Отчёт по лабораторной работе 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Радимов Игорь

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.

# Задание

Лабораторная работа подразумевает изучение влияния дополнительных атрибутов на файлы пользователя и изучение механизмов изменения идентификаторов.

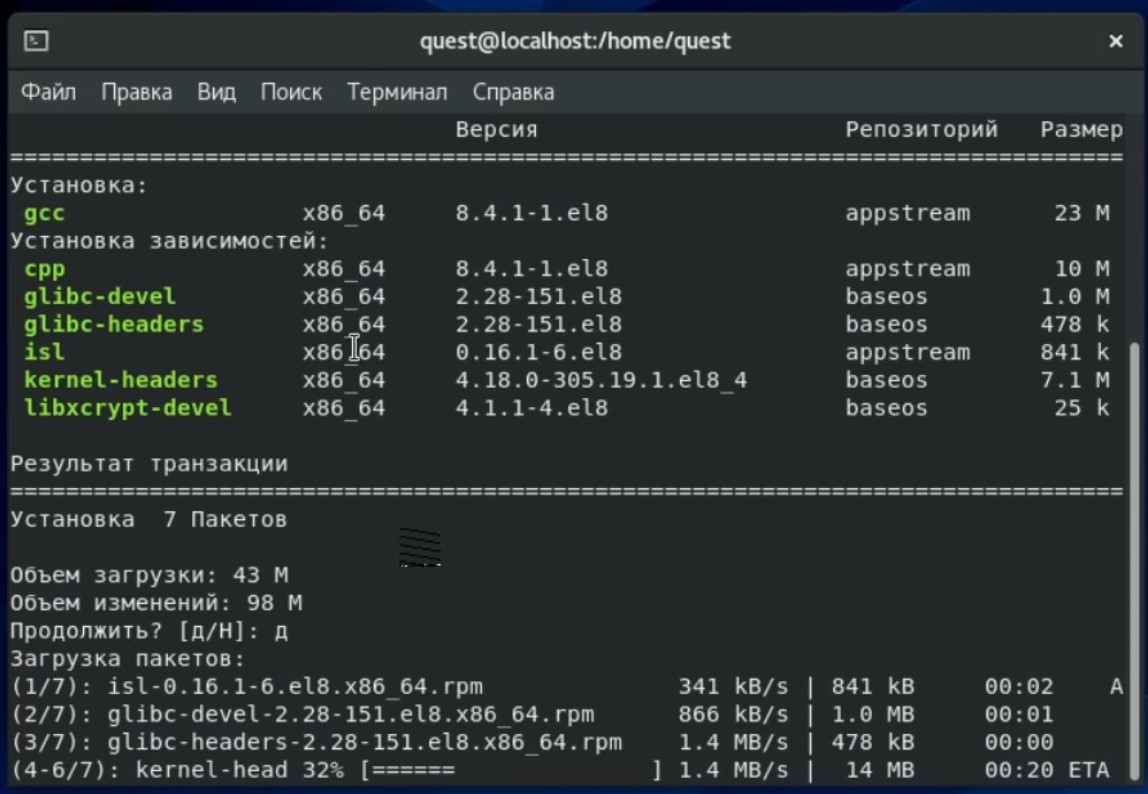
# Теория

Биты SUID, SGID и Sticky Unix отслеживает не символьные имена владельцев и групп, а их идентификаторы (UID - для пользователей и GID для групп). Эти идентификаторы хранятся в файлах /etc/passwd и /etc/group соответственно. Установка битов SUID или SGID позволит пользователям запускать исполняемые файлы от имени владельца (или группы) запускаемого файла. Если мы установим SUID на исполняемый файл /bin/chmod, то обычный пользователь сможет использовать эту команду без использования sudo, так, что она будет выполнятся от имени пользователя root. Каталог с установленным sticky-битом означает, что удалить файл из этого каталога может только владелец файла или суперпользователь. Другие пользователи лишаются права удалять файлы.

