Отчёта по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Акондзо Жордани Лади Гаэл

Содержание

# Цель работы

* Изучение механизмов изменения идентификаторов, применение SetUID- и SetGID-битов.
* Изучение действия Sticky-бита на запись и удаление файлов в общей директории.

# Выполнение лабораторной работы

## Часть 1: Изучение SetUID- и SetGID-битов

### Вопросы и Решение

1. Вошёл в систему от имени пользователя guest (рис. 1).

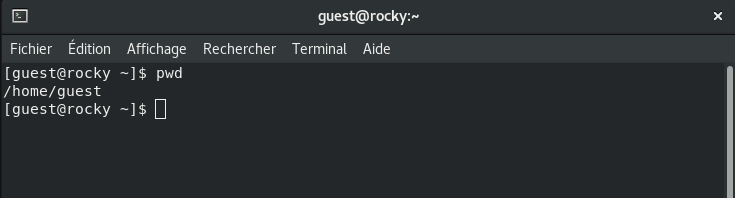


Рис. 1: Вход в систему

1. **Создание программы simpleid.c и её компиляция:**

* Программа simpleid.c выводит эффективный идентификатор пользователя (UID) и группы (GID). Она компилируется командой: (рис. 2) и (рис. 3)

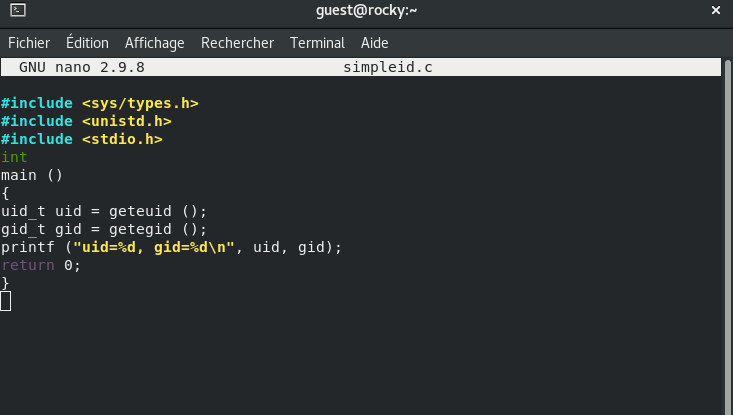


Рис. 2: Создание программы simpleid.c и её компиляция

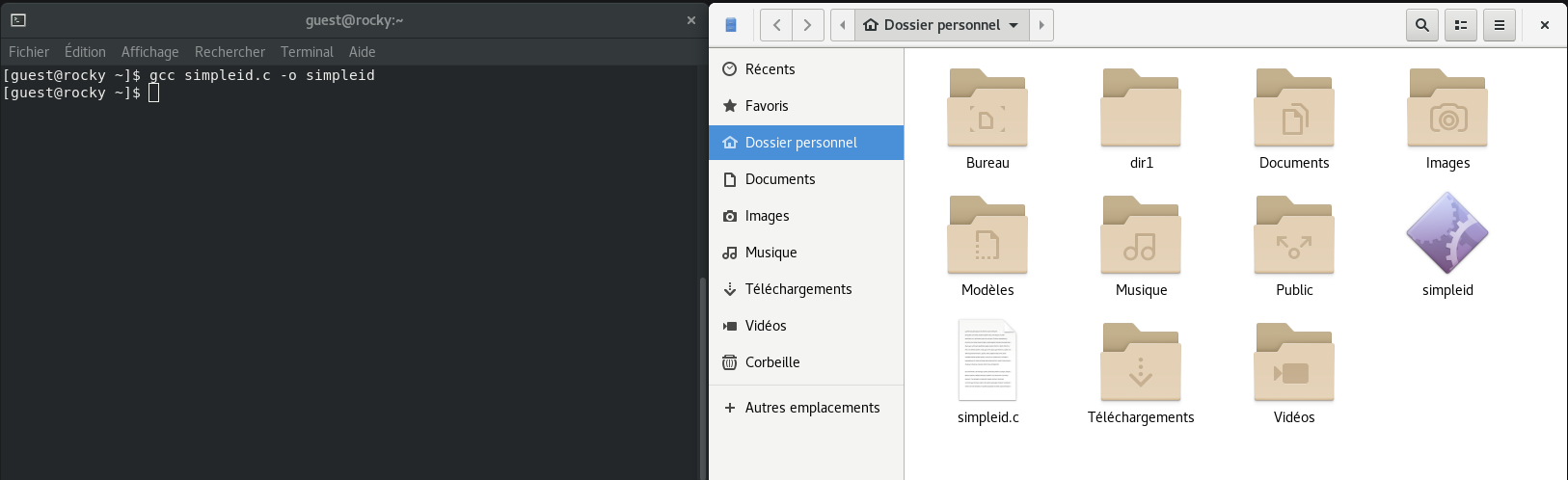


Рис. 3: Создание программы simpleid.c и её компиляция

1. **Запуск программы simpleid и сравнение результата с системной командой id:**

* Программа simpleid и команда id показывают текущий эффективный UID и GID пользователя, что должно совпадать. (рис. 4) и (рис. 5)

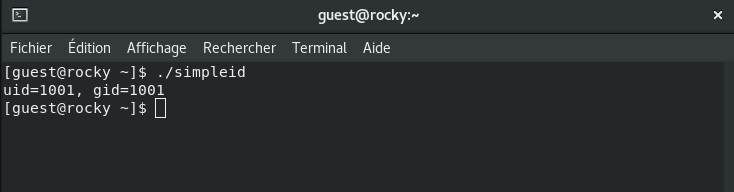


Рис. 4: Запуск программы

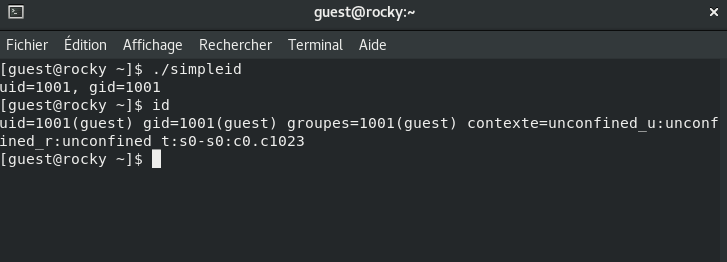


Рис. 5: Сравнение результата с системной командой id

1. **Усложнение программы, создание simpleid2.c:**

* Добавил вывод реальных идентификаторов (UID и GID) и повторно компилируйте: (рис. 6), (рис. 7) и (рис. 8)

