

# **Лабораторная работа №5**

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

**Панченко Денис Дмитриевич**

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	10
4	Контрольные вопросы	11

# Список иллюстраций

2.1	Файл . . . . .	5
2.2	Директория . . . . .	5
2.3	Перемещение . . . . .	5
2.4	Переименование . . . . .	5
2.5	Создание файла . . . . .	6
2.6	Создание каталога . . . . .	6
2.7	Перемещение файлов . . . . .	6
2.8	Новый каталог . . . . .	6
2.9	Права доступа . . . . .	6
2.10	Копирование файла . . . . .	7
2.11	Перемещение файла . . . . .	7
2.12	Копирование каталога . . . . .	7
2.13	Перемещение каталога . . . . .	7
2.14	Право на чтение . . . . .	7
2.15	Просмотр файла . . . . .	7
2.16	Право на чтение . . . . .	8
2.17	Право на выполнение . . . . .	8
2.18	Переход в каталог . . . . .	8
2.19	Право на чтение . . . . .	8
2.20	Описание команды . . . . .	8
2.21	Описание команды . . . . .	9
2.22	Описание команды . . . . .	9
2.23	Описание команды . . . . .	9

# 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 лабораторной работы

Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовем его `equipment` (рис. 2.1).

```
[ddpanchenko@ddpanchenko ~]$ cp /usr/include/sys/io.h ~/
[ddpanchenko@ddpanchenko ~]$ mv io.h equipment
```

Рис. 2.1: Файл

В домашнем каталоге создадим директорию `~/ski.places` (рис. 2.2).

```
[ddpanchenko@ddpanchenko ~]$ mkdir ~/ski.places
```

Рис. 2.2: Директория

Переместим файл `equipment` в каталог `~/ski.places` (рис. 2.3).

```
[ddpanchenko@ddpanchenko ~]$ mv equipment ~/ski.places
```

Рис. 2.3: Перемещение

Переименуем файл `~/ski.places/equipment` в `~/ski.places/equiplist` (рис. 2.4).

```
[ddpanchenko@ddpanchenko ~]$ mv ~/ski.places/equipment ~/ski.places/equiplist
```

Рис. 2.4: Переименование

Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.plases, назвав equiplist2 (рис. 2.5).

```
[ddpanchenko@ddpanchenko ~]$ touch abc1  
[ddpanchenko@ddpanchenko ~]$ mv abc1 ~/ski.plases/equiplist2
```

Рис. 2.5: Создание файла

Создадим каталог с именем equipment в каталоге ~/ski.plases (рис. 2.6).

```
[ddpanchenko@ddpanchenko ~]$ mkdir ~/ski.plases/equipment
```

Рис. 2.6: Создание каталога

Переместите файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment (рис. 2.7).

```
[ddpanchenko@ddpanchenko ~]$ mv ~/ski.plases/equiplist ~/ski.plases/equipment/  
[ddpanchenko@ddpanchenko ~]$ mv ~/ski.plases/equiplist2 ~/ski.plases/equipment/
```

Рис. 2.7: Перемещение файлов

Создадим и переместим каталог ~/newdir в каталог ~/ski.plases и назовем его plans (рис. 2.8).

```
[ddpanchenko@ddpanchenko ~]$ mkdir newdir  
[ddpanchenko@ddpanchenko ~]$ mv newdir ~/ski.plases/plans
```

Рис. 2.8: Новый каталог

Присваиваем определенным файлам определенные права доступа (рис. 2.9).

```
drwxr--r--. 1 ddpanchenko ddpanchenko 0 map 6 14:38 australia  
-rw-rw-r--. 1 ddpanchenko ddpanchenko 0 map 6 15:12 feathers  
-r-xr--r--. 1 ddpanchenko ddpanchenko 0 map 6 15:13 my_os  
drwx--x--x. 1 ddpanchenko ddpanchenko 0 map 6 14:47 play
```

Рис. 2.9: Права доступа

Скопируем файл ~/feathers в файл ~/file.old (рис. 2.10).

```
[ddpanchenko@ddpanchenko ~]$ mv ~/feathers ~/file.old
```

Рис. 2.10: Копирование файла

Переместим файл ~/file.old в каталог ~/play (рис. 2.11).

```
[ddpanchenko@ddpanchenko ~]$ mv ~/file.old ~/play/
```

Рис. 2.11: Перемещение файла

Скопируем каталог ~/play в каталог ~/fun (рис. 2.12).

```
[ddpanchenko@ddpanchenko ~]$ cp ~/play/ ~/fun/
```

Рис. 2.12: Копирование каталога

Переместим каталог ~/fun в каталог ~/play и назовем его games (рис. 2.13).

```
[ddpanchenko@ddpanchenko ~]$ mv ~/fun/ ~/play/games
```

Рис. 2.13: Перемещение каталога

Лишим владельца файла ~/feathers права на чтение (рис. 2.14).

```
[ddpanchenko@ddpanchenko ~]$ chmod u-r ~/feathers
```

Рис. 2.14: Право на чтение

Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat? Будет отказано в доступе (рис. 2.15).

```
[ddpanchenko@ddpanchenko ~]$ cat ~/feathers  
cat: /home/ddpanchenko/feathers: Отказано в доступе
```

Рис. 2.15: Просмотр файла

Вернем владельцу файла ~/feathers право на чтение (рис. 2.16).

```
[ddpanchenko@ddpanchenko ~]$ chmod u+r ~/feathers
```

Рис. 2.16: Право на чтение

Лишим владельца каталога ~/play права на выполнение (рис. 2.17).

```
[ddpanchenko@ddpanchenko ~]$ chmod u-x ~/play
```

Рис. 2.17: Право на выполнение

Перейдем в каталог ~/play. Что произошло? Нам отказано в доступе (рис. 2.18).

```
[ddpanchenko@ddpanchenko ~]$ cd play
bash: cd: play: Отказано в доступе
```

Рис. 2.18: Переход в каталог

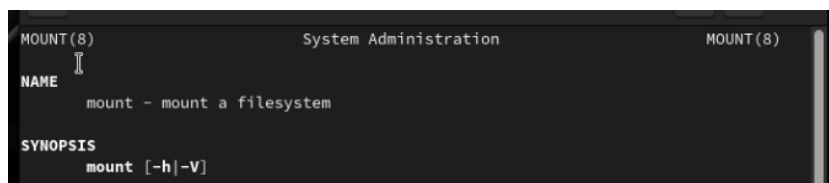
Вернем владельцу каталога ~/play право на выполнение (рис. 2.19).

```
[ddpanchenko@ddpanchenko ~]$ chmod u+x ~/play
```

Рис. 2.19: Право на чтение

Используем команду man для просмотра описания следующих команд: mount, fsck, mkfs, kill.

mount применяется для монтирования файловых систем (рис. 2.20).



```
MOUNT(8)                                System Administration                                MOUNT(8)
NAME
  mount - mount a filesystem

SYNOPSIS
  mount [-h|-V]
```

Рис. 2.20: Описание команды



fsck восстанавливает поврежденную файловую систему или проверяет ее на целостность (рис. 2.21).

```
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. 2.21: Описание команды

mkfs создает новую файловую систему (рис. 2.22).

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

NAME
    mkfs - build a Linux filesystem

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

Рис. 2.22: Описание команды

kill используется для принудительного завершения работы приложений (рис. 2.23).

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

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds
    signal] [--] pid|name...

    kill -l [number] | -L
```

Рис. 2.23: Описание команды

## 3 Вывод

Я ознакомился с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрел практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

## 4 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.

- Ext2, Ext3, Ext4 или Extended Filesystem - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений. В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзабайта.
- JFS или Journaled File System была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.
- ReiserFS - была разработана намного позже, в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она

была разработана под руководством Ганса Райзера и поддерживает только Linux. Из особенностей можно отметить динамический размер блока, что позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами. Еще одно преимущество - в возможности изменять размеры разделов на лету. Но минус в некоторой нестабильности и риске потери данных при отключении энергии. Раньше ReiserFS применялась по умолчанию в SUSE Linux, но сейчас разработчики перешли на Btrfs.

- XFS - это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы еще в 2001 году. Она изначально была рассчитана на файлы большого размера, и поддерживала диски до 2 Терабайт. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету и незначительный размер служебной информации.
- XFS - журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по умолчанию в дистрибутивах на основе Red Hat. Из недостатков - это невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находится в памяти.
- Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера на лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и

SUSE Linux.

2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.

- / — root каталог. Содержит в себе всю иерархию системы;
- /bin — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: pwd, ls, cat, ps);
- /boot — тут расположены файлы, используемые для загрузки системы (образ initrd, ядро vmlinuz);
- /dev — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;
- /etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;
- /home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;
- /lib — содержит системные библиотеки, с которыми работают программы и модули ядра;
- /lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;
- /media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;

- /mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;
- /opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);
- /proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;
- /root — директория, которая содержит файлы и личные настройки суперпользователя;
- /run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;
- /sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;
- /srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);
- /sys — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;
- /tmp — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;
- /usr — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой;
- /var — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.

3. Какая операция должна быть выполнена, чтобы содержимое некоторой

файловой системы было доступно операционной системе? Монтирование тома.

4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?
  - Один блок адресуется несколькими mode (принадлежит нескольким файлам).
  - Блок помечен как свободный, но в то же время занят (на него ссылается onode).
  - Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).
  - Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).
  - Несовпадение между размером файла и суммарным размером адресуемых inode блоков.
  - Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
  - “Потерянные” файлы (правильные inode, на которые не ссылаются записи каталогов).
  - Недопустимые или неразмещенные номера inode в записях каталогов.
5. Как создаётся файловая система? mkfs - позволяет создать файловую систему Linux.
6. Дайте характеристику командам для просмотра текстовых файлов. cat - выводит содержимое файла на стандартное устройство вывода.
7. Приведите основные возможности команды cp в Linux. cp – копирует или перемещает директорию, файлы.
8. Приведите основные возможности команды mv в Linux. mv - переименовать или переместить файл или директорию.

9. Что такое права доступа? Как они могут быть изменены? Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.