

Отчёт по лабораторной работе №7

дисциплина: Информационная безопасность

Никитаева Александра Семеновна, НПИбд-02-18

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	21
	Список литературы	22

List of Tables

List of Figures

2.1	Программа <i>simpleid.c</i>	6
2.2	Компиляция и выполнение программы <i>simpleid</i>	7
2.3	Программа <i>simpleid2.c</i>	8
2.4	Компиляция и выполнение программы <i>simpleid2</i>	9
2.5	Смена пользователя. Установка SetUID-бита. Выполнение программы <i>simpleid2</i>	10
2.6	Установка SetGID-бита. Выполнение программы <i>simpleid2</i>	11
2.7	Программа <i>readfile.c</i>	11
2.8	Работа с программой <i>readfile.c</i>	12
2.9	Запрет на чтение программы <i>readfile.c</i> для <i>guest</i>	13
2.10	Установка SetUID-бита на программу <i>readfile</i>	13
2.11	Программа <i>readfile</i> читает <i>readfile.c</i>	14
2.12	Программа <i>readfile</i> читает <i>/etc/shadow</i>	15
2.13	Исследование Sticky-бита от имени <i>guest</i>	16
2.14	Работа с <i>file01.txt</i> от имени <i>guest2</i> при наличии Sticky-бита	17
2.15	Снятие Sticky-бита с <i>/tmp</i>	18
2.16	Работа с <i>file01.txt</i> от имени <i>guest2</i> без Sticky-бита	19
2.17	Возвращение Sticky-бита на <i>/tmp</i>	20

1 Цель работы

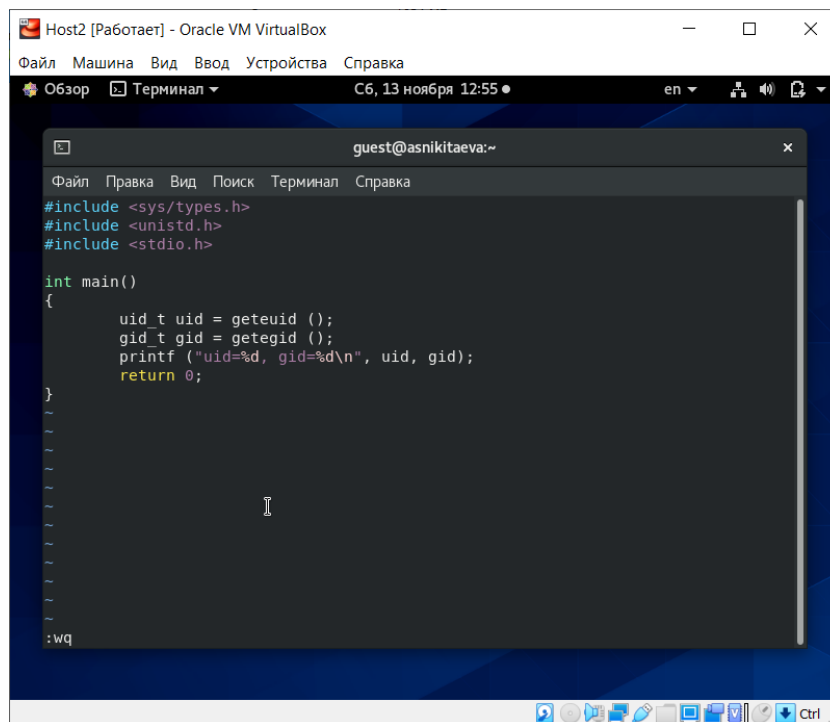
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Выполнение лабораторной работы

1. Создание программы

1.1. Вошла в систему от имени пользователя guest.

1.2. Создала программу *simpleid.c* по шаблону из методички. (рис. 2.1)



The screenshot shows a terminal window titled 'guest@asnikitaeva:~' within an Oracle VM VirtualBox environment. The terminal displays the following C code for the *simpleid.c* program:

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

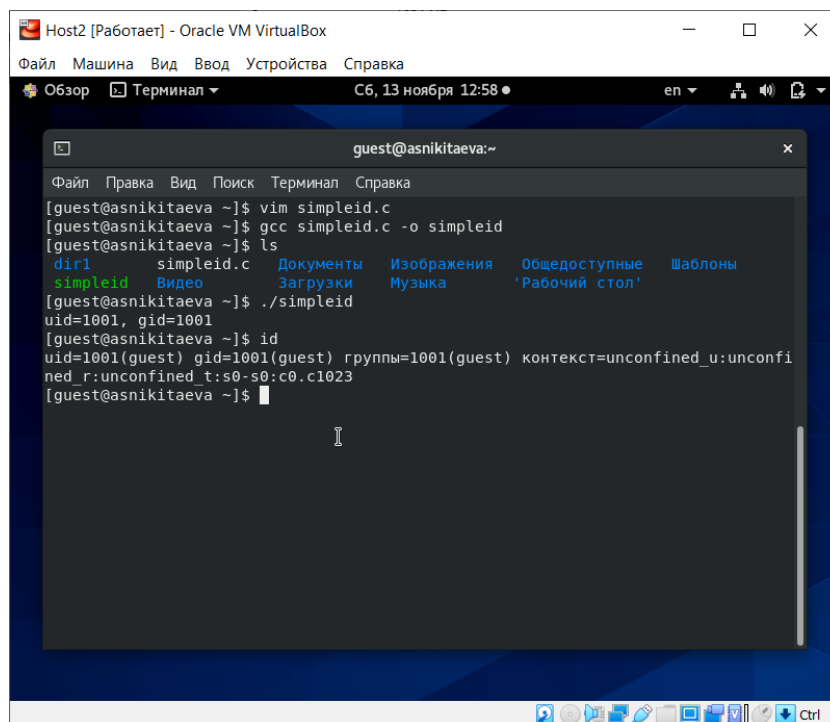
int main()
{
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Figure 2.1: Программа *simpleid.c*

