Отчёт по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Абузярова Лейла Джамилевна НБИбд-01-19

Содержание

# 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

## 2.1 Подготовка

Для выполнения части заданий требуются средства разработки приложений. Поэтому установим копилятор gcc командой yum install gcc.

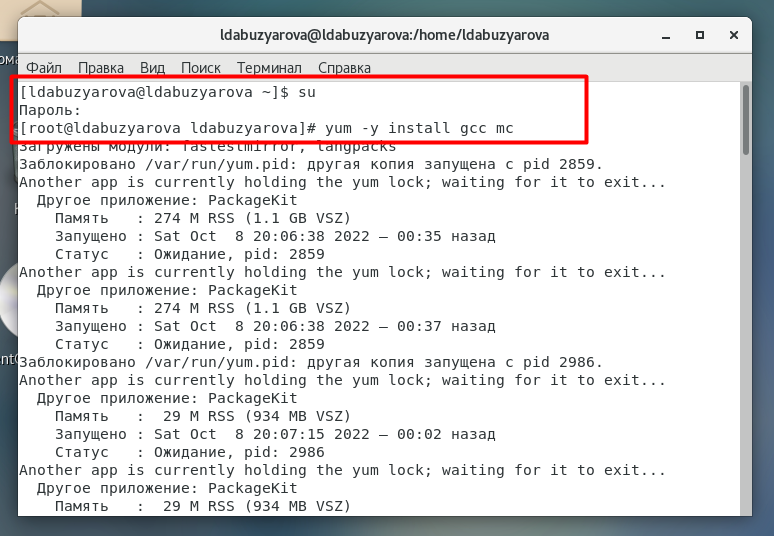


Figure 1: Установка компилятора gcc

Чтобы система защиты SELinux не мешала выполнению заданий работы, отключили систему запретов до очередной перезагрузки системы командой setenforce 0. Команда getenforce вывела Permissive:

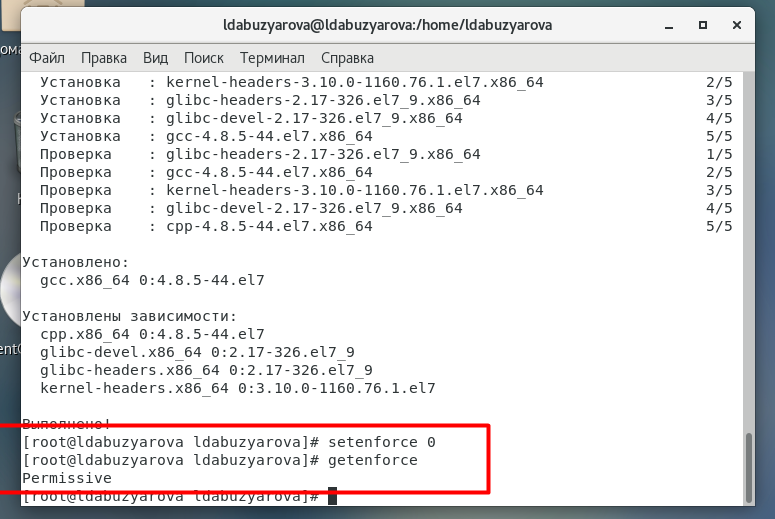


Figure 2: Подготовка к работе

Теперь проверим корректность установки компилятора, как показано на скриншоте номер 3.



Figure 3: Проверка установки компилятора

## 2.2 Изучение механики SetUID

Вошли в систему от имени пользователя guest. Сздали папку laba5, создали файл simpleid.c. Начали редактирование данного файла.

