

Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов

Сухарев Кирилл

Содержание

Цель работы	5
Условные обозначения и термины	6
Теоретические вводные данные	7
Техническое оснащение и выбранные методы проведения работы	8
Выполнение работы	9
Создание программы	9
Исследование Sticky-бита	23
Выводы	34
Библиография	35

List of Figures

0.1	Создание simpleid.c	9
0.2	Проверка работоспособности	10
0.3	Создание simpleid2.c	11
0.4	Создание simpleid2.c	12
0.5	chown и chmod	13
0.6	Проверка правильности	14
0.7	Выполнение simpleid2	15
0.8	Установка SetGID-бита	16
0.9	Создание readfile.c	17
0.10	Компиляция файла readfile.c	18
0.11	Смена прав у файла	19
0.12	Проверка недоступности readfile.c для guest	20
0.13	Смена владельца и установка SetUID-бита	21
0.14	Попытка прочитать readfile.c	22
0.15	Попытка прочитать etc/shadow	23
0.16	Проверка наличия атрибута Sticky	24
0.17	Создание file01.txt	25
0.18	Попытка чтения file01.txt	26
0.19	Дозапись file01.txt	27
0.20	Перезапись file01.txt	28
0.21	Попытка удаления file01.txt	29
0.22	Снятие Sticky-бита	30
0.23	Проверка снятия Sticky-бита	31
0.24	Проверка предыдущих команд	32
0.25	Проверка предыдущих команд	33

List of Tables

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.

Условные обозначения и термины

Утилита - сервисная программа, облегчающая пользование другими программами, работу с компьютером.

Учетная запись - хранимая в компьютерной системе совокупность данных о пользователе, необходимая для его опознавания (аутентификации) и предоставления доступа к его личным данным и настройкам.

Директория - объект в файловой системе, упрощающий организацию файлов.

Теоретические вводные данные

setuid (от англ. set user ID upon execution — «установка ID пользователя во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца исполняемого файла. Иногда файлы требуют разрешения на выполнение для пользователей, которые не являются членами группы владельца, в этом случае вам потребуется предоставить специальные разрешения на выполнение. Когда SUID установлен, пользователь может запускать любую программу, такую как владелец программы.

Если SUID бит установлен на файл и пользователь выполнил его, процесс будет иметь те же права что и владелец файла.

setgid (от англ. set group ID upon execution — «установка ID группы во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами группы исполняемого файла.

Так же, как SUID, установив SGID бит для файла он устанавливает ваш идентификатор группы для группы файла в то время как файл выполняется. Это действительно полезно в случае когда у вас есть реальные установки в многопользовательском режиме где у пользователей есть доступ к файлом. В одной домашней категории я действительно не нашел использования для SGID. Но основная концепция является такой же, как и у SUID, файлы у которых SGID бит устанавливается, то они принадлежат к этой группе , а не к этому пользователю.

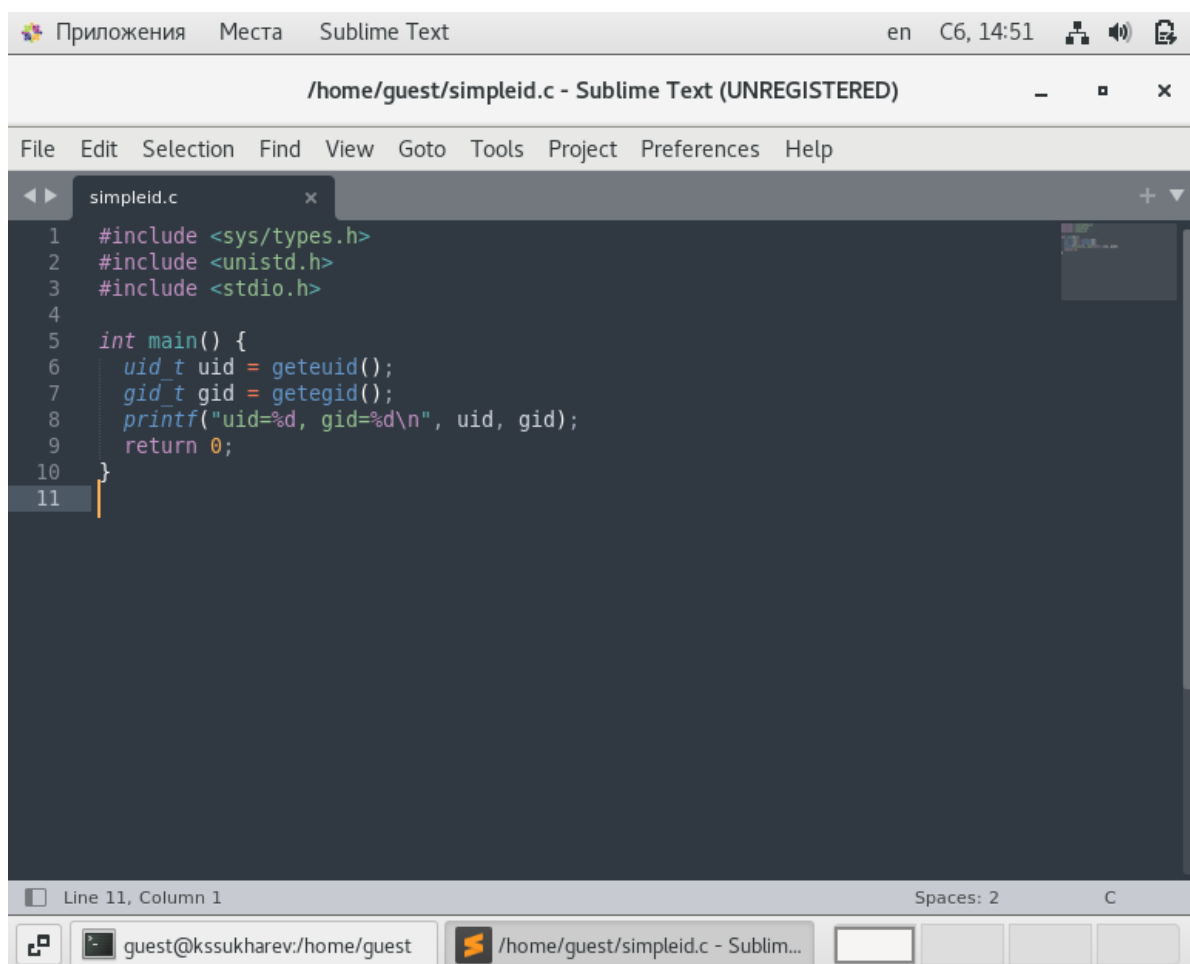
Техническое оснащение и выбранные методы проведения работы

В качестве среды выполнения лабораторной работы используется менеджер виртуальных машин VirtualBox и установленная с его помощью ОС Centos 7 на базе Linux.

Выполнение работы

Создание программы

1. Войдем в систему под пользователем `guest` и внесем туда программу на языке C (fig. 0.1).



```
Приложения  Места  Sublime Text  en  C6, 14:51  [icons]

/home/guest/simpleid.c - Sublime Text (UNREGISTERED)  -  [icons]

File  Edit  Selection  Find  View  Goto  Tools  Project  Preferences  Help

simpleid.c  x  +  v

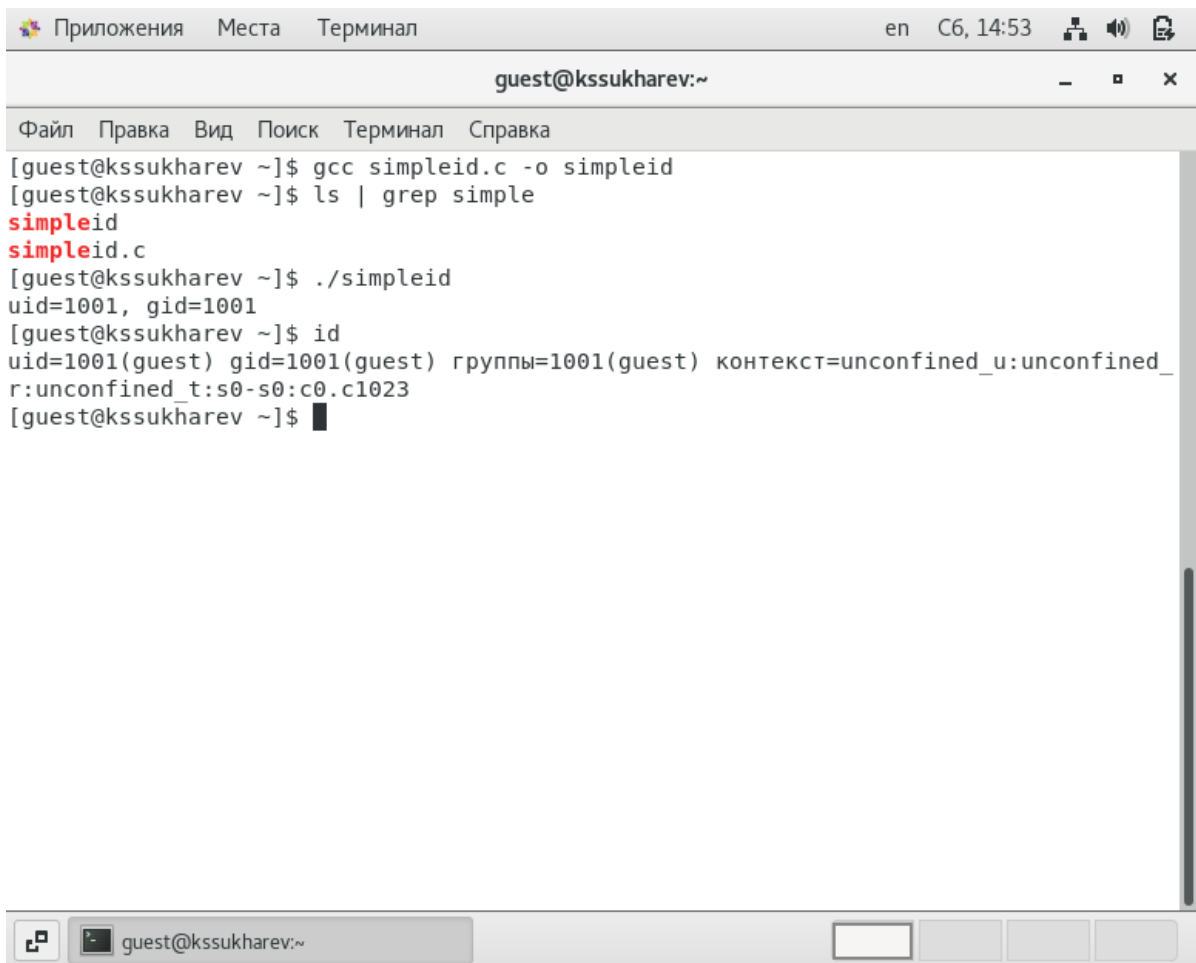
1  #include <sys/types.h>
2  #include <unistd.h>
3  #include <stdio.h>
4
5  int main() {
6      uid_t uid = geteuid();
7      gid_t gid = getegid();
8      printf("uid=%d, gid=%d\n", uid, gid);
9      return 0;
10 }
11

Line 11, Column 1  Spaces: 2  C

[icons]  guest@kssukharev:/home/guest  /home/guest/simpleid.c - Sublim...
```

Figure 0.1: Создание `simpleid.c`

2. Скомпилируем программу и выполним ее. Затем выполним программу `id` и убедимся, что выведенные группы соответствуют действительности (fig. 0.2).



```
Приложения  Места  Терминал  en  C6, 14:53  [иконки]
guest@kssukharev:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@kssukharev ~]$ gcc simpleid.c -o simpleid
[guest@kssukharev ~]$ ls | grep simple
simpleid
simpleid.c
[guest@kssukharev ~]$ ./simpleid
uid=1001, gid=1001
[guest@kssukharev ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@kssukharev ~]$
```

Figure 0.2: Проверка работоспособности

3. Создадим файл `simpleid2.c`, где дополнительно будем выводить действительные идентификаторы (fig. 0.3).