1.3. Скомпилировала программу и убедилась, что файл программы создан:
`gcc simpleid.c -o simpleid.` (рис. 2.2)

1.4. Выполнила программу *simpleid*: `./simpleid.` (рис. 2.2)

1.5. Выполнила системную программу id: id. (рис. 2.2) Полученный мной результат совпадает с данными предыдущего пункта задания.

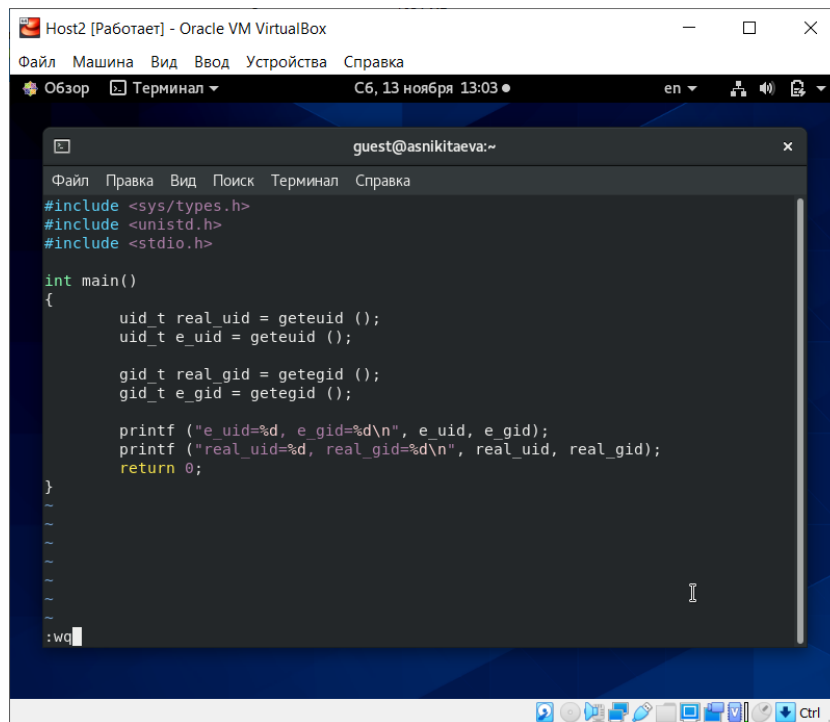


```
Host2 [Работаer] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Обзор  Терминал  C6, 13 ноября 12:58  en  [иконки]

guest@asnikitaeva:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@asnikitaeva ~]$ vim simpleid.c
[guest@asnikitaeva ~]$ gcc simpleid.c -o simpleid
[guest@asnikitaeva ~]$ ls
dir1  simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
simpleid  Видео  Загрузки  Музыка  'Рабочий стол'
[guest@asnikitaeva ~]$ ./simpleid
uid=1001, gid=1001
[guest@asnikitaeva ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@asnikitaeva ~]$
```

Figure 2.2: Компиляция и выполнение программы simpleid

1.6. Усложнила программу, добавив вывод действительных идентификаторов согласно шаблону из методички. Получившуюся программу назвала simpleid2.c. (рис. 2.3)



The image shows a screenshot of a terminal window within an Oracle VM VirtualBox environment. The window title is "Host2 [Работает] - Oracle VM VirtualBox". The terminal interface has a menu bar with "Файл", "Правка", "Вид", "Поиск", "Терминал", and "Справка". The terminal prompt is "guest@asnikitaeva:~". The code displayed in the terminal is as follows:

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int main()
{
    uid_t real_uid = geteuid ();
    uid_t e_uid = geteuid ();