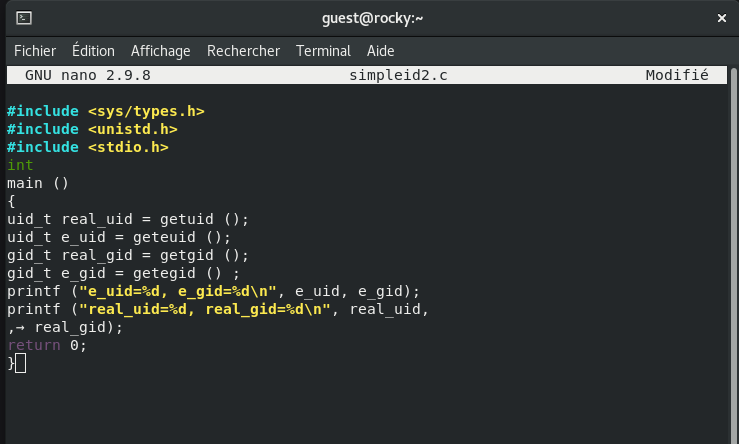


Рис. 6: Усложнение программы, создание simpleid2.c

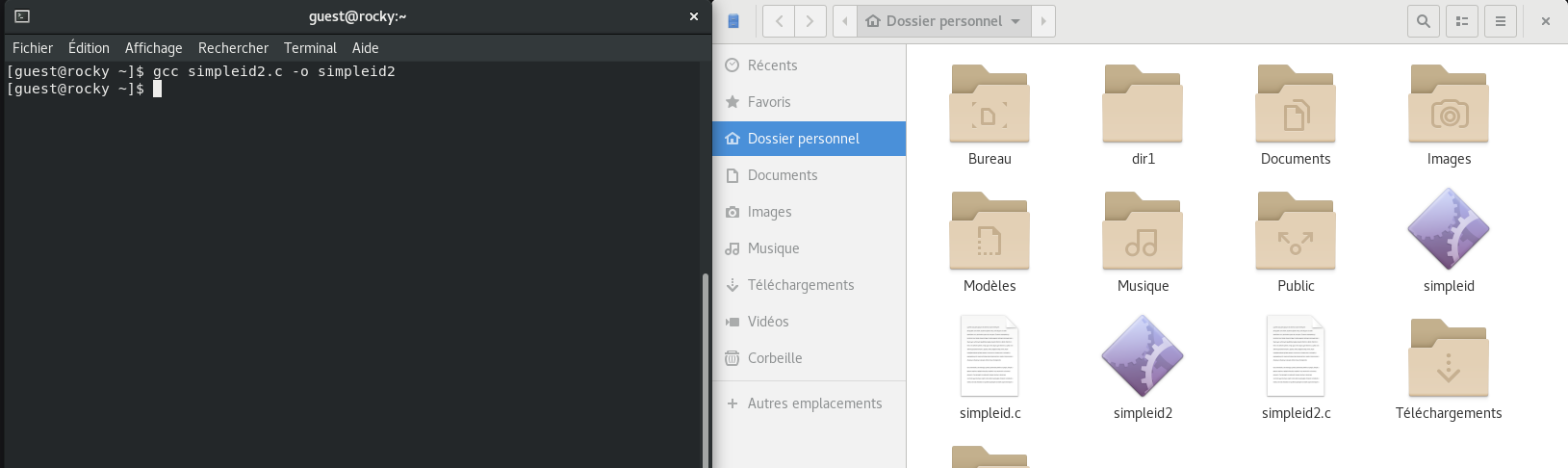


Рис. 7: Усложнение программы, создание simpleid2.c

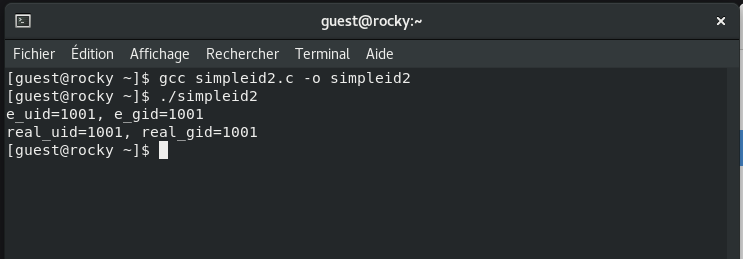


Рис. 8: Усложнение программы, создание simpleid2.c

1. **Изменение владельца программы simpleid2 на root и установка SetUID-бита:**

* Использовал следующие команды: (рис. 9)

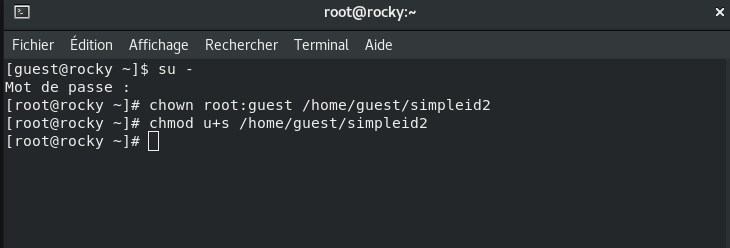


Рис. 9: Изменение владельца программы

* Использовал sudo или повысьте временно свои права с помощью su (рис. 10).

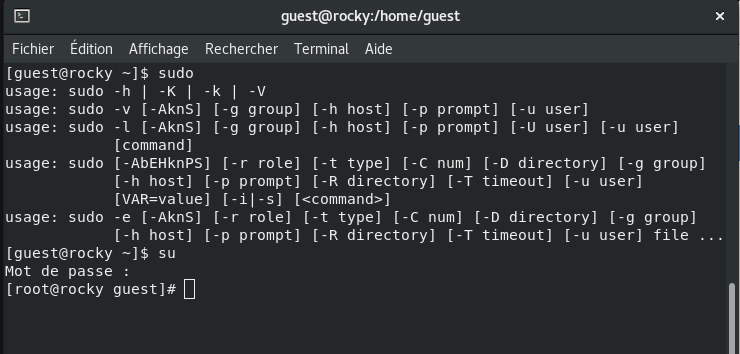


Рис. 10: Изменение владельца программы

* Это позволит программе выполняться с правами пользователя root, независимо от того, кто её запускает.

