Отчет по лабораторной работе №4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Евсеева Дарья Олеговна

30 сентября, 2022

Содержание

# Цель работы

Целью данной работы является получение практических навыков работы в консоли с расширенными атрибутами файлов.

# Задание

Провести работу с двумя расширенными атрибутами от имени пользователя guest и администратора.

# Теоретическое введение

Дискреционное разграничение доступа — подход к разграничению доступа, предполагающий назначение владельцев объектов, которые по собственному усмотрению определяют права доступа субъектов (других пользователей) к объектам (файлам), которыми владеют.

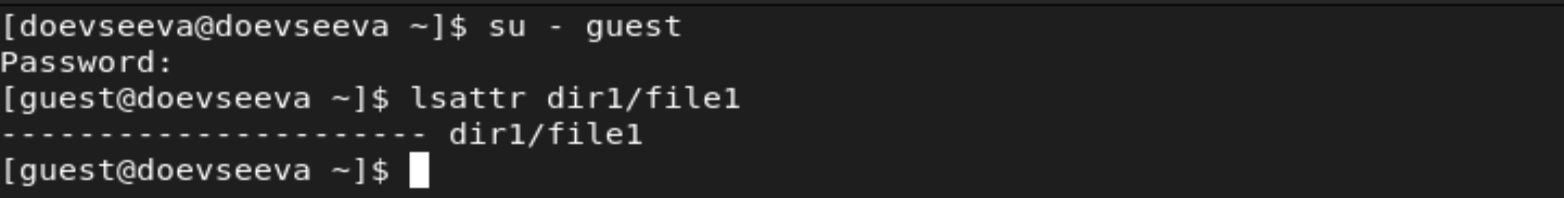
Дискреционные механизмы разграничения доступа используются для разграничения прав доступа процессов как обычных пользователей, так и для ограничения прав системных программ (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей.

# Выполнение лабораторной работы

Выполнять работу будем в операционной системе, установленной при выполнении первой лабораторной работы.

## 1. Работа с расширенным атрибутом a

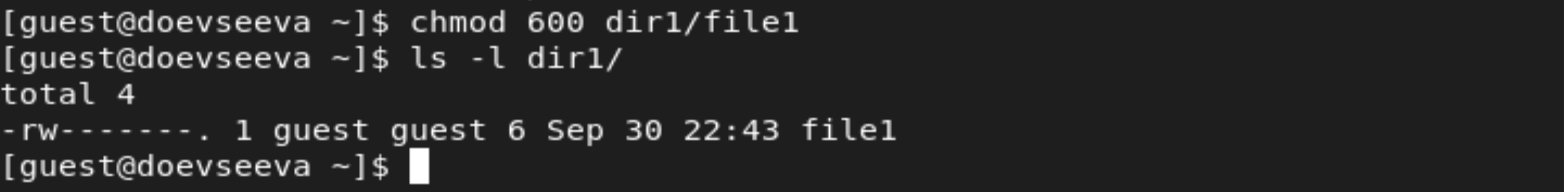
Для начала войдем в систему от имени пользователя guest и определим расширенные атрибуты файла file1.



Определение расширенных атрибутов

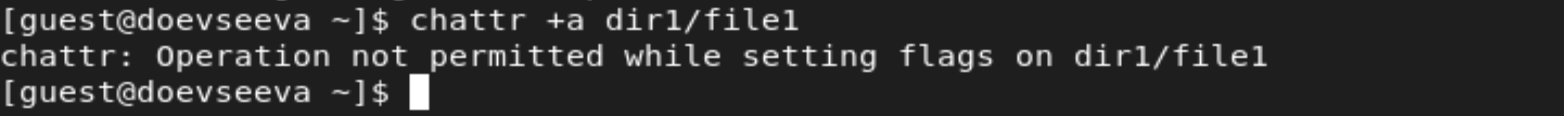
Как мы можем видеть, изначально у файла нет никаких расширенных атрибутов.

Далее установим на файл права, разрешающие чтение и запись для владельца файла.



Установка прав на чтение и запись

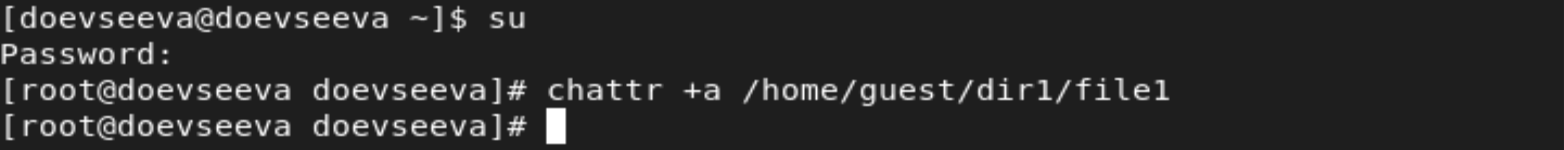
Попробуем установить на файл расширенный атрибут a от имени пользователя guest.



Попытка установки атрибута a

Мы видим, что в выполнении операции было отказано.

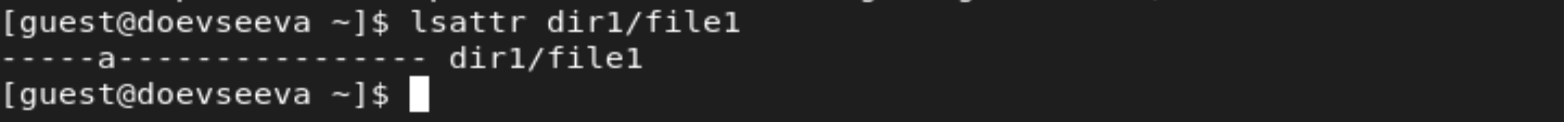
Далее, открыв новую консоль, зайдем в систему с правами администратора и снова попробуем установить на файл расширенный атрибут a.



Установка атрибута a

Здесь мы видим, что отказа в выполнении не последовало.

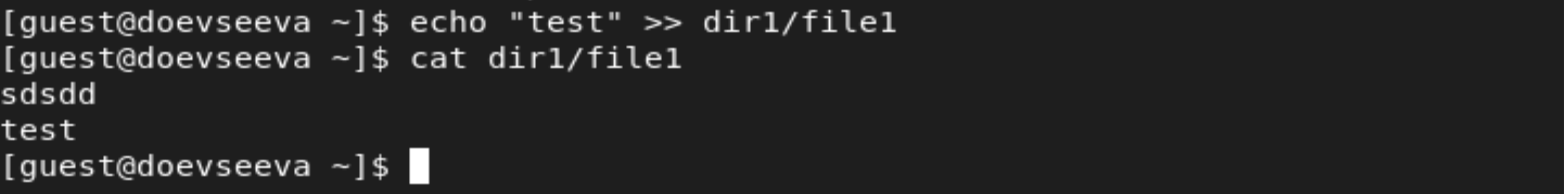
Проверим правильность установки атрибута от имени пользователя guest.



Проверка атрибутов

Мы видим, что атрибут действительно был успешно установлен.

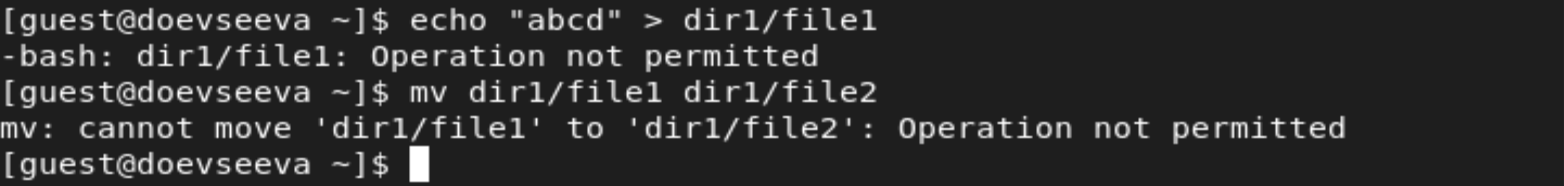
Выполним дозапись в файл слова ‘test’ и проверим успешность выполнения операции.



Дозапись в файл и его чтение

Как мы видим, слово было успешно записано в файл.

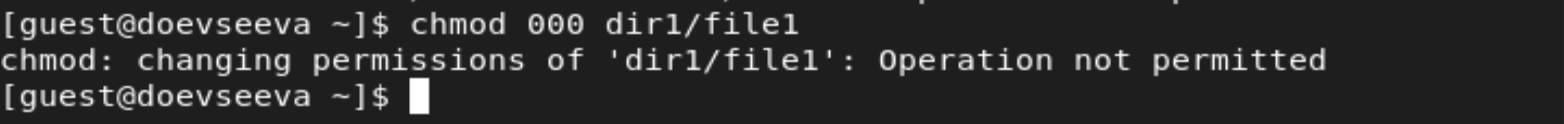
Далее попробуем перезаписать имеющуюся в файле информацию, а также попробуем переименовать файл.



Попытка перезаписи и переименования файла

Для обеих операций было отказано в доступе.

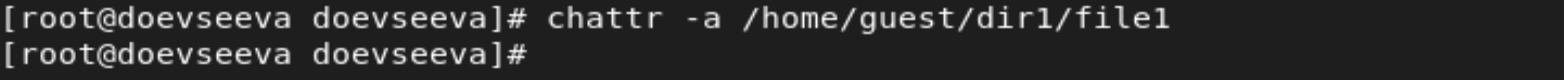
Также попробуем установить на файл права, запрещающие чтение и запись для владельца файла.



Попытка изменения прав на файл

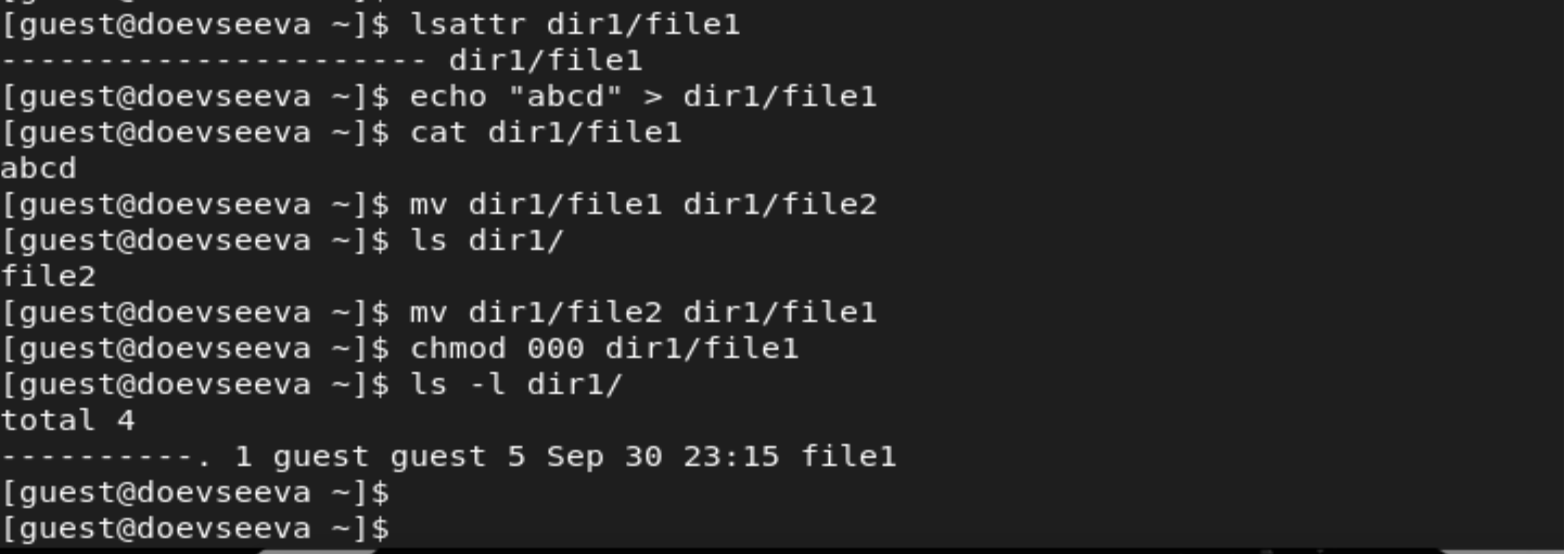
В доступе снова было отказано.

Снимем с файла расширенный атрибут a от имени администратора.



Снятие атрибута a

Далее повторим операции, которые ранее не смогли выполнить.



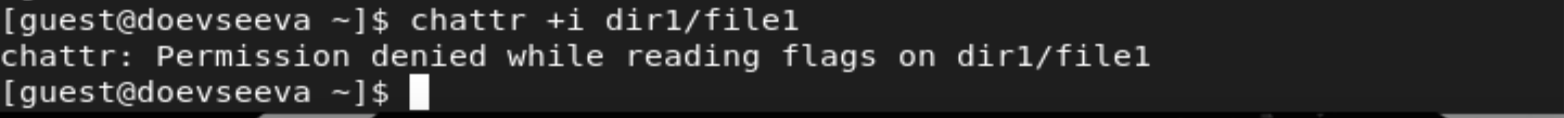
Повторное выполнение операций

Как мы видим, теперь все операции были успешно выполнены.

## 2. Работа с расширенным атрибутом i

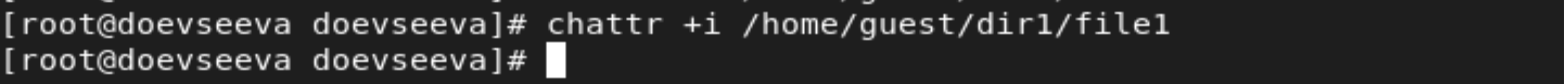
Теперь проделаем те же действия для расширенного атрибута i.

Попробуем добавить атрибут от имени пользователя guest.

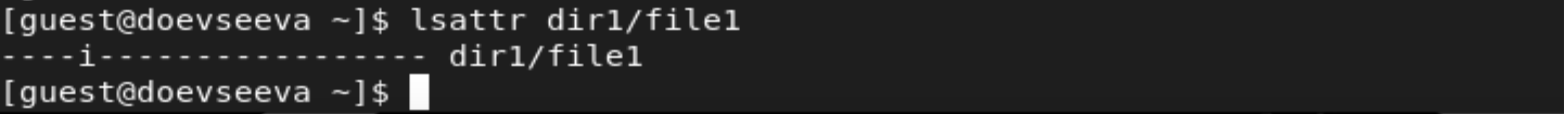


Попытка установки атрибута i

Добавим атрибут от имени администратора и проверим его наличие от имени пользователя guest.

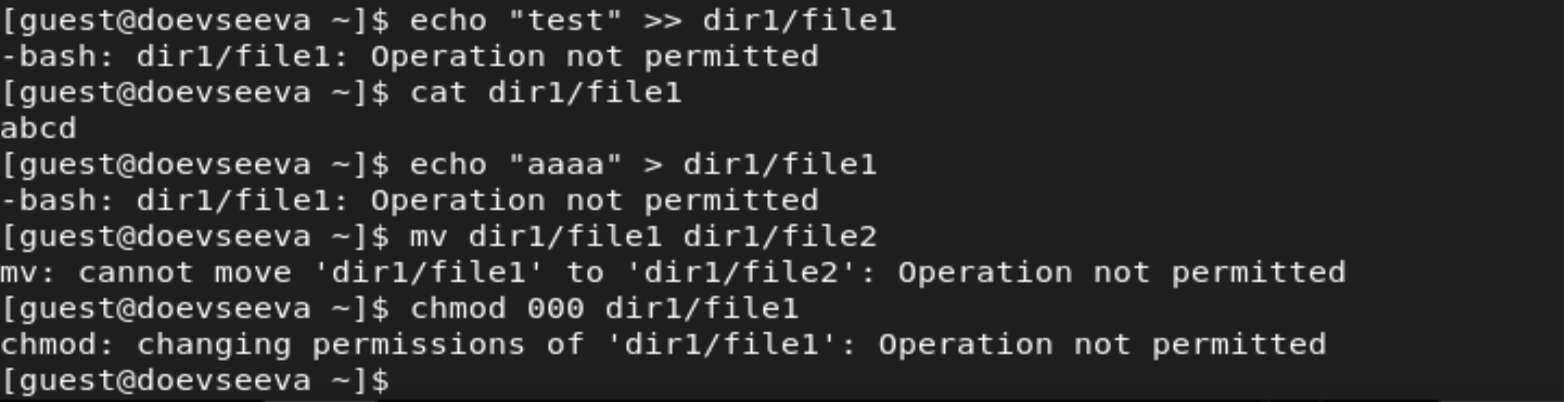


Установка атрибута i



Проверка атрибутов

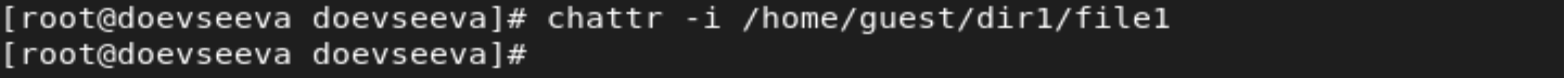
Попробуем повторить выполнение различных команд.



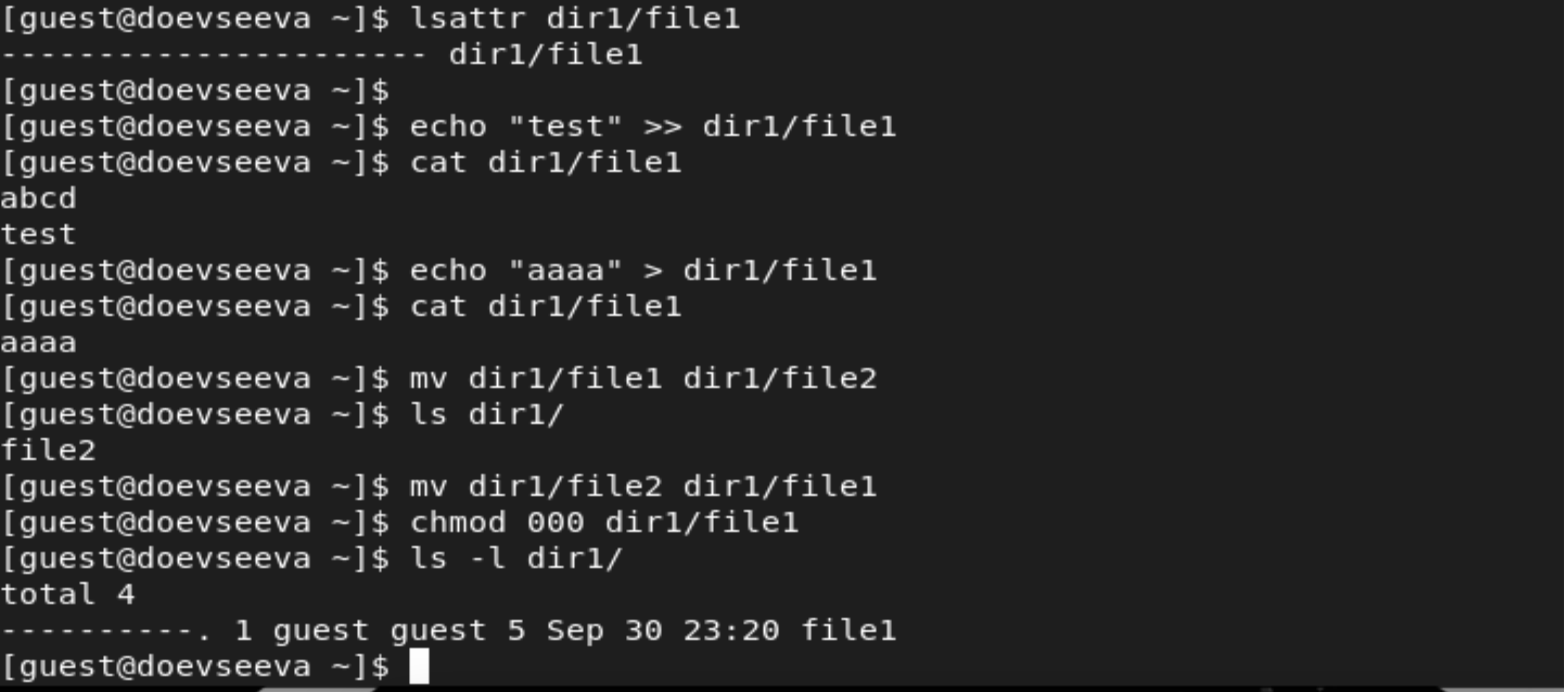
Попытка выполнения различных команд

Как мы видим, в данном случае в доступе было отказано для всех операций, кроме чтения файла, и, в отличие от расширенного атрибута a, дозапись в файл осуществить уже не удалось.

Снимем с файла расширенный атрибут i и повторим операции, которые не удалось выполнить.



Снятие атрибута i



Повторное выполнение операций

Как мы видим, теперь все операции были успешно выполнены.

# Выводы

В результате проделанной работы мы приобрели практические навыки работы в консоли с расширенными атрибутами файлов.

# Список литературы

* Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН” https://esystem.rudn.ru/