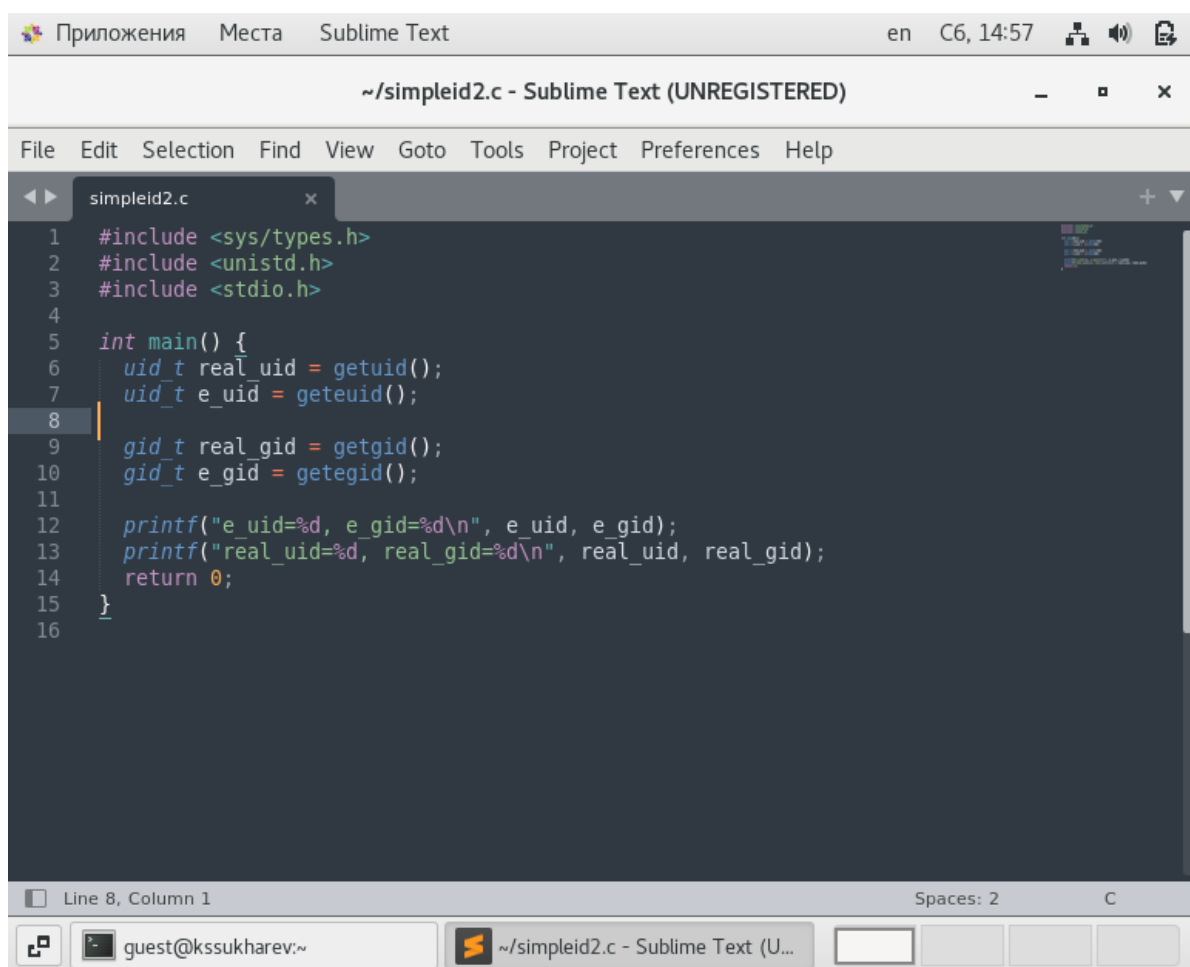
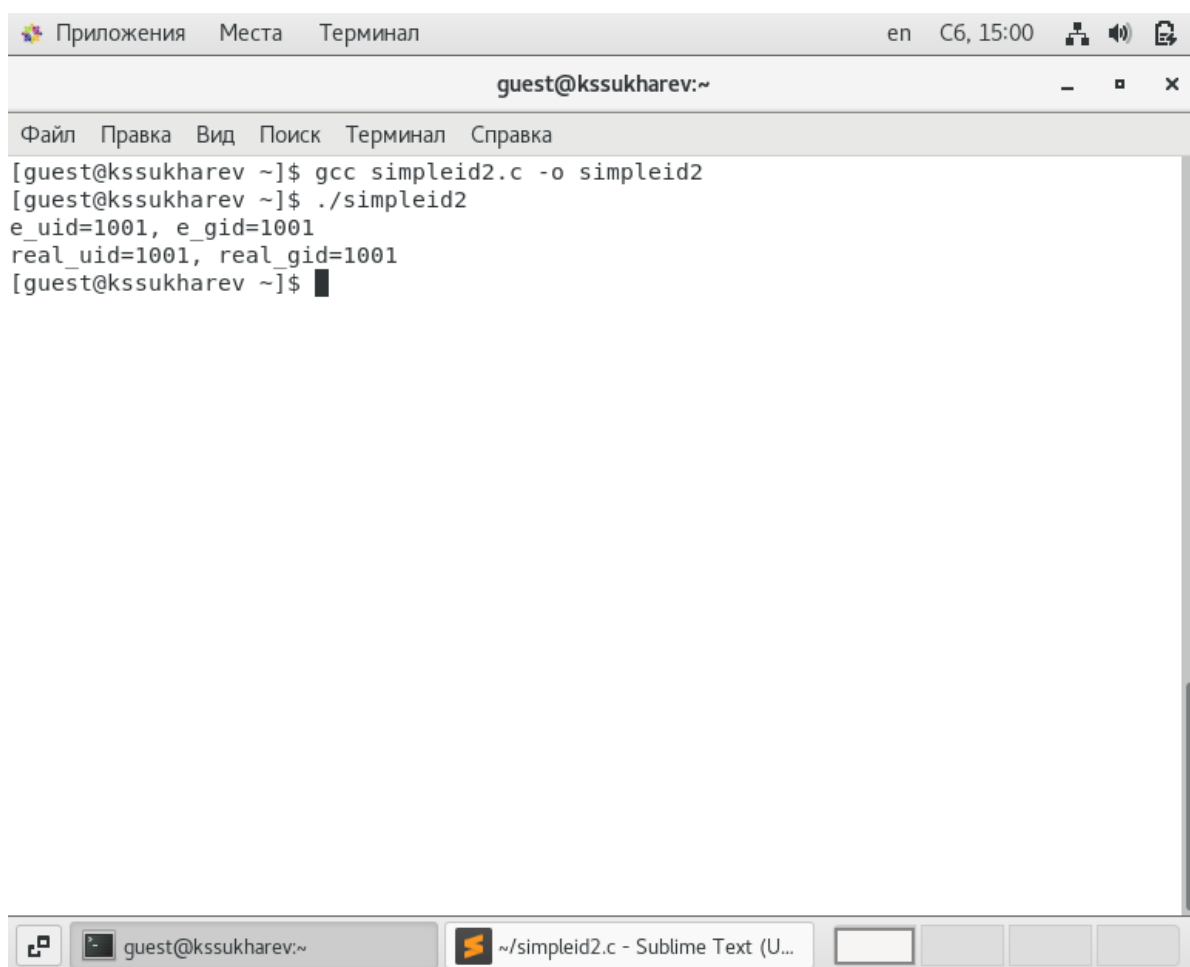


Figure 0.3: Создание simpleid2.c

4. Скомпилируем и запустим файл simpleid2 (fig. 0.4).



The image shows a terminal window titled "guest@kssukharev:~". The window has a menu bar with "Файл", "Правка", "Вид", "Поиск", "Терминал", and "Справка". The terminal output is as follows:

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

The terminal window is part of a desktop environment. The top bar shows "Приложения", "Места", "Терминал", "en", "C6, 15:00", and system icons. The bottom taskbar shows icons for a window manager, the terminal, and a Sublime Text editor window titled "~/simpleid2.c - Sublime Text (U...".

Figure 0.4: Создание simpleid2.c

5. Выполним по отношению к файлу simpleid2 команды chown и chmod. Команда chown меняет владельца и группу файла. То есть в данном случае мы устанавливаем файлу simpleid2 владельца root и группу guest. Командой chmod u+s устанавливается SetUID-бит (fig. 0.5).

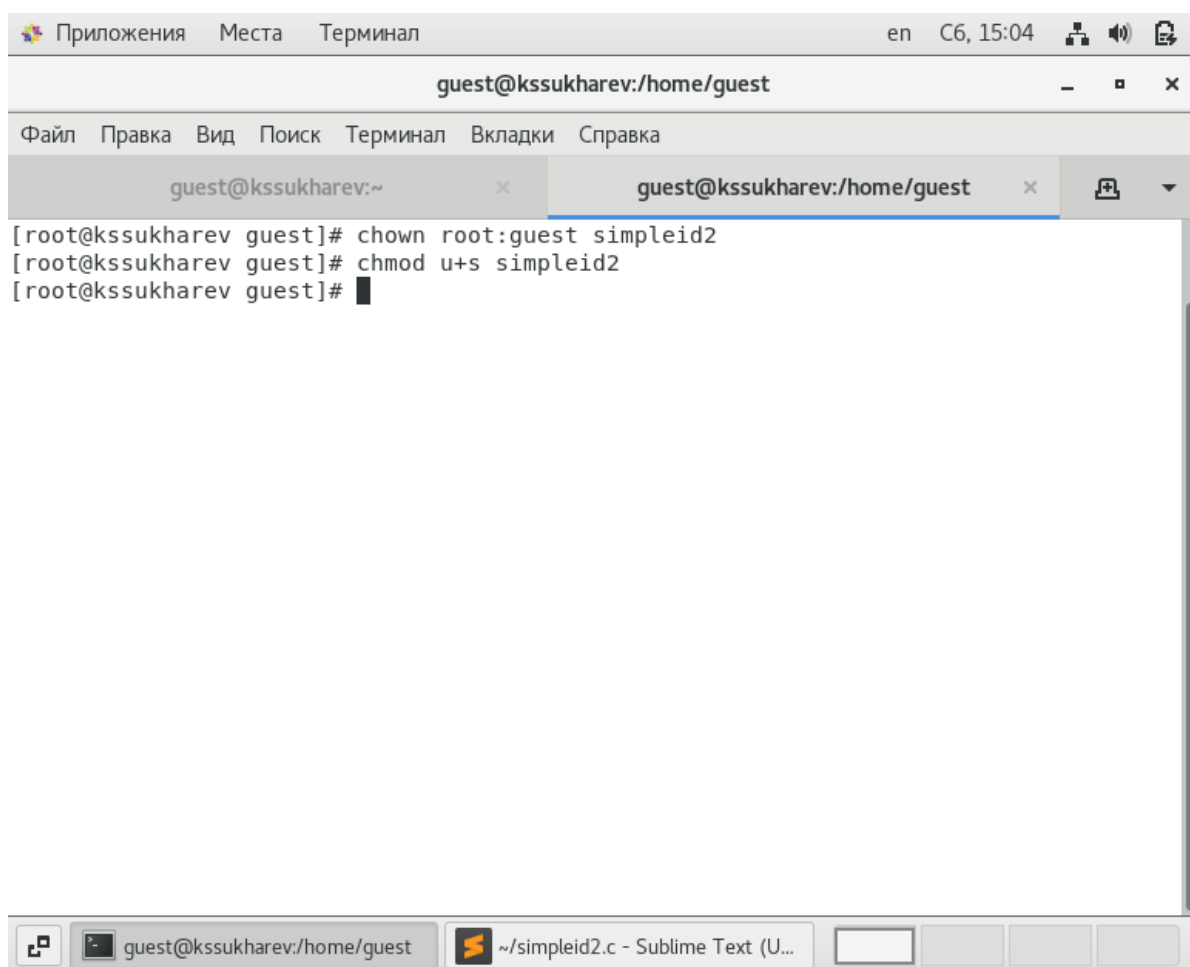


Figure 0.5: chown и chmod

6. Проверим правильность выполненных командой при помощи `ls -l`. Видим, что новые атрибуты и владелец файла были выполнены корректно (fig. 0.6).

