

# Отчёт по лабораторной работе 5

## Простейший вариант

Еленга Невлора Люглеш

### Содержание

|     |                                      |   |
|-----|--------------------------------------|---|
| 1   | Цель работы.....                     | 1 |
| 2   | Выполнение лабораторной работы ..... | 1 |
| 2.1 | Создание программы .....             | 1 |
| 2.2 | Исследование Sticky-бита .....       | 6 |
| 3   | Вывод .....                          | 7 |

## 1 Цель работы

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

## 2 Выполнение лабораторной работы

Setuid, Setgid и Sticky Bit. Это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги. Статья будет полезна пользователям и администраторам, которые уже знакомы с настройкой базовых прав в операционных системах Linux.

### 2.1 Создание программы

1.Вошли в систему от имени пользователя guest.

```
su: соои при проверке подлинности
[elenga@newlora ~]$ su guest
Пароль:
[guest@newlora elenga]$ emacs simpleid.c
bash: emacs: команда не найдена...
[guest@newlora elenga]$ vim simpleid.c
[guest@newlora elenga]$ █
```

Рис. 1.1.

2.Создали программу simpleid.c:

```

Файл  Правка  Вид  Поиск  Терминал  Справка
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t uid = getuid ();
    gid_t gid = getgid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
~

```

Рис. 1.2.

### 3.Скомпилировали программу

создан: gcc simpleid.c-osimpleid

### 4.Выполнили программу simpleid: ./simpleid

5.Выполнили системную программу id: id исравните полученный вами результат с данными предыдущего пункта задания.

```

./simpleid.c: line 8:      uid_t uid = getuid (); ,
[guest@newlora ~]$ vim simplified.c
[guest@newlora ~]$ gcc simplified.c -o simplified
[guest@newlora ~]$ ./simplified.c
./simplified.c: line 5: int: команда не найдена
./simplified.c: line 8: syntax error near unexpected token `('
./simplified.c: line 8: `    uid t uid = qetuid ();'

```

Рис. 1.3.

### 6.Усложнили программу,добавив вывод действительных идентификаторов:

```

#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. 1.4.

Получившуюся программу назвали simpleid2.c.

### 7.Скомпилировали и запустили simpleid2.c:

```

[guest@newlora ~]$ vim simpleid2.c
[guest@newlora ~]$ gcc simpleid2.c -o simpleid2
[guest@newlora ~]$ ./simpleid2.c
bash: ./simpleid2.c: Отказано в доступе
[guest@newlora ~]$ vim simpleid2.c
[guest@newlora ~]$

```

Рис. 1.5.

## 8. От имени супер пользователя выполнили команды:

```

Последний вход в систему: Сб окт 7 17:27:20 MSK 2023 на pts/0
root@newlora ~]# chown root:guest /home/guest/simpleid2
bash: chown: команда не найдена...
root@newlora ~]# chown root:guest /home/guest/simpleid2
root@newlora ~]# chmod u+s /home/guest/simpleid2

```

Рис. 1.6.

9. Использовали `sudo` или повысили временно свои права с помощью `su`.  
Поясните, что делают эти команды.

10. Выполнили проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`:

```

[guest@newlora ~]$ ./simpleid2.c
-bash: ./simpleid2.c: Отказано в доступе
[guest@newlora ~]$ ls -l
bash: ls-l: команда не найдена...
[guest@newlora ~]$ ls -l
итого 48
-rwxrwxr-x. 1 guest guest 8512 окт 7 17:55 a.out
drwxr-xr-x. 2 guest guest 32 сен 21 14:29 dir1
-rw-r--r--. 1 guest2 guest 5 сен 18 22:27 file1
-rw-r--r--. 1 guest2 guest 0 сен 18 22:27 file2
-rwsrwxr-x. 1 root guest 8616 окт 7 18:38 simpleid2
-rwSrwsr--. 1 guest guest 363 окт 7 18:40 simpleid2.c
-rwxrwxr-x. 1 guest guest 8512 окт 7 18:13 simplified
-rwsrwsr--. 1 guest guest 192 окт 7 18:12 simplified.c
drwxr-xr-x. 2 guest guest 6 сен 21 14:28 Видео
drwxr-xr-x. 2 guest guest 6 сен 21 14:28 Документы
drwxr-xr-x. 2 guest guest 6 сен 21 14:28 Загрузки
drwxr-xr-x. 2 guest guest 6 сен 21 14:28 Изображения
drwxr-xr-x. 2 guest guest 6 сен 21 14:28 Музыка
drwxr-xr-x. 2 guest guest 6 сен 21 14:28 Общедоступные
drwxr-xr-x. 2 guest guest 6 сен 21 14:28 Рабочий стол
drwxr-xr-x. 2 guest _ guest 6 сен 21 14:28 Шаблоны

```

Рис. 1.7.

## 11. Запустите `simpleid2` и `id`:

```

guest@newlora ~]# ./simpleid2.c
./simpleid2.c: line 5: int: команда не найдена
./simpleid2.c: line 8: syntax error near unexpected token `('
./simpleid2.c: line 8: `      uid_t real_uid = getuid ();'

```

Рис. 1.8.

## 12. Проделали тоже самое относительно SetGID-бита.

```

guest@newlora ~]$ vim simpleid2.c
guest@newlora ~]$ ./simpleid2.c
/simpleid2.c: line 5: int: команда не найдена
/simpleid2.c: line 8: syntax error near unexpected token `('
/simpleid2.c: line 8: `      uid_t real_uid = getuid ();'
guest@newlora ~]$ vim readfile.c

```

Рис. 1.9.

### 13.Создали программу readfile.c:

```

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

```

Рис. 1.10.

### 14.Откомпилировали её. gcc read file.c-o readfile

```

guest@newlora ~]$ vim readfile.c
guest@newlora ~]$ gcc readfile.c -o readfile
guest@newlora ~]$ vim readfile.c
guest@newlora ~]$
guest@newlora ~]$ gcc readfile.c -o readfile
guest@newlora ~]$ ./readfile.c
bash: ./readfile.c: Отказано в доступе
guest@newlora ~]$ █

```

Рис. 1.11.

15.Сменили владельца у файла read file.c(или любого другого текстового файла в системе)иизмените права так,чтобы только супер пользователь (root) мог прочитать его,aguest не мог.



```
38;5;9:*.z=38;5;9:*.Z=38;5;9:*.dz=38;5;9:*.gz=38;5;9:*.lrz=38;
zo=38;5;9:*.xz=38;5;9:*.bz2=38;5;9:*.bz=38;5;9:*.tbz=38;5;9:*.
;5;9:*.deb=38;5;9:*.rpm=38;5;9:*.jar=38;5;9:*.war=38;5;9:*.ear
9:*.rar=38;5;9:*.alz=38;5;9:*.ace=38;5;9:*.zoo=38;5;9:*.cpio=3
.rz=38;5;9:*.cab=38;5;9:*.jpg=38;5;13:*.jpeg=38;5;13:*.gif=38;
*.pbm=38;5;13:*.pgm=38;5;13:*.ppm=38;5;13:*.tga=38;5;13:*.xbm=
13:*.tif=38;5;13:*.tiff=38;5;13:*.png=38;5;13:*.svg=38;5;13:*.
38;5;13:*.pcx=38;5;13:*.mov=38;5;13:*.mpg=38;5;13:*.mpeg=38;5;
nkx=38;5;13:*.webm=38;5;13:*.ogm=38;5;13:*.mp4=38;5;13:*.m4v=3
13:*.vob=38;5;13:*.qt=38;5;13:*.nuv=38;5;13:*.wmv=38;5;13:*.as
;13:*.rmvb=38;5;13:*.flc=38;5;13:*.avi=38;5;13:*.fli=38;5;13:*.
3;5;13:*.dl=38;5;13:*.xcf=38;5;13:*.xwd=38;5;13:*.yuv=38;5;13:
=38;5;13:*.axv=38;5;13:*.anx=38;5;13:*.ogv=38;5;13:*.ogx=38;5;
au=38;5;45:*.flac=38;5;45:*.mid=38;5;45:*.midi=38;5;45:*.mka=3
5:*.mpc=38;5;45:*.ogg=38;5;45:*.ra=38;5;45:*.wav=38;5;45:Ошибка
core dumped)
[guest@newlora ~]$ ./readfile /etc/shadow
```

Рис. 1.14.

## 2.2 Исследование Sticky-бита

1.Выяснили,установленли атрибут Sticky надиректории/tmp

2.Отимени пользователя guest создали файлfile01.txt в директории /tmp сословом test:

3.Просмотрели атрибуты у только то созданного файла и разрешили чтение и запись для категории пользователей «все остальные»:

```
l/guestPATH=/usr/лошибка сегментирования (core dumped)
[guest@newlora ~]$ ls -l / | grep tmp
drwxrwxrwt. 22 root root 4096 окт 7 19:59 tmp
[guest@newlora ~]$ echo "test" > /tmp/file01.txt
[guest@newlora ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт 7 20:14 /tmp/file01.txt
[guest@newlora ~]$ chmod o+rw /tmp/file01.txt
[guest@newlora ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт 7 20:14 /tmp/file01.txt
```

Рис. 2.1.

4.От пользователя guest2 (не являющегося владельцем) получилось прочитать файл

5.От пользователя guest2 получилось дозаписать в файл

6.Проверили содержимое файла командой

7.От пользователя guest2 попробовали /tmp/file01.txt слово test3,стерев при этом всю имеющуюся в файле информацию командой

Удалось выполнить операцию

8.Проверили содержимое файла командой

9.От пользователя guest2 попробовали удалить файл/tmp/file01.txt командой rm/tmp/file 0l.txt

```
[guest@newlora ~]$ su - guest2
Пароль:
Последний вход в систему: Пн сен 18 22:14:04 MSK 2023 на pts/2
[guest2@newlora ~]$ cat /tmp/file01.txt
test
[guest2@newlora ~]$ echo "test3" > /tmp/file01.txt
[guest2@newlora ~]$ cat /tmp/file01.txt
test3
[guest2@newlora ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Рис. 2.2.

Не Удалось удалить файл

10.Повысьли свои права до супер пользователя следующей командой su и выполнили после этого команду,снимающую атрибут t(Sticky-бит)с директории /tmp:

11.Покинули режим супер пользователя командой exit

12.Отпользователя guest2 проверули,что атрибут at у директории /tmp нет:

13.Повторили предыдущие шаги.Какие наблюдают сяиз мененияю

14.Удалось удалить файл от имени пользователя,не являющегося его владельцем .

15.Повысьли свои права до супер пользователя и верните атрибут t на директорию/tmp:

```
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@newlora ~]$ su - root
Пароль:
Последний вход в систему: Сб окт 7 19:59:03 MSK 2023 на pts/1
[root@newlora ~]# chmod -t /tmp
[root@newlora ~]# exit
logout
[guest2@newlora ~]$ ls -l / |grep tmp
drwxrwxrwx. 22 root root 4096 окт 7 20:20 tmp
[guest2@newlora ~]$ cat /tmp/file01.txt
test3
[guest2@newlora ~]$ rm /tmp/file01.txt
[guest2@newlora ~]$ su -
Пароль:
Последний вход в систему: Сб окт 7 20:20:11 MSK 2023 на pts/1
[root@newlora ~]# chmod +t /tmp
[root@newlora ~]# exit
logout
[guest2@newlora ~]$ cat /tmp/file01.txt
cat: /tmp/file01.txt: Нет такого файла или каталога
[guest2@newlora ~]$ ls -l / |grep tmp
```

Рис. 2.3.

### 3 Вывод

В ходе выполнения лабораторной работы мы рассмотрели работы механизма смены идентификатора процессов пользователей,а также влияние бита Sticky на запись и удаление файлов.