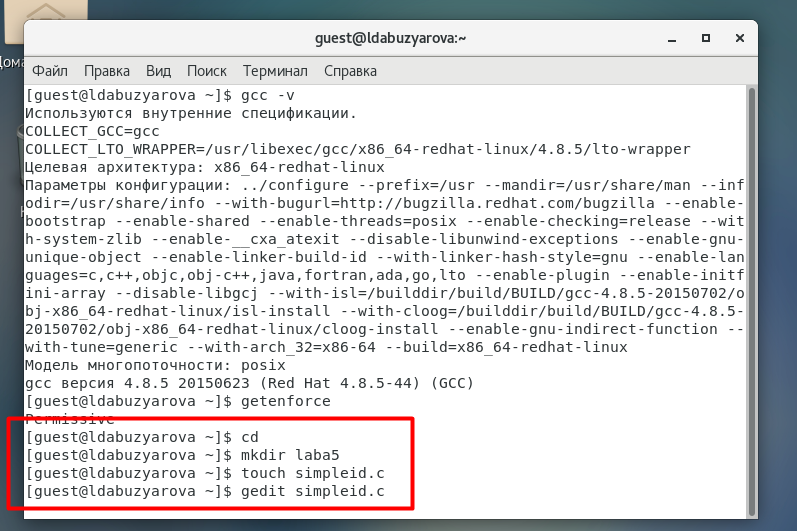


Figure 4: Создание и редактирование файла simpleid.c

Написали следующую программу в файле simpleid.c.

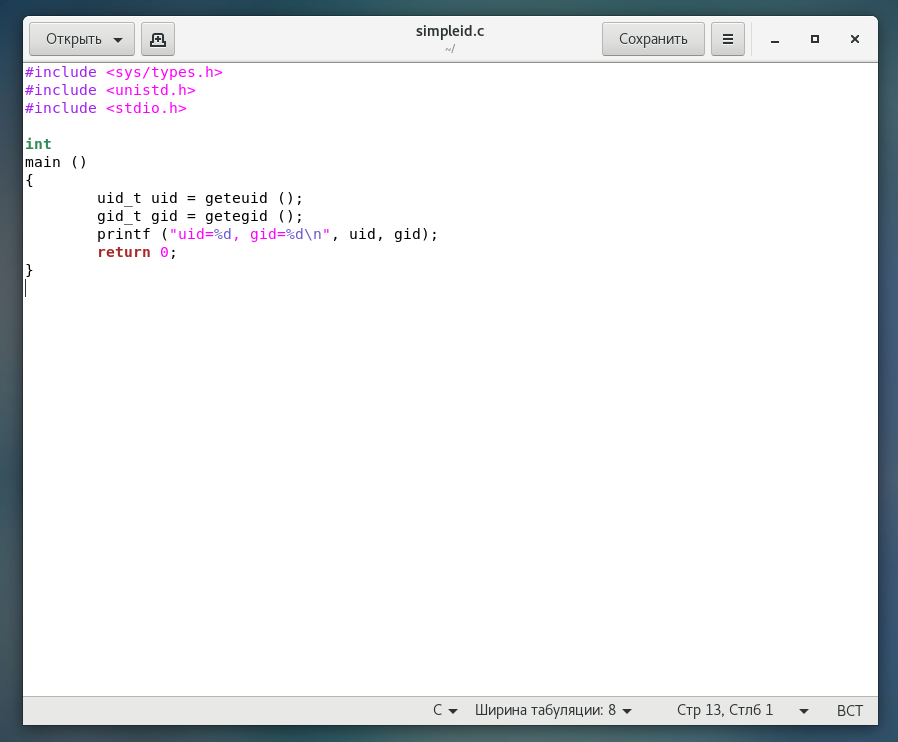


Figure 5: Написание программы

Скомпилировали программу и убедились, что файл программы создан: gcc simpleid.c -o simpleid. Выполнили программу simpleid командой ./simpleid. Выполнили системную программу id с помощью команды id. uid и gid совпадает в обеих программах