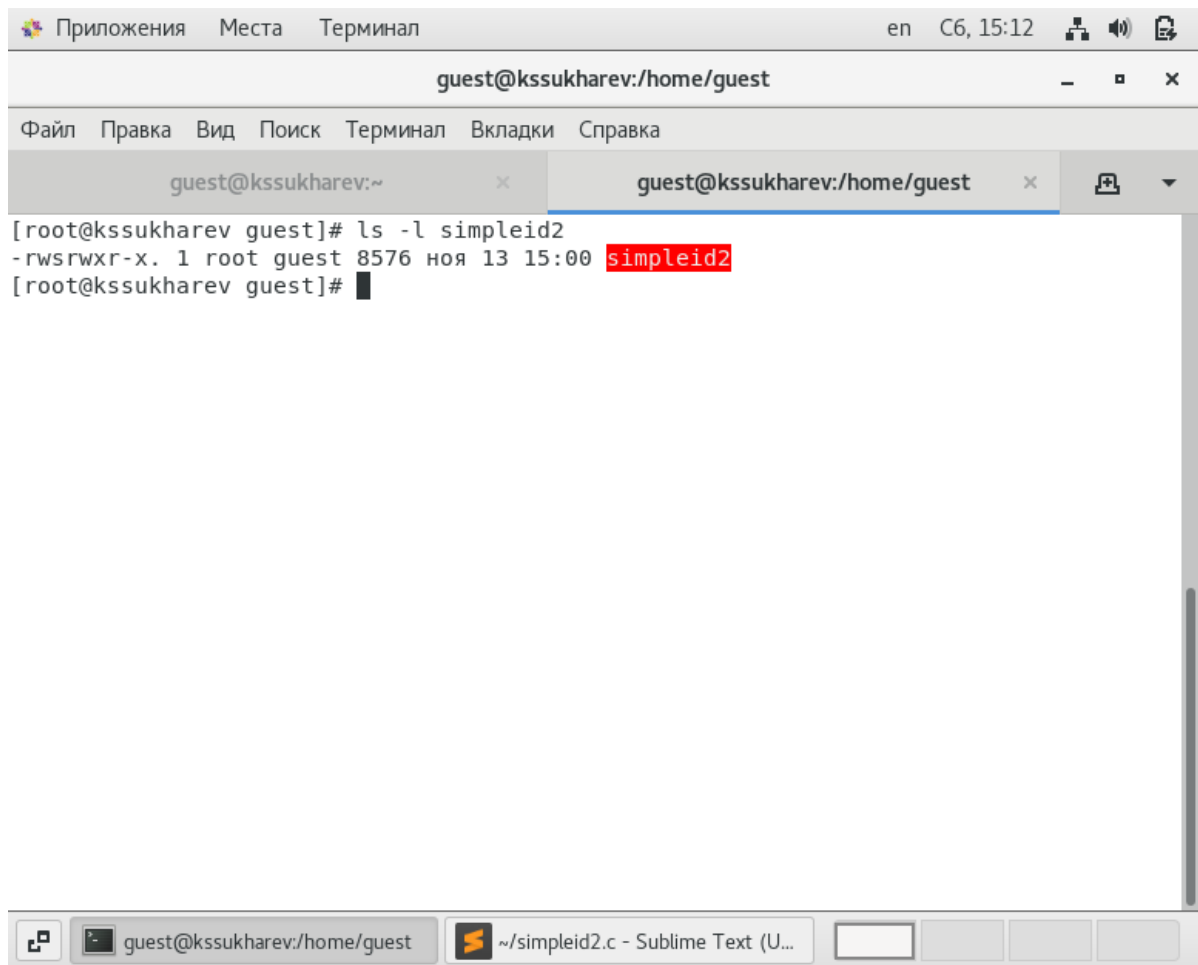
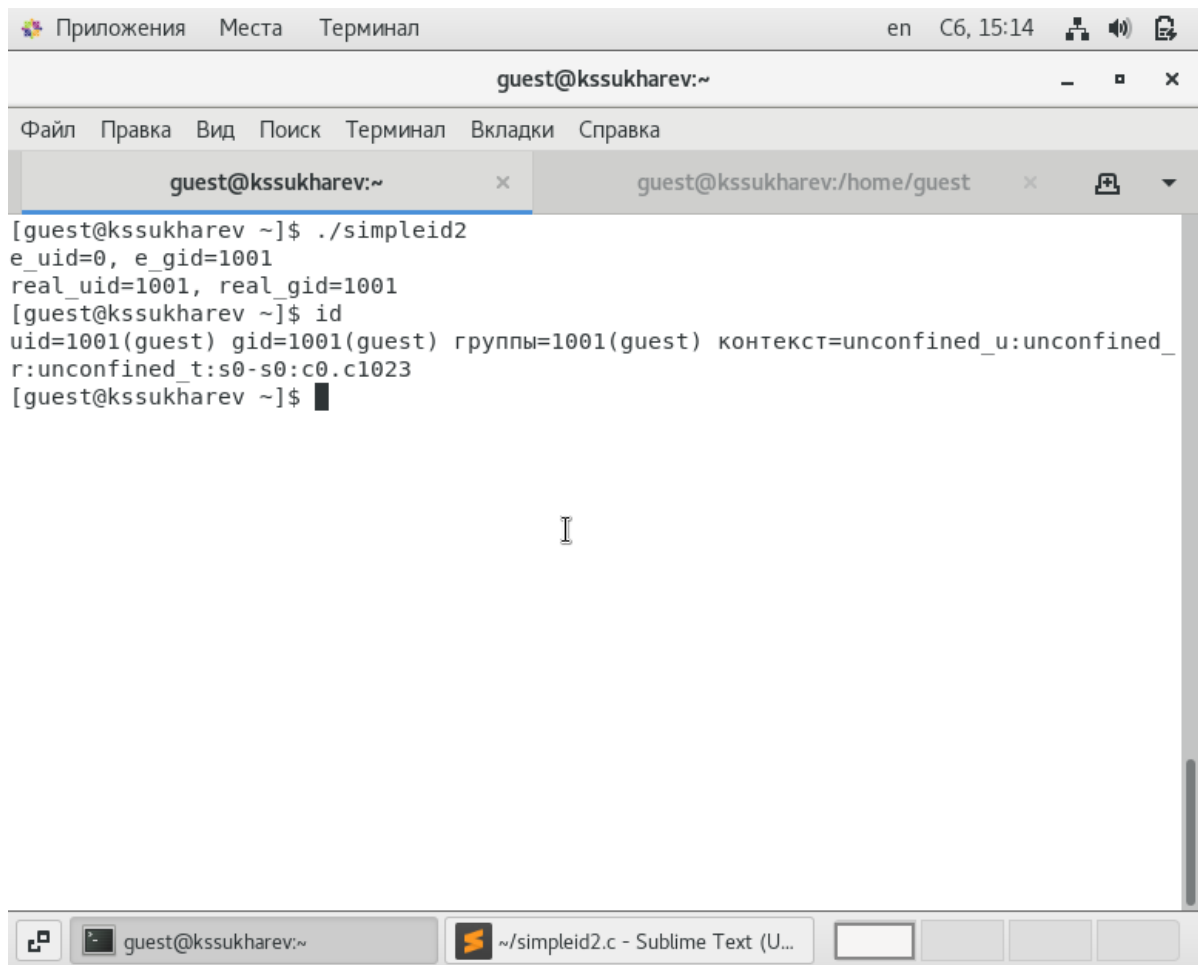


Figure 0.6: Проверка правильности

7. Запустим `simpleid2` и `id`. Видим, что `real_uid` и `real_gid` соответствуют данным `id`, а `SetUID`-бит установлен в 0 (суперпользователь) (fig. 0.7).



The screenshot shows a terminal window titled "guest@kssukharev:~". The window has a menu bar with "Файл", "Правка", "Вид", "Поиск", "Терминал", "Вкладки", and "Справка". The terminal content is as follows:

```
[guest@kssukharev ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@kssukharev ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@kssukharev ~]$
```

The terminal window has a taskbar at the bottom with icons for a window, the terminal, and a file named "~/simpleid2.c - Sublime Text (U...".

Figure 0.7: Выполнение simpleid2

8. Проделаем то же самое для SetGID-бита. Для этого выполним команду `chmod g+s`. Снова выполним `simpleid2` и убедимся, что группа файла равно `1001(guest)` (fig. 0.8).

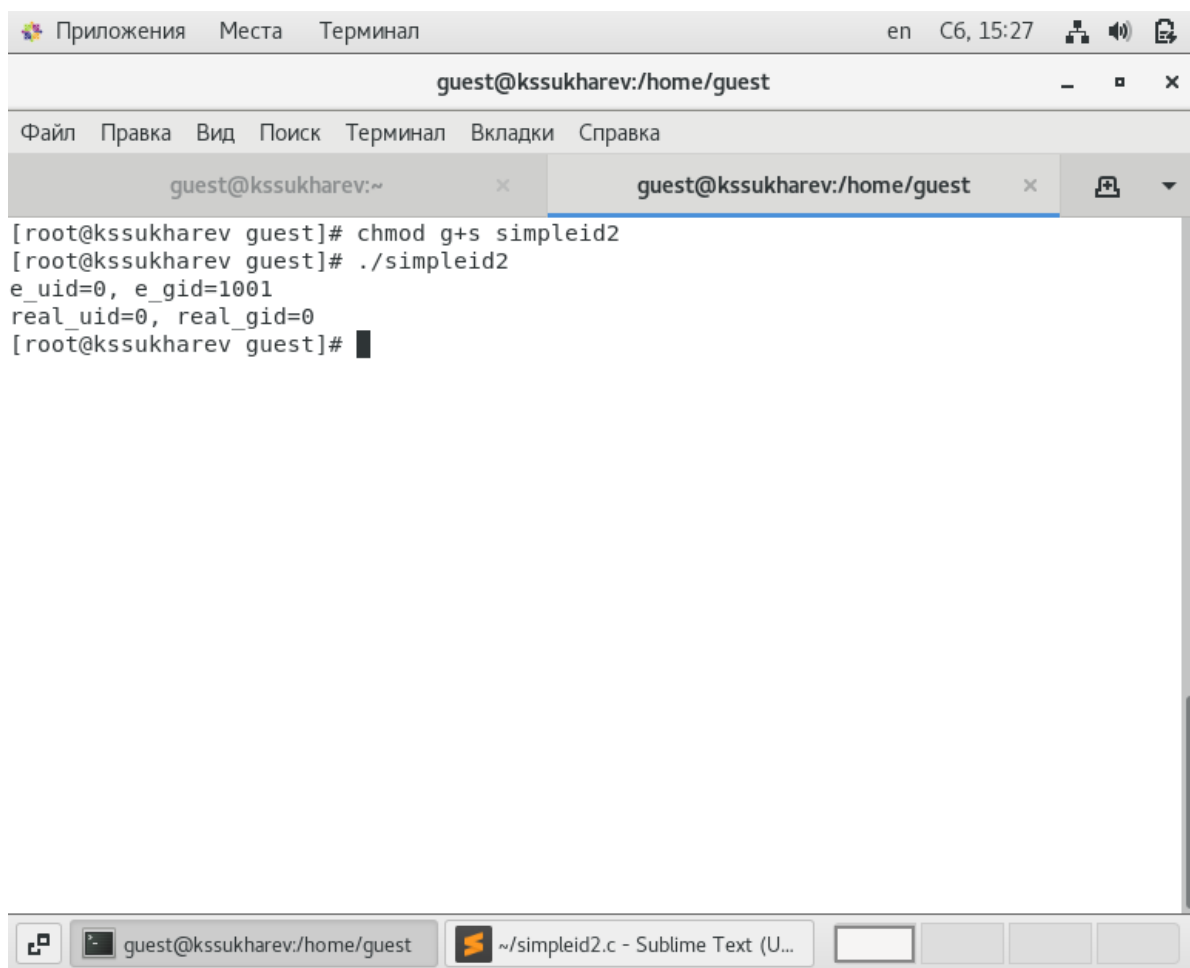


Figure 0.8: Установка SetGID-бита

9. Создадим программу readfile.c (fig. 0.9).

