Информационная безопасность

Л.5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Ширяев Кирилл Владимирович

2021

Содержание

[Цель работы 2](#_Toc96437920)

[Подготовка лабораторного стенда 2](#_Toc96437921)

[Ход работы 2](#_Toc96437924)

[Программа simpleid.c 2](#_Toc96437925)

[Компиляция simpleid.c 3](#_Toc96437926)

[Выполнение simpleid.c и id 4](#_Toc96437927)

[Программа simpleid2.c 5](#_Toc96437928)

[Компиляция и выполнение simpleid2.c 6](#_Toc96437929)

[Смена владельца и атрибут s 7](#_Toc96437930)

[Запуск simpleid2 с SetUID 8](#_Toc96437931)

[Программа readfile.c 9](#_Toc96437933)

[Компиляция и изменение readfile.c 10](#_Toc96437934)

[Атрибут Sticky на /tmp 11](#_Toc96437937)

[Создание file01.txt 12](#_Toc96437938)

[Атрибуты file01.txt 13](#_Toc96437939)

[Работа с файлом file01.txt в директории с t 14](#_Toc96437940)

[Снятие атрибута t 15](#_Toc96437941)

[Работа с файлом file01.txt в директории без t 16](#_Toc96437942)

[Возврат атрибута t 17](#_Toc96437943)

[Вывод 18](#_Toc96437944)

# 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.

# Подготовка лабораторного стенда

# Ход работы

## Программа simpleid.c

Вошли в систему от имени пользователя guest и создали программу simpleid.c:

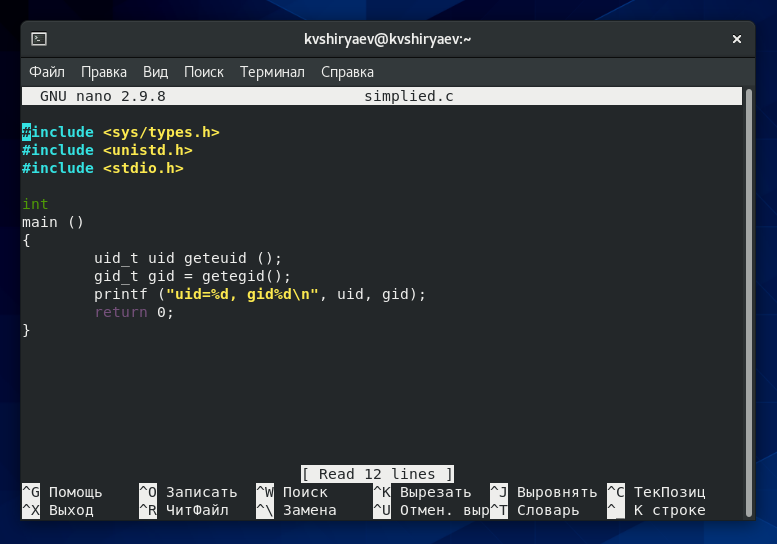
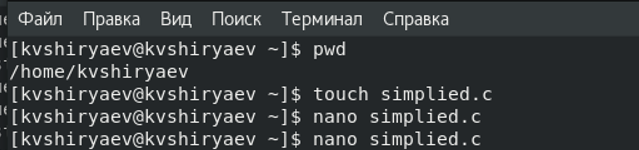


Figure 3: Программа simpleid.c

## Компиляция simpleid.c

Скомплилировали программу и убедились, что файл программы создан:

