

Лабораторная работа №7

Анализ файловой системы Linux. Команды для работы с файлами и каталогами

Буллер Татьяна Александровна

Содержание

1	Цель работы	4
2	Выполнение заданий примеров	5
2.1	Копирование и создание файлов.	5
2.2	Перемещение и переименование файлов и каталогов	8
2.3	Права доступа и их изменение	9
3	Выполнение лабораторной работы	12
3.1	Перемещение, копирование и переименование файлов и каталогов	12
3.2	Права доступа и их изменение	13
3.3	Продолжение работы с файлами и каталогами	15
3.4	Вызов ошибок доступа	15
3.5	Руководства man	16
4	Выводы	18

Список иллюстраций

2.1	Копирование файла в текущем каталоге	5
2.2	Копирование нескольких файлов в каталог	6
2.3	Копирование файлов в произвольном каталоге	6
2.4	Копирование каталогов в текущем каталоге	7
2.5	Копирование каталогов в произвольном каталоге	7
2.6	Переименование файлов в текущем каталоге	8
2.7	Перемещение файлов в другой каталог	8
2.8	Переименование каталогов в текущем каталоге	9
2.9	Перемещение каталога в другой каталог. Переименование каталога, не являющегося текущим	9
2.10	Добавление и удаление права исполнения	10
2.11	Изъятие права чтения для группы и других пользователей	10
2.12	Добавление права записи для группы	11
3.1	Копирование и перемещение файла	12
3.2	Копирование и перемещение файлов	13
3.3	Создание и перемещение нового каталога	13
3.4	Редактирование доступа к файлам	14
3.5	Редактирование доступа к каталогам	14
3.6	password, не passwd	15
3.7	Махинации с копированием и перемещением файлов и каталогов	15
3.8	Махинации с правами доступа файлов и каталогов	16
3.9	руководство к команде mount	16
3.10	руководство к команде fsck	17
3.11	руководство к команде mkfs	17
3.12	руководство к команде kill	17

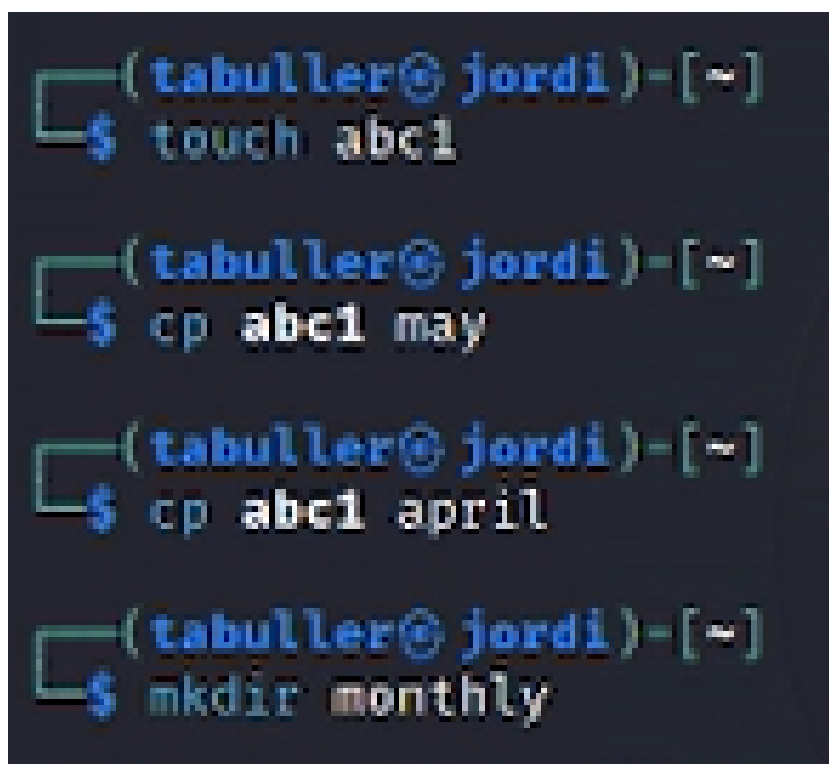
1 Цель работы

Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

2 Выполнение заданий примеров

2.1 Копирование и создание файлов.