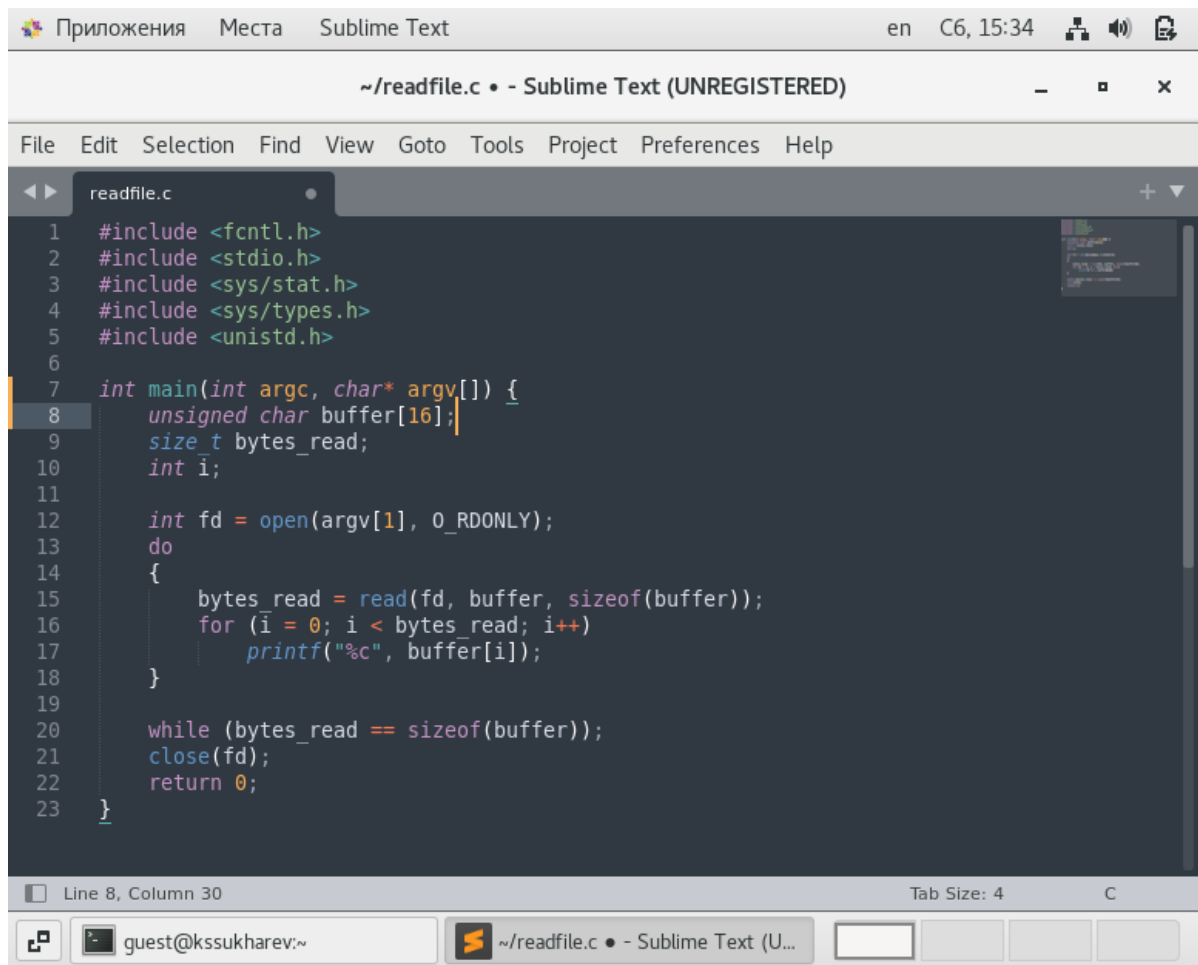


Figure 0.9: Создание readfile.c

10. Откомпилируем ее (fig. 0.10).

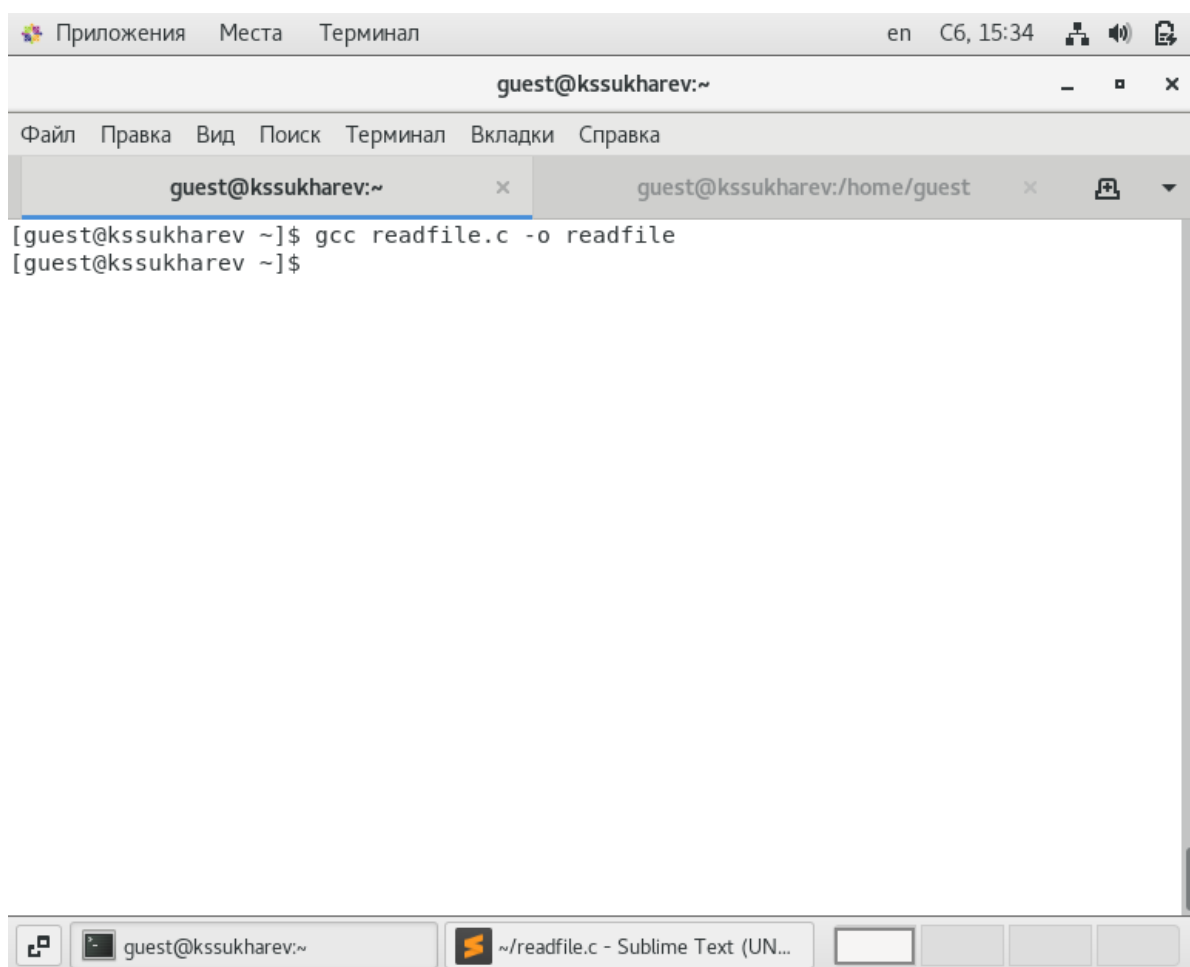


Figure 0.10: Компиляция файла `readfile.c`

11. Изменим права у файла `readfile.c` так, чтобы его мог прочитать только супер-пользователь (fig. 0.11).

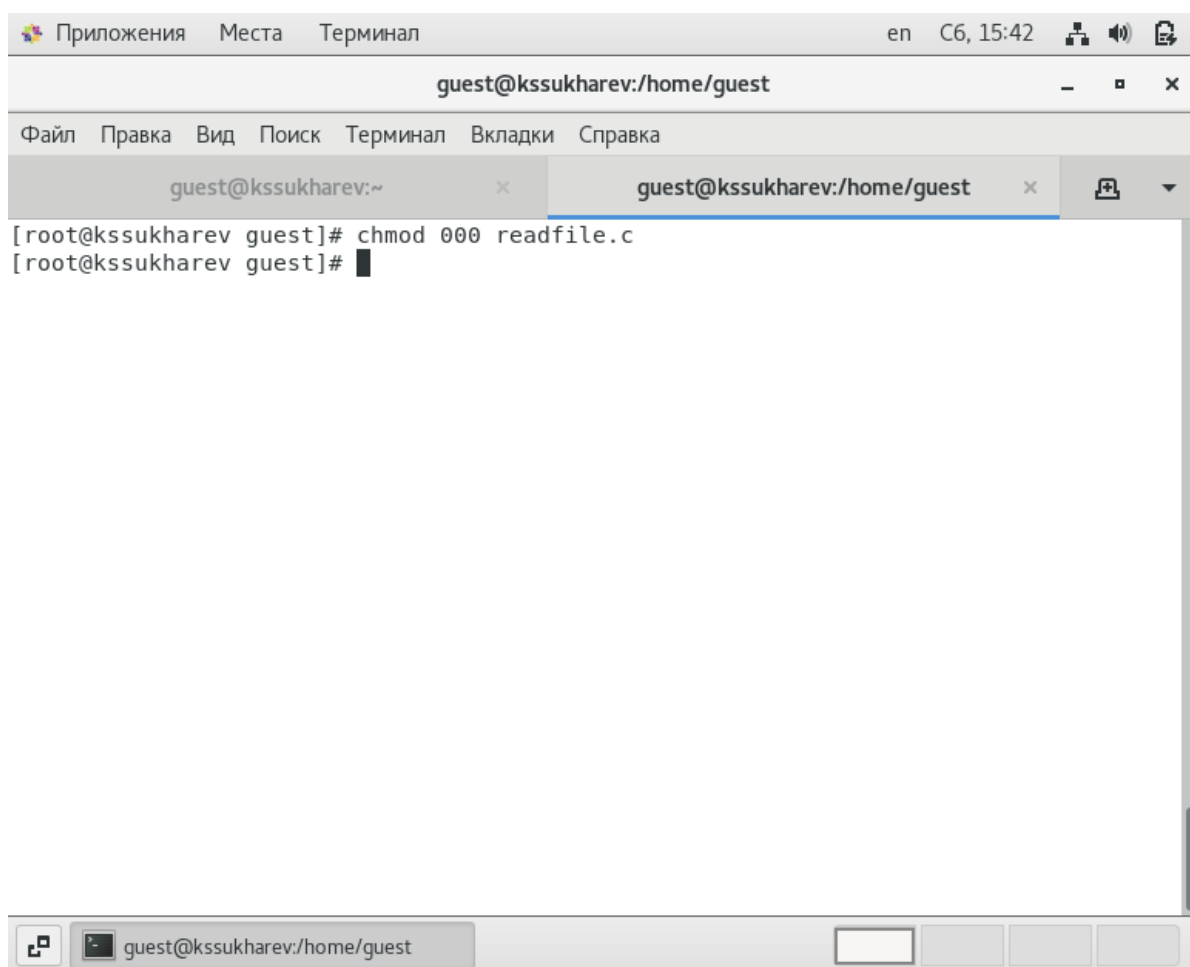


Figure 0.11: Смена прав у файла

12. Убедимся что пользователь guest не может прочитать файл readfile.c (fig. 0.12).

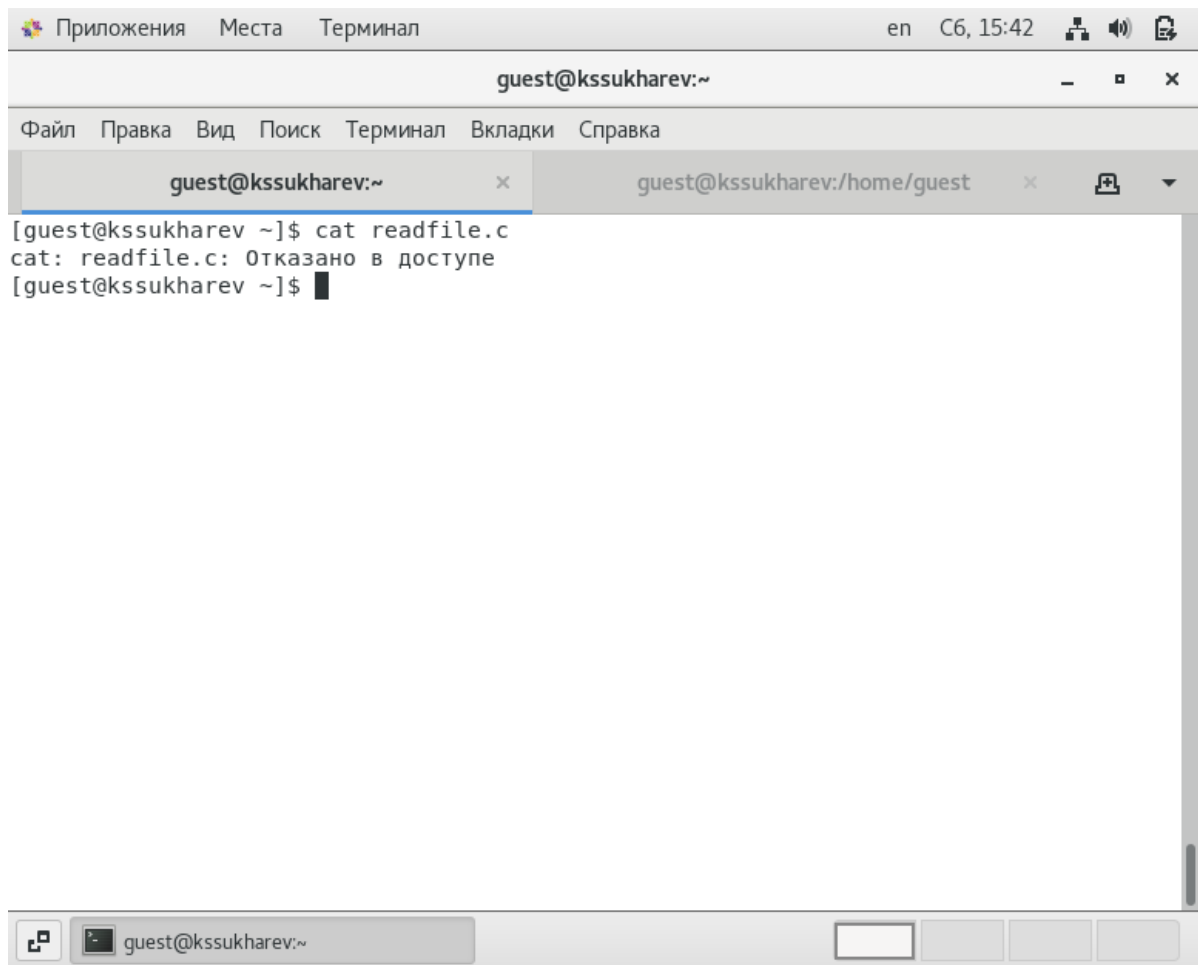


Figure 0.12: Проверка недоступности readfile.c для guest

13. Сменим владельца программы readfile и установим SetUID-бит (fig. 0.13).

