Лабораторная работа №4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Саргсян Арам Грачьяевич

Содержание

Список иллюстраций

1 Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов.

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Теоретическое введение

Есть 3 вида разрешений. Они определяют права пользователя на 3 действия: чтение, запись и выполнение. В Linux эти действия обозначаются вот так:

- **r** — read (чтение) — право просматривать содержимое файла;
- **w** — write (запись) — право изменять содержимое файла;
- **x** — execute (выполнение) — право запускать файл, если это программа или скрипт.

У каждого файла есть 3 группы пользователей, для которых можно устанавливать права доступа.

- **owner** (владелец) — отдельный человек, который владеет файлом. Обычно это тот, кто создал файл, но владельцем можно сделать и кого-то другого.
- **group** (группа) — пользователи с общими заданными правами.
- **others** (другие) — все остальные пользователи, не относящиеся к группе и не являющиеся владельцами.[1]

Чтобы увидеть текущие назначения владельца, нужно использовать команду `ls -l`. Эта команда показывает пользователя и группу-владельца.

Чтобы применить соответствующие разрешения, первое, что нужно учитывать, это владение. Для этого есть команда `chown`.[2]

Для того, чтобы позволить обычным пользователям выполнять программы от имени суперпользователя без знания его пароля была придумана такая вещь, как SUID и SGID биты. Рассмотрим эти полномочия подробнее.

- **SUID** - если этот бит установлен, то при выполнении программы, id пользователя, от которого она запущена заменяется на id владельца файла. Фактически, это позволяет обычным пользователям запускать программы от имени суперпользователя;
- **SGID** - этот флаг работает аналогичным образом, только разница в том, что пользователь считается членом группы, с которой связан файл, а не групп, к которым он действительно принадлежит. Если SGID флаг установлен на каталог, все файлы, созданные в нем, будут связаны с группой каталога, а не пользователя. Такое поведение используется для организации общих папок;
- **Sticky-bit** - этот бит тоже используется для создания общих папок. Если он установлен, то пользователи могут только создавать, читать и выполнять файлы, но не могут удалять файлы, принадлежащие другим пользователям.[3]

4 Выполнение лабораторной работы

1. От имени пользователя guest определите расширенные атрибуты файла file1, на файл file1 установил права, разрешающие чтение и запись для владельца файла, попробовал установить расширенный атрибут a (рис. 4.1).

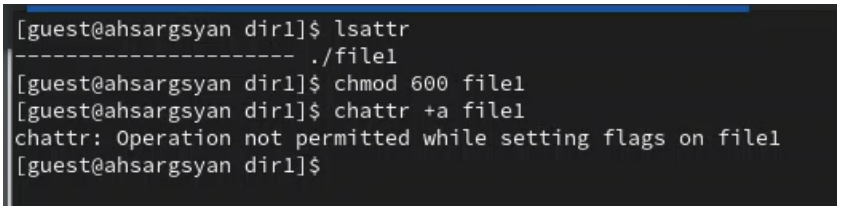

Рис. 4.1: Попытка установки прав

2. Установил права от суперпользователя, добавил информацию в file1 (рис. 4.2).

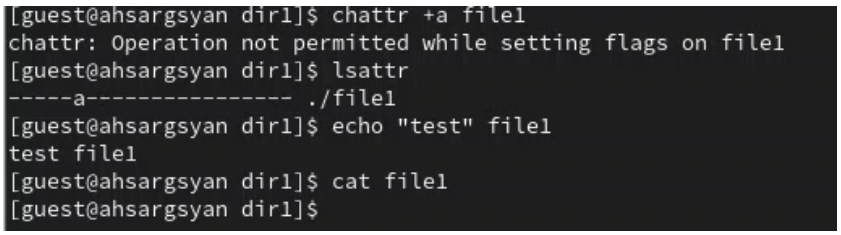

Рис. 4.2: Установка прав

3. Изучил атрибут i (рис. 4.3).

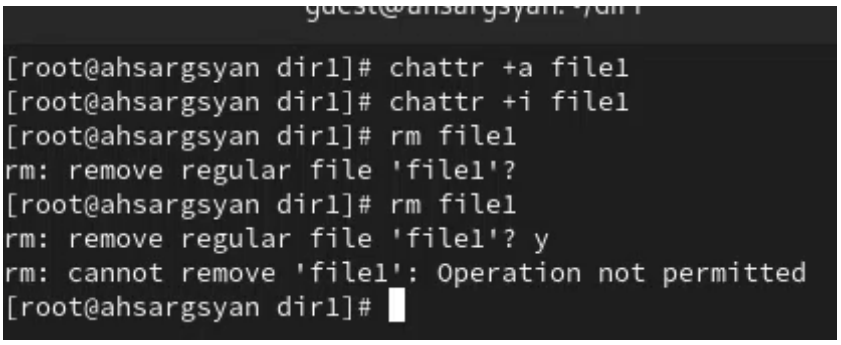

Рис. 4.3: Атрибут i

5 Выводы

Мы получили практические навыки работы в консоли с расширенными атрибутами файлов.

Список литературы

1. Права доступа в Linux [Электронный ресурс]. 2023. URL: https://codechick.io/tutorials/unix-linux/unix-linux-permissions.
2. Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask) [Электронный ресурс]. 2023. URL: https://habr.com/ru/articles/469667/.
3. Права доступа к файлам в Linux [Электронный ресурс]. 2023. URL: https://losst.pro/prava-dostupa-k-fajlam-v-linux.