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

Ход работы:

Установим компилятор gcc (Рис. 1)

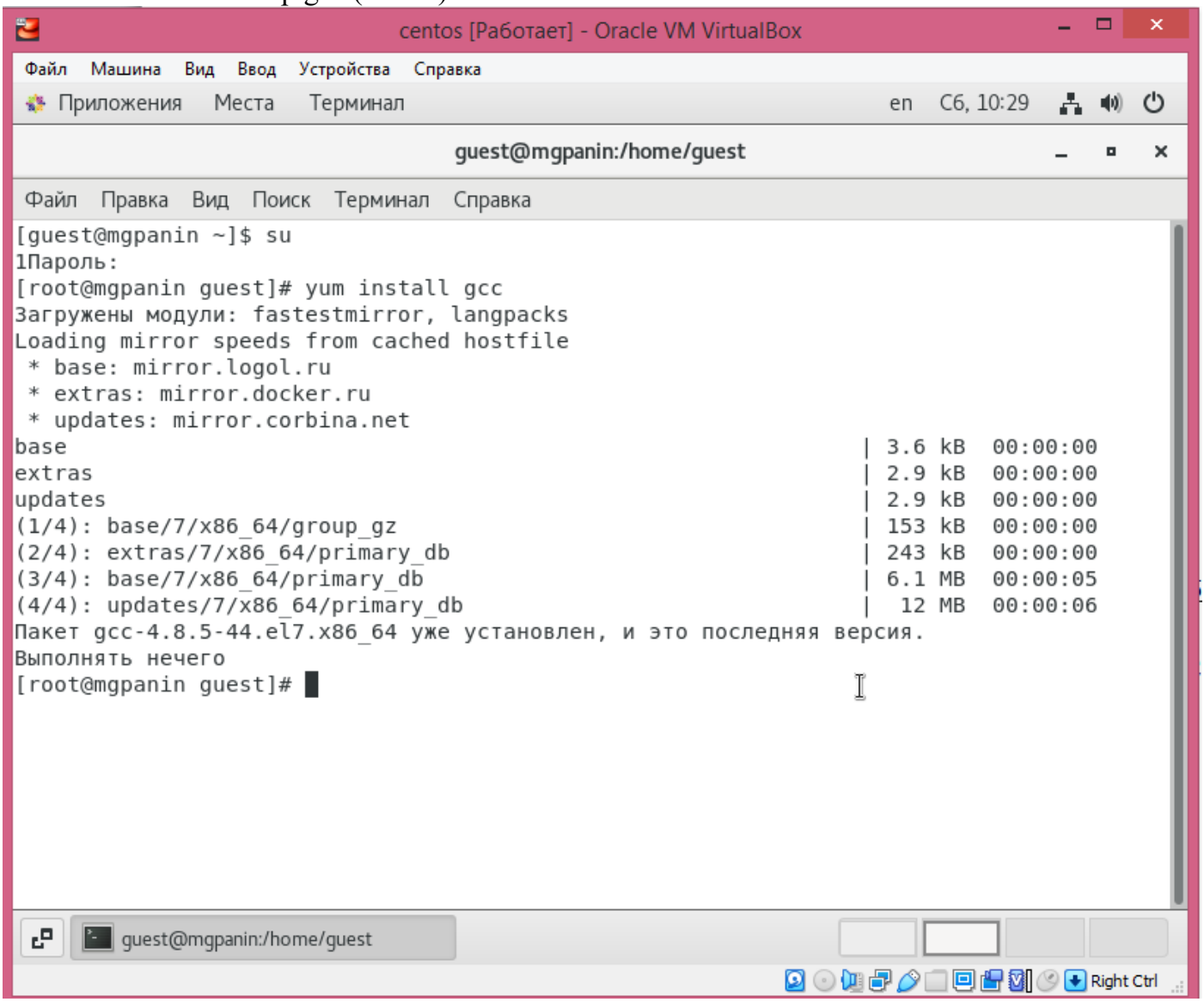

Рис. 1

Создадим программу simpleid.c (Рис. 2)