Необходимо скопировать файл ~/abc1 в файл april и в файл may. Для этого командой touch создаем сам файл, далее с помощью команды cp копируем его содержимое в новые файлы.



```
(tabuller@jordi)-[~]  
$ touch abc1  
  
(tabuller@jordi)-[~]  
$ cp abc1 may  
  
(tabuller@jordi)-[~]  
$ cp abc1 april  
  
(tabuller@jordi)-[~]  
$ mkdir monthly
```

Рис. 2.1: Копирование файла в текущем каталоге

Теперь нужно скопировать файлы april и may в каталог monthly - для этого с помощью mkdir создаем каталог monthly, с помощью уже известной cp копиру-

ем файлы в каталог. Припишем названия файлов в одну строку - так они скопируются в рамках одной команды.

```
(tabuller@jordi)-[~]  
$ mkdir monthly  
  
(tabuller@jordi)-[~]  
$ cd april may monthly  
cd: too many arguments  
  
(tabuller@jordi)-[~]  
$ cp april may monthly
```

Рис. 2.2: Копирование нескольких файлов в каталог

После этого скопируем файл monthly/may в файл с именем june - для этого используем команду cp и прописываем полный путь до файла.

```
(tabuller@jordi)-[~]  
$ cp april may monthly  
  
(tabuller@jordi)-[~]  
$ cp monthly/may monthly/june  
  
(tabuller@jordi)-[~]  
$ ls monthly  
april june may
```

Рис. 2.3: Копирование файлов в произвольном каталоге

Теперь скопируем каталог monthly в каталог monthly.00: для этого используем cp -r. Опция копируем рекурсивно - весь каталог перемещается в созданный

НОВЫЙ.

```
(tabuller@jordi)-[~]  
$ mkdir monthly.00  
  
(tabuller@jordi)-[~]  
$ cp -r monthly monthly.00  
  
(tabuller@jordi)-[~]  
$ ls monthly.00  
monthly
```

Рис. 2.4: Копирование каталогов в текущем каталоге

Далее по заданию нужно скопировать каталог monthly.00 в каталог /tmp. Для этого опять используем опцию рекурсивного копирования и прописываем полный путь к файлу (каталогу), куда хотим копировать.

```
(tabuller@jordi)-[~]  
$ cp -r monthly.00 /tmp  
  
(tabuller@jordi)-[~]  
$ ls  
abc1 Desktop Downloads LIC  
april Documents image may  
  
(tabuller@jordi)-[~]  
$ ls /tmp  
monthly.00
```

Рис. 2.5: Копирование каталогов в произвольном каталоге

2.2 Перемещение и переименование файлов и каталогов

Задание: изменить название файла `april` на `july` в домашнем каталоге. Для этого используем команду `mv`: де-юре название будет изменено, де-факто мы перемещаем файл в файл с другим названием и удаляем предыдущий одной командой.

```
(tabuller@jordi)-[~]  
$ mv april july  
  
(tabuller@jordi)-[~]  
$ ls  
abc1  Documents  image  LICENSE  monthly  Music  Public  ssh1.pub  Videos  
Desktop Downloads  july    may      monthly.00 Pictures ssh1    Templates work
```

Рис. 2.6: Переименование файлов в текущем каталоге

Чтобы переместить файл `july` в каталог `monthly.00` используем ту же команду, на этот раз прописывая полный путь до места назначения. При проверке новый файл появляется вместо с остальным содержимым каталога.

```
(tabuller@jordi)-[~]  
$ mv july monthly.00  
  
(tabuller@jordi)-[~]  
$ ls monthly.00  
july  monthly
```

Рис. 2.7: Перемещение файлов в другой каталог

Необходимо также переименовать каталог `monthly.00` в `monthly.01`. Для этого снова используем `mv`, работаем точно так же, как когда нам нужно было переименовать файл.


```
(tabuller@jordi)-[~]
$ mv monthly.00 monthly.01

(tabuller@jordi)-[~] u become, the more you are able to hear"
$ ls
abc1    Documents  image    may      monthly.01  Pictures  ssh1     Templates  work
Desktop Downloads  LICENSE  monthly   Music       Public    ssh1.pub  Videos
```