    gid_t real_gid = getegid ();
    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;
}
```

Figure 2.3: Программа *simpleid2.c*

1.7. Скомпилировала и запустила *simpleid2.c*: `gcc simpleid2.c -o simpleid2` и `./simpleid2`. (рис. 2.4)

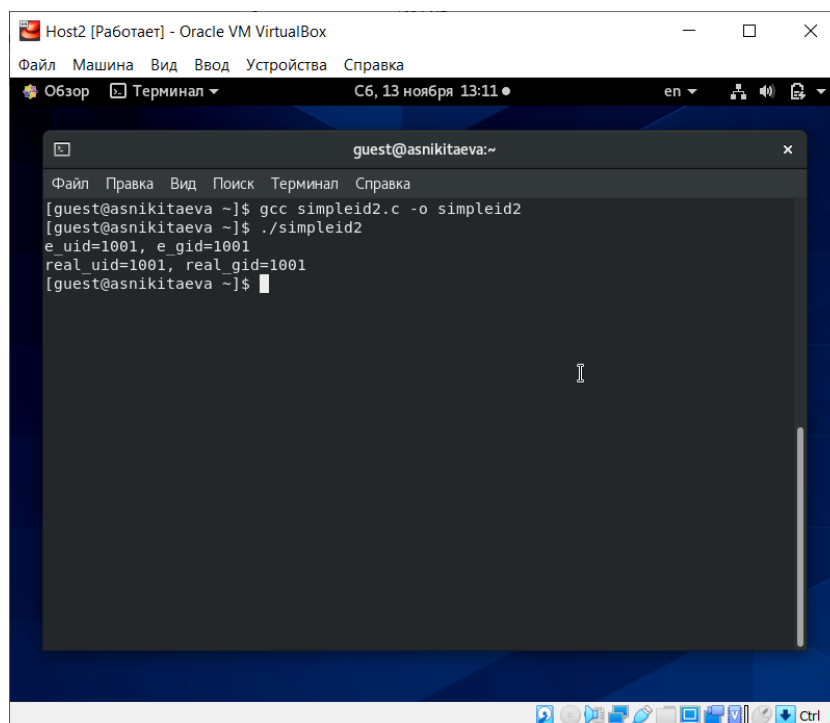


Figure 2.4: Компиляция и выполнение программы simpleid2

1.8. От имени суперпользователя выполнила команды: `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`. (рис. 2.5)

1.9. Повысила временно свои права с помощью `su`. (рис. 2.5) Первая команда меняет владельца файла, а вторая добавляет SetUID-бит.

1.10. Выполнила проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`: `ls -l simpleid2`. (рис. 2.5)

