

Лабораторная работа №5

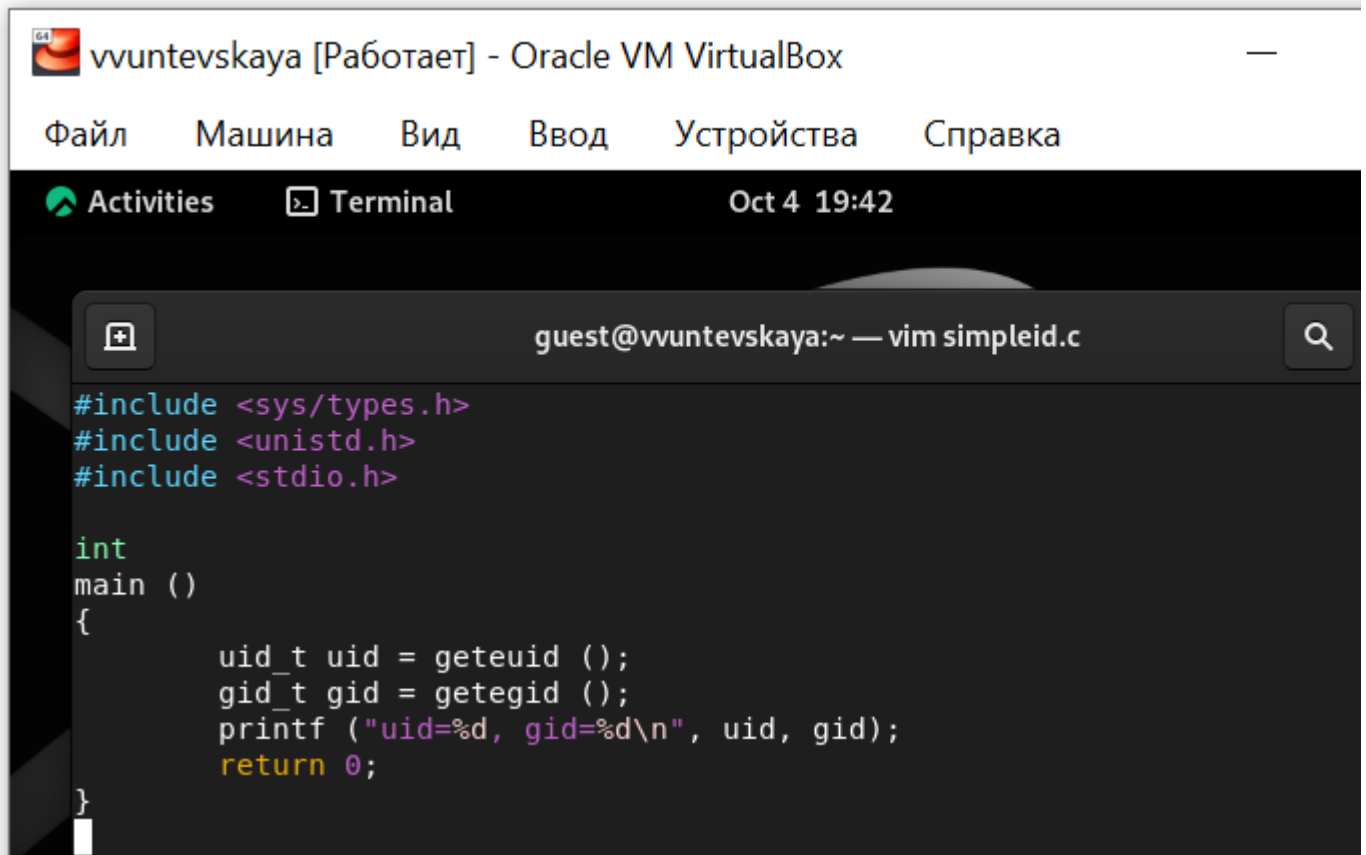
Унтевская Валерия НПИбд-02-19

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.