# Выполнение лабораторной работы

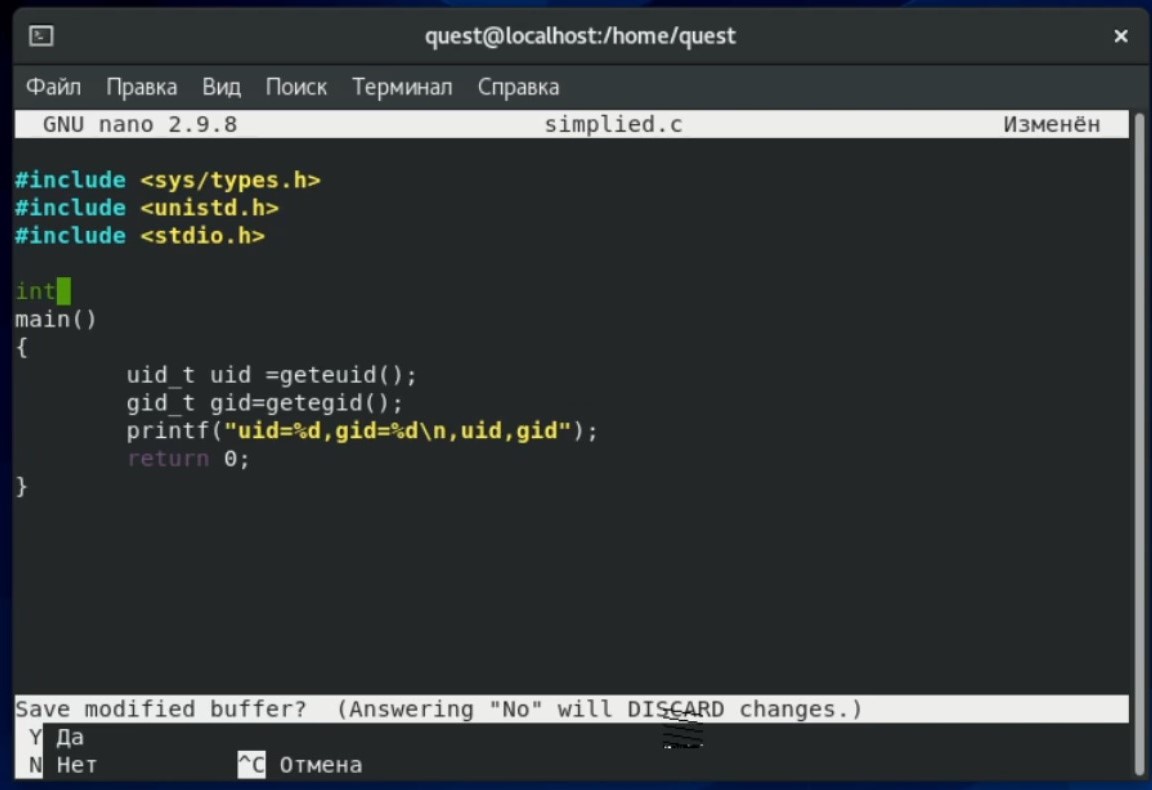
## Подготовка к выполнению

1. Установил компилятор gcc.

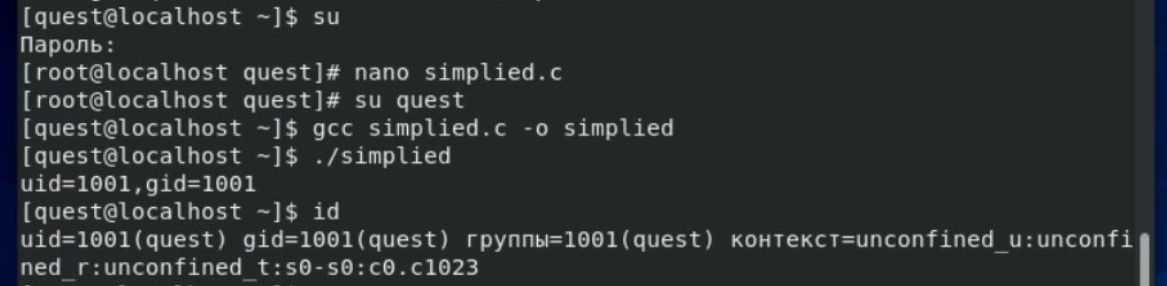
* 
* Figure 1: рис.1. Установка компилятора gcc.

