Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Сухарев Кирилл

Содержание

# 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 OC Centos 7 на базе Linux.

# Выполнение работы

## Создание программы

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

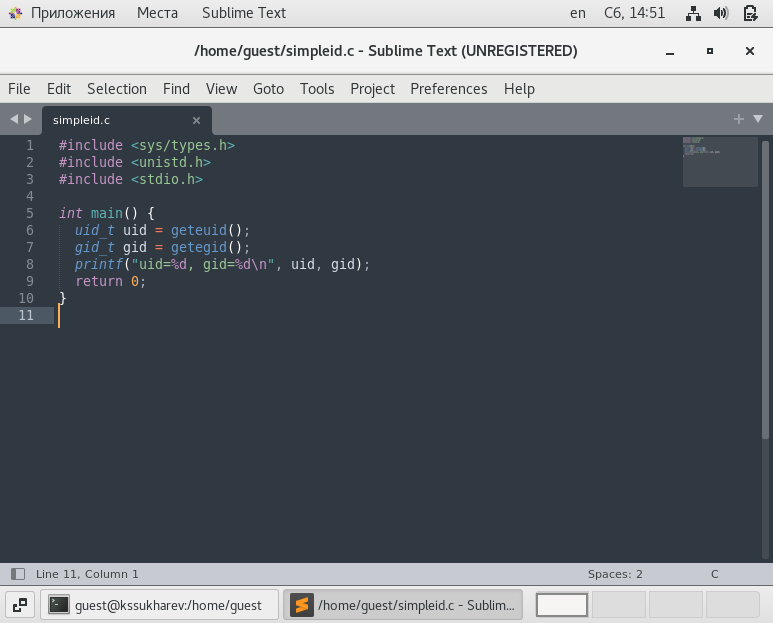


Figure 1: Создание simpleid.c

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

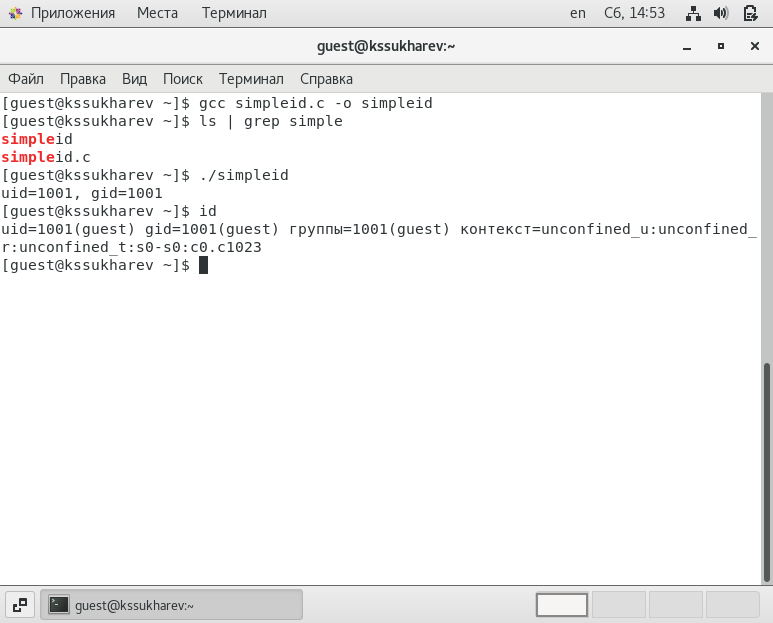


Figure 2: Проверка работоспособности

1. Создадим файл simpleid2.c, где дополнительно будем выводить действительные идентиикаторы (fig. 3).

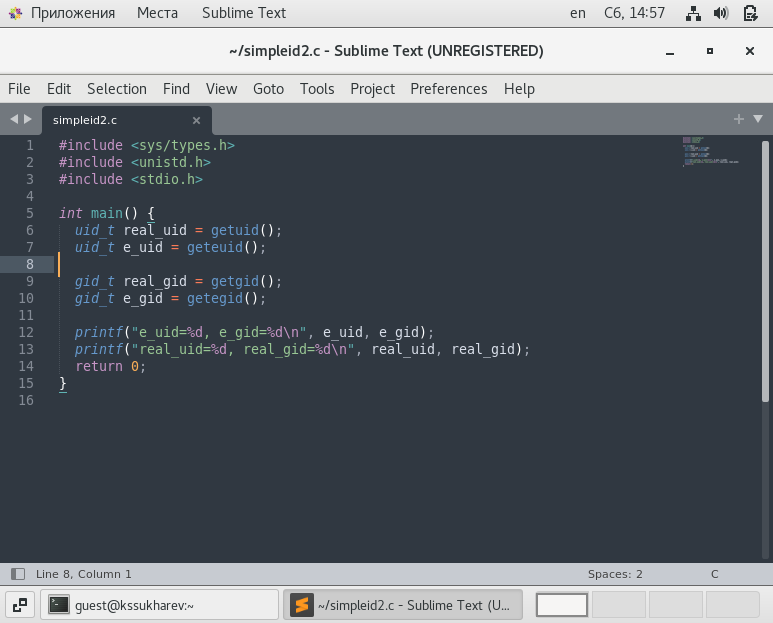


Figure 3: Создание simpleid2.c

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

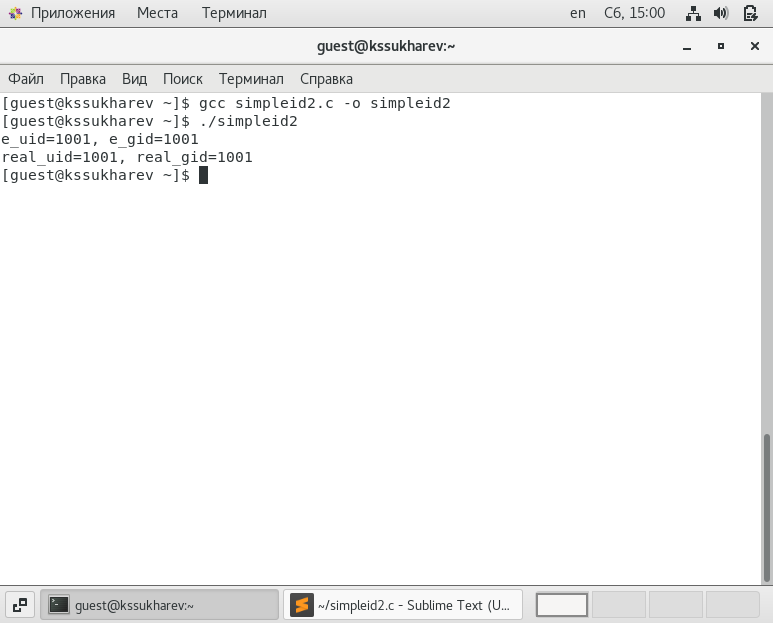


Figure 4: Создание simpleid2.c

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

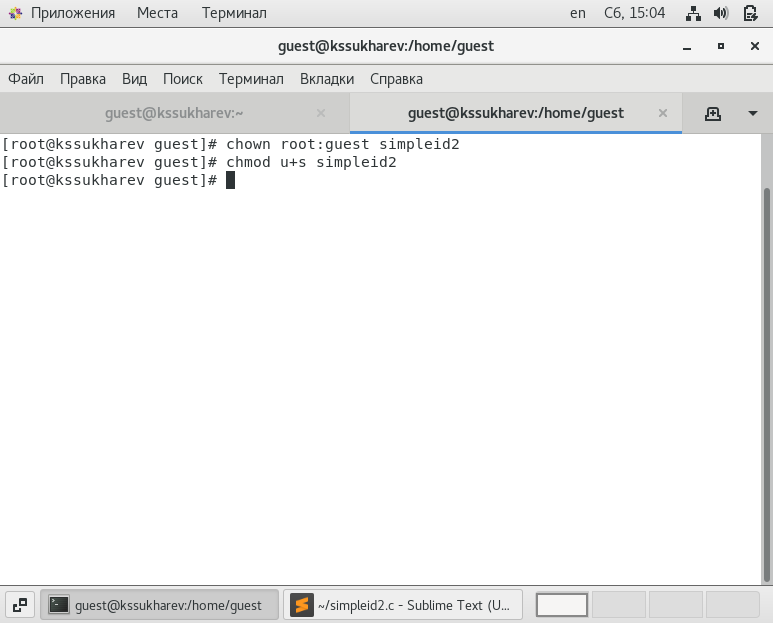


Figure 5: chown и chmod

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

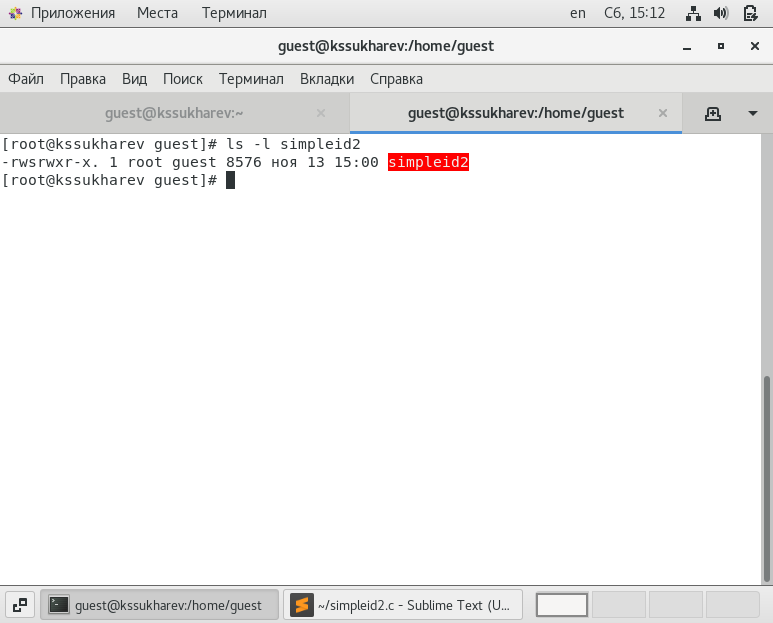


Figure 6: Проверка правильности

1. Запустим simpleid2 и id. Видим, что real\_uid и real\_gid соответствуют данным id, а SetUID-бит установлен в 0 (суперпользователь) (fig. 7).

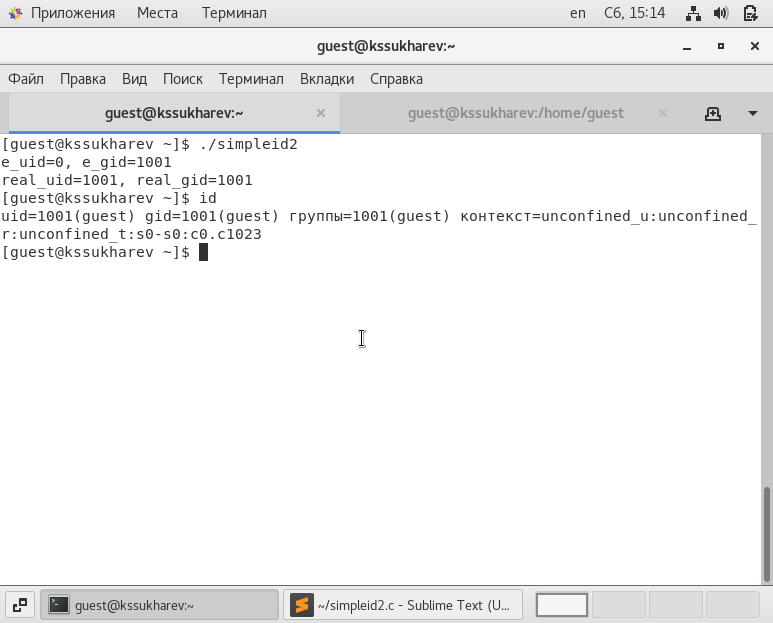


Figure 7: Выполнение simpleid2

1. 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. 8).

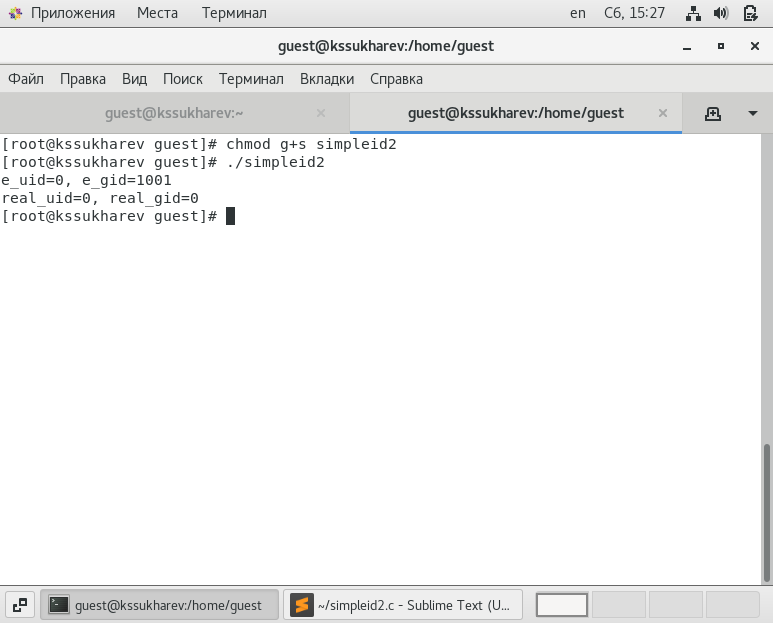


Figure 8: Установка SetGID-бита

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

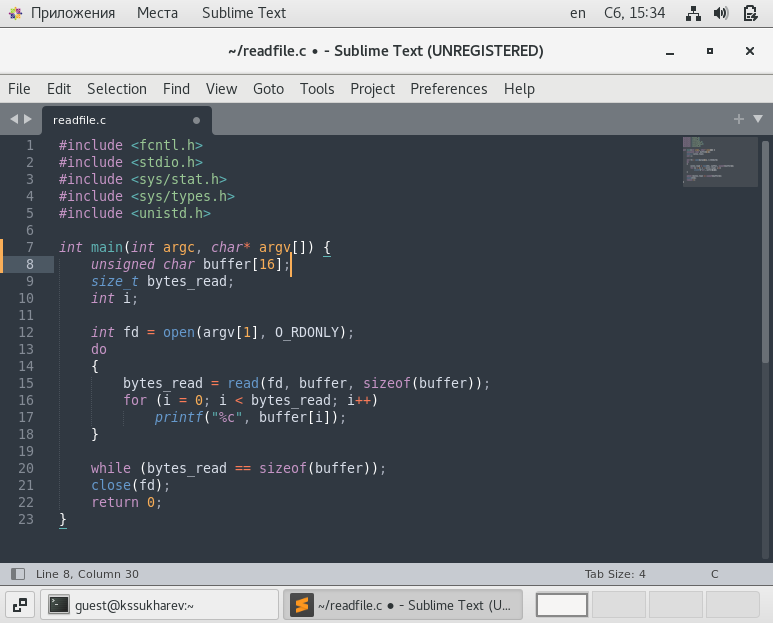


Figure 9: Создание readfile.c

1. Откомпилируем ее (fig. 10).

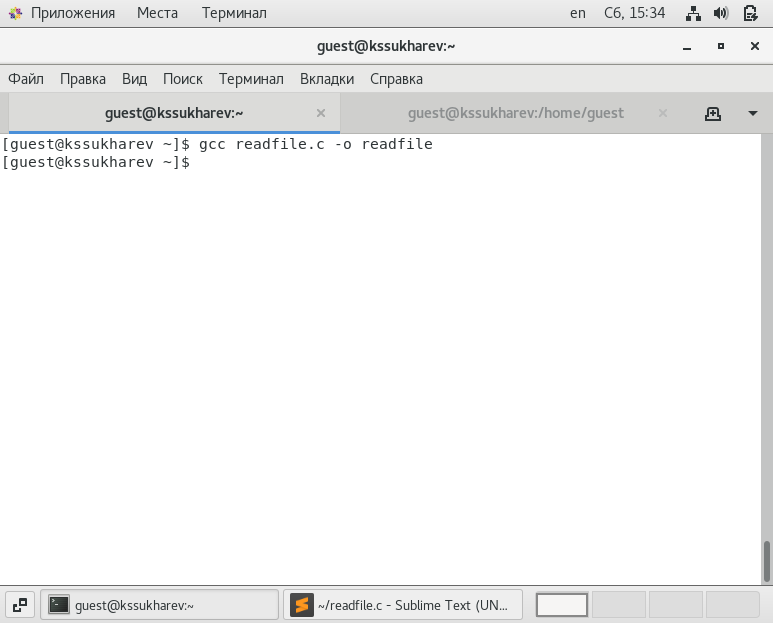


Figure 10: Компиляция файла readfile.c

1. Изменим права у файла readfile.c так, чтобы его мог прочитать только суперпользователь (fig. 11).

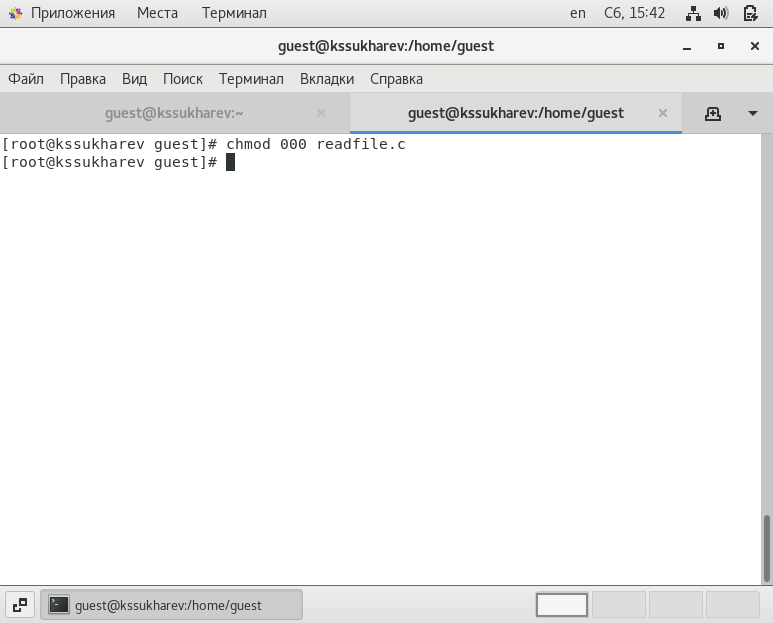


Figure 11: Смена прав у файла

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

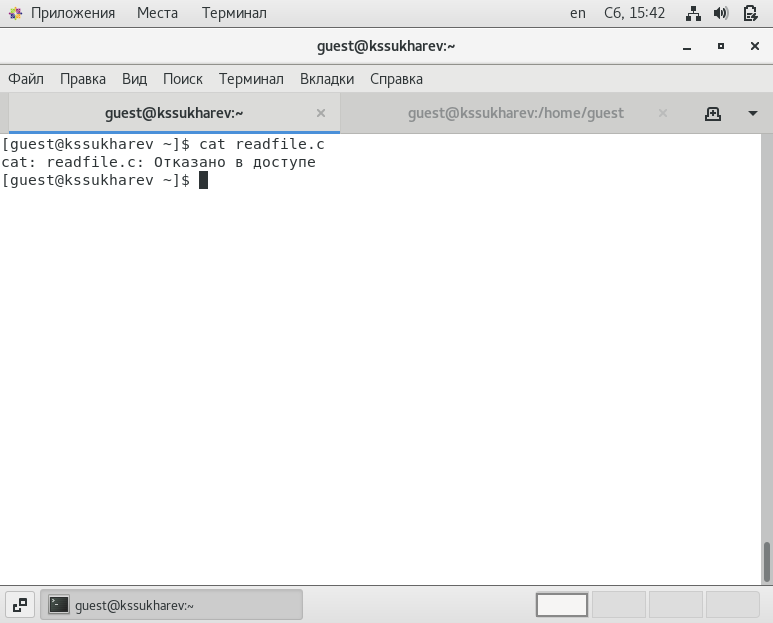


Figure 12: Проверка недоступности readfile.c для guest

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

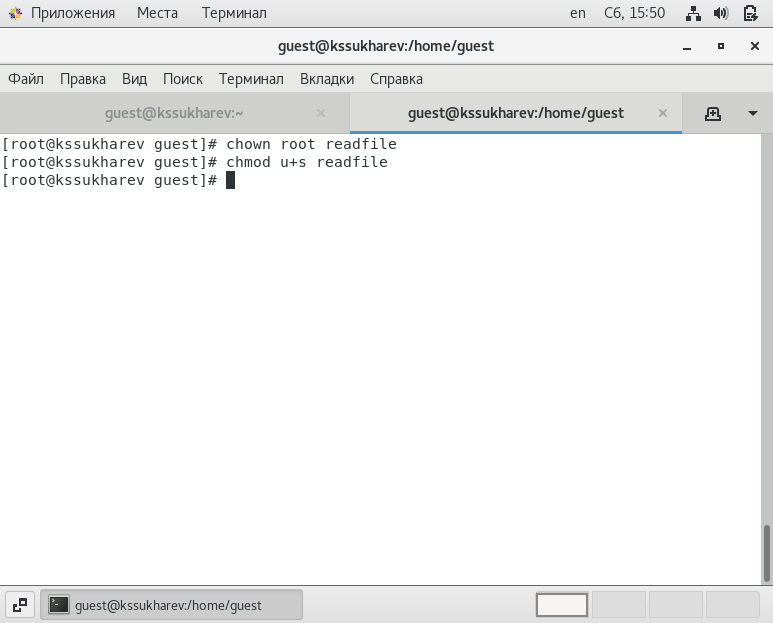


Figure 13: Смена владельца и установка SetUID-бита

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

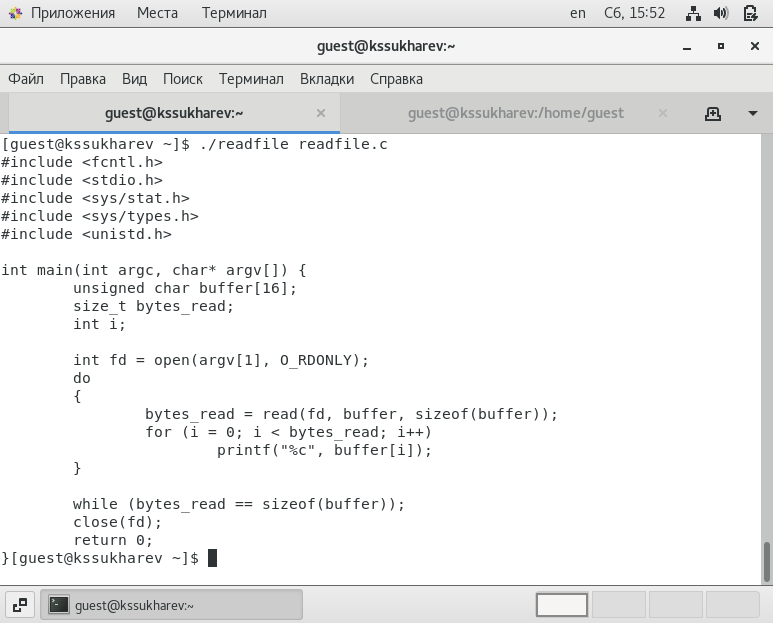


Figure 14: Попытка прочитать readfile.c

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

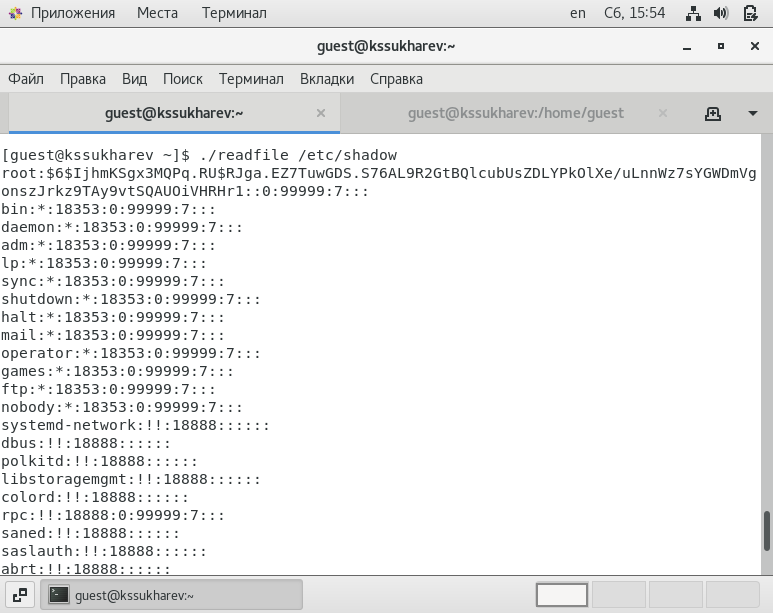


Figure 15: Попытка прочитать etc/shadow

## Исследование Sticky-бита

1. Выясним, установлне ли атрибут Sticky на директории /tmp. По результатам выполнения команды ls -l видим, что Sticky-бит установлен (fig. 16).

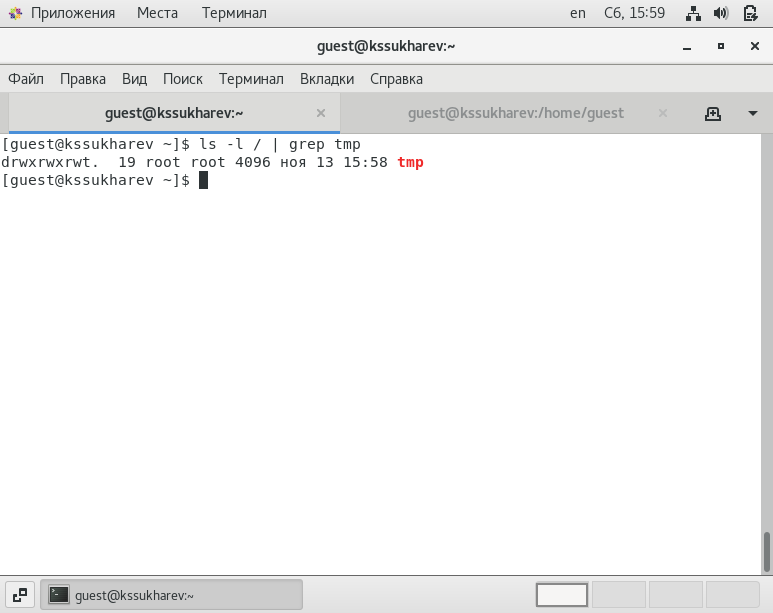


Figure 16: Проверка налиичия атрибута Sticky

1. Создадим в директории /tmp файл file01.txt. Посмотрим атрибуты этого файла, а затем разрешим остальным пользователям чтение и запись (fig. 17).

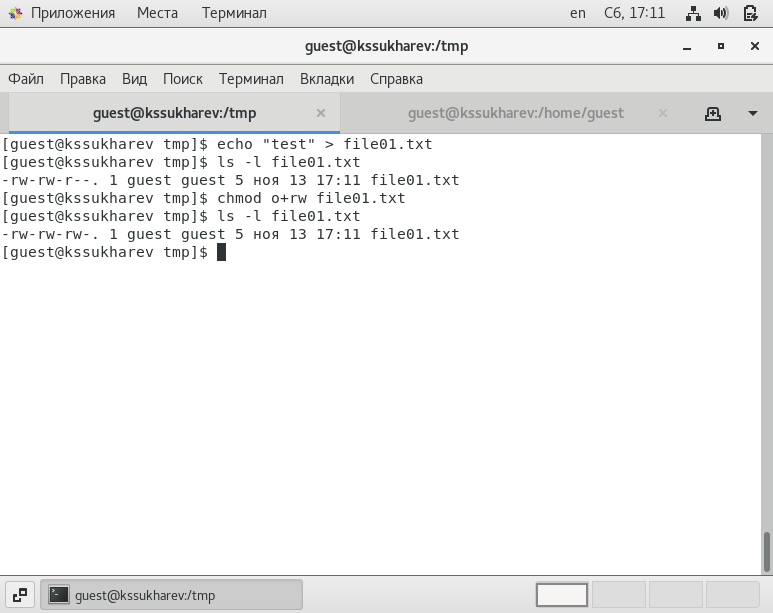


Figure 17: Создание file01.txt

1. От имени пользователя guest2 попробуем прочитать созданный файл. Никаких ошибок не возникло (fig. 18).

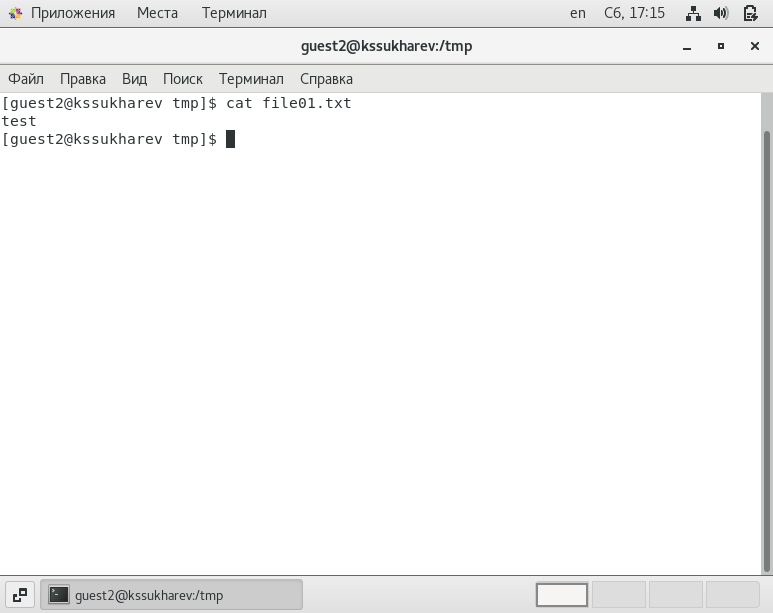


Figure 18: Попытка чтения file01.txt

1. Теперь попробуем дозаписать в этот файл слово test2. Как можно видеть, дозапись прошла успешно (fig. 19).

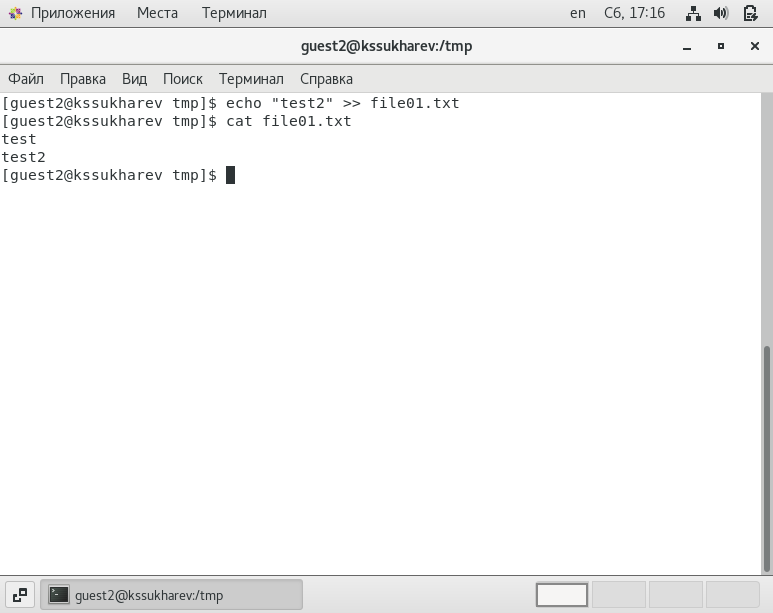


Figure 19: Дозапись file01.txt

1. Попробуем заменить содержимое файла на “test3”. Видим, что перезапись файла также прошла успешно (fig. 20).

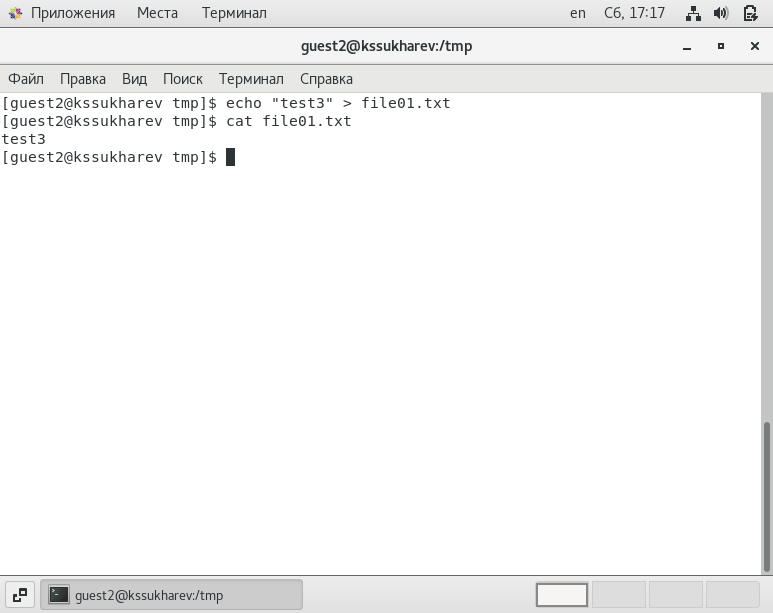


Figure 20: Перезапись file01.txt

1. Попробуем удалить файл. Данная операция не позволена (fig. 21).

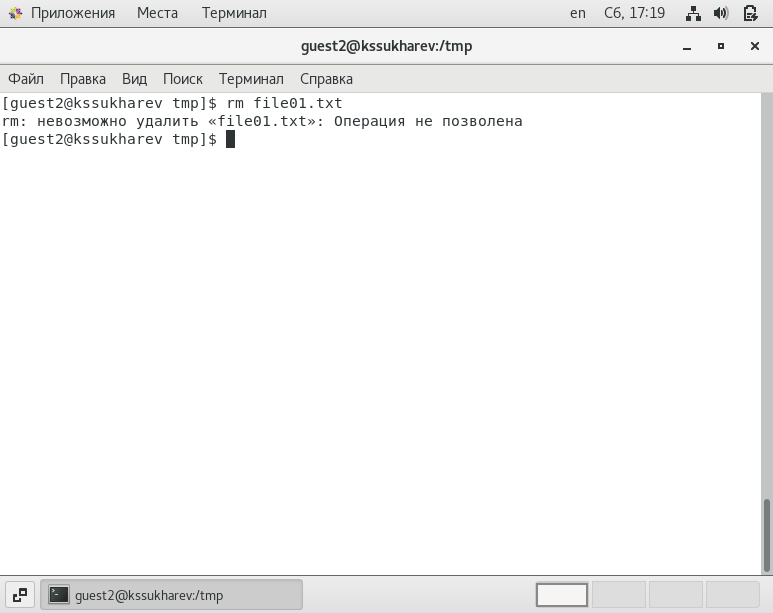


Figure 21: Попытка удаления file01.txt

1. Повысим свои права до суперпользователя и снимем Sticky-бит с директории /tmp (fig. 22).

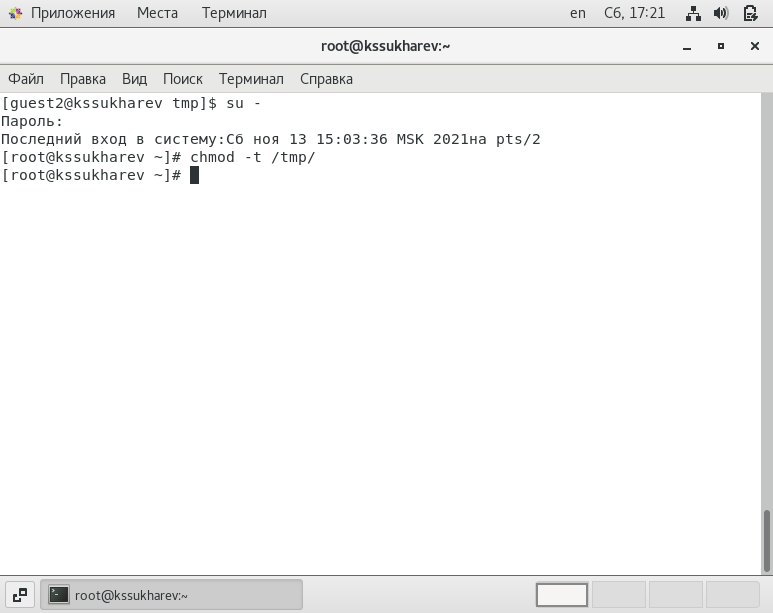


Figure 22: Снятие Sticky-бита

1. Выйдем из режима суперпользователя и убедимся, что атрибута t у директории /tmp больше нет (fig. 23).