Рис. 2.8: Переименование каталогов в текущем каталоге

Переместить каталог `monthly.01` в каталог `reports`. Для этого сперва создаем каталог с нужным именем, после этого перемещаем каталог, как до этого делали с файлами. Чтобы далее переименовать каталог `reports/monthly.01` в `reports/monthly` снова используем ту же команду, но на этот раз, так как мы переименовываем файлы не в рабочем каталоге, прописываем полный путь до них:

```
(tabuller@jordi)-[~]
$ mkdir reports | mv monthly.01 reports

(tabuller@jordi)-[~]
$ ls reports
monthly.01

(tabuller@jordi)-[~]
$ mv reports/monthly.01 reports/monthly

(tabuller@jordi)-[~]
$ ls reports
monthly
```

Рис. 2.9: Перемещение каталога в другой каталог. Переименование каталога, не являющегося текущим

2.3 Права доступа и их изменение

В каталоге существует файл `may`. Дадим владельцу права на его исполнение, после чего заберем назад и посмотрим, что получилось. После добавления права исполнения (+u) для владельца (u) при просмотре файла он подсвечивается

зеленым - теперь его можно исполнять, а в строке прав появляется дополнительный символ. Когда же право изымается, файл возвращается к исходному состоянию.

```
(tabuller@jordi)-[~]
$ ls -l may
-rw-r--r-- 1 tabuller buller 0 Mar 23 03:06 may

(tabuller@jordi)-[~]
$ chmod u+x may

(tabuller@jordi)-[~]
$ ls -l may
-rwxr--r-- 1 tabuller buller 0 Mar 23 03:06 may

(tabuller@jordi)-[~]
$ chmod u-x may

(tabuller@jordi)-[~]
$ ls -l may
-rw-r--r-- 1 tabuller buller 0 Mar 23 03:06 may
```

Рис. 2.10: Добавление и удаление права исполнения

Требуется создать каталог `monthly` с запретом на чтение для членов группы и всех остальных пользователей. Каталог уже создан, для того, чтобы лишить пользователей прав чтения, используем `chmod`. `-r` - лишение права чтения, для группы (`g`) и всех пользователей (`o`) соответственно.

```
(tabuller@jordi)-[~]
$ chmod g-r monthly | chmod o-r monthly
```

Рис. 2.11: Изъятие права чтения для группы и других пользователей

Теперь создадим новый файл и дадим группе права на запись в него. Пользуемся тем же, чем пользовались, когда добавляли владельцу права на исполнение, но теперь работаем с группой (`g`) и правом записи (`+w`)

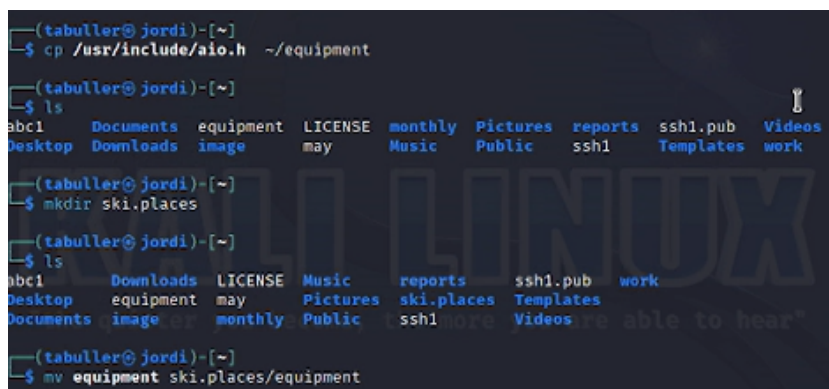
```
(tabuller@jordi)-[~]  
$ touch abc1  
  
(tabuller@jordi)-[~]  
$ chmod g+w abc1  
  
(tabuller@jordi)-[~]  
$ ls -l abc1  
-rw-rw-r-- 1 tabuller buller 0 Mar 23 03:17 abc1
```

Рис. 2.12: Добавление права записи для группы

3 Выполнение лабораторной работы

3.1 Перемещение, копирование и переименование файлов и каталогов

Задание: скопируйте файл (...) в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него. Проблема: моя система не имеет каталога `/usr/include/sys/` в принципе. Вместо этого копируем файл из `usr/include/`, проверяем его наличие - в выводе команды `ls` он отображается - и переходим к следующему шагу: создадим новую директорию и переместим скопированный файл в него.



```
(tabuller@jordi)-[~]
$ cp /usr/include/aio.h ~/equipment

(tabuller@jordi)-[~]
$ ls
abc1  Documents  equipment  LICENSE  monthly  Pictures  reports  ssh1.pub  Videos
Desktop Downloads  image     may      Music    Public   ssh1     Templates work

(tabuller@jordi)-[~]
$ mkdir ski.places

(tabuller@jordi)-[~]
$ ls
abc1  Downloads  LICENSE  Music  reports  ssh1.pub  work
Desktop  equipment  may      Pictures  ski.places  Templates
Documents image     monthly  Public   ssh1     more      Videos
are able to hear"

(tabuller@jordi)-[~]
$ mv equipment ski.places/equipment
```

Рис. 3.1: Копирование и перемещение файла

Далее скопируем под другим названием файл `abc1`, для этого используем команду `cp`. Проверим - файлы скопировались, все на месте. Далее создаем новую директорию и переносим новые файлы туда.

```

(tabuller@jordi)-[~]
$ cp abc1 ski.places/equiplist2

(tabuller@jordi)-[~]
$ ls ski.places
equiplist  equiplist2

(tabuller@jordi)-[~]
$ mkdir ski.places/equipment

(tabuller@jordi)-[~]
$ ls ski.places
equiplist  equiplist2  equipment

(tabuller@jordi)-[~]
$ mv ski.places/equiplist ski.places/equiplist2 ski.places/equipment

(tabuller@jordi)-[~]
$ ls ski.places
equipment

```

Рис. 3.2: Копирование и перемещение файлов

Теперь создадим в домашнем каталоге новую директорию и переместим ее (уже под другим именем) в созданный каталог `ski.places` - для этого используем `mkdir` и `mv` соответственно.

```

(tabuller@jordi)-[~]
$ mkdir newdir

(tabuller@jordi)-[~]
$ mv newdir ski.places/plans

(tabuller@jordi)-[~]
$ ls ski.places
equipment  plans

```

Рис. 3.3: Создание и перемещение нового каталога

3.2 Права доступа и их изменение

Далее определим опции, необходимые для изменение прав на файлы и директории. Что имеем по заданию: `- drwxr--r-- ... australia - drwx--x--x ... play - -r-xr--r-- ... my_os - -rw-rw-r-- ... feathers`

Видим, что первые два файла являются директориями, последние - отдельными файлами. Начнем с конца. Изначально владелец файла имеет право на чтение и запись, все остальные (группа и прочие пользователи) - только на чтение. Для файлов нам необходимо: - убрать у владельца my_os право на запись, добавить право на исполнение (u-w+x) - добавить группе владельца файла feathers право на запись в файл (g+w)

```
(tabuller@jordi)-[~/ski.places/plans]
$ chmod u+x-w my_os

(tabuller@jordi)-[~/ski.places/plans]
$ chmod g+w feathers
```

Рис. 3.4: Редактирование доступа к файлам

australia и play являются каталогами, о чем говорит буква d в начале строки прав доступа. По умолчанию владелец директории имеет все права к ней, группа и другие пользователи - права на чтение и исполнение.

Для первого каталога необходимо убрать у всех, кроме владельца, права на исполнение. Для этого используем go-x. Для второго каталога убираем у всех, кроме владельца, права на чтение go-w.

```
(tabuller@jordi)-[~/ski.places/plans]
$ mkdir australia play

(tabuller@jordi)-[~/ski.places/plans]
$ chmod u+x australia

(tabuller@jordi)-[~/ski.places/plans]
$ chmod u+x play | chmod go+x-r play

(tabuller@jordi)-[~/ski.places/plans]
$ ls -l
total 8
drwxr-xr-x 2 tabuller buller 4096 Mar 23 03:29 australia
-rw-rw-r-- 1 tabuller buller   0 Mar 23 03:23 feathers
-r-xr--r-- 1 tabuller buller   0 Mar 23 03:23 my_os
drwx--x--x 2 tabuller buller 4096 Mar 23 03:29 play

(tabuller@jordi)-[~/ski.places/plans]
$ chmod go-x australia
```