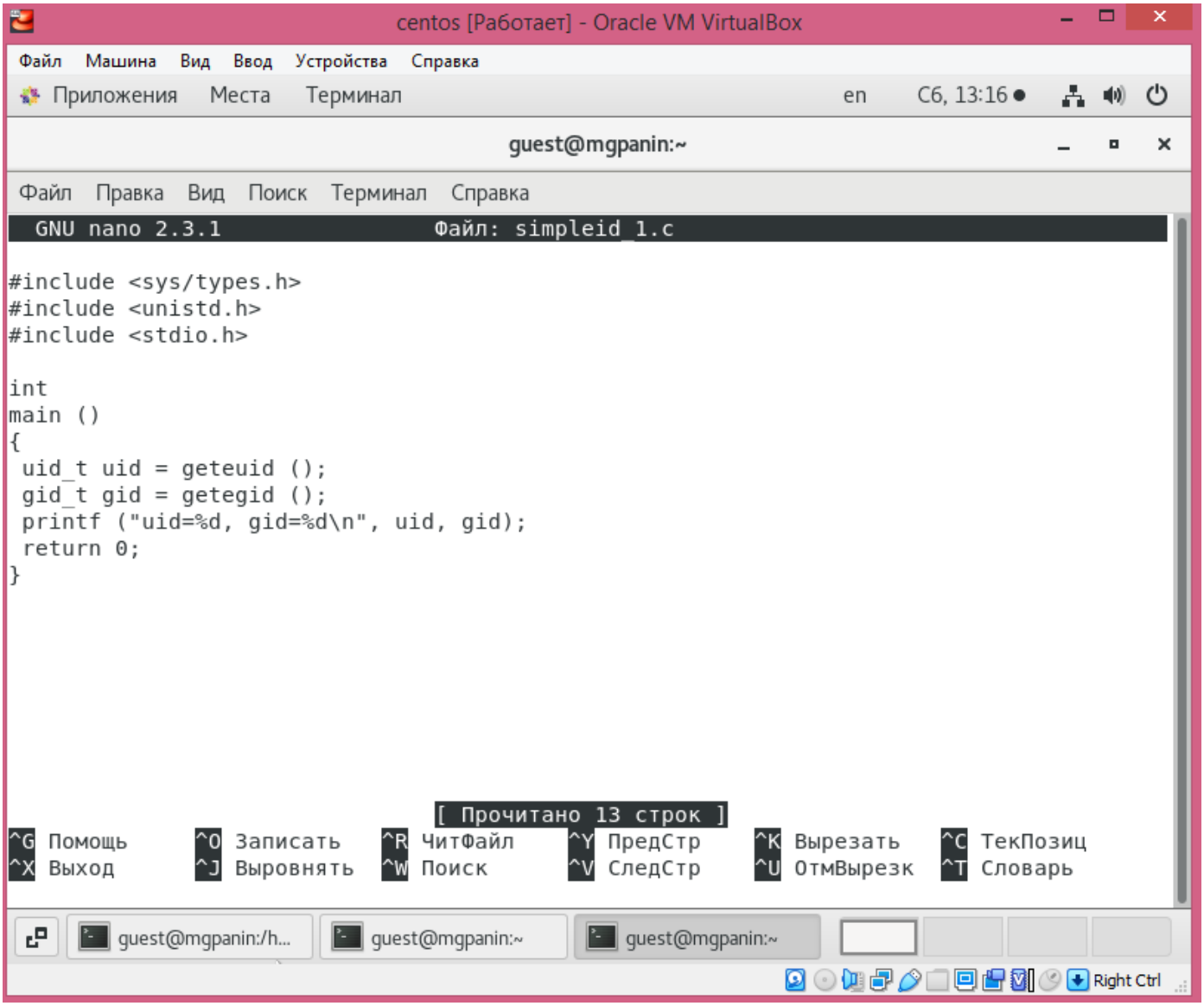

Рис. 2

1) Скомпилируем программу (Рис. 3)
   gcc simpleid.c –o simpleid
2) Выполним програму simpleid (Рис. 3)
   ./simpleid

