

Лабораторная работа №5

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

Кузнецова София Вадимовна

Содержание

Цель работы	5
Теоретическое введение	6
Команды для работы с файлами и каталогами	6
Копирование файлов и каталогов	6
Перемещение и переименование файлов и каталогов	6
Права доступа	7
Изменение прав доступа	7
Анализ файловой системы	7
Выполнение лабораторной работы	10
Выводы	17
Контрольные вопросы	18

Список иллюстраций

0.1	Файл equipment	10
0.2	Перемещение файла equipment в каталог ~/ski.places	10
0.3	Файл equiplist	10
0.4	Каталог ski.places	11
0.5	Создание каталога equipment	11
0.6	Перенос файлов equiplist и equiplist2	11
0.7	Каталог plans	11
0.8	Создание необходимых файлов и каталогов	12
0.9	Изменение прав доступа для файлов и каталогов	12
0.10	Файл passwd	12
0.11	Копирование файла ~/feathers	13
0.12	Перемещение файла ~/file.old	13
0.13	Копирование каталога ~/play	13
0.14	Каталога games	13
0.15	Изменение прав доступа на чтение к файду	14
0.16	Попытка изменить файл и скопировать его	14
0.17	Изменение прав доступа на выполнение каталогу play	14
0.18	man mount	14
0.19	man fsck	15
0.20	man mkfs	15
0.21	man kill	16

Список таблиц

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.

Теоретическое введение

Команды для работы с файлами и каталогами

Для создания текстового файла можно использовать команду `touch`. Формат команды: `touch имя-файла` Для просмотра файлов небольшого размера можно использовать команду `cat`. Формат команды: `cat имя-файла` Для просмотра файлов постранично удобнее использовать команду `less`. Формат команды: `less имя-файла` Следующие клавиши используются для управления процессом просмотра: — `Space` — переход к следующей странице, — `ENTER` — сдвиг вперёд на одну строку, — `b` — возврат на предыдущую страницу, — `h` — обращение за подсказкой, — `q` — выход из режима просмотра файла. Команда `head` выводит по умолчанию первые 10 строк файла. Формат команды: `head [-n] имя-файла`, где `n` — количество выводимых строк. Команда `tail` выводит по умолчанию 10 последних строк файла. Формат команды: `tail [-n] имя-файла`, где `n` — количество выводимых строк.

Копирование файлов и каталогов

Команда `cp` используется для копирования файлов и каталогов. Формат команды: `cp [-опции] исходный_файл целевой_файл`

Перемещение и переименование файлов и каталогов

Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов. Формат команды `mv`: `mv [-опции] старый_файл новый_файл`

Права доступа

Каждый файл или каталог имеет права доступа. В сведениях о файле или каталоге указываются: – 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).

Изменение прав доступа

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Формат команды: `chmod режим имя_файла` 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

Анализ файловой системы

Файловая система в Linux состоит из файлов 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

результате её применения можно получить примерно следующее: `mount proc on /proc type proc (rw) sysfs on /sys type sysfs (rw,nosuid,nodev,noexec) udev on /dev type tmpfs (rw,nosuid) devpts on /dev/pts type devpts (rw,nosuid,noexec) /dev/sda1 on /mnt/a type ext3 (rw,noatime) /dev/sdb2 on /mnt/docs type reiserfs (rw,noatime) shm on /dev/shm type tmpfs (rw,noexec,nosuid,nodev) usbfs on /proc/bus/usb type usbfs (rw,noexec,nosuid,devmode=0664,devgid=85) binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev) nfsd on /proc/fs/nfs type nfsd (rw,noexec,nosuid,nodev)` В данном случае указаны имена устройств, названия соответствующих им точек монтирования (путь), тип файловой системы и параметрами монтирования. 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`: `cat /etc/fstab /dev/hda1 / ext2 defaults 1 1 /dev/hda5 /home ext2 defaults 1 2 /dev/hda6 swap swap defaults 0 0 /dev/hdc /mnt/cdrom auto umask=0,user,noauto,ro,exec,users 0 0 none /mnt/floppy supermount dev=/dev/fd0,fs=ext2:vfat,-, sync,umask=0 0 0 none /proc proc defaults 0 0 none /dev/pts devpts mode=0622 0 0` В каждой строке этого файла 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. Например: `df Filesystem 1024-blocks Used Available Capacity Mounted on /dev/hda3 297635 169499 112764 60% / C`

помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы. Формат команды: `fsck имя_устройства` Пример: `fsck /dev/sda1`

Выполнение лабораторной работы

Копируем файл `/usr/include/sys/io.h` в домашний каталог и назовём его `equipment`, используя команду `cp`.

```
[svkuznecova@fedora ~]$ cp /usr/include/sys/io.h ~/equipment
[svkuznecova@fedora ~]$ ls
abc1  id_rsa.pub  monthly  -o.pub  reports  Видео  Загрузки  Музыка  'Рабочий стол'
equipment  may  -o  os-intro  work  Документы  Изображения  Общедоступные  Шаблоны
[svkuznecova@fedora ~]$
```

Рис. 0.1: Файл `equipment`

Создаём каталог `~/ski.places` и переместив в созданную директорию файл `equipment`.

```
[svkuznecova@fedora ~]$ mkdir ski.places
[svkuznecova@fedora ~]$ ls
abc1  id_rsa.pub  monthly  -o.pub  reports  work  Документы  Изображения  Общедоступные  Шаблоны
equipment  may  -o  os-intro  ski.places  Видео  Загрузки  Музыка  'Рабочий стол'
[svkuznecova@fedora ~]$ mv equipment ski.places
[svkuznecova@fedora ~]$ ls ski.places
equipment
[svkuznecova@fedora ~]$
```

Рис. 0.2: Перемещение файла `equipment` в каталог `~/ski.places`

Переименуем файл `equipment` в `equiulist` (команда `mv`).

```
[svkuznecova@fedora ~]$ mv ski.places/equipment ski.places/equiulist
[svkuznecova@fedora ~]$ ls ski.places
equiulist
[svkuznecova@fedora ~]$
```

Рис. 0.3: Файл `equiulist`

Создаём файл `abc1` (команда `touch`) и копируем его в каталог `ski.places`, назовём `equiulist2`.

```
[svkuznecova@fedora ~]$ touch abc1
[svkuznecova@fedora ~]$ cp abc1 ski.plases/equiplist2
[svkuznecova@fedora ~]$ ls ski.plases
equiplist  equiplist2
[svkuznecova@fedora ~]$
```

Рис. 0.4: Каталог ski.plases

Создаём каталог с именем equipment в каталоге ski.plases и переместим `~/ski.plases/equiplist` и `equiplist2`.

```
[svkuznecova@fedora ~]$ mkdir ski.plases/equipment
[svkuznecova@fedora ~]$ ls ski.plases
equiplist  equiplist2  equipment
[svkuznecova@fedora ~]$
```

Рис. 0.5: Создание каталога equipment

```
[svkuznecova@fedora ~]$ mv ski.plases/equiplist ski.plases/equiplist2 ski.plases/equipment
[svkuznecova@fedora ~]$ ls ski.plases
equipment
[svkuznecova@fedora ~]$ ls ski.plases/ equipment
ls: невозможно получить доступ к 'equipment': Нет такого файла или каталога
ski.plases/:
equipment
[svkuznecova@fedora ~]$ ls ski.plases/equipment
equiplist  equiplist2
[svkuznecova@fedora ~]$
```

Рис. 0.6: Перенос файлов equiplist и equiplist2

Создаём новый каталог newdir и переименуем в plans.

```
[svkuznecova@fedora ~]$ mkdir newdir
[svkuznecova@fedora ~]$ ls
abc1      monthly  -o.pub   ski.plases  Документы  Музыка  Шаблоны
id_rsa.pub newdir   os-intro work        Загрузки  Общедоступные
may       -o       reports  Видео       Изображения 'Рабочий стол'
[svkuznecova@fedora ~]$ mv newdir ski.plases/plans
[svkuznecova@fedora ~]$ ls ski.plases
equipment  plans
[svkuznecova@fedora ~]$
```

Рис. 0.7: Каталог plans

Создаём файлы `my_os` и `feathers` и каталоги `australia` и `play`, присвоим им следующие права доступа используя команду `chmod`.

```
[svkuznecova@fedora ~]$ mkdir australia play
[svkuznecova@fedora ~]$ touch my_os feathers
[svkuznecova@fedora ~]$ ls
abcl id_rsa.pub my_os os-intro ski.plases документы Музыка Шаблоны
australia may -o play work Загрузки Общедоступные
feathers monthly -o.pub reports Видео Изображения 'Рабочий стол'
```

Рис. 0.8: Создание необходимых файлов и каталогов

```
[svkuznecova@fedora ~]$ chmod 744 australia
[svkuznecova@fedora ~]$ chmod 711 play
[svkuznecova@fedora ~]$ chmod 544 my_os
[svkuznecova@fedora ~]$ chmod 644 feathers
[svkuznecova@fedora ~]$ ls -lt
итого 12
-rw-r--r--. 1 svkuznecova svkuznecova 0 map 8 02:36 feathers
-r-xr--r--. 1 svkuznecova svkuznecova 0 map 8 02:36 my_os
drwxr--r--. 1 svkuznecova svkuznecova 0 map 8 02:35 australia
drwx--x--x. 1 svkuznecova svkuznecova 0 map 8 02:35 play
```

Рис. 0.9: Изменение прав доступа для файлов и каталогов

Посмотрим содержимое файла `/etc/passwd`.

```
[svkuznecova@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
```

Рис. 0.10: Файл passwd

Копируем файл `~/feathers` в файл `~/file.old`.

```
[svkuznecova@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
```

Рис. 0.11: Копирование файла ~/feathers

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

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

Рис. 0.12: Перемещение файла ~/file.old

Копируем каталог ~/play в каталог ~/fun.

```
[svkuznecova@fedora ~]$ mkdir fun
[svkuznecova@fedora ~]$ cp -r play fun
[svkuznecova@fedora ~]$ ls fun
play
[svkuznecova@fedora ~]$
```

Рис. 0.13: Копирование каталога ~/play

Переместим каталог fun и назовём его games.

```
[svkuznecova@fedora ~]$ mv fun play/games
[svkuznecova@fedora ~]$ ls play
file.old games
[svkuznecova@fedora ~]$
```

Рис. 0.14: Каталога games

Лишим владельца файла ~/feathers права на чтение, попробуем просмотреть его и скопировать. В результате без прав на чтение владелец файла не сможет просмотреть и скопировать этот файл.

```
[svkuznecova@fedora ~]$ chmod u-r feathers
[svkuznecova@fedora ~]$ ls -lt
итого 12
drwx--x--x. 1 svkuznecova svkuznecova 26 map 8 02:46 play
--w-r--r--. 1 svkuznecova svkuznecova  0 map 8 02:36 feathers
```

Рис. 0.15: Изменение прав доступа на чтение к файлу

```
[svkuznecova@fedora ~]$ cat feathers
cat: feathers: Отказано в доступе
[svkuznecova@fedora ~]$ cp feathers feathers2
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
[svkuznecova@fedora ~]$
```

Рис. 0.16: Попытка изменить файл и скопировать его

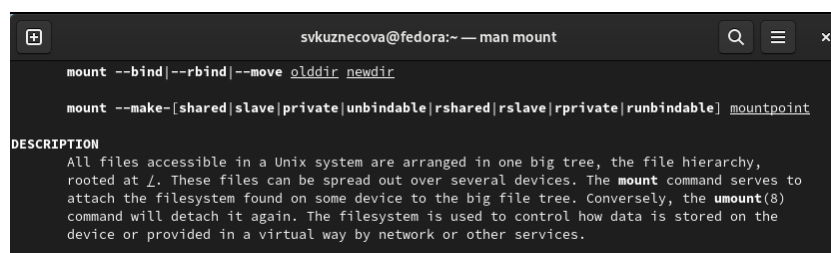
Лишаем владельца каталога `~/play` права на выполнение и попробуем перейти в него. Без прав на выполнение владелец не сможет перейти в каталог. Далее восстановим владельцу права на выполнение.

```
[svkuznecova@fedora ~]$ chmod u-x play
[svkuznecova@fedora ~]$ cd play
bash: cd: play: Отказано в доступе
[svkuznecova@fedora ~]$ chmod u+x play
[svkuznecova@fedora ~]$
```

Рис. 0.17: Изменение прав доступа на выполнение каталогу play

Используя команду `man` посмотрим описание команд `mount`, `fsck`, `mkfs`, `kill`.

Команда `mount` используется для просмотра используемых в операционной системе файловых систем.



```
svkuznecova@fedora:~ — man mount

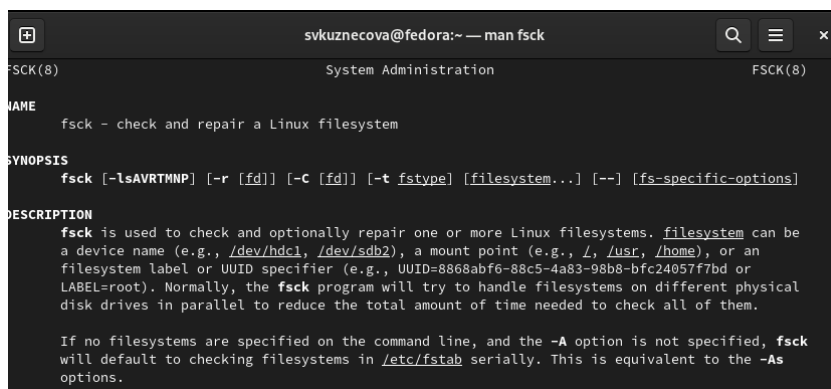
mount --bind|--rbind|--move olddir newdir

mount --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable] 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. Conversely, the umount(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.
```

Рис. 0.18: man mount

Команда `fsck` нужна для проверки и восстановления файловой системы.



```
svkuznecova@fedora:~ — man fsck
FSCK(8)                                System Administration                                FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

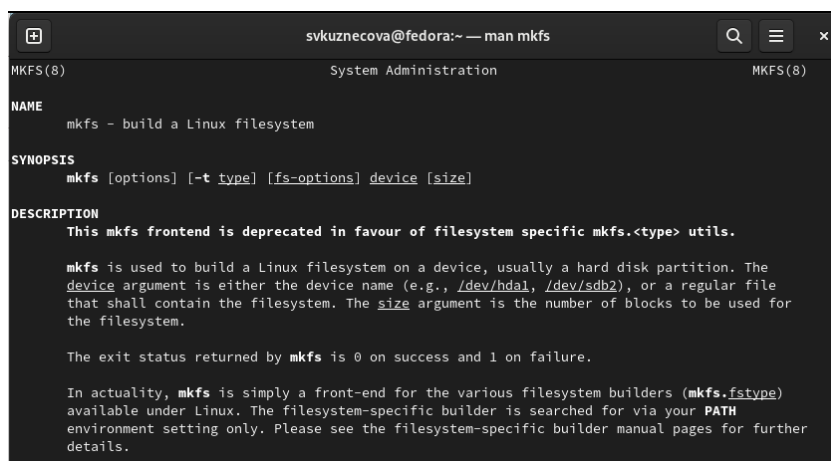
SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystem can be a device name (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). 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 -A option is not specified, fsck will default to checking filesystems in /etc/fstab serially. This is equivalent to the -As options.
```

Рис. 0.19: `man fsck`

Команда `mkfs` используется для создания файловой системы Linux на устройстве, обычно в разделе диска.



```
svkuznecova@fedora:~ — man mkfs
MKFS(8)                                System Administration                                MKFS(8)

NAME
    mkfs - build a Linux filesystem

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

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific mkfs.<type> utils.

    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.fstype) 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.
```

Рис. 0.20: `man mkfs`

Команда `kill` отправляет сигнал процессу, указанному с помощью каждого из операндов идентификатора процесса.

```
svkuznecova@fedora:~ — man kill
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

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 --queue 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.
```

Рис. 0.21: man kill

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.

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 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 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?

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 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.