

Лабораторная работа №5

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

Рытов Алексей Константинович НФИбд-02-21

Список иллюстраций

3.1	Создание файла 123 и его вывод	7
3.2	Вывод файла 123 с помощью less	8
3.3	Вывод строк файла 123 с помощью head и tail	8
3.4	Копирование файла io.h в домашний каталог	9
3.5	Переименовывание файла io.h	9
3.6	Создание директории ski.plases, перемещение файла equipment и его переименовывание	9
3.7	Создание файла abc1	9
3.8	Копирование abc1 в ski.plases и его переименовывание	10
3.9	Создание каталога equipment, перемещение в него файлов, создание каталога newdir и его перемещение	10
3.10	Создание каталогов и файлов	11
3.11	Изменение прав доступа	11
3.12	Вывод файла passwd с помощью команды cat	11
3.13	Копирование файла old, его перемещение	12
3.14	Перемещение и переименовывание каталога play	12
3.15	Проверка команды chmod	12
3.16	Использование команды man	13
3.17	Вывод команды man mount	13
3.18	Вывод команды man fsck	14
3.19	Вывод команды man mkfs	14
3.20	Вывод команды man kill	15

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 Теоретическое введение

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.

Команда `cp` используется для копирования файлов и каталогов. Формат команды: `cp [-опции] исходный_файл целевой_файл` Примеры: Копирование файла в текущем каталоге. Скопировать файл `~/abc1` в файл `april` и в файл `may`: `cd touch abc1 cp abc1 april cp abc1 may` Копирование нескольких файлов в каталог. Скопировать файлы `april` и `may` в каталог `monthly`: `mkdir monthly cp april may monthly` 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. Для рекурсивного копирования каталогов, содержащих файлы, используется команда `cp` с опцией `r`. Примеры: Копирование каталогов в текущем каталоге. Скопировать каталог `monthly` в каталог `monthly.00`: `mkdir monthly.00 cp -r monthly monthly.00` Копи-

рование каталогов в произвольном каталоге. Скопировать каталог monthly.00 в каталог /tmp `cp -r monthly.00 /tmp`

Команды `mv` и `mvdir` предназначены для перемещения и переименования файлов и каталогов. Формат команды `mv`: 1 `mv [-опции] старый_файл новый_файл`

Примеры: 1. Переименование файлов в текущем каталоге. Изменить название файла `april` на `july` в домашнем каталоге: 1 `cd` 2 `mv april july` 2. Перемещение файлов в другой каталог. Переместить файл `july` в каталог `monthly.00`: 1 `mv july monthly.00` 2 `ls monthly.00` Результат: 1 `april july june may` Если необходим запрос подтверждения о перезаписи файла, то нужно использовать опцию `i`. 3. Переименование каталогов в текущем каталоге. Переименовать каталог `monthly.00` в `monthly.01` 1 `mv monthly.00 monthly.01` 4. Перемещение каталога в другой каталог. Переместить каталог `monthly.01` в каталог `reports`: 1 `mkdir reports` 2 `mv monthly.01 reports` 5. Переименование каталога, не являющегося текущим. Переименовать каталог `reports/monthly.01` в `reports/monthly`: 1 `mv reports/monthly.01 reports/monthly`

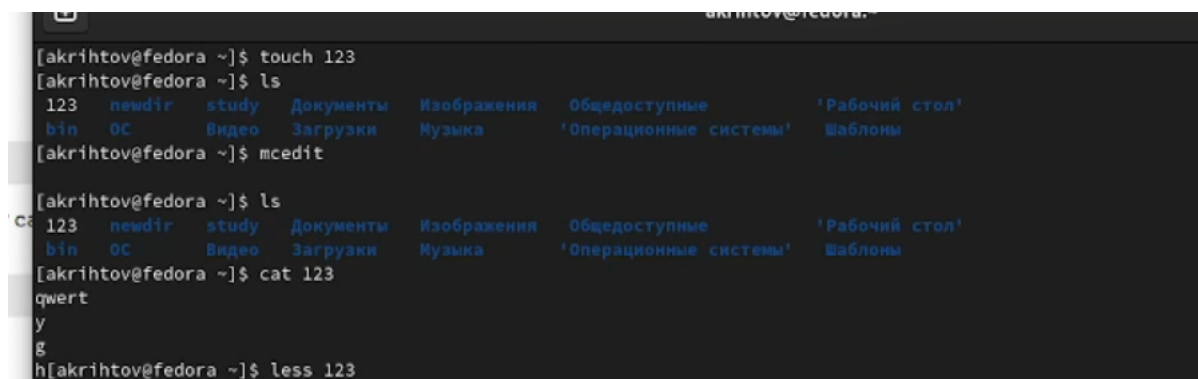
5.2.4. Права доступа Каждый файл или каталог имеет права доступа (табл. 5.1). В сведениях о файле или каталоге указываются: – тип файла (символ `(-)` обозначает файл, а символ `(d)` — каталог); – права для владельца файла (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — 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). Примеры: Кулябов Д. С. и др. Операционные системы 49

Право	Обозначение
Чтение	<code>r</code>
Разрешены просмотр и копирование	Разрешён просмотр списка входящих файлов
Запись	<code>w</code>
Разрешены изменение и переименование	Разрешены создание и удаление файлов
Выполнение	<code>x</code>
Разрешено выполнение файла (скриптов и/или программ)	Разрешён доступ в каталог и есть возможность сделать его текущим
Для файла (крайнее левое поле имеет значение <code>-</code>) владелец файла имеет право на чтение и запись (<code>rw-</code>), группа, в которую входит владелец	

файла, может читать файл (r-), все остальные могут читать файл (r-): -rw-r-r-
Только владелец файла имеет право на чтение, изменение и выполнение файла:
-rwx— Владелец каталога (крайнее левое поле имеет значение d) имеет право на просмотр, изменение и доступа в каталог, члены группы могут входить и просматривать его, все остальные — только входить в каталог: drwxr-x-x 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

3 Выполнение лабораторной работы

1. Выполнили все примеры, приведённые в первой части описания лабораторной работы.(рис. 1-3)



```
[akrihtov@fedora ~]$ touch 123
[akrihtov@fedora ~]$ ls
123  newdir  study  Документы  Изображения  Общедоступные  'Рабочий стол'
bin  OC     Видео  Загрузки  Музыка       'Операционные системы'  Шаблоны
[akrihtov@fedora ~]$ mcedit
[akrihtov@fedora ~]$ ls
123  newdir  study  Документы  Изображения  Общедоступные  'Рабочий стол'
bin  OC     Видео  Загрузки  Музыка       'Операционные системы'  Шаблоны
[akrihtov@fedora ~]$ cat 123
qwert
y
g
[akrihtov@fedora ~]$ less 123
```

Рис. 3.1: Создание файла 123 и его вывод

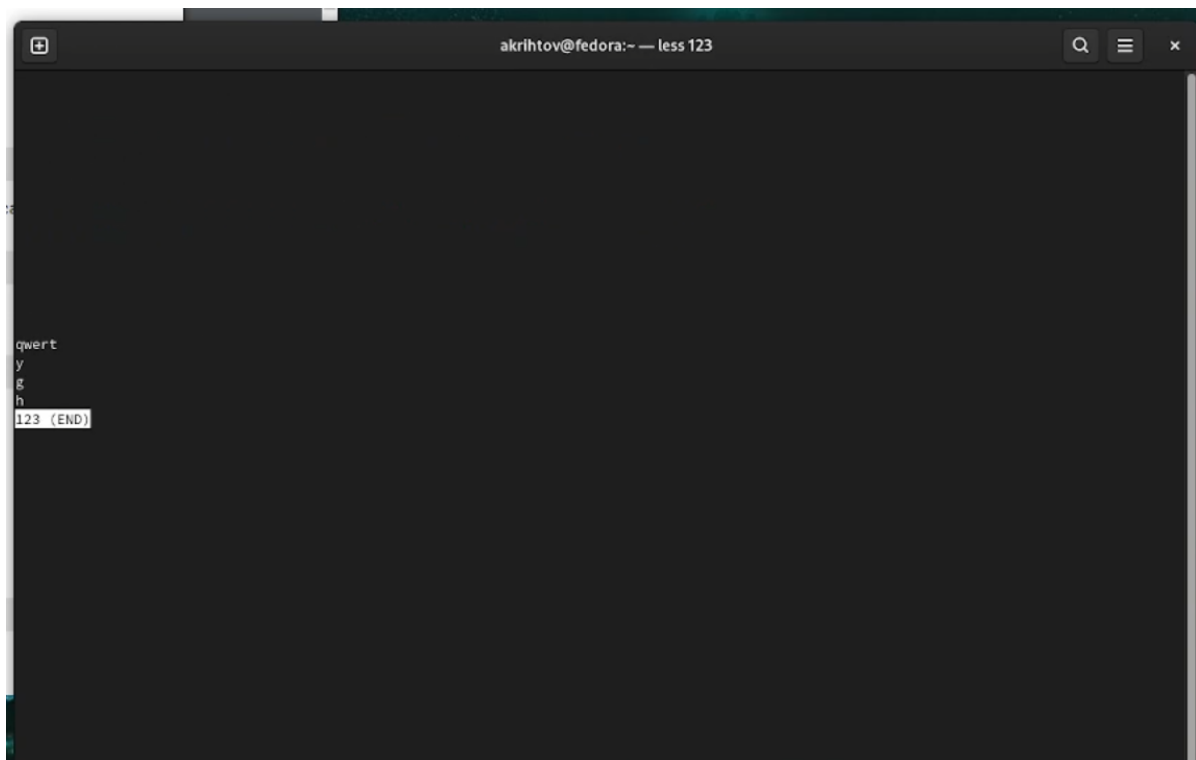


Рис. 3.2: Вывод файла 123 с помощью less

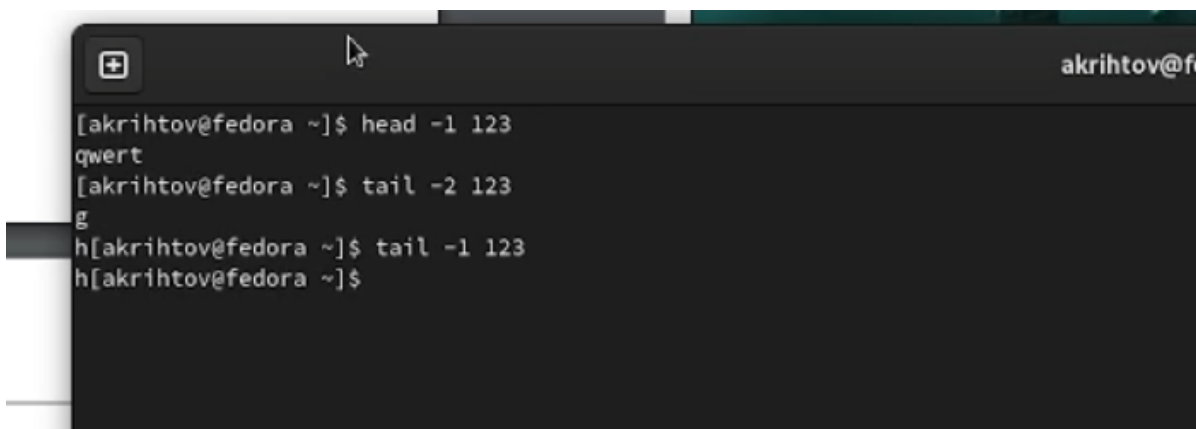


Рис. 3.3: Вывод строк файла 123 с помощью head и tail

2.1. Скопировали файл /usr/include/sys/io.h в домашний каталог и назвали его equipment.(рис. 4-5)


```
[akrihtov@fedora ~]$ cd /
[akrihtov@fedora /]$ cd usr/include/sys
[akrihtov@fedora sys]$ cp io.h /home/akrihtov
[akrihtov@fedora sys]$ cd ~
[akrihtov@fedora ~]$ ls
123  io.h  0C  Видео  Загрузки  Музыка  'Операционные системы'
bin  newdir  study  Документы  Изображения  Общедоступные  'Рабочий стол'
[akrihtov@fedora ~]$
```

Рис. 3.4: Копирование файла io.h в домашний каталог

```
[akrihtov@fedora ~]$ mv io.h equipment
```

Рис. 3.5: Переименовывание файла io.h

2.2. В домашнем каталоге создали директорию ~/ski.plases. 2.3. Переместили файл equipment в каталог ~/ski.plases. 2.4. Переименовали файл ~/ski.plases/equipment в ~/ski.plases/equiplist.(рис. 6)

```
[akrihtov@fedora ~]$ mkdir ski.plases
[akrihtov@fedora ~]$ ls
123  io.h  0C  study  Документы  Изображения  Общедоступные  'Рабочий
bin  newdir  ski.plases  Видео  Загрузки  Музыка  'Операционные системы'  Шаблоны
[akrihtov@fedora ~]$ touch equipment
[akrihtov@fedora ~]$ mv equipment ski.plases
[akrihtov@fedora ~]$ ls
123  io.h  0C  study  Документы  Изображения  Общедоступные  'Рабочий
bin  newdir  ski.plases  Видео  Загрузки  Музыка  'Операционные системы'  Шаблоны
[akrihtov@fedora ~]$ cd ski.plases
[akrihtov@fedora ski.plases]$ ls
equipment
[akrihtov@fedora ski.plases]$ cd ..
[akrihtov@fedora ~]$ cd ski.plases
[akrihtov@fedora ski.plases]$ mv equipment equiplist
[akrihtov@fedora ski.plases]$ ls
equiplist
```

Рис. 3.6: Создание директории ski.plases, перемещение файла equipment и его переименовывание

2.5. Создали в домашнем каталоге файл abc1 и скопировали его в каталог ~/ski.plases, назовите его equiplist2.(рис. 7-8)

```
[akrihtov@fedora ~]$ touch abc1
```

Рис. 3.7: Создание файла abc1

```

[akrihtov@fedora ~]$ ls
123  bin  newdir  ski.plases  Видео  Загрузки  Музыка  'Операционные системы'  Шаблоны
abc1 io.h  OC      study      Документы  Изображения  Общедоступные  'Рабочий стол'
[akrihtov@fedora ~]$ cp abc1 ski.plases
[akrihtov@fedora ~]$ mv abc1 equiplist2
[akrihtov@fedora ~]$ cd ski.plases
[akrihtov@fedora ski.plases]$ ls
abc1  equiplist
[akrihtov@fedora ski.plases]$ mv abc1 equiplist2
[akrihtov@fedora ski.plases]$ ls
equiplist  equiplist2
[akrihtov@fedora ski.plases]$

```

Рис. 3.8: Копирование abc1 в ski.plases и его переименовывание

2.6. Создали каталог с именем equipment в каталоге ~/ski.plases. 2.7. Переместили файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment. 2.8. Создали и переместили каталог ~/newdir в каталог ~/ski.plases и назвали его plans.(рис. 9)

```

[akrihtov@fedora ski.plases]$ mkdir equipment
[akrihtov@fedora ski.plases]$ ls
equiplist  equiplist2  equipment
[akrihtov@fedora ski.plases]$ mv equiplist equipment; mv equiplist2 equipment
[akrihtov@fedora ski.plases]$ cd equipment
[akrihtov@fedora equipment]$ ls
equiplist  equiplist2
[akrihtov@fedora equipment]$ cd ..
[akrihtov@fedora ski.plases]$ cd ..
[akrihtov@fedora ~]$ mkdir newdir
mkdir: невозможно создать каталог «newdir»: Файл существует
[akrihtov@fedora ~]$ rm -r newdir
[akrihtov@fedora ~]$ mkdir newdir
[akrihtov@fedora ~]$ ls
123  equiplist2  newdir  ski.plases  Видео  Загрузки  Музыка  'Операционные системы'  Шаблоны
bin  io.h      OC      study      Документы  Изображения  Общедоступные  'Рабочий стол'
[akrihtov@fedora ~]$ mv newdir ski.plases
[akrihtov@fedora ~]$ ls
123  equiplist2  OC      study  Документы  Изображения  Общедоступные  'Рабочий стол'
bin  io.h      ski.plases  Видео  Загрузки  Музыка  'Операционные системы'  Шаблоны
[akrihtov@fedora ~]$ cd ski.plases
[akrihtov@fedora ski.plases]$ ls
equipment  newdir
[akrihtov@fedora ski.plases]$ mv newdir plans
[akrihtov@fedora ski.plases]$ ls
equipment  plans
[akrihtov@fedora ski.plases]$

```

Рис. 3.9: Создание каталога equipment, перемещение в него файлов, создание каталога newdir и его перемещение

3. Определили опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа. 3.1. drwxr-r- ... australia 3.2. drwx-x-x ... play 3.3. -r-xr-r- ... my_os 3.4. -rw-rw-r- ... feathers(рис. 10-11)

```
[akrihtov@fedora ~]$ mkdir australia
[akrihtov@fedora ~]$ mkdir play
[akrihtov@fedora ~]$ touch my_os feathers
[akrihtov@fedora ~]$ ls
123      bin      feathers  my_os    play      study  Документы  Изображения  Общедоступные  'Рабочий стол'
australia equiplist2 io.h      OC       ski.places Видео      Загрузки    Музыка      'Операционные системы'  Шаблоны
[akrihtov@fedora ~]$ chmod +x
```

Рис. 3.10: Создание каталогов и файлов

```
[akrihtov@fedora ~]$ chmod 744 australia
[akrihtov@fedora ~]$ chmod 711 play
[akrihtov@fedora ~]$ chmod 544 my_os
[akrihtov@fedora ~]$ chmod 664 feathers
[akrihtov@fedora ~]$
```

Рис. 3.11: Изменение прав доступа

4.1. Просмотрели содержимое файла /etc/passwd.(рис. 12)

```
[akrihtov@fedora etc]$ cd passwd
bash: cd: passwd: Это не каталог
[akrihtov@fedora etc]$ cat 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
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:usr/sbin/nologin
```

Рис. 3.12: Вывод файла passwd с помощью команды cat

4.2. Скопировали файл ~/feathers в файл ~/file.old. 4.3. Переместили файл ~/file.old в каталог ~/play. 4.4. Скопировали каталог ~/play в каталог ~/fun.(рис. 13)

```

[akrihtov@fedora ~]$ touch file.old
[akrihtov@fedora ~]$ cp feathers file.old
[akrihtov@fedora ~]$ mkdir play
[akrihtov@fedora ~]$ mv file.old play
[akrihtov@fedora play]$ ls
file.old
[akrihtov@fedora play]$ cd ..
[akrihtov@fedora ~]$ mkdir fun
[akrihtov@fedora ~]$ cp play fun
cp: не указан -r; пропускается каталог 'play'
[akrihtov@fedora ~]$ ls
123      bin      feathers  io.h     oc      ski.places  Видео  Загрузки  Музыка  'Операционные системы'  Шаблоны
australia equiplist2 fun      my_os    play     study      Документы  Изображения  Общедоступные  'Рабочий стол'
[akrihtov@fedora ~]$ cp -r play fun
[akrihtov@fedora ~]$ cd play
[akrihtov@fedora play]$ ls
file.old
[akrihtov@fedora play]$ cd ..
[akrihtov@fedora ~]$ ls
123      bin      feathers  io.h     oc      ski.places  Видео  Загрузки  Музыка  'Операционные системы'  Шаблоны
australia equiplist2 fun      my_os    play     study      Документы  Изображения  Общедоступные  'Рабочий стол'
[akrihtov@fedora ~]$ cd fun
[akrihtov@fedora fun]$ ls
play

```

Рис. 3.13: Копирование файла old, его перемещение

4.5. Переместили каталог ~/fun в каталог ~/play и назвали его games.(рис.14)

```

[akrihtov@fedora ~]$ mv fun play
[akrihtov@fedora ~]$ ls
123      bin      feathers  my_os    play     study  Документы  Изображения  Общедоступные  'Рабочий стол'
australia equiplist2 io.h      oc      ski.places  Видео  Загрузки  Музыка  'Операционные системы'  Шаблоны
[akrihtov@fedora ~]$ cd play
[akrihtov@fedora play]$ ls
file.old fun
[akrihtov@fedora play]$ mv fun games
[akrihtov@fedora play]$ ls
file.old games

```

Рис. 3.14: Перемещение и переименовывание каталога play

4.6 - 4.8 Лишили владельца файла ~/feathers права на чтение. Попытались просмотреть файл командой cat. В итоге нав выдало ошибку. Попробовав скопировать файл ~/feathers мы также получили в консоли ошибку. 4.9. Дали владельцу файла ~/feathers право на чтение. 4.10. Лишили владельца каталога ~/play права на выполнение. 4.11. Попытались перейти в каталог ~/play. Нам вывело ошибку. 4.12. Дали владельцу каталога ~/play право на выполнение.(рис.15)

```

[akrihtov@fedora ~]$ chmod u-r feathers
[akrihtov@fedora ~]$ cat feathers
cat: feathers: Отказано в доступе
[akrihtov@fedora ~]$ cp feathers equiplist2
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
[akrihtov@fedora ~]$ chmod u+r feathers
[akrihtov@fedora ~]$ chmod u-x play
[akrihtov@fedora ~]$ cd play
bash: cd: play: Отказано в доступе
[akrihtov@fedora ~]$ chmod u+x play

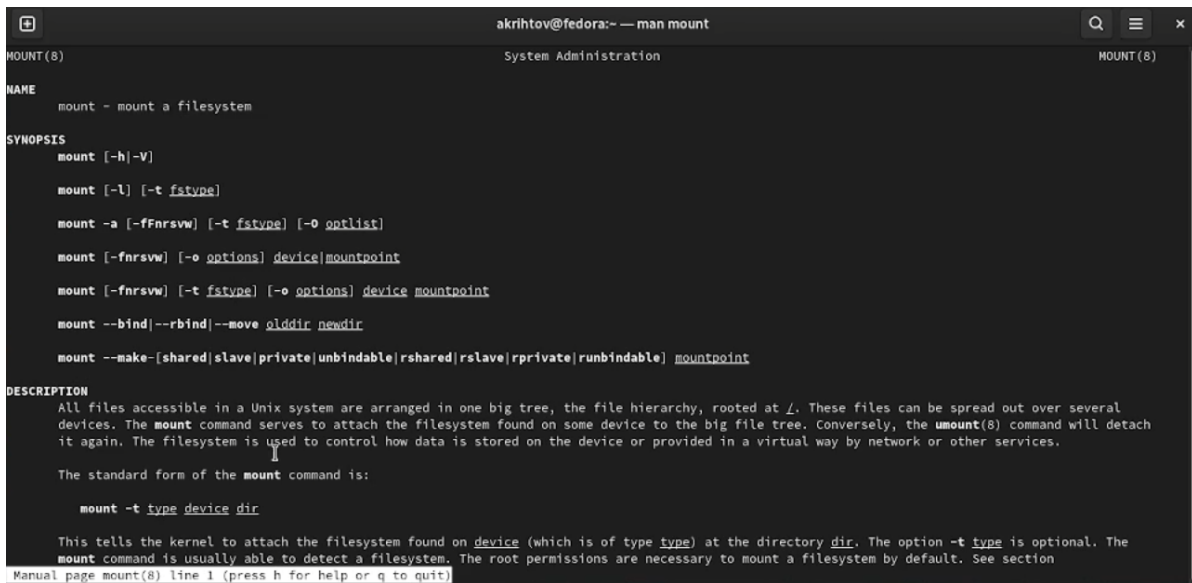
```

Рис. 3.15: Проверка команды chmod

5. Использовали команду `man` для команд `mount`, `fsck`, `mkfs`, `kill`. (рис. 16-20)

```
[akrihtov@fedora ~]$ man mount
[akrihtov@fedora ~]$ man fsck
[akrihtov@fedora ~]$ man mkfs
[akrihtov@fedora ~]$ man kill
[akrihtov@fedora ~]$
```

Рис. 3.16: Использование команды `man`



The screenshot shows a terminal window titled "akrihtov@fedora:~ — man mount". The window displays the manual page for the `mount` command. The page is divided into sections: NAME, SYNOPSIS, and DESCRIPTION. The NAME section states "mount - mount a filesystem". The SYNOPSIS section lists various command-line options and their syntax, including `mount [-h|-V]`, `mount [-l] [-t fstype]`, `mount -a [-fnrsvw] [-t fstype] [-o optlist]`, `mount [-fnrsvw] [-o options] device|mountpoint`, `mount [-fnrsvw] [-t fstype] [-o options] device mountpoint`, `mount --bind|--rbind|--move olddir newdir`, and `mount --make-[shared|slave|private|unbindable|rshared|rsave|rprivate|runbindable] mountpoint`. The DESCRIPTION section explains that 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. The standard form of the `mount` command is: `mount -t type device dir`. This tells the kernel to attach the filesystem found on `device` (which is of type `type`) at the directory `dir`. The option `-t type` is optional. The `mount` command is usually able to detect a filesystem. The root permissions are necessary to mount a filesystem by default. See section "Manual page mount(8) line 1 (press h for help or q to quit)".

Рис. 3.17: Вывод команды `man mount`

```
akrihtov@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.

  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

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

Рис. 3.18: Вывод команды man fsck

```
akrihtov@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.

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

  fs-options
    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.

  -V, --version

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

Рис. 3.19: Вывод команды man mkfs


```
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.

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

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

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

Manual page kill(1) line 1 (press h for help or q to quit)
право на выполнение.
```

Рис. 3.20: Вывод команды man kill

mount: Эта команда позволяет монтировать отделы жёсткого диска, к файловой структуре ОС.

mount -t vfstype device dir Такая команда предлагает ядру смонтировать (подключить) файловую систему указанного типа vfstype, расположенную на устройстве device, к заданному каталогу dir, который часто называют точкой монтирования. Предыдущее содержимое, владелец и режим доступа к каталогу dir становятся недоступными (исчезают), а вновь появившиеся продолжают действовать, пока файловая система device смонтирована (подключена) к dir.

fsck: Используется для восстановления поврежденных файловых систем в ситуациях, когда система не загружается или раздел не может быть смонтирован.

fsck [OPTIONS] [FILESYSTEM]

mkfs: Команда обычно используется для управления устройствами хранения в Linux. Вы можете рассматривать mkfs как инструмент командной строки для форматирования диска в определенной файловой системе.

mkfs -t [fs type] [target device]

kill: Чаще всего данная команда используется для принудительного завершения работы определенных процессов.

kill [параметры] идентификатор-процесса

4 Выводы

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, по проверке использования диска и обслуживанию файловой системы.

5 Ответы на 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. 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:

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.
9. Как создаётся файловая система?

С помощью команды `mkfs`.

6. Дайте характеристику командам для просмотра текстовых файлов.

`Cat` - выводит содержимое файла на стандартное устройство вывода. Для просмотра файлов постранично удобнее использовать команду `less`.

7. Приведите основные возможности команды `cp` в Linux.

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

8. Приведите основные возможности команды `mv` в Linux.

Команда `mv` предназначена для перемещения и переименования файлов и каталогов. Формат команды `mv`:

`mv [-опции] старый_файл новый_файл`

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.