Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.

Пономарева Лилия Михайловна

Содержание

# Цель работы

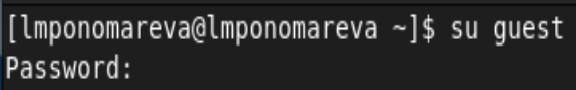
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов [[1]](#список-литературы).

# Теоретическое введение

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. Символьные эквиваленты идентификаторов используются только для удобства, например, при использовании команды ls, идентификаторы заменяются соответствующими символьными обозначениями.  
Что касается процессов, то с ними связано не два идентификатора, а 4-е: реальный и эффективный пользовательский (UID), а также реальный и эффективный групповой (GID). Реальные номера применяются для учета использования системных ресурсов, а эффективные для определения прав доступа к процессам. Как правило, реальные и эффективные идентификаторы совпадают. Владелец процесса может посылать ему сигналы, а также изменять приоритет.  
Процесс не может явно изменить ни одного из своих четырех идентификаторов, но есть ситуации когда происходит косвенная установка новых эффективных идентификаторов процесса. Дело в том, что существуют два специальных бита: SUID (Set User ID - бит смены идентификатора пользователя) и SGID (Set Group ID - бит смены идентификатора группы). Когда пользователь или процесс запускает исполняемый файл с установленным одним из этих битов, файлу временно назначаются права его (файла) владельца или группы (в зависимости от того, какой бит задан). Таким образом, пользователь может даже запускать файлы от имени суперпользователя.  
Вобщем, одним словом установка битов SUID или SGID позволит пользователям запускать исполняемые файлы от имени владельца (или группы) запускаемого файла. Например, как говорилось выше, команду chmod по умолчанию может запускать только root. 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. В некоторых случаях очень удобное решение. Кстати по такому принципу работает команда passwd, c помощью которой пользователь может изменить свой пароль.  
Еще одно важное усовершенствование касается использования sticky-бита в каталогах. Каталог с установленным sticky-битом означает, что удалить файл из этого каталога может только владелец файла или суперпользователь. Другие пользователи лишаются права удалять файлы. Установить sticky-бит в каталоге может только суперпользователь. Sticky-бит каталога, в отличие от sticky-бита файла, остается в каталоге до тех пор, пока владелец каталога или суперпользователь не удалит каталог явно или не применит к нему chmod. Заметьте, что владелец может удалить sticky-бит, но не может его установить.

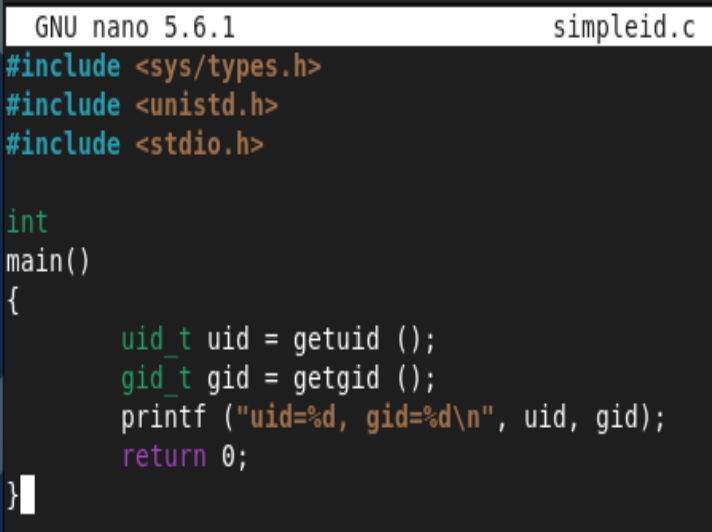
# Выполнение лабораторной работы

1. Вошла в систему от имени пользователя guest([рис. 1](../../image/1.png)).



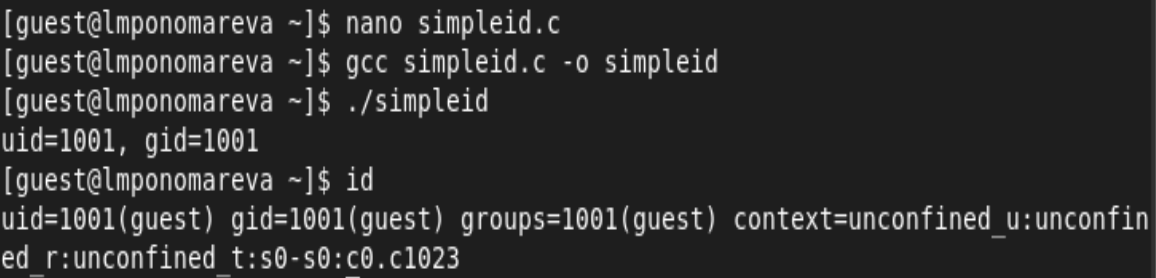
Вход под пользователем guest

1. Создала программу simpleid.c ([рис. 2](../../image/2.png)).



Код программы simpleid.c

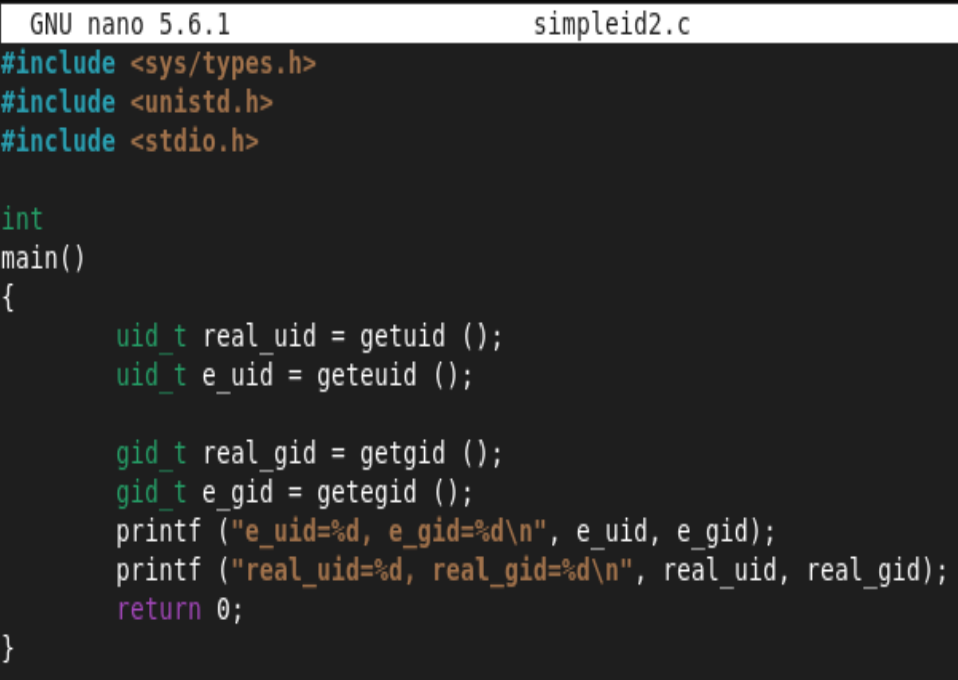
1. Скомплилировала программу simpleid, выполнила её и сравнила результат с выводом команды id ([рис. 3](../../image/3.png)).



Установка расширенного атрибута от имени обычного пользователя

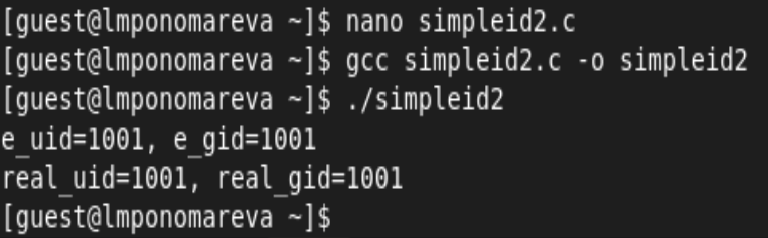
Видим совпадение id пользователя и группы.

1. Усложнила программу, добавив вывод действительных идентификаторов ([рис. 4](../../image/4.png)).



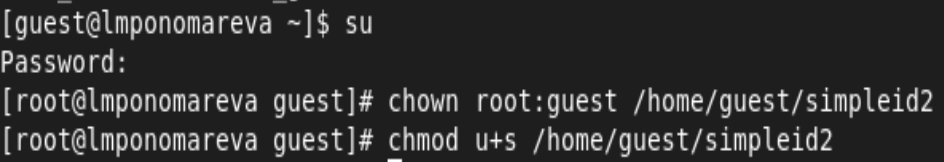
Программа simpleid2.c

1. Скомпилировала и запустила программу simpleid2.c ([рис. 5](../../image/5.png)).



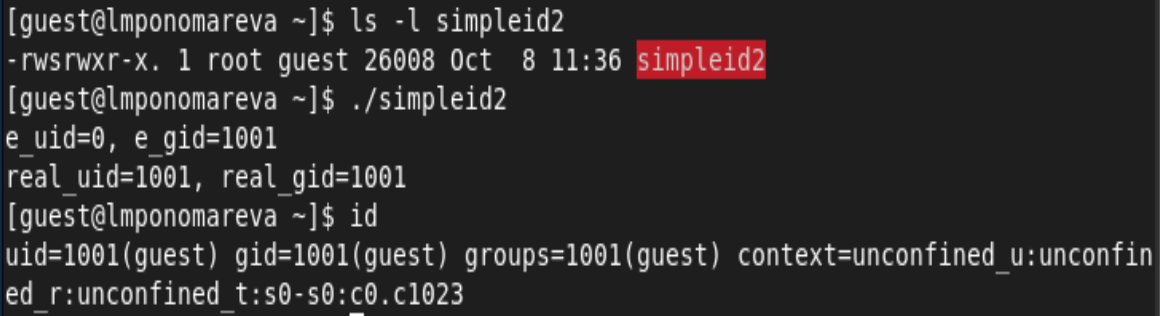
Вывод реальных и действительных идентификаторов

1. От имени суперпользователя выполнила команды смены владельца и группы файла simpleid2 и его прав доступа ([рис. 6](../../image/6.png)).



Смена владельца файла и прав доступа

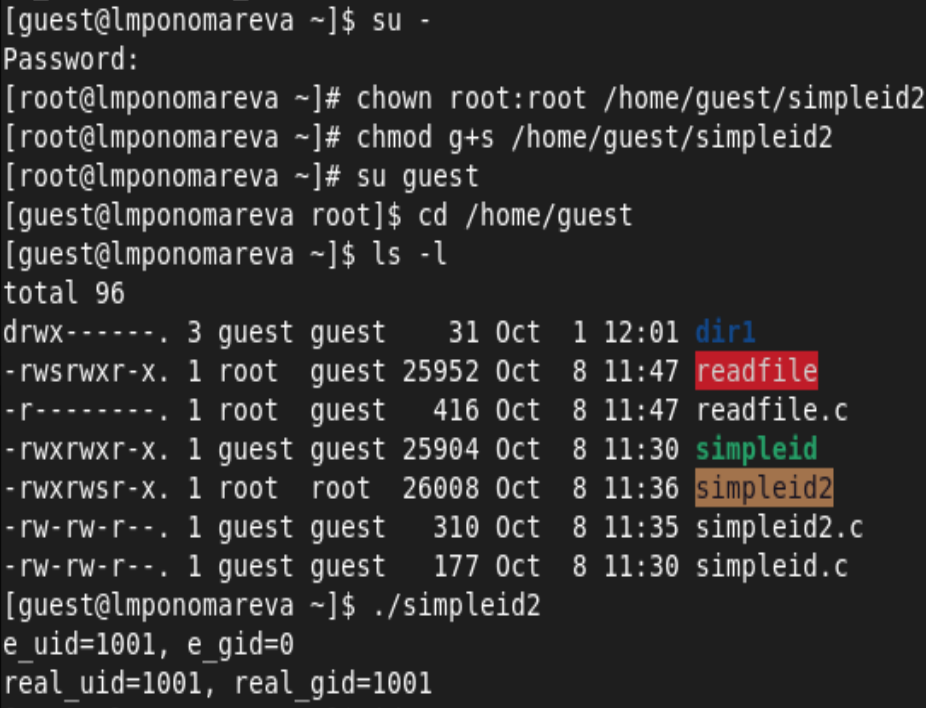
1. Проверила правильность установки новых атрибутов и смены владельца файла simpleid2, запустила программу и сравнила её вывод с результатом ввода команды id: ([рис. 7](../../image/7.png)).



Вывод программы simpleid2

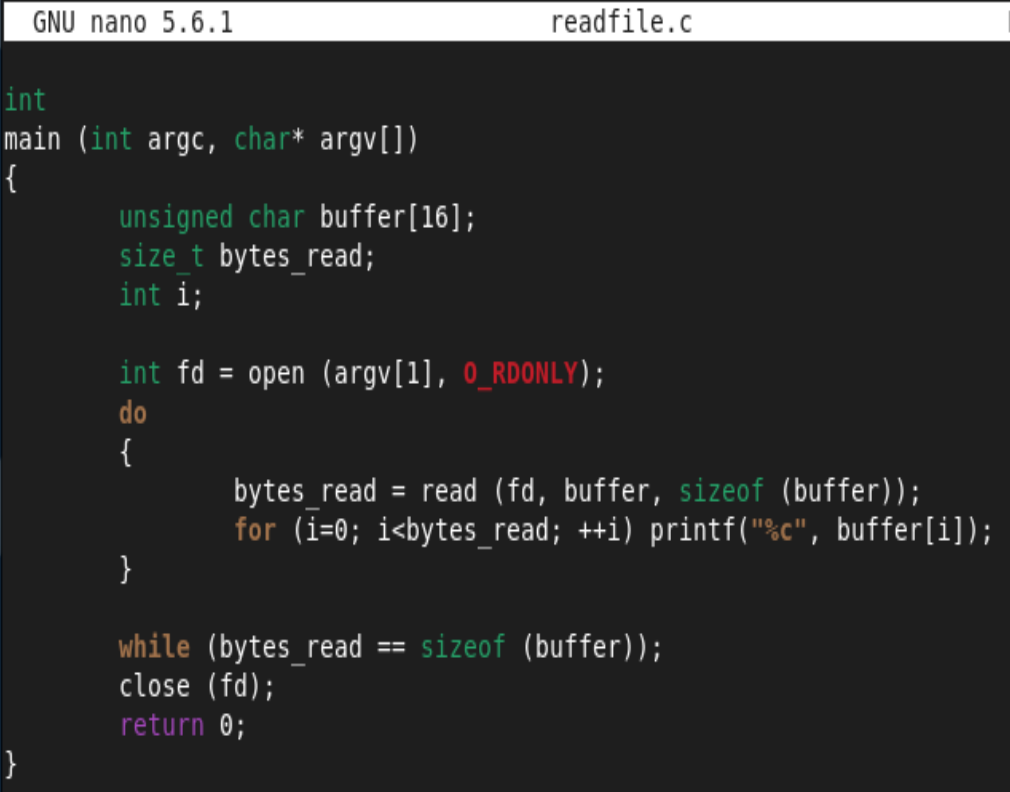
Видим, что действительный идентификатор владельца файла сменился на 0 (root), а реальный не изменился.

1. Проделала тоже самое относительно SetGID-бита. ([рис. 8](../../image/26.png)).



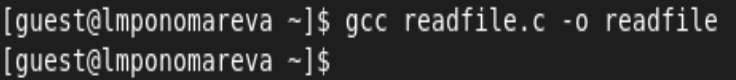
SetGID-бит

1. Создала программу readfile.c ([рис. 9](../../image/8.png)).



Код программы readfile.c

1. Откомпилировала её ([рис. 10](../../image/9.png)).



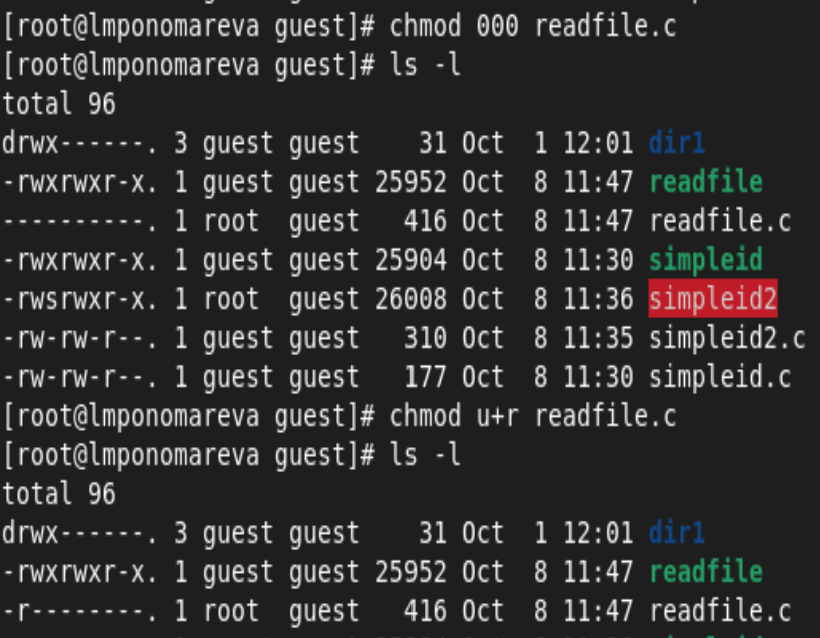
Попытка изменения прав доступа файла

1. Сменила владельца у файла readfile.c ([рис. 11](../../image/10.png)).



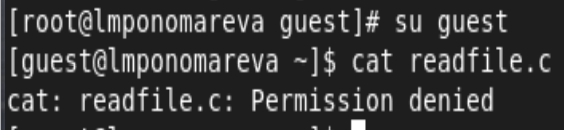
Смена владельца файла

1. Изменила права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог ([рис. 12](../../image/11.png)).



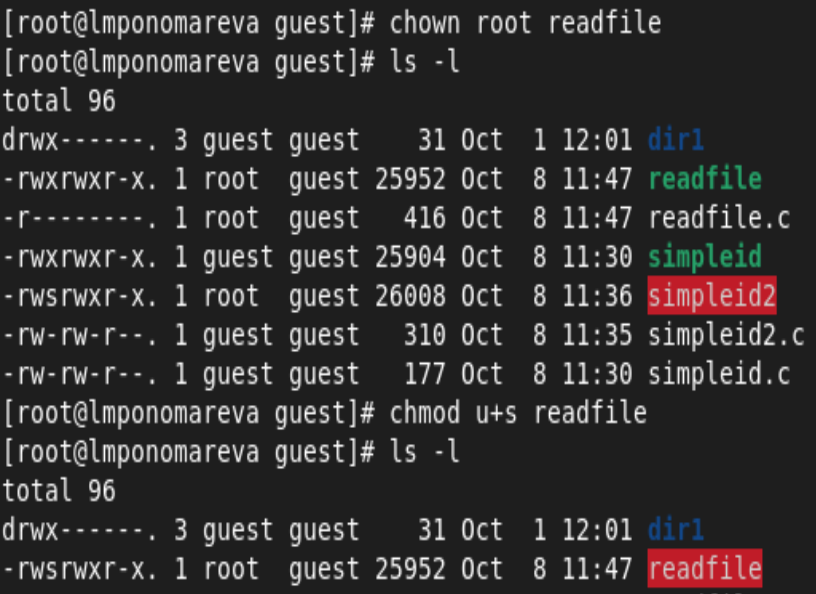
Смена атрибутов

1. Проверила, что пользователь guest не может прочитать файл readfile.c ([рис. 13](../../image/12.png)).



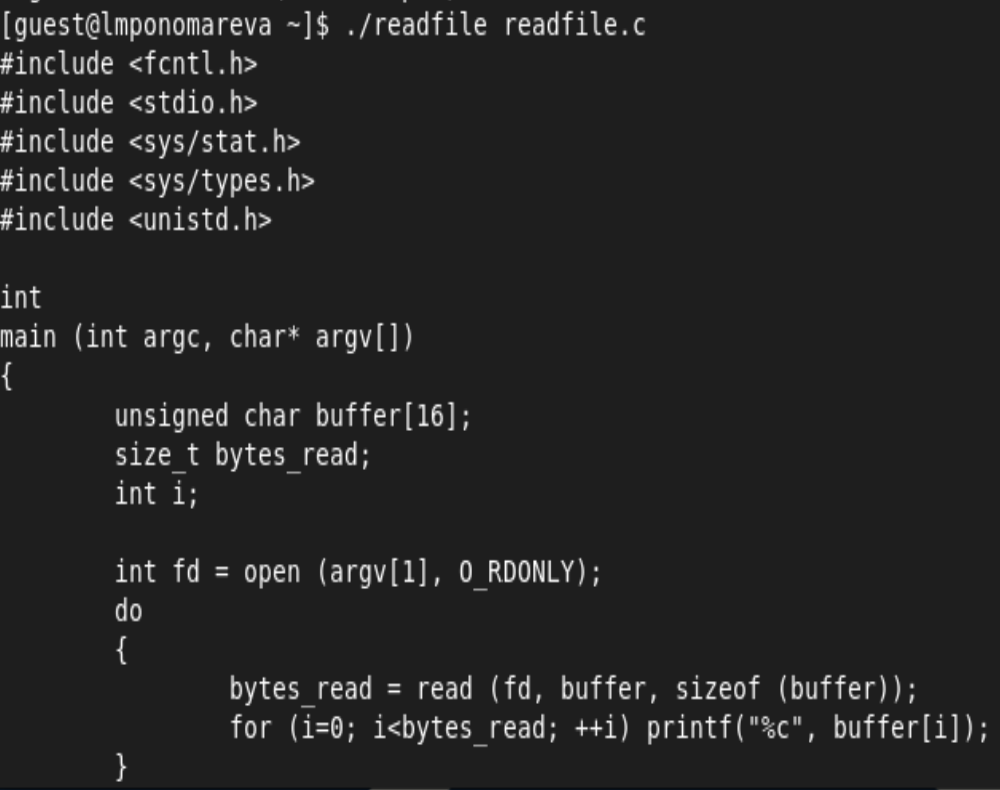
Проверка доступа к файлу

1. Сменила у программы readfile владельца и установила SetUID-бит ([рис. 14](../../image/13.png)).



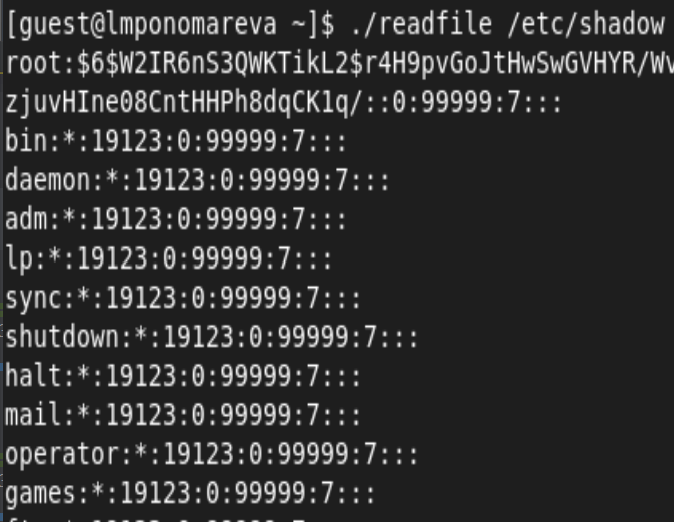
Смена владельца и добавление SetUID у readfile

1. Проверила, может ли программа readfile прочитать файл readfile.c ([рис. 15](../../image/14.png)).



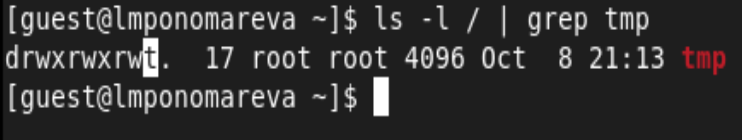
Чтение файла readfile.c программой readfile

1. Проверила, может ли программа readfile прочитать файл /etc/shadow ([рис. 16](../../image/15.png)).



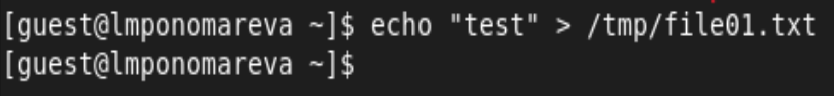
Чтение файла /etc/shadow программой из readfile

1. Выяснила, установлен ли атрибут Sticky на директории /tmp ([рис. 17](../../image/16.png)).



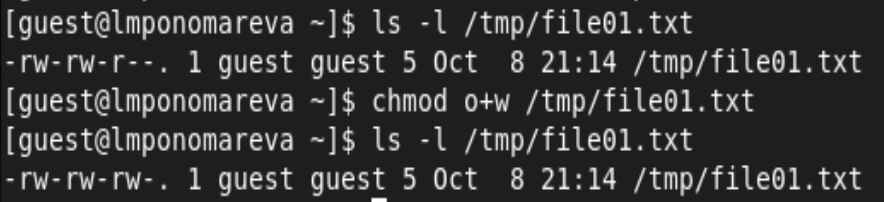
Атрибут Sticky на директории /tmp

1. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test ([рис. 18](../../image/17.png)).



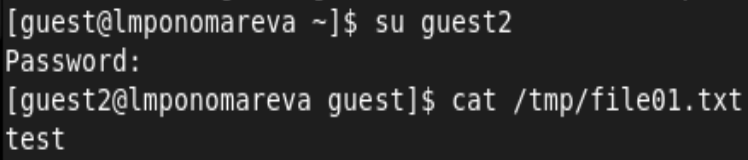
Создание файла file01.txt в директории /tmp

1. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» ([рис. 19](../../image/18.png)).



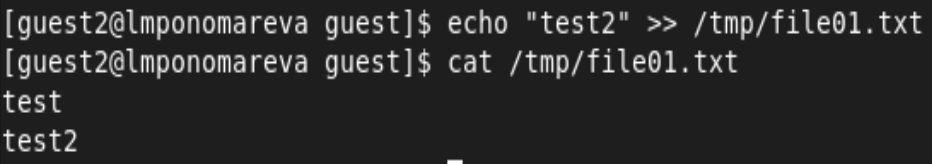
Чтение и запись для категории пользователей «все остальные»

1. От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt ([рис. 20](../../image/19.png)).



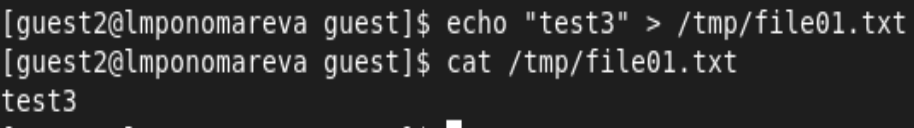
Чтение файла /tmp/file01.txt

1. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2 ([рис. 21](../../image/27.png)).



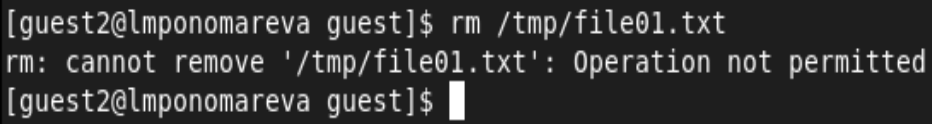
Дозапись файла /tmp/file01.txt

1. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию ([рис. 22](../../image/20.png)).



Перезапись файла /tmp/file01.txt

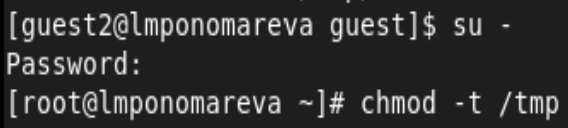
1. От пользователя guest2 попробовала удалить файл /tmp/file01.txt ([рис. 23](../../image/21.png)).



Попытка удаления файла /tmp/file01.txt

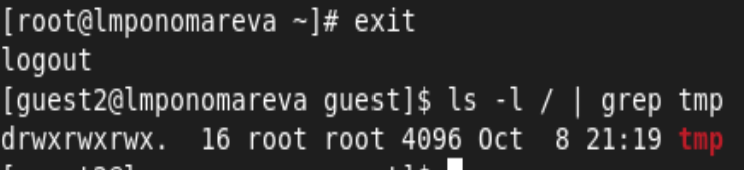
Файл удалить не удалось.

1. Повысила свои права до суперпользователя следующей командой и сняла атрибут t (Sticky-бит) с директории /tmp ([рис. 24](../../image/22.png)).



Снятие Sticky-бит

1. От пользователя guest2 проверила, что атрибута t у директории /tmp нет ([рис. 25](../../image/23.png)).



Проверка атрибутов

1. Повторила предыдущие шаги ([рис. 26](../../image/24.png)).



Проверка атрибутов

Теперь можем удалять файлы находящиеся в каталоге tmp.

# Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов.

# Список литературы

1. [Основы безопасности информационных систем : Учеб. пособие для студентов вузов, обучающихся по специальностям “Компьютер. безопасность” и “Комплекс. обеспечение информ. безопасности автоматизир. систем” / Д.А. Зегжда, А.М. Ивашко. - М. : Горячая линия - Телеком, 2000. - 449, [2] с. : ил., табл.; 21 см.; ISBN 5-93517-018-3](https://search.rsl.ru/ru/record/01000682756).