

Отчёт по лабораторной работе №5

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

Морозов Михаил Евгеньевич

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	9
	Список литературы	10

Список иллюстраций

3.1	Результат 1	7
3.2	Результат 2	8
3.3	Результат 3	8

Список таблиц

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 Теоретическое введение

В настоящее время sticky bit используется в основном для каталогов, чтобы защитить в них файлы. Из такого каталога пользователь может удалить только те файлы, владельцем которых он является. Примером может служить каталог /tmp, в который запись открыта для всех пользователей, но нежелательно удаление чужих файлов. Установка атрибута производится утилитой `chmod`. [1]

3 Выполнение лабораторной работы

Создали файл `simple.id` и записали в него код из лабораторной. После запуска получили `uid` и `gid` нашего пользователя. Усложним скрипт, добавив вывод `real uid` и `gid`. Теперь выводятся и `real uid` и `gid`, все совпадает с результатами предыдущих шагов. Пропишем `chown` и `chmod`. `chown` изменяет владельца файла, а `chmod u+s` позволяет запускать файл с правами владельца. Теперь при запуске файла от имени `guest` получаем `e_uid root`. Прделаем то же самое с `SetGID`-битом. Вывод такой же. Создадим файл `readfile.c` как в лабораторной и скомпилируем его. Меняем владельца на `root` и забираем все права у всех кроме владельца. Проверяем. `guest` не может прочесть `readfile.c`. Попробуем прочитать `readfile.c` с помощью `readfile`.

Результаты работы 1 (рис. 3.1)

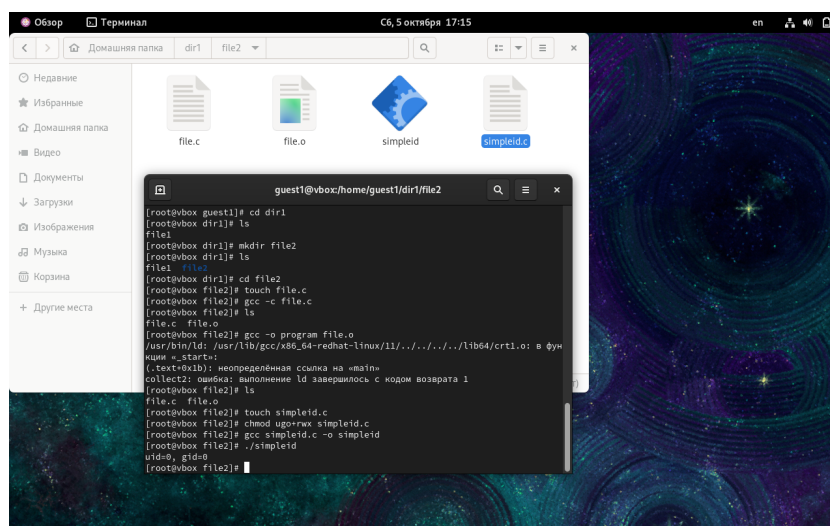
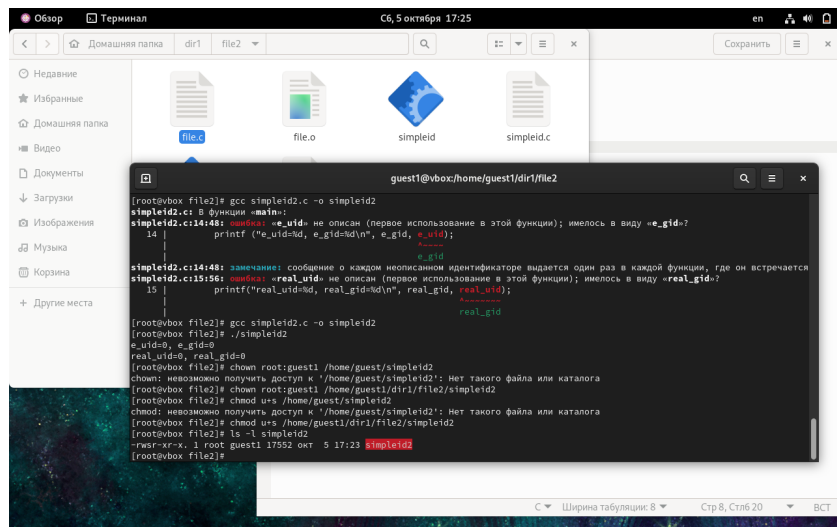