## Выполнение основной частии лабораторной работы

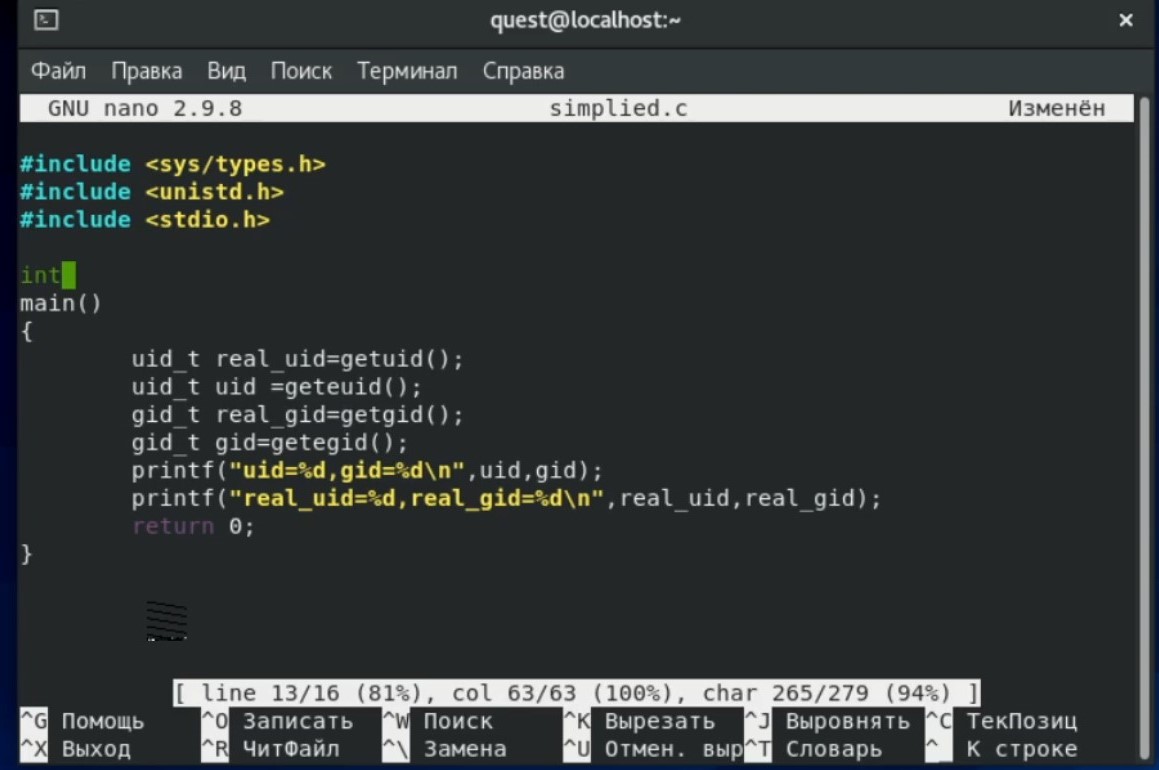
1. Вошёл в систему от имени пользователя quest.
2. Создал программу simpleid.c (рис. 2).

* 
* Figure 2: рис.2. Программа simpleid.c

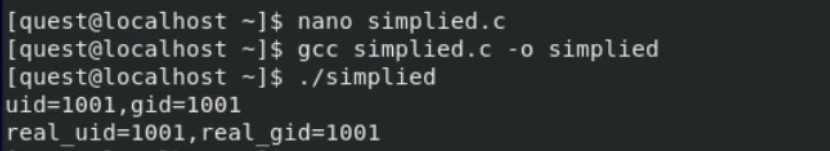
1. Скомплилировал программу и убедился, что файл программы создан командой gcc simpleid.c -o simpleid (рис. 3).
2. Выполнил программу simpleid командой ./simpleid (рис. 3).
3. Выполнил системную программу id и сравнил полученный результат с данными предыдущего пункта задания (рис. 3). Видим, что пользователи и группы совпадают. При этом команда id вывела действительные идентификаторы, а программа вывел эффективные, но при этом они совпадают и выводят 1001, то есть пользователя quest.

* 
* Figure 3: рис.3. Компиляция программы simpleid, её выполнение и сравнение с командой id.

1. Усложнил программу, добавив вывод действительных идентификаторов, получившуюся программу назвал simpleid2.c (рис. 4).

* 
* Figure 4: рис.4. Программа simpleid2.c

1. Скомпилировал и запустил simpleid2.c командами gcc simpleid2.c -o simpleid2 и ./simpleid2 (рис. 5). Видим, что программа выводит эффективные и действительные идентификаторы пользователя и группы для файла. Видим, что везде это 1001, то есть пользователь quest.

* 
* Figure 5: рис.5. Компиляция программы simpleid2, её выполнение.

8-9. От имени суперпользователя выполнил команды:chown root:quest /home/quest/simpleid2 и chmod u+s /home/quest/simpleid2. Временно поменяв свои права с помощью su (рис. 6).

С помощью этих команд файлу simpleid2 изменил владельца и группу на root и quest соответственно (chown), а также установил на файл SetUID-бит (chmod).

1. Выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой ls -l simpleid2 (рис. 6).
2. Запустил simpleid2 и id командами ./simpleid2 и id (рис. 6). Сравнил результаты: действительные идентификаторы совпадают с выводом команды id - везде 0, то есть рут-пользователь. Так же важно заметить, что эффективные идентификаторы совпадают с действительными.