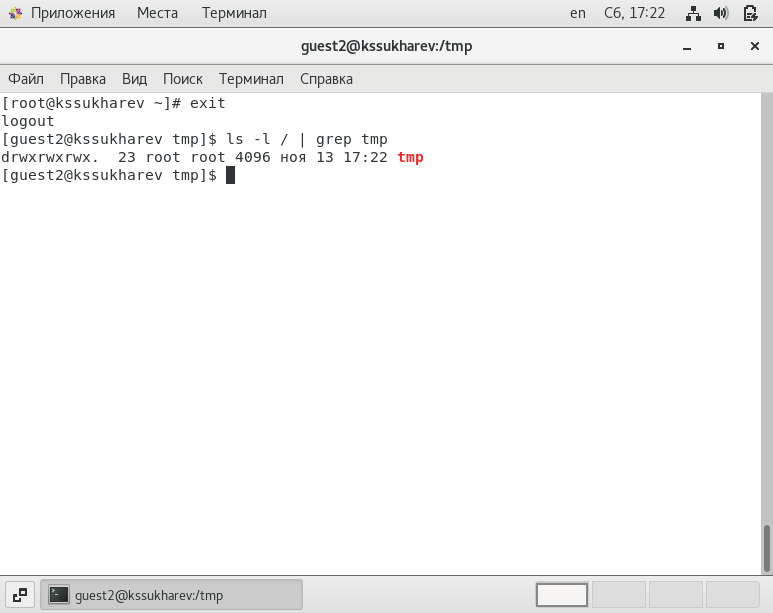


Figure 23: Проверка снятия Sticky-бита

1. Повторим предыдущие шаги. Видим, что все ограничения были сняты. Нам даже удалось удалить этот файл (fig. 24).

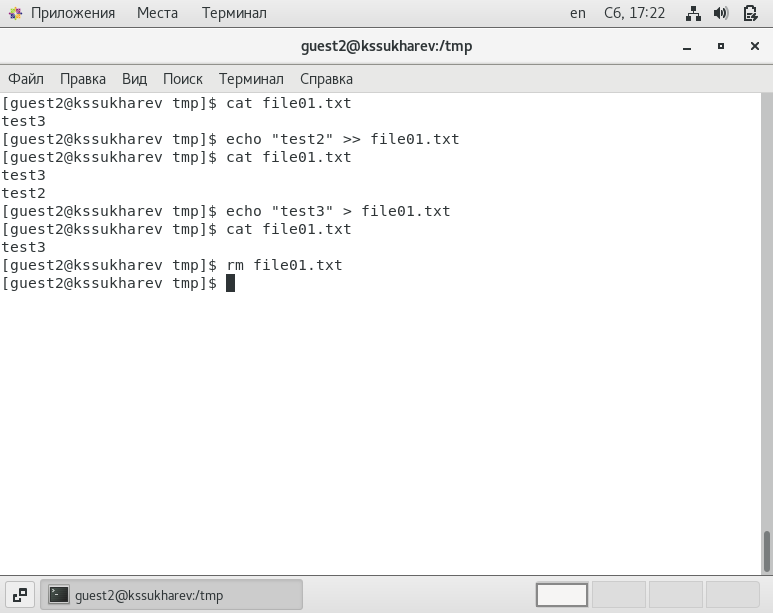


Figure 24: Проверка предыдущих команд

1. Вернем Sticky-бит на директорию /tmp (fig. 25).

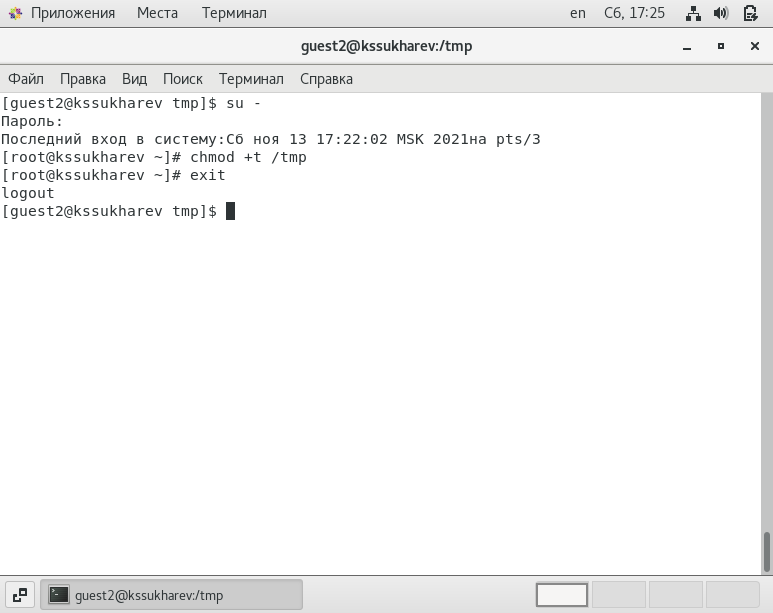


Figure 25: Проверка предыдущих команд

# 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 // ВикиЧтение. 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. M.: РУДН, 2015. 64 с..