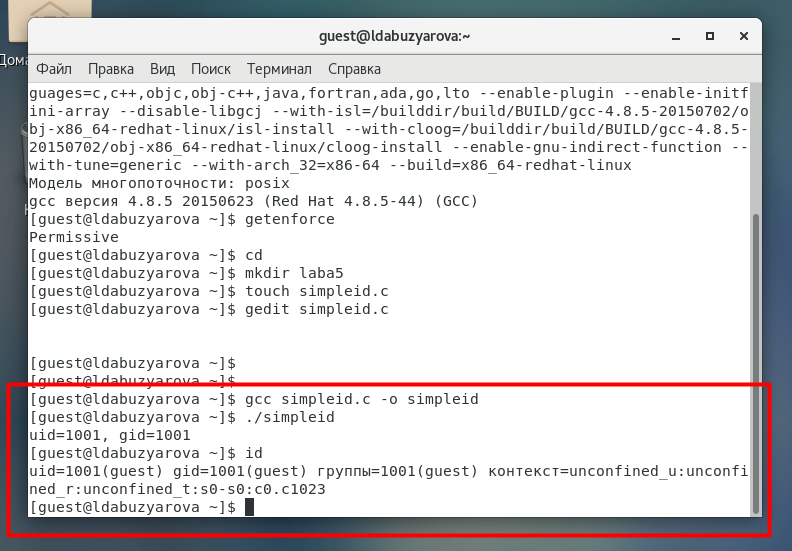


Figure 6: Результат программы simpleid

Усложнили программу, добавив вывод действительных идентификаторов. Проделываем те же операции, что и с предыдущим файлом.

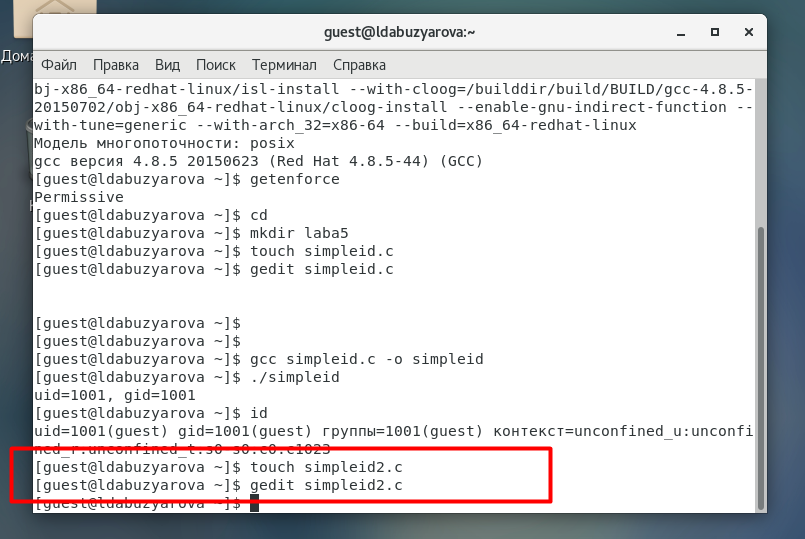


Figure 7: Создание и редактирование файла simpleid2.c

Написали следующую программу в файле simpleid2.c.

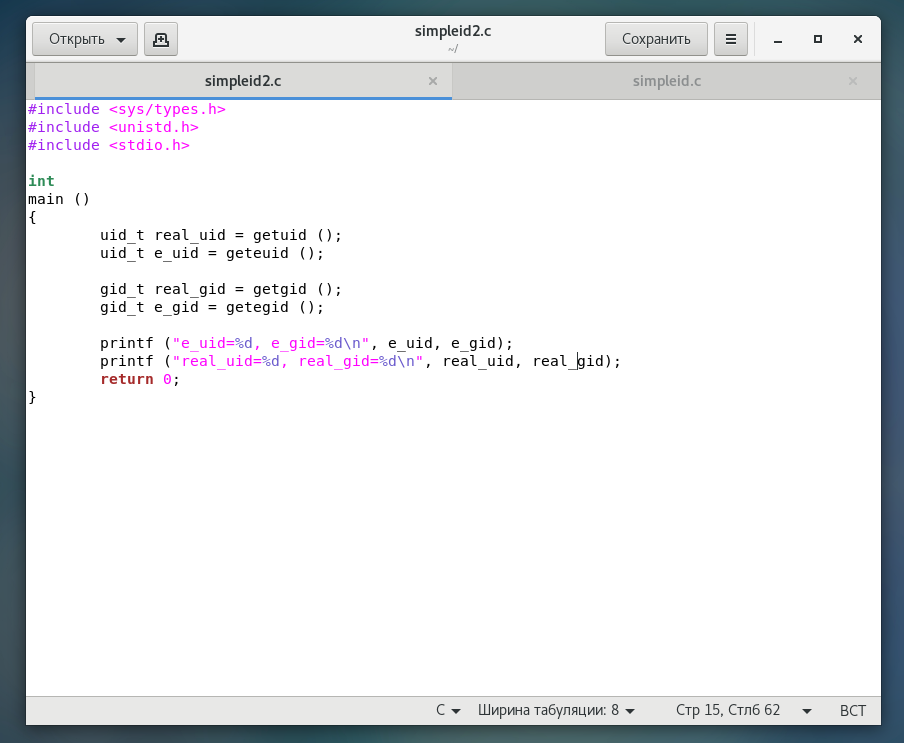


Figure 8: Написание программы

Скомпилировали и запустили simpleid2.c.