Рис. 3.5: Редактирование доступа к каталогам

3.3 Продолжение работы с файлами и каталогами

Теперь продолжим работу с файлами. Попробуем вывести на экран содержимое файла `/etc/password` - и столкнемся с ошибкой, потому что на данном дистрибутиве такого файла в системе нет. Его функции выполняет `/etc/passwd`.

```
(tabuller@jordi)-[~/ski.places/plans]
$ cat /etc/password
cat: /etc/password: No such file or directory
```

Рис. 3.6: password, не passwd

Далее скопируем уже имеющийся файл `feathers` под новым названием в домашнюю директорию. `file.old` теперь перемещаем в директорию `play`, копируем этот каталог в созданный в домашнем каталоге `fun` и уже его переносим назад в `play/games`

```
(tabuller@jordi)-[~]
$ cp ski.places/plans/feathers ~/file.old

(tabuller@jordi)-[~]
$ mv file.old ski.places/plans/play

(tabuller@jordi)-[~]
$ cp ski.places/plans/play ~/fun
cp: -r not specified; omitting directory 'ski.places/plans/play'

(tabuller@jordi)-[~]
$ cp -r ski.places/plans/play ~/fun

(tabuller@jordi)-[~]
$ mv fun ski.places/plans/play/games
```

Рис. 3.7: Махинации с копированием и перемещением файлов и каталогов

3.4 Вызов ошибок доступа

Теперь поработаем с правами доступа. Лишим владельца файла `~/feathers` права на чтение. При просмотре файла `~/feathers` командой `cat` мы встречаемся с ошибкой: доступ запрещен, мы только что отобрали у себя это право. То же самое будет при попытке скопировать файл `~/feathers`. Смилюемся и вернем право

на чтение: u+r.

Далее лишаем владельца каталога ~/play права на выполнение. При попытке перейти в этот каталог снова видим ту же ошибку: доступ запрещен, так как я запретила владельцу (себе) исполнять этот файл (каталог). Возвращаю права на выполнение - теперь все будет работать.

```
(tabuller@jordi)~$ chmod u-r ski.places/plans/feathers
(tabuller@jordi)~$ cat ski.places/plans/feathers
cat: ski.places/plans/feathers: Permission denied
(tabuller@jordi)~$ chmod u+x ski.places/plans/feathers
(tabuller@jordi)~$ cp ski.places/plans/feathers feathers
cp: cannot open 'ski.places/plans/feathers' for reading: Permission denied
(tabuller@jordi)~$ chmod u+r ski.places/plans/feathers | chmod u-x ski.places/plans/play
(tabuller@jordi)~$ cd ski.places/plans/play
cd: permission denied: ski.places/plans/play
(tabuller@jordi)~$ chmod u+x ski.places/plans/play
```

Рис. 3.8: Махинации с правами доступа файлов и каталогов

3.5 Руководства man

После этого посмотрим на несколько руководств. Команда mount, судя по описанию, монтирует файловую систему:

```

MOUNT(8)                                     System Admini
NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]
```

Рис. 3.9: руководство к команде mount

Команда `fsck` проверяет файловую систему и репарировает ее, если были найдены ошибки.

```
FSCK(8)                                System Administration          FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem ...] [--]
    [fs-specific-options]
```

Рис. 3.10: руководство к команде `fsck`

Команда `mkfs` создает файловую систему в принципе.

```
MKFS(8)                                System Administration          MKFS(8)

NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs [options] [-t type] [fs-options] device [size]
```

Рис. 3.11: руководство к команде `mkfs`

Команда `kill` “убивает” (отключает) процессы или задачи.

```
KILL(1)                                User Commands                  KILL(1)

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [ ... ]
```

Рис. 3.12: руководство к команде `kill`

4 Выводы

Приобретены практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.