

Отчет по второй лабораторной работе №6

Тулеева Валерия, НБИбд-01-20

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра прикладной
информатики и теории вероятностей

1 Анализ файловой системы Linux.

Команды для работы с файлами и каталогами

Оглавление:

1. Введение:
 - a) Цель работы
2. Описание результатов выполнения задания;
3. Вывод;
4. Контрольные вопросы.

2 Введение:

Команда для создания текстовых файлов.

Для создания текстового файла удобно воспользоваться командой touch.

Команды просмотра текстовых файлов.

Для просмотра небольших файлов удобно пользоваться командой cat.

Для просмотра больших файлов используйте команду less — она позволяет осуществлять постраничный просмотр файлов (длина страницы соответствует размеру экрана).

Команда tail выводит несколько (по умолчанию 10) последних строк файла.

Копирование файлов и каталогов осуществляется при помощи команды cp.

Команды mv и mvdir предназначены для перемещения и переименования файлов и каталогов.

3 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.

4 Описание результатов выполнения

Задания:

- Выполнила все примеры, приведенные в описании лабораторной работы (рис 1.1):

Один из примеров:

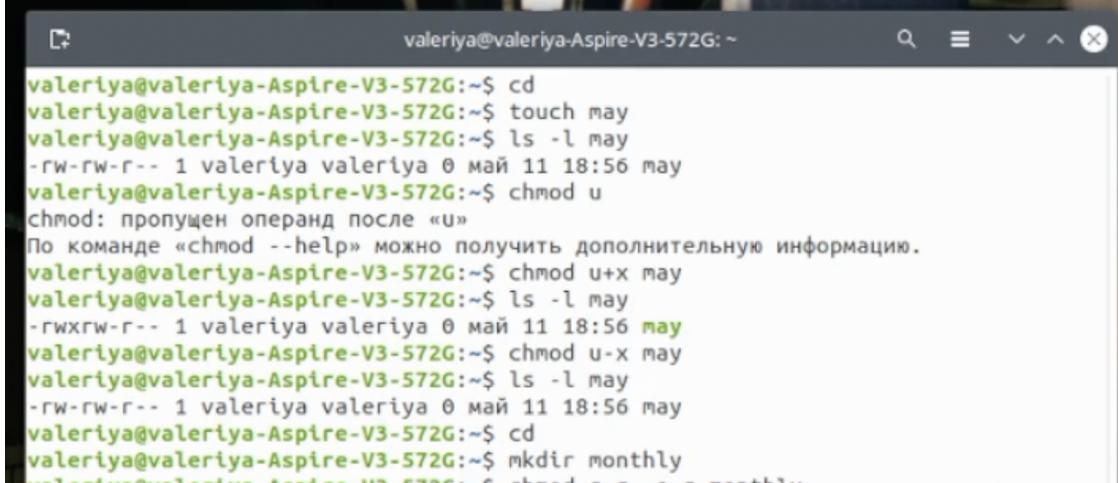
- cd

```
touch may
```

```
ls -l may
```

```
chmod u+x may
```

```
ls -l may
```



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "valeriya@valeriya-Aspire-V3-572G: ~". The window contains the following command-line session:

```
valeriya@valeriya-Aspire-V3-572G:~$ cd  
valeriya@valeriya-Aspire-V3-572G:~$ touch may  
valeriya@valeriya-Aspire-V3-572G:~$ ls -l may  
-rw-rw-r-- 1 valeriya valeriya 0 май 11 18:56 may  
valeriya@valeriya-Aspire-V3-572G:~$ chmod u  
chmod: пропущен операнд после «u»  
По команде «chmod --help» можно получить дополнительную информацию.  
valeriya@valeriya-Aspire-V3-572G:~$ chmod u+x may  

```

Рис 1.1. Выполнение примеров

2. Скопировала /usr/include/ в домашний каталог и назвала его equipment (рис 2.1):

```
cd /usr/include
```

```
sudo cp signal.h /home
```

```
valeriya@valeriya-Aspire-V3-572G:/usr/include$ sudo cp signal.h /home  
[sudo] пароль для valeriya:  
valeriya@valeriya-Aspire-V3-572G:/usr/include$ cd /home  
valeriya@valeriya-Aspire-V3-572G:/home$ ls  
signal.h valeriya  
valeriya@valeriya-Aspire-V3-572G:/home$
```

Рис 2.1. Копирование и переименование

3. В домашнем каталоге создала директорию ski.plases: (рис 3.1)

```
sudo mkdir ski.plases
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo mkdir ski.plases
```

Рис 3.1. Создание директории ski.plases

4. Переместила файл equipment в каталог ski.plases (рис 4.1):

```
cd /home
```

```
sudo mv equipment ski.plases
```

```
valeriya@valeriya-Aspire-V3-572G:~$ cd /home  
valeriya@valeriya-Aspire-V3-572G:/home$ sudo mv equipment ski.plases  
valeriya@valeriya-Aspire-V3-572G:/home$ ls ski.plases  
equipment  
valeriya@valeriya-Aspire-V3-572G:/home$
```

Рис 4.1. Перемещение файла в каталог

5. Переименовала файл ~/ski.plases/equipment в ~/ski.plases/equiplist: (рис 5.1)

```
sudo mv equipment equiplist
```

```
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ sudo mv equipment equiplist  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ ls  
equiplist
```

Рис 5.1. Переименовала файл

6. Создала в домашнем каталоге файл abc1 и скопировала его в каталог ~/ski.plases, назвав его equiplist2: (рис 6.1, 6.2, 6.3)

```
sudo touch abc1
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo touch abc1
```

Рис 6.1. Создала файл

```
sudo cp abc1 ski.plases
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo cp abc1 ski.plases  
valeriya@valeriya-Aspire-V3-572G:/home$ ls ski.plases  
abc1 equiplist
```

Рис 6.2. Скопировала его

```
cd ski.plases
```

```
sudo mv abc1 equiplist2
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ cd ski.plases  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ sudo mv abc1 equiplist2  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ ls  
equiplist equiplist2
```

Рис 6.3. Переименовала файл

7. Создала каталог с именем equipment в каталоге ~/ski.plases: (рис 7.1)

```
sudo mkdir equipment
```

```
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ sudo mkdir equipment  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ ls  
equiplist equiplist2 equipment
```

Рис 7.1. Создание нового каталога в другом каталоге

8. Переместила файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment: (рис 8.1)

```
sudo mv equiplist equiplist2 equipment
```

```
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ sudo mv equiplist equiplist2 equipment  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ ls equipment  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ ls equipment  
equiplist equiplist2
```

Рис 8.1. Перемещение файлов

9. Создала и переместила каталог `~/newdir` в каталог `~/ski.plases` и назвала его `plans`: (рис 9.1, 9.2, 9.3)

```
cd /home
```

```
sudo mkdir newdir
```

```
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ cd /home  
valeriya@valeriya-Aspire-V3-572G:/home$ sudo mkdir newdir
```

Рис 9.1. Создание каталога

```
sudo mv newdir ski.plases
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo mv newdir ski.plases  
valeriya@valeriya-Aspire-V3-572G:/home$ ls ski.plases  
equipment newdir
```

Рис 9.2. Перемещение в другой каталог

```
cd ski.plases
```

```
sudo mv newdir plans
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ cd ski.plases  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ mv newdir plans  
mv: невозможно переместить 'newdir' в 'plans': Отказано в доступе  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ sudo mv newdir plans  
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ ls  
equipment plans
```

Рис 9.3. Переименование каталога

10. Определила опции команды chmod:

- drwxr--r-- ... australia

sudo touch australia (рис 10.1)

sudo chmod 744 australia

```
valeriya@valeriya-Aspire-V3-572G:/home/ski.plases$ cd /home  
valeriya@valeriya-Aspire-V3-572G:/home$ sudo touch australia  
valeriya@valeriya-Aspire-V3-572G:/home$ chmod 744 australia  
chmod: изменение прав доступа для 'australia': Операция не позволена  
valeriya@valeriya-Aspire-V3-572G:/home$ sudo chmod 744 australia
```

Рис 10.1. Определение опции команды chmod

- drwxr-xr-x ... play (рис 10.2)

```
sudo touch play
```

```
sudo chmod 711 play
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo touch play  
valeriya@valeriya-Aspire-V3-572G:/home$ sudo chmod 711 play
```

Рис 10.2. Определение опции команды chmod

- -r-xr--r-- ... my_os (рис 10.3)

```
sudo touch my_os
```

```
sudo chmod 154 my_os
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo touch my_os  
valeriya@valeriya-Aspire-V3-572G:/home$ sudo chmod 154 my_os
```

Рис 10.3. Определение опции команды chmod

- -rw-rw-r-- ... feathers (рис 10.4)

```
sudo touch feathers
```

```
sudo chmod 664 feathers
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo touch feathers  
valeriya@valeriya-Aspire-V3-572G:/home$ sudo chmod 664 feathers  
valeriya@valeriya-Aspire-V3-572G:/home$ cd
```

Рис 10.4. Определение опции команды chmod

11. Просмотрела содержимое файла /etc/xfce4 (рис 11.1):

```
cd /etc/xfce4
```

```
ls
```

```
valeriya@valeriya-Aspire-V3-572G:/etc$ cd xfce4  
valeriya@valeriya-Aspire-V3-572G:/etc/xfce4$ ls  
defaults.list
```

Рис 11.1. Просмотр содержимого файла

12. Скопировала файл ~/feathers в файл ~/file.old:

```
sudo touch file.old; (рис 12.1)
```

```
sudo cp feathers file.old
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo touch file.old  
valeriya@valeriya-Aspire-V3-572G:/home$ sudo cp feathers file.old
```

Рис 12.1. Скопировала файл

13. Переместила файл `~/file.old` в каталог `~/play`: (рис 13.1)

```
sudo mv feathers play
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo mv feathers play
valeriya@valeriya-Aspire-V3-572G:/home$ ls play
feathers
```

Рис 13.1. Перемещение файла в каталог

14. Скопировала каталог `~/play` в каталог `~/fun` (рис 14.1):

```
sudo mkdir fun
```

```
sudo cp -r play fun
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo mkdir fun
valeriya@valeriya-Aspire-V3-572G:/home$ cp -r play fun
cp: невозможно создать каталог 'fun/play': Отказано в доступе
valeriya@valeriya-Aspire-V3-572G:/home$ sudo cp -r play fun
valeriya@valeriya-Aspire-V3-572G:/home$ ls fun
play
```

Рис 14.1. Копирование каталога в каталог

15. Переместила каталог `~/fun` в каталог `~/play` и назвала его `games` (рис 15.1):

```
sudo mv fun play
```

```
ls play
```

```
cd play
```

```
sudo mv fun games
```

```
ls
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo mv fun play
valeriya@valeriya-Aspire-V3-572G:/home$ ls play
feathers fun
valeriya@valeriya-Aspire-V3-572G:/home$ cd play
valeriya@valeriya-Aspire-V3-572G:/home/play$ sudo mv fun games
valeriya@valeriya-Aspire-V3-572G:/home/play$ ls
feathers games
```

Рис 15.1. Перемещение каталога в каталог и переименование

16. Лишила владельца файла `~/feathers` права на чтение (рис 16.1):

```
sudo chmod -r feathers
```

```
valeriya@valeriya-Aspire-V3-572G:/home/play$ sudo chmod -r feathers
```

Рис 16.1. Лишение владельца права на чтение

17. Ввела команду cat и попыталась просмотреть файл `~/feathers`, на что система мне ответила, что у меня нет прав (рис 17.1):

```
cat feathers
```

```
valeriya@valeriya-Aspire-V3-572G:/home/play$ sudo chmod -r feathers
valeriya@valeriya-Aspire-V3-572G:/home/play$ cat feathers
cat: feathers: Отказано в доступе
```

Рис 17.1. Попытка просмотра файла

18. Даала владельцу файла `~/feathers` право на чтение (рис 18.1):

```
sudo chmod +r feathers
```

19. Лишила владельца каталога `~/play` права на выполнение (рис 19.1):

```
sudo chmod -x play
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo chmod -x play
```

Рис 19.1. Лишение права на выполнение

20. Попыталась перейти в каталог `~/play`, на что система выдала ошибку, так как нет прав на выполнение у владельца (рис 20.1):

```
cd play
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ cd play
bash: cd: play: Отказано в доступе
```

Рис 20.1. Попытка перейти в каталог без прав на выполнение

21. Даала владельцу каталога `~/play` право на выполнение (рис 18.1):

```
sudo chmod +x play
```

```
valeriya@valeriya-Aspire-V3-572G:/home$ sudo chmod +x play
valeriya@valeriya-Aspire-V3-572G:/home$ cd play
valeriya@valeriya-Aspire-V3-572G:/home/play$ man_mount
```

Рис 18.1. Получение права на выполнение

22. Использовала команду man для просмотра описания следующих команд: mount, fsck, mkfs, kill. Кратко их охарактеризовала:

man mount; (рис 12.1 и рис 12.2)

Присоединение файловой системы, расположенной на запоминающем устройстве, к основному дереву каталогов называется монтированном устройства. Монтирование устройства осуществляется командой mount.

The screenshot shows a terminal window titled 'valeriya@valeriya-Aspire-V3-572G: /home/play\$ man_mount'. The window displays the 'MOUNT(8)' manual page under the 'System Administration' category. The page is divided into sections: NAME, SYNOPSIS, and DESCRIPTION. The NAME section contains the command 'mount - mount a filesystem'. The SYNOPSIS section shows four usage examples of the 'mount' command with various options like '-l', '-h', '-V', '-a', and '-t fstype'. The DESCRIPTION section provides a detailed explanation of what the 'mount' command does, stating that it attaches a filesystem from a device to a directory in the file hierarchy. At the bottom of the terminal window, there is a status bar with the text 'Manual page mount(8) line 1 (press h for help or q to quit)'.

```
valeriya@valeriya-Aspire-V3-572G: /home/play$ man_mount
(F) valeriya@valeriya-Aspire-V3-572G: /home/play MOUNT(8)
MOUNT(8) System Administration MOUNT(8)

NAME
    mount - mount a filesystem

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

    mount -a [-fFnrvw] [-t fstype] [-o optlist]

    mount [-fnrvw] [-o options] device|dir

    mount [-fnrvw] [-t fstype] [-o options] device dir

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 sev-
    eral 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 another
    services.

Manual page mount(8) line 1 (press h for help or q to quit)
```

Рис 12.1 и 12.2. Команда man mount

man fsck; (рис 12.3 и рис 12.4)

fsck (“file system consistency check”) — команда UNIX, которая проверяет и устраивает ошибки в файловой системе.

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

Формат команды:

fsck имя_устройства

Пример:

fsck /dev/sda1

```
valeriya@valeriya-Aspire-V3-572G: /home/play$ man fsck
valeriya@valeriya-Aspire-V3-572G: /home/play
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.

    The exit code returned by fsck is the sum of the following conditions:
Manual page fsck(8) line 1 (press h for help or q to quit)
```

Рис 12.3 и 12.4. Команда man fsck

man mkfs; (рис 12.5 и рис 12.6)

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**).

```
valeriya@valeriya-Aspire-V3-572G: /home/play$ man mkfs
valeriya@valeriya-Aspire-V3-572G: /home/play
MKFS(8)                               System Administration                               MKFS(8)

NAME
    mkfs - create a new file system

SYNOPSIS
    mkfs [-t fstype] [-F] [-m reserved%] [-l label] [-v] [-c count] [-C count] [-E]
    [-L label] [-I size] [-B size] [-S size] [-D] [-G size] [-L label] [-I size] [-B size] [-S size] [-D] [-G size] [-C count] [-F] [-m reserved%] [-l label] [-v] [-c count] [-E]
    [-t fstype]
```

```
valeriya@valeriya-Aspire-V3-572G: /home/play
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 code 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 manual pages for the individual builders for details.
```

Manual page **mkfs(8)** line 1 (press **h** for help or **q** to quit)

Рис 12.5 и 12.6. Команда man mkfs

man kill; (рис 12.7 и 12.8)

Команда Kill посылает указанный сигнал указанному процессу. Если не указано ни одного сигнала, посыпается сигнал SIGTERM. Сигнал SIGTERM завершает лишь те процессы, которые не обрабатывают его приход. Для других процессов может быть необходимым послать сигнал SIGKILL, поскольку этот сигнал перехватить невозможно.



```
valeriya@valeriya-Aspire-V3-572G: /home/play
```

KILL(1) User Commands KILL(1)

NAME

kill - send a signal to a process

SYNOPSIS

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

DESCRIPTION

The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process groups; see the PGID column in ps command output. A PID of -1 is special; it indicates all processes except the kill process itself and init.

OPTIONS

<pid> [...] Send signal to every <pid> listed.

-<signal>

-s <signal>

Manual page kill(1) line 1 (press h for help or q to quit)

Рис 12.7 и 12.8. Команда man kill

5 Вывод:

В данной лабораторной работе, я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

6 Контрольные вопросы (лабораторная работа №6)

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

FAT32

Файловая система FAT32 является самой старой и опытной в истории компьютерных технологий. Ее путь начался с оригинальной 8-битной системы FAT в 1977 году, которая функционировала внутри автономного диска Microsoft Standalone Disk Basic-80. Он был запущен специально для Intel 8080 NCR 7200 в 1977/1978 году, работая терминалом ввода данных с 8-дюймовыми гибкими дисками.

Основным отличием FAT32 от ее предшественников является преодоление ограниченного объема доступной для хранения информации. 32-разрядная система была выпущена в августе 1995 года вместе с релизом Windows 95 и в своем обновленном варианте позволила увеличить верхние пределы размера файлов и хранилища данных до 4 Гб и 16 Тб.

Таким образом, FAT32 не предназначена для хранения больших объемов данных и установки тяжелых приложений. Именно по этой причине на жестких дисках используется файловая система NTFS, которая позволяет пользователям перестать думать о загружаемых объемах информации.

Резюмируя, система FAT32 идеально подойдет для хранения данных, объем которых не превышает 4 Гб, на любых съемных носителях. Ее популярность не ограничивается только компьютерной сферой. Она используется в игровых консо-

лях, телевизорах с высокой четкостью изображения, DVD-проигрывателях, Blu-Ray плеерах и любых других устройствах с USB-портом. FAT32 поддерживают все версии Windows, Linux и MacOS.

ExFAT

ExFAT (Extended FAT) — новая, расширенная файловая система от Microsoft, которая с успехом заменяет своего предшественника на поле, когда дело доходит до больших объемов информации.

Как вы наверняка знаете, большинство современных цифровых фотокамер используют систему exFAT, поскольку она существенно легче NTFS, но, в то же время, позволяет сохранять файлы размером более 4 Гб, в отличие от FAT32.

Таким образом, копируя на Flash-накопитель с файловой системой exFAT документ размером 6 Гб, вы не столкнетесь с негативными последствиями, которые можно наблюдать, используя предшествующую версию системы.

Формат exFAT набирает все большую популярность и используется преимущественно с высокоемкими картами памяти SDXC. Основной причиной тому является небольшой размер файловой системы и, как ранее описывалось, возможность сохранять документы объемом более 4 Гб.

NTFS

Файловая система NTFS имеет важную особенность, когда дело доходит до ее восстановления, вследствие каких-либо повреждений. Она содержит в себе определенную структуру данных, которая отслеживает любые изменения в системе и с помощью которой всегда можно вернуть работоспособность NTFS.

Данная файловая система поддерживается всеми версиями Windows, начиная с Windows XP. К сожалению, MacOS не разделяет стремление к совместимости, продвигаемое Microsoft. Apple оставили для пользователей возможность чтения данных с дисков NTFS, однако записывать на них не получится. Поддержка данной файловой системы от Linux ограничивается лишь несколькими ее версиями.

Теоретический размер файла NTFS — 16 Эб — 1 Кб, что составляет 18 446 744 073 709 550 502 байта. В команду разработчиков входили Том Миллер, Гарри Кимуру,

Брайан Эндрю, Девид Гебель.

Следующей версией файловой системы стала NTFS v3.1, запущенная специально для Microsoft Windows XP. В дальнейшем она не претерпевала особых изменений, хотя в нее и было внесено множество различных дополнений. Например, появилась возможность сжатия логических разделов, восстановление и символические ссылки NTFS. Кроме того начальная емкость файловой системы составляла всего 256 Мб из колоссальных 16 Эб — 1 Кб в новых версиях, запущенных с выходом Windows 8.

Говоря о полезных функциях, внедренных в NTFS v3.1, можно отметить расширение поддерживаемых форматов файлов, квоты использования диска, шифрование файлов и создание точек повторной обработки. Примечательным является тот факт, что новые версии NTFS полностью совместимы с предыдущими.

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

Файловая система имеет древовидную структуру. Вершинами (промежуточными узлами) дерева являются каталоги со ссылками на другие каталоги или файлы. Листья дерева соответствуют файлам или пустым каталогам.

В Linux файловая система организована иначе чем в Windows.

Первое, что бросается в глаза, нет дисков C, D, E и так далее.

Второе, про что часто говорят, но не совсем просто осознать — всё в Linux является файлами. То есть сами файлы, очевидно, являются файлами; директории также являются файлом особого вида; физические устройства (диски, клавиатура, мышь и т. д. — это всё тоже файлы) и даже сетевые соединения — это тоже файлы.

Третье — в Linux есть root. Имеется ввиду корень файловой системы, который обозначается как «/». Ещё в Linux есть главный пользователь, имя которого тоже root — не нужно путаться, это довольно разные вещи. Кстати, ведь ещё есть и директория с именем /root. Говоря про корень файловой системы, имеется ввиду «/», а не «/root».

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).

3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?

Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой `mount` без параметров.

4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?

Нарушение целостности файловой системы

Ядро выбирает порядок совершения операций с метаданными таким образом, чтобы вред от ошибок в случае аварии был минимальным. Однако проблема нарушения этого порядка все же остается, т.к. драйвер может изменять очередность выполнения запросов для оптимизации ввода/вывода. Единственной возможностью сохранить выбранный порядок является синхронизация операций со стороны файловой подсистемы.

В нашем примере файловая подсистема будет ожидать, пока на диск не будет записано содержимое индексного дескриптора, и только после этого произведет изменения каталога.

Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:

1. Один блок адресуется несколькими mode (принадлежит нескольким файлам).

2. Блок помечен как свободный, но в то же время занят (на него ссылается onode).
3. Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).
4. Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).
5. Несовпадение между размером файла и суммарным размером адресуемых inode блоков.
6. Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
7. “Потерянные” файлы (правильные inode, на которые не ссылаются записи каталогов).
8. Недопустимые или неразмещенные номера inode в записях каталогов. Эти ошибки схематически показаны на рис.

Если нарушение все же произошло, на помощь может прийти утилита fsck(1M), производящая исправление файловой системы. Запуск этой утилиты может производиться автоматически каждый раз при запуске системы, или администратором, с помощью команды:

```
fsck [options] filesystem
```

где filesystem — специальный файл устройства, на котором находится файловая система.

Проверка и исправление должны производиться только на размонтированной файловой системе. Это связано с необходимостью исключения синхронизации таблиц в памяти (ошибочных) с их дисковыми эквивалентами (исправленными). Исключение составляет корневая файловая система, которая не может быть размонтирована.

Для ее исправления необходимо использовать опцию обеспечивающую немедленный перезапуск системы после проведения проверки.

5. Как создаётся файловая система?

В общем виде команда выглядит так:

```
mkfs -t ext2 /dev/hda5
```

Для создания FAT32 можно воспользоваться такой командой:

```
mkfs -t vfat /dev/hda5
```

В Debian, чтобы было доступно создание файловых систем FAT и FAT32 требуется установить пакет dosfstools.

6. Дайте характеристику командам, которые позволяют просмотреть текстовые файлы.

Команды просмотра текстовых файлов.

Для просмотра небольших файлов удобно пользоваться командой cat.

Формат команды:

```
cat имя-файла
```

Для просмотра больших файлов используйте команду less — она позволяет осуществлять постраничный просмотр файлов (длина страницы соответствует размеру экрана).

Формат команды:

```
less имя-файла
```

Для просмотра начала файла можно воспользоваться командой head. По умолчанию она выводит первые 10 строк файла.

Формат команды:

```
head [-n] имя-файла,
```

где n — количество выводимых строк . Команда tail выводит несколько (по умолчанию 10) последних строк файла.

Формат команды:

```
tail [-n] имя-файла,
```

где n — количество выводимых строк.

7. Приведите основные возможности команды cp в Linux.

Копирование файлов и каталогов осуществляется при помощи команды cp.

Формат команды:

cp [-опции] исходный_файл целевой_файл

Примеры:

1. Копирование файла в текущем каталоге. Скопировать файл ~/abc1 в файл april и в файл may:

```
cd  
touch abc1  
cp abc1 april  
cp abc1 may
```

2. Копирование нескольких файлов в каталог. Скопировать файлы april и may в каталог monthly:

```
mkdir monthly  
cp april may monthly
```

3. Копирование файлов в произвольном каталоге. Скопировать файл monthly/may в файл с именем june:

```
cp monthly/may monthly/june  
ls monthly
```

Опция i в команде cp выведет на экран запрос подтверждения о перезаписи файла, если на место целевого файла вы поставите имя уже существующего файла. Команда cp с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами.

8. Назовите и дайте характеристику командам перемещения и переименования файлов и каталогов.

Команды mv и mvdir предназначены для перемещения и переименования файлов и каталогов.

Формат команды mv:

mv [-опции] старый_файл новый_файл

Примеры:

1. Переименование файлов в текущем каталоге. Изменить название файла april на july в домашнем каталоге:

```
cd
```

```
mv april july
```

2. Перемещение файлов в другой каталог. Переместить файл july в каталог monthly.00:

```
mv july monthly.00
```

```
ls monthly.00
```

Результат:

```
april july june may
```

Если необходим запрос подтверждения о перезаписи файла, то нужно использовать опцию i.

9. Что такое права доступа? Как они могут быть изменены?

Права доступа — совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы установленных правовыми документами или собственником, владельцем информации. Права доступа определяют набор действий, разрешённых для выполнения субъектам над объектами данных.

Каждый файл или каталог имеет права доступа.

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

г чтение

w запись

x выполнение

u (user) владелец файла

g (group) группа, к которой принадлежит владелец файла

o (others) все остальные