

Лабораторная работа № 5

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

Соболевский Денис Андреевич

Содержание

Цель работы	4
Задание	5
Теоретическое введение	6
Выполнение лабораторной работы	7
Выводы	13
Список литературы	14

Список иллюстраций

1	Выполнение команд ./simpleid и id	7
2	Создание и программы simpleid2	8
3	Установка новых атрибутов и смена владельца файла simpleid2	8
4	Использование команд ./simpleid2 и id	9
5	Выполнение команды ls -l simpleid2	9
6	Создание программы readfile.c	9
7	Изменение владельца и прав файла readfile.c	10
8	Проверка, что пользователь guest не может прочитать файл readfile.c . . .	10
9	Смена прав файла	10
10	Чтение файла readfile.c	10
11	Чтение файла /etc/shadow	10
12	Выполнение команды ls -l / grep tmp	11
13	Создание файла file01.txt	11
14	Чтение файла file01.txt	11
15	Дозапись в файл /tmp/file01.txt	12
16	Перезапись в файле /tmp/file01.txt	12
17	Удаление файла /tmp/file01.txt	12
18	Удаление атрибута t	12
19	Повторение предыдущих шагов	12
20	Повышение прав и возвращение атрибута	12

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.

Задание

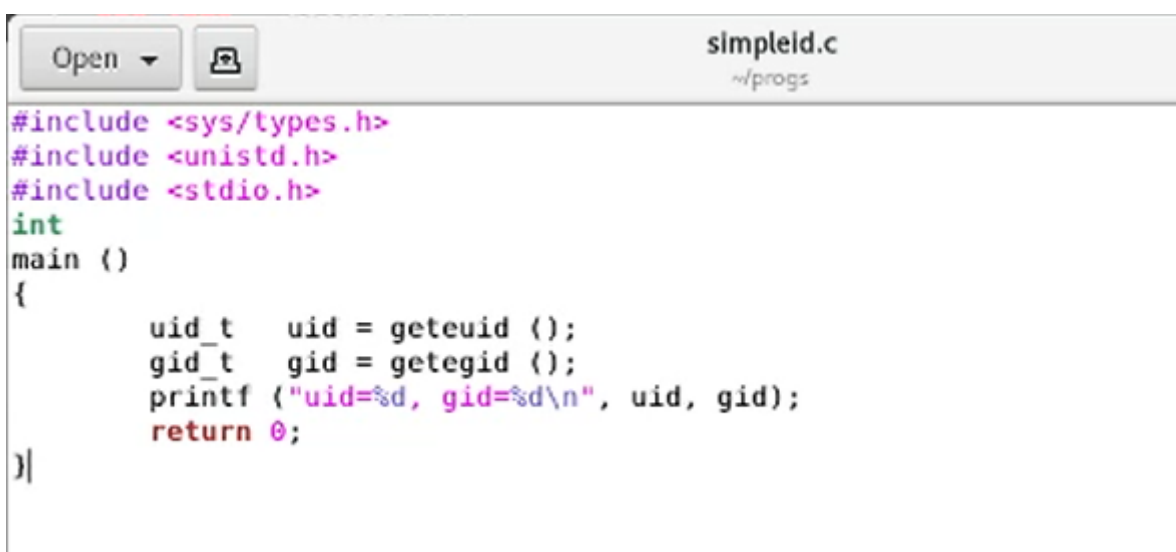
1. Исследовать влияние дополнительных атрибутов.
2. Исследовать Sticky-бит.

Теоретическое введение

- Операционная система — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.
- Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами. С помощью разрешений можно создать надежную среду — такую, в которой никто не может поменять содержимое ваших документов или повредить системные файлы.

Выполнение лабораторной работы

1. От имени пользователя guest создадим программу simpleid.c, скомпилируем ее и убедимся, что файл программы создан.



```
simpleid.c
~/progs

#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;
}
```