Выполним системную программу id (Рис. 3)
id

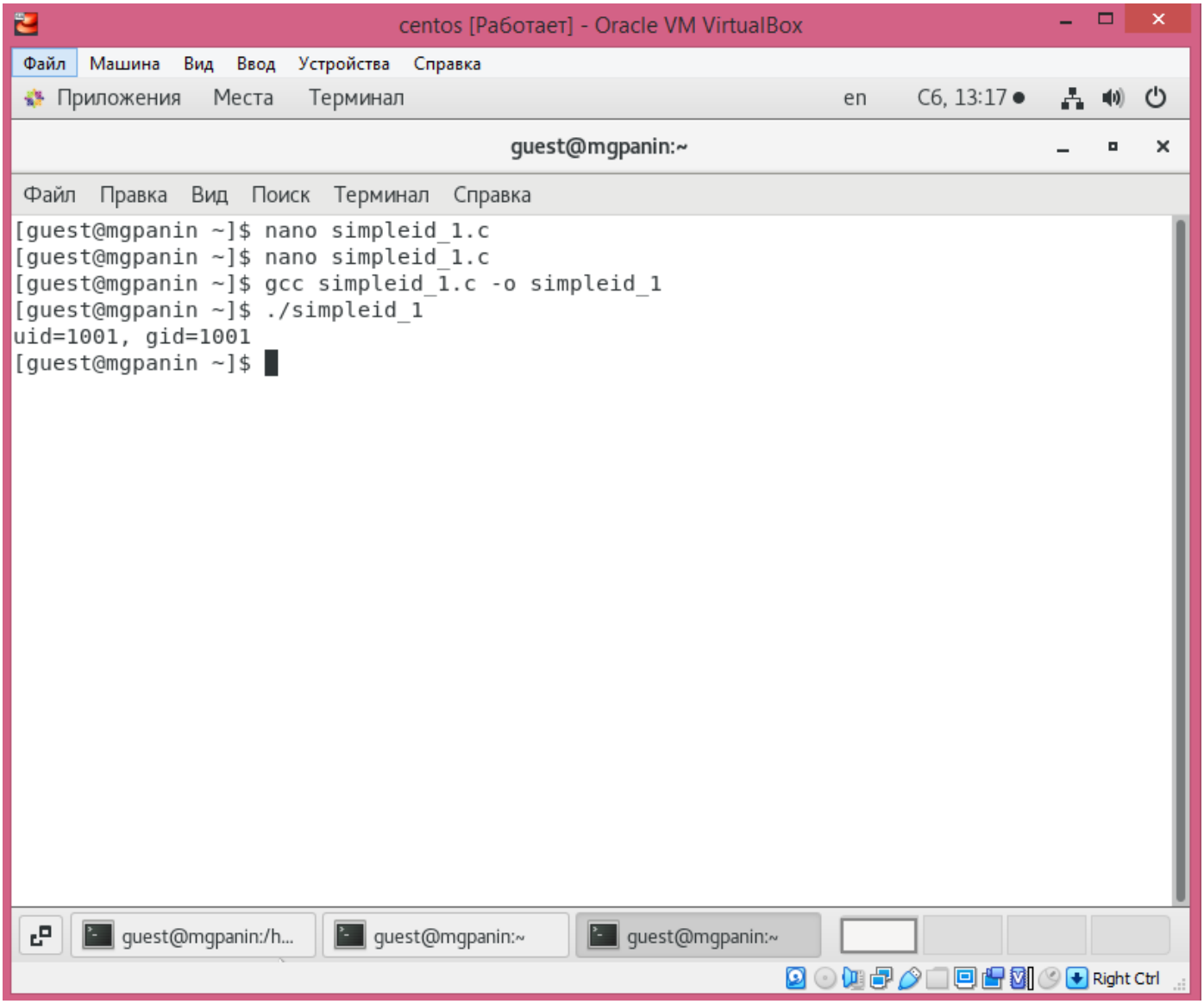

Рис. 3

Результаты совпадают

Усложним программу, добавив вывод действительных идентификаторов, сохранив как simpleid2.c (Рис. 4)

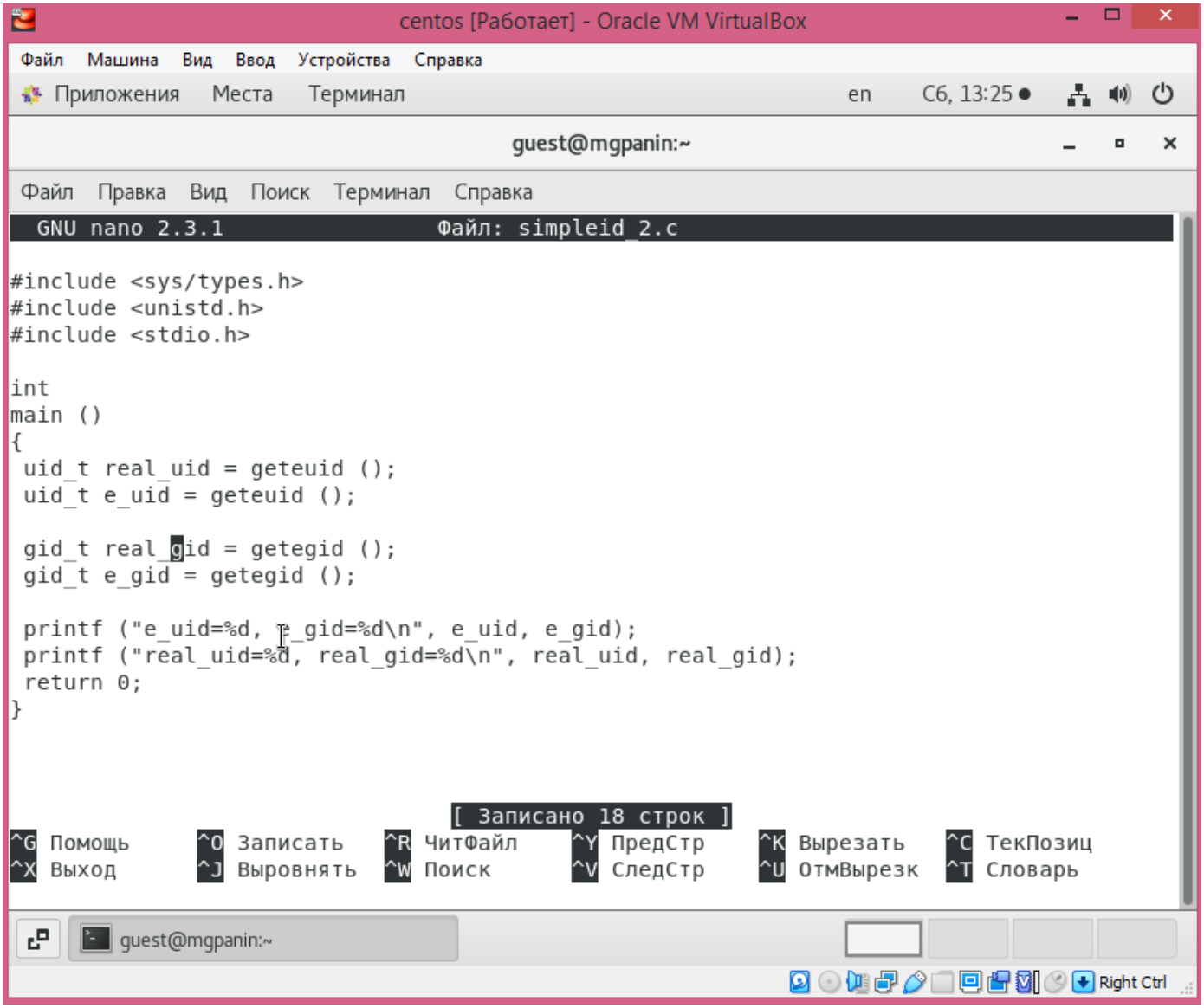

Рис. 4

3) Скомпилируем и запустим simpleid2.c (Рис. 5)
   gcc simpleid2.c –o simpleid
   ./simpleid2
4) От имени суперпользователя выполним команды (Рис. 5)
   chown root:guest /home/guest/simpleid2
   chmod u+s /home/guest/simpleid2
   С помощью этих команд файлу simpleid2 изменяем владельца и группу на root и guest соответственно, а также устанавливаем на файл SetUID-бит
5) Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (Рис. 5)
   ls –l simpleid2

