

Отчёт по лабораторной работе №5

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

Аскеров Александр Эдуардович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
3.1	Контрольные вопросы	18
4	Выводы	24

Список иллюстраций

3.1	Копирование файла в текущем каталоге	8
3.2	Копирование нескольких файлов в каталог	8
3.3	Копирование файлов в произвольном каталоге	9
3.4	Копирование каталогов в текущем каталоге	9
3.5	Копирование каталогов в произвольном каталоге	9
3.6	Переименование файлов в текущем каталоге	9
3.7	Перемещение файлов в другой каталог	10
3.8	Переименование каталогов в текущем каталоге	10
3.9	Перемещение каталога в другой каталог	10
3.10	Переименование каталога, не являющегося текущим	10
3.11	Различные права доступа	11
3.12	Создание файла ~/may с правом выполнения для владельца . . .	11
3.13	Лишение владельца файла ~/may права на выполнение	11
3.14	Создание каталога monthly с запретом на чтение для членов группы и всех остальных пользователей	11
3.15	Файл ~/abc1 с правом записи для членов группы	12
3.16	Просмотр используемой в операционной системе файловой системы	12
3.17	Просмотр объёма свободного пространства на файловой системе	13
3.18	Копирование и переименование файла /usr/include/sys/io.h	13
3.19	В домашнем каталоге создаём директорию ~/ski.places	13
3.20	Перемещение файла equipment в каталог ~/ski.places	13
3.21	Переименование файла ~/ski.places/equipment в ~/ski.places/equiplist	14
3.22	Копирование и переименование abc1	14
3.23	Создайте каталога с именем equipment в каталоге ~/ski.places . .	14
3.24	Перемещение файлов ~/ski.places/equiplist и equiplist2	14
3.25	Перемещение и переименование каталога ~/newdir	14
3.26	Присвоение прав доступа	15
3.27	Просмотр содержимого файла /etc/passwd	15
3.28	Копирование файла ~/feathers в файл ~/file.old	15
3.29	Перемещение файла ~/file.old в каталог ~/play	16
3.30	Копирование каталога ~/play в каталог ~/fun	16
3.31	Перемещение и переименование каталога ~/fun	16
3.32	Лишение владельца файла ~/feathers права на чтение	16
3.33	Попытка просмотреть файл ~/feathers	16
3.34	Попытка скопировать файл ~/feathers	17
3.35	Предоставление владельцу файла ~/feathers права на чтение . . .	17
3.36	Лишение владельца каталога ~/play права на выполнение	17

3.37 Попытка перейти в каталог ~/play	17
3.38 Предоставление владельцу каталога ~/play права на выполнение .	17
3.39 Команда mount	18
3.40 Команда fsck	18
3.41 Команда mksf	18
3.42 Команда kill	18

1 Цель работы

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

2 Теоретическое введение

Команда `touch` для создания файла.

Команда `cat` для просмотра содержимого небольшого файла.

Команда `less` для постраничного просмотра.

Команда `head` по умолчанию выводит первые 10 строк файла.

Команда `tail` по умолчанию выводит последние 10 строк файла.

Команда `cp` для копирования файлов и каталогов.

Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов.

Права доступа: `read`, `write`, `execute`, `-`. Тип файла (символ `(-)` обозначает файл, а символ `(d)` — каталог); `-` права для владельца файла (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует); `-` права для членов группы (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует); `-` права для всех остальных (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует).

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Режим (в формате команды) имеет следующие компоненты структуры и способ записи: `=` установить право; `-` лишить права; `+` дать право; `r` чтение; `w` запись; `x` выполнение; `u` (user) владелец файла; `g` (group) группа, к которой принадлежит владелец файла; `o` (others) все остальные. В работе с правами доступа можно использовать их цифровую запись (восьмеричное

значение) вместо символьной.

Файловая система в Linux состоит из фалов и каталогов. Каждому физическому носителю соответствует своя файловая система. Существует несколько типов файловых систем. Перечислим наиболее часто встречающиеся типы: – ext2fs (second extended filesystem); – ext3fs (third extended file system); – ext4 (fourth extended file system); – ReiserFS; – xfs; – fat (file allocation table); – ntfs (new technology file system). Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой `mount` без параметров.

В контексте команды `mount` устройство — специальный файл устройства, с помощью которого операционная система получает доступ к аппаратному устройству. Файлы устройств обычно располагаются в каталоге `/dev`, имеют сокращённые имена (например, `sdaN`, `sdbN` или `hdaN`, `hdbN`, где `N` — порядковый номер устройства, `sd` — устройства SCSI, `hd` — устройства MFM/IDE).

Точка монтирования — каталог (путь к каталогу), к которому присоединяются файлы устройств.

Другой способ определения смонтированных в операционной системе файловых систем — просмотр файла `/etc/fstab`. Сделать это можно например с помощью команды `cat`. В таком случае, в каждой строке этого файла будет указано: – имя устройство; – точка монтирования; – тип файловой системы; – опции монтирования; – специальные флаги для утилиты `dump`; – порядок проверки целостности файловой системы с помощью утилиты `fsck`.

Для определения объёма свободного пространства на файловой системе можно воспользоваться командой `df`, которая выведет на экран список всех файловых систем в соответствии с именами устройств, с указанием размера и точки монтирования.

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

3 Выполнение лабораторной работы

1. Выполним все примеры, приведённые в первой части описания лабораторной работы.

1.1. Копирование файла в текущем каталоге. Скопируем файл ~/abc1 в файл april и в файл may.

```
[aeaskerov@fedora ~]$ touch abc1  
[aeaskerov@fedora ~]$ cp abc1 april  
[aeaskerov@fedora ~]$ cp abc1 may
```

Рис. 3.1: Копирование файла в текущем каталоге

1.2. Копирование нескольких файлов в каталог. Скопируем файлы april и may в каталог monthly.

```
[aeaskerov@fedora ~]$ mkdir monthly  
[aeaskerov@fedora ~]$ cp april may monthly  
[aeaskerov@fedora ~]$
```

Рис. 3.2: Копирование нескольких файлов в каталог

1.3. Копирование файлов в произвольном каталоге. Скопируем файл monthly/may в файл с именем june.


```
[aeaskerov@fedora ~]$ cp monthly/may monthly/june
[aeaskerov@fedora ~]$ ls monthly
april  june  may
[aeaskerov@fedora ~]$
```

Рис. 3.3: Копирование файлов в произвольном каталоге

1.1. Копирование каталогов в текущем каталоге. Скопируем каталог `monthly` в каталог `monthly.00`.

```
[aeaskerov@fedora ~]$ mkdir monthly.00
[aeaskerov@fedora ~]$ cp -r monthly monthly.00
[aeaskerov@fedora ~]$
```

Рис. 3.4: Копирование каталогов в текущем каталоге

1.2. Копирование каталогов в произвольном каталоге. Скопируем каталог `monthly.00` в каталог `/tmp`.

```
[aeaskerov@fedora ~]$ cp -r monthly.00 /tmp
[aeaskerov@fedora ~]$
```

Рис. 3.5: Копирование каталогов в произвольном каталоге

1.1. Переименование файлов в текущем каталоге. Изменим название файла `april` на `july` в домашнем каталоге.

```
[aeaskerov@fedora ~]$ mv april july
```

Рис. 3.6: Переименование файлов в текущем каталоге

1.2. Перемещение файлов в другой каталог. Переместим файл `july` в каталог `monthly.00`.

```
[aeaskerov@fedora ~]$ ls monthly.00
july  monthly
[aeaskerov@fedora ~]$
```

Рис. 3.7: Перемещение файлов в другой каталог

1.3. Переименование каталогов в текущем каталоге. Переименуем каталог monthly.00 в monthly.01.

```
[aeaskerov@fedora ~]$ mv monthly.00 monthly.01
[aeaskerov@fedora ~]$
```

Рис. 3.8: Переименование каталогов в текущем каталоге

1.4. Перемещение каталога в другой каталог. Переместим каталог monthly.01 в каталог reports.

```
[aeaskerov@fedora ~]$ mkdir reports
[aeaskerov@fedora ~]$ mv monthly.01 reports
[aeaskerov@fedora ~]$
```

Рис. 3.9: Перемещение каталога в другой каталог

1.5. Переименование каталога, не являющегося текущим. Переименуем каталог reports/monthly.01 в reports/monthly.

```
[aeaskerov@fedora ~]$ mv reports/monthly.01 reports/monthly
```

Рис. 3.10: Переименование каталога, не являющегося текущим

1.1. Права доступа.

```
[aeaskerov@fedora ~]$ ls -l
итого 8
-rw-rw-r--. 1 aeaskerov aeaskerov  0 мар 11 13:50 abc1
drwxr-xr-x. 1 aeaskerov aeaskerov  8 фев 24 12:51 bin
-rw-----. 1 aeaskerov aeaskerov 2635 окт  9 12:11 keys
-rw-r--r--. 1 aeaskerov aeaskerov  596 окт  9 12:11 keys.pub
-rw-rw-r--. 1 aeaskerov aeaskerov  0 мар 11 13:50 may
drwxrwxr-x. 1 aeaskerov aeaskerov  24 мар 11 13:52 monthly
```

Рис. 3.11: Различные права доступа

1.1. Создадим файл ~/may с правом выполнения для владельца.

```
[aeaskerov@fedora ~]$ touch may
[aeaskerov@fedora ~]$ ls -l may
-rw-rw-r--. 1 aeaskerov aeaskerov 0 мар 11 14:18 may
[aeaskerov@fedora ~]$ chmod u+x may
[aeaskerov@fedora ~]$ ls -l may
-rwxrw-r--. 1 aeaskerov aeaskerov 0 мар 11 14:18 may
[aeaskerov@fedora ~]$
```

Рис. 3.12: Создание файла ~/may с правом выполнения для владельца

1.2. Лишим владельца файла ~/may права на выполнение.

```
[aeaskerov@fedora ~]$ chmod u-x may
```

Рис. 3.13: Лишение владельца файла ~/may права на выполнение

1.3. Создадим каталог monthly с запретом на чтение для членов группы и всех остальных пользователей.

```
[aeaskerov@fedora ~]$ chmod g-r monthly
[aeaskerov@fedora ~]$ chmod o-r monthly
```

Рис. 3.14: Создание каталога monthly с запретом на чтение для членов группы и всех остальных пользователей

1.4. Создадим файл ~/abc1 с правом записи для членов группы.

```
[aeaskerov@fedora ~]$ chmod g+w abc1
```

Рис. 3.15: Файл ~/abc1 с правом записи для членов группы

1.1. Посмотрим используемую в операционной системе файловую систему.

```
[aeaskerov@fedora ~]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=1048576,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=802696k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
```

Рис. 3.16: Просмотр используемой в операционной системе файловой системы

1.2. Определим объём свободного пространства на файловой системе с помощью команды `df`, которая выведет на экран список всех файловых систем в соответствии с именами устройств, с указанием размера и точки монтирования.

```
[aeaskerov@fedora ~]$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           4096             0      4096           0% /dev
tmpfs              2006732         4884    2001848         1% /dev/shm
tmpfs              802696         1388    801308         1% /run
/dev/sda2          82836480       22371600 57940672        28% /
tmpfs              2006732          16    2006716         1% /tmp
/dev/sda2          82836480       22371600 57940672        28% /home
/dev/sda1          996780         175748   752220         19% /boot
tmpfs              401344          148    401196         1% /run/user/1000
[aeaskerov@fedora ~]$
```

Рис. 3.17: Просмотр объёма свободного пространства на файловой системе

2. Выполним следующие действия, зафиксировав используемые при этом команды и результаты их выполнения.

2.1. Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовём его `equipment`.

```
[aeaskerov@fedora ~]$ cp /usr/include/sys/io.h ~/equipment
```

Рис. 3.18: Копирование и переименование файла `/usr/include/sys/io.h`

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

```
[aeaskerov@fedora ~]$ mkdir ~/ski.places
```

Рис. 3.19: В домашнем каталоге создаём директорию `~/ski.places`

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

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

Рис. 3.20: Перемещение файла `equipment` в каталог `~/ski.places`

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

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

Рис. 3.21: Переименование файла ~/ski.places/equipment в ~/ski.places/equiplist

2.5. Создадим в домашнем каталоге файл abc1 (уже создан) и скопируем его в каталог ~/ski.places, назовём его equiplist2.

```
[aeaskerov@fedora ~]$ cp abc1 ski.places/equiplist2  
[aeaskerov@fedora ~]$
```

Рис. 3.22: Копирование и переименование abc1

2.6. Создадим каталог с именем equipment в каталоге ~/ski.places.

```
[aeaskerov@fedora ~]$ mkdir ski.places/equipment
```

Рис. 3.23: Создайте каталога с именем equipment в каталоге ~/ski.places

2.7. Переместим файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment.

```
[aeaskerov@fedora ~]$ mv ski.places/equiplist ski.places/equiplist2 ski.places/equipment
```

Рис. 3.24: Перемещение файлов ~/ski.places/equiplist и equiplist2

2.8. Создадим (уже существует) и переместим каталог ~/newdir в каталог ~/ski.places и назовём его plans.

```
[aeaskerov@fedora ~]$ mv newdir ski.places/plans
```

Рис. 3.25: Перемещение и переименование каталога ~/newdir

3. Определим 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`

```
[aeaskerov@fedora ~]$ touch my_os feathers
[aeaskerov@fedora ~]$ chmod 744 australia
[aeaskerov@fedora ~]$ chmod 711 play
[aeaskerov@fedora ~]$ chmod 544 my_os
[aeaskerov@fedora ~]$ chmod 664 feathers
[aeaskerov@fedora ~]$ ls -tl
итого 8
-rw-rw-r--. 1 aeaskerov aeaskerov 0 map 11 14:34 feathers
-r-xr--r--. 1 aeaskerov aeaskerov 0 map 11 14:34 my_os
drwxr--r--. 1 aeaskerov aeaskerov 0 map 11 14:34 australia
drwx--x--x. 1 aeaskerov aeaskerov 0 map 11 14:34 play
```

Рис. 3.26: Присвоение прав доступа

4. Прделаем приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды.

4.1. Просмотрим содержимое файла `/etc/passwd`.

```
[aeaskerov@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
```

Рис. 3.27: Просмотр содержимого файла `/etc/passwd`

4.2. Скопируем файл `~/feathers` в файл `~/file.old`.

```
[aeaskerov@fedora ~]$ cp feathers file.old
```

Рис. 3.28: Копирование файла `~/feathers` в файл `~/file.old`

4.3. Переместим файл ~/file.old в каталог ~/play.

```
[aeaskerov@fedora ~]$ mv file.old play
```

Рис. 3.29: Перемещение файла ~/file.old в каталог ~/play

4.4. Скопируем каталог ~/play в каталог ~/fun.

```
[aeaskerov@fedora ~]$ cp -r play fun
```

Рис. 3.30: Копирование каталога ~/play в каталог ~/fun

4.5. Переместим каталог ~/fun в каталог ~/play и назовём его games.

```
[aeaskerov@fedora ~]$ mv fun play/games
```

Рис. 3.31: Перемещение и переименование каталога ~/fun

4.6. Лишим владельца файла ~/feathers права на чтение.

```
[aeaskerov@fedora ~]$ chmod u-r feathers
```

Рис. 3.32: Лишение владельца файла ~/feathers права на чтение

4.7. Что произойдёт, если мы попытаемся просмотреть файл ~/feathers командой cat? (Будет отказано в доступе)

```
[aeaskerov@fedora ~]$ cat feathers  
cat: feathers: Отказано в доступе
```

Рис. 3.33: Попытка просмотреть файл ~/feathers

4.8. Что произойдёт, если мы попытаемся скопировать файл ~/feathers? (Будет отказано в доступе)


```
[aeaskerov@fedora ~]$ cp feathers feathers1  
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
```

Рис. 3.34: Попытка скопировать файл ~/feathers

4.9. Дадим владельцу файла ~/feathers право на чтение.

```
[aeaskerov@fedora ~]$ chmod u+r feathers
```

Рис. 3.35: Предоставление владельцу файла ~/feathers права на чтение

4.10. Лишим владельца каталога ~/play права на выполнение.

```
[aeaskerov@fedora ~]$ chmod u-x play
```

Рис. 3.36: Лишение владельца каталога ~/play права на выполнение

4.11. Перейдём в каталог ~/play. Что произошло? (Произошёл отказ в доступе)

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

Рис. 3.37: Попытка перейти в каталог ~/play

4.12. Дадим владельцу каталога ~/play право на выполнение.

```
[aeaskerov@fedora ~]$ chmod u+x play
```

Рис. 3.38: Предоставление владельцу каталога ~/play права на выполнение

5. Прочитаем man по командам mount, fsck, mkfs, kill и кратко их охарактеризуем, приведя примеры.

Команда `mount` – позволяет просмотреть используемые в операционной системе файловые системы.

```
mount - mount a filesystem
```

Рис. 3.39: Команда `mount`

Команда `fsck` – проверка и починка файловой системы Linux.

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

Рис. 3.40: Команда `fsck`

Команда `mkfs` – построение файловой системы Linux.

```
mkfs - build a Linux filesystem
```

Рис. 3.41: Команда `mkfs`

Команда `kill` – прекращение процесса.

```
kill - terminate a process
```

Рис. 3.42: Команда `kill`

3.1 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 ещё больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В неё 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 сбоя.

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. Ещё 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 ещё 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 – невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находятся в памяти.

Btrfs или B-Tree File System – это совершенно новая файловая система, сосредоточенная на отказоустойчивости, лёгкости администрирования и восстановлении 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 многие пользователи считают файловую систему Btrfs 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?

Отсутствие синхронизации между образом файловой системы в памяти и её 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 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.

4 Выводы

Успешно проведено ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Были 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.