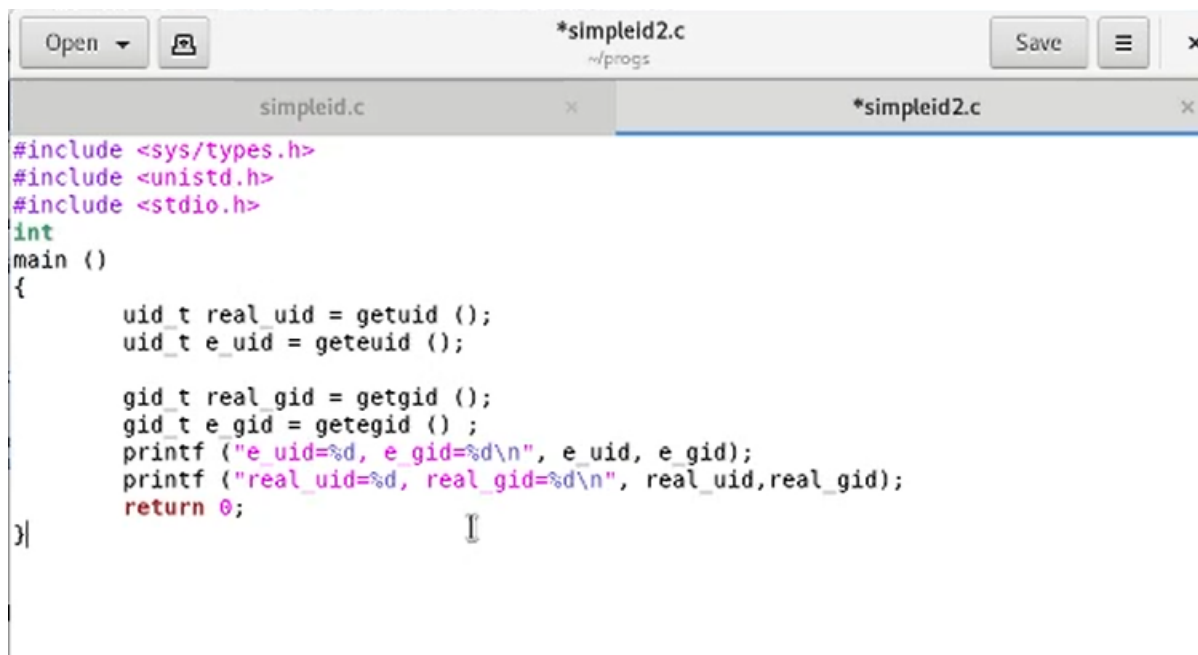
2. Выполним команды ./simpleid и id и убедимся, что полученные данные совпадают.



```
[guest@user progs]$ ./simpleid
uid=1001, gid=1001
[guest@user progs]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 1: ./simpleid id

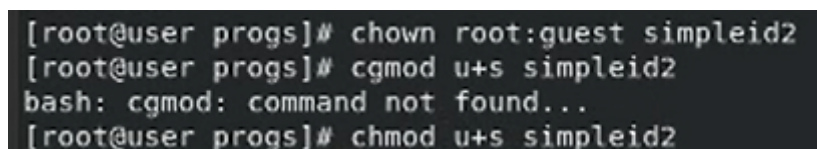
3. Усложним программу, добавив вывод действительных идентификаторов.



```
#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. 2: simpleid2



```
[root@user progs]# chown root:guest simpleid2
[root@user progs]# cgmod u+s simpleid2
bash: cgmod: command not found...
[root@user progs]# chmod u+s simpleid2
```

Рис. 3: simpleid2

4. От имени суперпользователя выполним команды.
5. Выполним команды `./simpleid2` и `id` и убедимся, что полученные данные совпадают.

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

Рис. 4: `./simpleid2 id`

6. Выполним проверку правильности установки новых атрибутов.

```
[root@user progs]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 13064 Oct  7 12:32 simpleid2
[root@user progs]# exit
```

Рис. 5: `ls -l simpleid2`

7. Создадим и скомпилируем программу `readfile.c`.

```
[guest@user progs]$ touch readfile.c
[guest@user progs]$
```

Рис. 6: `readfile.c`

8. Сменим владельца у файла `readfile.c` и изменим права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог.
9. Проверим, что пользователь `guest` не может прочитать файл `readfile.c`.
10. Сменим владельца и установим SetUID-бит.

```

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

```

Рис. 7: readfile.c

```

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

```

Рис. 8: , guest readfile.c

```

Password:
[root@user progs]# chown root:guest readfile
[root@user progs]# chmod u+s readfile

```

Рис. 9:

```

[guest@user progs]$ ./readfile readfile.c

```

Рис. 10: readfile.c

```

saslauth:!!:19609::::::
dnsmasq:!!:19609::::::
radvd:!!:19609::::::
sssd:!!:19609::::::
clevis:!!:19609::::::
cockpit-ws:!!:19609::::::
cockpit-wsinstance:!!:19609::::::
flatpak:!!:19609::::::
colord:!!:19609::::::
rpcuser:!!:19609::::::
gdm:!!:19609::::::
gnome-initial-setup:!!:19609::::::
pesign:!!:19609::::::
sshd:!!:19609::::::
rngd:!!:19609::::::
tcpdump:!!:19609::::::
dasobolevskiy:$6$sr2S3AypLds/Taj5$NbcWgUvi.vrDoHiTCXGZL5ffb6q9Q4IJZzoYfLLjFrL0TA
THiY8qIu2HFgiLFektEWhBt4hDzFcEBRztSnr8J.:19609:0:99999:7:::
vboxadd:!!:19609::::::
guest:$6$c3dRz0m0AeFS4wvY$cBlilhy2whTYTSsXs/grLkgzQLVabA2vVtZPB3TUtEthhi0gaN.bU.
kTNp8TWmY1znKQliaVMih.d0NQkR3Gj.:19616:0:99999:7:::
guest2:$6$stzI.j08NDvN2odV$Fv/b9VKuDEERewhJk4c9S5QpseDyFFnBs1D5yM905/986Xe4dicel
F9pjeWo8vTV2uzY4A/.U.mXI/oyVqW9/0:19623:0:99999:7:::
[guest@user progs]$

```

Рис. 11: /etc/shadow

```
[guest@user progs]$ ls -l / | grep tmp  
drwxrwxrwt. 19 root root 4096 Oct  7 12:43 tmp
```

Рис. 12: `ls -l / | grep tmp`

11. Проверим, может ли программа `readfile` прочитать файл `readfile.c`.
12. Проверим, может ли программа `readfile` прочитать файл `/etc/shadow`.
13. Выясним, установлен ли атрибут `Sticky` на директории `/tmp`.
14. От имени пользователя `guest` создадим файл `file01.txt` в директории `/tmp`.

```
[guest@user progs]$ echo "test" > /tmp/file01.txt
```

Рис. 13: `file01.txt`

15. От пользователя `guest2` попробуем прочитать файл `file01.txt`.

```
[guest@user progs]$ su guest2  
Password:  
[guest2@user progs]$ cat /tmp/file01.txt  
test
```

Рис. 14: `file01.txt`

16. От пользователя `guest2` попробуем дозаписать файл `file01.txt`.
17. От пользователя `guest2` попробуем записать в файл `file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию.
18. От пользователя `guest2` попробуем удалить файл `file01.txt`.
19. От имени суперпользователя снимем атрибут `t` с директории `/tmp`.
20. Повторим предыдущие шаги. Теперь файл удален успешно.
21. Повысим свои права до суперпользователя и вернем атрибут `t` на директорию `/tmp`.

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

Рис. 15: /tmp/file01.txt

```
test2  
[guest2@user progs]$ echo "test3" > /tmp/file01.txt  
[guest2@user progs]$ cat /tmp/file01.txt  
test3
```

Рис. 16: /tmp/file01.txt

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

Рис. 17: /tmp/file01.txt

```
[root@user progs]# chmod -t /tmp  
[root@user progs]# exit
```

Рис. 18: t

```
[guest2@user progs]$ ls -l / | grep tmp  
drwxrwxrwx. 20 root root 4096 Oct 7 12:47 tmp  
[guest2@user progs]$ cat /tmp/file01.txt  
test3  
[guest2@user progs]$ echo "test2" > /tmp/file01.txt  
[guest2@user progs]$ cat /tmp/file01.txt  
test2  
[guest2@user progs]$ echo "test3" > /tmp/file01.txt  
[guest2@user progs]$ cat /tmp/file01.txt  
test3  
[guest2@user progs]$ rm /tmp/file01.txt
```

Рис. 19:

```
[guest2@user progs]$ su  
Password:  
[root@user progs]# chmod +t
```

Рис. 20:

Выводы

В рамках данной лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрены принципы работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

[1] <https://codeby.school/blog/informacionnaya-bezopasnost/razgranichenie-dostupa-v-linux-znakomstvo-s-astra-linux>

[2] https://www.researchgate.net/profile/Dmitry-Kulyabov/publication/339290917_Informacionnaa_bezopasnost-komputernyh-setej-laboratornye-raboty.pdf