

Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного образовательного  
учреждения высшего образования  
**«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)**

Ю.С. Белов, Е.А. Черепков

**АДМИНИСТРИРОВАНИЕ ФАЙЛОВОЙ СИСТЕМЫ LINUX**  
Методические указания к выполнению лабораторной работы  
по курсу «Операционные системы»

Калуга – 2018

УДК 004.62  
ББК 32.972.1  
Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий».

Методические указания рассмотрены и одобрены:

- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол № 3 от «24» октября 2018 г.

Зав. кафедрой ИУ4-КФ

 к.т.н., доцент Ю.Е. Гагарин

- Методической комиссией факультета ИУ-КФ протокол № 3 от «29» сентября 2018 г.


Председатель методической  
комиссии факультета ИУ-КФ

 к.т.н., доцент М.Ю. Адкин

- Методической комиссией

КФ МГТУ им.Н.Э. Баумана протокол № 2 от «6» кабеля 2018 г.

Председатель методической комиссии  
КФ МГТУ им.Н.Э. Баумана

 д.э.н., профессор О.Л. Перерва

Рецензент:

к.т.н., доцент кафедры ИУ3-КФ

 А.В. Финюшин

Авторы

к.ф.-м.н., доцент кафедры ИУ4-КФ  
ассистент кафедры ИУ4-КФ

 Ю.С. Белов  
 Е.А. Черепков

#### Аннотация

Методические указания к выполнению лабораторной работы по курсу «Операционные системы» содержат общие сведения о системных каталогах в ОС семейства Linux, а также синтаксис основных команд для монтирования устройств, архивации и сжатия файлов.

Предназначены для студентов 3-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2018 г.  
© Ю.С. Белов, Е.А. Черепков, 2018 г.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ.....	5
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ .....	6
СТАНДАРТ FHS (FILESYSTEM HIERARCHY STANDARD) .....	10
МОНТИРОВАНИЕ И ДЕМОНТИРОВАНИЕ ФАЙЛОВЫХ СИСТЕМ .....	16
АРХИВАЦИЯ ДАННЫХ.....	26
СЖАТИЕ ДАННЫХ.....	31
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ .....	36
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ .....	37
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ .....	37
ОСНОВНАЯ ЛИТЕРАТУРА.....	39
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА .....	39

## **ВВЕДЕНИЕ**

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Операционные системы» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 3-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткое описание команд для работы с устройствами, команды монтирования файловых систем, а также для архивации и сжатия файлов в операционной системе Linux.

Методические указания составлены для ознакомления студентов с работой с устройствами, каталогами и файлами, утилитами для архивации и сжатия файлов в операционной системе Linux. Для выполнения лабораторной работы студенту необходимы минимальные знания об операционной системе Linux.

## **ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ**

Целью выполнения лабораторной работы является приобретение практических навыков по работе с устройствами и файловой системой ОС Linux.

Основными задачами выполнения лабораторной работы являются:

1. Ознакомиться с устройством файловой системы ОС Linux.
2. Получить навыки работы с устройствами ОС Linux.
3. Изучить основные команды монтирования файловых систем ОС Linux.
4. Получить навыки работы с утилитами для архивации и сжатия файлов в ОС Linux.
5. Изучить основные опции команд для архивации и сжатия файлов в ОС Linux.

Результатами работы являются:

1. Демонстрация выполнения команд для работы с файловой системой, архивации и сжатия файлов.
2. Подготовленный отчет.

## **КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ**

Все файлы в системе Linux логически соединены в одно общее дерево, но сами файлы размещаются на различных запоминающих устройствах, например на жестких дисках и компакт-дисках. Файлы, записанные на запоминающих устройствах, организованы в файловые системы. Дерево каталогов в ОС Linux может охватывать несколько файловых систем, каждая из которых размещена на отдельном устройстве. Сами файлы организованы в единое файловое дерево, вершиной которого является корневой каталог.

Файлы той или иной файловой системы остаются отделенными от дерева каталогов до тех пор, пока вы явно не присоедините их к этому дереву. В каждой файловой системе файлы организованы в отдельное дерево каталогов. Это дерево можно рассматривать как поддерево, которое необходимо присоединить к основному дереву каталогов. Например, на диске с файлами ОС Linux — свое дерево каталогов. Это поддерево нужно присоединить к основному дереву, расположенному в разделе жесткого диска. Пока это не сделано, доступа к файлам на диске у вас не будет.

Присоединение файловой системы, расположенной на запоминающем устройстве, к основному дереву каталогов называется монтированием устройства. Операция монтирования предполагает присоединение дерева каталогов, находящегося на устройстве памяти, к указанному пользователем каталогу. Лишь после этого можно перейти в присоединенный каталог и обращаться к его файлам. Каталог, к которому вы хотите присоединить файлы, находящиеся на запоминающем устройстве, называется точкой монтирования.

Монтирование файловой системы может осуществляться только привилегированным пользователем, поскольку эта задача относится к функциям системного администратора. После монтирования компакт-диска его невозможно будет свободно извлечь и вставить новый (также не следует вынимать из дисковода смонтированную дискету). Так, привод компакт-диска останется запертым до тех пор, пока вы не

демонтируете компакт-диск. После этого диск можно извлечь и вставить другой, который, прежде чем использовать, необходимо смонтировать. При каждой замене компакт-диска или дискеты вам необходимо будет повторять операцию монтирования - демонтирования.