Рис. 3.1: Результат 1

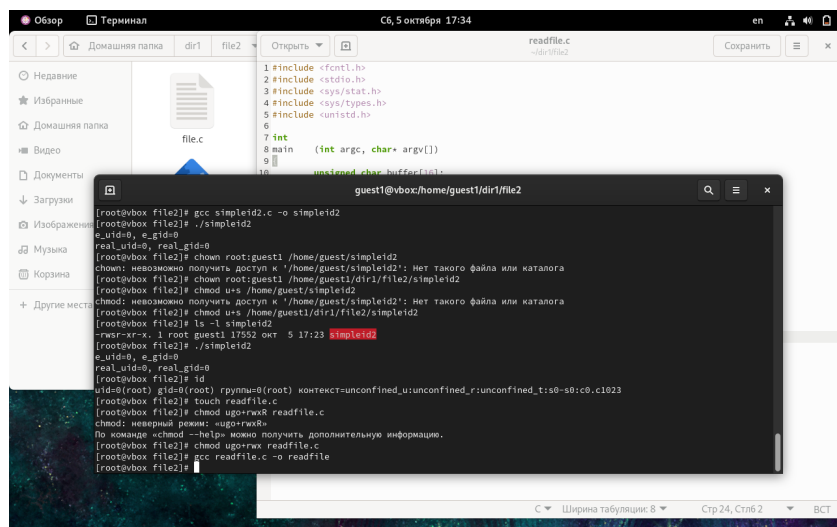
Результаты работы 2 (рис. 3.2)



```
guest1@vbox:/home/guest1/dir1/file2
[root@vbox file2]# gcc simpleid2.c -o simpleid2
simpleid2.c:14:48: warning: 'e_gid' не описан (первое использование в этой функции); имелось в виду 'e_uid'?
14 |         printf("e_uid=%d, e_gid=%d\n", e_gid, e_gid);
    |                                                ^~~~~~
simpleid2.c:14:48: warning: 'e_gid' не описан (первое использование в этой функции); имелось в виду 'e_uid'?
15 |         printf("real_uid=%d, real_gid=%d\n", real_gid, real_gid);
    |                                                ^~~~~~
    |
    = real_gid
[root@vbox file2]# gcc simpleid2.c -o simpleid2
[root@vbox file2]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@vbox file2]# chown root:guest1 /home/guest1/simpleid2
chown: невозможно получить доступ к '/home/guest1/simpleid2': Нет такого файла или каталога
[root@vbox file2]# chown root:guest1 /home/guest1/dir1/file2/simpleid2
[root@vbox file2]# chmod u+s /home/guest1/simpleid2
chmod: невозможно получить доступ к '/home/guest1/simpleid2': Нет такого файла или каталога
[root@vbox file2]# chmod u+s /home/guest1/dir1/file2/simpleid2
[root@vbox file2]# ls -l simpleid2
-rwxr-xr-x. 1 root guest1 17552 окт  5 17:23 simpleid2
[root@vbox file2]#
```

Рис. 3.2: Результат 2

Результаты работы 3 (рис. 3.3)



```
guest1@vbox:/home/guest1/dir1/file2
[root@vbox file2]# gcc simpleid2.c -o simpleid2
[root@vbox file2]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@vbox file2]# chown root:guest1 /home/guest1/simpleid2
chown: невозможно получить доступ к '/home/guest1/simpleid2': Нет такого файла или каталога
[root@vbox file2]# chown root:guest1 /home/guest1/dir1/file2/simpleid2
[root@vbox file2]# chmod u+s /home/guest1/simpleid2
chmod: невозможно получить доступ к '/home/guest1/simpleid2': Нет такого файла или каталога
[root@vbox file2]# chmod u+s /home/guest1/dir1/file2/simpleid2
[root@vbox file2]# ls -l simpleid2
-rwxr-xr-x. 1 root guest1 17552 окт  5 17:23 simpleid2
[root@vbox file2]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@vbox file2]# id
uid=0(root) gid=0(root) rpnpn=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@vbox file2]# touch readfile.c
[root@vbox file2]# chmod ugo+rwx readfile.c
chmod: неверный режим: ugo+rwxR
По команде 'chmod --help' можно получить дополнительную информацию.
[root@vbox file2]# chmod ugo+rwx readfile.c
[root@vbox file2]# gcc readfile.c -o readfile
[root@vbox file2]#
```

Рис. 3.3: Результат 3

4 Выводы

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

Список литературы

1. Sticky bit [Электронный ресурс]. Wikimedia Foundation, 2024. URL: https://ru.wikipedia.org/wiki/Sticky_bit.