Запустим simpleid2 и id (Рис. 5)

./simpleid2

id

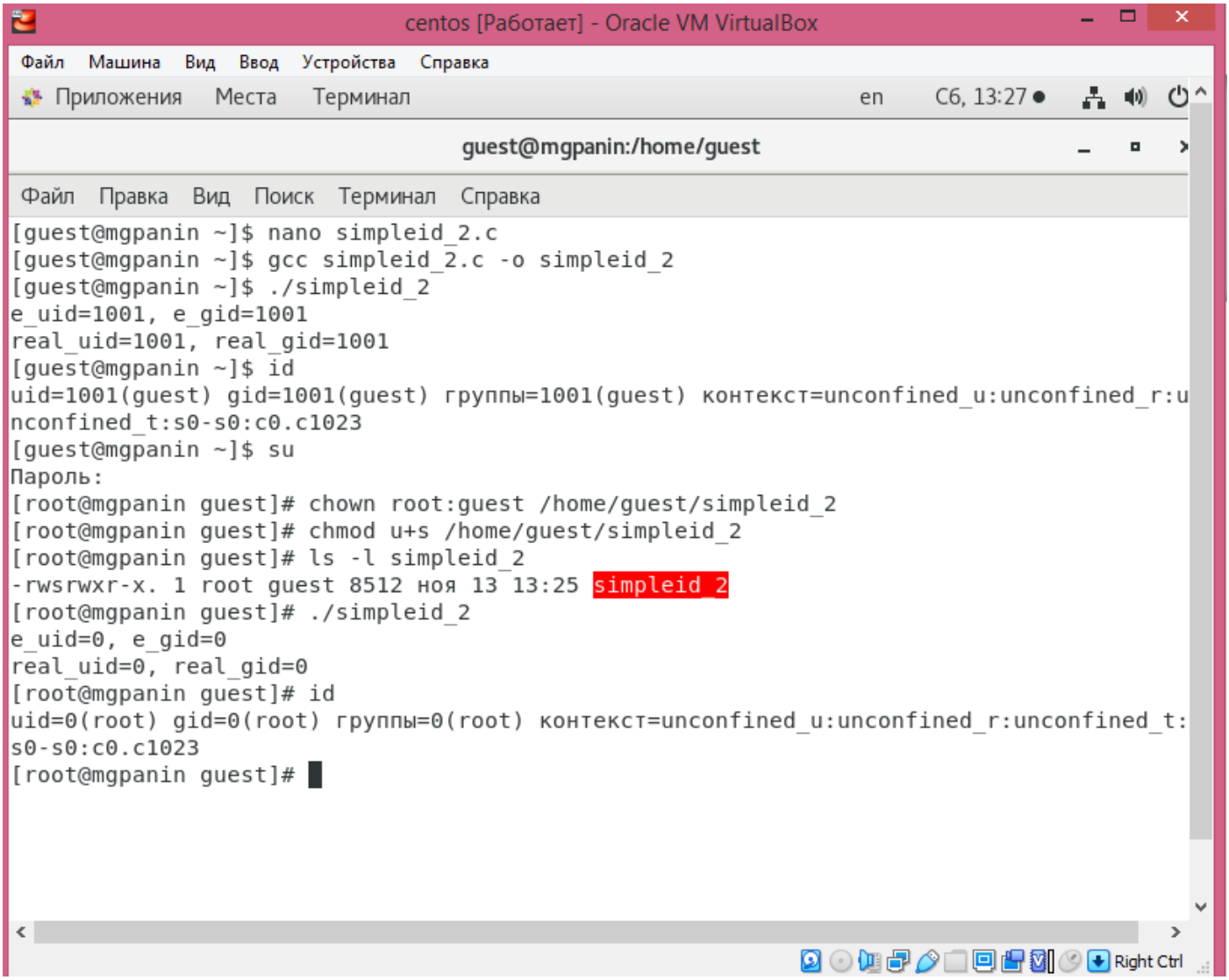

Рис. 5

Результаты также одинаковы

Проделаем тоже самое относительно SetGID-бита (Рис. 6)