**Таблица 1.** Типы файловых систем

<b>Опция</b>	<b>Файловая система</b>
minix	Файловые системы Minux; длина имени файла ограничена 30 символами
ext	Ранние версии файловой системы Linux уже не используются
ext2	Стандартная файловая система Linux, поддерживающая длинные имена и большие размеры файлов
xiaf	Файловая система Xiaf
msdos	Файловая система для разделов MS-DOS
hpfs	Файловая система для разделов OS/2
proc	Специальная файловая система для процессов
nfs	Сетевая файловая система, монтируемая из удаленных систем
umsdos	Файловая система UMS-DOS
swap	Раздел или файл подкачки ОС Linux
sysv	Файловая система UNIX System V
Iso9660	Файловая система для монтирования CD-ROM

При работе с Linux вы, может быть, еще не скоро встретитесь с необходимостью работать с большинством консольных команд, поскольку имеются такие оболочки, как Midnight Commander или графические оболочки типа KDE. Но с командами архивирования (точнее, разархивирования) вам работать придется обязательно, хотя бы потому, что вы будете часто встречать архивированные файлы в Интернете.

Основным средством архивирования в UNIX (а, следовательно, и в Linux) является комплекс из двух программ — tar и gzip. Хотя никто не запрещает пользоваться arj, pkzip, lha, rar и т. д. — версии этих программ для Linux общедоступны. Просто уж исторически сложилось, что пользователи Unix чаще применяют именно tar и gzip, и именно в таком формате распространяется большая часть программного обеспечения для Unix. Поэтому овладеть работой с tar и gzip — дело чести любого пользователя Linux.

**Сжатие информации** - это процесс преобразования информации, хранящейся в файле, к виду, при котором уменьшается избыточность в ее представлении и соответственно требуется меньший объем памяти для хранения.

Сжатие информации в файлах производится за счет устранения избыточности различными способами, например за счет упрощения кодов, исключения из них постоянных битов или представления повторяющихся символов или повторяющейся последовательности символов в виде коэффициента повторения и соответствующих символов. Применяются различные алгоритмы подобного сжатия информации.

Сжиматься могут как один, так и несколько файлов, которые в сжатом виде помещаются в так называемый архивный файл (архив).

**Архивный файл** - это специальным образом организованный файл, содержащий в себе один или несколько файлов в сжатом или несжатом виде и служебную информацию об именах файлов, дате и времени их создания или модификации, размерах и т.д.

Целью упаковки файлов обычно являются обеспечение более компактного размещения информации на диске, сокращение времени и соответственно стоимости передачи информации по каналам связи в компьютерных сетях. Кроме того, упаковка в один архивный файл группы файлов существенно упрощает их перенос с одного компьютера на другой, сокращает время копирования файлов на диски, позволяет защитить информацию от несанкционированного доступа, способствует защите от несанкционированного доступа.



**Степень сжатия** файлов характеризуется коэффициентом  $K_c$ , определяемым как отношение объема сжатого файла  $V_c$  к объему исходного файла  $V_0$  выраженное в процентах:  $K_c = \frac{V_c}{V_0} * 100\%$

Степень сжатия зависит от используемой программы, метода сжатия и типа исходного файла. Наиболее хорошо сжимаются файлы графических образов, текстовые файлы и файлы данных, для которых степень сжатия может достигать 5 - 40 %, меньше сжимаются файлы исполняемых программ и загрузочных модулей - 60 - 90 %. Почти не сжимаются архивные файлы. Программы для архивации отличаются используемыми методами сжатия, что соответственно влияет на степень сжатия.

**Архивация (упаковка)** - помещение (загрузка) исходных файлов в архивный файл в сжатом или несжатом виде.

**Разархивация (распаковка)** - процесс восстановления файлов из архива точно в таком же виде, какой они имели до загрузки в архив. При распаковке файлы извлекаются из архива и помещаются на диск или в оперативную память.

## СТАНДАРТ FHS (FILESYSTEM HIERARCHY STANDARD)

Структура системы каталогов определяется стандартом Filesystem Hierarchy Standard (FHS). FHS определяет стандартизованную структуру системных каталогов, которой должны соответствовать все дистрибутивы Linux.

В Linux используется множество каталогов со стандартными именами, предназначенных для выполнения стандартных административных задач. Все они расположены на самом верхнем уровне файловой системы — в корневом каталоге, представленном косой чертой (/). Например, в каталоге /dev хранятся файлы устройств (терминалов, дисков, модемов), в каталоге /etc — конфигурационные файлы и файлы запуска, а в каталоге /home — пользовательские начальные каталоги и пользовательские файлы.

Все каталоги, хранящиеся в корневом каталоге системы (/), перечислены в табл. 2. Кроме того, в этой таблице приведены имена некоторых полезных подкаталогов.

**Таблица 2.** Каталоги файловой системы Linux

Каталог	Описание
/	Начало структуры файловой системы, называемое корневым каталогом
/boot	Содержит файл образа ядра и модули, загружаемые при загрузке системы
/home	Содержит начальные каталоги пользователей
/sbin	Содержит команды администрирования и команды, используемые привилегированным пользователем
/dev	Содержит файлы устройств, например терминала и принтера
/etc	Содержит конфигурационные и другие важные системные файлы

Продолжение таблицы 2

<b>Каталог</b>	<b>Описание</b>
/etc/opt	Содержит системные конфигурационные файлы для приложений, хранящихся в каталоге /opt
/etc/X11	Содержит системные конфигурационные файлы для X Window System и ее приложений
/etc/sqlml	Содержит системные конфигурационные файлы установленных на компьютер SQLML- или XML-систем
/bin	Содержит важные пользовательские команды и утилиты
/lib	Содержит важные совместно используемые библиотеки и модули ядра
/lib/modules	Содержит модули ядра
/mnt	Содержит каталоги для монтирования файловых систем временно используемых накопителей, таких как компакт-диски и дискеты
/opt	Содержит добавляемые приложения (например, в некоторых дистрибутива включает KDE)
/proc	Каталог процессов — резидентно хранящийся в памяти каталог с файлами содержащими информацию о системе
/tmp	Содержит временные файлы
/usr	Содержит файлы и команды, используемые системой. Этот каталог разделен на несколько подкаталогов
/var	Содержит постоянно изменяющиеся файлы, например файлы почтовых ящиков

Каталог `/usr` имеет важные подкаталоги, предназначенные для поддержки пользователей, предоставления приложений, библиотек и документации (табл. 3).

**Таблица 3.** Подкаталоги каталога `/usr`

<b>Каталог</b>	<b>Описание</b>
<code>/usr/bin</code>	Содержит большинство пользовательских команд и утилит
<code>/usr/sbin</code>	Содержит не самые важные административные приложения
<code>/usr/lib</code>	Содержит библиотеки для приложений, языков программирования, рабочих столов и т. п.
<code>/usr/games</code>	Содержит игры и обучающие программы
<code>/usr/include</code>	Содержит файлы заголовков языка программирования C (.h)
<code>/usr/local</code>	Каталог для программного обеспечения, установленного на локальной машине (все, что устанавлируется пользователем)
<code>/usr/share</code>	Содержит архитектурно-независимые данные, общие для различных систем. Например, к ним относятся страницы электронной документации, такие как <code>Man</code> и <code>info</code>
<code>/usr/src</code>	Содержит исходный код, включая и исходный код ядра
<code>/usr/X11R6</code>	Содержит приложения и библиотеки для X Window System (версия 11, редакция 6).

Каталоги `/var` предназначены для хранения часто изменяющихся данных (табл. 4)

**Таблица 4.** Подкаталоги каталога **/var**

<b>Каталог</b>	<b>Описание</b>
/var/account	Содержит журналы регистрации процессов
/var/cache	Содержит кэшируемые данные приложений для man-страниц, данные прокси-сервера, шрифты и данные, специфические для приложений
/var/crash	Содержит дампы системных сбоев
/var/games	Содержит данные игр
/var/lib	Содержит файлы с информацией о состоянии конкретных приложений
/var/local	Используется для изменяющихся данных программ, установленных в каталог /usr/local
/var/lock	Содержит файлы блокировки, позволяющие определить, используется ли в данный момент конкретный файл или программа
/var/log	Содержит такие файлы журналов, как /var/log/messages, в котором хранятся все сообщения ядра и системных программ
/var/mail	Содержит файлы почтовых ящиков пользователей
/var/opt	Содержит часто изменяющиеся данные приложений, установленных в каталоге /opt
/var/run	Содержит файлы с информацией о работающих в системе процессах
/var/spool	Буферные каталоги приложений, например данные электронной почты, новостей, очередей печати, а также программ cron и at
/var/tmp	Содержит временные файлы, которые должны сохраняться между перезагрузками системы
/var/yp	Содержит файлы данных Network Information Service (NIS)

## Файлы устройств: /dev

Для того чтобы смонтировать файловую систему, нужно задать имя устройства. Интерфейсы устройств, которые монтируются в системе, обеспечиваются специальными файлами, называемыми *файлами устройств*. Имена этих файлов отображают имена соответствующих устройств. Файлы устройств хранятся в каталоге /dev и обычно имеют сокращенные имена, заканчивающиеся номером устройства. В табл. 5 приведен список часто используемых имен устройств.

Файл устройства дисководов НГМД имеет имя fd0. Он хранится в каталоге /dev. Путь к этому файлу — /dev/fd0. Обратите внимание на цифру 0 после «fd». Если в системе больше одного дисковода НГМД, то они будут представлены файлами с именами fd1, fd2 и т. д.

В именах файлов устройств жестких IDE-дисков присутствует префикс hd, а SCSI-дисков sd. После префикса имени жесткого диска следуют буквенное имя диска и номер раздела. Например, файл с именем hda2 относится ко второму разделу первого IDE-диска.

От других IDE-устройств имя файла устройства CD-ROM отличается последним символом. Например, файл IDE-устройства CD-ROM, подключенного ко второму IDE-порту, как правило, будет называться hdc, а IDE-устройства CD-ROM, подключенного в качестве второго устройства ко второму порту, — hdd.

**Таблица 5.** Префиксы имен устройств

Имя устройства	Описание
hd	Жесткие диски с интерфейсом IDE. Цифры 1—4 — первичные разделы, 5 и выше — логические разделы
sd	Жесткие диски с интерфейсом SCSI
sr	Приводы CD-ROM с интерфейсом SCSI
fd	Дисководы НГМД
st	Стримеры с интерфейсом SCSI

Продолжение таблицы 5

<b>Имя устройства</b>	<b>Описание</b>
ht	Стримеры с интерфейсом IDE
tty	Терминалы
lp	Порты принтера
pty	Псевдотерминалы (используются для удаленной регистрации)
js	Аналоговый джойстик
midi	Порты цифрового интерфейса музыкальных инструментов
ttyS	Последовательные порты
cua	Внешние устройства (COM-порты)
cdrom	Ссылка на файл устройства CD-ROM
modem	Ссылка на файл устройства модема

## МОНТИРОВАНИЕ И ДЕМОНТИРОВАНИЕ ФАЙЛОВЫХ СИСТЕМ

Установленные и созданные в файловой системе файлы занимают какую-то часть этого пространства, а остальная часть остается доступной для новых файлов и каталогов. Для того чтобы определить, сколько свободного пространства осталось в файловой системе, можно воспользоваться командой **df**.

Она выдает список всех файловых систем по именам устройств, сообщает их размер и точки монтирования.

```
$ df
Filesystem 1024-blocks Used Available Capacity Mounted
on
/dev/hda3 297635 169499 112764 60% /
/dev/hdal 205380 182320 23060 89% /mnt/dos
/dev/hdc 637986 637986 0 100% /mnt/cdrom
```

### Команды Mount и Umount

Любую файловую систему можно также монтировать и демонтировать при помощи команд **mount** и **umount**. В табл. 6 приведен список опций команды mount.

Команда mount принимает два аргумента: имя устройства, через которое Linux получает доступ к файловой системе, и каталог в файловой структуре, к которому присоединяется новая файловая система. *Точка\_монтирования* — это каталог, к которому вы хотите присоединить файлы, находящиеся на запоминающем устройстве. *Устройство* — это специальный файл, с помощью которого система получает доступ к аппаратным средствам. Команда mount имеет следующий синтаксис:

```
# mount устройство точка_монтирования
```



Файлы устройств находятся в каталогах /dev и обычно имеют сокращенные имена, которые заканчиваются номером устройства. Например, fd0 может обозначать первый дисковод гибких дисков, присоединенный к системе.

**Таблица 6.** Команда mount

<b>Опция</b>	<b>Назначение</b>
-f	Имитирует монтирование файловой системы. Используется для проверки возможности монтирования файловой системы
-v	Текстовый режим. Команда mount выдает описания действий, которые выполняет. Совместно с -f используется для выявления проблем, возникающих при монтировании файловой системы
-w	Монтирует файловую систему с правом на чтение и запись
-r	Монтирует файловую систему с правом только на чтение
-n	Монтирует файловую систему без записи соответствующей строки в файле fstab
-t тип	Задаёт тип монтируемой файловой системы.
-a	Монтирует все файловые системы, перечисленные в файле /etc/fstab
-o список_опций	Монтирует файловую систему с учетом списка опций.

Чтобы к файловой системе был возможен доступ, она должна быть смонтирована. При инсталляции Linux и создании на жестком диске раздела Linux система автоматически конфигурируется на монтирование основных файловых систем при каждом запуске. Для дискет и компакт-дисков такая возможность не предусмотрена, и их придется монтировать явно.

```
$ mount /dev/fd0 /mydir
```

Перед тем как останавливать систему, необходимо демонтировать все смонтированные файловые системы. Основные файловые системы демонтируются автоматически. Если, однако, вы хотите заменить смонтированную файловую систему другой, сначала нужно Демонтировать первую явно. Скажем, вы смонтировали файлы, содержащиеся на дискете, а теперь хотите заменить ее на другую. Для этого нужно сначала демонтировать файловую систему установленной дискеты. Файловая система демонтируется командой `umount`. В качестве аргументов эта команда использует имя устройства и каталог, в котором оно было смонтировано. Вот синтаксис команды `umount`:

```
# umount устройство точка_монтирования
```

В следующем примере демонтируется гибкий диск, смонтированный в каталоге `/mydir`:

```
$ mount /dev/fd0 mydir
```

Демонтировать файловую систему можно было бы, воспользовавшись именем этого каталога:

```
$ umount /mydir
```

Для команды `umount` установлено одно существенное ограничение. Нельзя демонтировать файловую систему, в которой вы работаете в настоящий момент. Если вы перейдете в какой-либо каталог файловой системы и затем попытаетесь демонтировать ее, то получите сообщение об ошибке — вас уведомят о том, что файловая система занята.

## Монтирование дискет

При монтировании и демонтировании устройств их имена и имена каталогов можно указывать явно, как это делается во всех версиях Linux. Имя устройства для дисководов гибких дисков — `fd0`, имя каталога — `/dev`. Таким образом, дисковод гибких дисков обозначается как `/dev/fd0`. В принципе, монтировать устройство можно в любой каталог, но в версии Red Hat уже создан удобный каталог для файлов дискет, `/mnt/floppy`. В следующем примере файлы дискеты монтируются в этот каталог.

```
$ mount /dev/fd0 /mnt/floppy
```

Помните о том, что вы монтируете не дисковод, а конкретную дискету. Просто вынуть ее и вставить другую нельзя. Команда `mount` присоединила находящиеся на дискете файлы к основному дереву каталогов, и система надеется найти их на дискете, установленной в дисководе. Вынув одну дискету и вставив другую, вы при попытке доступа к ней получите сообщение об ошибке.

Для того чтобы поменять дискету, сначала нужно демонтировать файловую систему дискеты, вставленной в дисковод. Затем следует вставить новую дискету и явно смонтировать ее файлы. Для демонтирования используется команда `umount`.

```
$ umount /dev/fd0
```

В операции `umount` можно указать либо каталог, в котором смонтирован гибкий диск, либо устройство `/dev/fd0`.

```
$ umount /mnt/floppy
```

Теперь вы можете извлечь дискету, вставить новую и затем смонтировать ее.

```
$ mount /mnt/floppy
```

При останове системы производится автоматическое демонтирование всех смонтированных дисков. Явно демонтировать их не нужно.

### **Монтирование компакт-дисков**

В Red Hat каталог /mnt/cdrom зарезервирован для файловых систем компакт-писков. Строка для этого каталога имеется в файле /etc/fstab. Чтобы смонтировать компакт-диск, достаточно ввести команду mount и имя каталога /mnt/cdrom. Имя устройства указывать не надо. После монтирования можно обращаться к компакт-диску через каталог /mnt/cdrom.

```
$ mount /mnt/cdrom
```

Чтобы поменять диск, сначала нужно демонтировать компакт-диск, установленный в дисковом, с помощью команды umount. Вот как выполняется демонтирование диска, который вы хотите вынуть из дискового:

```
$ umount /mnt/cdrom
```

Если вы желаете смонтировать компакт-диск в другом каталоге, необходимо указать в команде mount имя устройства. Ниже приведен пример, в котором диск, находящийся в дисковом, монтируется в каталоге /mydir. В данном примере для компакт-диска файл устройства имеет имя /dev/hdc.

```
$ mount /dev/hdc /mydir
```

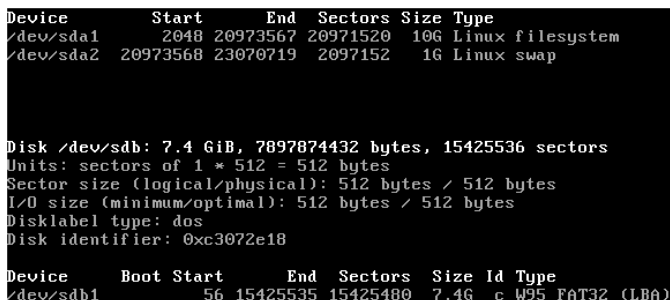
## Монтирование флешки в linux

Самое первое, что нужно сделать после того, как вы подключили флешку — узнать имя ее файла в системе.

Так как флешка подключается последней у нее будет старшая буква. Например, sdb или sdc. Имя sda принадлежит жесткому диску. Можно, конечно, просто посмотреть содержимое каталога /dev/, но такой способ ненадежный. Мы воспользуемся утилитой fdisk.

Подключите флешку и выполните:

```
$ fdisk -l
```



```
Device            Start       End   Sectors  Size Type
/dev/sda1          2048 20973567 20971520   10G Linux filesystem
/dev/sda2          20973568 23070719 2097152    1G Linux swap

Disk /dev/sdb: 7.4 GiB, 7897874432 bytes, 15425536 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device      Boot Start      End   Sectors  Size Id Type
/dev/sdb1             56 15425535 15425480   7.4G  c W95 FAT32 (LBA)
```

Рис.1. Информация о флешке

Как видите, здесь мы можем посмотреть детальную информацию об устройстве. Имя файла, список разделов, формат таблицы разделов, список разделов. А главное для нас, размер раздела и его файловую систему. Теперь не сложно понять какая из них флешка. В этом примере это /dev/sdb1. Создаем папку для монтирования.

Теперь монтируем флешку с помощью команды mount:

```
$ mount /dev/sdb1 /mnt/usb
```

После завершения работы с флешкой не забудьте ее [демонтировать](#). Потому что иначе данные могут быть не сохранены или вообще повреждена файловая система флешки:

```
$ umount /dev/sdb1
```

## Автоматическое монтирование файловых систем: `/etc/fstab`

Хотя монтирование файловой системы можно выполнять при помощи команды [mount](#), вы можете упростить этот процесс, если поместите команды монтирования в файл конфигурации `/etc/fstab`. Для того чтобы ОС Linux автоматически монтировала файловую систему, вам нужно просто добавить имя соответствующего раздела жесткого диска в файл `fstab`. Это можно сделать, непосредственно редактируя файл `/etc/fstab`, путем ввода в него новой команды.

Элемент файла `fstab` содержит несколько полей, разделенных пробелами или знаками табуляции. Первое поле — имя монтируемой файловой системы. Оно обычно начинается с `/dev`, например, `/dev/hda3`, — третий раздел жесткого диска. Следующее поле — каталог в файловой структуре, к которому вы хотите присоединить файловую систему, Находящуюся на данном устройстве. Третье поле — тип монтируемой файловой системы. Перечень типов файловых систем приведен в [табл. 1](#). Тип раздела жесткого диска для стандартной файловой системы Linux — `ext2`. В следующем примере показана строка файла `fstab`, соответствующая первичному разделу жесткого диска для Linux. Он монтируется в корневом каталоге (`/`) и имеет тип `ext2`.

```
/dev/hda3 / ext2 defaults 0 1
```

В поле, следующем после типа файловой системы, указываются различные опции монтирования (табл. 7). Имеется стандартный набор опций, устанавливаемых по умолчанию; их можно задать вводом одной опции `defaults`. Другие опции в списке разделяются запятыми (без пробелов). Опция `defaults` обозначает устройство как устройство чтения/записи, асинхронное, блок-ориентированное, без возможности монтирования для обычных пользователей, с возможностью выполнения на нем программ. Для CD-ROM указываются всего две опции, `ro` и `noauto`. Опция `ro` показывает, что файлы, находящиеся на этом устройстве, предназначены только для чтения, `noauto` — что оно

автоматически не монтируется. Опция `noauto` используется как для CD-ROM, так и для дискет, чтобы они не монтировались автоматически, поскольку вы не знаете, будут ли они установлены при запуске. Ниже приводится пример записей для CD-ROM и дискет. Тип файловой системы на CD-ROM, `iso9660`, отличается от типа файловой системы, размещаемой на жестком диске. Для дисководов гибких дисков также предусмотрены все стандартные опции, используемые для разделов жесткого диска.

```
/dev/fd0 /mnt/floppy ext2 defaults,noauto 0 0
/dev/hdc /mnt/cdrom iso9660 ro,noauto 0 0
```

**Таблица 6.** Опции монтирования файловых систем, используемые в команде `mount` (при установленной опции `-o`) и в файле `/etc/fstab`

Опция	Описание
<code>async</code>	Весь ввод/вывод файловой системы должен выполняться асинхронно
<code>auto</code>	Возможно монтирование с опцией <code>-a</code>
<code>defaults</code>	Использовать опции, применяемые по умолчанию: <code>rw</code> , <code>suid</code> , <code>dev</code> , <code>exec</code> , <code>auto</code> , <code>nouser</code> и <code>async</code>
<code>dev</code>	Файлы байт-ориентированных и блок-ориентированных устройств в файловой системе интерпретируются как специальные файлы
<code>noauto</code>	Файловая система может монтироваться только явно. Опция <code>-a</code> не приведет к автоматическому монтированию файловой системы
<code>exec</code>	Разрешить выполнение двоичных файлов
<code>nouser</code>	Запретить обычным (непривилегированным) пользователям монтирование файловой системы

Продолжение таблицы 6

<b>Опция</b>	<b>Описание</b>
remount	Попытаться повторно смонтировать уже смонтированную файловую систему. Эта опция часто используется для изменения опций монтирования файловой системы, и в частности для того, чтобы разрешить запись для файловой системы, ранее смонтированной только для чтения
ro	Монтировать файловую систему с доступом только для чтения
rw	Монтировать файловую систему с доступом для чтения и для записи
suid	Разрешить использование битов смены идентификатора пользователя (SUID) и смены идентификатора группы (SGID)
sync	Весь ввод/вывод файловой системы должен выполняться в синхронном режиме
user	Разрешить обычным (непривилегированным) пользователям монтирование файловой системы. Для обычных пользователей всегда активизируются следующие опции: noexec, nosuid и nodev
nodev	Блокировать специальные устройства в файловой системе
nosuid	Запретить использование битов смены идентификатора пользователя (SUID) и смены идентификатора группы (SGID)

Ниже приведена копия файла `/etc/fstab`. Первая его строка — комментарий. Все строки комментариев начинаются со знака `#`. Файловая система `/proc` — это специальная файловая система, которую операционная система Linux использует для управления системными процессами. Никакому реальному устройству она не соответствует.



Для того чтобы сделать запись в файле `/etc/fstab`, вы можете просто отредактировать его.

```
#<<device>> <<mountpoint>> <<filesystemtype>> <<options>>
<<dump>> <<fack>>
/dev/hda3 / ext2 defaults 0 1
/dev/hdc /mnt/cdrora iso9660 ro,noauto 0 0
/dev/fd0 /mnt/floppy ext2 defaults,noauto 0 0
/proc /proc proc defaults
/dev/hda2 none swap sw
/dev/hdal /mnt/windows vfat defaults 0 0
```

В файловую структуру Linux можно монтировать разделы MS-DOS и Windows 95, используемые операционными системами MS-DOS и Windows. Делается это так же как монтирование любой файловой системы Linux. Нужно только указать тип раздела `msdos` — для MS-DOS или `vfat` — для Windows 95.

```
/dev/hdal /mnt/windows vfat defaults 0 0
```

Файловые системы, перечисленные в файле `/etc/fstab`, автоматически монтируются при каждой загрузке системы, если вы явным образом не укажете обратное, задав для соответствующего устройства аргумент `noauto`. Если вы хотите, чтобы рядовой пользователь мог монтировать компакт-диск, задайте опцию `user`.

```
/dev/hdc /mnt/cdrom iso9660 ro,noauto,user 0 0
```

## Свободная память

Команда **free** выводит информацию о свободной и занятой памяти. Общий объем памяти изменяется в блоках (1024 байта):

```
$ free
```

## АРХИВАЦИЯ ДАННЫХ

Утилита **tar** предназначена для создания [архивов файлов и каталогов](#). С помощью данной программы можно архивировать файлы, обновлять их в архиве и вводить в этот архив новые файлы. Можно архивировать и целые каталоги со всеми их файлами и подкаталогами. При необходимости все эти файлы и подкаталоги можно восстановить из архива. Первоначально программа tar предназначалась для создания архивов на лентах, отсюда и название tar (tape archive, то есть «архив на ленте»).

При создании файла для tar-архива к имени этого файла обычно добавляется расширение .tar. Это условное обозначение; оно не обязательно. В команде можно указать сколько угодно имен файлов. Если указано имя каталога, то в архив включаются и все его подкаталоги.

# tar опции f имя\_архива.tar имена\_файлов и каталогов

Для создания архива используется опция с. В сочетании с опцией f опция с приводит к созданию архива в файле или на устройстве. Эта опция ставится непосредственно перед опцией f. Обратите внимание на то, что дефиса перед опцией нет. В табл. 7. перечислены все опции, которые можно использовать с программой tar. В следующем примере в каталоге mydir содержатся два файла, mymeeting и party, а также катало reports с тремя файлами: weather, monday и Friday; все они будут заархивированы.

```
$ tar cvf myarch.tar mydir  
mydir/  
mydir/reports/  
mydir/reports/weather  
mydir/reports/monday  
mydir/reports/friday
```

mydir/mymeeting  
mydir/party

**Таблица 7.** Опции команды tar

<b>Команда или опция</b>	<b>Назначение</b>
tar <i>опции</i> <i>файлы</i>	Создает резервные копии файлов в архивном файле, на ленте или на другом устройстве
tar <i>опции</i> f <i>имя архива</i> <i>список_файлов</i>	Создает резервные копии файлов в файле <i>имя_архива</i> ; <i>список_файлов</i> может содержать имена файлов и каталогов
c	Создает новый архив
t	Выдает список файлов, имеющихся в архиве
r	Добавляет файлы в архив
u	Обновляет архив новыми и исправленными файлами. Добавляет только те файлы, которые изменялись после последнего архивирования, и файлы, которые в архиве отсутствуют
w	Ожидает от пользователя подтверждения на архивирование каждого файла; позволяет обновлять архив избирательно
x	Извлекает файлы из архива
m	Извлекая файл из архива, не изменяет дату его последней модификации
M	Создает многотомный архив, который может храниться на нескольких носителях
f <i>имя_архива</i>	Записывает архив не на устройство по умолчанию, а в файл <i>имя_архива</i> ; если <i>имя_архива</i> задано, опция f указывает записывать tar-архив в файл с этим именем

Продолжение таблицы 7

Команда или опция	Назначение
f <i>имя_устройства</i>	Записывает tar-архив на устройство, например на дискету или ленту; /dev/fd0— имя устройства для дискет; имя устройства по умолчанию содержится в файле/etc/default/tar
v	Отображает имя каждого архивируемого в данный момент файла
z	Осуществляет сжатие и распаковку архивированных файлов с помощью программы gzip

Пользователь может извлекать каталоги и файлы из архива, заданного по умолчанию, применяя команду tar с опцией x. Опция xf позволяет извлечь каталоги и файлы из конкретного архива. При их извлечении формируются и все подкаталоги. В следующем примере посредством опции xf команде tar дается указание извлечь все файлы и подкаталоги из файла myarch.tar.

```
$ tar xvf myarch.tar
mydir/
mydir/reports/
mydir/reports/weather
mydir/reports/monday
mydir/reports/friday
mydir/mymeeting
mydir/party
```

Для добавления файлов в существующий архив используется опция r. В приведенном ниже примере пользователь добавляет файлы из каталога letters в архив myarch.tar. В результате в архив добавляется каталог mydocs и все имеющиеся в нем файлы:

```
$ tar rvf myarch.tar letters
mydocs/
mydocs/doc1
```

Если нужно заменить какой-либо файл в архивированных ранее каталогах, можно с помощью опции `u` дать программе `tar` указание обновить архив, заменив модифицированные файлы их новыми версиями. Программа сравнивает время последнего изменения каждого архивированного файла и соответствующего файла в каталоге и копирует в архив все файлы с более поздней датой модификации. В архив будут добавлены и все вновь созданные в этих каталогах файлы. В следующем примере пользователь обновляет файл `myarch.tar`, заменяя в нем все измененные файлы каталога `mydir` и добавляя новые файлы, созданные после создания архива.

```
$ tar uvf myarch.tar mydir
mydir/
mydir/gifts
```

Если вы хотите посмотреть, какие файлы хранятся в архиве, дайте команду `tar` с опцией `t`. В следующем примере показано, как с помощью этой команды можно получить список всех файлов, хранящихся в архиве `myarch.tar`.

```
$ tar tvf myarch.tar
drwxr-xr-x root/root 0 2000-10-24 21:38:18 mydir/
drwxr-xr-x root/root 0 2000-10-24 21:38:51 raydir/reports
-rw-r--r-- root/root 22 2000-10-24 21:38:40 raydir/reports/weather
-rw-r--r-- root/root 22 2000-10-24 21:38:45 mydir/reports/monday
-rw-r--r-- root/root 22 2000-10-24 21:38:51 mydir/reports/friday
-rw-r--r-- root/root 22 2000-10-24 21:38:18 mydir/mymeeeting
-rw-r--r-- root/root 22 2000-10-24 21:38:42 mydir/party
drwxr--r-x root/root 0 2000-10-24 21:38:45 mydocs/
```

```
-rw-r--r--  root/root 22 2000-10-24 21:38:45 mydocs/doc1  
drwxr--r-x  root/root 0 2000-10-24 21:38:03 mydir/  
-rw-r--r--  root/root 22 2000-10-24 21:38:03 mydir/gifts
```

При использовании команды `tar` операция сжатия архивных файлов не выполняется. Если вы хотите сжать файлы, дайте `tar` указание вызвать утилиту `gzip`. Если команда `tar` применяется с опцией `z`, то сначала программа `gzip` выполняет сжатие, а затем `tar` архивирует файлы. Та же опция `z` обеспечит вызов `gzip` для распаковки файлов при извлечении их из архива.

```
$ tar czf myarch.tar mydir
```

## СЖАТИЕ ДАННЫХ

Существует несколько утилит [сжатия](#), которые могут применяться в системах Linux UNIX. Большая часть программного обеспечения ОС Linux использует утилиты GNU **gzip** и **gunzip**. Утилита **gzip** сжимает файл, а **gunzip** - распаковывает. Для того чтобы сжать файл, введите команду **gzip** и имя этого файла. В результате файл будет заменен его сжатой версией с расширением **.gz**.

```
$ gzip mydata
$ ls
mydata.gz
```

**Таблица 8.** Опции команды **gzip**

Опция	Назначение
-c	Посылает сжатую версию файла на стандартный вывод; каждый указанный файл сжимается отдельно  \$ gzip -c raydata preface >> myfiles.gz
-d	Распаковывает сжатый файл; можно также использовать команду <b>gunzip</b>  \$ gzip -d myfiles.gz \$ gunzip myfiles.gz
-h	Выдает перечень справочной информации
-l список_файлов	Выдает размер каждого из указанных файлов (в сжатом и несжатом виде)  \$ gzip -l myfiles.gz

Продолжение таблицы 8

Опция	Назначение
-t имя_каталога	Производит рекурсивную обработку указанных каталогов и сжимает в них все файлы. Обработка начинается с текущего рабочего каталога. При использовании этой опции с программой gunzip сжатые файлы из указанного каталога будут распакованы
-v список_файлов	Сообщает степень сжатия в процентах по каждому обработанному файлу
-число	<p>Определяет скорость и степень сжатия. Диапазон чисел - от -1 до -9.</p> <p>Меньшее число означает более высокую скорость, но меньшую степень сжатия - в итоге получается большой файл, который быстро сжимается и распаковывается. Число -1 означает самое быстрое сжатие, но и максимальный размер. Число -9 дает очень маленький файл, который сжимается и распаковывается медленно. По умолчанию принимается -6</p>

Для распаковки gzip-файла введите либо команду gzip с опцией -d, либо команду gunzip. Полный перечень опций команды gunzip приведен в табл. 8.

```
$ gunzip mydata.gz
```

```
$ ls
```

```
mydata
```

Можно сжимать и архивированные файлы. Эта операция дает в результате файлы с расширением «.tar.gz.». Сжатые архивированные



файлы часто используются для передачи очень больших файлов по сетям.

```
$ gzip myarch.tar
```

```
$ ls
```

```
myarch.tar.gz
```

Другой популярной утилитой для сжатия файлов является **bzip2**. Для сжатия файлов в ней используется алгоритм сжатия текста с сортировкой по блокам Берроуза-Вилера (Burrows-Wheeler) и кодирование Хоффмана (Huffman). Умышленно сделано так, что параметры командной строки очень похожи на параметры `gzip`, однако они не совпадают полностью. Вы можете сжимать файлы при помощи команды `bzip2` и восстанавливать их при помощи `bunzip2`. В результате выполнения команды `bzip2` создаются файлы с расширением `.bz2`.

