

Отчёт по лабораторной работе 5

Кочетов Андрей Владимирович

13 ноября, 2021

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.

Улучшить навыки работы с консолью и атрибутами.
Научиться писать программы и работать с ними.

Выполнение лабораторной работы

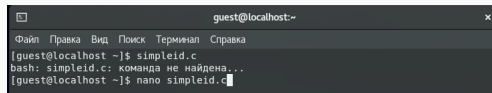
Выполнение лабораторной работы

1. Убедился, что программа установлена и выключил систему запретов(рис.1).

```
Модель многопоточности: posix
gcc версия 8.4.1 20200928 (Red Hat 8.4.1-1) (GCC)
[avkochetov@localhost ~]$ getenforce
Enforcing
[avkochetov@localhost ~]$ setenforce 0
setenforce: setenforce() failed
[avkochetov@localhost ~]$ getenforce
Enforcing
[avkochetov@localhost ~]$ setenforce 0
setenforce: setenforce() failed
[avkochetov@localhost ~]$ su
Пароль:
[root@localhost avkochetov]# setenforce 0
[root@localhost avkochetov]# getenforce
Permissive
[root@localhost avkochetov]#
```

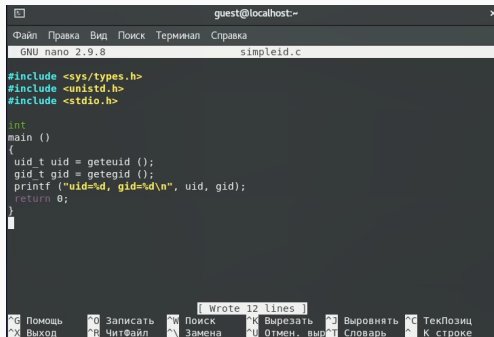
Figure 1: рис.1. Подготовка

2. Вошел в систему от guest и создал программу simpleid.c(рис.2-3).

A terminal window titled 'guest@localhost:~' with a menu bar containing 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows the following commands and output:

```
[guest@localhost ~]$ simpleid.c
bash: simpleid.c: команда не найдена...
[guest@localhost ~]$ nano simpleid.c
```

Figure 2: рис.2. Создание программы



```
guest@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
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;
}

[Wrote 12 lines]
^G Помощь      ^O Записать   ^W Поиск      ^K Вырезать   ^J Выворнять  ^C ТекПозиц
^X Выход      ^R ЧитФайл   ^N Замена     ^U Отмен. выр ^I Словарь    ^_ К строке
```

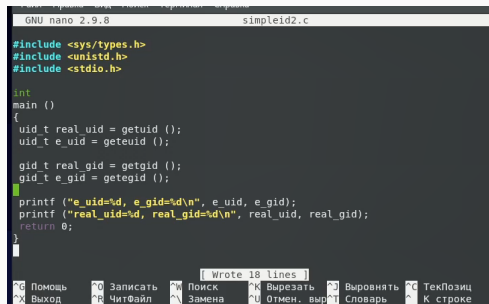
Figure 3: рис.3. Код программы

3. Скомпилировал программу, выполнил ее и выполнил системную программу id. Сравнил полученные результаты(рис.4).

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

Figure 4: рис.4. Выполнение программ

4. Усложнил программу и дал ей новое название(рис.5).



```
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;
}

[ Wrote 18 lines ]
⌘ Помощь  ⌘ Записать  ⌘ Поиск  ⌘ Вырезать  ⌘ Выводить  ⌘ ТекПозиц
⌘ Выход  ⌘ ЧитФайл  ⌘ Замена  ⌘ Отмен. выр  ⌘ Словарь  ⌘ К строке
```

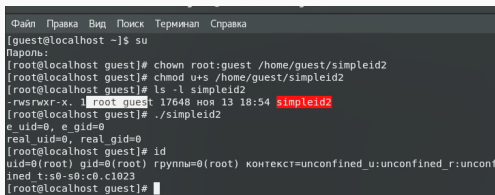
Figure 5: рис.5. Новая программа

5. Скомпилировал и запустил новую программу(рис.6).