1. **Запуск simpleid2 и id для сравнения результатов:**

* Сначала выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2. (рис. 11)



Рис. 11: Запуск simpleid2 и id для сравнения результатов

* Потом запустил программу и убедился, что эффективный UID соответствует root, даже если программа запускается обычным пользователем, что подтверждает работу SetUID (рис. 12).

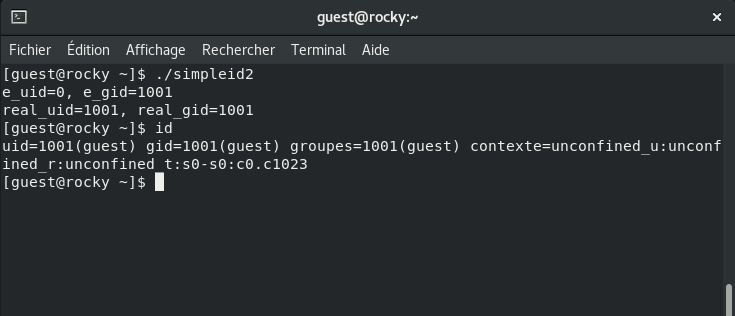


Рис. 12: Запуск simpleid2 и id для сравнения результатов

1. **Применение SetGID-бита к simpleid2:**

* Изменил группу файла и установил SetGID: (рис. 13)

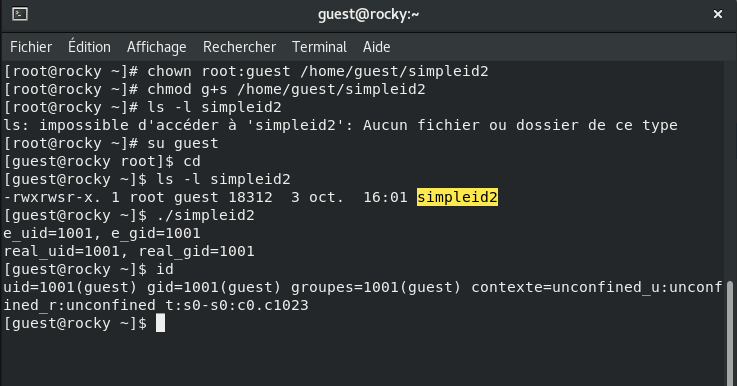


Рис. 13: Применение SetGID-бита к simpleid2

* Программа будет выполняться с правами группы guest. Сравниk результаты ./simpleid2 и id, чтобы увидел, что GID соответствует группе владельца, указанной в SetGID.

1. **Создание программы readfile.c для чтения файла:**

* Создал программу readfile.c, которая читает содержимое файла, и скомпилируйте её: (рис. 14) и (рис. 15)