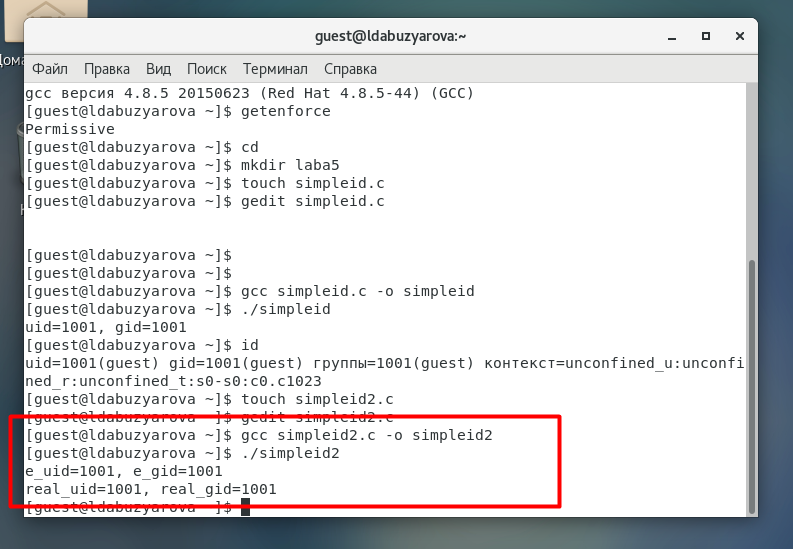


Figure 9: Результат программы simpleid2 от guest

На скриншоте 10 видно следующее: теперь от имени суперпользователя выполнили команды, которые меняют владельца файла. Использовали su для повышения прав до суперпользователя. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Также запустили simpleid2 и id.

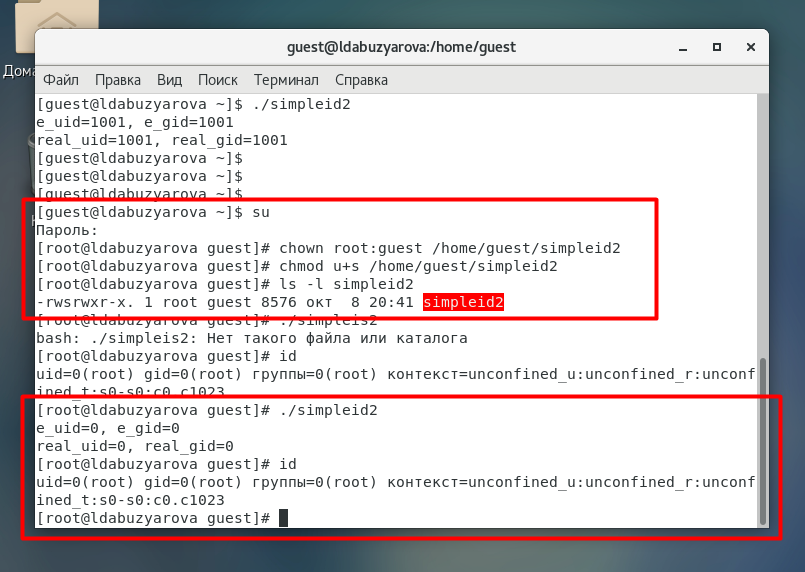


Figure 10: Результат программы simpleid2 от root

Приходим к выводу, что результат выполнения программ при заходе от пользователя guest и через root отличается.

Теперь создадим файл readfile.c, применяя те же команды, что и в предыдущий раз.

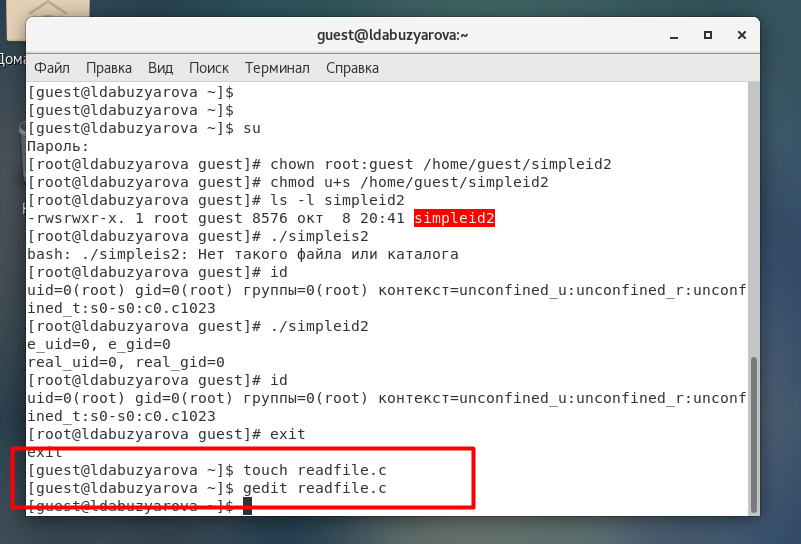


Figure 11: Создание и редактирование файла readfile.c

Написали программу readfile.c

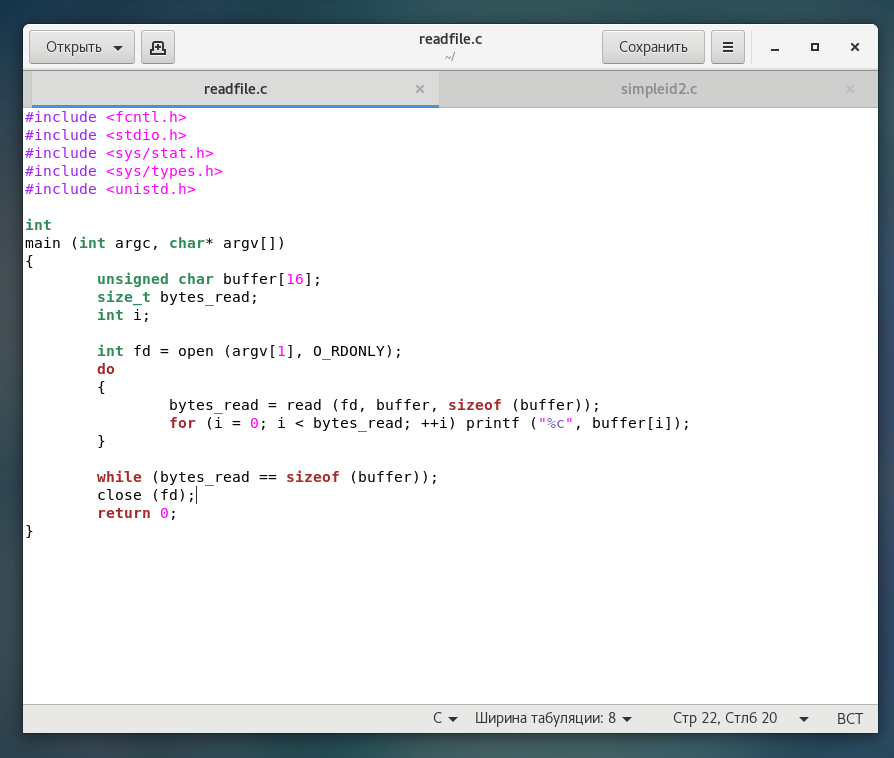


Figure 12: Программа readfile

Откомпилировали программу. Затем сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог.