Выполнение лабораторной работы

1. Вошла в систему от имени пользователя guest и создала программу simpleid.c (рис. 1).

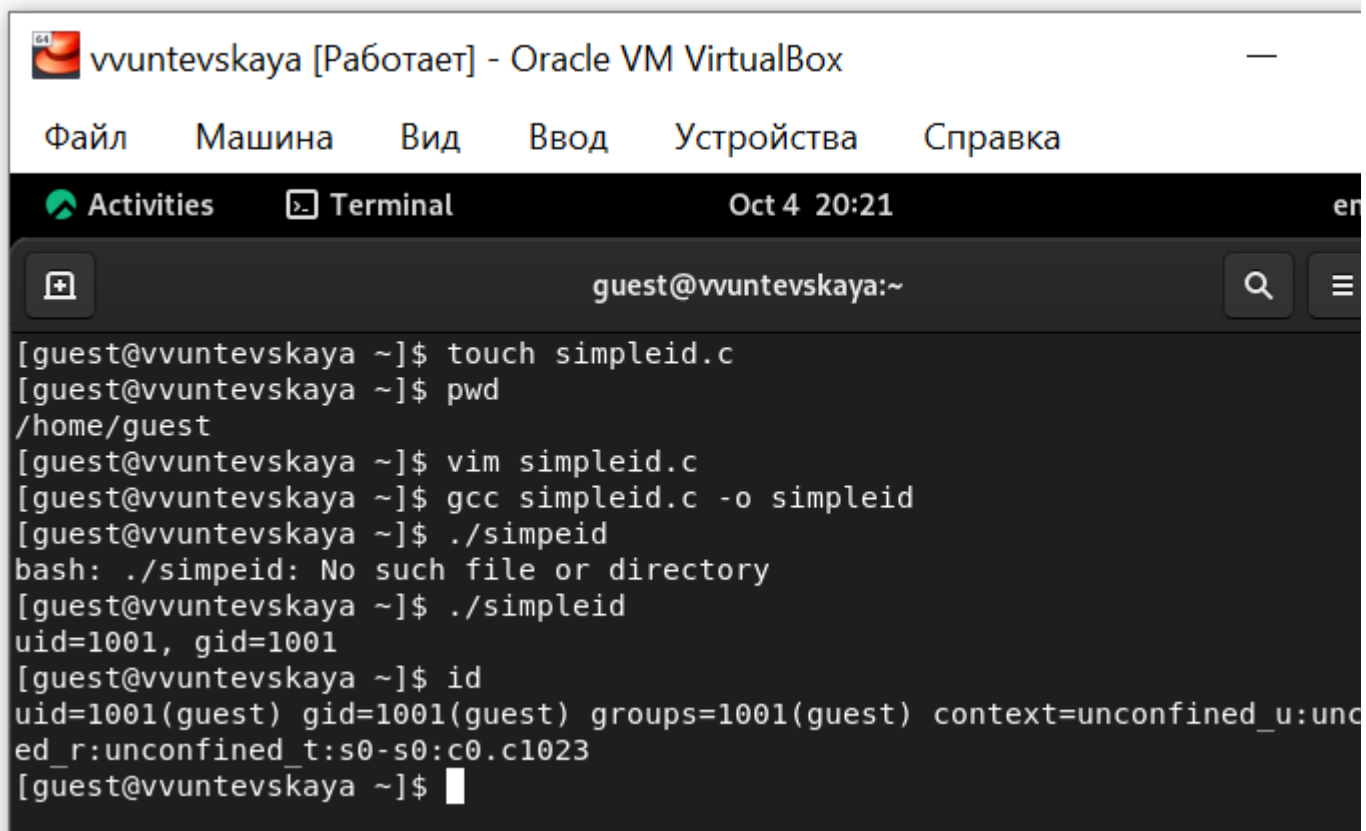


The screenshot shows a VirtualBox window titled "vvuntevskaya [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The top bar shows "Activities", "Terminal", and the date "Oct 4 19:42". A terminal window is open, displaying the contents of a C file named "simpleid.c" in the vim editor. The code is as follows:

```
guest@vvuntevskaya:~ — vim 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;
}
```