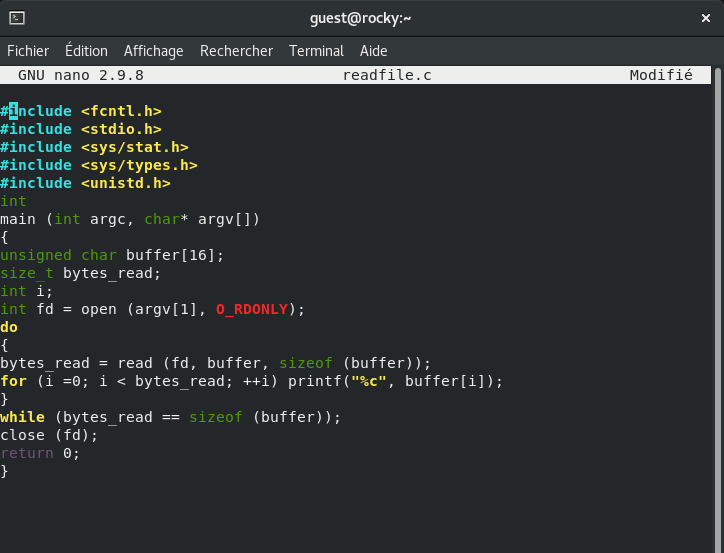


Рис. 14: Создание программы readfile.c для чтения файла

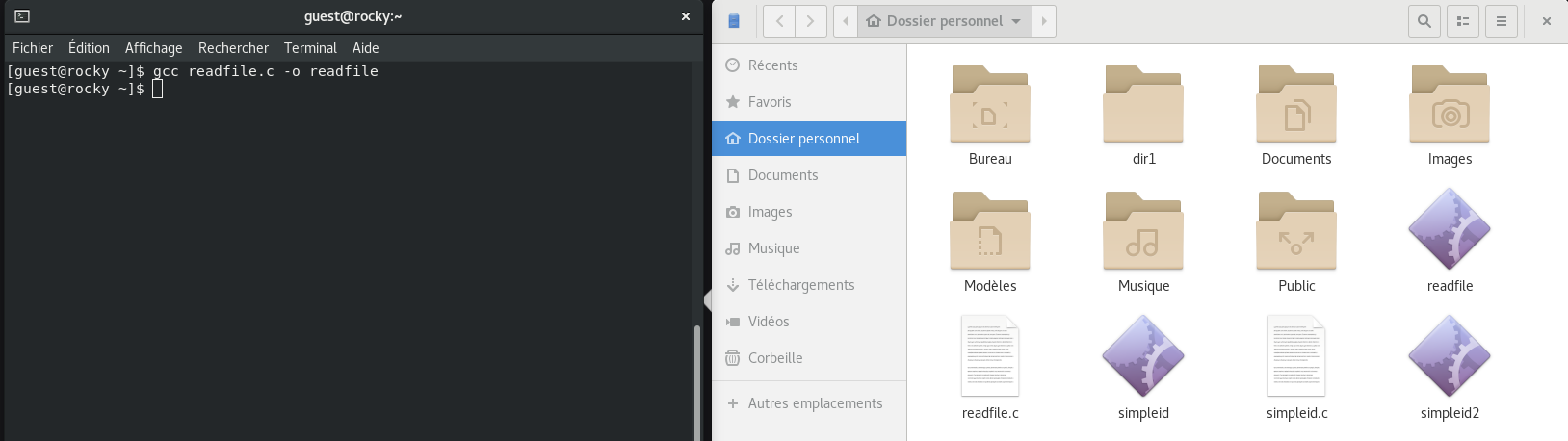


Рис. 15: Создание программы readfile.c для чтения файла

1. **Изменение владельца файла и прав доступа:**

* Сменил владельца файла readfile.c на root, установил права так, чтобы только root мог его читать: (рис. 16)

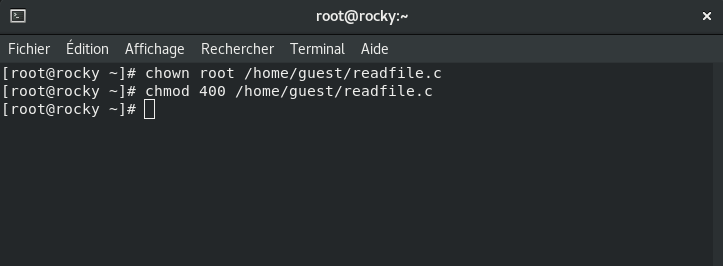


Рис. 16: Изменение владельца файла и прав доступа

* Проверил, что пользователь guest не может прочитать файл readfile.c (должна быть ошибка “Permission denied”) (рис. 17).

