# Лабораторная работа 7. Монтирование

**Аннотация:** Цель работы: изучить специальные команды файловых систем, в том числе команды монтирования в Linux на примере операционной системы Ubuntu.

**Задание 1.** Изучите назначение и синтаксис команды *fdisk*.

Ход выполнения:

1. *fdisk* (сокращение от “format disk“, форматировать диск) – это наиболее распространенная утилита командной строки для работы с дисками в Linux/Unix системах. При помощи этой команды можно просматривать, создавать, удалять, изменять, копировать и перемещать разделы на жёстком диске, используя простой интерфейс текстовых меню.

Для просмотра всех доступных разделов используется опция -*l* (listing – перечисление). Разделы перечисляются по именам, например, /dev/sda, /dev/sdb или /dev/sdc.

2. Выведите на экран все доступных разделы дисков.

Для выполнения данного задания необходимо иметь права пользователя. Для этого нужно либо зарегистрироваться в системе под учетной записью *root*:

*su root*

а затем ввести пароль пользователя *root*, который был задан при установке операционной системы.

Либо перед командами, на которые недостаточно прав обычного пользователя вводить команду *sudo*, а затем уже пароль пользователя.

Выполняя команды с правами администратора, будьте очень внимательны, так как ошибки или необдуманные действия могут привести к неожиданным результатам и даже к выходу из строя операционной системы.

*sudo fdisk –l*

3. Определите, какой раздел является загрузочным.

**Задание 2.** Изучите назначение и синтаксис команды *blkid.*

Ход выполнения:

1. Во многих дистрибутивах Linux разделы называются не по «именам», скажем, /dev/hda2, а используя UUID'ы - Уникальные Идентификаторы. Выглядят они как многозначные шестнадцатеричные номера, например: UUID="5a179614-0415-48c6-a9ad-3f6ad9596619".

Программа *blkid*, используя библиотеку *libblkid*, умеет читать содержимое файлов блочных устройств. Эти специальные файлы, находящиеся в директории */dev*, имеют весьма специфичную структуру и не могут быть прочитаны обычными средствами (скажем, командой *[cat](http://rus-linux.net/lib.php?name=/MyLDP/consol/HuMan/cat-ru.html)*).

Однако программа *blkid* по умолчанию не считывает информацию о разделах непосредственно из соответствующих файлов устройств, а считывает ее из файла */etc/blkid.tab*. В файле */etc/blkid.tab* со временем накапливается устаревшая информация о давно несуществующих разделах.

Параметр *-c* заставляет *blkid* читать не из файла *blkid.tab*, а из любого указанного файла. Скажем, /dev/null. Не имея никакой информации из данного файла, команда вынуждена будет прочитать информацию из файла устройств, то есть выдать нам реальную картину на текущий момент

2. Выполните команду *blkid* с правами администратора. Какая информация появилась в терминале?

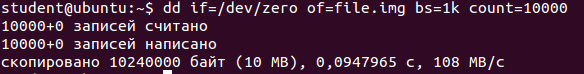
3. Выполните команду *blkid* с параметром *-c* с правами администратора.

Поменялся ли результат выполнения команды от использования параметра?

**Задание 3.** Создайте файловую систему в файле, расположенном в существующей файловой системе, чтобы продемонстрировать возможности уровня файловой системы Linux (и монтирования).

Ход выполнения:

1. Создайте файл заданного размера с помощью команды *dd* путем копирования файла из источника */dev/zero*– другими словами, инициализируя файл нулями, как показано на рисунке 1.



*Рис. 1*. Создание инициализированного файла

В результате выполнения команды создан файл file.img размером 10 МБ.

2. Свяжем с файлом блочное устройство-заглушку (loop) с помощью команды *losetup* (чтобы он выглядел как блочное устройство, а не как обычный файл файловой системы):

*losetup /dev/loop0 file.img*

Если вы попытаетесь выполнить данную команду с правами обычного пользователя, то увидите сообщение о том, что вам отказано в доступе. Для решения этой проблемы используйте один из вариантов, перечисленных в п.1 данного задания. Например, чтобы выполнить команду с правами администратора, не регистрируясь при этом под пользователем *root*, введите:

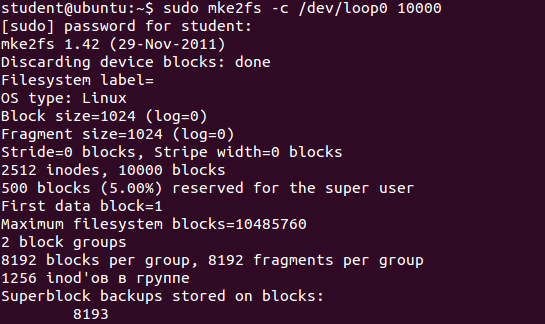
*sudo losetup /dev/loop0 file.img*

На экране появится приглашение ввести пароль пользователя (рис. 2):



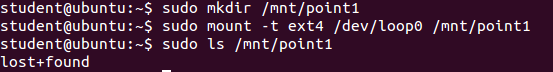
*Рис. 2.* Установка связи между файлом и блочным устройством-заглушкой

3. Создайте на блочном устройстве файловую систему определенного размера с помощью команды *mke2fs* (рис. 3).



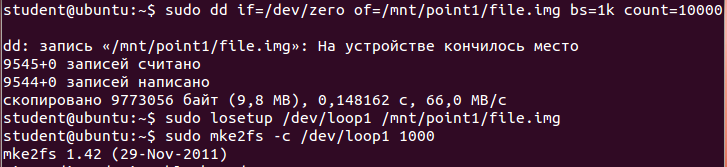
*Рис. 3*. Создание файловой системы на устройстве loop

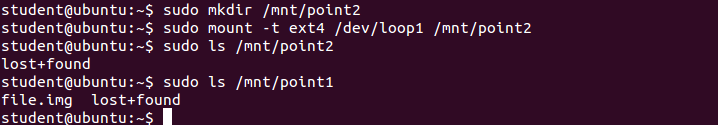
4. Примонтируем файл *file.img*, представленный блочным устройством (*/dev/loop0*) к точке */mnt/point1* с помощью команды *mount*. Обратите внимание, что указанный тип файловой системы – ext4. После монтирования вы можете обращаться к точке монтирования как к новой файловой системе с помощью команды *ls*, как видно из рисунка 4.



*Рис. 4.* Создание точки монтирования и монтирование файловой системы

5. Как показано на рисунке 5, этот процесс можно продолжить, создавая новый файл в новой файловой системе, связывая его с устройством *loop* и создавая в нем еще одну файловую систему.





*Рис. 5*. Создание новой файловой системы *loop* в уже существующей

Из рассмотренного примера легко понять, насколько большие возможности предоставляет файловая система и устройство *loop* в Linux. Аналогичным образом с помощью устройства *loop* можно создавать в файле файловые системы с шифрованием. Это может быть полезно для защиты ваших данных. В случае необходимости такой файл можно быстро смонтировать с помощью устройства *loop*.

6. Изменилось ли что-то в списках разделов диска системы и уникальных идентификаторов разделов?

7. Создайте файл *file1* в созданной файловой системе.

8. Выполните команду:

*sudo mount -t ext4 -o remount,ro /dev/loop0 /mnt/point1*

9. Создайте файл *file2* в созданной файловой системе. Что произошло c файловой системой после выполнения последней команды *mount*?

10. Что надо сделать, чтобы создать *file2*?

**Задание 4.** Выполните автоматическое монтирование файловой системы Linux.

Ход выполнения:

1. Ознакомьтесь c назначением конфигурационного файла */etc/fstab*.

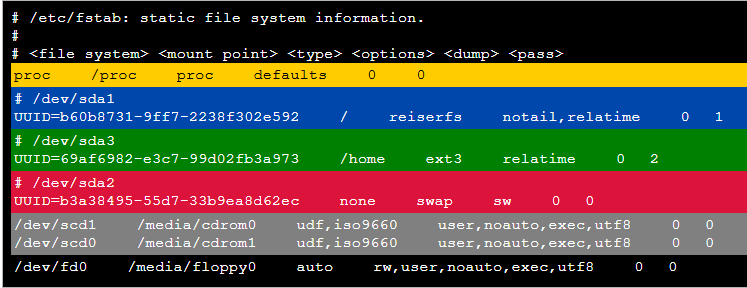
Монтирование может быть выполнено вручную с помощью команды *mount*. Но для начала работы ОС нужно во время загрузки подключить все важные разделы с системными файлами, проверить их файловые системы на ошибки и сделать их готовыми к работе.

Все эти задачи выполняет система инициализации. Но сама система инициализации не знает куда монтировать тот или иной раздел. Для получения этой информации она использует конфигурационный файл */etc/fstab*.

Файл *fstab* - это текстовый файл, который содержит информацию о различных файловых системах и устройствах хранения информации в вашем компьютере. Этот файл можно открыть в любом текстовом редакторе, но редактировать его возможно только от имени [суперпользователя](https://help.ubuntu.ru/wiki/%D1%81%D1%83%D0%BF%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C_%D0%B2_ubuntu), т.к. файл является важной, неотъемлемой частью системы, без него система не загрузится.

1. Посмотрите содержимое файла *fstab.*

Файл *fstab* состоит из строк. Каждая строка – это устройство. Символ решетки (#) в начале строки, как и во всех Unix системах, обозначает комментарий и поэтому значимой строкой не считается.



Желтая строка монтирует виртуальную файловую систему *procfs* к директории */proc*. Это стандартная процедура ОС, поэтому лучше ее не трогать.

Синяя строка присоединяет корневой раздел с параметрами *notail,relatime* (значение параметров будет рассмотрено ниже). Это тоже лучше не трогать.

Зеленая строка монтирует раздел */home* с параметром *relatime*.

Красная монтирует *SWAP* раздел.

Серые строки задают параметры *user,noauto,exec,utf8* для ручного (параметр *noauto*) монтирования CD/DVD-приводов.

Черные строки задают параметры ручного монтирования floppy-диска.

Все строки обладают одинаковым числом блоков. Каждый блок в строке отделен минимум одним пробелом.



В желтом поле находятся названия или универсальные идентификаторы устройств.

В синем столбике отображены точки монтирования. Точка монтирования — это директория, где нужно искать данное устройство. В зеленом столбике описаны типы файловых систем.

В красном столбике находятся параметры монтирования. Если параметров несколько, то они перечисляются через запятую без пробелов.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Параметр** | **Действие** | **Значение по умолчанию** |
| **1** | exec | Разрешить запуск исполняемых файлов. | включена |
| noexec | Запретить запуск исполняемых файлов | – |
| **2** | auto | Раздел будет автоматически монтироваться при загрузке операционной системы. | включена |
| noauto | Раздел не будет автоматически монтироваться при загрузке операционной системы. | – |
| **3** | rw | Выставить права доступа на чтение и запись. | включена |
| ro | Выставить права доступа только на чтение. | – |
| **4** | nouser | Запретить простым пользователям монтировать/демонтировать устройство. | включена |
| user | Разрешить простым пользователям монтировать/демонтировать устройство. | – |
| **5** | sw или swap | Специальный параметр SWAP области | – |
| **6** | async | Включение опции асинхронного ввода/вывода. Любая операция (копирование файла, удаление и т.д.) будет происходить немного позже, чем дана команда. Помогает в распределении нагрузки ОС, последняя сама выбирает подходящее время. | включена |
| sync | Включение опции синхронного ввода/вывода. Любая операция происходит синхронно с командой. | – |
| **7** | suid | Разрешить работу SUID и SGID битов. Бит SUID, у исполняемого файла, повышает запустившему пользователю права до владельца этого файла. К примеру, если root создал исполняемый файл с битом SUID, то пользователь, запустивший этот файл, получает на время исполнения файла права суперпользователя. Бит SGID, у исполняемого файла, повышает запустившему пользователю права до группы владельца этого файла. | – |
| nosuid | Заблокировать работу SUID и SGID битов для устройства. | включена |
| **8** | iocharset=koi8-r codepage=866 | Добавляет поддержку кодировки koi8-r в названиях файлов и директорий. Применять при необходимости. | – |
| **9** | errors=remount-ro | При ошибке перемонтировать с параметром только для чтения (ro). | – |
| **10** | notail | Запрещает хранить маленькие файлы в хвостах больших. Увеличивает быстродействие. | – |
| **11** | atime | Производить запись времени последнего доступа к файлу. | включена |
| noatime | Отключение записи времени последнего доступа к файлу. Увеличивает быстродействие файловой системы. Эта опция не рекомендуется стандартом POSIX, так как некоторые приложения требуют этой функции (к примеру, почтовые клиенты и программы нотификации о новой почте перестанут правильно работать). | – |
| relatime | Включение обновления времени последнего обращения к файлу только в том случае, если предыдущее время доступа было раньше, чем текущее время изменения файла. Это более лояльный подход, чем noatime. | – |
| **12** | defaults | Использование всех параметров по-умолчанию: exec, auto, rw, nouser, async, nosuid, atime | – |

Серое поле указывает на включение/исключение устройства хранения информации в список резервного копирования программы DUMP, если последняя используется (0 – не выполнять резервное копирование; 1 – выполнять резервное копирование).

Черное поле устанавливает порядок проверки раздела на наличие ошибок. Если установить один и тот же порядок для двух разделов, они будут проверяться одновременно (0 – раздел не проверяется; 1 – раздел проверяется первым; 2 – раздел проверяется вторым и т.д).

1. Создайте резервную копию файла *fstab*.
2. Добавьте в файл fstab строку для автоматического монтирования файловой системы, созданной в предыдущем задании. Строку можно добавить как комментарий, так как ошибка в файле fstab может привести к тому, что операционная система не загрузится после перезагрузки.

**Задание 5.** Изучите монтирование устройств к дереву файловой системы Linux.

Ход выполнения:

1. Создайте в домашнем каталоге папку *floppy*. Создайте файловую систему для дисковода гибких дисков. Размер и тип файловой системы должен соответствовать дискете. Выберите тип файловой системы, который можно использовать и в Linux, и в Windows.
2. Примонтируйте в папку floppy созданную файловую систему с помощью команды *mount*.
3. Добавьте в файл fstab строку для автоматического монтирования файловой системы дисковода гибких дисков.
4. Отмонтируйте (демонтируйте) файловую систему дисковода гибких дисков.

Обычно операционная система выполняет демонтирование всех смонтированных файловых систем автоматически при перезагрузке или выключении компьютера. При демонтировании файловой системы все ее кэшированные данные, находящиеся в оперативной памяти, сбрасываются на диск.

Файловые системы можно демонтировать и вручную. Более того, вы должны делать это при извлечении доступных для записи съемных устройств, таких как дискеты или USB-накопители.

Для демонтирования файловой системы выполните команду *umount*, указав в качестве аргумента *либо* имя устройства, *либо* точку монтирования.

**Задание 6.** Изучите команду создания синонима каталога в дереве файловой системы Linux.

Ход выполнения:

1. Ознакомьтесь с назначением и использованием ключа *--bind* команды *mount*.

Команда *mount* с ключом *--bind* применяется в системах на ядре Linux (начиная с 2.4) для монтирования каталога в другой каталог. Такое бывает необходимо для перемещения содержимого в несколько мест файловой системы одновременно.

К примеру, команда:

*mount --bind /mnt/Files /var/ftp*

позволит обращаться к файлам из */mnt/Files* через путь */var/ftp*, где */var/ftp* – некий уже существующий (возможно, пустой) каталог (его настоящее содержимое будет недоступно до момента размонтирования).

Преимуществом данного способа создания ссылок на каталоги над символьными ссылками является возможность обходить ограничения доступа к файловой системе.

2. Создайте каталог *folder1* в домашнем каталоге. Создайте в каталоге *folder1* файл с именем *file1*.

3. Создайте каталог *folder2* в домашнем каталоге. Создайте в каталоге *folder2* файл с именем *file2*.

4. Назначьте разные права доступа к каталогам *folder1* и *folder2*, а также к их содержимому.

5. Примонтируйте каталог *folder2* к каталогу *folder1*. Проверьте содержимое каталогов. Изменилось ли оно, почему?

6. Обратите внимание, изменились ли права доступа к каталогам и файлам.

7. Отмонтируйте каталог *folder2* от каталога *folder1*. Обратите внимание на содержимое каталогов. Сделайте выводы.

**Задание 7.** Откройте доступ виртуальной машине с ОС Linux к папке, находящейся на физическом компьютере (например, на компьютере с ОС Windows).