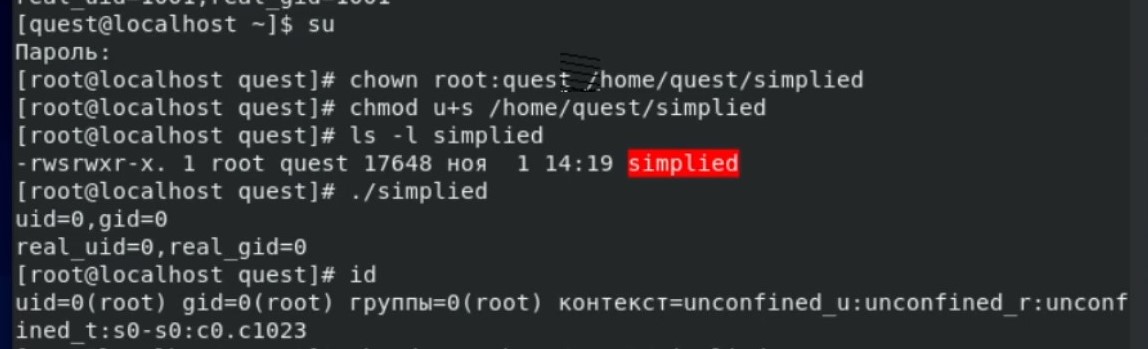


Figure 6: рис.6. Изменение владельца программы и установка SetUID-бита, проверка установки и изменения, запуск программы и команды id.

1. Проделал тоже самое относительно SetGID-бита (рис. 7)

Установка SetGID-бита отражается к команде ls, а сравнение выполнения программы и команды id дало следующие результаты: действительные идентификаторы совпадают с выводом команды id - везде 0, то есть рут-пользователь. Но так же важно заметить, что эффективные идентификаторы отличны от действительных: пользователь - 0, группа - 1001.

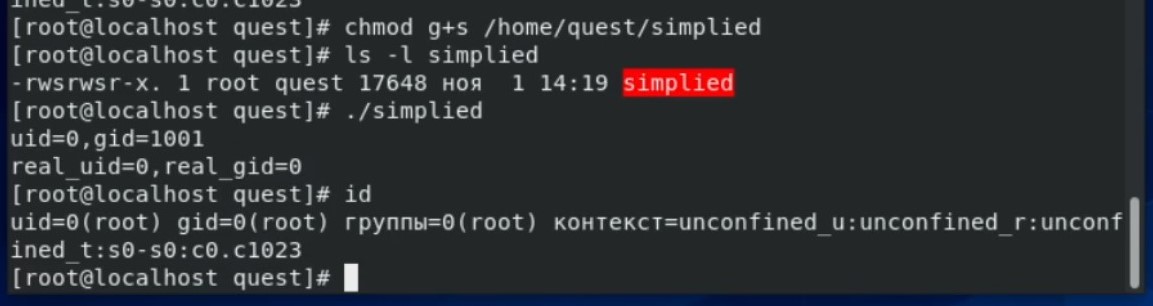


Figure 7: рис.7. Установка SetGID-бита, проверка установки, запуск программы и команды id.

1. Создал программу readfile.c (рис. 8).