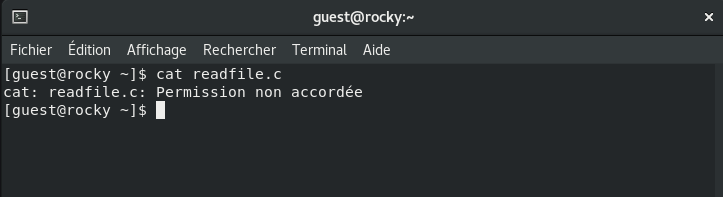


Рис. 17: Проверка прочитания

1. **Установка SetUID-бита на программу readfile:**

* Сменил владельца на root и установите SetUID: (рис. 18)

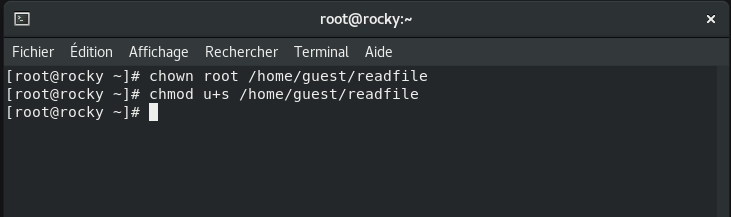


Рис. 18: Установка SetUID-бита на программу readfile

* Теперь программа будет выполняться с правами root.

1. **Проверка, может ли программа readfile прочитать файл readfile.c:**

* Запустите readfile как guest: (рис. 19)

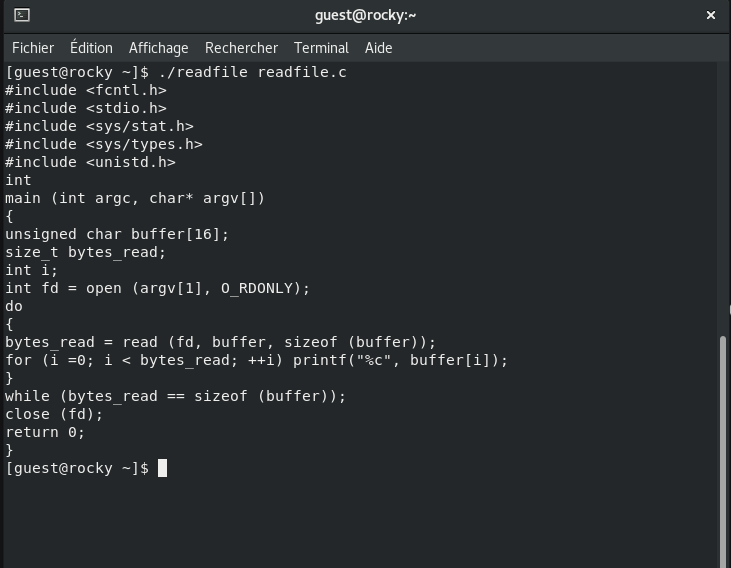


Рис. 19: Проверка, может ли программа readfile прочитать файл readfile.c

* Программа должна успешно прочитать файл благодаря SetUID-биту.

1. **Проверка, может ли программа readfile прочитать файл /etc/shadow:**

* Запустил ./readfile /etc/shadow. Если SetUID-бит установлен, программа сможет читать этот файл, что демонстрирует возможность использования SetUID для доступа к привилегированным данным (рис. 20).

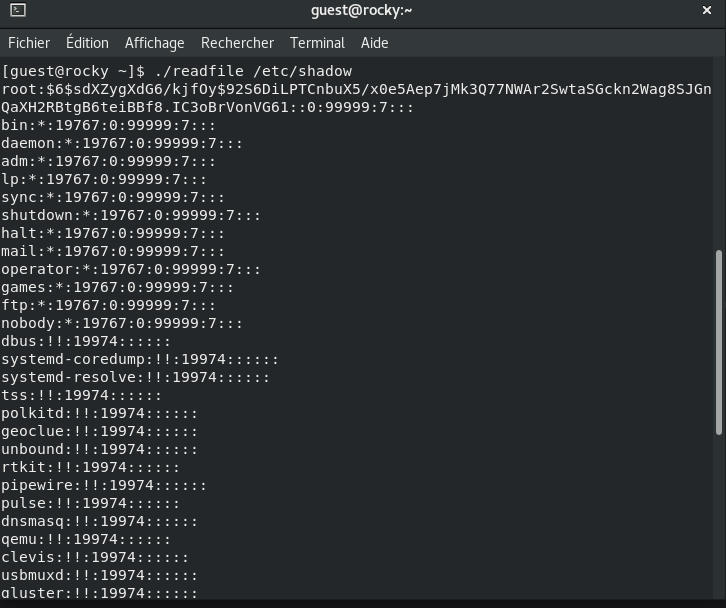


Рис. 20: Проверка, может ли программа readfile прочитать файл /etc/shadow

## Часть 2: Исследование Sticky-бита

### Вопросы и Решение

1. **Проверка наличия Sticky-бита на директории /tmp:**

* Выполнил команду: (рис. 21)

