# Лабораторная работа №5

# Дисциплина: информационная безопасность

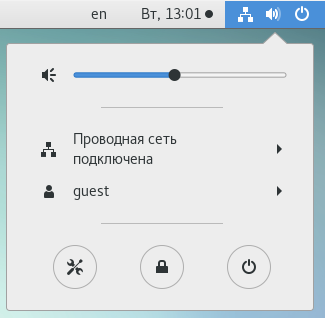
# Студент: Подорога Виктор Александрович

# Цель работы

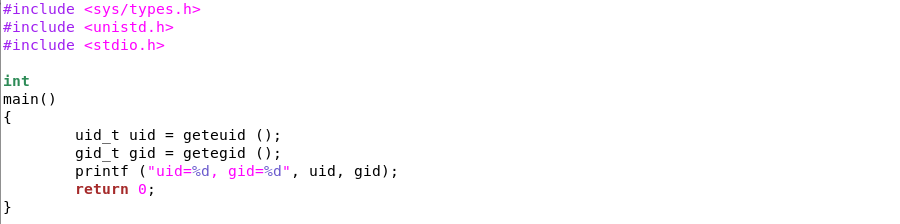
Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# Выполнение лабораторной работы

1. Вошел в систему от имени пользователя guest:

* 
* *Рис. 1. Вход в систему*

1. Создал файл с кодом программы на языке С:

* 
* *Рис. 2.1. Создание файла*
* 
* *Рис. 2.2. Код программы*

1. Скомпилировал программу:

* 
* \*Рис. 3. Компиляция

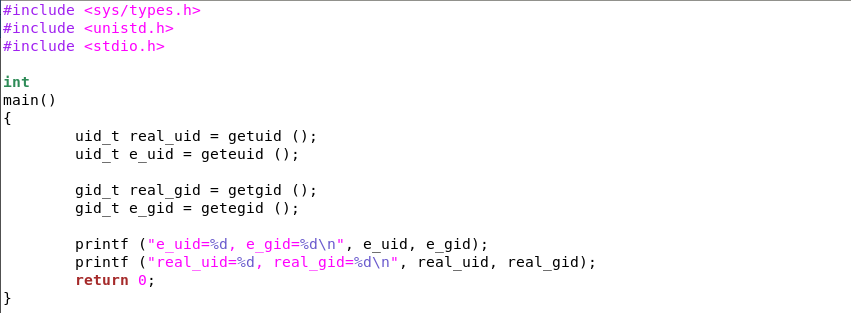
1. Выполнил программу командой ./simpleid:

* 
* *Рис. 4. Результат работы*

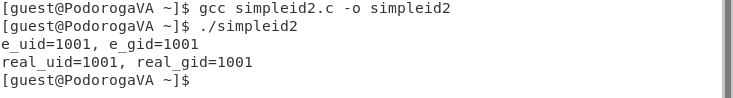
1. Выполнил системную программу id:

* 
* *Рис. 5. Выполнение id*

1. Усложнил программу, добавив вывод действительных идентификаторов:

* 
* *Рис. 6.1. Создание файла*
* 
* *Рис. 6.2. Код программы*

1. Скомпилировал и запустил:

* 
* *Рис. 7. Компиляция и запуск*

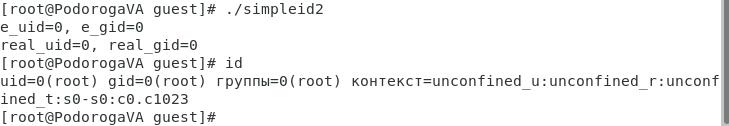
1. От имени суперпользователя выполнил следующие команды:

* 
* *Рис. 8. Изменение атрибутов*

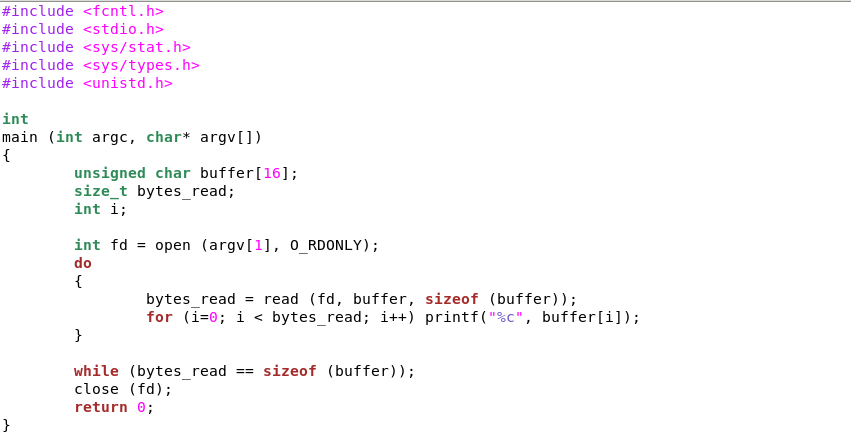
1. С помощью команды chown root:guest /home/guest/simpleid2 удалось изменить права на файл с guest на root. Следующая команда позволяет установить атрибут SetUID, при наличии которого программа запускается с правами владельца файла.
2. Выполнил проверку правильности установки прав и атрибутов:

* 
* *Рис. 10. Установка атрибута i на файл file1*

1. Запустил simpleid2 и id:

* 
* *Рис. 11. Запуск simpleid2 и id*

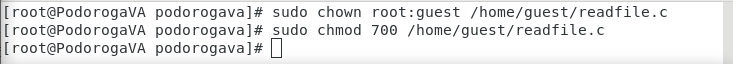
1. Проделал то же самое относительно SetGID-бита.
2. Создал программу readfile.c:

* 
* *Рис. 13.1. Создание файла*
* 
* *Рис. 13.2. Код программы*

1. Выполнил компиляцию:

* 
* *Рис. 14. Компиляция*

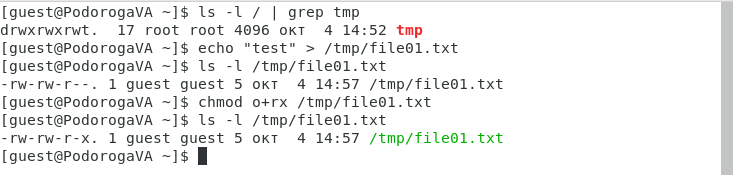
1. Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь мог прочитать его, а guest не мог:

* 
* *Рис. 15. Смена прав и владельца*

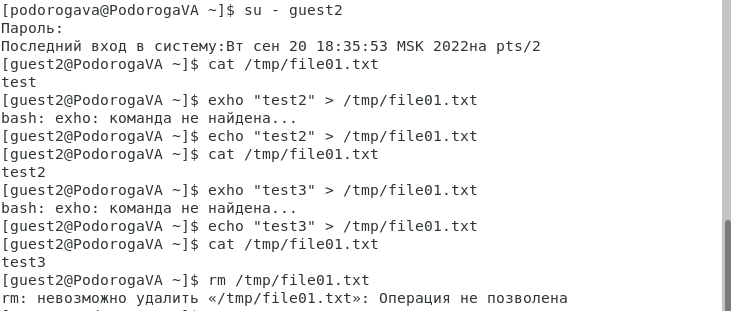
1. Проверил, что пользователь guest не может прочитать файл readfile.c:

* 
* *Рис. 16. Установка атрибута i на файл file1*
* При этом программа может прочитать файл readfile.c и /etc/shadow

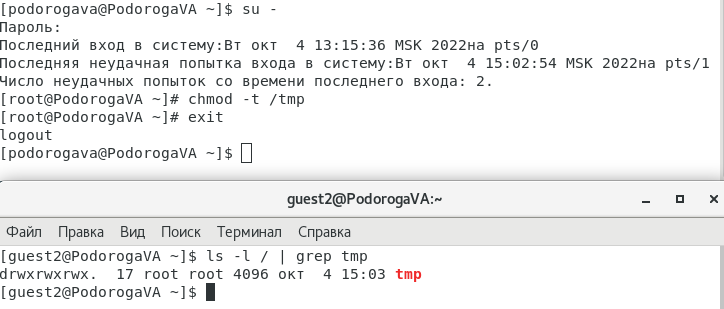
1. Выяснил, установлен ли атрибут Stucky на директории /tmp, далее от имени пользователя guest создал файл file01.txt со словом test, посмотрел атрибуты этого файла и разрешил чтение и запись для категории пользователей "все остальные", проверил правильность установки атрибутов:

* 
* *Рис. 17. Исследование Sticky-бита*

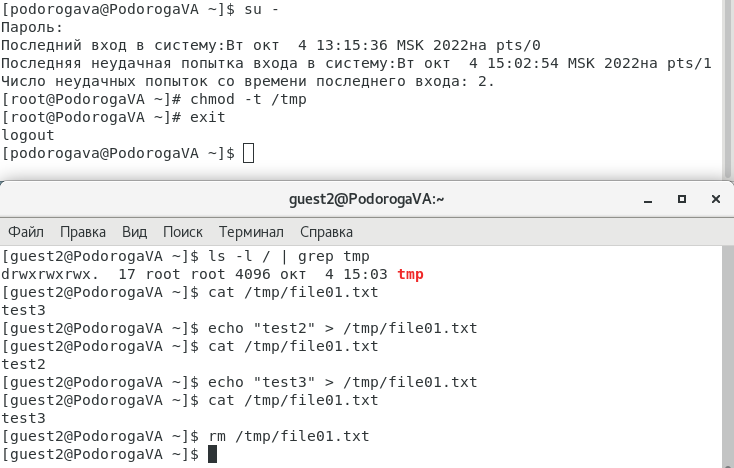
1. От пользователя guest2 попробовал прочитать файл (удалось), попробовал дозаписать в файл test2 (удалось перезаписать), снова проверил содержимое и заменил содержимое файла на test3. Все эти операции выполнить удалось, а удалить - не получилось:

* 
* *Рис. 18. Операции над файлом*

1. Повысил права до суперпользователя и снял Sticky-бит с использованием chmod, затем вышел из режима суперпользователя:

* 
* *Рис. 19. Снятие Sticky-бита*

1. Попробовал снова выполнить вышеописанные операции с файлом file01.txt, на этот раз всё получилось:

* 
* *Рис. 20. Операции над файлом без атрибута t на директории tmp*

1. Повысил права до суперпользователя, вернул атрибут t на директорию:

* 
* *Рис. 21. Возвращение атрибута t на директорию tmp*

# Вывод

В результате выполнения работы я изучил механизмы изменения идентификаторов, использования SetUID-бита и Sticky-бита, получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работу механизма смены идентификатора процессов пользователей, а также механизм влияния Sticky-бита на возможные операции над файлами внутри директории.