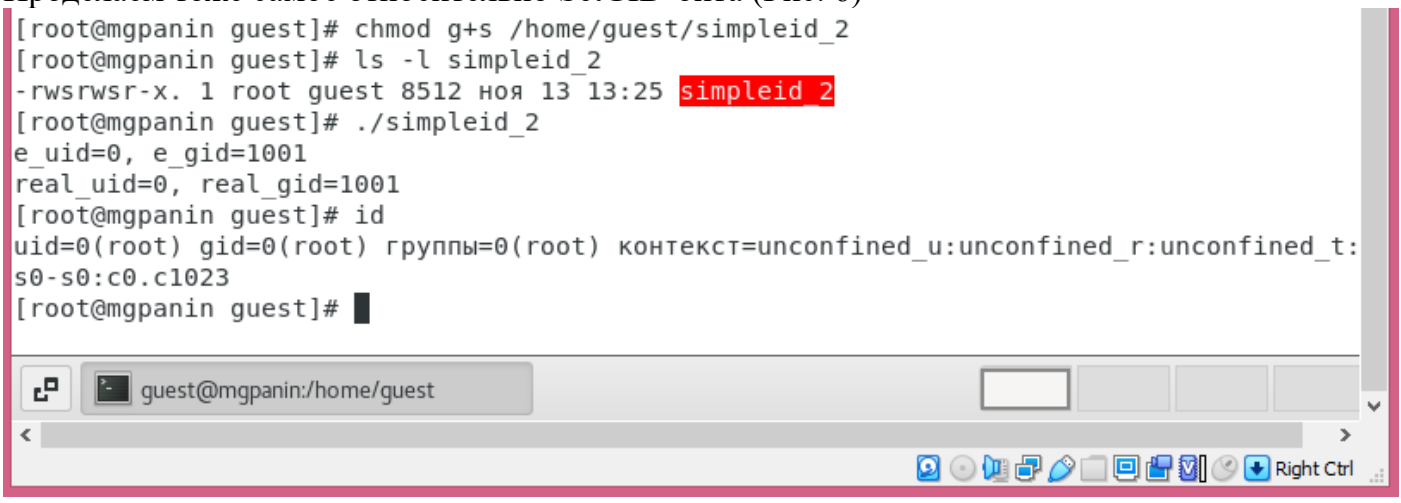

Рис. 6

Создадим readfile.c (Рис. 7)

Рис. 7

6) Откомпилируем ее (Рис. 8)
   gcc readfile.c –o readfile
7) Сменим владельца у файла readfile.c и изменим права так, чтобы только root мог прочитать его (Рис. 8)
8) Проверим, что пользователь guest не может прочитать файл readfile.c (Рис. 8)

Сменим у программы readfile владельца и установим SetUID-бит (Рис. 8)

```
[root@mgpanin guest]# nano readfile.c
[root@mgpanin guest]# nano readfile.c
[root@mgpanin guest]# gcc readfile.c -o readfile
[root@mgpanin guest]# chmod a-r readfile.c
[root@mgpanin guest]# exit
exit
[guest@mgpanin ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@mgpanin ~]$ su
1Пароль:
chsu: Сбой при проверке подлинности
[guest@mgpanin ~]$ chown root:guest readfile
chown: изменение владельца «readfile»: Операция не позволена
[guest@mgpanin ~]$ su
Пароль:
[root@mgpanin guest]# chown root:guest readfile
[root@mgpanin guest]# chmod u+s readfile
[root@mgpanin guest]# exit
exit
[guest@mgpanin ~]$
```

guest@mgpanin:~ | Messenger - Mozilla Firefox

Рис. 8

Проверим, может ли программа readfile прочитать файл readfile.c (Рис. 9)

Приложения Места Терминал en Сб, 13:41

guest@mgpanin:~

Файл Правка Вид Поиск Терминал Справка

```
[root@mgpanin guest]# exit
exit
[guest@mgpanin ~]$ ./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;
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;
}
[guest@mgpanin ~]$
```

guest@mgpanin:~ | Messenger - Mozilla Firefox

Рис. 9

Проверим, может ли программа readfile прочитать файл /etc/shadow (Рис. 10)

