Лабораторная работа № 2

Дискреционное разграничение прав в Linux. Основные атрибуты

Сухарев Кирилл

Содержание

# Цель работы

Получение практических навыков работы в консоли с атрибутами файлов, закрепление теоретических основ дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux.

# Условные обозначения и термины

**Учетная запись** - хранимая в компьютерной системе совокупность данных о пользователе, необходимая для его опознавания (аутентификации) и предоставления доступа к его личным данным и настройкам.

**Директория** - объект в файловой системе, упрощающий организацию файлов.

**uid** - номер, назначенный каждому пользователю Linux. Это представление пользователя в ядре Linux.

**gid** - идентификационный номер основной группы пользователя.

# Теоретические вводные данные

## Права доступа к файлам в Linux

В операционной системе Linux много функций безопасности. Одна из самых важных - это система прав доступа к файлам. Linux, как последователь идеологии ядра Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в linux продуманы очень хорошо.

## Основные права доступа к файлам в Linux

Изначально каждый файл имел три параметра доступа. Вот они:

* Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;
* Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;
* Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Но все эти права были бы бессмысленными, если бы применялись сразу для всех пользователей. Поэтому каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

* Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение.
* Группа - любая группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу.
* Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла.

Именно с помощью этих наборов полномочий устанавливаются права файлов в linux. Каждый пользователь может получить полный доступ только к файлам, владельцем которых он является или к тем, доступ к которым ему разрешен. Только пользователь Root может работать со всеми файлами независимо от их набора их полномочий.

## Просмотр прав доступа

Узнать права на файл linux можно командой **ls -l**. За права файлов в linux тут отвечают черточки. Первая это тип файла, который рассмотрен в отдельной статье. Дальше же идут группы прав сначала для владельца, для группы и для всех остальных. Всего девять черточек на права и одна на тип. Рассмотрим значения черточек:

* — - нет прав, совсем;
* –x - разрешено только выполнение файла, как программы но не изменение и не чтение;
* -w- - разрешена только запись и изменение файла;
* -wx - разрешено изменение и выполнение, но в случае с каталогом, вы не можете посмотреть его содержимое;
* r– - права только на чтение;
* r-x - только чтение и выполнение, без права на запись;
* rw- - права на чтение и запись, но без выполнения;
* rwx - все права;

# Техническое оснащение и выбранные методы проведения работы

В качестве среды выполнения лабораторной работы используется менеджер виртуальных машин VirtualBox и установленная с его помощью OC Centos 7 на базе Linux.

# Выполнение работы

1. Используя учётную запись администратора создадим учётную запись пользователя guest и зададим для него пароль (fig. 1).

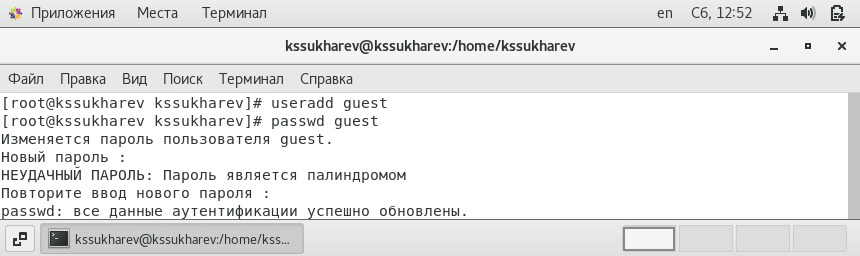


Figure 1: Создание нового учётного пользователя

1. Войдем в систему под пользователям guest (fig. 2).

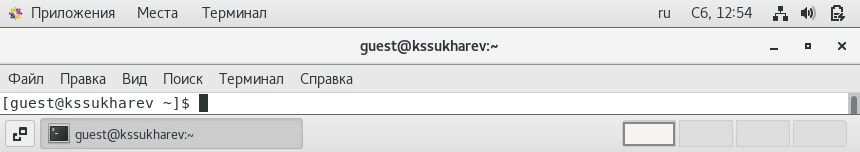


Figure 2: Вход в систему

1. Определим текущую директорию командой **pwd** и сравним ее с домашней директорией, которая выводится командой **echo ~**. Текущая директория является домашней (fig. 3).

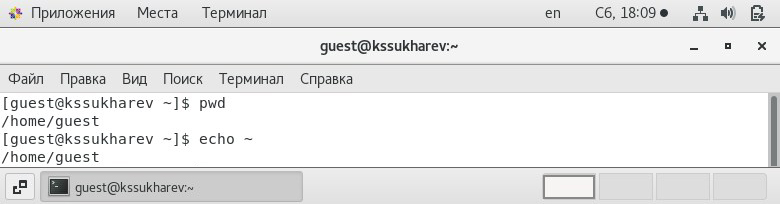


Figure 3: Определение текущей директории

1. Командой **whoami** уточним имя пользователя. Команда вывела guest (fig. 4).

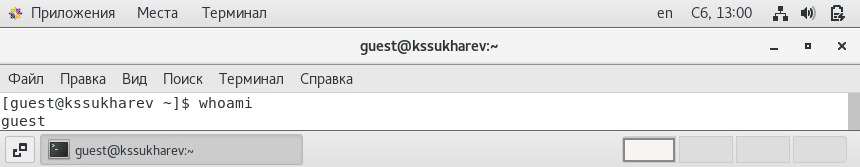


Figure 4: Определение имени пользователя

1. Теперь узнаем более подробные сведения о пользователе командой **id**. uid - 1001, gid - 1001. Сравним вывод с выводом команды **groups**. Команда вывела список всех групп, но поскольку новых групп не создавалось, команда вывела только guest. Вывод команды **id** полностью соответствует приглашению командной строки (fig. 5).

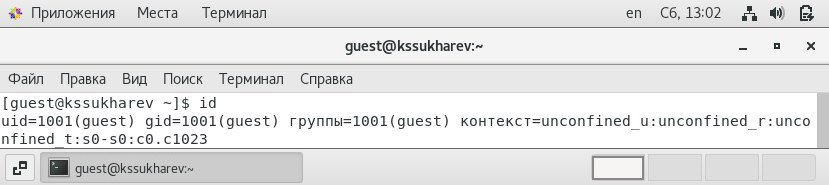


Figure 5: Команды **id** и **groups**

1. Просмотрим сведения о пользователе guest в файле */etc/passwd* командой **cat /etc/passwd | grep guest**. Выведенные uid и gid совпадают с определенными ранее (fig. 6).

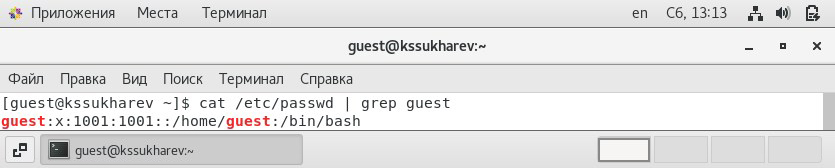


Figure 6: Файл */etc/passwd*

1. Определим существующие в системе директории командой **ls -R /home/**. Доступ к поддиректориям kssukharev получить не удалось. Выведем расширенную информацию о директориях командой **ls -l /home/**. Видим, что у владельцев есть полный доступ к каталогам. У остальных же пользователей доступа нет совсем (fig. 7).

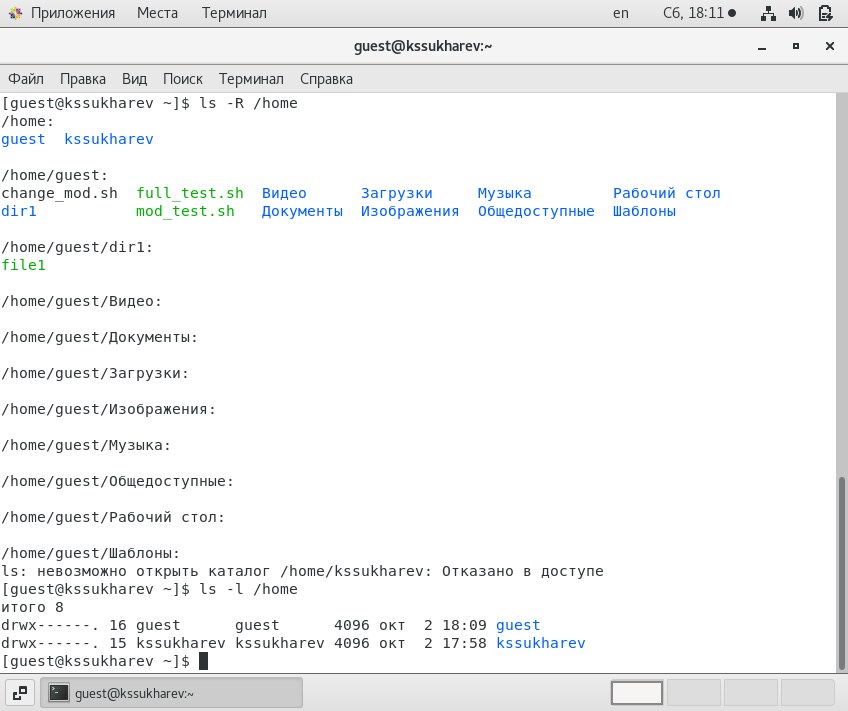


Figure 7: Права на директориях системы

1. Проверим атрибуты на поддиректориях в директории */home* командой **lsattr /home**. Нам удалось увидеть атрибуты лишь на директории guest (fig. 8).

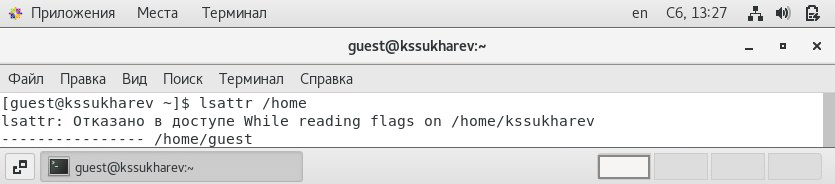


Figure 8: Атрибуты поддиректорий

1. Создадим в домашней директории поддиректорию dir1 командой **mkdir dir1**. Снова применим команды **ls -l** и **lsattr** чтобы увидеть, какие права доступа и расширенные атрибуты были выставлены на директорию dir1. Права доступа предоставлены пользователю guest и группе guest в полном размере, а остальным пользователям - только на чтение и выполнение. Каких-либо атрибутов на этой директории нет (fig. 9).

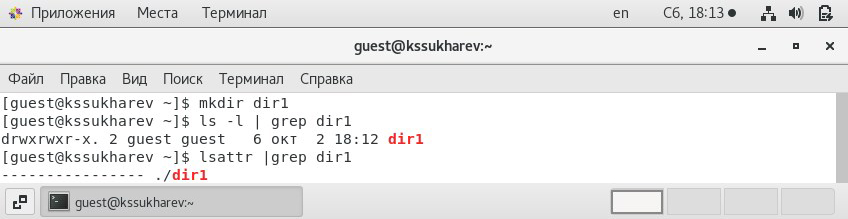


Figure 9: Новая директория

1. Снимем с созданной директории все права командой **chmod 000 dir1**. Командой **ls -l** убедимся в правильноси выполненных действий (fig. 10).

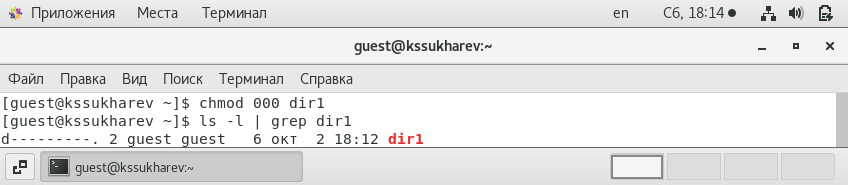


Figure 10: Снятие атрибутов

1. Попытаемся созадать в директории dir1 файл file1 командой **echo “test” > /home/guest/dir1/file1**. Получаем закономерную ошибку, поскольку мы запретили любые взаимодействия с директорией (fig. 11).

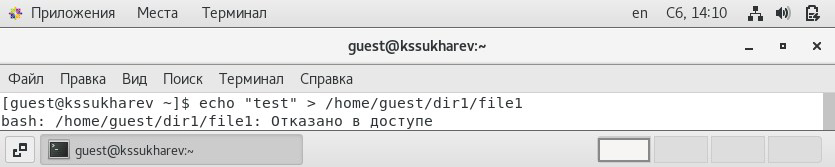


Figure 11: Попытка создания файла

1. Заполним таблицу «Установленные права и разрешённые действия» (fig. 12). До заполнения таблицы создадим в dir1 файл file1. При заполнении таблицы будем использовать следующие команды:

* Создание файла: **echo “test” > dir1/file2**
* Удаление файла: **rm dir1/file1**
* Запись в файл: **echo “test” > dir1/file1**
* Чтение файла: **cat dir1/file1**
* Смена директории: **cd dir1**
* Просмотр файлов в директории: **ls dir1**
* Переименование файла: **mv dir1/file1 dir1/file2**
* Смена атрибутов файла: **chattr +d dir1/file1**

Процесс заполнения таблицы подробно описан в скринкасте к данной лабораторной работе.

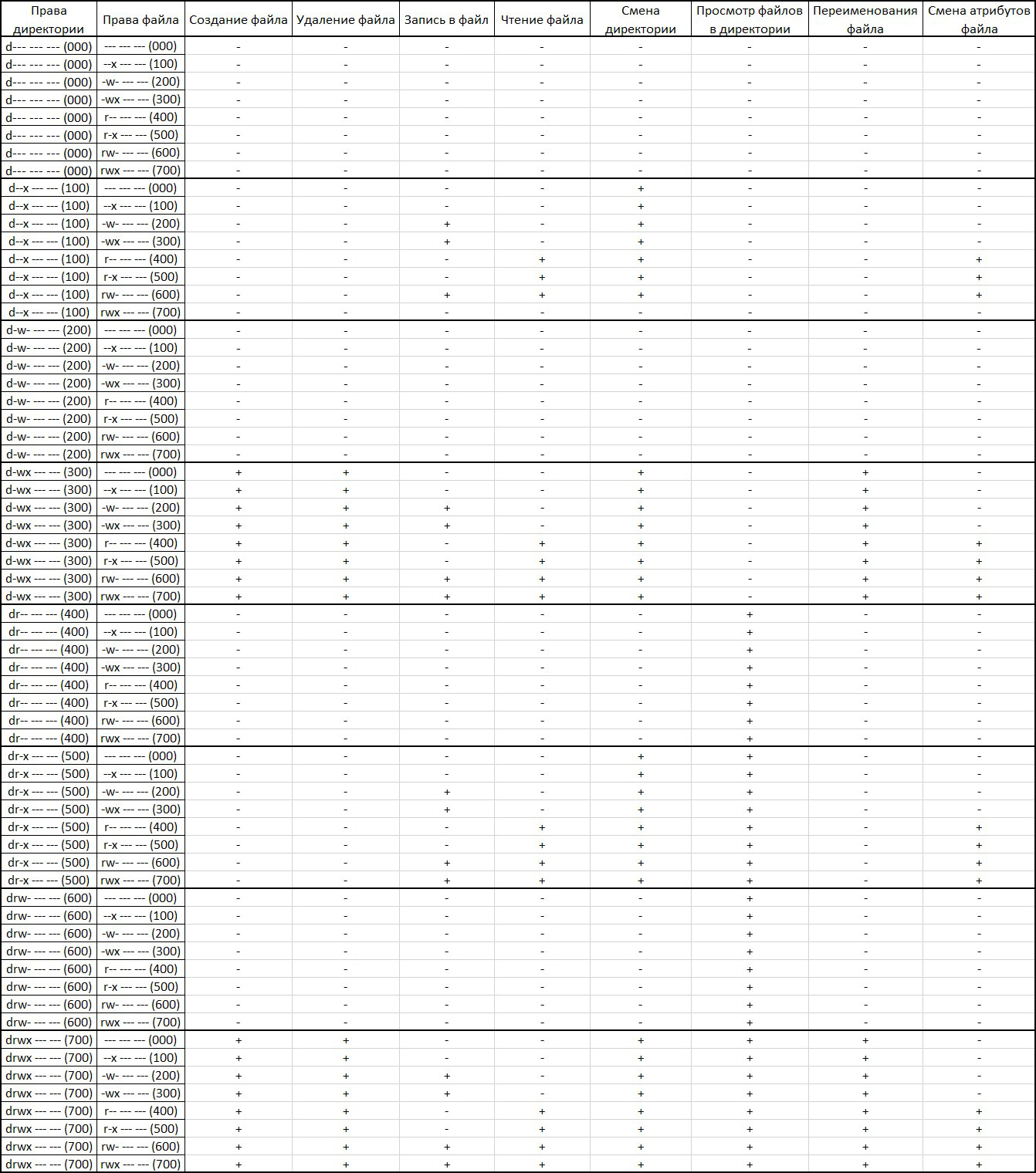


Figure 12: Таблица «Установленные права и разрешённые действия»

1. По результатам таблицы «Установленные права и разрешённые действия» заполним таблицу «Минимальные права для совершения операций» (fig. 13).



Figure 13: Таблица «Минимальные права для совершения операций»

# Выводы

Права доступа используются для управления возможностями различных групп пользователей системы по отношению к директориям и файлам.