**Таблица 9.** Основные опции программы `bzip2`

Опция	Значение
-d, --decompress	Принудительная разархивация. Эта опция необходима в силу того, что <code>bzip2</code> , <code>bunzip2</code> и <code>bzcat</code> — это на самом деле одна и та же программа, которая сама по расширению имени файла принимает решение о том, какое действие надо выполнить над указанным файлом. Опция <code>-d</code> отключает этот механизм и заставляет программу разархивировать указанные файлы
-z, --compress	Принудительная архивация

Продолжение таблицы 9

Опция	Значение
-t, --test	Проверка целостности указанного файла(ов) без разархивации
-f, --force	Перезапись существующего файла. По умолчанию bzip2 не перезаписывает существующие файлы. Если вы хотите перезаписать существующий файл, надо задать опцию -f
-k, --keep	Сохранять (не удалять) исходные файлы при архивации или разархивации
-s, --small	Снижает требования к объему используемой оперативной памяти за счет снижения скорости архивации. Эту опцию рекомендуется применять на компьютерах с малым объемом ОЗУ (8 Мбайт и меньше)
-q, --quiet	Не выводить малосущественные сообщения
-v, --verbose	Выводить дополнительную информацию в процессе работе (представляет интерес в диагностических целях)
-L, --license, -V, --version	Отобразить версию программы и лицензионное соглашение

В следующем примере файл mydata сжимается и помещается в архив mydata.bz2

```
$ bzip2 mydata
$ ls
mydata.bz2
```

Для распаковки bzip-файла применяйте команду bunbzip2.

```
$ bunbzip2 mydata.bz2
```

**Zip** является кроссплатформенной утилитой и используется в операционных системах Windows, Mac, MSDOS, OS/2, UNIX и Linux. Чтобы сжать каталог, нужно включить в команду параметр -r. В первом примере архивируется файл:

```
$ zip mydata
```

```
$ ls
```

```
mydata.zip
```

В следующем примере архивируется каталог:

```
$ zip -r reports
```

Для восстановления и извлечения Zip-файла можно использовать команду unzip.

```
$ unzip mydata.zip
```

## ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Научиться использовать команды для монтирования и демонтирования различных устройств, архивации и сжатия файлов в ОС Linux. Изучить расположение и назначение системных каталогов.

Продемонстрировать выполнение команд:

1. Для отображения свободного пространства в файловых системах
2. Монтирования и демонтирования компакт-дисков
3. Монтирования и демонтирования флеш-накопителя
4. Работы с файлами в смонтированной файловой системе
5. Автоматического монтирования устройств
6. Вывести информацию об использовании оперативной памяти
7. Создания архива утилитой tar (использовать различные опции при создании архива)
8. Работы с архивом (вывести список файлов, имеющихся в архиве, добавить файлы в архив, обновить архив новыми и исправленными файлами, извлечь файлы из архива)
9. Создания и распаковку архива утилитой gzip (использовать различные опции)
10. Создания и распаковку архива утилитой bzip2 (использовать различные опции)

## КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Опишите организацию файлов в ОС Linux.
2. Перечислите файловые системы, используемые в ОС Linux.
3. Раскройте понятие «FHS (filesystem hierarchy standard)».
4. Перечислите каталоги, хранящиеся в корневом каталоге системы.
5. Перечислите префиксы имен устройств.
6. Предложите вариант команд для монтирования и демонтирования устройств в ОС Linux.
7. Перечислите опции команды mount.
8. Назовите каталог, в котором хранятся файлы устройств.
9. Назовите назначение файла «/etc/fstab».
10. Перечислите опции монтирования файловых систем, используемые в команде mount (при установленной опции -o) и в файле /etc/fstab.
11. Предложите способ автоматического монтирования файловой системы.
12. Опишите понятие сжатия файлов.
13. Раскройте понятие архивации файлов.
14. Опишите понятие степень сжатия.
15. Опишите функции утилиты tar.
16. Перечислите опции команды tar.
17. Предложите команду для просмотра содержимого архива.
18. Назовите опции команды gzip.
19. Предложите варианты утилит для сжатия файлов.
20. Предложите вариант сжатия архивированного файла.

## ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 2 занятия (4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, ответы на контрольные вопросы, описание процесса выполнения лабораторной работы, выводы.

## ОСНОВНАЯ ЛИТЕРАТУРА

1. Вирт, Н. Разработка операционной системы и компилятора. Проект Оберон [Электронный ресурс] / Н. Вирт, Ю. Гуткнехт ; пер.с англ. Борисов Е.В., Чернышов Л.Н.. — Электрон. дан. — Москва : ДМК Пресс, 2012. — 560 с. — Режим доступа: <https://e.lanbook.com/book/39992>

## ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

2. Крищенко, В.А. Сервисы Windows [Электронный ресурс] : учебное пособие / В.А. Крищенко, Н.Ю. Рязанова. — Электрон. дан. — Москва : МГТУ им. Н.Э. Баумана, 2011. — 47 с. — Режим доступа: <https://e.lanbook.com/book/52416..>

3. Войтов, Н.М. Администрирование ОС Red Hat Enterprise Linux. Учебный курс [Электронный ресурс] : учебное пособие / Н.М. Войтов. — Электрон. дан. — Москва : ДМК Пресс, 2011. — 192 с. — Режим доступа: <https://e.lanbook.com/book/1081>

4. Стащук, П.В. Администрирование и безопасность рабочих станций под управлением Mandriva Linux: лабораторный практикум [Электронный ресурс] : учебно-методическое пособие / П.В. Стащук. — Электрон. дан. — Москва : ФЛИНТА, 2015. — 182 с. — Режим доступа: <https://e.lanbook.com/book/70397>

### Электронные ресурсы:

5. Научная электронная библиотека <http://eLIBRARY.RU>
6. Электронно-библиотечная система <http://e.lanbook.com>
7. Losst - Linux Open Source Software Technologies <https://losst.ru>