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

Figure 6: рис.6. Снятие атрибута

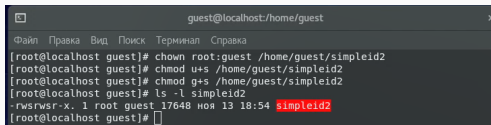
6. От root выполнил определенные команды, проверил проверку новых атрибутов и запустил программу. Сравнил результаты(рис.7).



```
Файл Правка Вид Поиск Терминал Справка
[guest@localhost ~]$ su
Пароль:
[root@localhost guest]# chown root:guest /home/guest/simpleid2
[root@localhost guest]# chmod u+s /home/guest/simpleid2
[root@localhost guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 13 18:54 simpleid2
[root@localhost guest]# ./simpleid2
e uid=0, e_gid=0
real uid=0, real_gid=0
[root@localhost guest]# id
uid=0(root) gid=0(root) rгруппы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost guest]#
```

Figure 7: рис.7. Выполнение команд

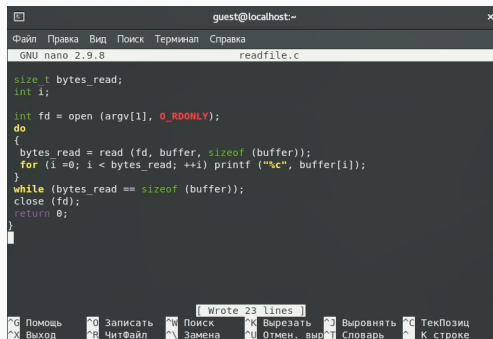
7. Проделал тоже самое относительно SetGID-бита(рис.8).



```
guest@localhost:/home/guest
Файл Правка Вид Поиск Терминал Справка
[root@localhost guest]# chown root:guest /home/guest/simpleid2
[root@localhost guest]# chmod u+s /home/guest/simpleid2
[root@localhost guest]# chmod g+s /home/guest/simpleid2
[root@localhost guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 17648 ноя 13 18:54 simpleid2
[root@localhost guest]#
```

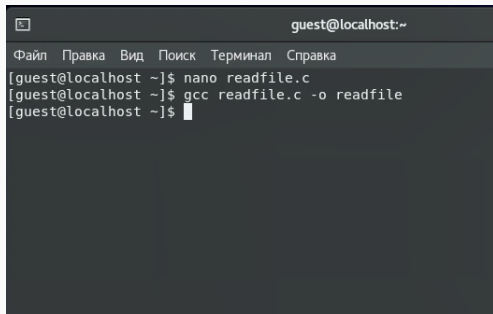
Figure 8: рис.8. Повторение

8. Создал еще одну программу с названием readfile.c и откомпилировал ее(рис.9-10).



```
guest@localhost:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
GNU nano 2.9.8  readfile.c  
  
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 23 lines  
^G Помощь  ^O Записать  ^W Поиск  ^K Вырезать  ^J Выводить  ^C ТекПозиц  
^X Выход  ^R ЧитФайл  ^\ Замена  ^U Отмен. выр  ^T Словарь  ^_ К строке
```

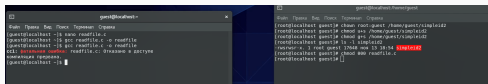
Figure 9: рис.9. readfile.c

A terminal window titled 'guest@localhost:~' with a menu bar containing 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows three lines of commands: '[guest@localhost ~]\$ nano readfile.c', '[guest@localhost ~]\$ gcc readfile.c -o readfile', and '[guest@localhost ~]\$' followed by a cursor.

```
guest@localhost:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[guest@localhost ~]$ nano readfile.c  
[guest@localhost ~]$ gcc readfile.c -o readfile  
[guest@localhost ~]$
```

Figure 10: рис.10. Компиляция

9. Сменил владельца у файла и изменил права. Убедился, что смена прошла успешно(рис.11).



The image shows two terminal windows side-by-side. The left window, titled 'guest@localhost: ~', shows the command 'ls -l /tmp/quest' being executed, resulting in a file 'readfile.c' with permissions '-rwxr-xr-x' and ownership 'quest:quest'. The right window, titled 'guest@localhost: /home/guest', shows the command 'ls -l /home/guest/readfile.c' being executed, resulting in a file 'readfile.c' with permissions '-rwxr-xr-x' and ownership 'quest:quest'. The output of the second command is highlighted in red.

```
guest@localhost: ~  
$ ls -l /tmp/quest  
-rwxr-xr-x 1 quest quest 10240 Aug 13 18:04 readfile.c  
quest@localhost: ~  
$
```

```
guest@localhost: /home/guest  
$ ls -l /home/guest/readfile.c  
-rwxr-xr-x 1 quest quest 10240 Aug 13 18:04 readfile.c  
quest@localhost: /home/guest  
$
```

