

# **Лабораторная работа №2**

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

Смородова Дарья Владимировна

2022 Sep 16th

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	16
5	Список литературы	17

## List of Tables

# List of Figures

3.1	Создание пользователя guest и задание ему пароля . . . . .	8
3.2	Вход в систему под guest . . . . .	8
3.3	Проверка директории и переход в домашнюю директорию . . . . .	9
3.4	Уточнение имени пользователя . . . . .	9
3.5	Проверка id . . . . .	9
3.6	Файл /etc/passwd . . . . .	10
3.7	Учётная запись guest в файле /etc/passwd . . . . .	10
3.8	Существующие директории . . . . .	10
3.9	Расширенные атрибуты поддиректорий . . . . .	11
3.10	Создание поддиректории dir1 и команда ls -l . . . . .	11
3.11	Просмотр расширенных атрибутов поддиректорий . . . . .	11
3.12	Снятие с директории dir1 всех атрибутов . . . . .	12
3.13	Попытка создать файл file1 в директории dir1 . . . . .	12
3.14	Проверка установленных прав и разрешенных действий . . . . .	13
3.15	Установленные права и разрешённые действия 1 . . . . .	13
3.16	Установленные права и разрешённые действия 2 . . . . .	14
3.17	Минимальные права для совершения операций . . . . .	14
3.18	Создание и удаление поддиректорий . . . . .	15

# 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

В операционной системе Linux есть много отличных функций безопасности, но она 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.

Изначально каждый файл имеет три параметра 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.

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. <sup>1</sup>

---

<sup>1</sup>Права доступа к файлам в Linux

## 3 Выполнение лабораторной работы <sup>1</sup>

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

```
[smorodovadv@smorodovadv ~]$ su root
Пароль:
[root@smorodovadv smorodovadv]# useradd guest
[root@smorodovadv smorodovadv]# passwd guest
Изменение пароля пользователя guest.
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: Пароль должен содержать не менее 8 символов
Повторите ввод нового пароля:
passwd: данные аутентификации успешно обновлены.
[root@smorodovadv smorodovadv]#
```

Figure 3.1: Создание пользователя guest и задание ему пароля

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

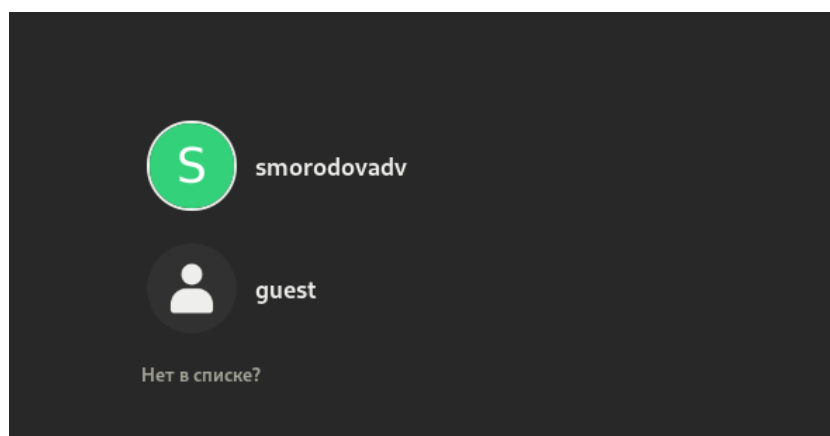


Figure 3.2: Вход в систему под guest

---

<sup>1</sup>Методические материалы к лабораторной работе



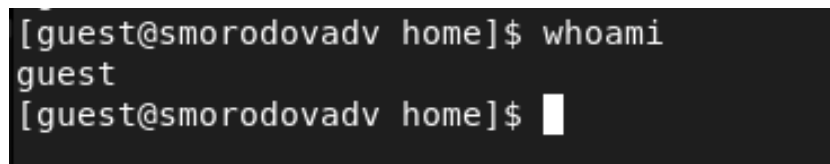
3. Определим директорию, в которой мы находимся при помощи команды `pwd`, и переход в домашнюю директорию (рис. 3.3):



```
guest@smorodovadv:/home
[guest@smorodovadv ~]$ pwd
/home/guest
[guest@smorodovadv ~]$ cd ..
[guest@smorodovadv home]$
```

