последний вход в систему: Сб апр 13 14:39:45 MSK 2024 на pts/0  
[root@localhost ~]# chmod -t /tmp  
[root@localhost ~]# exit  
выход  
[guest@localhost ~]$
```

Рис. 3.21: Дозапись

- 6) От пользователя guest2 попробуем удалить файл (рис. [3.22]). Операцию
выполнить не удалось.

```
[guest@localhost ~]$ echo "test2" /tmp/file01.txt  
test2 /tmp/file01.txt
```

Рис. 3.22: Удаление файла

- 7) Повысим свои права до суперпользователя и выполним после этого коман-
ду, снимающую атрибут t с директории /tmp [3.23].

```
[guest@localhost ~]$ cat /tmp/file01.txt  
cat: /tmp/file01.txt: Нет такого файла или каталога
```

Рис. 3.23: Снятие атрибута

- 8) Снова попробуем дозаписать в файл слово test2 (рис. [3.24]). Снова операцию
выполнить не удалось.

```
[guest@localhost ~]$ rm /tmp/file01.txt  
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
```

Рис. 3.24: Дозапись

- 9) Снова попробуем удалить файл (рис. [3.25]). Теперь операцию выполнить удалось.

```
[guest@localhost ~]$ su -  
Пароль:  
Последний вход в систему: Сб апр 13 14:56:23  
[root@localhost ~]# chmod +t /tmp  
[root@localhost ~]# exit  
выход  
[guest@localhost ~]$
```

Рис. 3.25: Удаление файла

- 10) Вернем атрибут t на директорию /tmp (рис. [3.26]).

```
[guest@localhost ~]$ su -  
Пароль:  
Последний вход в систему: Сб апр 13 14:56:23  
[root@localhost ~]# chmod +t /tmp  
[root@localhost ~]# exit  
выход  
[guest@localhost ~]$
```

Рис. 3.26: Возвращение атрибута

4 Вывод

В результате выполнения работы я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. 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.