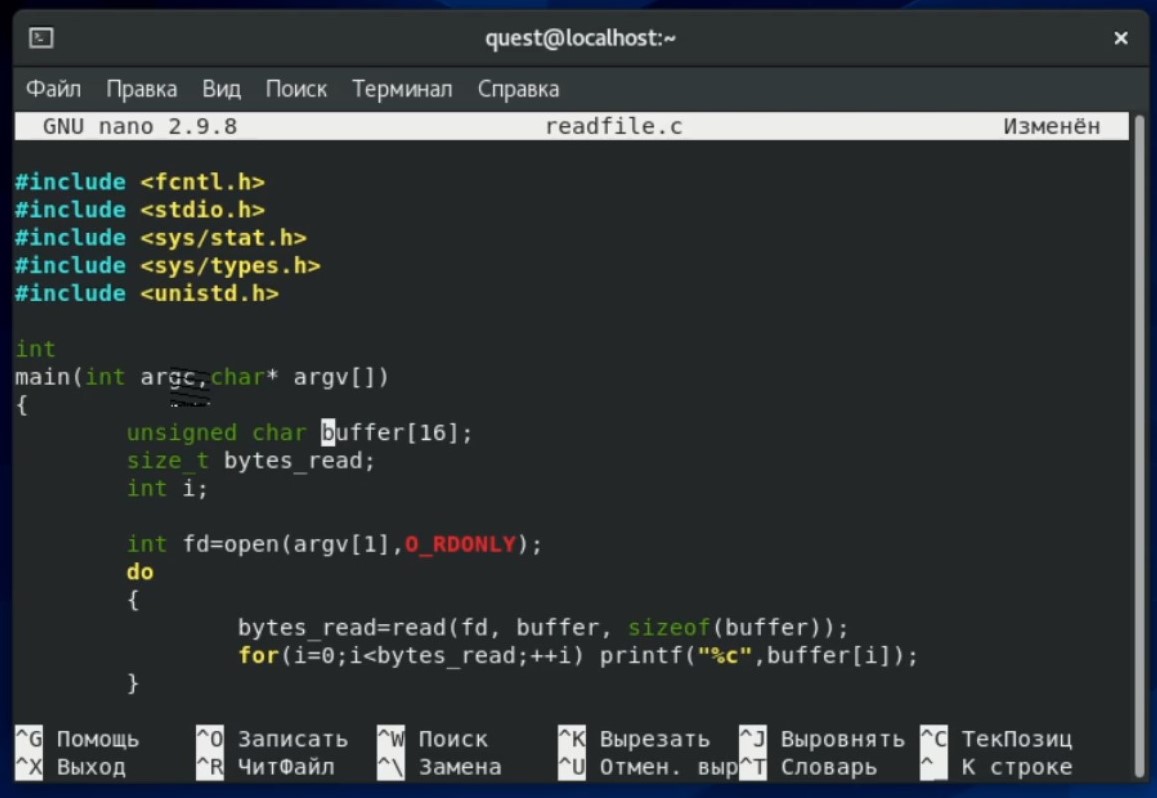


Figure 8: рис.8. Программа readfile.c

1. Откомпилировал её командой gcc readfile.c -o readfile (рис. 9).
2. Сменил владельца у файла readfile.c (chown) и изменил права так, чтобы только суперпользователь (root) мог прочитать его, a quest не мог. Использовал chmod a-r (рис. 9).
3. Проверил, что пользователь quest не может прочитать файл readfile.c командой cat (рис. 9).
4. Сменил у программы readfile владельца и установил SetUID-бит (рис. 9).

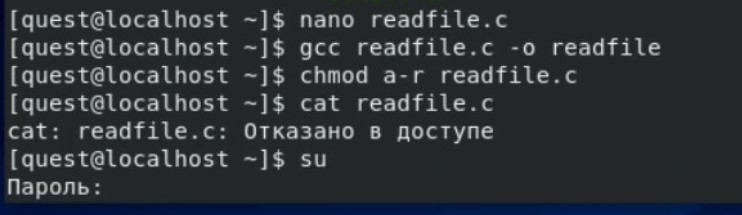


Figure 9: рис.9. Компиляция readfile и другие действия в соответствии с 14-17 пунктами.

1. Проверил, может ли программа readfile прочитать файл readfile.c. Да, может (рис. 10).

