

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

Операционные системы

Дмитрий Юрьевич Дымченко

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	13
5	Контрольные вопросы	14

Список иллюстраций

3.1	Выполнение пункта 2	8
3.2	Выполнение пункта 3	9
3.3	Выполнение пункта 4.1	9
3.4	Выполнение пунктов 4.2-4.12	10
3.5	Команда mount	10
3.6	Команда fsck	11
3.7	Команда mkfs	11
3.8	Команда kill	12

Список таблиц

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 Задание

1. Выполните все примеры, приведённые в первой части описания лабораторной работы.
2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
 - 2.1. Скопируйте файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него.
 - 2.2. В домашнем каталоге создайте директорию `~/ski.places`.
 - 2.3. Переместите файл `equipment` в каталог `~/ski.places`.
 - 2.4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 - 2.5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
 - 2.6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
 - 2.7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 - 2.8. Создайте и переместите каталог `~/newdir` в каталог `~/ski.places` и назовите его `plans`.
3. Определите опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:
 - 3.1. `drwxr-r- ... australia`
 - 3.2. `drwx-x-x ... play`
 - 3.3. `-r-xr-r- ... my_os`
 - 3.4. `-rw-rw-r- ... feathers`При необходимости создайте нужные файлы.
4. Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 - 4.1. Просмотрите содержи-

- мое файла `/etc/password`. 4.2. Скопируйте файл `~/feathers` в файл `~/file.old`. 4.3. Переместите файл `~/file.old` в каталог `~/play`. 4.4. Скопируйте каталог `~/play` в каталог `~/fun`. 4.5. Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`. 4.6. Лишите владельца файла `~/feathers` права на чтение. 4.7. Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`? 4.8. Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`? 4.9. Дайте владельцу файла `~/feathers` право на чтение. 4.10. Лишите владельца каталога `~/play` права на выполнение. 4.11. Перейдите в каталог `~/play`. Что произошло? 4.12. Дайте владельцу каталога `~/play` право на выполнение.
5. Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.

3 Выполнение лабораторной работы

Скопируем файл `io.h` в домашний каталог и назовём его `equipment`. Для этого воспользуемся командой `cp` и укажем путь к нашему файлу. Выполним проверку командой `ls`. В домашнем каталоге командой `mkdir` создаём директорию `ski.places`. Перемещаем файл `equipment` в каталог `ski.places` командой `mv`. Переименуем файл `equipment`, находящийся в каталоге `ski.places` в `equiplist` с помощью команды `mv`. Создаём в домашнем каталоге файл `abc1` командой `touch` и копируем его в каталог `ski.places` под названием `equiplist2`. Создаём каталог с именем `equipment` в каталоге `ski.places`. Перемещаем файлы `equiplist` и `equiplist2` из каталога `ski.places` в подкаталог `equipment`. Создаём каталог `newdir`. Далее перемещаем его в каталог `ski.places` под названием `plans` (рис. 3.1).

```
dydihchenko@fedora ~$ sudo cp /usr/include/sys/io.h /home/dydihchenko
dydihchenko@fedora ~$ ls
abc1  io.h  may  monthly  newdir  pandoc-2.18-linux-amd64.tar.gz  pandoc-crossref-linux.tar.xz  reports  work  Видео  Документы  Зарплата  П
dydihchenko@fedora ~$ mv io.h equipment
dydihchenko@fedora ~$ ls
abc1  equipment  may  monthly  newdir  pandoc-2.18-linux-amd64.tar.gz  pandoc-crossref-linux.tar.xz  reports  work  Видео  Документы  Зарплата
dydihchenko@fedora ~$ mkdir ski.places
dydihchenko@fedora ~$ ls
abc1  equipment  may  monthly  newdir  pandoc-2.18-linux-amd64.tar.gz  pandoc-crossref-linux.tar.xz  reports  ski.places  work  Видео  Докум
dydihchenko@fedora ~$ mv equipment ski.places
dydihchenko@fedora ~$ ls ski.places
equipment
dydihchenko@fedora ~$ mv ski.places/equipment ski.places/equiplist
dydihchenko@fedora ~$ ls ski.places
equiplist
dydihchenko@fedora ~$ cp abc1 ski.places
dydihchenko@fedora ~$ mv ski.places/abc1 ski.places/equiplist2
dydihchenko@fedora ~$ cd ski.places
dydihchenko@fedora ski.places$ mkdir equipment
dydihchenko@fedora ski.places$ mv equiplist1 equiplist2 equipment
mv: не удалось выполнить stat для 'equiplist1': Нет такого файла или каталога
dydihchenko@fedora ski.places$ mv equiplist equiplist2 equipment
mv: не удалось выполнить stat для 'equiplist2': Нет такого файла или каталога
dydihchenko@fedora ski.places$ ls equipment
equiplist  equiplist2
dydihchenko@fedora ski.places$ cd
dydihchenko@fedora ~$ ls
abc1  may  monthly  newdir  pandoc-2.18-linux-amd64.tar.gz  pandoc-crossref-linux.tar.xz  reports  ski.places  work  Видео  Документы  Зарплата
dydihchenko@fedora ~$ ls newdir
dydihchenko@fedora ~$ mv newdir ski.places
dydihchenko@fedora ~$ cd ski.places
dydihchenko@fedora ski.places$ mv newdir plans
```

Рис. 3.1: Выполнение пункта 2

Создаем каталоги `australia` и `play` и файлы `my_os` и `feathers`. Определяем необходимые опции команды `chmod` и присваиваем файлам и каталогам права, указанные в методической разработке (рис. 3.2).


```
[dydihmchenko@fedora ~]$ ls
abc1    feathers    monthly    pandoc-2.18-linux-amd64.tar.gz  play    ski.places  Видео    Зарядки    Музыка    'Рабочий cron'
australia  may        my_os      pandoc-crossref-linux.tar.xz    reports  work        Документы  Изображения  Общедоступные  Шаблоны

[dydihmchenko@fedora ~]$ rm my_os feathers
rm: невозможно удалить 'my_os': это каталог
rm: невозможно удалить 'feathers': это каталог
[dydihmchenko@fedora ~]$ rm -r my_os feathers
[dydihmchenko@fedora ~]$ touch my_os feathers
[dydihmchenko@fedora ~]$ ls
abc1    feathers    monthly    pandoc-2.18-linux-amd64.tar.gz  play    ski.places  Видео    Зарядки    Музыка    'Рабочий cron'
australia  may        my_os      pandoc-crossref-linux.tar.xz    reports  work        Документы  Изображения  Общедоступные  Шаблоны

[dydihmchenko@fedora ~]$ chmod 744 australia
[dydihmchenko@fedora ~]$ chmod 711 play
[dydihmchenko@fedora ~]$ chmod 544 my_os
[dydihmchenko@fedora ~]$ chmod 664 feathers
[dydihmchenko@fedora ~]$ ls -l
total 23148
-rw-rw-r--. 1 dydihmchenko dydihmchenko 0 map 10 01:21 abc1
drwxr--r--. 1 dydihmchenko dydihmchenko 0 map 10 01:30 australia
-rw-rw-r--. 1 dydihmchenko dydihmchenko 0 map 10 01:32 feathers
-rw-rw-r--. 1 dydihmchenko dydihmchenko 0 map 10 01:17 may
drwx--x--x. 1 dydihmchenko dydihmchenko 0 map 10 01:20 monthly
-rwxr--r--. 1 dydihmchenko dydihmchenko 0 map 10 01:32 my_os
-rw-rw-r--. 1 dydihmchenko dydihmchenko 16713899 anp 4 2022 pandoc-2.18-linux-amd64.tar.gz
-rw-rw-r--. 1 dydihmchenko dydihmchenko 6984764 мая 21 2022 pandoc-crossref-linux.tar.xz
drwx--x--x. 1 dydihmchenko dydihmchenko 0 map 10 01:30 play
drwxr-xr-x. 1 dydihmchenko dydihmchenko 14 map 10 01:17 reports
drwxr-xr-x. 1 dydihmchenko dydihmchenko 28 map 10 01:29 ski.places
```

Рис. 3.2: Выполнение пункта 3

Посмотрим содержимое файла passwd командой cat (рис. 3.3).

```
[dydihmchenko@fedora ~]$ 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
```

Рис. 3.3: Выполнение пункта 4.1

Скопируем файл feathers в file.old. Переместим файл file.old в каталог play. Скопируем каталог play в каталог fun. Переместим каталог fun в каталог play и назовём его games. Лишаем владельца файла права на чтение. При попытке просмотреть файл мы получаем отказ в доступе, такой же отказ мы получаем при попытке скопировать этот файл. В конце возвращаем владельцу файла право на чтение. Лишаем владельца каталога play права на выполнение. При попытке перейти в этот каталог мы получаем отказ в доступе. Возвращаем владельцу каталога право на выполнение (рис. 3.4).

```

[dydimchenko@fedora ~]$ cp feathers ~/file.old
[dydimchenko@fedora ~]$ ls ~/file.old
/home/dydimchenko/file.old
[dydimchenko@fedora ~]$ ls file.old
file.old
[dydimchenko@fedora ~]$ cp feathers file.old
[dydimchenko@fedora ~]$ ls file.old
file.old
[dydimchenko@fedora ~]$ mv file.old play
[dydimchenko@fedora ~]$ ls play
file.old
[dydimchenko@fedora ~]$ cp play fun
cp: не указан -r; пропускается каталог 'play'
[dydimchenko@fedora ~]$ cp -r play fun
[dydimchenko@fedora ~]$ ls fun
file.old
[dydimchenko@fedora ~]$ mv fun play
[dydimchenko@fedora ~]$ cd play
[dydimchenko@fedora play]$ ls
file.old fun
[dydimchenko@fedora play]$ mv fun games
[dydimchenko@fedora play]$ ls
file.old games
[dydimchenko@fedora play]$ cd
[dydimchenko@fedora ~]$ ls
abc1 feathers monthly pandoc-2.18-linux-amd64.tar.gz play ski_places Бегеи Загрузки Музыка 'Рабочий стол'
australis may my_os pandoc-crossref-linux.tar.xz reports work Документы Избранное Общедоступные Вибланы
[dydimchenko@fedora ~]$ chmod g-r feathers
[dydimchenko@fedora ~]$ chmod g+r feathers
[dydimchenko@fedora ~]$ cat feathers
[dydimchenko@fedora ~]$ cp feathers may
[dydimchenko@fedora ~]$ ls may
may
[dydimchenko@fedora ~]$ chmod g+r feathers
[dydimchenko@fedora ~]$ chmod g+r feathers
[dydimchenko@fedora ~]$ chmod u-x play
[dydimchenko@fedora ~]$ cd play
bash: cd: play: Отказано в доступе
[dydimchenko@fedora ~]$ chmod u+x play

```

Рис. 3.4: Выполнение пунктов 4.2-4.12

Прочитаем с помощью команды `man` следующие команды: `mount`, `fsck`, `mkfs`, `kill`. Кратко охарактеризуем эти команды. Команда `mount` используется для просмотра используемых в операционной системе файловых систем (рис. 3.5).

```

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

SYNOPSIS
  mount [-h -V]

  mount [-t fstype] [-o options] [-i fstype] [-o options]
  mount [-fstype] [-o options] device mountpoint
  mount [-fstype] [-t fstype] [-o options] device mountpoint
  mount --bind|--bind|--rbind|--rbind olddir newdir
  mount --make-shared|--private|--bindable|--shared|--private|--bindable mountpoint

DESCRIPTION
  All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files can be spread out over several devices. The mount command serves to attach the filesystem found on some device to the big file tree. Generally, the mount(8) command will detach it again. The filesystem is used to control how data is stored on the device or provided in a virtual way by network or other services.

  The standard form of the mount command is:
  mount -t type device dir

```

Рис. 3.5: Команда mount

С помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы (рис. 3.6).

```
NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-haviynp] [-w [fs]] [-k [fs]] [-t fstype] [filesystem...] [-=] [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystem can be a device name (e.g., /dev/hda1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=da39a3db-5442-496b-af50-940000000000 or label=fs). Normally, the fsck program will try to handle filesystems on different physical disk drives in parallel to reduce the total amount of time needed to check all of them. If no filesystems are specified on the command line, and the -k option is not specified, fsck will default to checking filesystems to ext3/ext4 serially. This is equivalent to the -As options.

    The exit status returned by fsck is the sum of the following conditions:

    0      No errors
    1      Filesystem errors corrected
    2      System should be rebooted
    4      Filesystem errors left uncorrected
    8      Operational error
    16     Usage or syntax error
```

Рис. 3.6: Команда fsck

mkfs используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента filesys для файловой системы может выступать или название устройства (например, /dev/hda1, /dev/sdb2) или точка монтирования (например, /, /usr, /home) (рис. 3.7).

```
NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs (options) [-t type] [filesystem] device [size]

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific mkfs.-types utilities.

    mkfs is used to build a Linux filesystem on a device, usually a hard disk partition. The device argument is either the device name (e.g., /dev/hda1, /dev/sdb2), or a regular file that shall contain the filesystem. The size argument is the number of blocks to be used for the filesystem.

    The exit status returned by mkfs is 0 on success and 1 on failure.

    In actuality, mkfs is simply a front-end for the various filesystem builders (mkfs.-types) available under Linux. The filesystem-specific builder is searched for via your PATH environment setting only. Please see the filesystem-specific builder manual pages for further details.

OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2) is used.

    filesystem
        Filesystem-specific options to be passed to the real filesystem builder.

    -v, --verbose
        Produce verbose output, including all filesystem-specific commands that are executed. Specifying this option more than once inhibits execution of any filesystem-specific commands. This is really only useful for testing.

    -h, --help
        Display help text and exit.

    -V, --version
        Print version and exit. (mkfs will display version information only when it is the only parameter, otherwise it will work in conjunction.)
```

Рис. 3.7: Команда mkfs

Утилита kill отправляет сигнал процессу(-ам), указанному с помощью каждого из операндов идентификатор_процесса. По умолчанию утилита kill отправляет сигнал SIGTERM, но эту настройку по умолчанию можно переопределить путем определения имени сигнала для отправки (рис. 3.8).

```
kill(1)                                User Commands                                kill(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal=- signal+] [-q value] [-t timeout milliseconds signal] [--] pidname...
    kill -l [number] | -h

DESCRIPTION
    The command kill sends the specified signal to the specified processes or process groups.

    If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used, be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

    Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all, --pid, and --pgrp options, and the possibility to specify processes by command name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is still performed.

ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.

    Each pid can be expressed in one of the following ways:

    0
        where 0 is larger than 0. The process with PID 0 is signaled.

    0
        All processes in the current process group are signaled.
```

Рис. 3.8: Команда kill

4 Выводы

В ходе выполнения лабораторной работы мы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрели практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

5 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 экзбайта. 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. Со-

держит 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? Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:
 - Один блок адресуется несколькими `inode` (принадлежит нескольким файлам).
 - Блок помечен как свободный, но в то же время занят (на него ссылается `inode`).
 - 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 неразме-

щенные 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. `Ср` – 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.