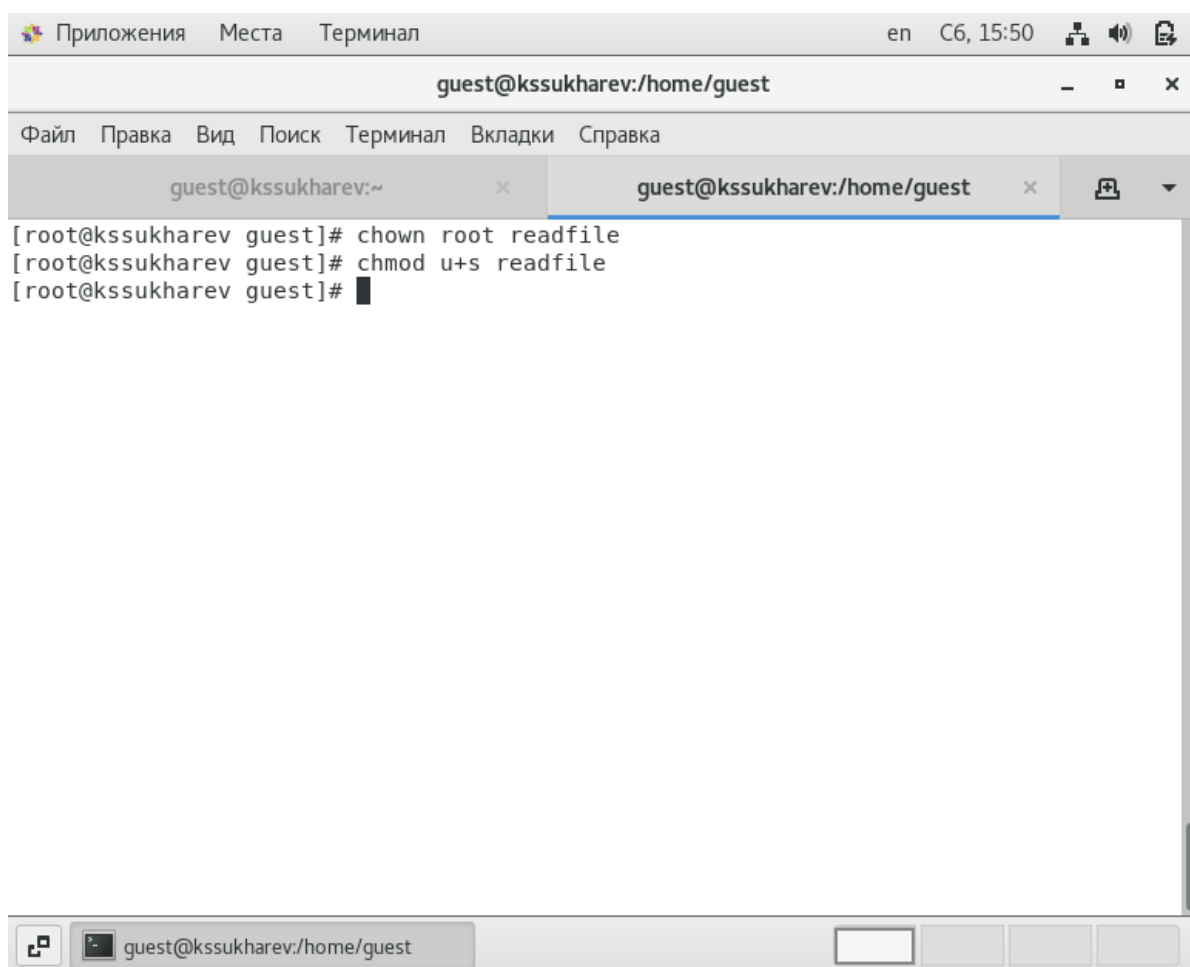
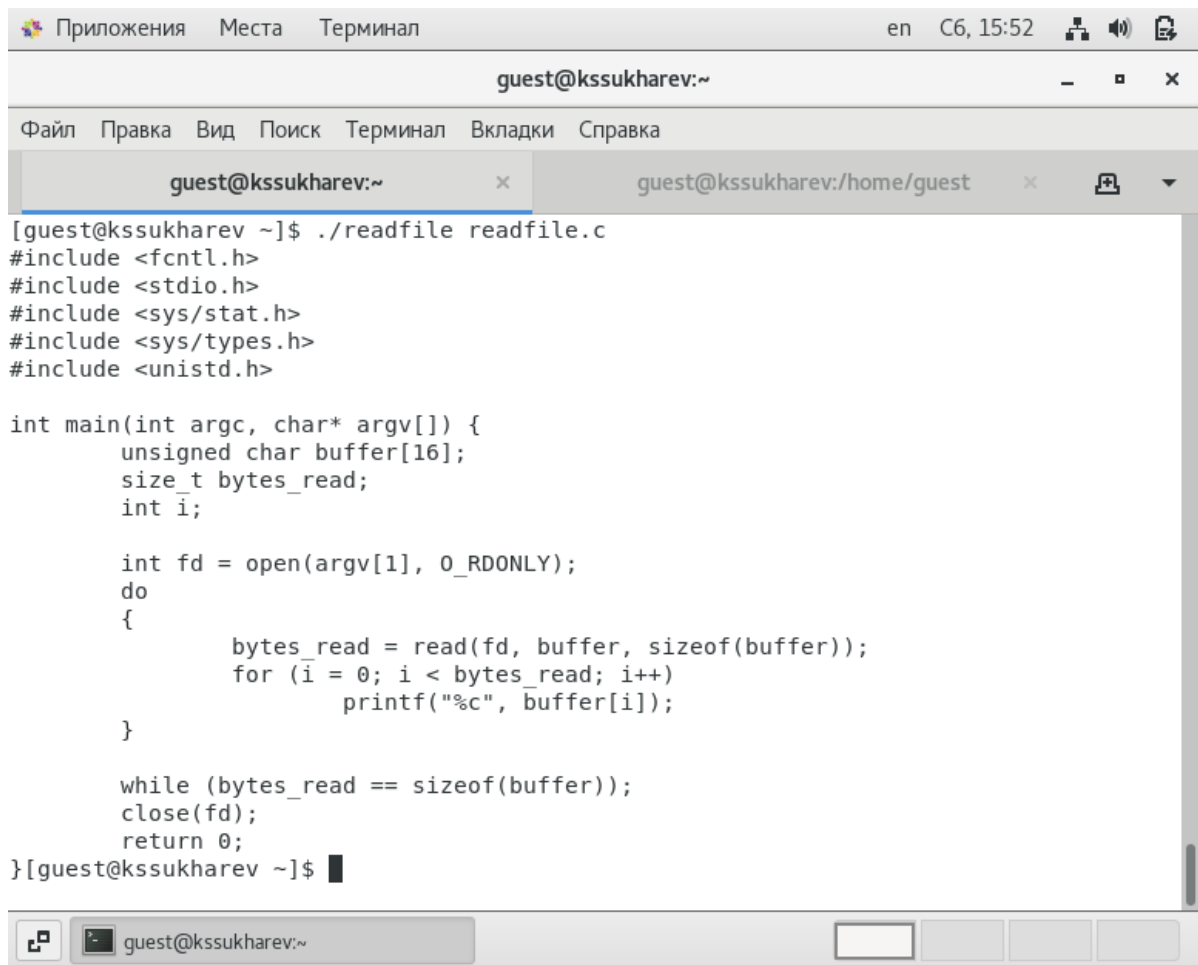


Figure 0.13: Смена владельца и установка SetUID-бита

14. Попробуем прочитать файл `readfile.c` программой `readfile`. Операция была выполнена успешно (fig. 0.14).



```
Приложения  Места  Терминал  en  C6, 15:52  [system icons]
guest@kssukharev:~
Файл  Правка  Вид  Поиск  Терминал  Вкладки  Справка
guest@kssukharev:~  x  guest@kssukharev:/home/guest  x  [icons]
[guest@kssukharev ~]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

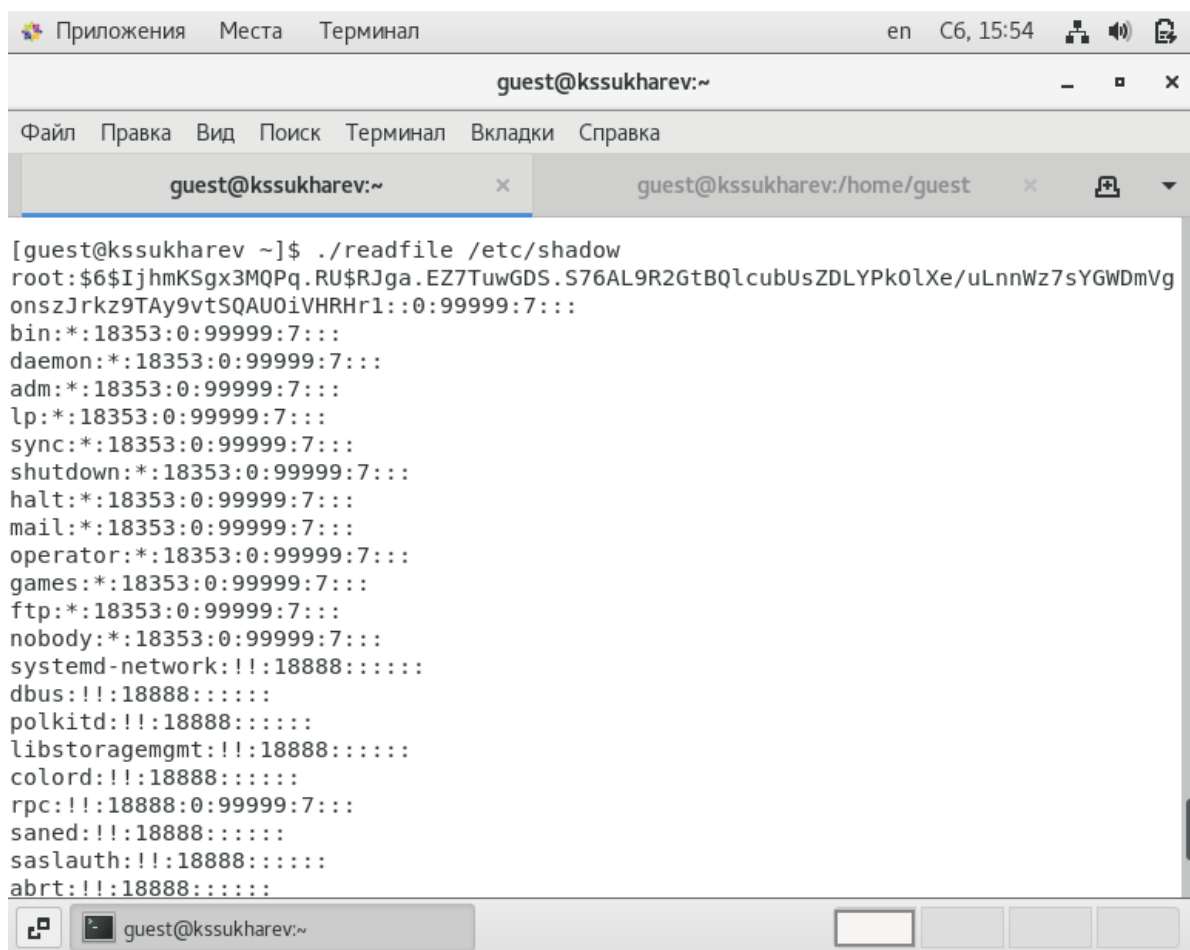
int main(int argc, char* argv[]) {
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    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@kssukharev ~]$
```

Figure 0.14: Попытка прочитать readfile.c

15. Попробуем прочитать файл `etc/shadow` программой `readfile`. Снова все прошло успешно (fig. 0.15).



```
[guest@kssukharev ~]$ ./readfile /etc/shadow
root:$6$IjhmKSgx3MQPq.RU$RJga.EZ7TuwGDS.S76AL9R2GtBQlcubUsZDLYPk0lXe/uLnnWz7sYGWDmVg
onszJrkz9TAy9vtSQAU0iVHRHr1::0:99999:7:::
bin:*:18353:0:99999:7:::
daemon:*:18353:0:99999:7:::
adm:*:18353:0:99999:7:::
lp:*:18353:0:99999:7:::
sync:*:18353:0:99999:7:::
shutdown:*:18353:0:99999:7:::
halt:*:18353:0:99999:7:::
mail:*:18353:0:99999:7:::
operator:*:18353:0:99999:7:::
games:*:18353:0:99999:7:::
ftp:*:18353:0:99999:7:::
nobody:*:18353:0:99999:7:::
systemd-network:!!:18888:~::~:
dbus:!!:18888:~::~:
polkitd:!!:18888:~::~:
libstoragemgmt:!!:18888:~::~:
colord:!!:18888:~::~:
rpc:!!:18888:0:99999:7:::
saned:!!:18888:~::~:
saslauth:!!:18888:~::~:
abrt:!!:18888:~::~:
```

Figure 0.15: Попытка прочитать `etc/shadow`

Исследование Sticky-бита

1. Выясним, установлен ли атрибут Sticky на директории `/tmp`. По результатам выполнения команды `ls -l` видим, что Sticky-бит установлен (fig. 0.16).