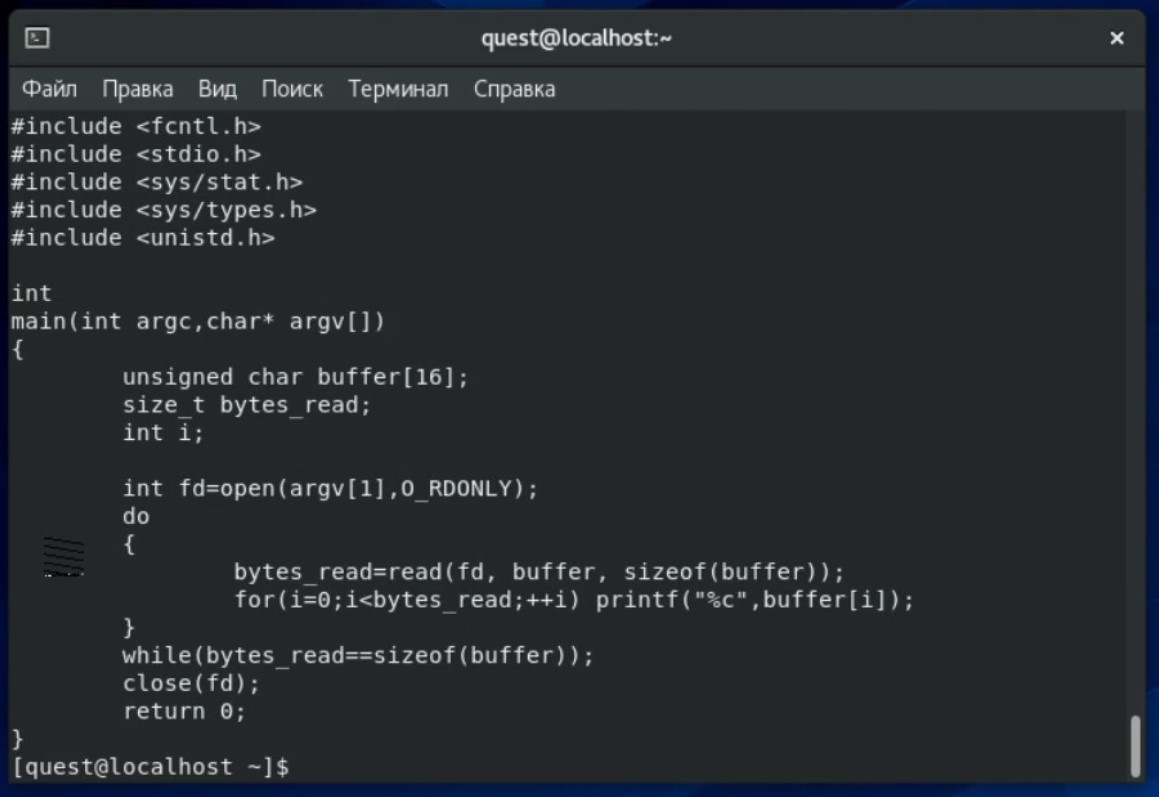


Figure 10: рис.10. Выполнение программы readfile с файлом readfile.c.

1. Проверил, может ли программа readfile прочитать файл /etc/shadow. Её выполненеи возможно в том числе, так как владельцем файла является root-пользователь (рис. 11).

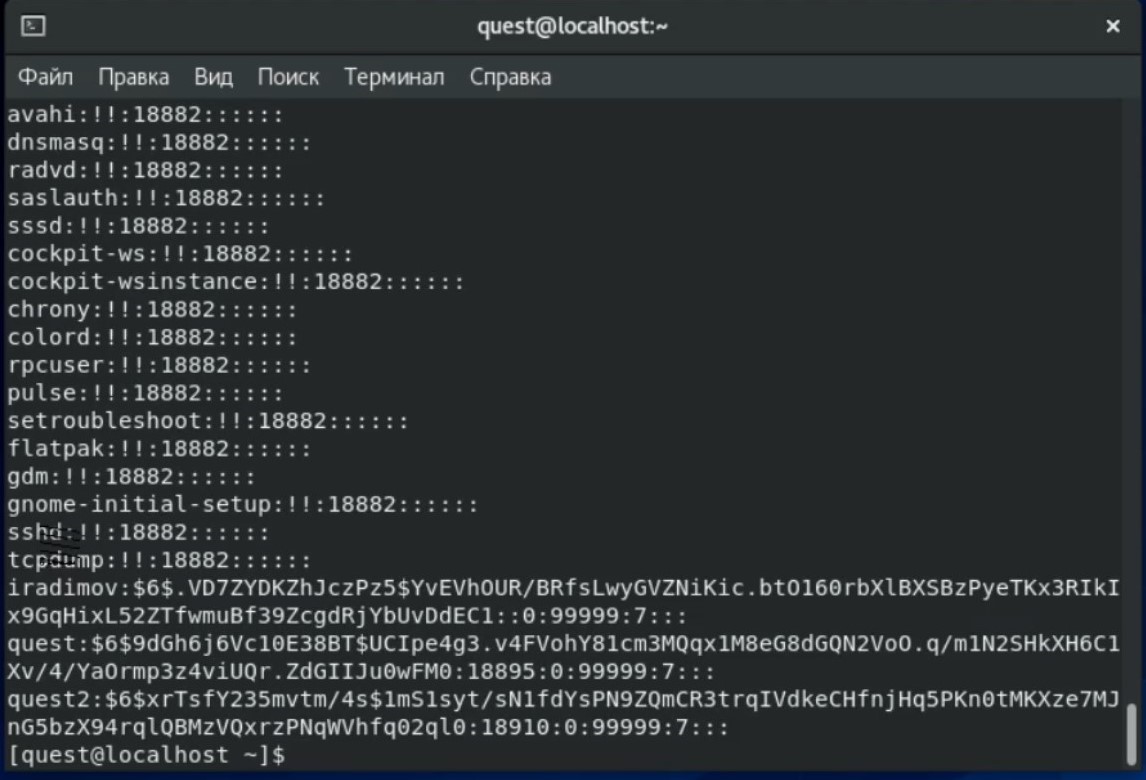
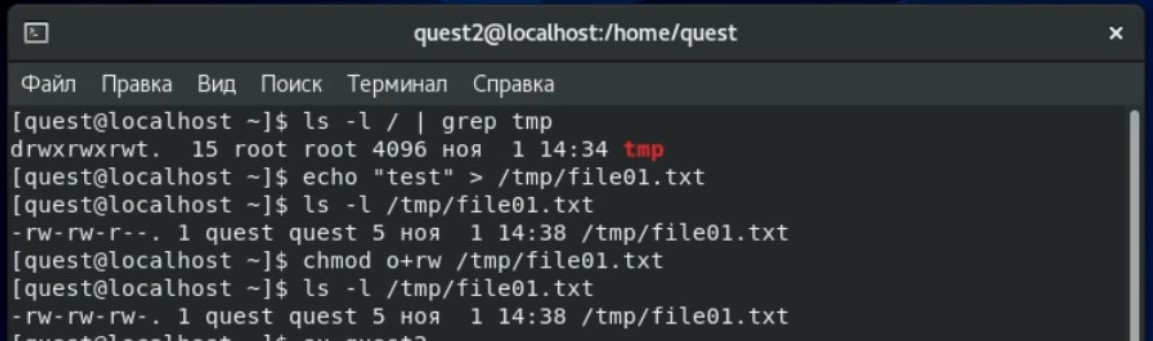


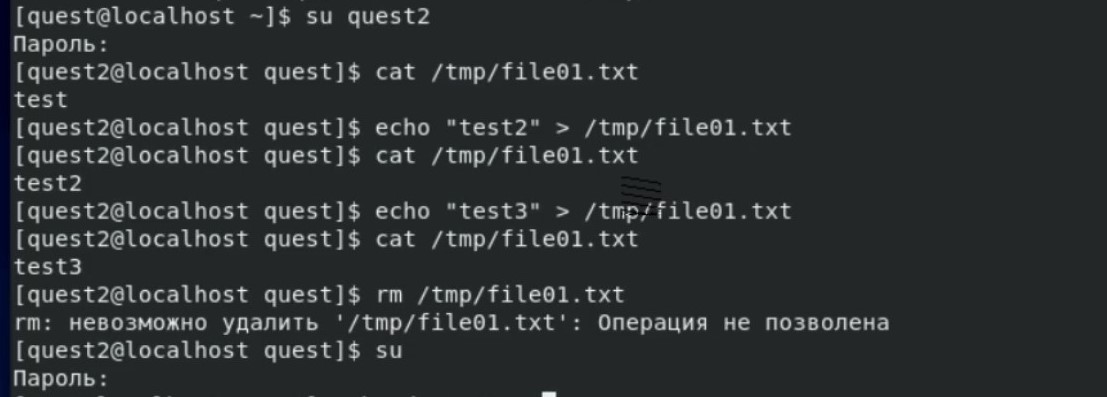
Figure 11: рис.11. Выполнение программы readfile с файлом /etc/shadow.

## Исследование Sticky-бита

1. Выяснилв, установлен ли атрибут Sticky на директории /tmp, для чего выполнил команду ls -l / | grep tmp (рис. 12). Видим, что установлен, так как есть буква t.
2. От имени пользователя quest создал файл file01.txt в директории /tmp со словом test командой echo “test” > /tmp/file01.txt (рис. 12).
3. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные». Выполнил команды ls -l /tmp/file01.txt и chmod o+rw /tmp/file01.txt и ls -l /tmp/file01.txt (рис. 12).
4. От пользователя quest2 (не являющегося владельцем) попробовал прочитать файл /tmp/file01.txt командой cat /tmp/file01.txt (рис. 12).

* 
* Figure 12: рис.12. Выполнение пунктов 1-4 исследования Sticky-бита .

1. От пользователя quest2 попробовал дозаписать в файл /tmp/file01.txt слово test2 командой echo “test2” >> /tmp/file01.txt. Мне удалось выполнить операцию (рис. 13).
2. Проверил содержимое файла командой cat /tmp/file01.txt (рис. 13).
3. От пользователя quest2 попробовал записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo “test3” > /tmp/file01.txt. Мне удалось выполнить операцию (рис. 13).
4. Проверил содержимое файла командой cat /tmp/file01.txt (рис. 13).
5. От пользователя quest2 попробовал удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt. Мне нее удалось удалить файл (рис. 13).

* 
* Figure 13: рис.13. Выполнение пунктов 5-9 исследования Sticky-бита .

Можем сделать вывод, что разрешена дозапись в файл, запись в файл, но мы не можем удалять файл из директории, на которую установлен атрибут Sticky.

1. Повысил свои права до суперпользователя следующей командой su - и выполнил после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp (рис. 14).
2. Покинул режим суперпользователя командой exit (рис. 14).
3. От пользователя quest2 проверил, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис. 14).
4. Повторил предыдущие шаги (рис. 14). Видим, что дозапись и запись так же разрешены, но при этом удалось и удалить файл.

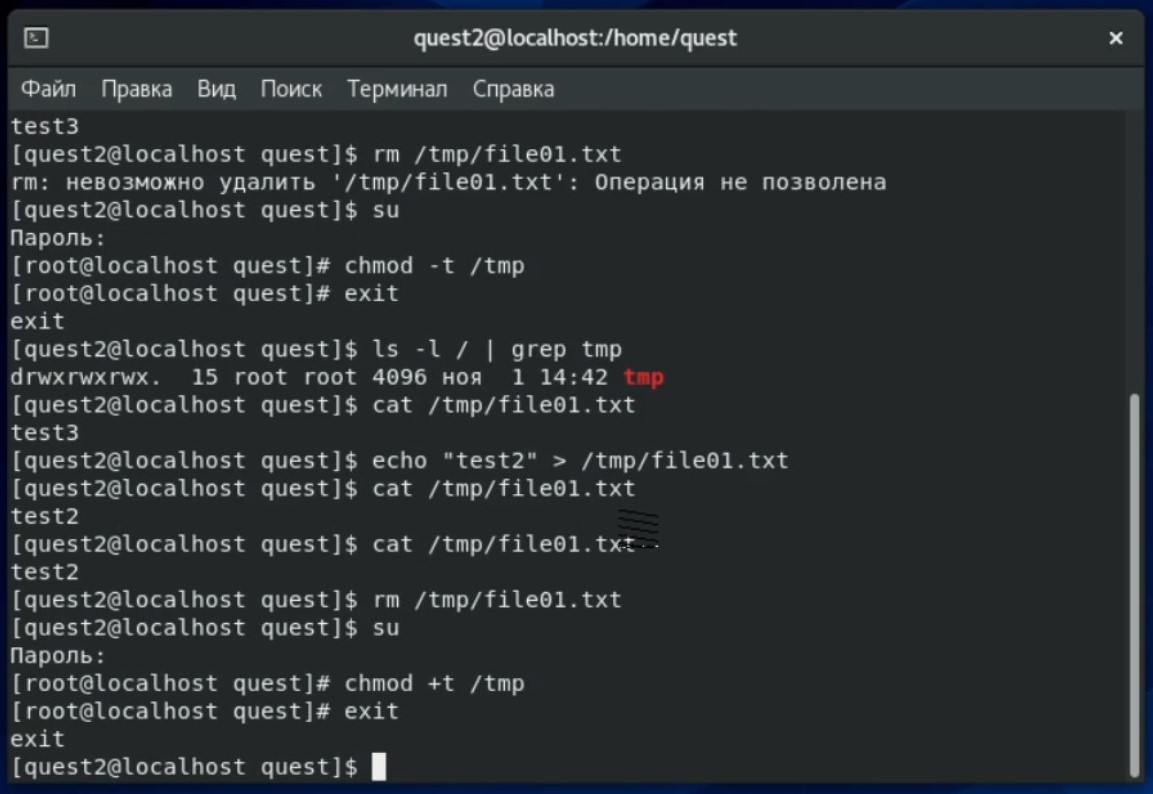


Figure 14: рис.14. Выполнение пунктов 10-13 исследования Sticky-бита .

1. Мне удалось удалить файл от имени пользователя, не являющегося его владельцем.
2. Повысил свои права до суперпользователя и верните атрибут t на директорию /tmp: su -, chmod +t /tmp, exit (рис. 15).

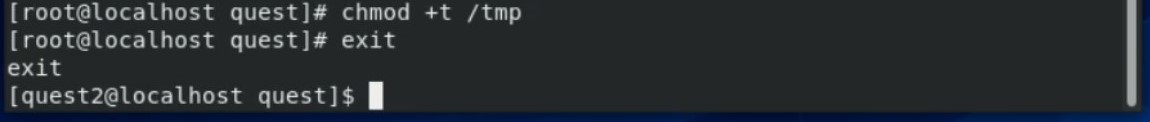


Figure 15: рис.15. Возврашение атрибута t на директорию.

# Библиография

1. ТУИС РУДН
2. https://help.ubuntu.ru/wiki/стандартные\_права\_unix

# Выводы

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