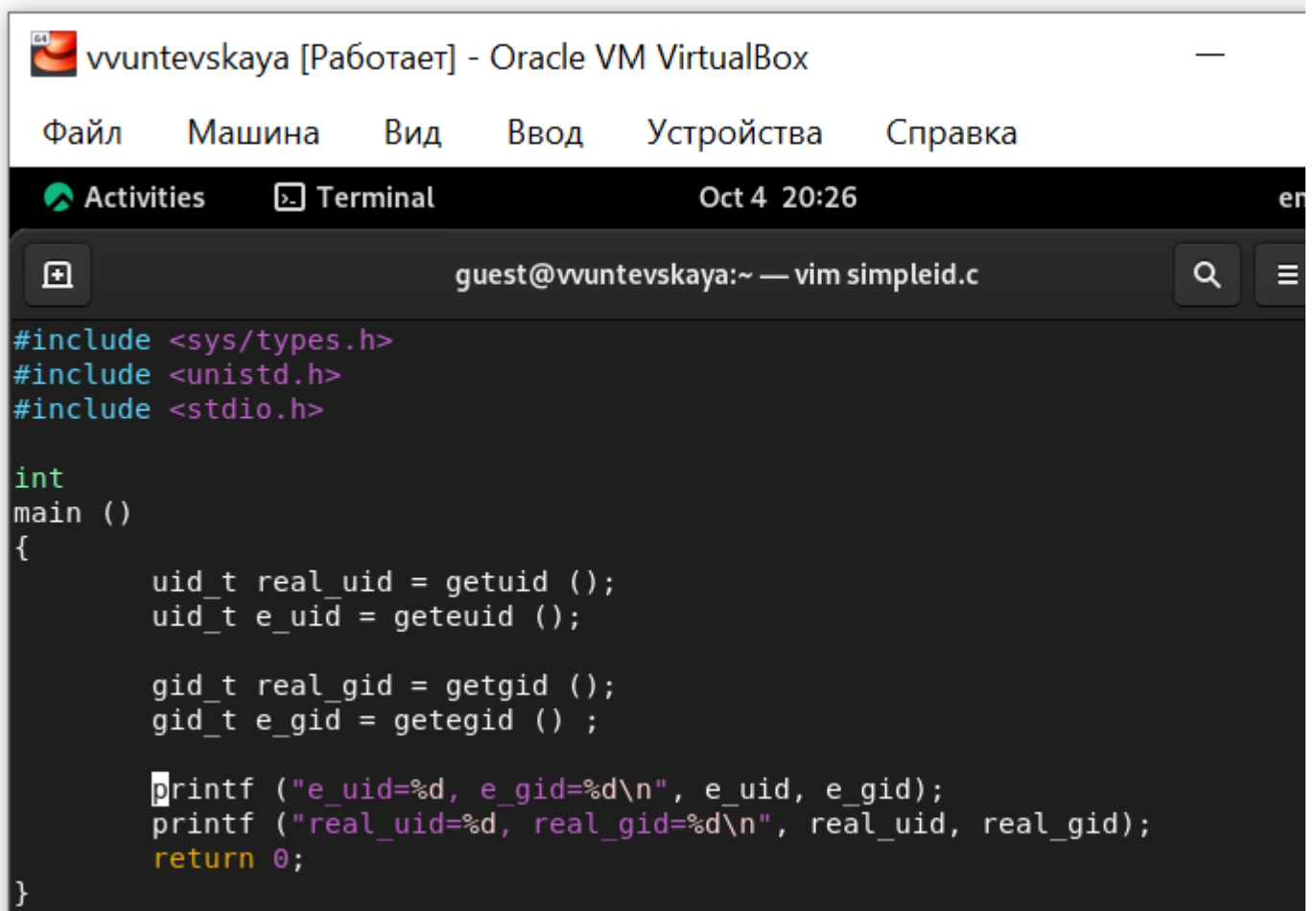
2. Скомпилировала программу и убедилась, что файл программы создан. Выполнила программу simpleid. Выполнила системную программу id. В отличие от команды id, моя программа не выводит контекст и все группы, в которые пользователь (рис. 2).



The screenshot shows the same VirtualBox window, but the terminal window now displays a series of commands and their outputs. The date is "Oct 4 20:21". The terminal output is as follows:

```
guest@vvuntevskaya:~
[guest@vvuntevskaya ~]$ touch simpleid.c
[guest@vvuntevskaya ~]$ pwd
/home/guest
[guest@vvuntevskaya ~]$ vim simpleid.c
[guest@vvuntevskaya ~]$ gcc simpleid.c -o simpleid
[guest@vvuntevskaya ~]$ ./simpeid
bash: ./simpeid: No such file or directory
[guest@vvuntevskaya ~]$ ./simpleid
uid=1001, gid=1001
[guest@vvuntevskaya ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unc
ed_r:unconfined_t:s0-s0:c0.c1023
[guest@vvuntevskaya ~]$
```

3. Усложнила программу, добавив вывод действительных идентификаторов (рис. 3).



The screenshot shows a VirtualBox window titled "vvuntevskaya [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The interface has tabs for "Activities" and "Terminal". The terminal window shows the command prompt "guest@vvuntevskaya:~ — vim simpleid.c". The code in the terminal is as follows:

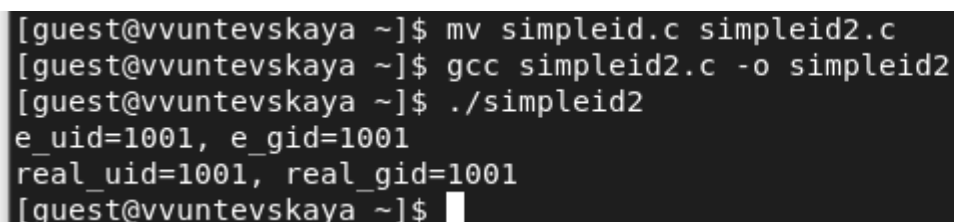
```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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;
}
```

4. Получившуюся программу назвала simpleid2.c. Скомпилировала и запустила simpleid2.c (рис. 4).



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

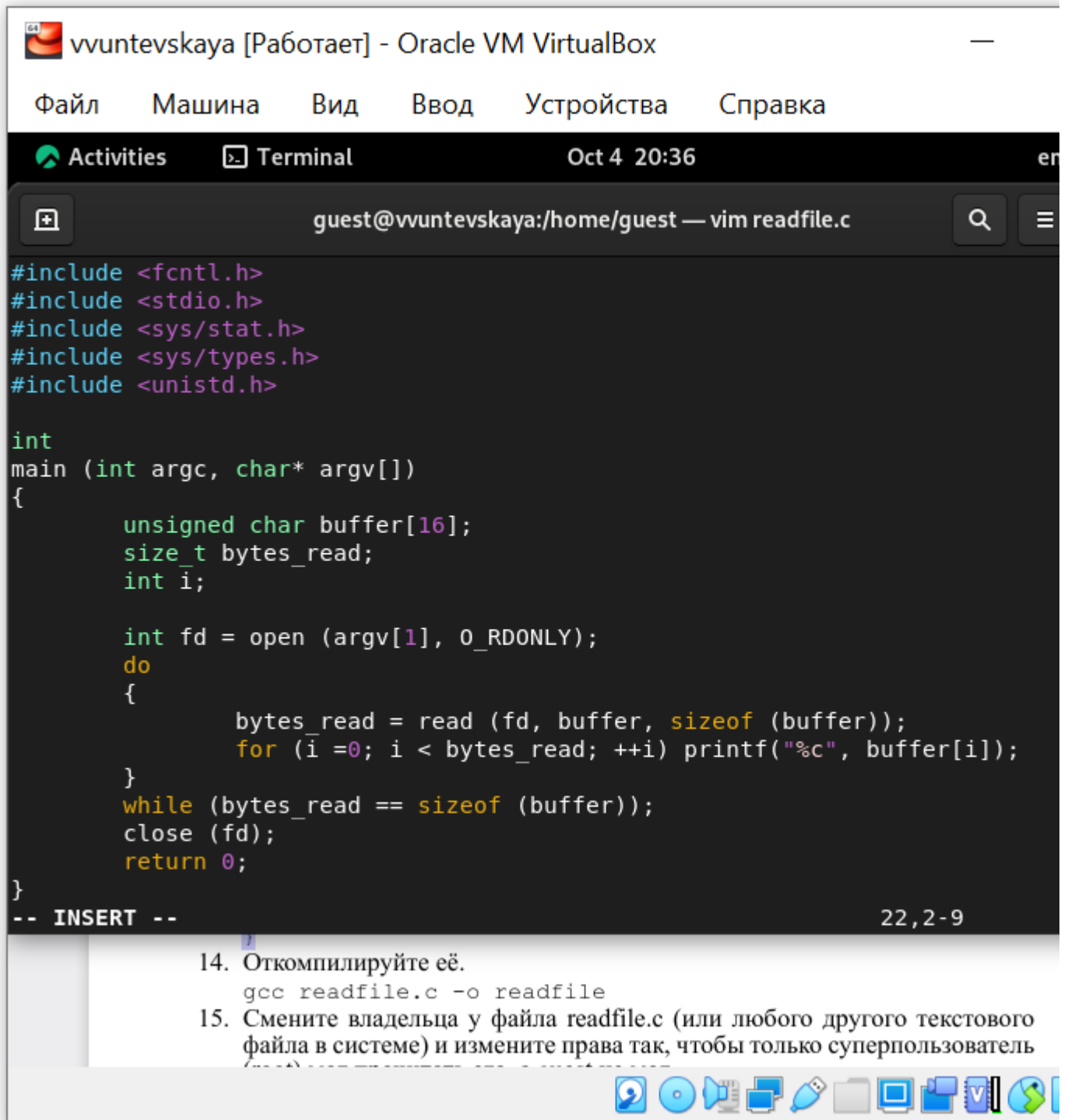
5. От имени суперпользователя выполнила команды `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`. Первая команда меняет владельца файла simpleid2 на группу guest. Вторая команда меняет права доступа к файлу simpleid2 для пользователя и установленные атрибуты SUID или SGID позволяют запускать файл на выполнение с правами владельца файла или группы соответственно. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Запустила simpleid2 и id. Сравнила результаты (рис. 5).

```
[guest@vvuntevskaya ~]$ su
Password:
[root@vvuntevskaya guest]# chown root:guest /home/guest/simpleid2
[root@vvuntevskaya guest]# chmod u+s /home/guest/simpleid2
[root@vvuntevskaya guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  4 20:27 simpleid2
[root@vvuntevskaya guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@vvuntevskaya guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@vvuntevskaya guest]#
```

6. Проделала тоже самое относительно SetGID-бита (рис. 6).

```
[guest@vvuntevskaya ~]$ su
Password:
[root@vvuntevskaya guest]# chown root:guest /home/guest/simpleid2
[root@vvuntevskaya guest]# chmod u+s /home/guest/simpleid2
[root@vvuntevskaya guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  4 20:27 simpleid2
[root@vvuntevskaya guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@vvuntevskaya guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@vvuntevskaya guest]#
```

7. Создала программу readfile.c (рис. 7).



vvuntevskaya [Работает] - Oracle VM VirtualBox

Файл Машина Вид Ввод Устройства Справка

Activities Terminal Oct 4 20:36

guest@vvuntevskaya:/home/guest — vim 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;

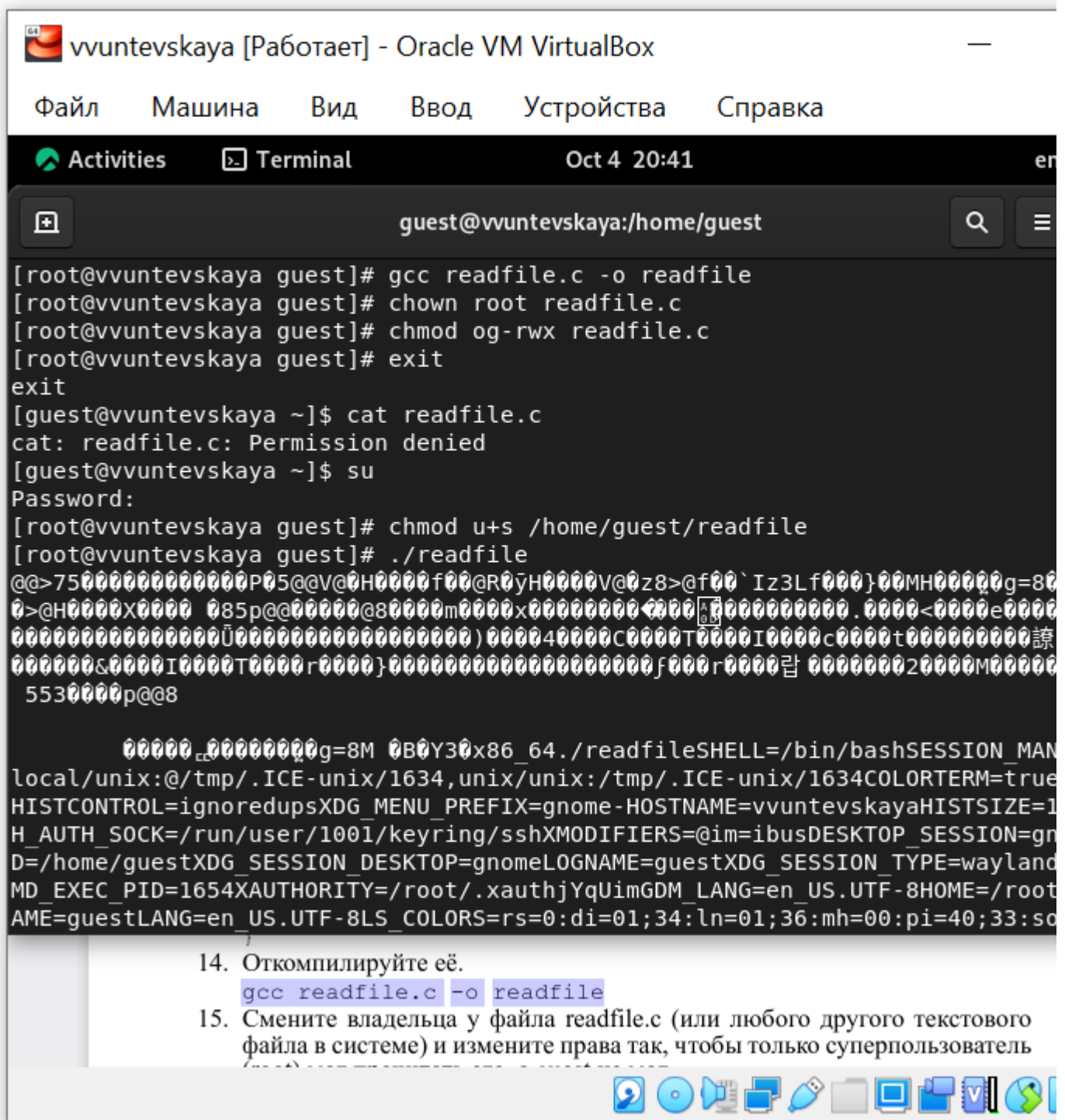
    int fd = open (argv[1], O_RDONLY);
    do
    {
        bytes_read = read (fd, buffer, sizeof (buffer));
        for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
    }
    while (bytes_read == sizeof (buffer));
    close (fd);
    return 0;
}
```

-- INSERT -- 22,2-9

14. Откомпилируйте её.
gcc readfile.c -o readfile

15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Проверила, что пользователь guest не может прочитать файл readfile.c. Сменила у программы readfile владельца и установила SetU'D-бит. Проверила, может ли программа readfile прочитать файл readfile.c (рис. 8).

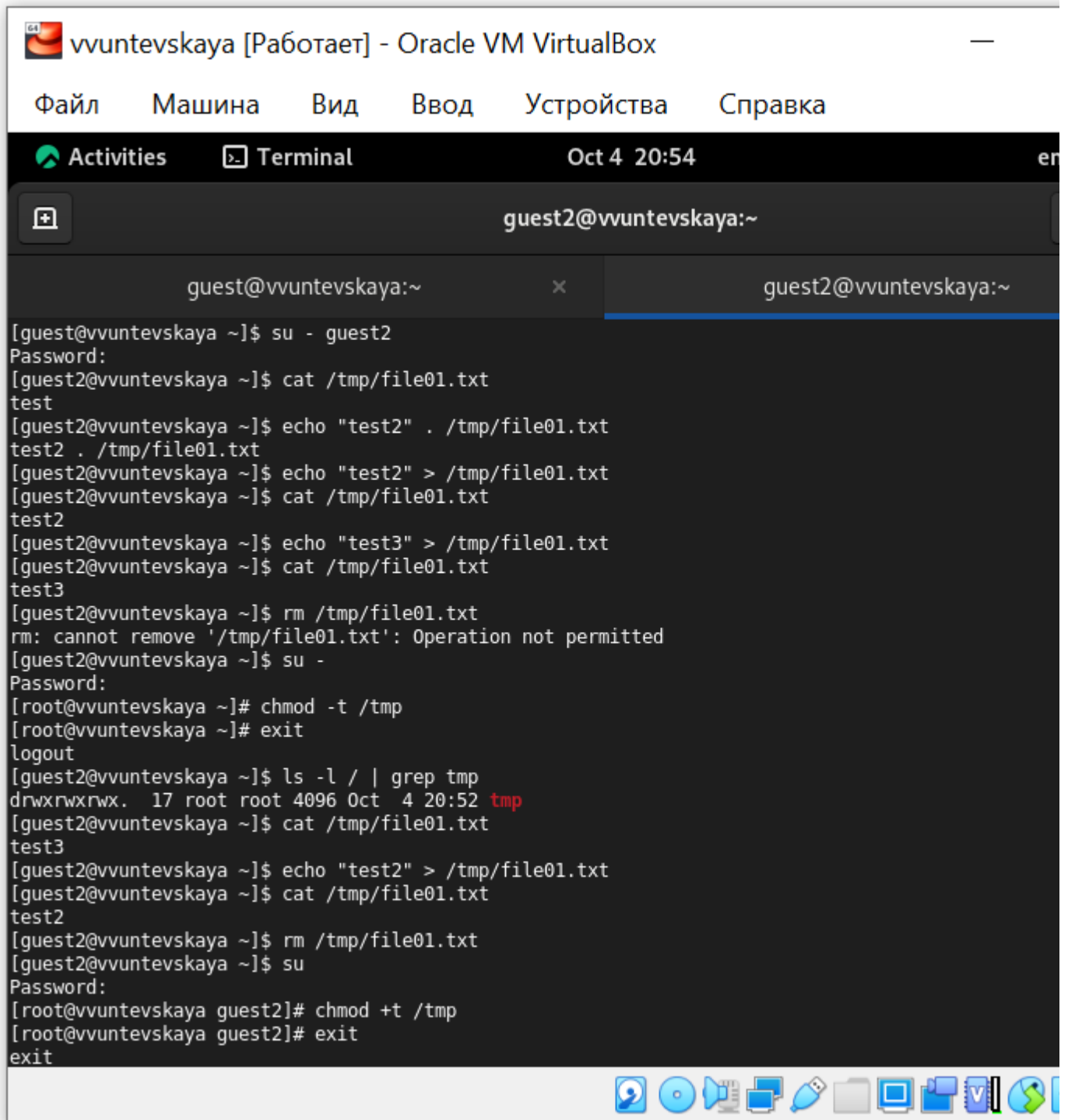
8. Откомпилировала программу. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Проверила, что пользователь guest не может прочитать файл readfile.c. Сменила у программы readfile владельца и установила SetU'D-бит. Проверила, может ли программа readfile прочитать файл readfile.c (рис. 8).



9. Выяснила, установлен ли атрибут Sticky на директории /tmp. От имени пользователя guest создала файл file01.txt в директории/tmp со словом test. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» (рис. 9).

```
[guest@vvuntevskaya ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  4 20:40 tmp
[guest@vvuntevskaya ~]$ echo "test" > /tmp/file01.txt
[guest@vvuntevskaya ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  4 20:46 /tmp/file01.txt
[guest@vvuntevskaya ~]$ chmod 0+rw /tmp/file01.txt
chmod: invalid mode: '0+rw'
Try 'chmod --help' for more information.
[guest@vvuntevskaya ~]$ chmod o+rw /tmp/file01.txt
[guest@vvuntevskaya ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  4 20:46 /tmp/file01.txt
[guest@vvuntevskaya ~]$
```

10. От пользователя guest2 попробовала прочитать файл /tmp/file01.txt. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2. Удалось выполнить операцию. Проверила содержимое файла. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию. Удалось выполнить операцию. Проверила содержимое файла. От пользователя guest2 попробовала удалить файл /tmp/file01.tx. Не удалось выполнить операцию. Повысила свои права до суперпользователя и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Покинула режим суперпользователя. От пользователя guest2 проверила, что атрибута t у директории /tmp нет. Повторила предыдущие шаги. Удалось успешно выполнить каждый шаг. Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp (рис. 10).



```
vvuntevskaya [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Activities  Terminal  Oct 4 20:54
guest2@vvuntevskaya:~
guest@vvuntevskaya:~ x guest2@vvuntevskaya:~
[guest@vvuntevskaya ~]$ su - guest2
Password:
[guest2@vvuntevskaya ~]$ cat /tmp/file01.txt
test
[guest2@vvuntevskaya ~]$ echo "test2" . /tmp/file01.txt
test2 . /tmp/file01.txt
[guest2@vvuntevskaya ~]$ echo "test2" > /tmp/file01.txt
[guest2@vvuntevskaya ~]$ cat /tmp/file01.txt
test2
[guest2@vvuntevskaya ~]$ echo "test3" > /tmp/file01.txt
[guest2@vvuntevskaya ~]$ cat /tmp/file01.txt
test3
[guest2@vvuntevskaya ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@vvuntevskaya ~]$ su -
Password:
[root@vvuntevskaya ~]# chmod -t /tmp
[root@vvuntevskaya ~]# exit
logout
[guest2@vvuntevskaya ~]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Oct  4 20:52 tmp
[guest2@vvuntevskaya ~]$ cat /tmp/file01.txt
test3
[guest2@vvuntevskaya ~]$ echo "test2" > /tmp/file01.txt
[guest2@vvuntevskaya ~]$ cat /tmp/file01.txt
test2
[guest2@vvuntevskaya ~]$ rm /tmp/file01.txt
[guest2@vvuntevskaya ~]$ su
Password:
[root@vvuntevskaya guest2]# chmod +t /tmp
[root@vvuntevskaya guest2]# exit
exit
```

Выводы

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

влияние бита Sticky на запись и удаление файлов.