1.11. Запустила `simpleid2` и `id`: `./simpleid2` и `id`. (рис. 2.5)

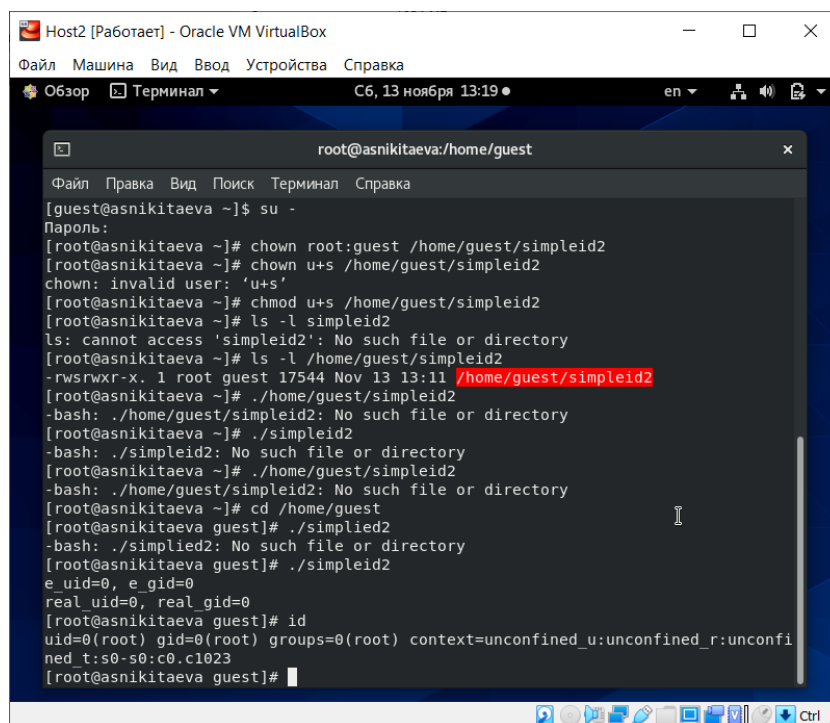


Figure 2.5: Смена пользователя. Установка SetUID-бита. Выполнение программы simpleid2

1.12. Проделала то же самое относительно SetGID-бита. (рис. 2.6)

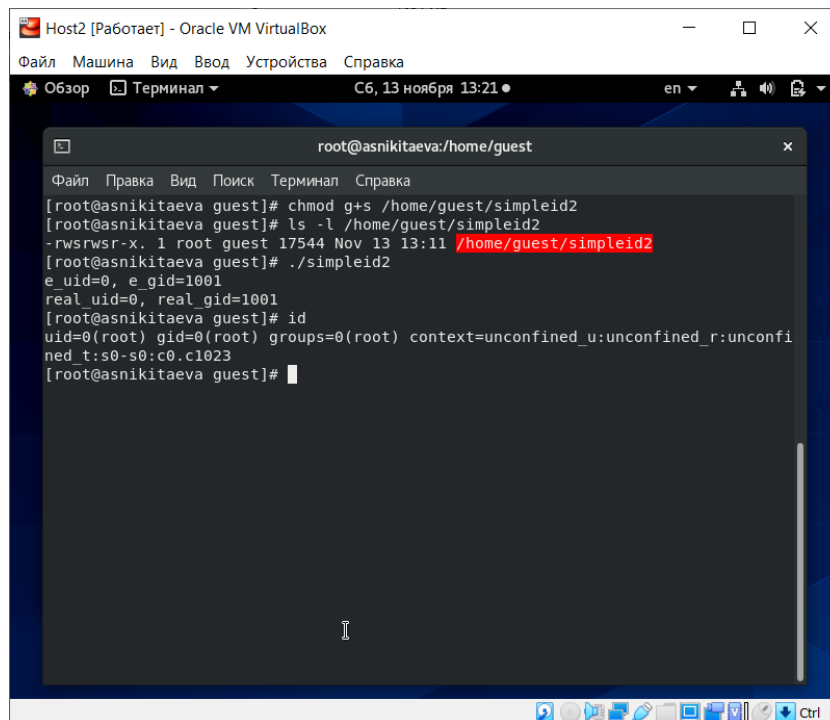


Figure 2.6: Установка SetGID-бита. Выполнение программы simpleid2

1.13. Создала программу readfile.c по шаблону из методички. (рис. 2.7)

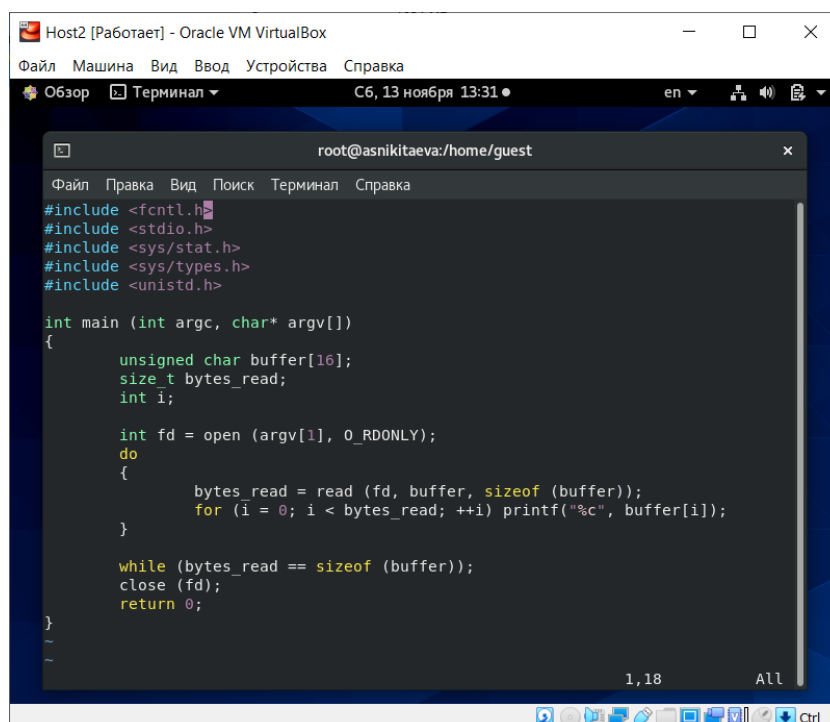
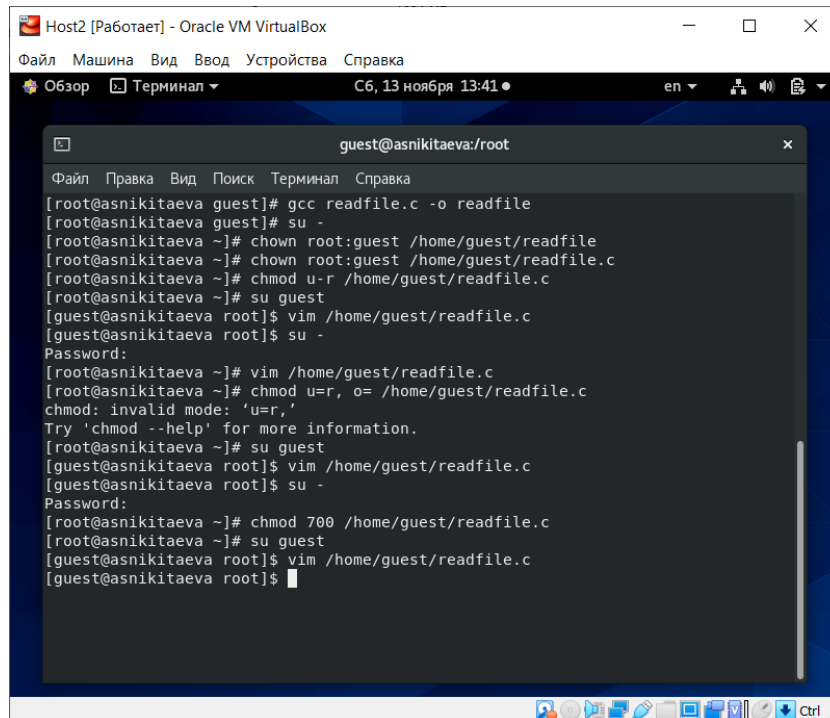


Figure 2.7: Программа *readfile.c*

1.14. Откомпилировала её: `gcc readfile.c -o readfile`. (рис. 2.8)

1.15. Сменила владельца у файла `readfile.c` и изменила права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог. (рис. 2.8)