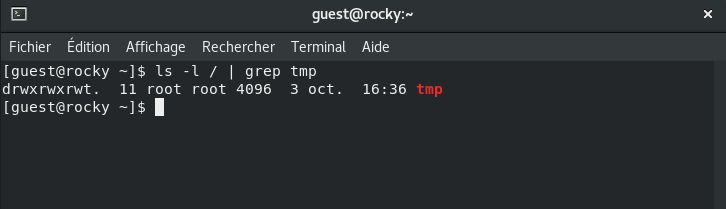


Рис. 21: Проверка наличия Sticky-бита на директории /tmp

* Sticky-бит (t) указывает, что только владелец файла или root может удалить его, даже если у других пользователей есть права на запись.

1. **Создание файла в /tmp и изменение прав доступа:**

* Как пользователь guest, создайте файл file01.txt в /tmp и разрешил запись всем: (рис. 22) и (рис. 23)

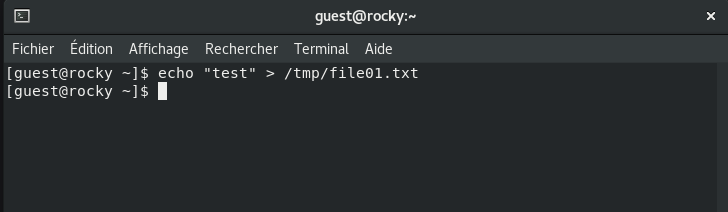


Рис. 22: Создание файла в /tmp и изменение прав доступа

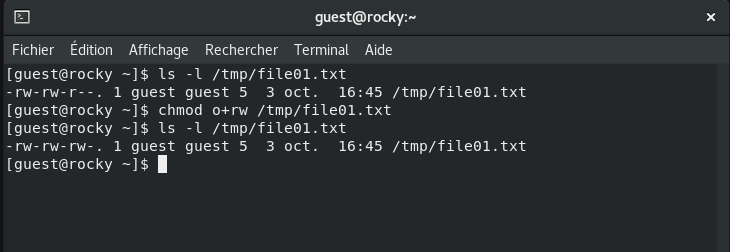


Рис. 23: Создание файла в /tmp и изменение прав доступа

1. **Чтение и изменение файла другим пользователем guest2:**

* Как пользователь guest2, попытался: (рис. 24)

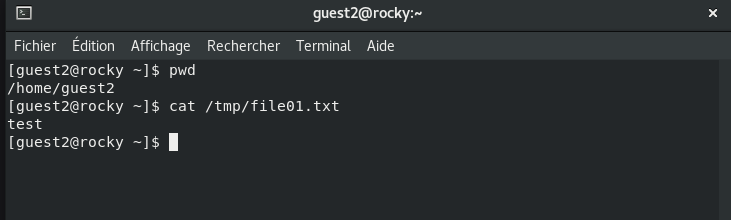


Рис. 24: Чтение и изменение файла другим пользователем guest2

* Чтение и запись должны быть успешными, но удаление файла будет невозможно благодаря Sticky-биту (рис. 25), (рис. 26), (рис. 27), (рис. 28) и (рис. 29).