Figure 11: рис.11. Права

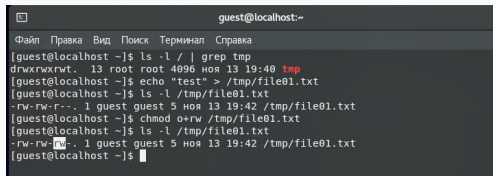
10. Сменил владельца и поставил SetU'D-бит. Таким образом смог прочитать файл и прочитать shadow(рис.12-13).

```
[guest@localhost ~]$ nano readfile.c
[guest@localhost ~]$ gcc readfile.c -o readfile
[guest@localhost ~]$ gcc readfile.c -o readfile
cc1: фатальная ошибка: readfile.c: Отказано в доступе
компиляция прервана.
[guest@localhost ~]$ chmod 777 readfile.c
[guest@localhost ~]$ chmod +s readfile.c
[guest@localhost ~]$ gcc readfile.c -o readfile
[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;
```

Figure 12: рис.12. Настройка

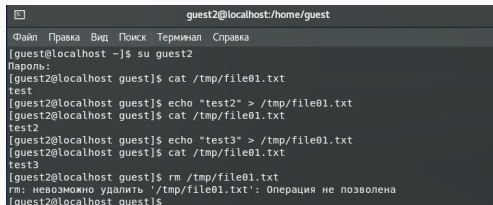
11. Проверил атрибут Sticky, создал файл file01.txt со словом, посмотрел атрибуты и разрешил запись для всех остальных пользователей(рис.14).

A terminal window titled 'guest@localhost:~' with a menu bar containing 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows the following commands and output:

```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 13 root root 4096 ноя 13 19:40 tmp
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 19:42 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 19:42 /tmp/file01.txt
[guest@localhost ~]$
```

Figure 14: рис.14. Работа с файлами

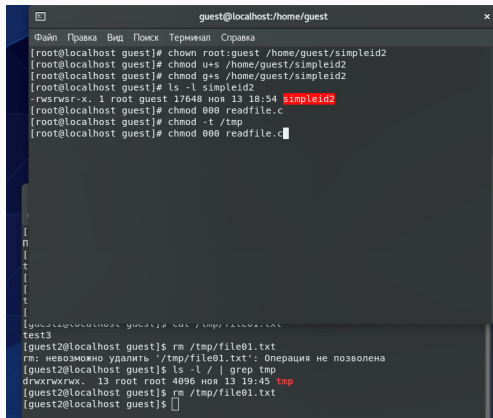
12. Выполнил ряд команд от пользователя Guest2. Не удалось выполнить команду rm(рис.15).



```
guest2@localhost:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@localhost ~]$ su guest2
Пароль:
[guest2@localhost guest]$ cat /tmp/file01.txt
test
[guest2@localhost guest]$ echo "test2" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test2
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@localhost guest]$
```

Figure 15: рис.15. Guest2

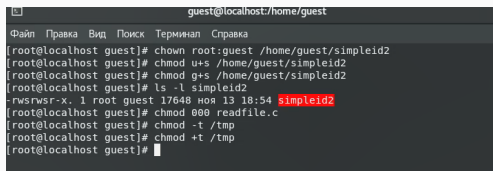
13. Снял атрибут t, проверил, что атрибут снят, и успешно выполнил программу rm(рис.16).



```
guest@localhost:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
[root@localhost guest]# chown root:guest /home/guest/simpleid2
[root@localhost guest]# chmod u+s /home/guest/simpleid2
[root@localhost guest]# chmod g+s /home/guest/simpleid2
[root@localhost guest]# ls -l simpleid2
-rwxrwxr-x. 1 root guest 17648 ноя 13 18:54 simpleid2
[root@localhost guest]# chmod 000 readfile.c
[root@localhost guest]# chmod -t /tmp
[root@localhost guest]# chmod 000 readfile.c
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@localhost guest]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 ноя 13 19:45 tmp
[guest2@localhost guest]$ rm /tmp/file01.txt
[guest2@localhost guest]$
```

Figure 16: рис.16. Выполнил команду

14. Вернул атрибут t(рис.17).



```
guest@localhost:/home/guest
Файл Правка Вид Поиск Терминал Справка
[root@localhost guest]# chown root:guest /home/guest/simpleid2
[root@localhost guest]# chmod u+s /home/guest/simpleid2
[root@localhost guest]# chmod g+s /home/guest/simpleid2
[root@localhost guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 17648 ноя 13 18:54 simpleid2
[root@localhost guest]# chmod 000 readfile.c
[root@localhost guest]# chmod -t /tmp
[root@localhost guest]# chmod +t /tmp
[root@localhost guest]#
```

Figure 17: рис.17. Возвращение атрибута

Выводы

Изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Спасибо за внимание