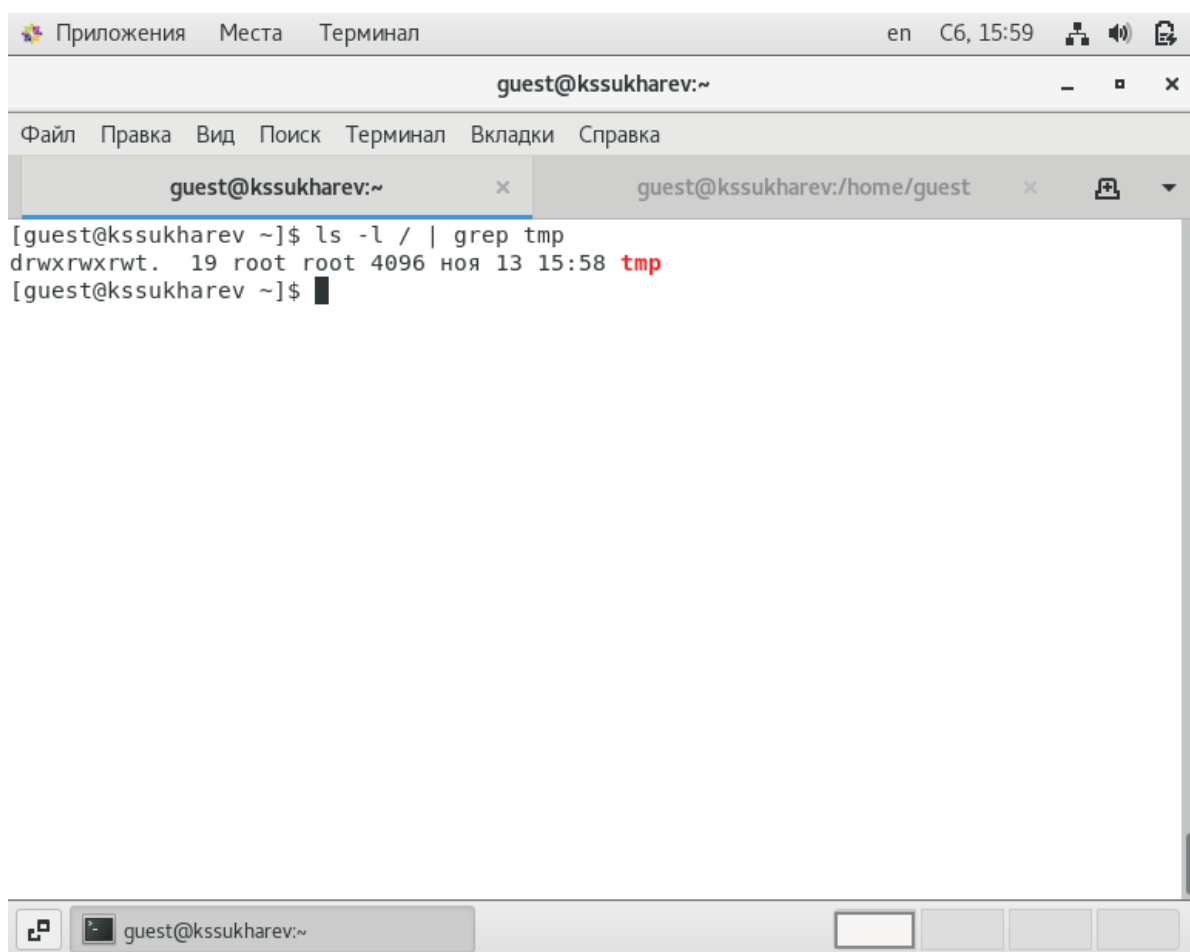


Figure 0.16: Проверка наличия атрибута Sticky

2. Создадим в директории `/tmp` файл `file01.txt`. Посмотрим атрибуты этого файла, а затем разрешим остальным пользователям чтение и запись (fig. 0.17).

The screenshot shows a terminal window titled "guest@kssukharev:/tmp". The window has a menu bar with "Файл", "Правка", "Вид", "Поиск", "Терминал", "Вкладки", and "Справка". The terminal content shows the following commands and output:

```
[guest@kssukharev tmp]$ echo "test" > file01.txt
[guest@kssukharev tmp]$ ls -l file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 17:11 file01.txt
[guest@kssukharev tmp]$ chmod o+rw file01.txt
[guest@kssukharev tmp]$ ls -l file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 17:11 file01.txt
[guest@kssukharev tmp]$
```

The terminal window has a tab bar at the top with two tabs: "guest@kssukharev:/tmp" (active) and "guest@kssukharev:/home/guest". The bottom status bar shows the active tab and some system icons.

Figure 0.17: Создание file01.txt

3. От имени пользователя guest2 попробуем прочитать созданный файл. Никаких ошибок не возникло (fig. 0.18).

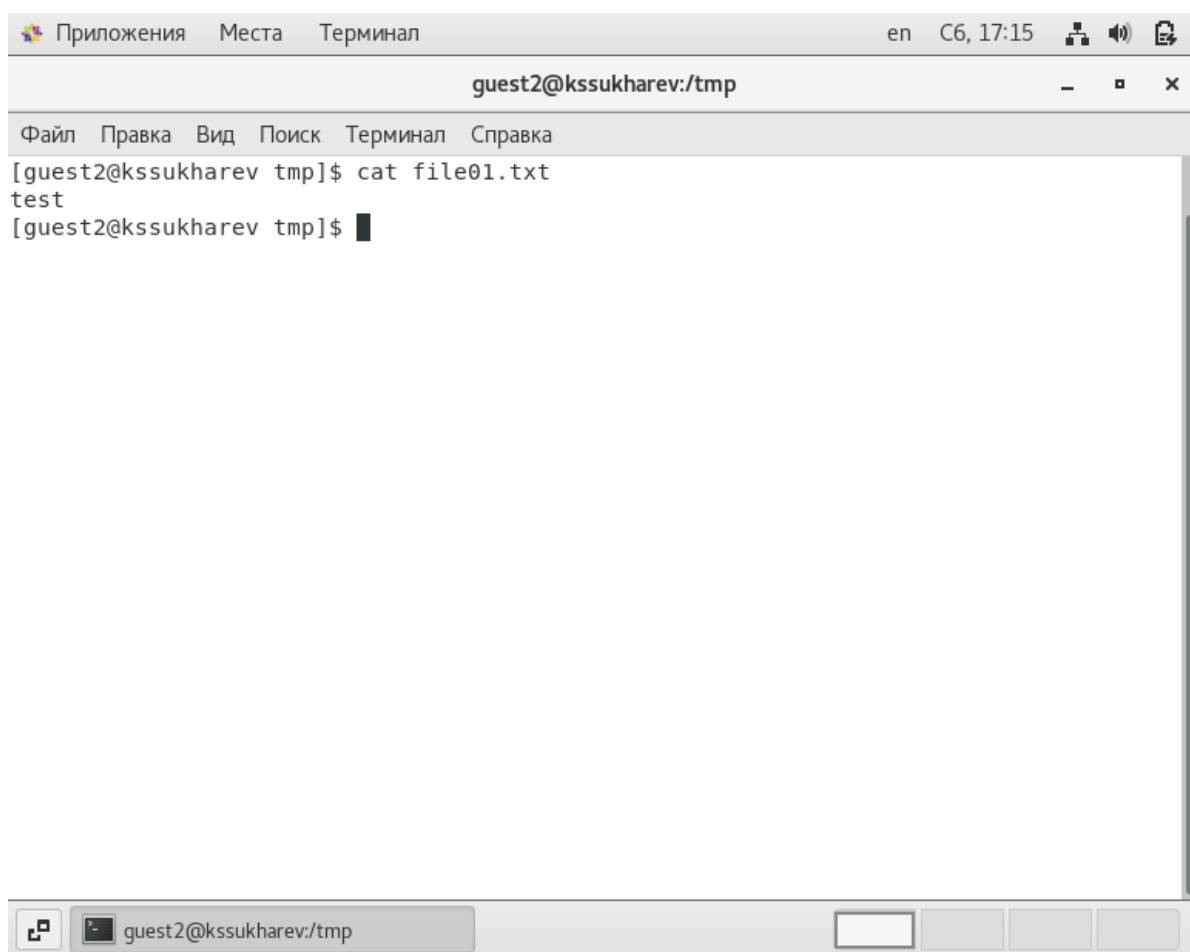


Figure 0.18: Попытка чтения file01.txt

4. Теперь попробуем дозаписать в этот файл слово test2. Как можно видеть, дозапись прошла успешно (fig. 0.19).

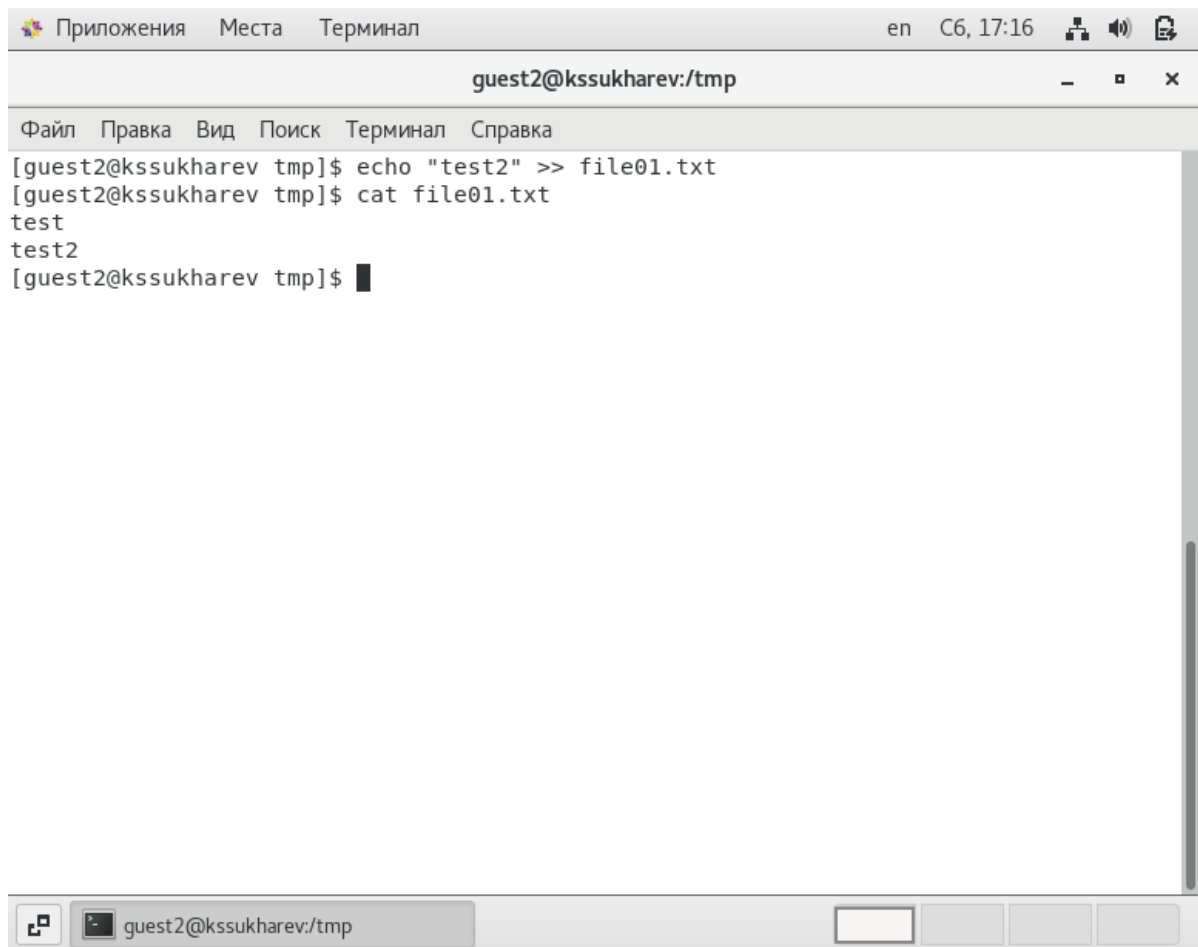


Figure 0.19: Дозапись file01.txt

5. Попробуем заменить содержимое файла на "test3". Видим, что перезапись файла также прошла успешно (fig. 0.20).

