

Лабораторная работа № 4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Абдуллаев Сайидазизхон Шухратович

Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Ход работы	9
Выводы	16

Список иллюстраций

1	Определение расширенных атрибутов	9
2	Установление прав на файл	10
3	Установление расширенного атрибута	10
4	Установление расширенного атрибута от имени суперпользователя	11
5	Дозапись в файл	12
6	Манипуляции с файлом	13
7	Изменение прав доступа	13
8	Снятие атрибута а	14
9	Выполнение действий	14
10	Замена атрибута «а» на «і»	15

Список таблиц

Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов.

Задание

Закрепить дискреционное разграничение прав в Linux с расширенными атрибутами.

Теоретическое введение

В Linux, как и в любой многопользовательской системе, абсолютно естественным образом возникает задача разграничения доступа субъектов — пользователей к объектам — файлам дерева каталогов. Один из подходов к разграничению доступа — так называемый дискреционный — предполагает 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 в (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей. Чтобы получить доступ к файлам в Linux, используются разрешения. Эти разрешения назначаются трем объектам: файлу, группе и другому объекту. Для управления правами используется команда `chmod`. При использовании `chmod` в относительном режиме вы работаете с тремя индикаторами, чтобы указать, что вы хотите сделать. Сначала вы указываете, для кого вы хотите изменить разрешения. Для этого вы можете выбрать между пользователем (u), группой (g) и другими (o). Затем вы используете оператор для добавления или удаления разрешений из текущего режима или устанавливаете их абсолютно. В конце вы используете r(read), w(write) и x(execute), чтобы указать, какие разрешения вы хотите установить. При использовании `chmod` вы можете устанавливать разрешения для пользователя (user), группы (group) и других (other). Помимо основных разрешений, о которых вы только что прочитали, в Linux также есть набор расширенных раз-

решений. Это не те разрешения, которые вы устанавливаете по умолчанию, но иногда они предоставляют полезное дополнение.

Ход работы

1. Проверим наличие file1 директории dir1, войдя в учетную запись пользователя guest. Определим расширенные атрибуты файла. (Рис. [-@fig:001]).

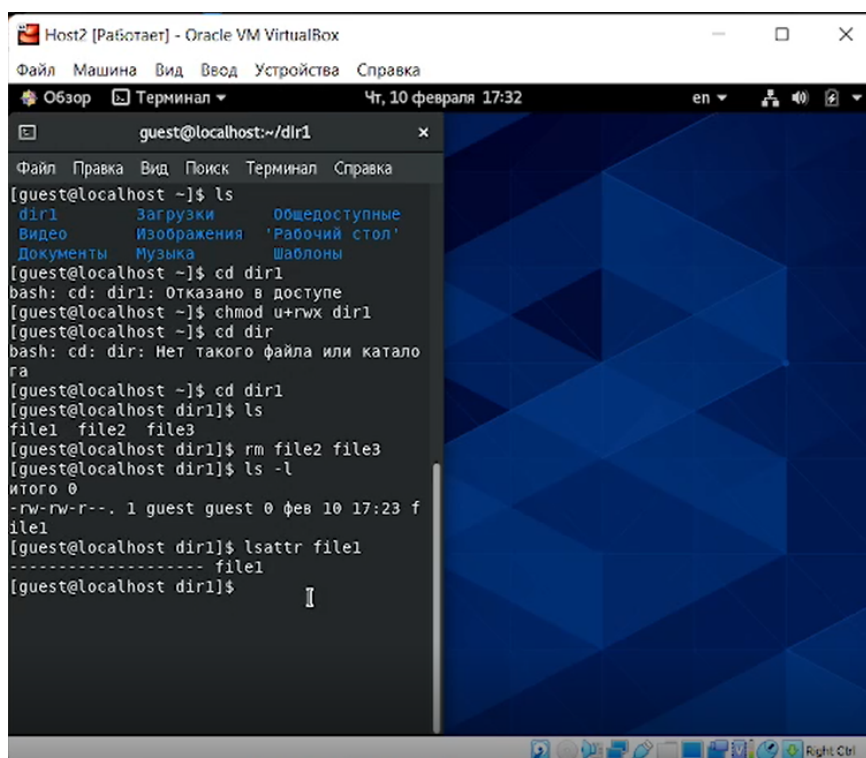


Рис. 1: Определение расширенных атрибутов

2. Установим на file1 права, разрешающие чтение и запись для владельца файла. Проверим правильность выполнения команды. (Рис. [-@fig:002]).

```

[guest@localhost dir1]$ lsattr file1
----- file1
[guest@localhost dir1]$ chmod 600 file1
[guest@localhost dir1]$ ls -l
итого 0
-rw-----. 1 guest guest 0 фев 10 17:23 f
ile1
[guest@localhost dir1]$ █ █

```

Рис. 2: Установление прав на файл

3. Попробуем установить на файл file1 расширенный атрибут а от имени пользователя guest. В результате получаем отказ на выполнение данного действия. (Рис. [-@fig:003]).

```

[guest@localhost dir1]$ chatter +a file1
chattr: Операция не позволена while settin
g flags on file1 █
[guest@localhost dir1]$ █

```

Рис. 3: Установление расширенного атрибута

4. С помощью команды su - заходим на второй консоли в учетную запись root . Попробуем установить расширенный атрибут а на файл /home/guest/dir1/file1 от имени суперпользователя, а затем проверяем правильность выполнения от имени guest. (Рис. [-@fig:004]).

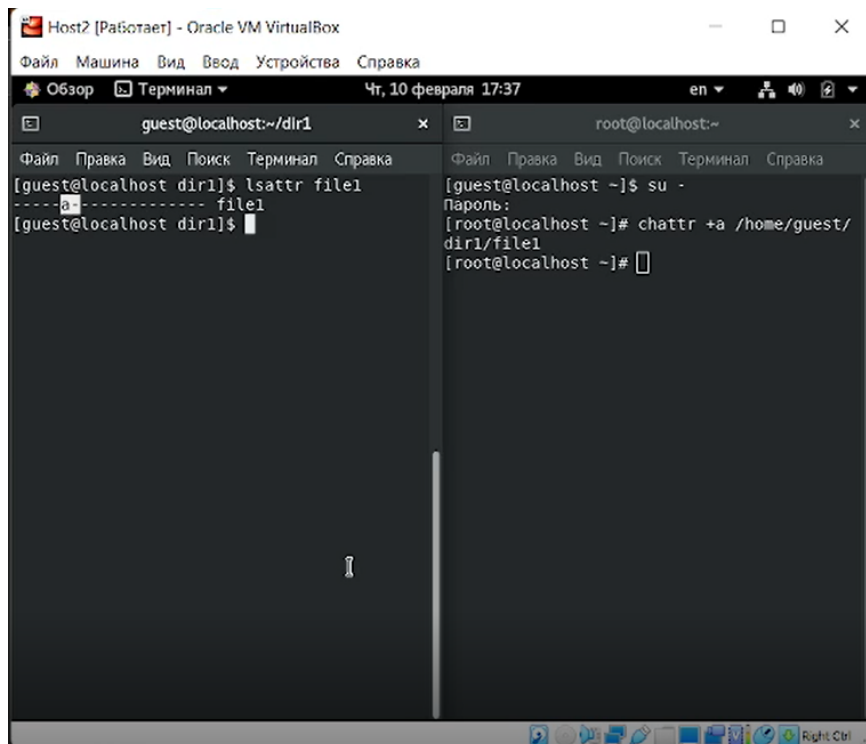
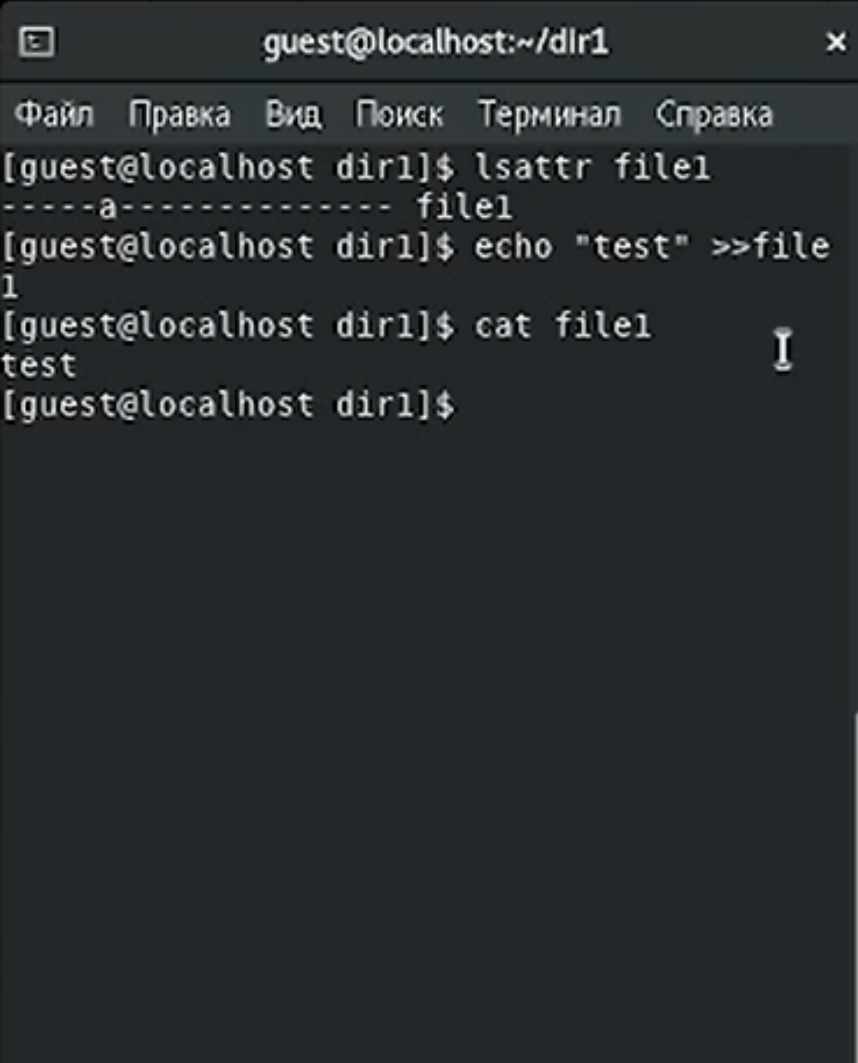


Рис. 4: Установление расширенного атрибута от имени суперпользователя

5. Выполним дозапись в файл file1 слова «test» командой `echo "test" /home/guest/dir1/file1` После этого выполним чтение файла file1 командой `cat`. (Рис. [-@fig:005]).

A screenshot of a terminal window titled "guest@localhost:~/dir1". The window has a menu bar with options: "Файл", "Правка", "Вид", "Поиск", "Терминал", and "Справка". The terminal shows the following commands and output:

```
[guest@localhost dir1]$ lsattr file1
-----a----- file1
[guest@localhost dir1]$ echo "test" >>file1
[guest@localhost dir1]$ cat file1
test
[guest@localhost dir1]$
```

Рис. 5: Дозапись в файл

6. Попробуем стереть имеющуюся в файле информацию командой `echo "abcd" > file1`, а также переименовать файл. (Рис. [-@fig:006]).

```
[guest@localhost dir1]$ echo "abcd" >file1
bash: file1: Операция не позволена
[guest@localhost dir1]$ mv file1 file2
mv: невозможно переместить 'file1' в 'file
2': Операция не позволена
[guest@localhost dir1]$
```

Рис. 6: Манипуляции с файлом

7. Попробуем установить на файл file1 права, запрещающие чтение и запись для владельца файла. В результате получаем отказ. (Рис. [-@fig:007]).

```
[guest@localhost dir1]$ chmod 000 file1
chmod: изменение прав доступа для 'file1':
Операция не позволена
[guest@localhost dir1]$
```

Рис. 7: Изменение прав доступа

8. Снимем расширенный атрибут а с файла /home/guest/dir1/file1 от имени суперпользователя. (Рис. [-@fig:008]).

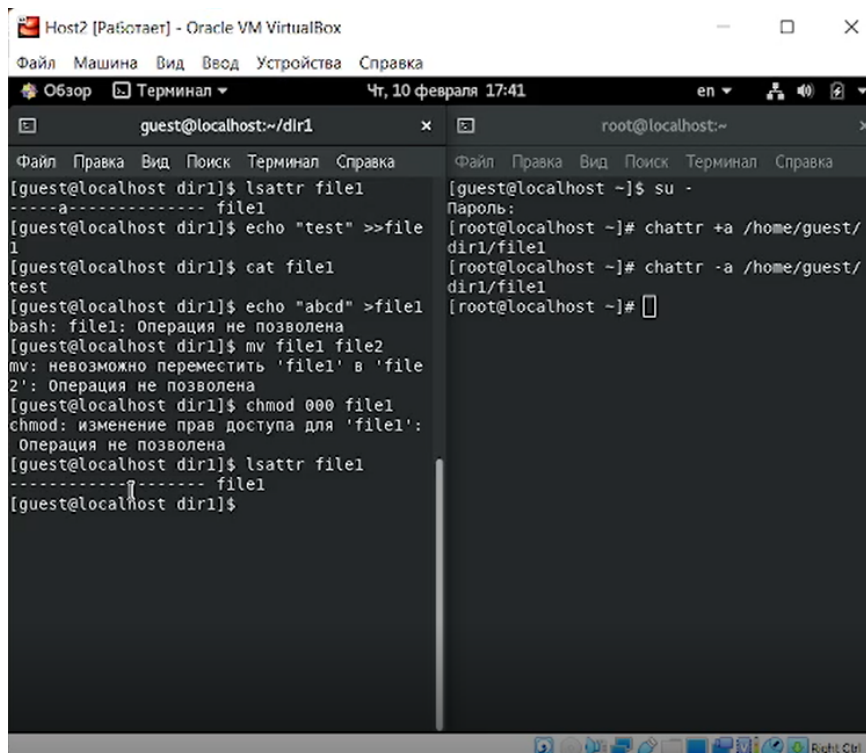


Рис. 8: Снятие атрибута а

9. Повторим не выполненные ранее действия. (Рис. [-@fig:009]).

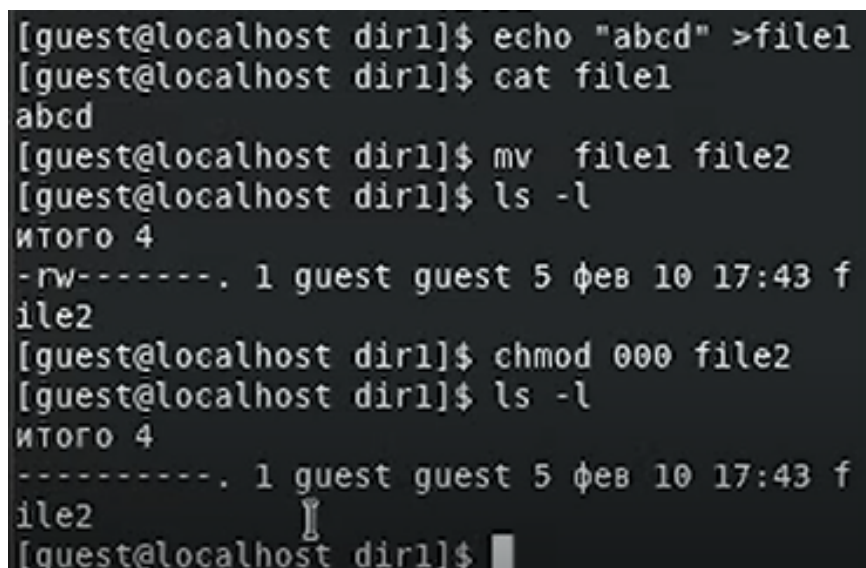


Рис. 9: Выполнение действий

10. Заменяем атрибут «а» атрибутом «і» от имени суперпользователя и выполним все действия по шагам. В результате придем к выводу, что никакие действия с файлом не разрешены.. (Рис. [-@fig:010]).

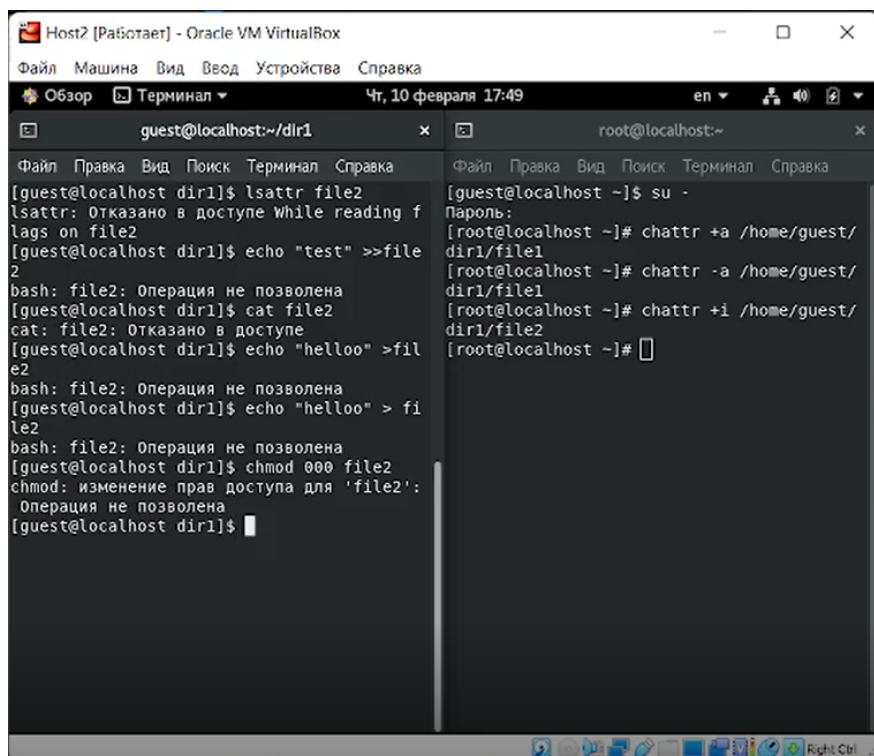


Рис. 10: Замена атрибута «а» на «і»

Выводы

В результате выполнения данной работы были практические навыки работы в консоли с расширенными атрибутами файлов.