Отчёт

по лабораторной работе 5

Лекомцева Алёна

Содержание

[Цель работы 1](#_Toc95083934)

[Задание 1](#_Toc95083935)

[Выполнение лабораторной работы 1](#_Toc95083936)

[Выводы 7](#_Toc95083937)

# Цель работы

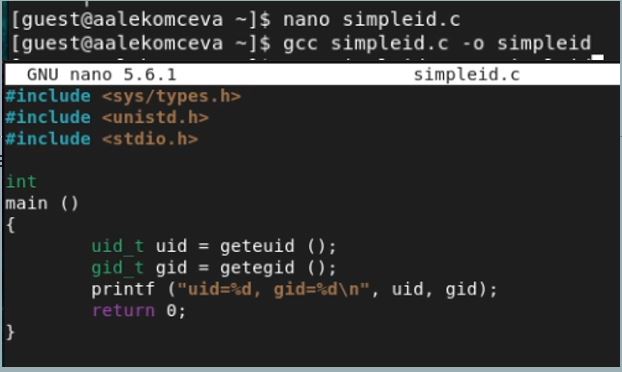
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в кон- соли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# Задание

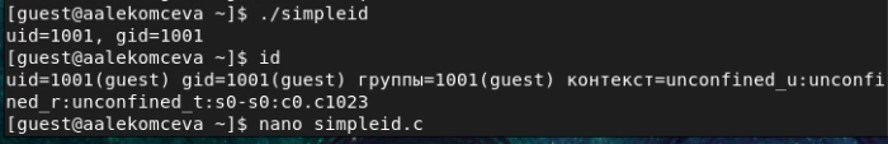
Лабораторная работа подразумевает создание программ и использование Sticky-бита.

# Выполнение лабораторной работы

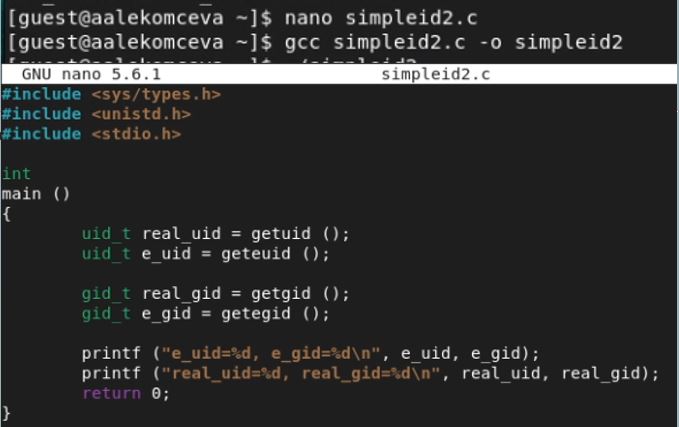
1. Войдем в систему от имени пользователя guest. (рис.1).
2. Создаем программу simpleid.c и компилируем ее. (рис.1).

* 
* рис.1. Программа simpleid.c.

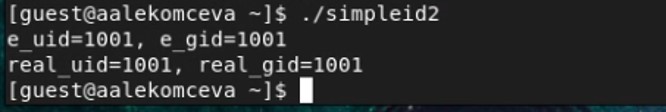
1. Выполним программы simpleid и id. Сравним полученныt нами результаты. Они совпадают. (рис.2).

* 
* рис.2. Выполнение программы.

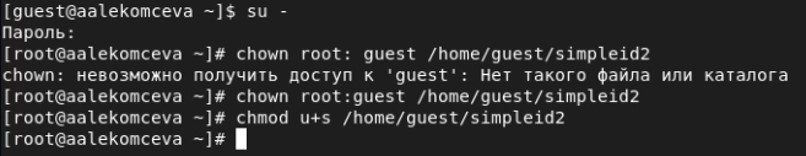
1. Усложним программу, добавив вывод действительных идентификаторов. Получившуюся программу назовем simpleid2.c. (рис.3).

* 
* рис.3. Программа simpleid2.c.

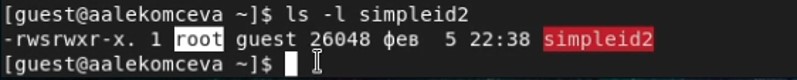
1. Скомпилируем и запустим simpleid2.c. (рис.4).

* 
* рис.4. Выполнение программы simpleid2.c.

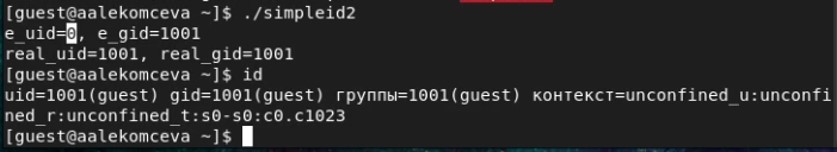
1. От имени суперпользователя выполним команды: chown root:guest /home/guest/simpleid2 и chmod u+s /home/guest/simpleid2. Первая меняет владельца файла, а вторая устанавливает SetUID-бит. (рис.5).

* 
* рис.5. Смена владельца и атрибутов.

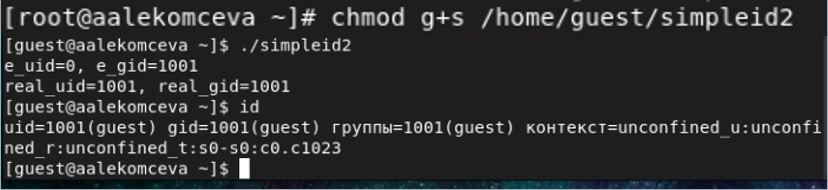
1. Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (рис.6).

* 
* рис.6. Проверка смены владельца и атрибутов.

1. Запустим simpleid2 и id и сравним результат. Результаты одинаковы. (рис.7).

* 
* рис.7. Программы simpleid2 и id.

1. Проделаем тоже самое относительно SetGID-бита (рис.8).

* 
* рис.8. Действия относительно SetGID-бита.

1. Создаем программу readfile.c и компилируем ее. (рис.9).

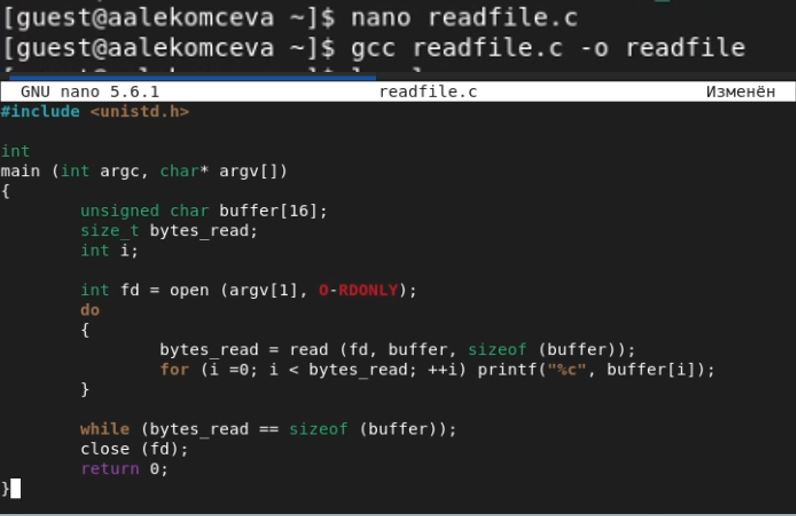


рис.9. Программа readfile.c.

1. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. Проверим, что пользователь guest не может прочитать файл readfile.c. (рис.10).

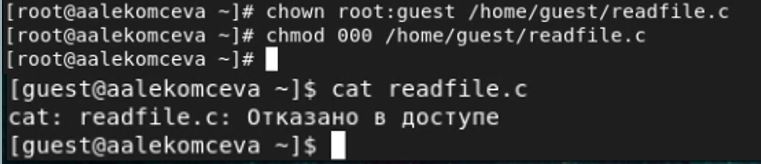


рис.10. Работа с программой readfile.с.

1. Сменим у программы readfile владельца и установим SetUID-бит. Проверим, может ли программа readfile прочитать файл readfile.c. Может. (рис.11).