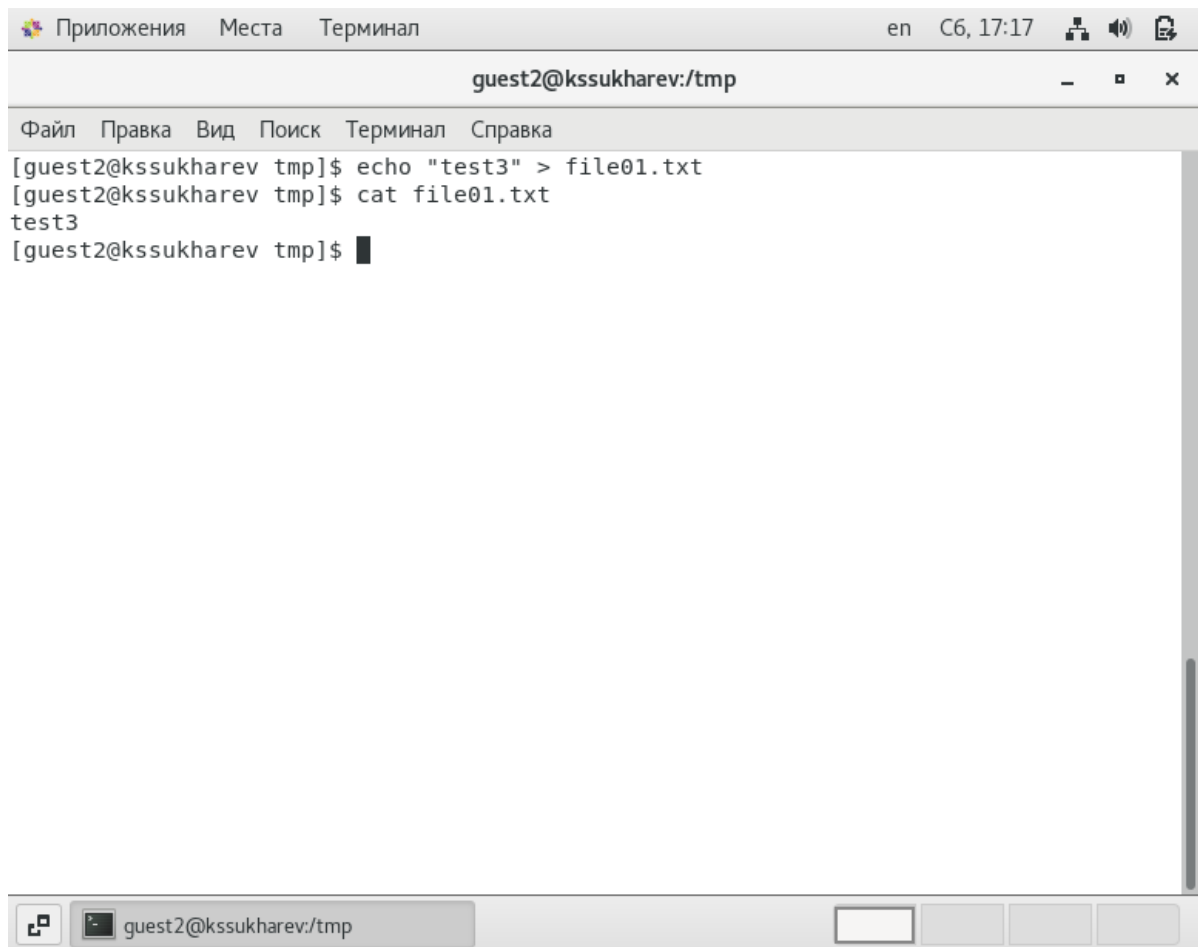


Figure 0.20: Перезапись file01.txt

6. Попробуем удалить файл. Данная операция не позволена (fig. 0.21).

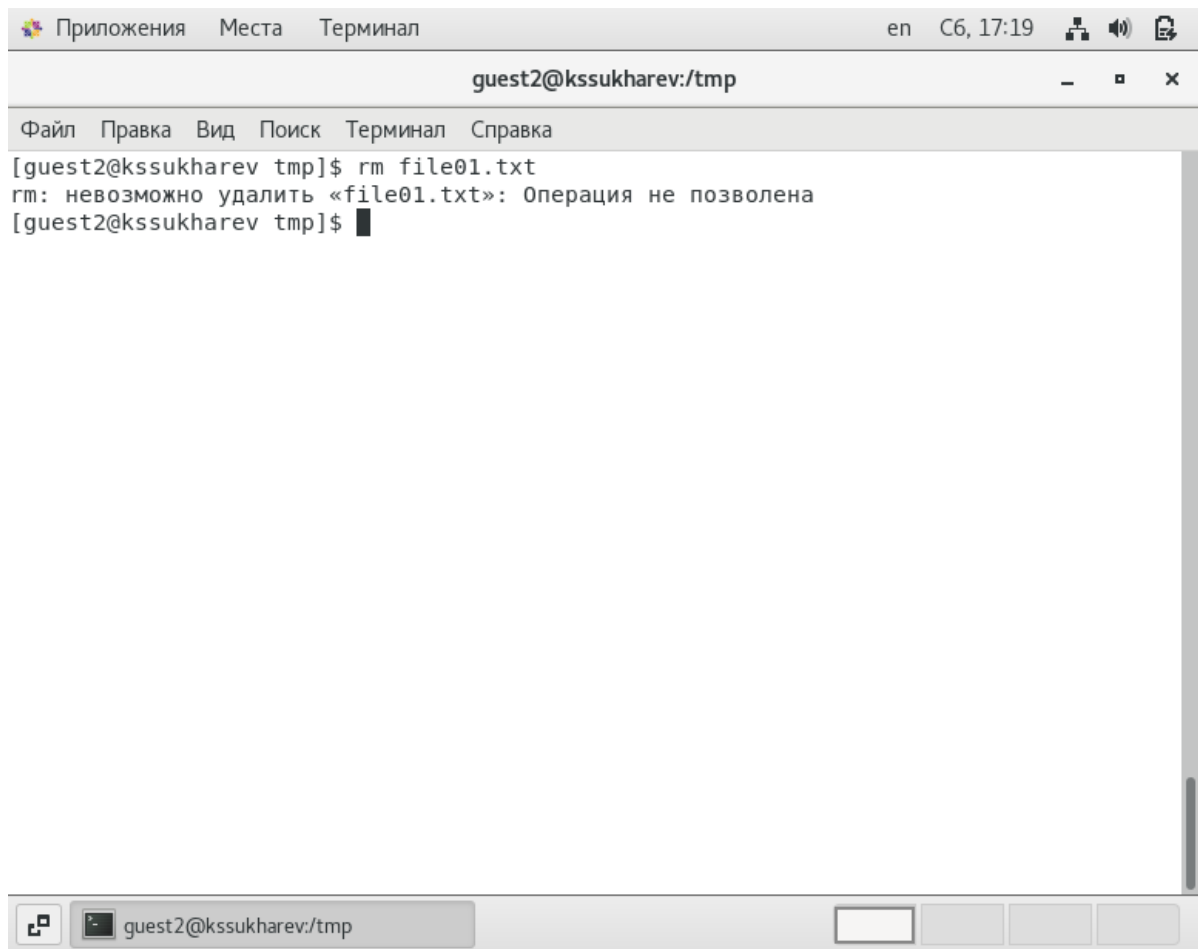


Figure 0.21: Попытка удаления file01.txt

7. Повысим свои права до суперпользователя и снимем Sticky-бит с директории /tmp (fig. 0.22).

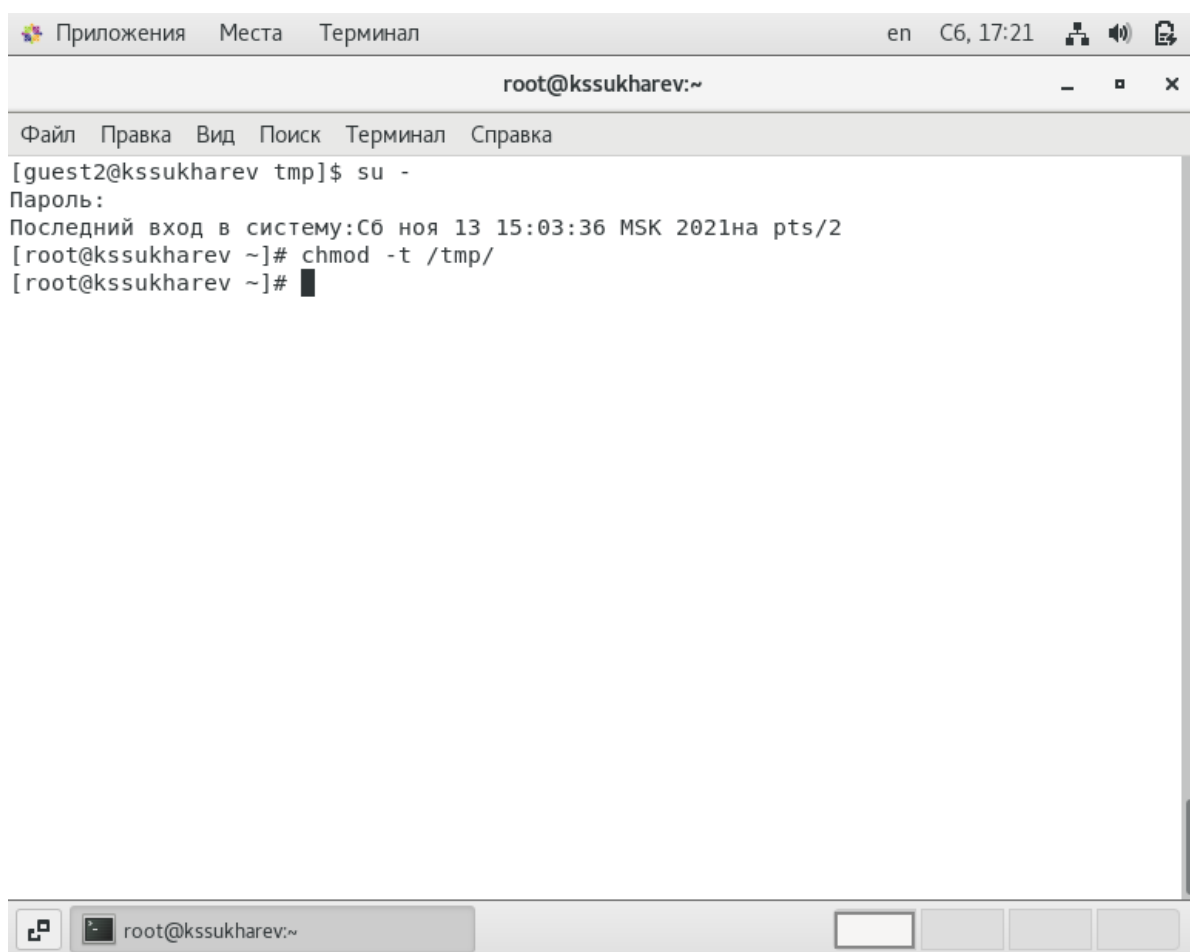
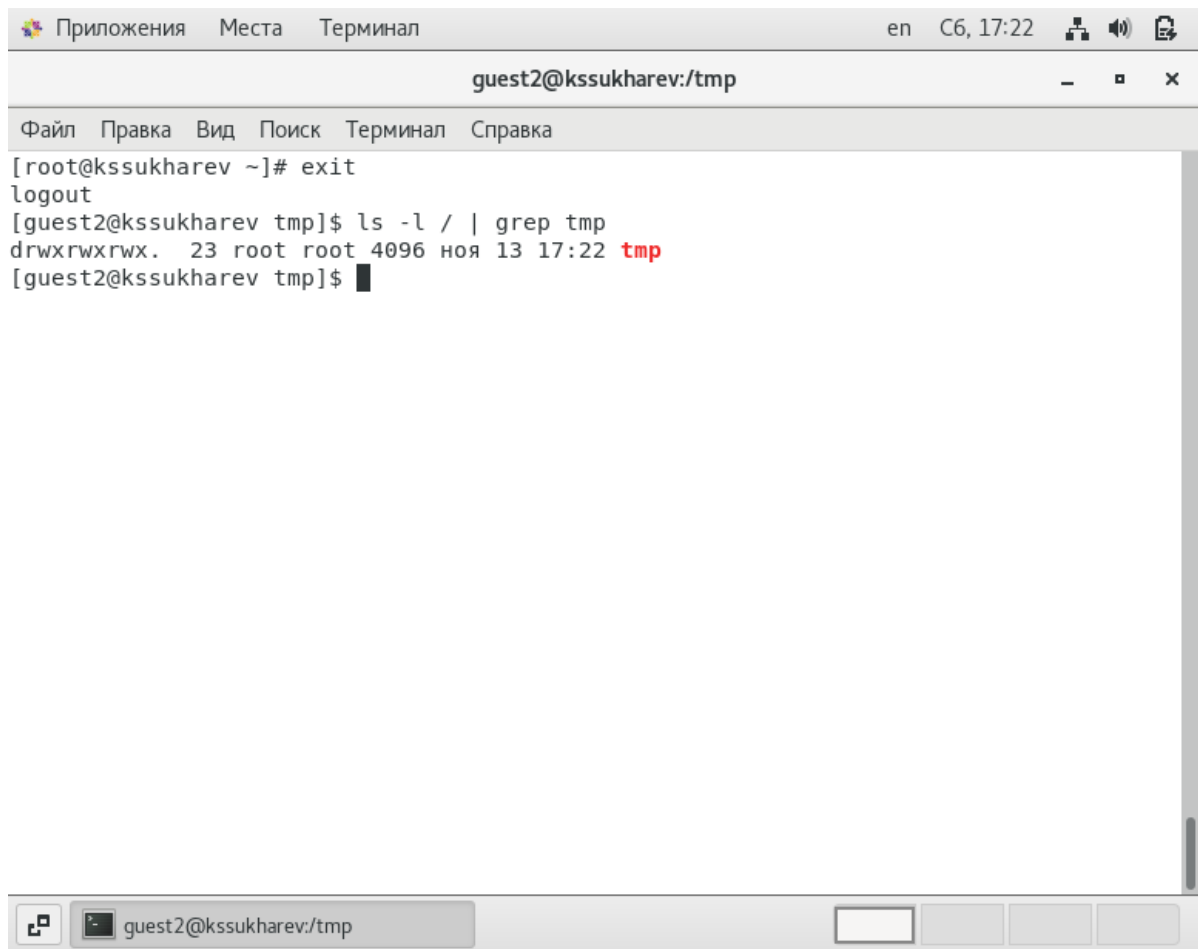