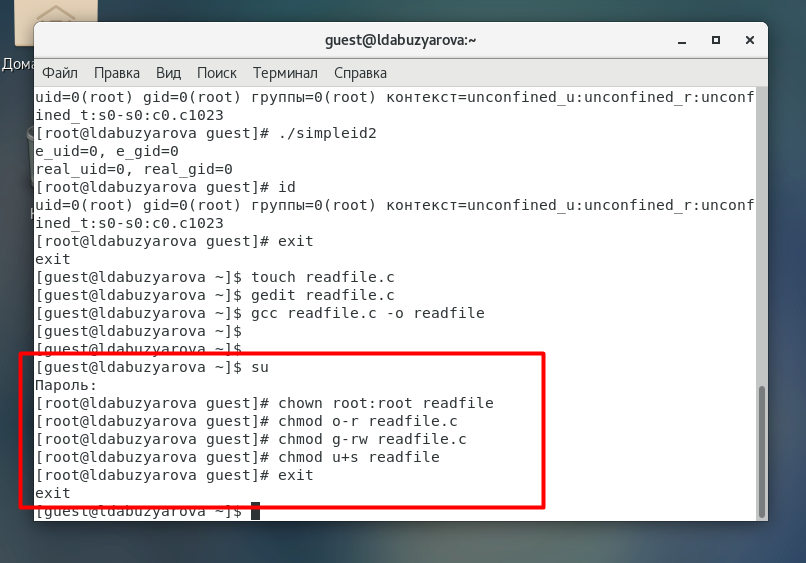


Figure 13: Смена прав на файле readfile

Проверили, что пользователь guest не может прочитать файл readfile.c. Но увидели, что файл читается.

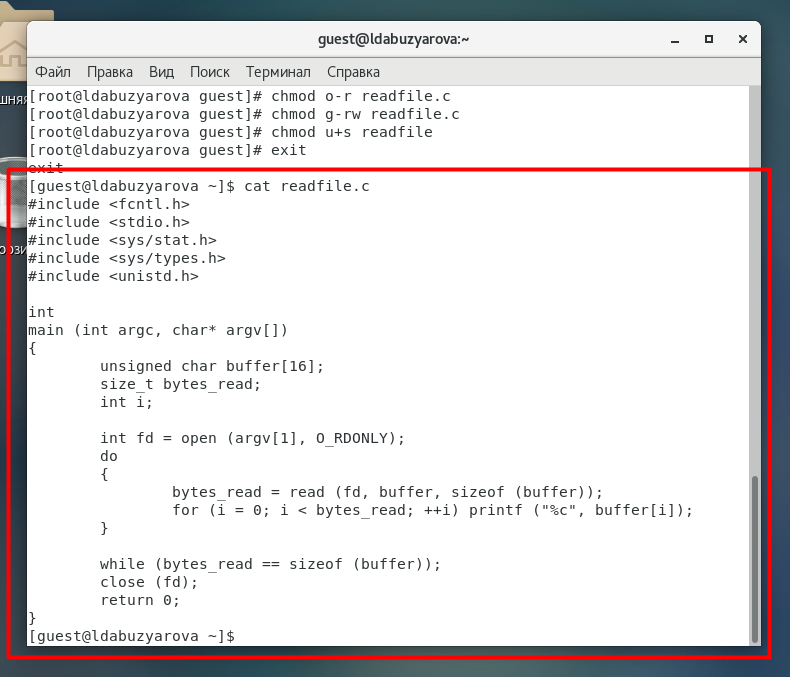


Figure 14: Проверка файла на чтение

Ищем ошибку. Оказалось, что была допущена ошибка в установке прав. Переделали команду с установкой прав и повторили проверку, на этот раз в доступе отказано, как и должно быть. А команда ./readfile сработала потому что она выполняет чтение от имени суперпользователя, а не guest.

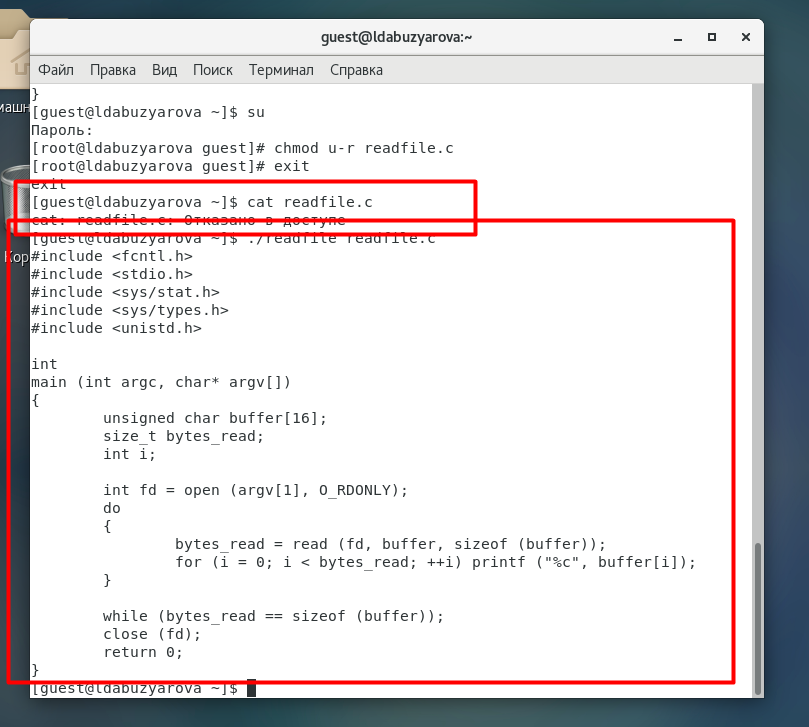


Figure 15: Повторная проверка файла на чтение

## 2.3 Исследование Sticky-бита

От имени пользователя guest создали файл file01.txt в директории /tmp со словом test. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные». Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».



Figure 16: Проверка прав на файле file01.txt

От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt. Мы видим, что в файле записано одно слово. далее к нему мы должны были поочередно дозаписать слова test2 и test3, но я забыла добавить по одной галочке и вместо дозаписи перезаписала слова.

От пользователя попробовали удалить файл /tmp/file01.txt командой rm /tmp/file01.txt, однако получила отказ.

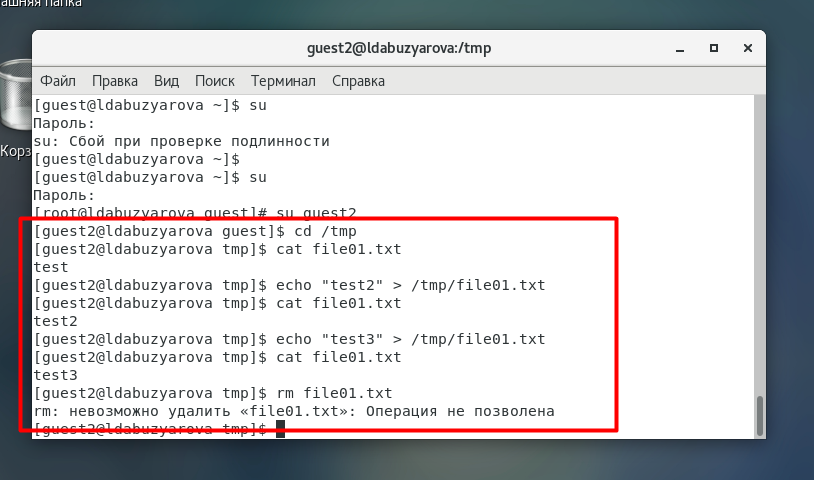


Figure 17: Дозапись и перезапись слов в файл file01.txt

От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Покинули режим суперпользователя командой exit. От пользователя проверили, что атрибута t у директории /tmp нет. Повторили предыдущие шаги. Получилось удалить файл.

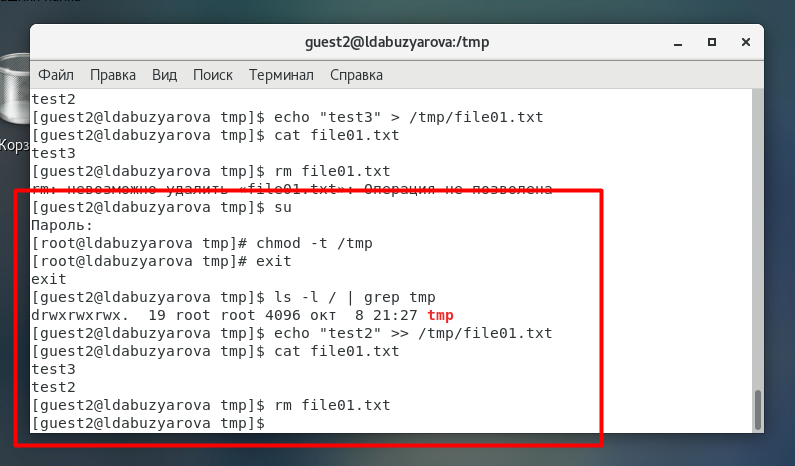


Figure 18: Снятие атрибута t

# 3 Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

# Список литературы

1. [КОМАНДА CHATTR В LINUX](https://losst.ru/neizmenyaemye-fajly-v-linux)
2. [chattr](https://en.wikipedia.org/wiki/Chattr)