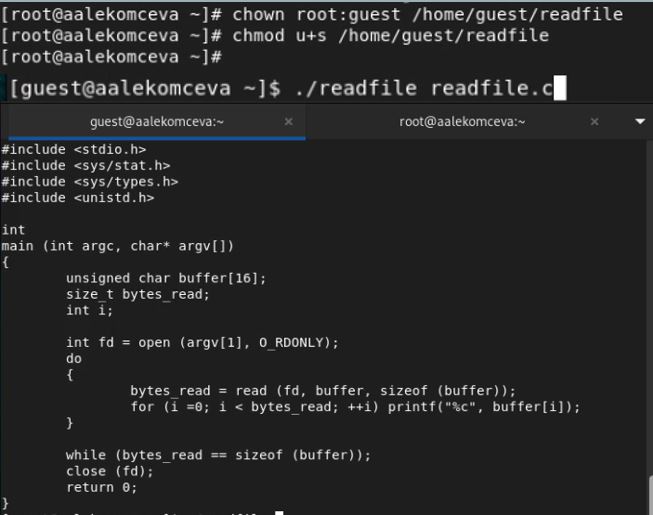


рис.11. Программа readfile читает файл readfile.с.

1. Проверим, может ли программа readfile прочитать файл /etc/shadow. Может. (рис.12).

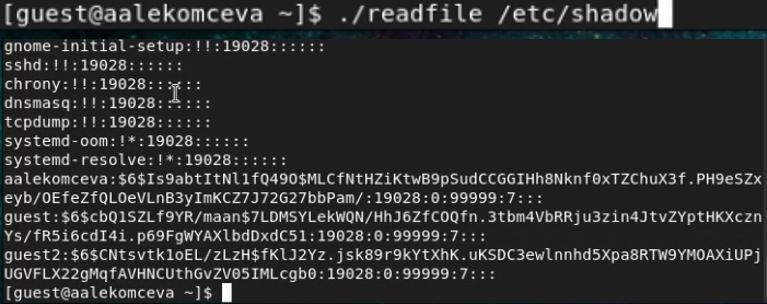


рис.12. Программа readfile читает файл /etc/shadow.

1. Выясним, установлен ли атрибут Sticky на директории /tmp/. Уствновлен. От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные». Проверим правильность выполнения команд. (рис.13).

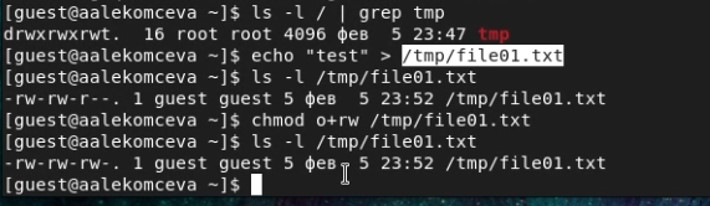


рис.13. Атрибут Sticky.

1. От пользователя guest2 (не являющегося владельцем) попробуем прочитать, дозаписать, перезаписать и удалить файл /tmp/file01.txt. Удалось выполнить все команды, кроме удаления файла. (рис.14).

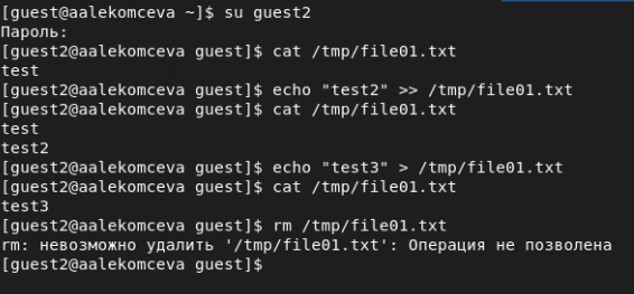
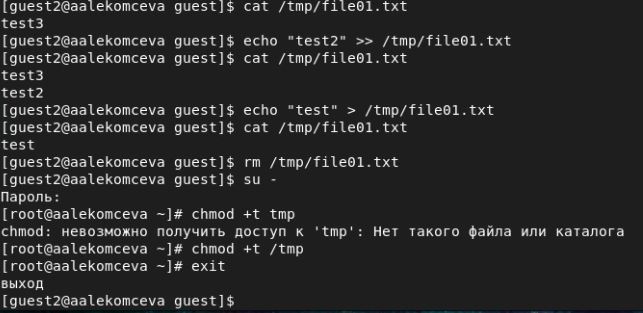


рис.14. Атрибут Sticky.

1. Повысим свои права до суперпользователя командой su - и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Повторим предыдущие шаги от пользователя guest2. Нам удалось удалить файл от имени пользователя, не являющегося его владельцем. (рис.15).

* 
* рис.15. Выполнение команд без атрибута Sticky.

# Выводы

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