gcc simpleid.c -o simpleid

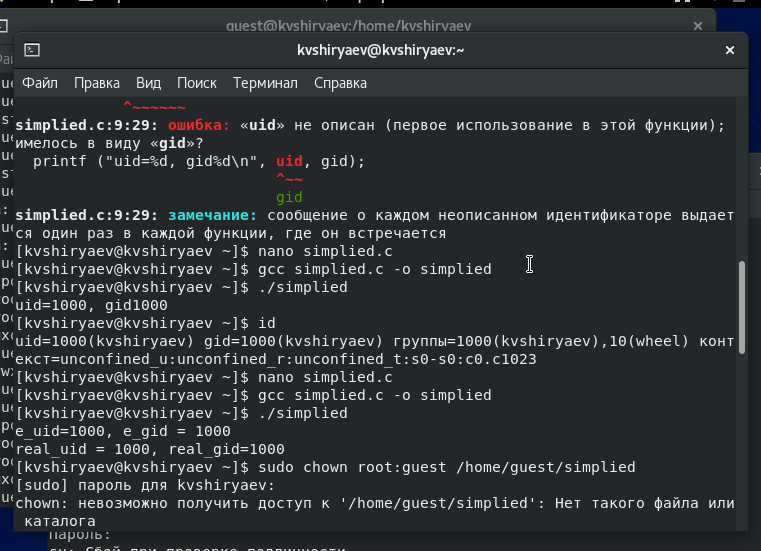


Figure 4: Компиляция simpleid.c

## Выполнение simpleid.c и id

Выполнили программу simpleid: ./simpleid, а также системную программу id:

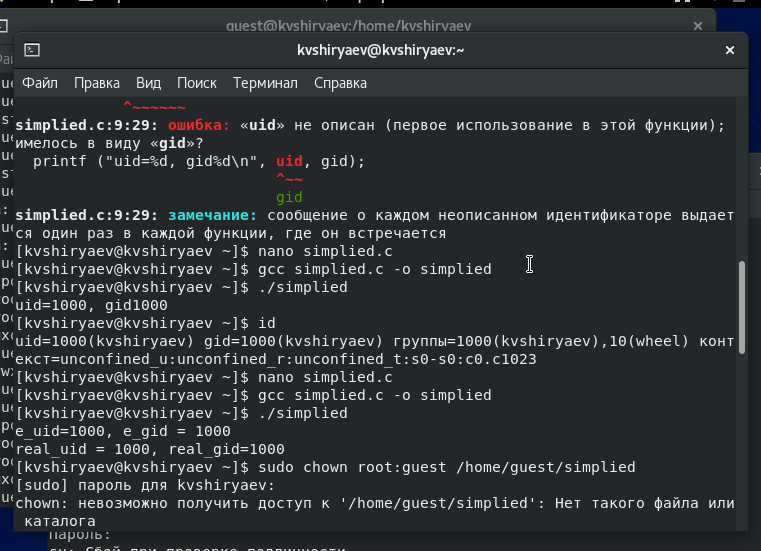


Figure 5: Выполнение simpleid.c и id

Получили идентичные результаты рассматриваемых параметров

## Программа simpleid2.c

Усложнили программу, добавив вывод действительных идентификаторов, и назвали её simpleid2.c:

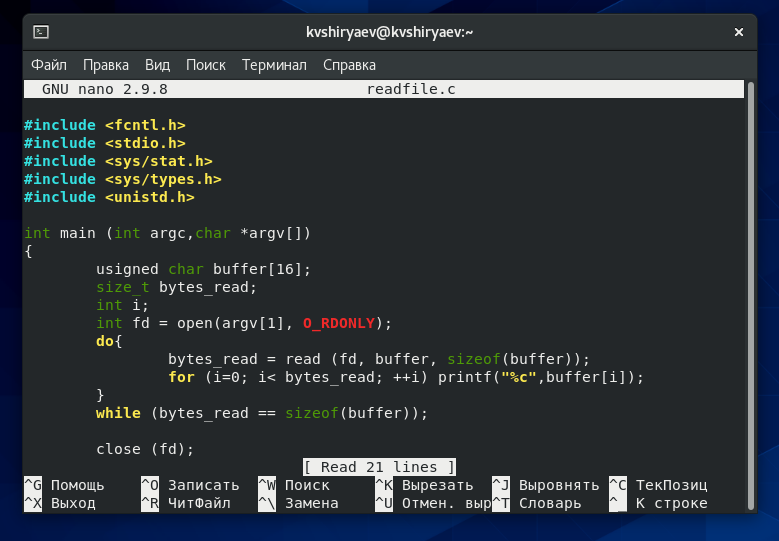


Figure 6: Программа simpleid2.c

## Компиляция и выполнение simpleid2.c

Скомпилировали и запустили программу simpleid2.c:

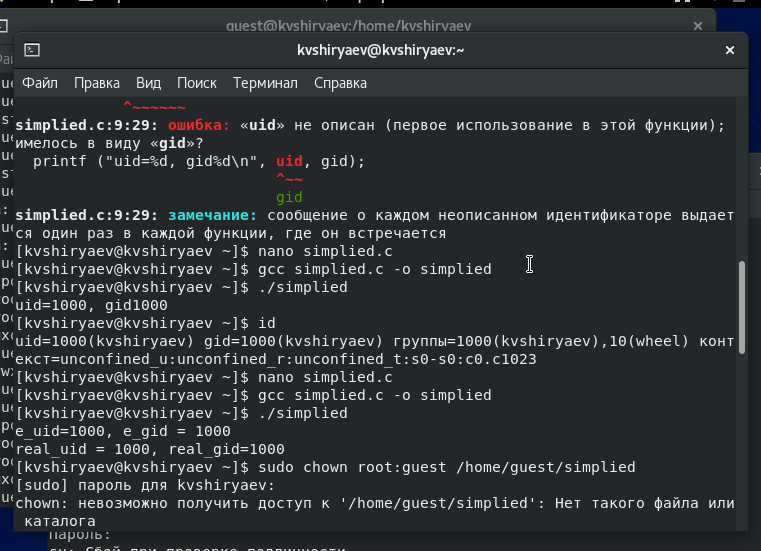


Figure 7: Компиляция и выполнение simpleid2.c

Теперь видим не только текущих группу и пользователя, но и владельца файла.

## Смена владельца и атрибут s

От имени суперпользователя выполнили команды:

chown root:guest /home/guest/simpleid2

chmod u+s /home/guest/simpleid2

Тем самым, сменили владельца файла и добавили ему дополнительный атрибут (SetUID).

Затем выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

ls -l simpleid2

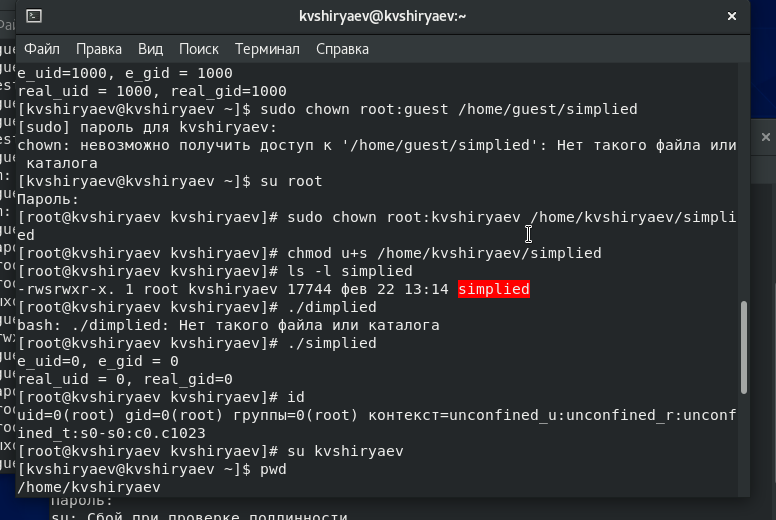


Figure 8: Смена владельца и атрибут s

## Запуск simpleid2 с SetUID

Запустили simpleid2 и id, вновь получили идентичные результаты.

Убедились в принадлежности файла пальзователю root.