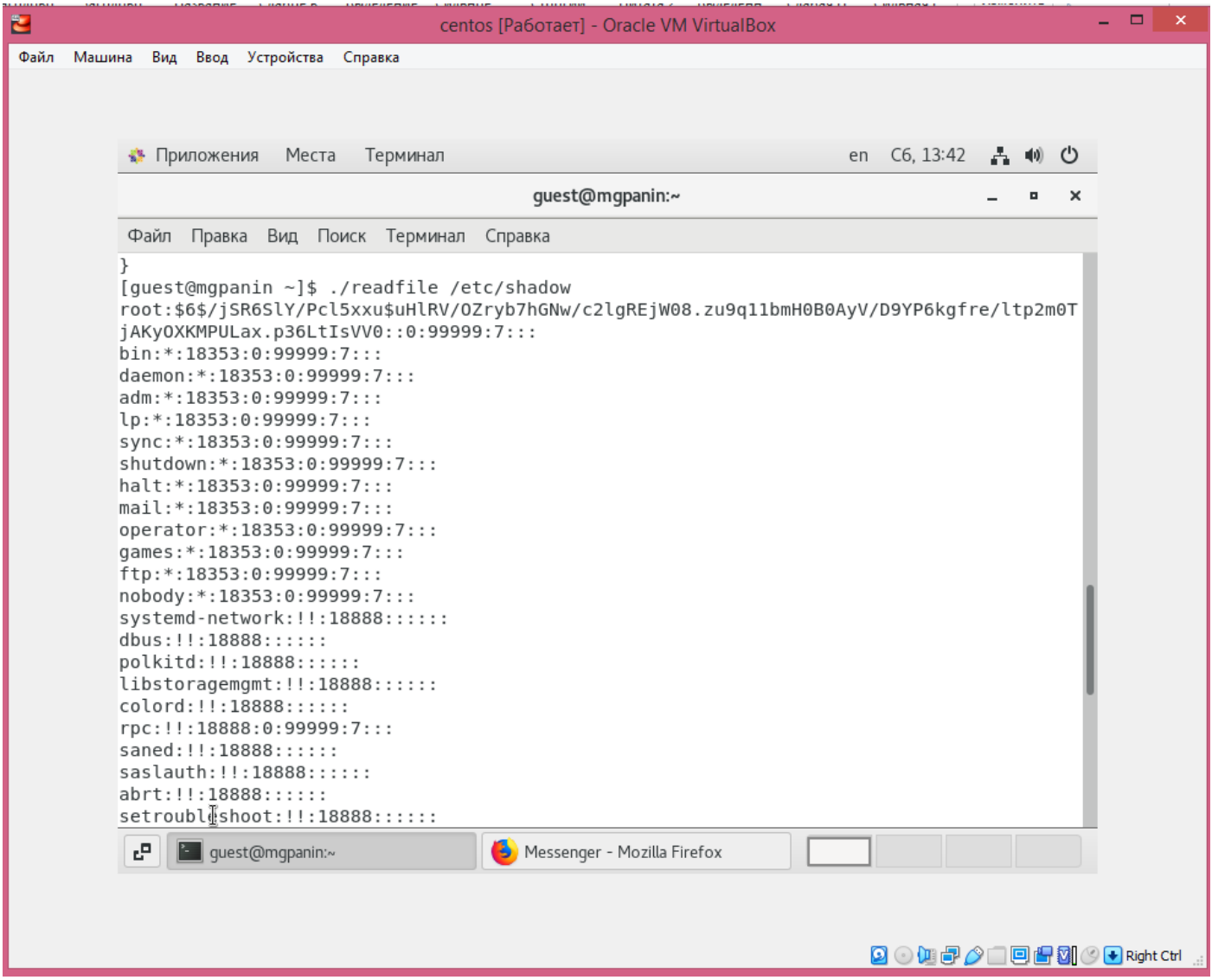

Рис. 10

9) Выясним, установлен ли атрибут Sticky на директории /tmp (Рис. 11)
   ls –l / | grep tmp
10) От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test (Рис. 11)
   echo "test" > /tmp/file01.txt

Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» (Рис. 11)