```
Host2 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Обзор  Терминал  Сб, 13 ноября 13:41  en  [иконки]

guest@asnikitaeva:/root
Файл  Правка  Вид  Поиск  Терминал  Справка

[root@asnikitaeva guest]# gcc readfile.c -o readfile
[root@asnikitaeva guest]# su -
[root@asnikitaeva ~]# chown root:guest /home/guest/readfile
[root@asnikitaeva ~]# chown root:guest /home/guest/readfile.c
[root@asnikitaeva ~]# chmod u-r /home/guest/readfile.c
[root@asnikitaeva ~]# su guest
[guest@asnikitaeva root]$ vim /home/guest/readfile.c
[guest@asnikitaeva root]$ su -
Password:
[root@asnikitaeva ~]# vim /home/guest/readfile.c
[root@asnikitaeva ~]# chmod u=r, o= /home/guest/readfile.c
chmod: invalid mode: 'u=r,'
Try 'chmod --help' for more information.
[root@asnikitaeva ~]# su guest
[guest@asnikitaeva root]$ vim /home/guest/readfile.c
[guest@asnikitaeva root]$ su -
Password:
[root@asnikitaeva ~]# chmod 700 /home/guest/readfile.c
[root@asnikitaeva ~]# su guest
[guest@asnikitaeva root]$ vim /home/guest/readfile.c
[guest@asnikitaeva root]$
```

Figure 2.8: Работа с программой *readfile.c*

1.16. Проверила, что пользователь `guest` не может прочитать файл `readfile.c`. (рис. 2.9)

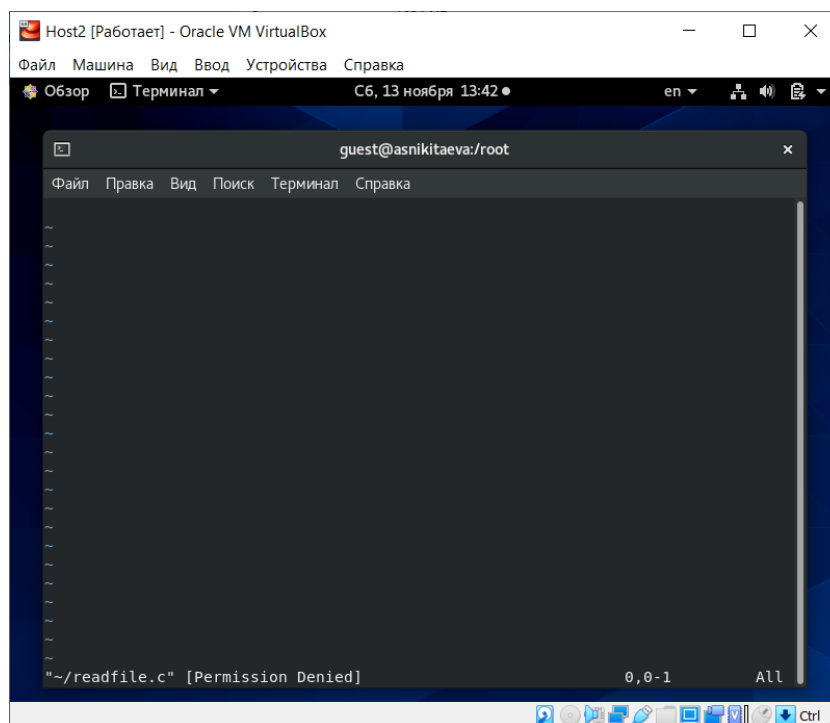


Figure 2.9: Запрет на чтение программы *readfile.c* для guest

1.17. Сменила у программы *readfile* владельца (рис. 2.8) и установила SetUID-бит (рис. 2.10).

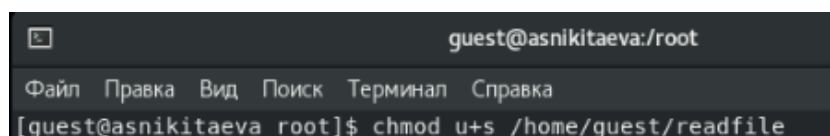


Figure 2.10: Установка SetUID-бита на программу *readfile*

1.18. Проверила, может ли программа *readfile* прочитать файл *readfile.c*. (рис. 2.11)

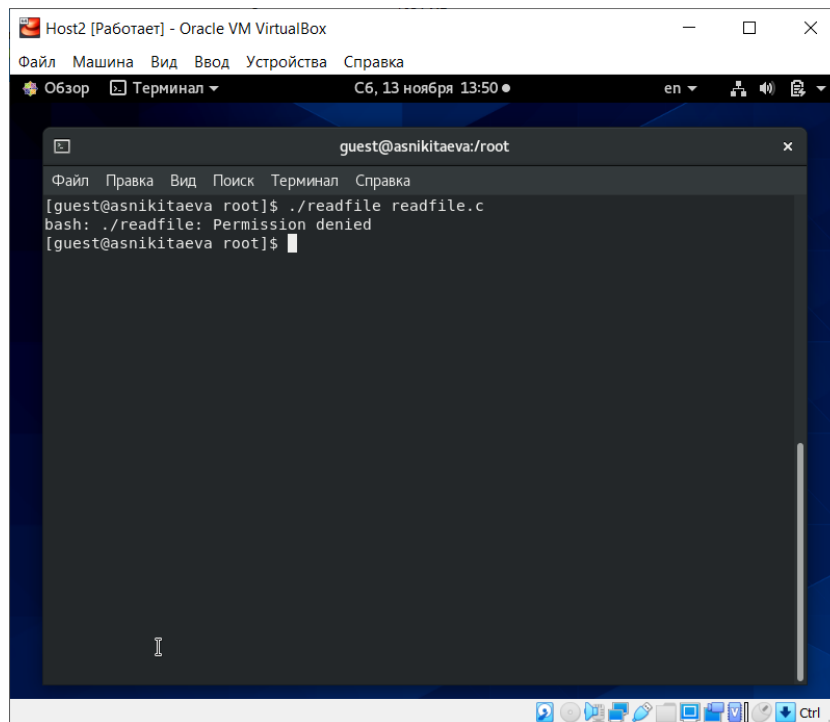


Figure 2.11: Программа readfile читает *readfile.c*

1.19. Проверила, может ли программа readfile прочитать файл */etc/shadow*. (рис. 2.12)

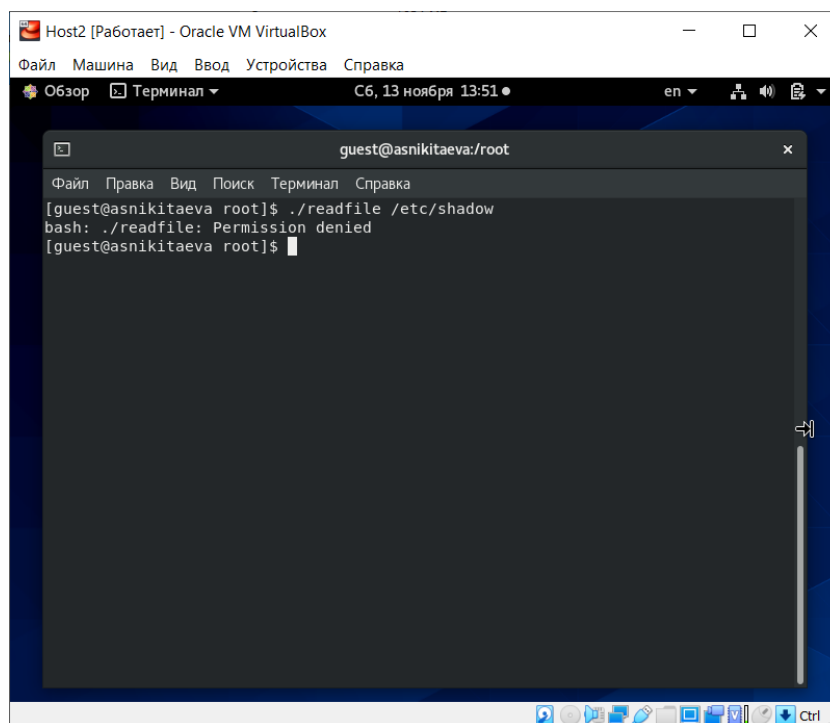


Figure 2.12: Программа readfile читает */etc/shadow*

2. Исследование Sticky-бита

2.1. Выяснила, установлен ли атрибут Sticky на директории */tmp*, для чего выполнила команду: `ls -l / | grep tmp`. (рис. 2.13)

2.2. От имени пользователя *guest* создала файл *file01.txt* в директории */tmp* со словом *test*: `echo "test" > /tmp/file01.txt`. (рис. 2.13)

2.3. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt, chmod o+rw /tmp/file01.txt и ls -l /tmp/file01.txt`. (рис. 2.13)

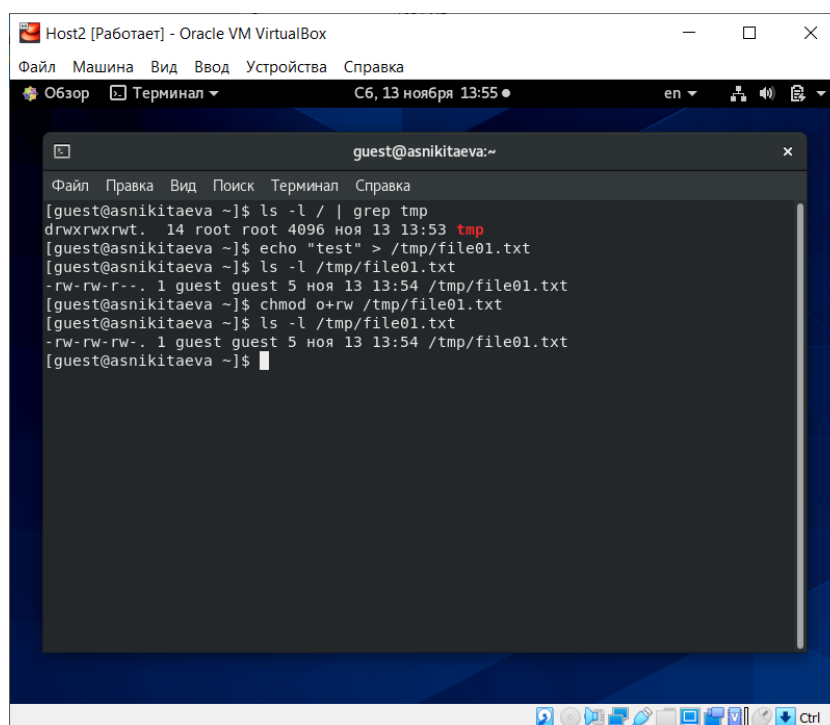


Figure 2.13: Исследование Sticky-бита от имени guest

2.4. От пользователя guest2 (не являющегося владельцем) попробовала прочитывать файл `/tmp/file01.txt`: `cat /tmp/file01.txt`. (рис. 2.14)

2.5. От пользователя guest2 попробовала дозаписать в файл `/tmp/file01.txt` слово `test2` командой: `echo "test2" >> /tmp/file01.txt`. (рис. 2.14) Операция прошла успешно.

2.6. Проверила содержимое файла командой: `cat /tmp/file01.txt`. (рис. 2.14)

2.7. От пользователя guest2 попробовала записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию командой: `echo "test3" > /tmp/file01.txt`. (рис. 2.14) Операция прошла успешно.

2.8. Проверила содержимое файла командой: `cat /tmp/file01.txt`. (рис. 2.14)

2.9. От пользователя guest2 попробовала удалить файл `/tmp/file01.txt` командой: `rm /tmp/file01.txt`. (рис. 2.14) Операция была не позволена.

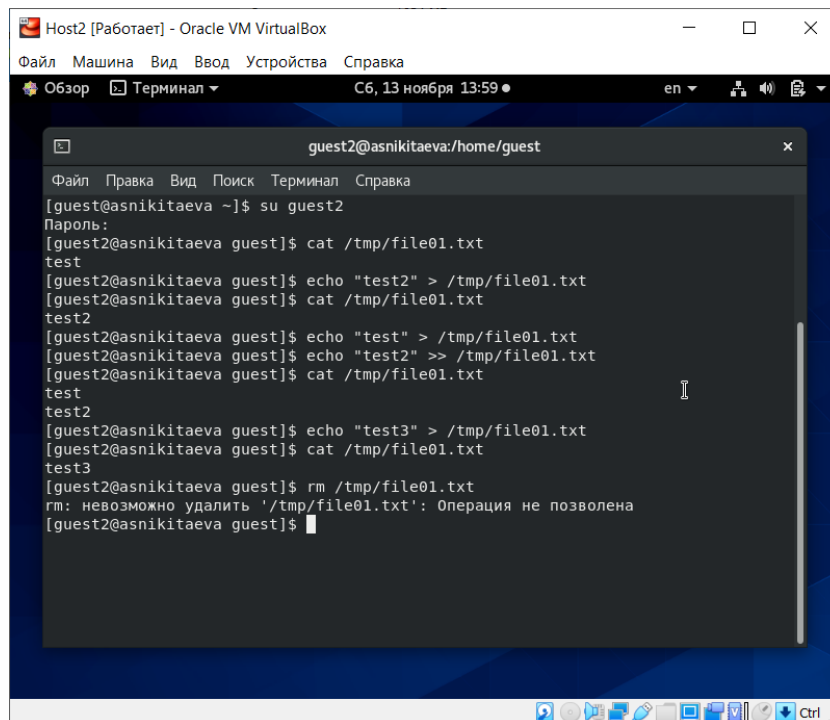


Figure 2.14: Работа с *file01.txt* от имени *guest2* при наличии Sticky-бита

2.10. Повысила свои права до суперпользователя следующей командой: `su -`, и выполнила после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`. (рис. 2.15)

2.11. Покинула режим суперпользователя командой: `exit`. (рис. 2.15)

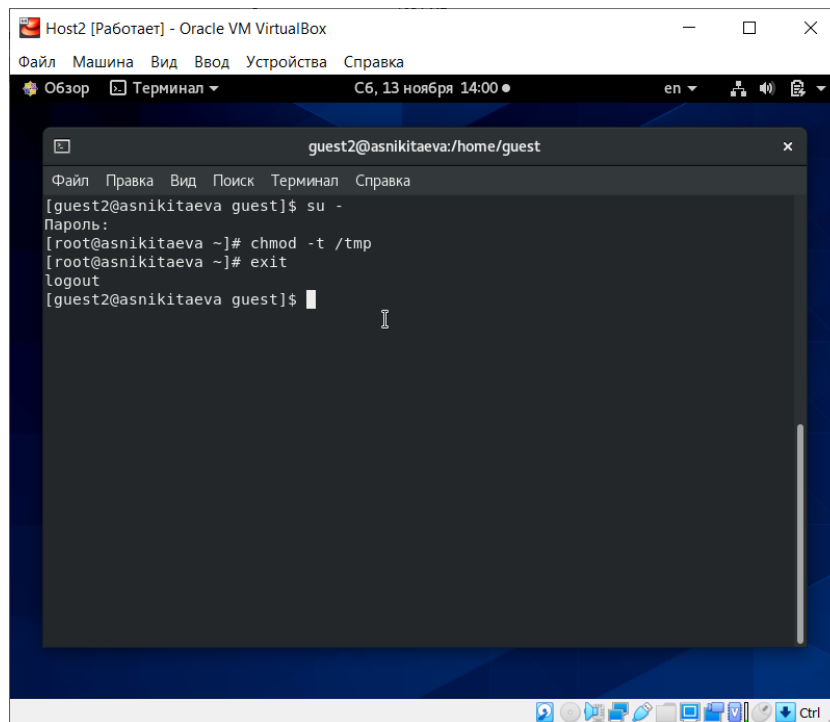
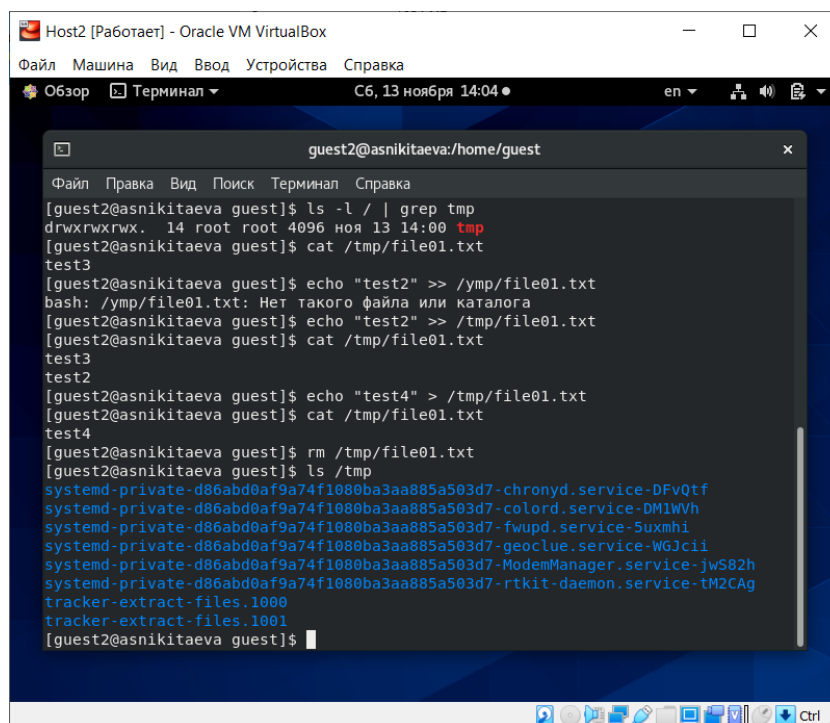


Figure 2.15: Снятие Sticky-бита с */tmp*

- 2.12. От пользователя *guest2* проверила, что атрибута *t* у директории */tmp* нет:
`ls -l / | grep tmp`. (рис. 2.16)
- 2.13. Повторила предыдущие шаги. (рис. 2.16) Теперь удалось удалить файл.