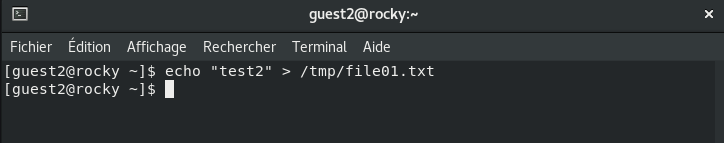


Рис. 25: Чтение и изменение файла другим пользователем guest2

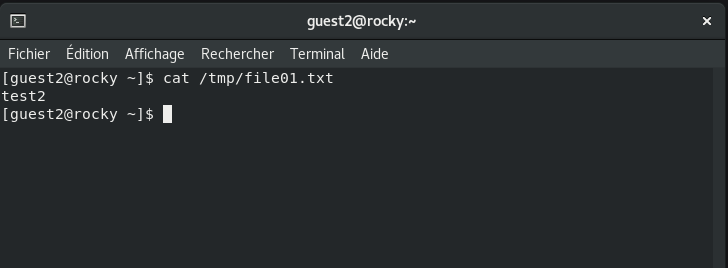


Рис. 26: Чтение и изменение файла другим пользователем guest2

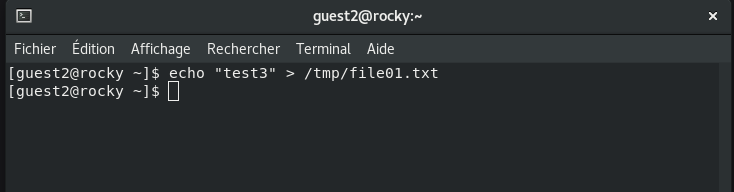


Рис. 27: Чтение и изменение файла другим пользователем guest2



Рис. 28: Чтение и изменение файла другим пользователем guest2

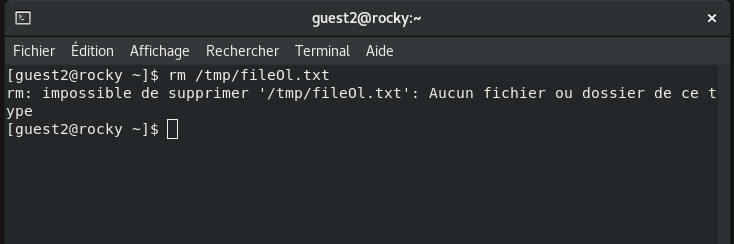


Рис. 29: Чтение и изменение файла другим пользователем guest2

1. **Удаление Sticky-бита с /tmp:**

* Как суперпользователь, снимите Sticky-бит: (рис. 30)

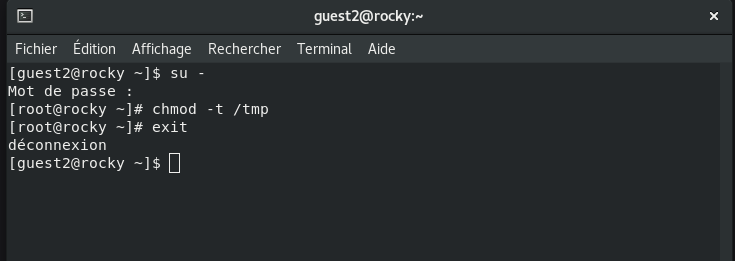


Рис. 30: Удаление Sticky-бита с /tmp

* От пользователя guest2 проверил, что атрибута t у директории /tmp нет: (рис. 31)



Рис. 31: Проверка отрибуты

* Повторил попытку удаления файла от пользователя guest2. Теперь файл должен быть удалён, так как Sticky-бит больше не защищает его, но что-то не так.

1. **Возвращение Sticky-бита на /tmp:**

* Повторно установил Sticky-бит: (рис. 32)



Рис. 32: Возвращение Sticky-бита на /tmp

* Теперь повторил действия и убедился, что удаление файла снова запрещено для пользователей, не являющихся его владельцем (рис. 33).



Рис. 33: Повторение действия

# Выводы

В ходе этой лабораторной работы было продемонстрировано, как использование **SetUID, SetGID, и Sticky-бито** позволяет управлять доступом и правами пользователей в системе Linux. **SetUID и SetGID** позволяют временно выполнять программы с правами владельца файла или его группы, что полезно для выполнения привилегированных задач. Sticky-бит защищает файлы в общих директориях, таких как /tmp, от удаления пользователями, не являющимися владельцами, что предотвращает потенциальные конфликты и нарушения безопасности.