

# **Отчет по лабораторной работе №2**

**Дискреционное разграничение прав в Linux. Основные атрибуты**

Исаханян Эдуард Тигранович

2022 Sep 13th

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	13
	Список литературы	14

# List of Tables

# List of Figures

3.1	Создание пользователя guest и задание ему пароля . . . . .	7
3.2	Вход в систему под guest . . . . .	8
3.3	Проверка директории . . . . .	8
3.4	Проверка имени пользователя . . . . .	8
3.5	Проверка ID . . . . .	9
3.6	Файл /etc/passwd . . . . .	9
3.7	Учётная запись guest в файле /etc/passwd . . . . .	9
3.8	Существующие директории . . . . .	10
3.9	Расширенные атрибуты поддиректорий . . . . .	10
3.10	Поддиректория dir1 . . . . .	10
3.11	Расширенные атрибуты поддиректорий . . . . .	11
3.12	Снятие с директорий все атрибуты . . . . .	11
3.13	Попытка создать файл . . . . .	11
3.14	Установленные права и разрешённые действия 1 . . . . .	12
3.15	Установленные права и разрешённые действия 2 . . . . .	12
3.16	Минимальные права для совершения операций . . . . .	12

# 1 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.

## 2 Теоретическое введение<sup>1</sup>

В Линуксе существует 3 основных права доступа: - 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.

---

<sup>1</sup>Открытый источник информации

### 3 Выполнение лабораторной работы

В установленной при выполнении предыдущей лабораторной работы операционной системе создадим учётную запись пользователя guest (используя учётную запись администратора) и зададим пароль (рис. 3.1).

```
[edikisakhanyan@edikisakhayan ~]$ su root
Пароль:
[root@edikisakhayan edikisakhayan]# useradd guest
[root@edikisakhayan edikisakhayan]# passwd guest
Изменяется пароль пользователя guest.
Новый пароль :
НЕУДАЧНЫЙ ПАРОЛЬ: В пароле должно быть не меньше 8 символов
Повторите ввод нового пароля :
passwd: все данные аутентификации успешно обновлены.
```

Figure 3.1: Создание пользователя guest и задание ему пароля

Войдем в систему от имени пользователя guest (рис. 3.2).

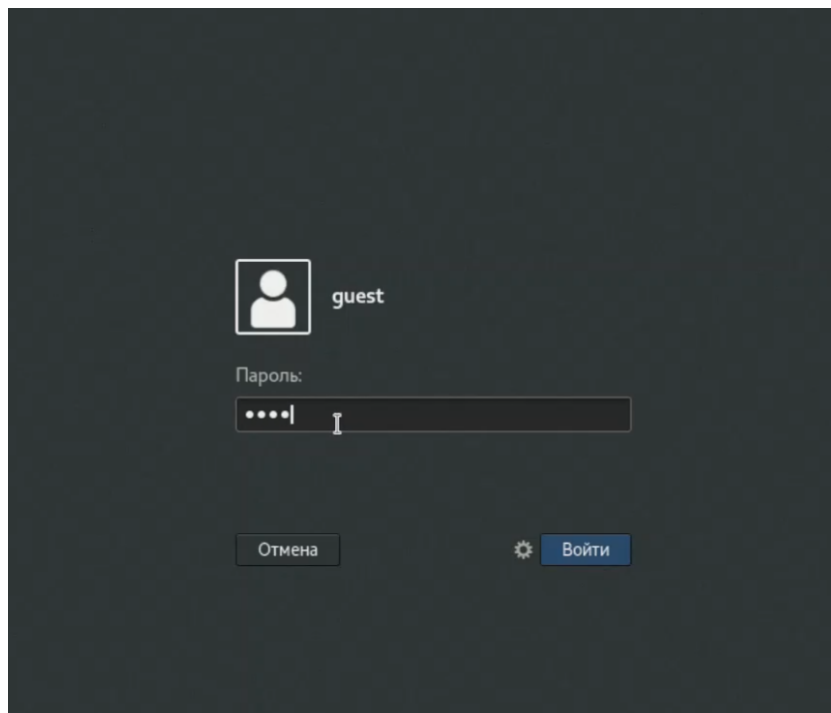


Figure 3.2: Вход в систему под guest

Определим директорию, в которой мы находимся (рис. 3.3).

```
[guest@edikisakhayan ~]$ pwd
/home/guest
```

Figure 3.3: Проверка директории

Как видно это наша домашняя директория. Она в целом совпадает с приглашением командной строки: в командной строке есть guest (пользователь) и ~ (указывает на то, что мы находимся в домашней директории).

Уточним имя нашего пользователя (рис. 3.4).

```
[guest@edikisakhayan ~]$ whoami
guest
```

Figure 3.4: Проверка имени пользователя

Уточним имя нашего пользователя, его группу, а также группы, куда он входит,



командой `id`. Выведенные значения `uid`, `gid` и др. запомнили. Выполним команду `groups`. Полученные значения совпадают с тем, что выдала `id` (рис. 3.5).

```
[guest@edikisakhayan ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@edikisakhayan ~]$ groups
guest
```

Figure 3.5: Проверка ID

Полученная информация об имени пользователя частично совпадает с данными, выводимыми в приглашении командной строки, но является более подробной.

Посмотрим файл `/etc/passwd` (рис. 3.6).

```
[guest@edikisakhayan ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
libstoragemgmt:x:998:996:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
colord:x:997:995:User for colord:/var/lib/colord:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
```

Figure 3.6: Файл `/etc/passwd`

Найдем в нем нашу учётную запись (рис. 3.7).

```
[guest@edikisakhayan ~]$ cat /etc/passwd | grep guest
guest:x:1001:1001:~/home/guest:/bin/bash
```

Figure 3.7: Учётная запись `guest` в файле `/etc/passwd`

Определим `uid` пользователя: 1001 и `gid` пользователя: 1001. Эти значения совпадают с полученными ранее значениями.

Определим существующие в системе директории (рис. 3.8).

```
[guest@edikisakhayan ~]$ ls -l /home/
итого 8
drwx-----. 15 edikisakhayan edikisakhayan 4096 сен 13 13:30 edikisakhayan
drwx-----. 15 guest guest 4096 сен 13 15:36 guest
```

Figure 3.8: Существующие директории

Как видим владельцы директорий имеют на них полные права, а группы и другие пользователи не имеют никаких прав на эти директории.

Проверим, какие расширенные атрибуты установлены на поддиректориях, находящихся в директории (рис. 3.9).

```
[guest@edikisakhayan ~]$ lsattr /home/
lsattr: Отказано в доступе While reading flags on /home/edikisakhayan
----- /home/guest
```

Figure 3.9: Расширенные атрибуты поддиректорий

Мы можем посмотреть только расширенные атрибуты директории guest, а расширенные атрибуты директорий других пользователей нам не доступны.

Создадим в домашней директории поддиректорию dir1: `mkdir dir1`. Определим командами `ls -l` и `lsattr`, какие права доступа и расширенные атрибуты были выставлены на директорию dir1 (рис. 3.10 - 3.11).

```
[guest@edikisakhayan ~]$ mkdir dir1
[guest@edikisakhayan ~]$ ls -l
итого 0
drwxrwxr-x. 2 guest guest 6 сен 13 15:41 dir1
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Видео
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Документы
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Загрузки
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Изображения
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Музыка
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Общедоступные
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Рабочий стол
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Шаблоны
[guest@edikisakhayan ~]$ lsattr /home/
lsattr: Отказано в доступе While reading flags on /home/edikisakhayan
----- /home/guest
```

Figure 3.10: Поддиректория dir1

```
[guest@edikisakhayan ~]$ lsattr /home/guest/
----- /home/guest/Рабочий стол
----- /home/guest/Загрузки
----- /home/guest/Шаблоны
----- /home/guest/Общедоступные
----- /home/guest/Документы
----- /home/guest/Музыка
----- /home/guest/Изображения
----- /home/guest/Видео
----- /home/guest/dir1
```

Figure 3.11: Расширенные атрибуты поддиректорий

Снимем с директории dir1 все атрибуты: `chmod 000 dir1`, и проверим с её помощью правильность выполнения команды `ls -l` (рис. 3.12).

```
[guest@edikisakhayan ~]$ chmod 000 dir1
[guest@edikisakhayan ~]$ ls -l
итого 0
d----- . 2 guest guest 6 сен 13 15:41 dir1
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Видео
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Документы
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Загрузки
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Изображения
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Музыка
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Öffentlich
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Рабочий стол
drwxr-xr-x. 2 guest guest 6 сен 13 15:35 Шаблоны
```

Figure 3.12: Снятие с директорий все атрибуты

Попытаемся создать в директории dir1 файл file1: `echo "test" > /home/guest/dir1/file1` (рис. 3.13).

```
[guest@edikisakhayan ~]$ echo "test" > /home/guest/dir1/file1
bash: /home/guest/dir1/file1: Отказано в доступе
```

Figure 3.13: Попытка создать файл

Я получим отказ в выполнении операции по созданию файла, т. к. мы сняли с директории все атрибуты (даже для владельцев). Сообщение об ошибке никак не отразилось на создании файла, потому что он не был создан.

Заполним таблицу «Установленные права и разрешённые действия», выполняя действия от имени владельца директории (файлов), определив опытным путём,

какие операции разрешены, а какие нет. Если операция разрешена, занесем в таблицу знак «+», если не разрешена – знак «-» (рис. 3.14 - 3.15).

Права директории	Права файла	Создание файла	Удаление файла	Запись в файл	Чтение файла	Смена директории	Просмотр файлов в директории	Переименование файла	Смена атрибутов файла
d----- (000)	(000)-(700)	-	-	-	-	-	-	-	-
d-x----- (100)	(000)	-	-	-	-	+	-	-	+
d-x----- (100)	(100)	-	-	-	-	+	-	-	+
d-x----- (100)	(200)	-	-	+	-	-	-	-	+
d-x----- (100)	(300)	-	-	+	-	+	-	-	+
d-x----- (100)	(400)	-	-	-	+	-	-	-	+
d-x----- (100)	(500)	-	-	-	+	-	-	-	+
d-x----- (100)	(600)	-	-	+	+	+	-	-	+
d-x----- (100)	(700)	-	-	+	+	+	-	-	+
d-x----- (100)	(000)-(700)	-	-	-	-	-	-	-	-
dwx----- (200)	(000)-(700)	+	+	-	-	+	-	+	+
dwx----- (300)	(000)	+	+	-	-	+	-	+	+
dwx----- (300)	(100)	+	+	-	-	+	-	+	+
dwx----- (300)	(200)	+	+	+	-	+	-	+	+
dwx----- (300)	(300)	+	+	+	-	+	-	+	+
dwx----- (300)	(400)	+	+	-	+	+	-	+	+
dwx----- (300)	(500)	+	+	-	+	+	-	+	+
dwx----- (300)	(600)	+	+	+	+	+	-	+	+
dwx----- (300)	(700)	+	+	+	+	+	-	+	+
dwx----- (400)	(000)-(700)	-	-	-	-	+	-	-	-
drwx----- (500)	(000)	-	-	-	-	+	-	-	+
drwx----- (500)	(100)	-	-	-	-	+	-	-	+
drwx----- (500)	(200)	-	-	+	-	+	-	-	+
drwx----- (500)	(300)	-	-	+	-	+	-	-	+
drwx----- (500)	(400)	-	-	-	+	+	-	-	+
drwx----- (500)	(500)	-	-	-	+	+	-	-	+
drwx----- (500)	(600)	-	-	+	+	+	-	-	+
drwx----- (500)	(700)	-	-	+	+	+	-	-	+
drwx----- (600)	(000)-(700)	-	-	-	-	+	-	-	-
drwx----- (700)	(000)	+	+	-	-	+	-	+	+
drwx----- (700)	(100)	+	+	-	-	+	-	+	+
drwx----- (700)	(200)	+	+	-	-	+	-	+	+
drwx----- (700)	(300)	+	+	+	-	+	-	+	+
drwx----- (700)	(400)	+	+	-	+	+	-	+	+

Figure 3.14: Установленные права и разрешённые действия 1

drwx----- (700)	(300)	+	+	+	-	+	+	+	+
drwx----- (700)	(400)	+	+	-	+	+	+	+	+
drwx----- (700)	(500)	+	+	-	+	+	+	+	+
drwx----- (700)	(600)	+	+	+	+	+	+	+	+
drwx----- (700)	(700)	+	+	+	+	+	+	+	+

Figure 3.15: Установленные права и разрешённые действия 2

На основании заполненной таблицы определим те или иные минимально необходимые права для выполнения операций внутри директории dir1, заполним таблицу «Минимальные права для совершения операций» (рис. 3.16).

Операции	Минимальные права на директорию	Минимальные права на файл
Создание файла	d-wx----- (300)	(000)
Удаление файла	d-wx----- (300)	(000)
Чтение файла	d-x----- (100)	(400)
Запись в файл	d-x----- (100)	(200)
Переименование файла	d-wx----- (300)	(000)
Создание поддиректории	d-wx----- (300)	(000)
удаление поддиректории	d-wx----- (300)	(000)

Figure 3.16: Минимальные права для совершения операций

## 4 Выводы

В ходе работы, мы получили практические навыки работы в консоли с атрибутами файлов, закрепили теоретические основы дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux.

# Список литературы

1. Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН” <https://esystem.rudn.ru/>  
::: {#refs} :::