

Отчёт по лабораторной работе №4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Федорина Эрнест Василевич

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	10
	Список литературы	11

Список иллюстраций

3.1	работа с атрибутами file1 от лица guest	7
3.2	ставим атрибут с помощью root пользователя	7
3.3	атрибут установлен правильно	8
3.4	работа с файлами после выставления различных атрибутов . . .	8
3.5	выставление и снятие атрибутов	9

Список таблиц

1 Цель работы

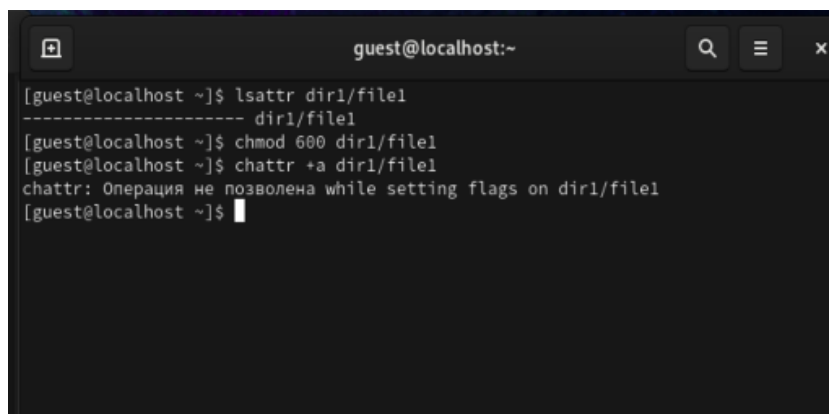
Получение практических навыков работы в консоли с расширенными атрибутами файлов

2 Теоретическое введение

`chmod` (от англ. *change mode*) — команда для изменения прав доступа к файлам и каталогам, используемая в Unix-подобных операционных системах. Входит в стандарт POSIX, в Coreutils.[1].

3 Выполнение лабораторной работы

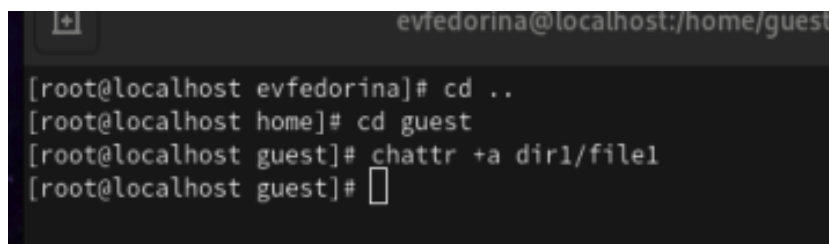
Для начала мы определим расширенные атрибуты файла file1, установим на файл определённые права командой `chmod`, а также попытаемся установить расширенный атрибут `+a`, от имени пользователя `guest` (рис. [3.1])



```
guest@localhost:~  
[guest@localhost ~]$ lsattr dir1/file1  
----- dir1/file1  
[guest@localhost ~]$ chmod 600 dir1/file1  
[guest@localhost ~]$ chattr +a dir1/file1  
chattr: Операция не позволена while setting flags on dir1/file1  
[guest@localhost ~]$
```

Рис. 3.1: работа с атрибутами file1 от лица guest

Как мы видим, установить атрибут пользователем `guest` не получается. Установим атрибут с помощью учётной записи `root` (рис. [3.2])



```
evfedorina@localhost:/home/guest  
[root@localhost evfedorina]# cd ..  
[root@localhost home]# cd guest  
[root@localhost guest]# chattr +a dir1/file1  
[root@localhost guest]#
```

Рис. 3.2: ставим атрибут с помощью root пользователя

Проверили правильность установления атрибута (рис. [3.3])

```
chmod: операция не позволена write setting  
[guest@localhost ~]$ lsattr dir1/file1  
-----a----- dir1/file1  
[guest@localhost ~]$
```

Рис. 3.3: атрибут установлен правильно

Проверили правильность установления атрибута (рис. [3.3])

Далее мы поработали с файлами - записывали, переименовывали и очищали содержимое файла file1 (file2). Затем снимали атрибут +a, и проделывали все шаги с атрибутом +i и без него (рис. [3.4]). Для этого приходилось ставить атрибуты с помощью пользователя root (рис. [3.5]).

```
guest@localhost dir1]$ cat file1  
est  
guest@localhost dir1]$ echo "abcd" > file1  
guest@localhost dir1]$ cat file1  
bcd  
guest@localhost dir1]$ mv file1 file2  
guest@localhost dir1]$ ls  
ile2  
guest@localhost dir1]$ echo "test" > file1  
guest@localhost dir1]$ echo "test" > file2  
guest@localhost dir1]$ ls  
ile1 file2  
guest@localhost dir1]$ cat file2  
est  
guest@localhost dir1]$ echo "abcd" > file2  
guest@localhost dir1]$ echo "test" > file2  
ash: file2: Операция не позволена  
guest@localhost dir1]$ echo "test" file2  
est file2  
guest@localhost dir1]$ echo "abcd" > file2  
ash: file2: Операция не позволена  
guest@localhost dir1]$ mv file2 file1  
v: невозможно переместить 'file2' в 'file1': Операция не позволена  
guest@localhost dir1]$
```

Рис. 3.4: работа с файлами после выставления различных атрибутов


```
[root@localhost guest]# chattr -a dir1/file1
[root@localhost guest]# chattr -i dir1/file1
chattr: Нет такого файла или каталога while trying to stat dir1/file1
[root@localhost guest]# chattr -i dir1/file2
[root@localhost guest]# chattr +i dir1/file2
[root@localhost guest]#
```

Рис. 3.5: выставление и снятие атрибутов

По итогу можно сказать, что выставление атрибутов +a или +i позволяет совершать множество манипуляций над файлами. Например, у нас получилось очистить содержимое файла или переименовать его.

4 Выводы

Получил практические навыки работы в консоли с расширенными атрибутами файлов

Список литературы

1. chmod [Электронный ресурс]. Wikimedia Foundation, Inc., 2024. URL: <https://en.wikipedia.org/wiki/Chmod>.