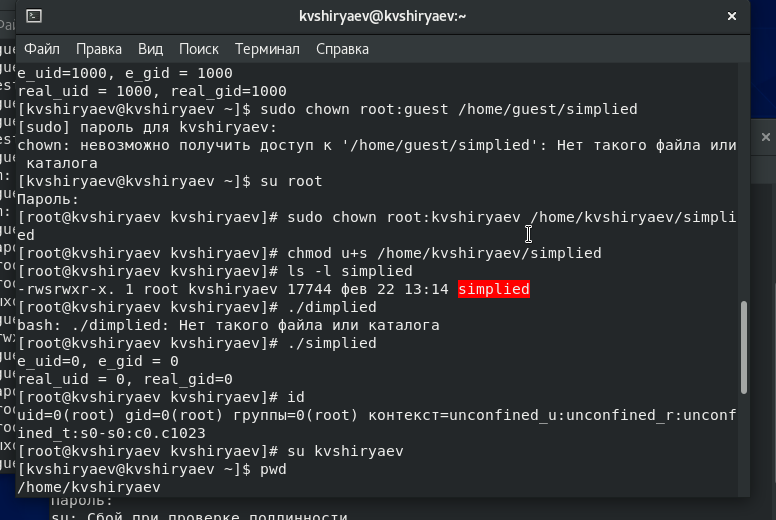


Figure 9: Запуск simpleid2 с SetUID

## Программа readfile.c

Создали программу readfile.c:

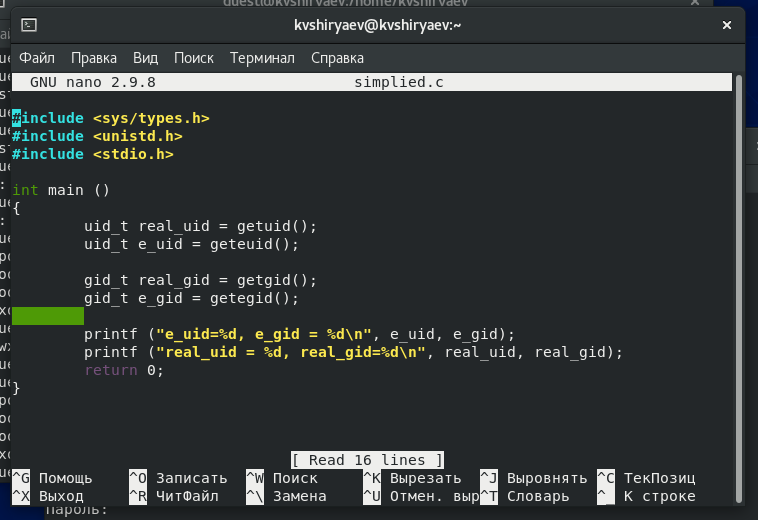


Figure 11: Программа readfile.c

## Компиляция и изменение readfile.c

Откомпилировали программу и изменили владельца у файла readfile.c и права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог, убедились в правильности, получив отказ в доступе:

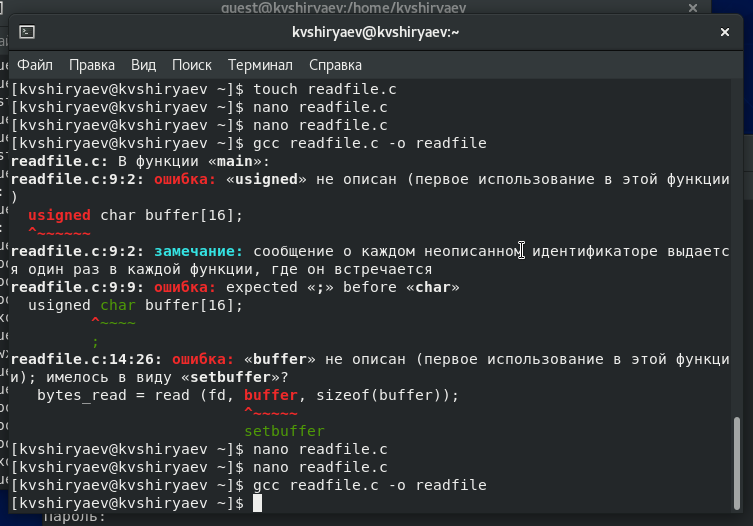


Figure 12: Компиляция и изменение readfile.c

## Атрибут Sticky на /tmp

Выяснили, что на директории /tmp установлен атрибут Sticky:

ls -l / | grep tmp

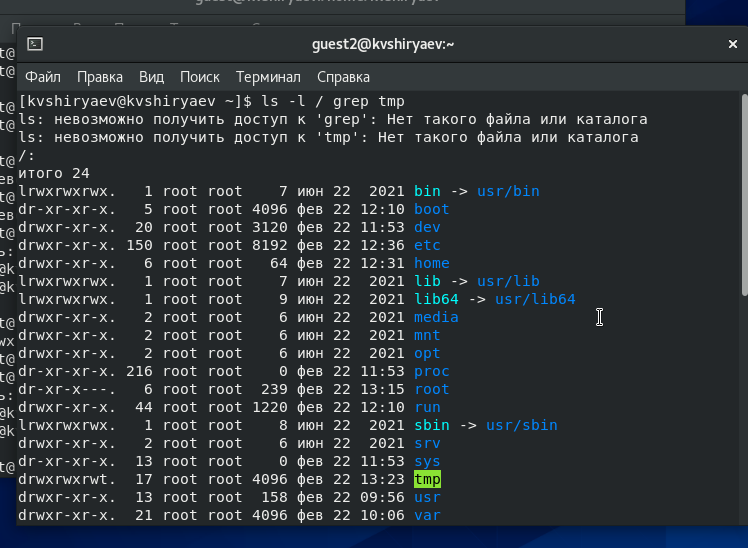


Figure 15: Атрибут Sticky на /tmp

## Создание file01.txt

От имени пользователя guest создали файл file01.txt в директории /tmp со словом test:

echo "test" > /tmp/file01.txt

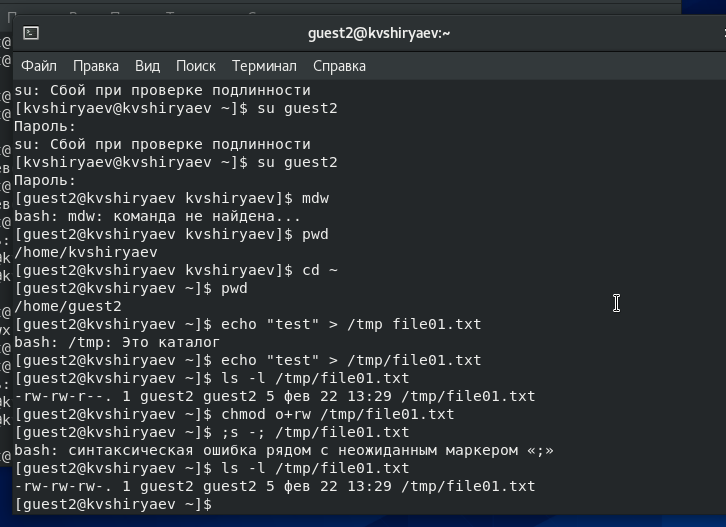


Figure 16: Создание file01.txt

## Атрибуты file01.txt

Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»:

chmod o+rw /tmp/file01.txt

ls -l /tmp/file01.txt

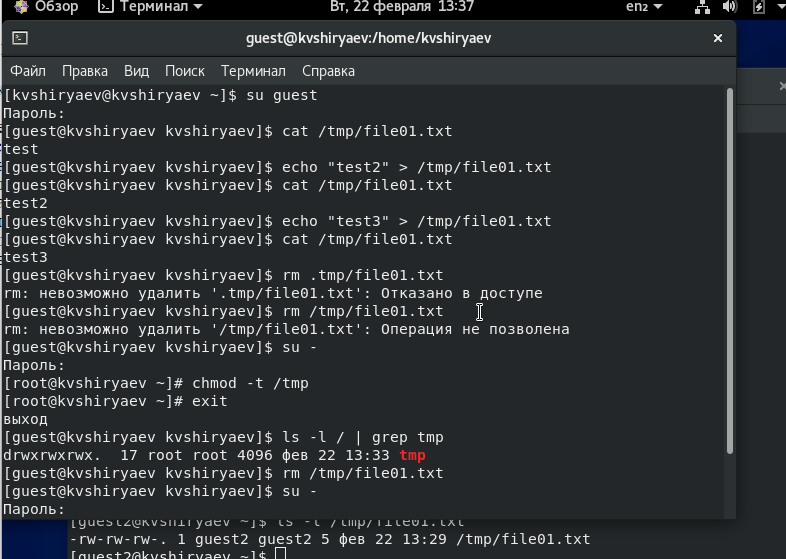


Figure 17: Атрибуты file01.txt

## Работа с файлом file01.txt в директории с t

От пользователя guest2 (не являющегося владельцем) попробовали:

1. прочитать файл /tmp/file01.txt: cat /tmp/file01.txt
2. дозаписать в файл /tmp/file01.txt слово test2: echo "test2" >> /tmp/file01.txt
3. проверить содержимое файла: cat /tmp/file01.txt
4. записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию: echo "test3" > /tmp/file01.txt
5. проверить содержимое файла: cat /tmp/file01.txt
6. удалить файл /tmp/file01.txt: rm /tmp/fileOl.txt

Удалось дозаписать информацию в файл, перезаписать, прочитать, но не удалось удалить файл:

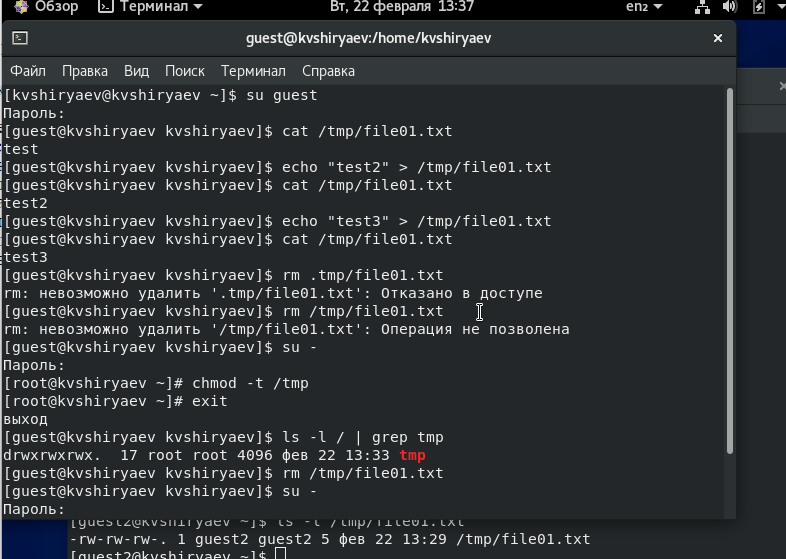


Figure 18: Работа с файлом file01.txt в директории с t

## Снятие атрибута t

Повысили свои права до суперпользователя командой su -

Выполнили после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp.

Покинули режим суперпользователя командой exit

От пользователя guest2 проверили, что атрибута t у директории /tmp нет: ls -l / | grep tmp

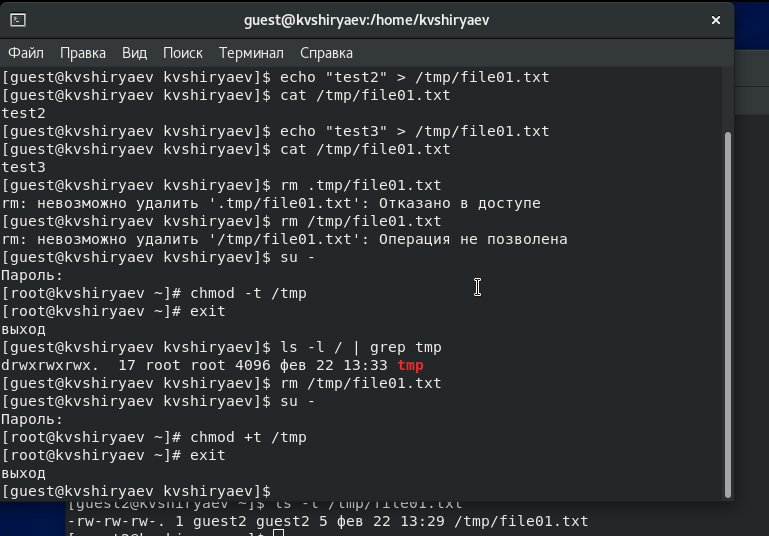


Figure 19: Снятие атрибута t

## Работа с файлом file01.txt в директории без t

Повторили все действия и, в отличие от предыдущего раза, теперь уже нам удалось удалить файл:

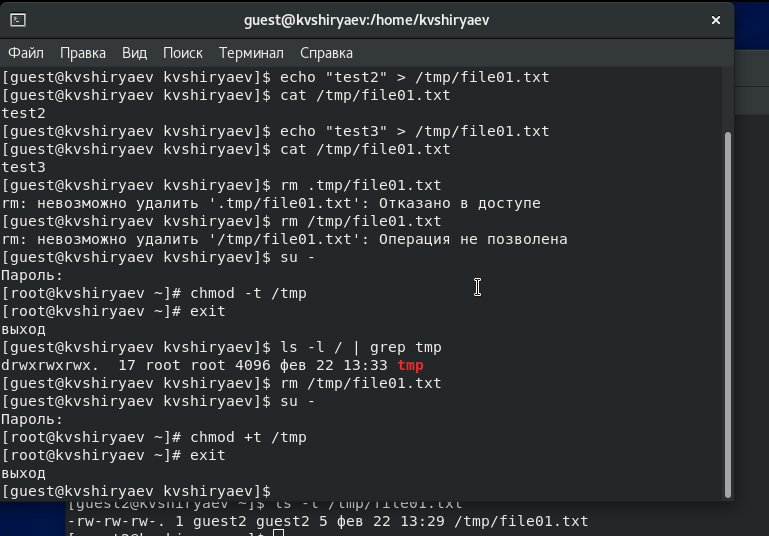


Figure 20: Работа с файлом file01.txt в директории без t

## Возврат атрибута t

Повысили свои права с помощью su, вернули директории /tmp атрибут t:

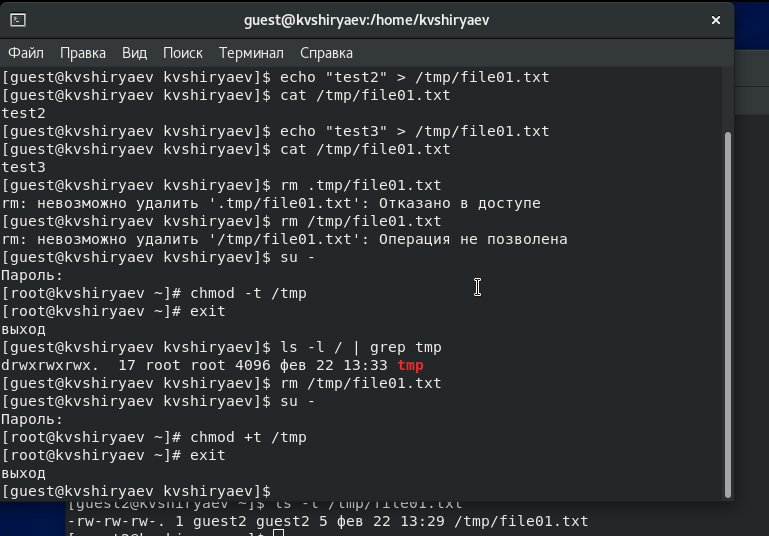


Figure 21: Возврат атрибута t

# Вывод

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов.

Получили практические навыки работы в консоли с дополнительными атрибутами.

Рассмотрели работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.