Figure 0.22: Снятие Sticky-бита

8. Выйдем из режима суперпользователя и убедимся, что атрибута `t` у директории `/tmp` больше нет (fig. 0.23).



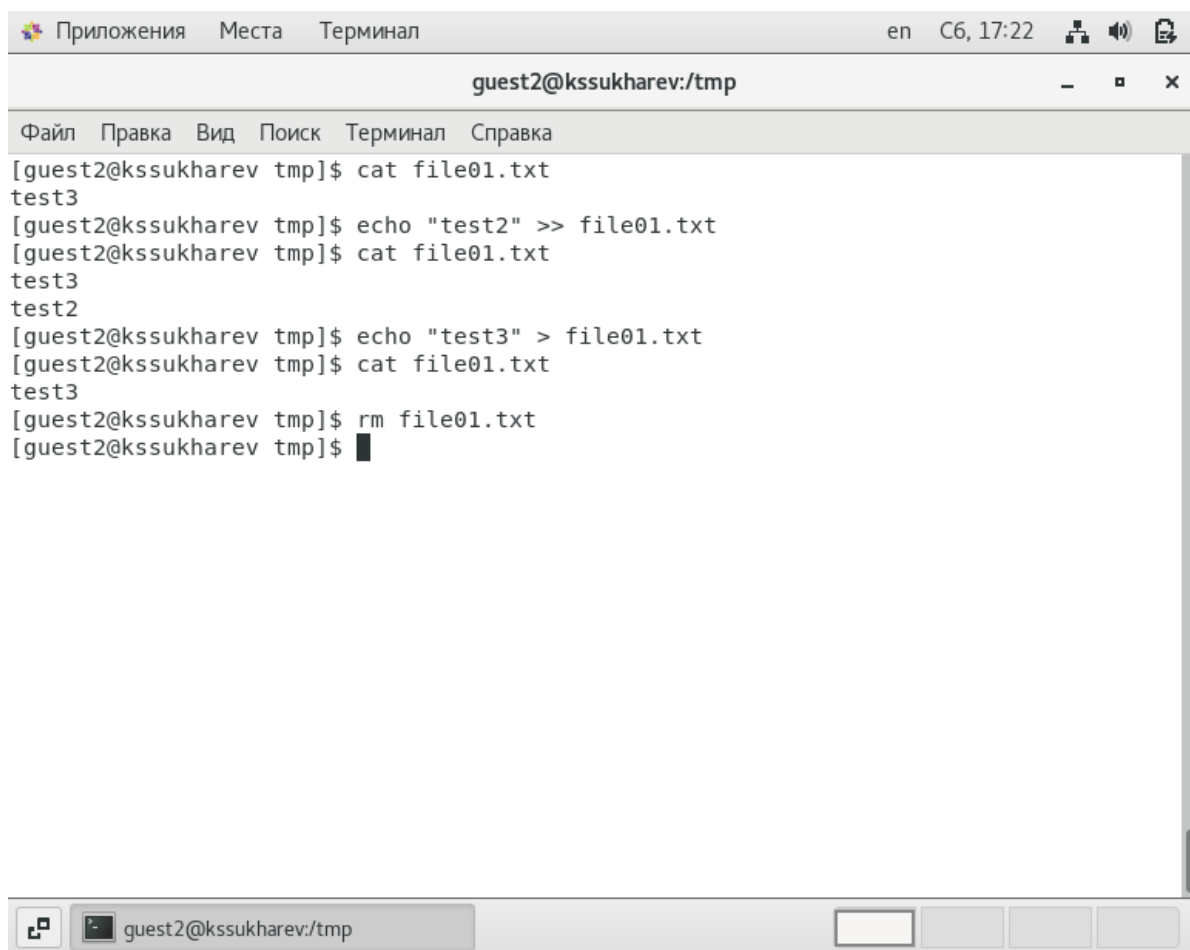
The screenshot shows a terminal window titled "guest2@kssukharev:/tmp". The window has a menu bar with "Файл", "Правка", "Вид", "Поиск", "Терминал", and "Справка". The terminal output is as follows:

```
[root@kssukharev ~]# exit
logout
[guest2@kssukharev tmp]$ ls -l / | grep tmp
drwxrwxrwx. 23 root root 4096 ноя 13 17:22 tmp
[guest2@kssukharev tmp]$
```

The terminal window has a title bar with "en C6, 17:22" and system icons. The bottom of the window shows a taskbar with a terminal icon and the text "guest2@kssukharev/tmp".

Figure 0.23: Проверка снятия Sticky-бита

9. Повторим предыдущие шаги. Видим, что все ограничения были сняты. Нам даже удалось удалить этот файл (fig. 0.24).



```
guest2@kssukharev:/tmp
[guest2@kssukharev tmp]$ cat file01.txt
test3
[guest2@kssukharev tmp]$ echo "test2" >> file01.txt
[guest2@kssukharev tmp]$ cat file01.txt
test3
test2
[guest2@kssukharev tmp]$ echo "test3" > file01.txt
[guest2@kssukharev tmp]$ cat file01.txt
test3
[guest2@kssukharev tmp]$ rm file01.txt
[guest2@kssukharev tmp]$
```

Figure 0.24: Проверка предыдущих команд

10. Вернем Sticky-бит на директорию /tmp (fig. 0.25).

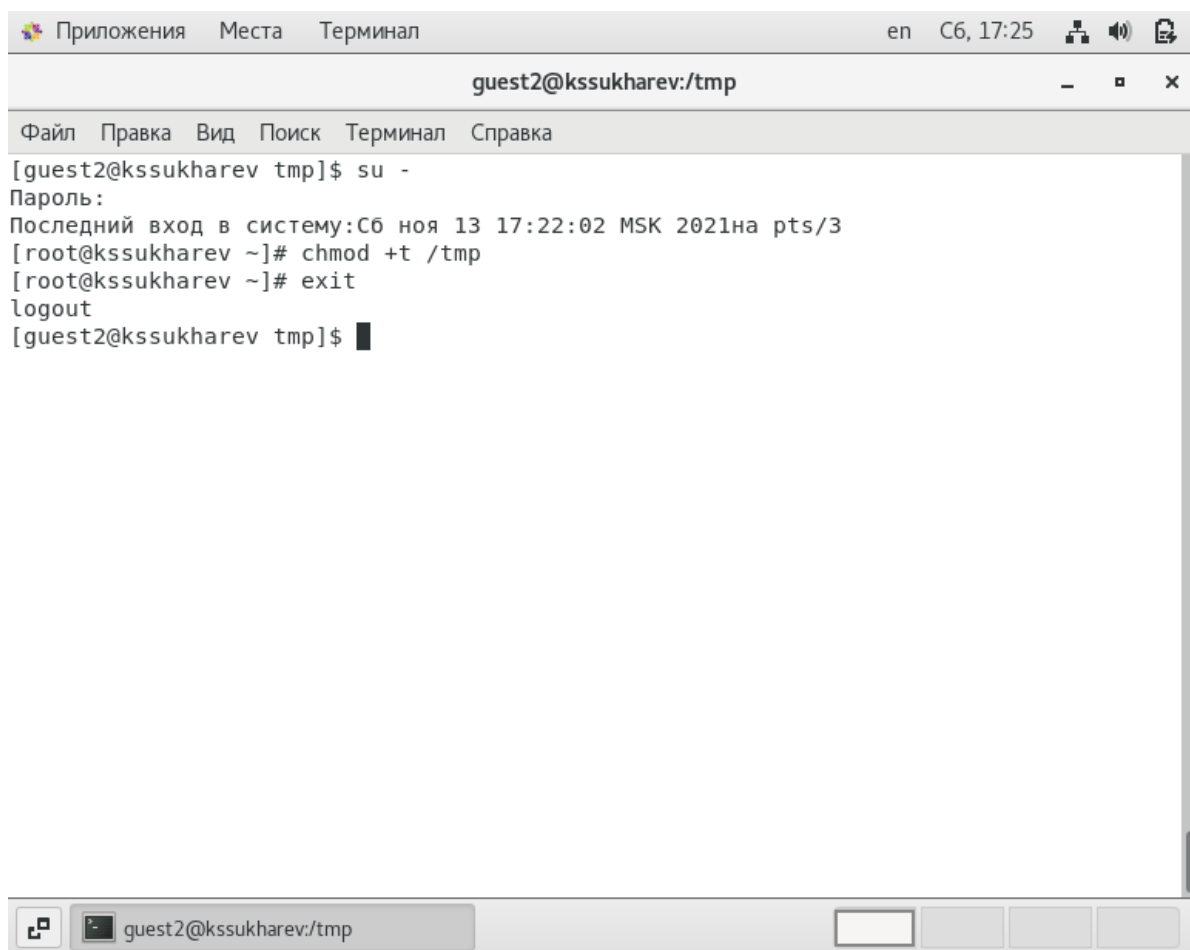


Figure 0.25: Проверка предыдущих команд

Выводы

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

Библиография

1. Права доступа и атрибуты файла. Команды `chown`, `chmod` и `chattr` // Вики-Чтение. URL: <https://it.wikireading.ru/38589> (Дата обращения: 13.11.2021).
2. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. М.: РУДН, 2015. 64 с..