ls –l /tmp/file01.txt
chmod o+rw /tmp/file01.txt
ls –l /tmp/file01.txt

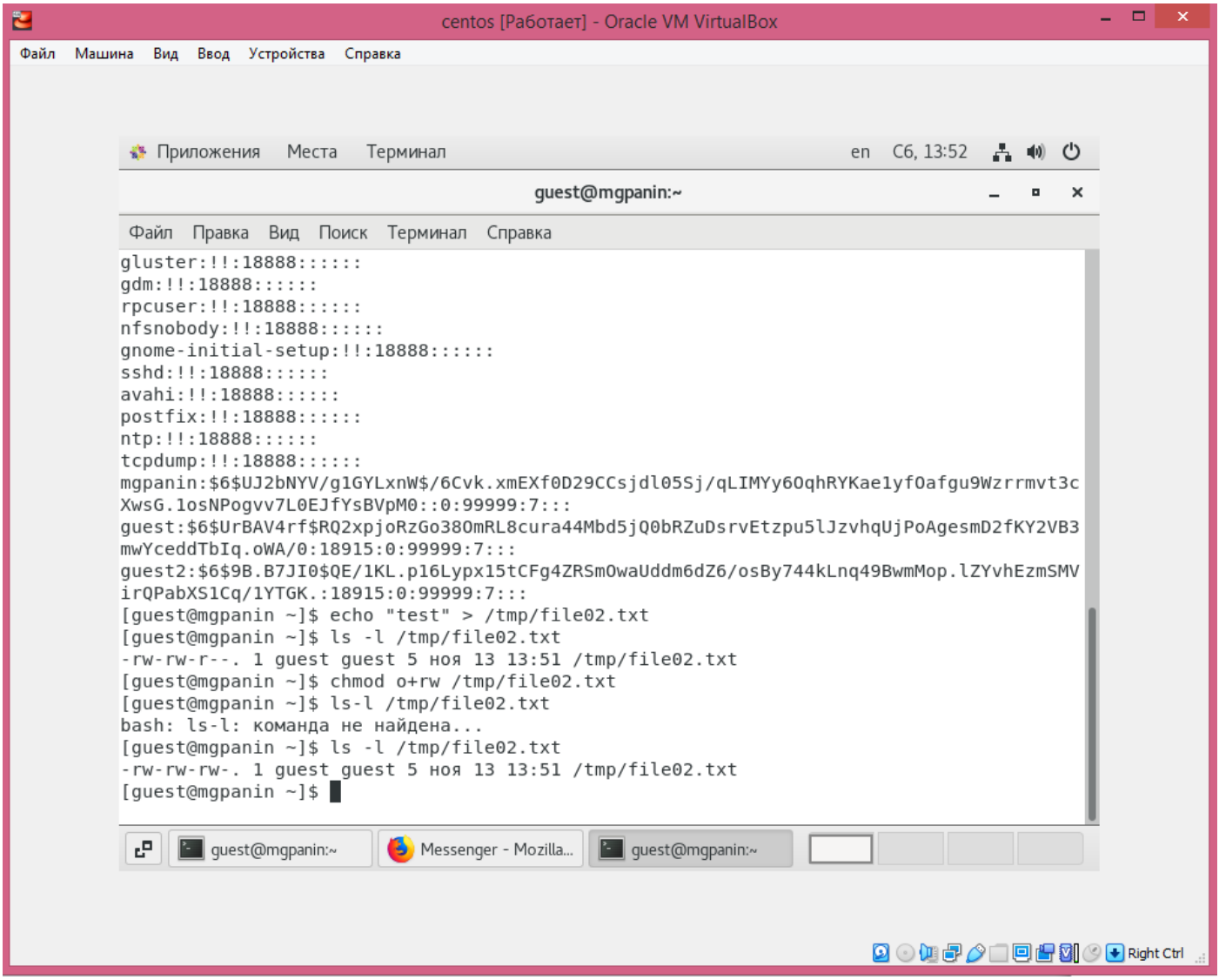

Рис. 11

11) От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt (Рис. 12)
    cat /tmp/file01.txt
12) От пользователей guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2 (Рис. 12)
    echo "test2" > /tmp/file01.txt
    Дозаписать не получилось
13) Проверим содержимое файла (Рис. 12)
    cat /tmp/file01.txt
14) От пользователей guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев всю информацию в файле (Рис. 12)
    echo "test3" > /tmp/file01.txt
    Перезаписать информацию получилось
15) Проверим содержимое файла (Рис. 12)
    cat /tmp/file01.txt
16) От пользователя guest2 попробуйте удалить файл /tmp/file01.txt (Рис. 12)
    rm /tmp/file01.txt
    Файл не удалился
17) Повысим свои права до суперпользователя (Рис. 12)
    su –
    И выполним после этого команду снимающую атрибут Sticky-бита с директории /tmp

chmod –t /tmp

18) Покинем режим суперпользователя (Рис. 12)

exit

19) От пользователя guest2 проверим, что атрибут t у директории /tmp нет (Рис. 12)

ls –l / | grep tmp

Повторим предыдущие шаги (Рис. 12)

Файл удалось удалить от имени пользователя, не являющегося его владельцем

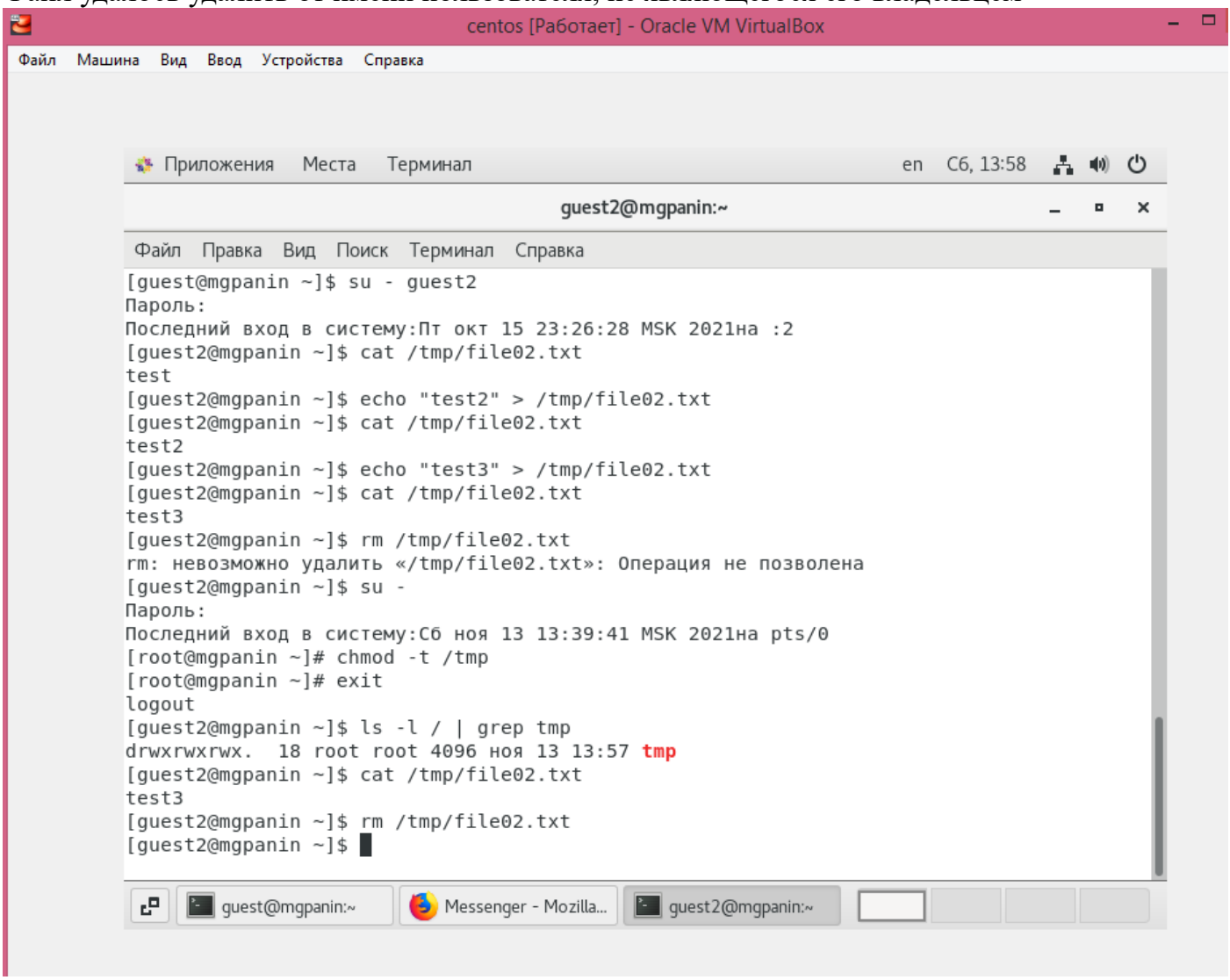

Рис. 12

Повысим свои права до суперпользователя и вернем атрибут t на директорию /tmp (Рис. 13)

su -

chmod +t /tmp

exit

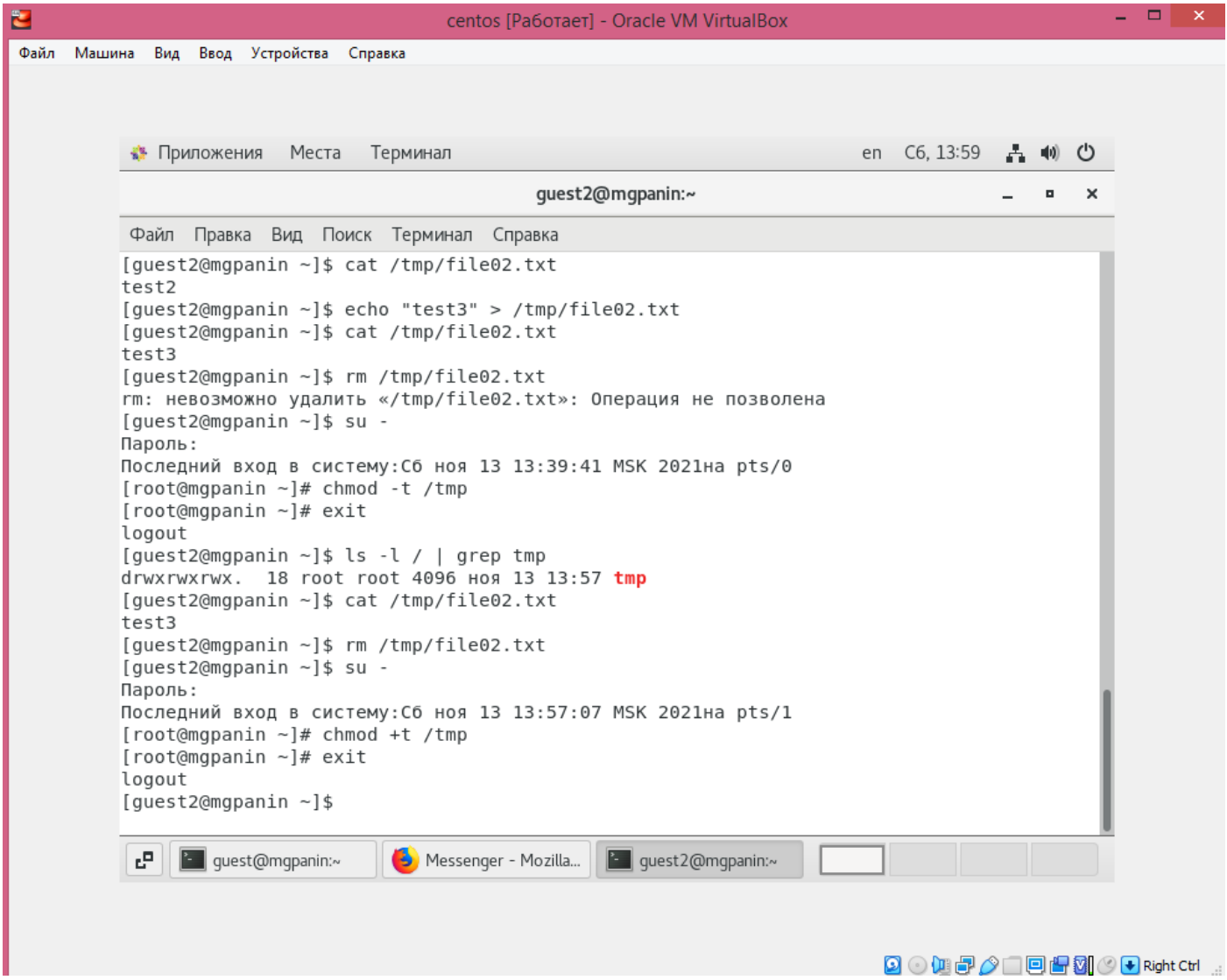

Рис. 13

Вывод:

Я изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, получил практические навыки работы в консоли с дополнительными атрибутами, а также рассмотрели работы механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов