

## ЛАБОРАТОРНАЯ РАБОТА № 5

### МОНТИРОВАНИЕ ДИСКОВ, КОМАНДА DD

Цель работы: Знакомство с командами по работе с дисками. Создание копии диска, его образа, монтирование и размонтирование файловых систем.

#### Подключение дисков (монтирование)

В ОС Linux содержимому любых устройств с данными, на которых создана файловая система, сопоставляется не отдельный том со своей буквой (как в Windows-системах), а отводится определённое место в существующем дереве каталогов. То есть корневой каталог файловой системы имеется только один, но к нему можно подключать (монтировать) другие каталоги с других томов.

Операция присоединения устройства хранения данных к дереву каталогов называется монтированием, а место присоединения - точкой монтирования. По умолчанию предполагается, что присоединяемые этой операцией данные монтируются в специальный каталог **/mnt**, для чего в нём создают соответствующие каталоги. Выполняют эту операцию с помощью команды **mount**. В общем случае команда **mount** требует двух аргументов - имени монтируемого устройства и точки монтирования. Например, команда

```
$ mount /dev/sdb1 /mnt/common
```

смонтирует первый раздел на втором диске в каталог **common** каталога **/mnt**; очевидно, что он расположен в штатном каталоге **/mnt**. В данном примере тип файловой системы в разделе **/dev/sdb1** не указан. В большинстве случаев он может быть определён автоматически. Но иногда его можно указать в явном виде с помощью опции **-t**. Например, команда могла бы быть такой, если бы на этом диске была файловая система типа FAT32:

```
$ mount -t vfat /dev/sdb1 /mnt/common
```

При необходимости указать кодировку, используемую на присоединяемом том, можно в эту команду добавить такие опции:

```
$ mount -t vfat -o codepage=866,iocharset=utf8 /dev/sdb1 /mnt/common
```

Полную справку о команде **mount** получите с помощью команды **man**

В POSIX-системах существует специальный файл **/etc/fstab**, в котором хранятся настройки монтирования различных разделов, включая корневой каталог файловой системы и swar-раздел, предназначенный для реализации виртуальной памяти (в этом разделе система сохраняет те страницы виртуальных адресных пространств исполняемых процессов, которые сейчас не удаётся разместить в реальной физической памяти). Существенная информация из файла **/etc/fstab** заключается в первую очередь в соответствии точек монтирования разделам (томам). При каждой загрузке Linux читает этот файл и монтирует все перечисленные в нём диски в указанные для них места. Больше нигде никакой информации о

точках монтирования в системе не хранится. Соответственно при установке в **/etc/fstab** записывается вся указанная Вами информация о монтировании Ваших разделов.

В этом файле каждому разделу винчестера отводится одна строка. Строчки, начинающиеся с символа **#**, являются комментариями и системой игнорируются. Формат каждой записи очень простой и использует 6 столбцов:

раздел\_диска    точка\_монтирования    файловая\_система    опции\_монтирования  
опции\_дампа    опции\_проверки\_файловой\_системы

В последнее время вместо собственно указания разделов диска в виде **/dev/sda1**, **/dev/sda2** и т. д. в файле **/etc/fstab** используется несколько другой механизм. Для этой цели используются так называемые **UUID** - достаточно длинные строчки из произвольных букв латинского алфавита и цифр. Для того, чтобы узнать UUID для нужного Вам раздела, выполните в терминале команду

```
ls -l /dev/disk/by-uuid
```

Необходимо отметить, что во-первых, точка монтирования должна существовать, это должен быть пустой каталог, а во-вторых, в записи имени этого каталога не должно быть спецсимволов и пробелов. В случае использования для монтирования не пустых каталогов их содержимое будет недоступно до размонтирования.

Различные флешки, внешние USB-диски, плееры, смартфоны и прочие устройства монтируются автоматически при подключении к вашему компьютеру. Точкой монтирования в этом случае служит один из подкаталогов **/media**, обычно - с именем, соответствующим метке устройства. Так, например, в Ubuntu за выделением точек монтирования, определением файловой системы на присоединяемых устройствах и решением других проблем следит сама система и Вам ничего делать не надо. Кроме того, все внешние устройства автоматически попадают в меню «Переход» (на старых версиях Ubuntu с Gnome-интерфейсом) и на рабочий стол, так что Вы всегда можете легко получить к ним доступ.

Права на запись файлов на подключаемые устройства определяются правами на запись в точку монтирования. То есть если Вы что-то монтируете в **/media/data**, то у вас должны быть нужным образом установлены владелец, группа и права доступа к каталогу **/media/data**.

## Команда dd

Команда **dd** предназначена для задействования одноименной утилиты, предназначенной для низкоуровневого копирования и преобразования данных. Ее название расшифровывается как «data duplicator» или «дубликатор данных». Данная утилита используется главным образом для записи образов установочных дисков дистрибутивов Linux на флеш-накопители и создания образов оптических носителей, тем не менее, круг ее функций не ограничивается перечисленными операциями. Например, **dd** может использоваться для простого поблочного копирования дисков, файлов или изменения регистра символов текстовых строк. Вообще, рассматриваемая утилита является в какой-то мере уникальной, ведь она предполагает использование собственного формата передачи параметров.

Стандартный синтаксис команды выглядит следующим образом:

**\$ dd if=<имя исходного файла> of=<имя целевого файла> [параметры]**

Несложно заметить, что для передачи параметров утилите используется формат записи <имя параметра>=<значение параметра>. Утилита может читать исходные данные из стандартного потока ввода и выводить результирующие данные с помощью стандартного потока вывода в случае отказа от использования параметров **if** и **of**, но в подавляющем большинстве случаев данные параметры являются необходимыми для указания имен файлов с соответствующими данными. Утилита читает и записывает данные блоками, причем размер блока может изменяться с помощью параметра **bs** (по умолчанию используются блоки размером в 512 Кб). Существуют отдельные параметры для задания размеров читаемых и записываемых блоков, а именно, **ibs** и **obs**. Количество читаемых блоков может ограничиваться в помощью параметра **count**. Для пропуска заданного количества блоков исходного файла может использоваться параметр **skip**, целевого файла — параметр **seek**. Для указания флагов чтения и записи через запятую может использоваться параметр **iflag**. Наиболее часто используемыми флагами являются:

- **append** — активация режима дописывания данных в целевой файл.
- **direct** — режим обработки данных в обход кэша файловых систем (повышает скорость).
- **dsync** — режим записи данных с синхронизацией (повышает надежность).
- **sync** — режим записи данных и метаданных с синхронизацией (повышает надежность).
- **fullblock** — чтение лишь полных блоков.
- **nonblock** — активация режима неблокируемого ввода/вывода (повышает скорость).
- **noatime** — отключение механизма обновления меток времени элементов файловой системы (повышает скорость).
- **nofollow** — отказ от перехода по символьным ссылкам.

Наконец, для указания флагов преобразования через запятую может использоваться параметр **conv**. Наиболее часто используемыми флагами являются:

- **lcase** — перевод символов строки в кодировке ASCII в нижний регистр.
- **ucase** — перевод символов строки в кодировке ASCII в верхний регистр.
- **nocreat** — вывод сообщения об ошибке в случае отсутствия целевого файла.
- **excl** — вывод сообщения об ошибке в случае существования целевого файла.
- **notrunc** — отказ от урезания целевого файла.
- **swab** — смена мест каждых двух байтов из исходного файла.
- **noerror** — продолжение работы даже в случае возникновения ошибок.
- **fdatasync** — активация режима записи данных в целевой файл перед завершением работы утилиты.
- **fsync** — активация режима записи данных и метаданных в целевой файл перед завершением работы утилиты.

## Примеры использования команды dd

### Создание резервной копии данных дискового накопителя

Предположим, мы используем жесткий диск, представленный файлом устройства **/dev/sda**, и нам нужно создать посекторную резервную копию всех размещенных на нем данных, сохранив ее в файле в разделе съемного диска с интерфейсом USB, представленном файлом устройства **/dev/sdb1** и смонтированным в директорию **/mnt/sdb1**. Обычно такие файлы резервных копий называются дампами или образами дисков. Наш файл образа диска будет носить имя **backup.img**. Это команда, с помощью которой его можно создать:

```
# dd if=/dev/sda of=/mnt/sdb1/backup.img
```

В данной команде с помощью параметра **if** задается путь к исходному файлу, а с помощью параметра **of** — к целевому.

### Восстановление данных из резервной копии

Для восстановления данных из созданной резервной копии следует загрузить систему с установочного диска дистрибутива и выполнить обратную команду.

```
# dd if=/mnt/sdb1/backup1.img of=/dev/sda
```

***Предупреждение:** при выполнении данной команды будет перезаписано все содержимое указанного жесткого диска, поэтому стоит относиться к подобным командам с особым вниманием.*

### Клонирование жесткого диска

Перед клонированием жесткого диска вы должны убедиться в том, что у вас имеется жесткий диск того же объема, что и исходный. Эта же операция может выполняться и в случае флеш-накопителей с интерфейсом USB аналогичных объемов. Предположим, что исходный флеш-накопитель представлен файлом устройства **/dev/sdb**, а целевой — файлом устройства **/dev/sdc**. В этом случае вы можете клонировать исходный накопитель с помощью следующей команды:

```
# dd if=/dev/sdb of=/dev/sdc
```

Даже в том случае, если целевой накопитель имеет больший объем, вам будет доступен лишь объем исходного флеш-накопителя, сохраненный на уровне файловой системы.

### Создание образа оптического диска формата ISO

Для создания образа оптического диска CD, DVD или BD достаточно поблочно прочитать его содержимое и сохранить это содержимое в файле:

```
# dd if=/dev/sr0 of=image.iso bs=2048
```

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Добавьте в Ваш виртуальный компьютер ещё один жесткий диск объёмом 500 МБ. Эту операцию необходимо выполнить на выключенном виртуальном компьютере. После включения компьютера зайдите в Windows и создайте на новом устройстве ещё один

логический диск, отформатировав его для конкретики в FAT32. Скопируйте на новый диск несколько файлов.

2. Перегрузите систему и зайдите в систему Linux. Обеспечьте, чтобы новый логический диск был теперь доступен через каталог **/mnt/disk2** при каждом запуске системы Linux. Опишите в отчёте выполненные для этого действия.

3. Записи о всех смонтированных дисках попадают в специальный файл **/etc/mtab**, а формат записей этого файла ровно такой же, как и у **/etc/fstab**. Посмотрите на смонтированные файловые системы с помощью команды **cat /etc/mtab**. Опубликуйте информацию, полученную с помощью этой команды, в своём отчёте.

4. Добавьте в свой виртуальный компьютер (напомним, что это можно сделать только на выключенной виртуальной машине) еще один диск и тоже размером 500МБ. С помощью команды **dd** обеспечьте, чтобы он стал точной копией предыдущего диска. Смонтируйте его в каталог **/mnt/disk3** и убедитесь, что там расположены те же файлы, что и в каталоге **/mnt/disk2**

5. Отмонтируйте оба новых диска и с помощью просмотра смонтированных систем убедитесь в этом.