**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ по лабораторной работе 5

ТЕМА «Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов»

### по дисциплине «Информационная безопасность»

**Выполнил/ла:**

**Студент/ка группы:** НПИбд-02-21

**Студенческий билет №** 1032205421

**Студент/ка:** Стелина Петрити

# Список содержания

Список содержания.

Список изображений Цель работы.

*Последовательность выполнения работы*

Выводы

# Список изображений

рис. 1 cоздание simpleid.c рис. 2 компиляция и id

рис. 3 Создание simpleid2.c

рис. 4 Компиляция и запуск simpleid2.c рис. 5 команд chown,chmod

рис. 6 правильности установки атрибутов рис. 7 запуск, id

рис. 8 SetGID-бита

рис. 9 Создание readfile.c рис. 10 Пункты 14,15,16 рис. 11 Пункты 17,18,19 рис. 12 Пункты 1,2,3

рис. 13 Пункты 4,5,6,7,8 рис. 14 Пункты 10,11 рис. 15 Нет атрибута

рис. 16 Возвращающий атрибут t рис. 17 Проверка

## Цель работы

Получение практических навыков работы в консоли с дополнительными атрибутами. Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

## Последовательность выполнения работы

### Создание программы

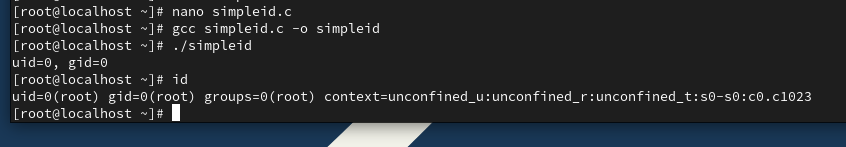
* + - 1. Войдите в систему от имени пользователя guest.
      2. Создайте программу simpleid.c:



*рис. 1 Создание simpleid.c*

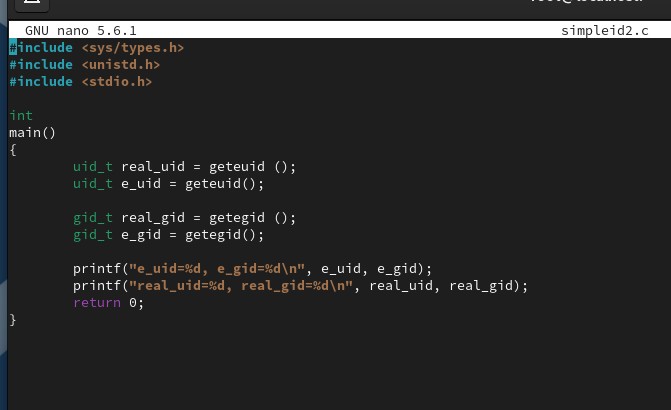
* + - 1. Скомплилируйте программу и убедитесь, что файл программы создан:gcc simpleid.c -o simpleid
      2. Выполните программу simpleid:./simpleid
      3. Выполните системную программу id:id и сравните полученный вами результат с данными предыдущего пункта задания.

Один и тот же uid и gid



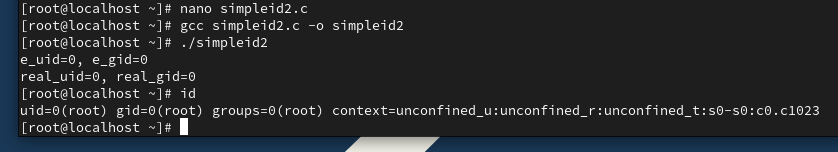
*рис. 2 компиляция и id*

* + - 1. Усложните программу, добавив вывод действительных идентификаторов: Получившуюся программу назовите simpleid2.c.



*рис. 3 Создание simpleid2.c*

* + - 1. Скомпилируйте и запустите simpleid2.c: gcc simpleid2.c -o simpleid2, ./simpleid2

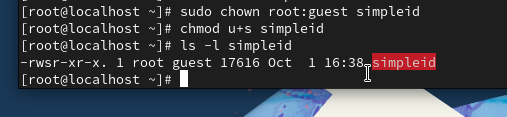


*рис. 4 Компиляция и запуск simpleid2.c*

* + - 1. От имени суперпользователя выполните команды: chown root:guest

/home/guest/simpleid2, chmod u+s /home/guest/simpleid2

* + - 1. Используйте sudo или повысьте временно свои права с помощью su.



*рис. 5 команд chown,chmod*

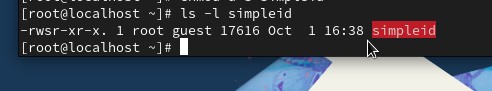
Поясните, что делают эти команды.

### sudo (SuperUser DO)

**Что это?** sudo — это команда, которая позволяет пользователям выполнять команды с привилегиями суперпользователя (root) или другого пользователя, указанного в конфигурационном файле /etc/sudoers.

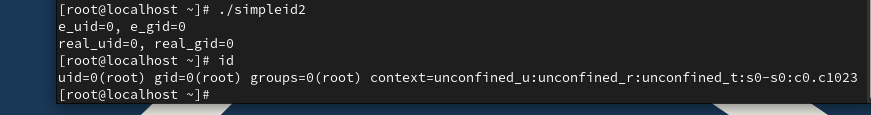
### su (Substitute User)

**Что это?** su — это команда, которая позволяет пользователю переключиться на другого пользователя, обычно на суперпользователя (root).

Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2

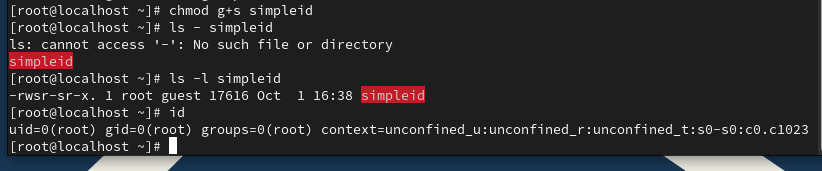
*рис. 6 правильности установки атрибутов*

1. Запустите simpleid2 и id: ./simpleid2, id Сравните результаты.



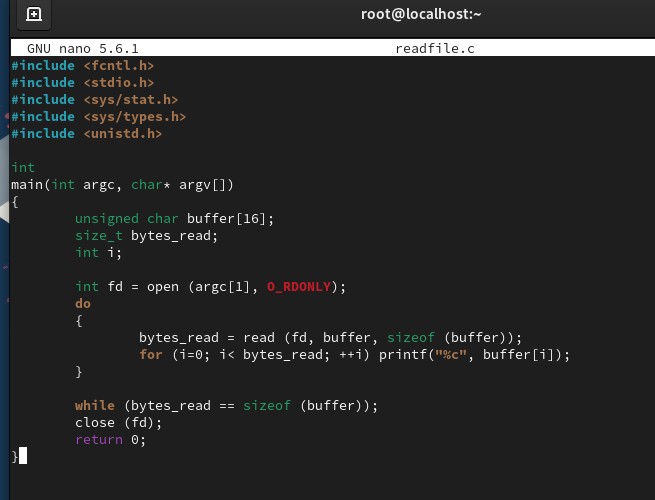
*рис. 7 запуск, id*

1. Проделайте тоже самое относительно SetGID-бита.



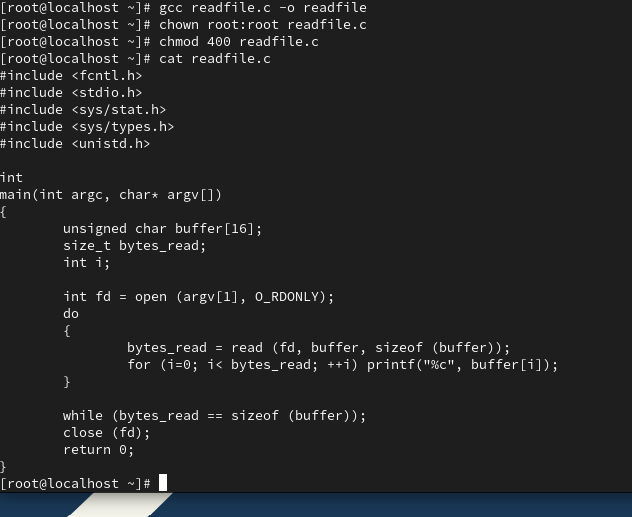
*рис. 8 SetGID-бита*

1. Создайте программу readfile.c:



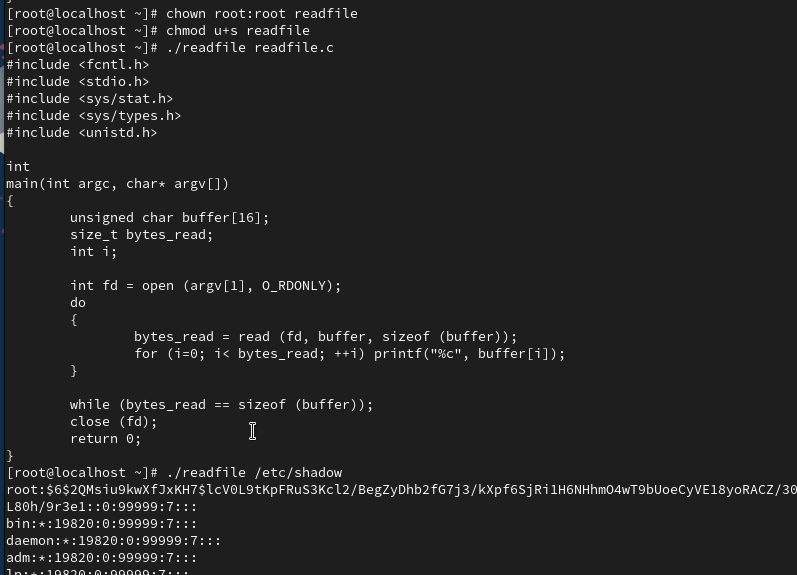
*рис. 9 Создание readfile.c*

1. Откомпилируйте её. gcc readfile.c -o readfile
2. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог.
3. Проверьте, что пользователь guest не может прочитать файл readfile.c.



*рис. 10 Пункты 14,15,16*

1. Смените у программы readfile владельца и установите SetU’D-бит.
2. Проверьте, может ли программа readfile прочитать файл readfile.c?
3. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Да, прочитается.



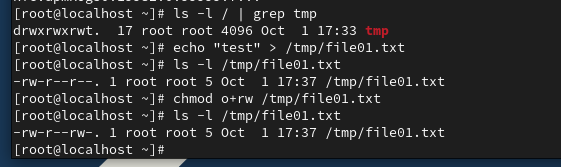
*рис. 11 Пункты 17,18,19*

* + 1. **Исследование Sticky\***\*-бита\*\*
       1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp
       2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом

test: echo "test" > /tmp/file01.txt

* + - 1. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt, chmod o+rw

/tmp/file01.txt, ls -l /tmp/file01.txt



*рис. 12 Пункты 1,2,3*

* + - 1. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл

/tmp/file01.txt: cat /tmp/file01.txt

* + - 1. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2

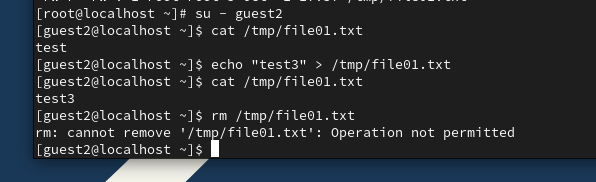
командой echo "test2" > /tmp/file01.txt Удалось ли вам выполнить операцию?

* + - 1. Проверьте содержимое файла командой cat /tmp/file01.txt
      2. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo "test3" > /tmp/file01.txt

Удалось ли вам выполнить операцию?

* + - 1. Проверьте содержимое файла командой cat /tmp/file01.txt
      2. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm

/tmp/fileOl.txt Удалось ли вам удалить файл? Нет, операция была запрещена.

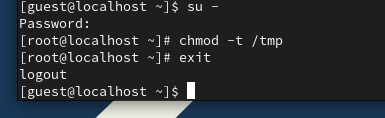


*рис. 13 Пункты 4,5,6,7,8*

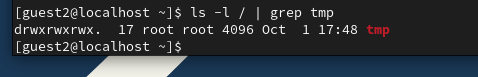
* + - 1. Повысьте свои права до суперпользователя следующей командой su - и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t

/tmp

* + - 1. Покиньте режим суперпользователя командой Exit



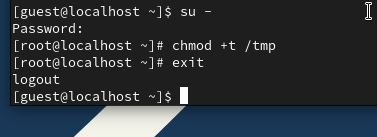
*рис. 14 Пункты 10,11*

* + - 1. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep tmp

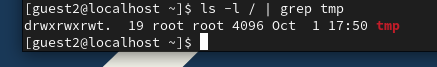
*рис. 15 Нет атрибута*

* + - 1. Повторите предыдущие шаги. Какие наблюдаются изменения?
      2. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.
      3. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp: su

- chmod +t /tmp, exit



*рис. 16 Возвращающий атрибут t*



*рис. 17 Проверка*

## Вывод

В этой лабораторной работе мы изучили, как работают специальные атрибуты файлов в Linux. Мы увидели, как они помогают контролировать доступ к файлам и программам, а также защищают общие папки от удаления файлов другими пользователями.