

Clonehdd

Любой из нас хорошенько задумывается над тем, как правильно разбить HDD при установке FreeBSD. Действительно, потом будет весьма проблематично изменить размер партиции при необходимости. Проблема заключается в том, что на жестком диске находятся, так называемые, слайсы, а уже в них инкапсулированы партиции. Это не всегда так, потому что есть еще и второй метод разметки HDD без слайсов, но в данной статье он не рассматривается. Популярные программы для работы с разделами HDD, такие как Partition Magic, Acronis могут удалить слайс, скопировать/переместить его посекторно, но никак не заглянуть внутрь и изменить размер той или иной партиции.

Итак, есть система (далее, система – это установленная ОС FreeBSD, включающая в себя MBR, корневой раздел, SWAP, дополнительные разделы /tmp, /var и т.д.) на жестком диске объемом 80 ГБ.

Задача: перенести систему на другой жесткий диск объемом 250 ГБ, увеличивая размеры партиций пропорционально увеличению объема HDD. В нашем случае это $250-80=170$ ГБ.

Вторая часть задачи: повторять эту процедуру автоматически, с определенным интервалом (для создания резервной копии системы). В обоих случаях, нужно скопировать систему полностью, т.е. не только файлы и каталоги, а и загрузочный сектор, загрузчик для сохранения возможности загрузки системы с полученной копии.

```
# whereis clonehdd clonehdd: /usr/ports/sysutils/clonehdd
# cd /usr/ports/sysutils/clonehdd
# make install clean
```

После установки мы видим грозное предупреждение о том, что запуская clonehdd нужно семь раз отмерить а потом уже отрезать =)) Но все не так страшно, как кажется: программа изначально

рассчитана так, что данные на исходном винте никогда и ни при каких условиях не подвергнутся записи. Это сообщение больше относится к нашему новому, чистому винту. Поэтому, если вы купили чистый винт и хотите на него переехать, то вряд ли `clonehdd` чем то сможет навредить.

Работа с программой предельно проста. Для осуществления задуманного нам нужно всего один раз запустить утилиту с нужными параметрами. Вот о параметрах теперь поподробнее.

`-src=[device]` – Это исходный девайс на котором находится наша система. Чтобы узнать имя текущего винта, нужно выполнить команду:

```
/sbin/mount
```

Мы увидим таблицу, в первой колонке которой указаны имена примонтированных устройств. Вот, например: `/dev/ad0s1f`

`/dev/»` это папка, в которой находятся все устройства, «`ad0`» это нужное нам имя самого винта,

«`s1`» слайс под номером 1, «`f`» имя партиции на слайсе.

Вывод: значение этому параметру присваиваем «`ad0`»

`-dst=[device]` Это девайс назначения. Именно сюда переедет ваша система. Узнать его можно, например, командой:

```
cat /var/run/dmesg.boot
```

На экране появится список сообщений ядра при загрузке. Обычно, сообщения о найденных винтах находятся в самом конце. Ни одна партиция не должна быть примонтирована с этого винта, иначе `CloneHDD` выдаст ошибку.

`-swap=[size in MB]` Размер будущего раздела SWAP.

`-freespace=[Required free space on dst partition]` Если клонирование производится на винт меньшего размера, на DST винте после клонирования должен остаться небольшой кусочек

свободного места, для того чтобы винт небыл на 100% заполнен после клонирования. По умолчанию – это 100МБ. Этим параметром его можно изменить.

`-safe` [Required safe mode for `dump`] Включает режим безопасного копирования. В этом режиме сначала создается образ каждого раздела в папке `.snap`, а потом делается перенос. В «небезопасном» режиме, копирование производится на лету. Вся проблема в том что размер этого образа равен размеру данных на разделе. А т.к. образ будет записан на тот же раздел с которого производится копирование, на разделе должно быть минимум 50% свободного места. Вернемся к параметру `-safe`. Если параметр отсутствует, клонирование будет произведено в безопасном режиме если достаточно свободного места или в небезопасном режиме если этого места недостаточно. Если параметр задан, небезопасного копирования не будет, и разделы с недостатком свободного места склонированы не будут.

`-fstab`=[Device name to write in fstab] После клонирования будет сгенерирован файл `/etc/fstab` на полученном винте. Параметр задает Имя девайса, который будет записан в этот файл. По умолчанию: девайс, заданный параметром `-src`.

`-force` [Do not ask any questions] При выполнении второй части задачи, утилита будет выполняться из `cron`'а и нам не нужны лишние вопросы, в ходе выполнения программы. Данный параметр отключает интерактивный режим и вопрос «Are you sure?» задаваться не будет.

Вернемся к нашей задаче. Мой исходный винт: `ad0`, новый винт: `ad1`.

```
# clonehdd -src=ad0 -dst=ad1 -swap=1024
Clone parameters:
Source partition: /dev/ad0
Dest partition: /dev/ad1
Swap size: 1024 MB
Safe dumping: Disabled
Free space on DST: 100 MB
```

Fstab device name: ad0

—

[OK] Found devices for clone procedure

[OK] DST partition is not in use

—

Source partition

/usr size: 64489MB, used: 10563MB

/var size: 4958MB, used: 144MB

/ size: 1483MB, used: 82MB

/tmp size: 1483MB, used: MB

Total: 72415 MB, used: 10790 MB

—

[OK] Device ad1 has enough free space

Wait 10 seconds before start: 10 9 8 7 6 5 4 3 2 1

[OK] Device /dev/ad1 made clean

[OK] New slice created

—

Destination device partitions:

SWAP size: 1024 MB

/ size 1588 MB

/tmp size 1588 MB

/var size 5306 MB

/usr size 69026 MB

—

[INF] Last partition were increased for 3 blocks

[OK] Partitions were created successfully

—

[OK] Partition /tmp was formatted successfully

Starting dump/restore procedure... [OK]

[OK] Partition /var was formatted successfully

Starting dump/restore procedure... [OK]

[OK] Partition /usr was formatted successfully

Starting dump/restore procedure... [OK]

[OK] Partition / was formatted successfully

Starting dump/restore procedure... [OK]

[OK] file /etc/fstab generated successfully

Видим сообщения об успешном клонировании. Теперь сделаем выполнение задачи по расписанию: выполнять каждый день в 2 часа ночи клонирование рабочего винта на запасной. Добавляем в файл `/etc/crontab` строку

```
0 2 * * * root clonehdd -src=ad0 -dst=ad1 -swap=1024 -force  
>/dev/null
```

Обратите внимание, что `>/dev/null` убивает не все сообщения. Только обычные сообщения будут опущены. Все сообщения об ошибках отсылаются на выход `STDERR`, не попадают на `/dev/null` и будут отправлены `cron`'ом по почте системному администратору.

<http://subnets.ru/blog/?p=905>

<https://anikin.pw/all/perenos-freebsd-na-drugoy-zhestkiy-disk/>

<http://www.opennet.ru/base/sys/clonehdd.txt.html>