Figure 3.3: Проверка директории и переход в домашнюю директорию

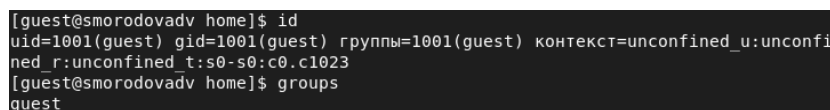
4. Уточним имя нашего пользователя при помощи команды `whoami` (рис. 3.4):



```
[guest@smorodovadv home]$ whoami
guest
[guest@smorodovadv home]$
```

Figure 3.4: Уточнение имени пользователя

5. Уточним имя нашего пользователя, его группу, а также группы, куда он входит, командой `id`. Выведенные значения `uid`, `gid` и др. запомним. Выполним команду `groups`. Полученные значения совпадают с тем, что выдала `id`. Полученная информация об имени пользователя частично совпадает с данными, выводимыми в приглашении командной строки, но является более подробной (рис. 3.5):



```
[guest@smorodovadv home]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@smorodovadv home]$ groups
guest
```

Figure 3.5: Проверка `id`

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

```
[guest@smorodovadv home]$ 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:65534:65534:Kernel Overflow User:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
sssd:x:997:993:User for sssd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
pipewire:x:996:992:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
libstoragemgmt:x:995:991:daemon account for libstoragemgmt:/var/run/lsm:/sbin
login
```

Figure 3.6: Файл /etc/passwd

7. Найдем в нем нашу учётную запись при помощи команды `cat /etc/passwd | grep guest`. Uid пользователя: 1001, gid пользователя: 1001. Эти значения совпадают с полученными в ранее. (рис. 3.7):

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

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

8. При помощи команды `ls -l /home/` определим существующие в системе директории. Владельцы директорий имеют на них полные права, а группы и другие пользователи не имеют никаких прав на эти директории (рис. 3.8):

```
[guest@smorodovadv home]$ ls -l /home/
итого 8
drwx-----. 14 guest      guest      4096 сен 14 17:38 guest
drwx-----. 14 smorodovadv smorodovadv 4096 сен 14 17:30 smorodovadv
[guest@smorodovadv home]$
```

Figure 3.8: Существующие директории

9. Проверим, какие расширенные атрибуты установлены на поддиректориях, находящихся в директории /home при помощи команды `lsattr /home`. Нам удалось посмотреть только расширенные атрибуты директории guest, а

расширенные атрибуты директорий других пользователей нам не доступны (рис. 3.9):

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

Figure 3.9: Расширенные атрибуты поддиректорий

10. Создадим в домашней директории поддиректорию `dir1` при помощи команды `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@smorodovadv ~]$ mkdir dir1
[guest@smorodovadv ~]$ ls -l /home/guest/dir1
итого 0
[guest@smorodovadv ~]$ ls -l
итого 0
drwxrwxr-x. 2 guest guest 6 сен 14 17:49 dir1
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Видео
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Документы
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Загрузки
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Изображения
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Музыка
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Общедоступные
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 'Рабочий стол'
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Шаблоны
```

Figure 3.10: Создание поддиректории `dir1` и команда `ls -l`

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

Figure 3.11: Просмотр расширенных атрибутов поддиректорий

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

```
[guest@smorodovadv ~]$ chmod 000 dir1
[guest@smorodovadv ~]$ ls -l
итого 0
d----- . 2 guest guest 6 сен 14 17:49 dir1
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Видео
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Документы
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Загрузки
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Изображения
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Музыка
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Общедоступные
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 'Рабочий стол'
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Шаблоны
[guest@smorodovadv ~]$
```

Figure 3.12: Снятие с директории dir1 всех атрибутов

12. Попытаемся создать в директории dir1 файл file1 при помощи команды `echo "test" > /home/guest/dir1/file1`. При попытке создания был получен отказ, так как до этого я сняла с директории все атрибуты. Сообщение об ошибке никак не отразилось на создании файла, потому что он не был создан (рис. 3.13):