```
Host2 [Работаet] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Обзор Терминал C6, 13 ноября 14:04 en
guest2@asnikitaeva:/home/guest
Файл Правка Вид Поиск Терминал Справка
[guest2@asnikitaeva guest]$ ls -l / | grep tmp
drwxrwxrwx. 14 root root 4096 ноя 13 14:00 tmp
[guest2@asnikitaeva guest]$ cat /tmp/file01.txt
test3
[guest2@asnikitaeva guest]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Нет такого файла или каталога
[guest2@asnikitaeva guest]$ echo "test2" >> /tmp/file01.txt
[guest2@asnikitaeva guest]$ cat /tmp/file01.txt
test3
test2
[guest2@asnikitaeva guest]$ echo "test4" > /tmp/file01.txt
[guest2@asnikitaeva guest]$ cat /tmp/file01.txt
test4
[guest2@asnikitaeva guest]$ rm /tmp/file01.txt
[guest2@asnikitaeva guest]$ ls /tmp
systemd-private-d86abd0af9a74f1080ba3aa885a503d7-chrond.service-DFvQtf
systemd-private-d86abd0af9a74f1080ba3aa885a503d7-colord.service-DM1WVh
systemd-private-d86abd0af9a74f1080ba3aa885a503d7-fwupd.service-5uxmhi
systemd-private-d86abd0af9a74f1080ba3aa885a503d7-geoclue.service-W6Jcii
systemd-private-d86abd0af9a74f1080ba3aa885a503d7-ModemManager.service-jwS82h
systemd-private-d86abd0af9a74f1080ba3aa885a503d7-rtkit-daemon.service-tM2CAg
tracker-extract-files.1000
tracker-extract-files.1001
[guest2@asnikitaeva guest]$
```

Figure 2.16: Работа с *file01.txt* от имени guest2 без Sticky-бита

2.14. Да, мне удалось удалить файл от имени пользователя, не являющегося его владельцем.

2.15. Повысила свои права до суперпользователя и вернула атрибут *t* на директорию */tmp*: `su -, chmod +t /tmp` и `exit`. (рис. 2.17)

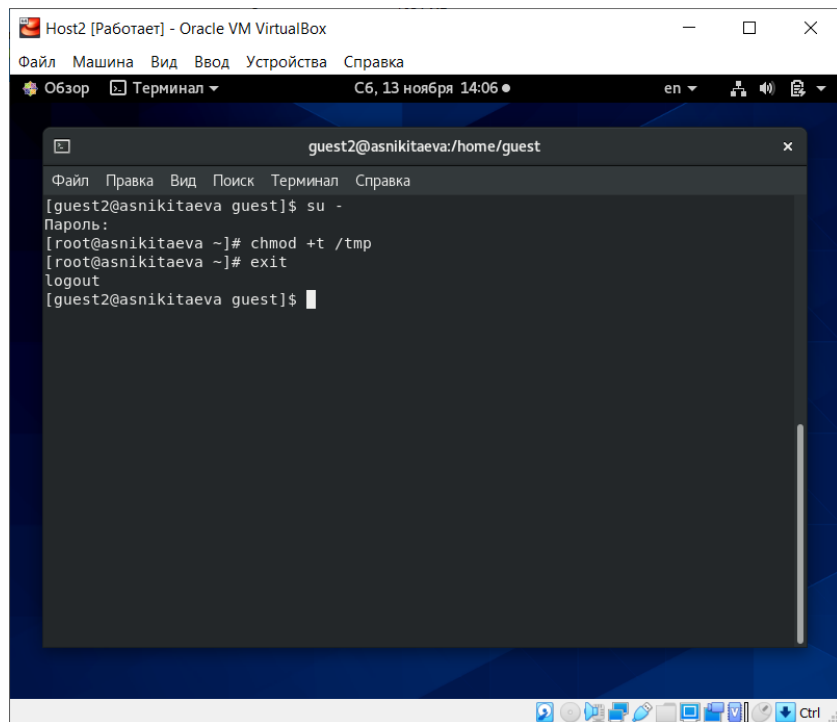


Figure 2.17: Возвращение Sticky-бита на */tmp*

3 Выводы

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

Список литературы

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Информационная безопасность компьютерных сетей. Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов