

Отчет по лабораторной работе №6

Дисциплина

Филиппова Анна Дмитриевна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	15
4	Контрольные вопросы	16

List of Figures

2.1	Копирование файлов и каталогов	5
2.2	Перемещение и переименовывание файлов и каталогов	6
2.3	Изменение прав доступа	6
2.4	Копирование файлов создание, домашних каталогов, перемещение файлов и тд	7
2.5	Копирование файлов создание, домашних каталогов, перемещение файлов и тд	7
2.6	Присваиваем файлам права доступа	8
2.7	Присваиваем файлам права доступа	9
2.8	Просматриваем содержимое файла	10
2.9	Копируем файлы, каталоги даем и лишаем прав владельца	10
2.10	Команда man	11
2.11	Справка mount	12
2.12	Справка fsck	13
2.13	Справка mkfs	14
2.14	Справка kill	14

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 Выполнение лабораторной работы

1. Выполняем примеры, описанные в первой части описания лабораторной работы. (рис. 2.1) , (рис. 2.2) , (рис. 2.3)

```
[adfilippova@adfilippova ~]$ cd
[adfilippova@adfilippova ~]$ touch abc1
[adfilippova@adfilippova ~]$ cp abc1 april
[adfilippova@adfilippova ~]$ cp abc1 may
[adfilippova@adfilippova ~]$ mkdir monthly
[adfilippova@adfilippova ~]$ cp april may monthly
[adfilippova@adfilippova ~]$ cp monthly/may monthly/june
[adfilippova@adfilippova ~]$ ls monthly
april  june  may
[adfilippova@adfilippova ~]$ mkdir monthly.00
[adfilippova@adfilippova ~]$ cp -r monthly monthly.00
[adfilippova@adfilippova ~]$ cp -r monthly /tmp
[adfilippova@adfilippova ~]$ █
```

Figure 2.1: Копирование файлов и каталогов

Скопируем файл ~/abc1 в файл april и в файл may. Для этого создадим файл abc1, используя команду «touch abc1», далее осуществим копирование с помощью команд «cp abc1 april» и «cp abc1 may».

Скопируем файлы april и may в каталог monthly, используя команды «mkdir monthly» – для создания каталога monthly и «cp april may monthly» – для копирования.

Скопируем файл monthly/may в файл с именем june. Выполним команды «cp monthly/may monthly/june» и «ls monthly» (для просмотра содержимого каталога).

Скопируем каталог monthly в каталог monthly.00. Для этого создадим каталог monthly.00 командой «mkdir monthly.00» и осуществим копирование, используя команду «cp -r monthly monthly.00» (команда cp с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами).

Скопируем каталог `monthly.00` в каталог `/tmp`, используя команду «`cp -r monthly.00 /tmp`».

```
[adfilippova@adfilippova ~]$ cd
[adfilippova@adfilippova ~]$ mv april july
[adfilippova@adfilippova ~]$ mv july monthly.00
[adfilippova@adfilippova ~]$ ls monthly.00
july  monthly
[adfilippova@adfilippova ~]$ mv monthly.00 monthly.01
[adfilippova@adfilippova ~]$ mkdir reports
[adfilippova@adfilippova ~]$ mv monthly.01 reports
[adfilippova@adfilippova ~]$ mv reports/monthly.01 reports/monthly
[adfilippova@adfilippova ~]$
```

Figure 2.2: Перемещение и переименовывание файлов и каталогов

Изменим название файла `april` на `july` в домашнем каталоге, используя команду «`mv april july`».

Переместим файл `july` в каталог `monthly.00` с помощью команды «`mv july monthly.00`». Проверим результат командой «`ls monthly.00`».

Переименуем каталог `monthly.00` в `monthly.01`, используя команду «`mv monthly.00 monthly.01`».

Переместим каталог `monthly.01` в каталог `reports`. Для этого создадим каталог `reports` с помощью команды «`mkdir reports`» и выполним перемещение командой «`mv monthly.01 reports`».

Переименуем каталог `reports/monthly.01` в `reports/monthly` командой «`mv reports/monthly.01 reports/monthly`».

```
[adfilippova@adfilippova ~]$ cd
[adfilippova@adfilippova ~]$ touch may
[adfilippova@adfilippova ~]$ ls -l may
-rw-rw-r--. 1 adfilippova adfilippova 0 май  9 00:25 may
[adfilippova@adfilippova ~]$ chmod u+x may
[adfilippova@adfilippova ~]$ ls -l may
-rwxrw-r--. 1 adfilippova adfilippova 0 май  9 00:25 may
[adfilippova@adfilippova ~]$ chmod u-x may
[adfilippova@adfilippova ~]$ la -l may
bash: la: команда не найдена...
[adfilippova@adfilippova ~]$ ls -l may
-rw-rw-r--. 1 adfilippova adfilippova 0 май  9 00:25 may
[adfilippova@adfilippova ~]$ cd
[adfilippova@adfilippova ~]$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
[adfilippova@adfilippova ~]$ chmod go-r monthly
chmod: невозможно получить доступ к «monthly»: Нет такого файла или каталога
[adfilippova@adfilippova ~]$ chmod go-r monthly
[adfilippova@adfilippova ~]$ cd
[adfilippova@adfilippova ~]$ touch abc1
[adfilippova@adfilippova ~]$ chmod g+w abc1
```

Figure 2.3: Изменение прав доступа

Создадим файл ~/may с правом выполнения для владельца. Для этого выполним следующие команды: «touch may» (создание файла), «ls -l may» (просмотр сведений о файле), «chmod u+x may» (изменение прав), «ls -l may».

Лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may» (изменение прав), «ls -l may» (просмотр сведений о файле).

Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly» (создание каталога), «chmod go-r monthly» (изменение прав).

Создаем файл ~/abc1 с правом записи для членов группы, используя команды: «touch abc1» (создание файла), «chmod g+w abc1» (изменение прав).

2. Выполняем следующие действия из задания лабораторной работы.(рис. 2.4), (рис. 2.5)

```
[adfilippova@adfilippova ~]$ cp /usr/include/sys/io.h ~
[adfilippova@adfilippova ~]$ mv io.h equipment
[adfilippova@adfilippova ~]$ mkdir ski.places
[adfilippova@adfilippova ~]$ mv equipment ski.places
[adfilippova@adfilippova ~]$ mv ski.places/equipment ski.places/equipment
mv: «ski.places/equipment» и «ski.places/equipment» - один и тот же файл
[adfilippova@adfilippova ~]$ mv ski.places/equipment ski.places/equiplist
[adfilippova@adfilippova ~]$ touch abc1
[adfilippova@adfilippova ~]$ cp abc1 ski.places
[adfilippova@adfilippova ~]$ mv ski.places/abc1 ski.places/equiplist2
[adfilippova@adfilippova ~]$ mkdir ski.places/equipment
```

Figure 2.4: Копирование файлов создание, домашних каталогов, перемещение файлов и тд

```
[adfilippova@adfilippova ~]$ mv ski.places/equiplist ski.places/equiplist2 ski.places/e
quipment
[adfilippova@adfilippova ~]$ mkdir newdir
mkdir: невозможно создать каталог «newdir»: Файл существует
[adfilippova@adfilippova ~]$ mv newdir ski.places
[adfilippova@adfilippova ~]$ mv ski.places/newdir ski.places/plans
[adfilippova@adfilippova ~]$ █
```

Figure 2.5: Копирование файлов создание, домашних каталогов, перемещение файлов и тд

- 1) Копируем файл /usr/include/sys/io.h в домашний каталог (команда «cp /usr/include/sys/io.h ~»)и назоваем его equipment (команда «mv aio.h equipment»).

- 2) В домашнем каталоге создаем директорию ~/ski.places (команда «mkdir ski.places»).
 - 3) Перемещаем файл equipment в каталог ~/ski.places (команда «mv equipment ski.places»).
 - 4) Переименовываем файл ~/ski.places/equipment в ~/ski.places/equiplist (команда «mv ski.places/equipment ski.places/equiplist»).
 - 5) Создаем в домашнем каталоге файл abc1 (команда «touch abc1») и копируем его в каталог ~/ski.places (команда «cp abc1 ski.places»), называем его equiplist2 (команда «mv ski.places/abc1 ski.places/equiplist2»).
 - 6) Создаем каталог с именем equipment в каталоге ~/ski.places (команда «mkdir ski.places/equipment»).
 - 7) Перемещаем файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment (команда «mv ski.places/equiolist ski.places/equiplist2 ski.places/equipment»).
 - 8) Создаем (команда «mkdir newdir») и перемещаем каталог ~/newdir в каталог ~ ski.places (команда «mv newdir ski.places») и называем его plans (команда «mv ski.places/newdir ski.places/plans»).
3. Определяем опции команды chmod, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем необходимые файлы, используя команды: «mkdir australia», «mkdir play», «touch my_os», «touch feathers». (рис. 2.6), (рис. 2.7)

```
[adfilippova@adfilippova ~]$ mkdir australia
[adfilippova@adfilippova ~]$ mkdir play
[adfilippova@adfilippova ~]$ touch my_os
[adfilippova@adfilippova ~]$ touch feathers
[adfilippova@adfilippova ~]$ chmod 744 australia
[adfilippova@adfilippova ~]$ chmod 711 play
bash: chmod: команда не найдена...
Аналогичная команда: 'chmod'
[adfilippova@adfilippova ~]$ chmod 711 play
[adfilippova@adfilippova ~]$ chmod 544 my_os
[adfilippova@adfilippova ~]$ chmod 664 feathers
[adfilippova@adfilippova ~]$ ls -l
```

Figure 2.6: Присваиваем файлам права доступа


```

drwxr--r--. 2 adfilippova adfilippova      6 май  9 01:35 australia
-rw-rw-r--. 1 adfilippova adfilippova      0 май  9 01:36 feathers
-rw-rw-r--. 1 adfilippova adfilippova      0 май  9 00:25 may
drwx-wx--x. 2 adfilippova adfilippova     42 май  9 00:17 monthly
-r-xr--r--. 1 adfilippova adfilippova      0 май  9 01:36 my_os
drwxrwxr-x. 2 adfilippova adfilippova      6 май  8 16:53 new
drwxrwxr-x. 4 adfilippova adfilippova     35 апр 29 03:29 os-introl
drwxr-xr-x. 4 root        root            30 ноя 27 2018 pandoc-2.5
-rw-r--r--. 1 root        root            28483450 ноя 27 2018 pandoc-2.5-linux.tar.gz
-rw-rw-r--. 1 adfilippova adfilippova    13800700 мар 24 2020 pandoc-2.9.2.1-1-amd64.deb
-rw-rw-r--. 1 adfilippova adfilippova    13800700 мар 24 2020 pandoc-2.9.2.1-1-amd64.deb.
1
-rw-r--r--. 1 root        root            13800700 мар 24 2020 pandoc-2.9.2.1-1-amd64.deb.
2
-rw-rw-r--. 1 adfilippova adfilippova    6675912 апр 30 22:58 pandoc-crossref-Linux.tar.x
z
drwx--x--x. 2 adfilippova adfilippova      6 май  9 01:35 play

```

Figure 2.7: Присваиваем файлам права доступа

- 1) drwxr--r-- ... australia: команда «chmod 744 australia» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение).
- 2) drwx--x--x ... play: команда «chmod 711 play» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение).
- 3) -r-xr--r-- ... my_os: команды «chmod 544 my_os» (это файл, владелец имеет право на чтение и выполнение, группа владельца и остальные – только чтение).
- 4) -rw-rw-r-- ... feathers: команды «chmod 664 feathers» (это файл,владелец и группа владельца имеют право на чтение и запись,остальные – только чтение). Командой «ls -l» проверяем правильность выполненных действий.
4. Продلываем упражнения, которые даны в тексте лабораторной работы. (рис. 2.8), (рис. 2.9)

```
[adfilippova@adfilippova ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
libstoragemgmt:x:998:996:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
colord:x:997:995:User for colord:/var/lib/colord:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
sane:x:996:994:SANE scanner daemon user:/usr/share/sane:/sbin/nologin
sasauthd:x:995:76:Sasauthd user:/run/sasauthd:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
setroubleshoot:x:994:991::/var/lib/setroubleshoot:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
```

Figure 2.8: Просматриваем содержимое файла

```
[adfilippova@adfilippova ~]$ cp feathers file.old
[adfilippova@adfilippova ~]$ mv file.old play
[adfilippova@adfilippova ~]$ cp -r play fun
[adfilippova@adfilippova ~]$ mv fun play
[adfilippova@adfilippova ~]$ mv play/fun play/games
[adfilippova@adfilippova ~]$ chmod u-r feathers
[adfilippova@adfilippova ~]$ cat feathers
cat: feathers: Отказано в доступе
[adfilippova@adfilippova ~]$ cp feathers play
cp: невозможно открыть «feathers» для чтения: Отказано в доступе
[adfilippova@adfilippova ~]$ chmod u+r feathers
[adfilippova@adfilippova ~]$ chmod u-x play
[adfilippova@adfilippova ~]$ cd play
bash: cd: play: Отказано в доступе
[adfilippova@adfilippova ~]$ chmod u+x play
[adfilippova@adfilippova ~]$ █
```

Figure 2.9: Копируем файлы, каталоги даем и лишаем прав владельца

- 1) Просмотрим содержимое файла /etc/passwd (команда «cat /etc/passwd»).
- 2) Копируем файл ~/feathers в файл ~/file.old (команда «cp feathers file.old»).
- 3) Переместим файл ~/file.old в каталог ~/play (команда «mv file.ord play»).
- 4) Скопируем каталог ~/play в каталог ~/fun (команда «cp -r play fun»).
- 5) Переместим каталог ~/fun в каталог ~/play (команда «mv fun play») и назовем его games (команда «mv play/fun play/games»).
- 6) Лишим владельца файла ~/feathers права на чтение (команда «chmodu-r feathers»).

- 7) Если мы попытаемся просмотреть файл ~/feathers командой cat, то получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на чтение данного файла.
 - 8) Если мы попытаемся скопировать файл ~/feathers, например, в каталог monthly, то получим отказ в доступе, по причине, описанной в пункте 7.
 - 9) Дадим владельцу файла ~/feathers право на чтение (команда «chmod u+r feathers»).
 - 10) Лишим владельца каталога ~/play права на выполнение (команда «chmod u-x play»).
 - 11) Перейдем в каталог ~/play (команда «cd play»). Получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на выполнение данного каталога.
 - 12) Дадим владельцу каталога ~/play право на выполнение (команда «chmod u+x play»).
5. Используя команды «man mount», «man fsck», «man mkfs», «man kill», получим информацию о соответствующих командах. (рис. 2.10)

```
[adfilippova@adfilippova ~]$ man mount
[adfilippova@adfilippova ~]$ man fsck
[adfilippova@adfilippova ~]$ man mkfs
[adfilippova@adfilippova ~]$ man kill
```

Figure 2.10: Команда man

Команда mount: предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах).

Корень этого дерева обозначается как /. Физически файлы могут располагаться на различных устройствах.

Команда `mount` служит для подключения файловых систем разных устройств к этому большому дереву. Наиболее часто встречающаяся форма команды `mount` выглядит следующим образом: «`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. (рис. 2.11)

```

MOUNT(8)                                     System Administration                                     MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-lhv]

    mount -a [-fFnrsvw] [-t vfstype] [-O optlist]

    mount [-fnrsvw] [-o option[,option]...] device|dir

    mount [-fnrsvw] [-t vfstype] [-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 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 standard form of the mount command, is

        mount -t type device dir

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

```

Figure 2.11: Справка mount

Команда `fsck`: это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Он использует программы, специфичные для типа файловой системы, которую он проверяет.

У команды `fsck` следующий синтаксис: `fsck [параметр] – [параметры ФС] [. . .]` Например, если нужно восстановить («починить») файловую систему на некотором устройстве `/dev/sdb2`, следует воспользоваться командой: «`sudo fsck -y /dev/sdb2`».

Опция `-y` необходима, т. к. при её отсутствии придётся слишком часто давать подтверждение. (рис. 2.12)

```

FSCK(8)                                     System Administration                                     FSCK(8)
NAME
    fsck - check and repair a Linux filesystem

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

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems.
    filesystems can be a device name (e.g. /dev/hdc1, /dev/sdb2), a mount point
    (e.g. /, /usr, /home), or an ext2 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:

        0      No errors

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

```

Figure 2.12: Справка fsck

Команда **mkfs**: создаёт новую файловую систему Linux. Имеет следующий синтаксис: **mkfs [V] [-t fstype] [fs-options] filesystem [blocks]** **mkfs** используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента **filesystem** для файловой системы может выступать или название устройства (например, /dev/hda1, /dev/sdb2) или точка монтирования (например, /, usr, /home).

Аргументом **blocks** указывается количество блоков, которые выделяются для использования этой файловой системой.

По окончании работы **mkfs** возвращает 0 - в случае успеха, а 1 - при неудачной операции. Например, команда «**mkfs -t ext2 /dev/hdb1**» создаёт файловую систему типа **ext2** в разделе /dev/hdb1 (второй жёсткий диск). (рис. 2.13)

```

MKFS(8)                                System Administration                                MKFS(8)

NAME
    mkfs - build a Linux filesystem

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

DESCRIPTION
    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 in a number of directories, like perhaps /sbin, /sbin/fs,
/sbin/fs.d, /etc/fs, /etc (the precise list is defined at compile time but at
    least contains /sbin and /sbin/fs), and finally in the directories listed in
    the PATH environment variable. Please see the filesystem-specific builder
    manual pages for further details.

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

```

Figure 2.13: Справка mkfs

Команда **kill**: посылает сигнал процессу или выводит список допустимых сигналов. Имеет следующий синтаксис: **kill [опции] PID**, где PID – это PID (числовой идентификатор) процесса или несколько PID процессов, если требуется послать сигнал сразу нескольким процессам.

Например, команда «**kill -KILL 3121**» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс. (рис. 2.14)

```

KILL(1)                                User Commands                                KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-s signal|-p] [-q sigval] [-a] [--] pid...
    kill -l [signal]

DESCRIPTION
    The command kill sends the specified signal to the specified process or
    process group. If no signal is specified, the TERM signal is sent. The TERM
    signal will kill processes which do not catch this signal. For other pro-
    cesses, it may be necessary to use the KILL (9) signal, since this signal
    cannot be caught.

    Most modern shells have a builtin kill function, with a usage rather similar
    to that of the command described here. The '-a' and '-p' options, and the
    possibility to specify processes by command name are a local extension.

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

OPTIONS
    pid... Specify the list of processes that kill should signal. Each pid can
    Manual page kill(1) line 1 (press h for help or q to quit)

```

Figure 2.14: Справка kill

3 Выводы

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

4 Контрольные вопросы

1. Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th». После выполнения команды видно, что на моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. devtmpfs позволяет ядру создать экземпляр tmpfs с именем devtmpfs при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором / минором будет предоставлять узел устройства в devtmpfs. devtmpfs монтируется на /dev и содержит специальные файлы устройств для всех устройств. tmpfs – временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначена для быстрого и ненадёжного хранения временных данных. Хорошо подходит для /tmp и массовой сборки пакетов/образов. Предполагает наличие достаточного объёма виртуальной памяти. Файловая система tmpfs предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска. ext4 – имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «extent file system», который позволяет добиться меньшей фрагментации файлов и увеличить общую

производительность файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (delayed allocation – delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных. Характеристики: -максимальный размер файла: 16 TB; -максимальный размер раздела: 16 TB; -максимальный размер имени файла: 255 символов. Рекомендации по использованию: -наилучший выбор для SSD; -наилучшая производительность по сравнению с предыдущими Ext-системами; -она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3. ISO 9660 – стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

2. Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел – / (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам. / – корень Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом каталоге. /BIN – бинарные файлы пользователя Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском

режиме или режиме восстановления. /sbin – системные исполняемые файлы. Так же как и /bin, содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки, когда не примонтирован каталог /usr. Но здесь находятся программы, которые можно выполнять только с правами суперпользователя. /etc – конфигурационные файлы. В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, в системе инициализации Init Scripts, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ. /dev – файлы устройств. В Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры – это просто файлы в каталоге /dev/. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов. /proc – информация о процессах. По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его Pid, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов. /var – переменные файлы. Название каталога /var говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кешы, базы данных и так далее. /tmp – временные файлы. В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию. /usr – программы пользователя. Это самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию. /home – домашняя папка. В этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т. д. /boot – файлы загрузчика. Содержит все

файлы, связанные с загрузчиком системы. Это ядро `vmlinuz`, образ `initrd`, а также файлы загрузчика, находящиеся в каталоге `/boot/grub`. `/LIB` – системные библиотеки Содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах `/bin` и `/sbin`. `/OPT` – дополнительные программы В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями. `/MNT` – монтирование В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы. `/MEDIA` – съемные носители В этот каталог система монтирует все подключаемые внешние накопители –USB флешки, оптические диски и другие носители информации. `/SRV` – сервер В этом каталоге содержатся файлы серверов и сервисов. `/RUN` - процессы Каталог, содержащий PID файлы процессов, похожий на `/var/run`, но в отличие от него, он размещен в `TMPFS`, а поэтому после перезагрузки все файлы теряются.

3. Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой `mount`.
4. Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду `fsck`.
5. Файловую систему можно создать, используя команду `mkfs`.
6. Для просмотра текстовых файлов существуют следующие команды: `cat`
Задача команды `cat` очень проста – она читает данные из файла или стандартного ввода и выводит их на экран. Синтаксис утилиты: `cat [опции] файл1 файл2 ...` Основные опции: `-b` – нумеровать только непустые строки `-E` – показывать символ `$` в конце каждой строки `-n` – нумеровать все строки `-s` – удалять пустые повторяющиеся строки `-T` – отображать табуляции в виде `^I` `-h` – отобразить справку `-v` – версия утилиты `nl` Команда `nl`

действует аналогично команде `cat`, но выводит еще и номера строк в столбце слева. `less` Существенно более развитая команда для пролистывания текста. При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста. Синтаксис аналогичный синтаксису команды `cat`. Некоторые опции: `-g` – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения) `-N` – показывать номера строк `head` Команда `head` выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита. Синтаксис аналогичный синтаксису команды `cat`. Основные опции: `-c` (`-bytes`) – позволяет задавать количество текста не в строках, а в байтах `-n` (`-lines`) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию `-q` (`-quiet`, `-silent`) – выводит только текст, не добавляя к нему название файла `-v` (`-verbose`) – перед текстом выводит название файла `-z` (`-zero-terminated`) – символы перехода на новую строку заменяет символами завершения строк `tail` Эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды `cat`. Основные опции: `-c` – выводить указанное количество байт с конца файла `-f` – обновлять информацию по мере появления новых строк в файле `-n` – выводить указанное количество строк из конца файла `-pid` – используется с опцией `-f`, позволяет завершить работу утилиты, когда завершится указанный процесс `-q` – не выводить имена файлов `-retry` – повторять попытки открыть файл, если он недоступен `-v` – выводить подробную информацию о файле

7. Утилита `cp` позволяет полностью копировать файлы и директории. Синтаксис: `cp [опции] файл-источник файл-приемник` После выполнения команды файл-источник будет полностью перенесен в файл-приемник. Если в конце

указан слэш, файл будет записан в заданную директорию с оригинальным именем. Основные опции: `-attributes-only` – не копировать содержимое файла, а только флаги доступа и владельца `-f`, `-force` – перезаписывать существующие файлы `-i`, `-interactive` – спрашивать, нужно ли перезаписывать существующие файлы `-L` – копировать не символические ссылки, а то, на что они указывают `-n` – не перезаписывать существующие файлы `-P` – не следовать символическим ссылкам `-r` – копировать папку Linux рекурсивно `-s` – не выполнять копирование файлов в Linux, а создавать символические ссылки `-u` – скопировать файл, только если он был изменён `-x` – не выходить за пределы этой файловой системы `-p` – сохранять владельца, временные метки и флаги доступа при копировании `-t` – считать файл-приемник директорией и копировать файл-источник в эту директорию

8. Команда `mv` используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий. Синтаксис: `mv [-опции] старый_файл новый_файл`
- Основные опции: `-help` – выводит на экран официальную документацию об утилите `-version` – отображает версию `mv` `-b` – создает копию файлов, которые были перемещены или перезаписаны `-f` – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла `-i` – наоборот, будет спрашивать разрешение у владельца `-n` – отключает перезапись уже существующих объектов `-strip-trailing-slashes` – удаляет завершающий символ `/` у файла при его наличии `-t [директория]` – перемещает все файлы в указанную директорию `-u` – осуществляет перемещение только в том случае, если исходный файл новее объекта назначения `-v` – отображает сведения о каждом элементе во время обработки команды
- Команда `rename` также предназначена, чтобы переименовать файл. Синтаксис: `rename [опции] старое_имя новое_имя`
- файлы
- Основные опции: `-v` – вывести список обработанных файлов `-n` – тестовый режим, на самом деле никакие действия выполнены не будут `-f` –

принудительно перезаписывать существующие файлы

9. Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её установленных носителям, правовыми процессам и документами другим или ресурсам) собственником, владельцем информации. Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Синтаксис команды: `chmod режим имя_файла` Режим 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