```
[guest@smorodovadv ~]$ chmod 000 dir1
[guest@smorodovadv ~]$ ls -l
итого 0
d----- . 2 guest guest 6 сен 14 17:49 dir1
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Видео
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Документы
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Загрузки
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Изображения
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Музыка
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Общедоступные
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 'Рабочий стол'
drwxr-xr-x. 2 guest guest 6 сен 14 17:37 Шаблоны
[guest@smorodovadv ~]$ echo "test" > /home/guest/dir1/file1
bash: /home/guest/dir1/file1: Отказано в доступе
[guest@smorodovadv ~]$ ls -l /home/guest/dir1
ls: невозможно открыть каталог '/home/guest/dir1': Отказано в доступе
[guest@smorodovadv ~]$ ls
dir1  документы  Изображения  Общедоступные  Шаблоны
Видео  Загрузки  Музыка  'Рабочий стол'
[guest@smorodovadv ~]$ cd dir1
bash: cd: dir1: Отказано в доступе
[guest@smorodovadv ~]$
```

Figure 3.13: Попытка создать файл file1 в директории dir1

13. 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 «-».
14. Порядок команд, при помощи которых проводилась проверка (рис. 3.14):

```
[guest@smorodovadv dir1]$ mv f1 file
[guest@smorodovadv dir1]$ chmod 600 file
[guest@smorodovadv dir1]$ touch file1
[guest@smorodovadv dir1]$ rm file1
[guest@smorodovadv dir1]$ echo "text4" > file
[guest@smorodovadv dir1]$ cat file
text4
[guest@smorodovadv dir1]$ ls
file
[guest@smorodovadv dir1]$ mv file f1
[guest@smorodovadv dir1]$ mv f1 file
[guest@smorodovadv dir1]$ chmod 700 file
```

Figure 3.14: Проверка установленных прав и разрешенных действий

15. Получившаяся таблица «Установленные права и разрешённые действия» (рис. 3.15 - 3.16):

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)	-	-	-	-	-	-	-	-
d(000)	(100)	-	-	-	-	-	-	-	-
d(000)	(200)	-	-	-	-	-	-	-	-
d(000)	(300)	-	-	-	-	-	-	-	-
d(000)	(400)	-	-	-	-	-	-	-	-
d(000)	(500)	-	-	-	-	-	-	-	-
d(000)	(600)	-	-	-	-	-	-	-	-
d(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-w----- (200)	(000)	-	-	-	-	-	-	-	-
d-w----- (200)	(100)	-	-	-	-	-	-	-	-
d-w----- (200)	(200)	-	-	-	-	-	-	-	-
d-w----- (200)	(300)	-	-	-	-	-	-	-	-
d-w----- (200)	(400)	-	-	-	-	-	-	-	-
d-w----- (200)	(500)	-	-	-	-	-	-	-	-
d-w----- (200)	(600)	-	-	-	-	-	-	-	-
d-w----- (200)	(700)	-	-	-	-	-	-	-	-
d-wx----- (300)	(000)	+	+	-	-	+	-	+	+
d-wx----- (300)	(100)	+	+	-	-	+	-	+	+
d-wx----- (300)	(200)	+	+	+	-	+	-	+	+
d-wx----- (300)	(300)	+	+	+	-	+	-	+	+
d-wx----- (300)	(400)	+	+	-	+	+	-	+	+
d-wx----- (300)	(500)	+	+	-	+	+	-	+	+
d-wx----- (300)	(600)	+	+	+	+	+	-	+	+
d-wx----- (300)	(700)	+	+	+	+	+	-	+	+

Figure 3.15: Установленные права и разрешённые действия 1

dr-----	(400)	(000)	-	-	-	-	-	+	-	-
dr-----	(400)	(100)	-	-	-	-	-	+	-	-
dr-----	(400)	(200)	-	-	-	-	-	+	-	-
dr-----	(400)	(300)	-	-	-	-	-	+	-	-
dr-----	(400)	(400)	-	-	-	-	-	+	-	-
dr-----	(400)	(500)	-	-	-	-	-	+	-	-
dr-----	(400)	(600)	-	-	-	-	-	+	-	-
dr-----	(400)	(700)	-	-	-	-	-	+	-	-
dr-x-----	(500)	(000)	-	-	-	-	+	+	-	+
dr-x-----	(500)	(100)	-	-	-	-	+	+	-	+
dr-x-----	(500)	(200)	-	-	+	-	+	+	-	+
dr-x-----	(500)	(300)	-	-	+	-	+	+	-	+
dr-x-----	(500)	(400)	-	-	-	+	+	+	-	+
dr-x-----	(500)	(500)	-	-	-	+	+	+	-	+
dr-x-----	(500)	(600)	-	-	+	+	+	+	-	+
dr-x-----	(500)	(700)	-	-	+	+	+	+	-	+
drwx-----	(600)	(000)	-	-	-	-	-	+	-	-
drwx-----	(600)	(100)	-	-	-	-	-	+	-	-
drwx-----	(600)	(200)	-	-	-	-	-	+	-	-
drwx-----	(600)	(300)	-	-	-	-	-	+	-	-
drwx-----	(600)	(400)	-	-	-	-	-	+	-	-
drwx-----	(600)	(500)	-	-	-	-	-	+	-	-
drwx-----	(600)	(600)	-	-	-	-	-	+	-	-
drwx-----	(600)	(700)	-	-	-	-	-	+	-	-
drwx-----	(700)	(000)	+	+	-	-	+	+	+	+
drwx-----	(700)	(100)	+	+	-	-	+	+	+	+
drwx-----	(700)	(200)	+	+	+	-	+	+	+	+
drwx-----	(700)	(300)	+	+	+	-	+	+	+	+
drwx-----	(700)	(400)	+	+	-	+	+	+	+	+
drwx-----	(700)	(500)	+	+	-	+	+	+	+	+
drwx-----	(700)	(600)	+	+	+	+	+	+	+	+
drwx-----	(700)	(700)	+	+	+	+	+	+	+	+

Figure 3.16: Установленные права и разрешённые действия 2

16. 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. 3.17):

Операция	Минимальные права на директорию	Минимальные права на файл
Создание файла	d(300)	(000)
Удаление файла	d(300)	(000)
Чтение файла	d(100)	(400)
Запись в файл	d(100)	(200)
Переименование файла	d(300)	(000)
Создание поддиректории	d(300)	(000)
Удаление поддиректории	d(300)	(000)

Figure 3.17: Минимальные права для совершения операций

17. Для проверки минимальных прав на директорию и минимальных прав на файл при создании и удалении поддиректорий выполним следующие команды (рис. 3.18):

```

[guest@smorodovadv ~]$ chmod 000 dir1
[guest@smorodovadv ~]$ mkdir dir1/dir2
mkdir: невозможно создать каталог «dir1/dir2»: Отказано в доступе
[guest@smorodovadv ~]$ chmod 100 dir1
[guest@smorodovadv ~]$ mkdir dir1/dir2
mkdir: невозможно создать каталог «dir1/dir2»: Отказано в доступе
[guest@smorodovadv ~]$ chmod 200 dir1
[guest@smorodovadv ~]$ mkdir dir1/dir2
mkdir: невозможно создать каталог «dir1/dir2»: Отказано в доступе
[guest@smorodovadv ~]$ chmod 300 dir1
[guest@smorodovadv ~]$ mkdir dir1/dir2
[guest@smorodovadv ~]$ cd dir1
[guest@smorodovadv dir1]$ ls
ls: невозможно открыть каталог '.': Отказано в доступе
[guest@smorodovadv dir1]$ cd dir2
[guest@smorodovadv dir2]$ cd ..
[guest@smorodovadv dir1]$ cd ..
[guest@smorodovadv ~]$ chmod 000 dir1
[guest@smorodovadv ~]$ rmdir dir1/dir2
rmdir: не удалось удалить 'dir1/dir2': Отказано в доступе
[guest@smorodovadv ~]$ chmod 100 dir1
[guest@smorodovadv ~]$ rmdir dir1/dir2
rmdir: не удалось удалить 'dir1/dir2': Отказано в доступе
[guest@smorodovadv ~]$ chmod 200 dir1
[guest@smorodovadv ~]$ rmdir dir1/dir2
rmdir: не удалось удалить 'dir1/dir2': Отказано в доступе
[guest@smorodovadv ~]$ chmod 300 dir1
[guest@smorodovadv ~]$ rmdir dir1/dir2
[guest@smorodovadv ~]$ cd dir1
[guest@smorodovadv dir1]$ cd dir2
bash: cd: dir2: Нет такого файла или каталога
[guest@smorodovadv dir1]$

```

Figure 3.18: Создание и удаление поддиректорий

## 4 Выводы

В ходе данной лабораторной 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.



## 5 Список литературы

1. Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН”
2. Права доступа к файлам в Linux