# Как я FreeBSD на терабайтник ставил

При переносе системы на терабайтник и написании статьи использовались материалы по следующим ссылкам:

http://m8d.de/news/freebsd-on-gpt.php

http://www.lissyara.su/?id=1704

Возникла у меня как-то необходимость заменить в домашнем сервере, который работает под управлением ОС FreeBSD 7.1 Release, жёсткий диск Samsung SP0411N (40GB, IDE) на Seagate ST31000333AS (1TB, SATAII). Материнская плата, установленная на сервере – Asus P5V800-MX (чипсет VIA), поддерживает HDD IDE, SATA и SATAII.

Не долго думая, отключил старый винт, подсоединил новый, вставил в привод DVD с FreeBSD, запустил установку и… И обломался об утилиту fdisk, которой показалось, что винт имеет неправильную геометрию.

Для меня так и осталось загадкой, что же не понравилось этой утилите в моём новом HDD. Судя по хэндбуку, ОС должна нормально работать с разделами до 2TB.

По морально-этическим соображениям пропущу описание попыток заставить систему встать на этот жёсткий, так как к результату они не привели, но при этом в большом количестве сопровождались ненормативными фразами и выражениями.

Позадавав вопросы на разнообразных форумах, решил попробовать использовать GPT. Итак, к делу:

## 1. Подготовка

Загружаюсь со старого винта, на который установлена FreeBSD 7.1 Release при подключенном новом терабайтнике. У меня в ядре уже была включена поддержка GPT, но перед работой проверьте в конфигурации ядра наличие строчки

```
options   GEOM_GPT
```

терабайтник у меня определился как **ad4** (а как определился Ваш, можно посмотреть в файле **/var/run/dmesg.boot**).

## 2. Создаю схему разметки GPT

Зашёл под рутом и дал команду

```
gpart create -s GPT ad4
```
(1)

вывода команд привести не могу, не помню. Этой командой я создал схему разметки GPT на провайдере, а провайдером в данном случае выступает **ad4**, то есть мой новый винт – сигейт-терабайтник (Во как я загнул! Это вольный перевод **man gpart**).

Теперь я дам команду **gpart show**, которая покажет информацию о GPT в системе, в моём случае покажет, что диск свободен, начинается с 34 и имеет размером какое-то страшное число логических блоков, которое тут же и в гигах указывается. В процессе разметки диска я буду эту команду постоянно использовать, так что будьте готовы.

## 3. Создаю разделы

Теперь создам загрузочный раздел:

```
gpart add -b 34 -s 16 -t freebsd-boot ad4
```
(2)

эта команда создала раздел, начиная с логического блока **34** (блин, ну почему не сектора? Какие ещё логические блоки?), размером в **16** логических блоков, имеющий тип **freebsd-boot** и размещающийся на **ad4**, только **ad4** в документации теперь называет-

ся не «провайдер», а уже «геом». То есть до команды «**gpart create -s GPT ad4**» **ad4** был провайдером, а после этой команды волшебным образом стал геомом. Браво!

Ну а после этого я ка-а-а-ак начал разделы налево-направо создавать, тока успевал энтер нажимать! Пишу команды в том порядке, в котором я их давал:

```
gpart add -b 50 -s 2097152 -t freebsd-ufs ad4
```
(3)

этой командой я создал раздел для корневой файловой системы размером в 1 гиг, то есть раздел начинается с логического блока **50** и занимает **2097152** логических блока. Число, с которого должен начинаться следующий раздел берём из вывода команды **gpart show** (просто смотрим там, с какого блока начинается свободное место), а сколько блоков надо на один гиг, я выяснил, разделив количество блоков на количество гиг на винте (эти данные тоже можно посмотреть в выводе команды **gpart show**).

```
gpart add -b 2097202 -s 8388608 -t freebsd-swap ad4
```
(4)

это 4 гига под **swap** (объём свопника я задал, умножив количество имеющейся у меня в сервере оперативки на 2).

```
gpart add -b 10485810 -s 8388608 -t freebsd-ufs ad4
```
(5)

это 4 гига под **/tmp** – с большим запасом. Я запасливый.

```
gpart add -b 18874418 -s 20971520 -t freebsd-ufs ad4
```
(6)

это 10 гиг под **/var** на случай разрастания логов.

```
gpart add -b 39845938 -s 1913679197 -t freebsd-ufs ad4
```
(7)

и всё остальное под **/usr**. Как я выяснил, сколько там осталось этого самого остального? Да опять из вывода **gpart show**, в последней строчке вывода указано, с какого блока начинается свободное место и каков его размер.

После этих манипуляций вывод команды **gpart show** у меня стал такой:

```
=>        34  1953525101  ad4  GPT  (932G)
          34          16    1  freebsd-boot  (8.0K)
          50     2097152    2  freebsd-ufs  (1.0G)
     2097202     8388608    3  freebsd-swap  (4.0G)
    10485810     8388608    4  freebsd-ufs  (4.0G)
    18874418    20971520    5  freebsd-ufs  (10G)
    39845938  1913679197    6  freebsd-ufs  (913G)
```

То есть я получил такое размещение разделов на винте:

| Раздел | Тип | Размер | Файловая система |
|---|---|---|---|
| ad4p1 | freebsd-boot | 8KB | - |
| ad4p2 | freebsd-ufs | 1GB | / |
| ad4p3 | freebsd-swap | 4GB | /swap |
| ad4p4 | freebsd-ufs | 4GB | /tmp |
| ad4p5 | freebsd-ufs | 10GB | /var |
| ad4p6 | freebsd-ufs | 913GB | /usr |

## 4. Делаю диск загрузочным

Нужно сделать загрузку FreeBSD с нового винта.

Сделаю винт загрузочным:

```
gpart bootcode -b /dist/boot/pmbr ad4
```
(8)

Теперь запишу в загрузочный код загрузчик FreeBSD:

```
cp /boot/gptboot /tmp
```
(9)

```
dd if=/dev/zero bs=641 count=1 >> /tmp/gptboot
```
(10)

И запишу загрузочный код в загрузочный сектор:

```
dd if=/tmp/gptboot of=/dev/ad4p1 bs=512
```
(11)

## 5. Инициализирую разделы

Едем дальше. А дальше созданные разделы необходимо подготовить для работы, то есть проинициализировать. Если бы я в винде был, я б просто сказал «форматнуть, в никсах есть такое определение?

Итак, команды:

```
newfs -O2 /dev/ad4p2
```
(12)

```
newfs -O2 -U /dev/ad4p4
```
(13)

```
newfs -O2 -U /dev/ad4p5
```
(14)

```
newfs -O2 -U /dev/ad4p6
```
(15)

Подготовка диска завершена, перехожу к переносу системы на новый диск.

P.S. Хочу отметить, что вместо команды **gpart** можно было попробовать воспользоваться командой **gpt**.

## 6. Переношу систему на новый жёсткий диск

В качестве подготовки к переносу создал каталоги **/mnt/root**, **/mnt/var** и **/mnt/usr**:

```
mkdir /mnt/root; mkdir /mnt/var; mkdir /mnt/usr
```
(16)

смонтировал туда новые разделы:

```
mount /dev/ad4p2 /mnt/root
```
(17)

```
mount /dev/ad4p5 /mnt/var
```
(18)

```
mount /dev/ad4p6 /mnt/usr
```
(19)

И непосредственно перенос системы:

```
cd /; pax -p eme -X -rw . /mnt/root
```
(20)

```
cd /var; pax -p eme -X -rw . /mnt/var
```
(21)

```
cd /usr; pax -p eme -X -rw . /mnt/usr
```
(22)

Теперь редактирую файл **/etc/fstab** (на новом диске названия разделов не те, что на старом, так что система при загрузке с терабайтника попытается смонтировать те разделы, которые были на старом самсунге-сороковке, а так как сороковка в тот момент будет уже отключена, то… То не знаю что будет, не проверял. Думаю, что ничего хорошего.).

У меня получился вот такой файл:

```
# Device        Mountpoint      FStype    Options         Dump Pass#
/dev/ad4p3      none            swap      sw              0    0
/dev/ad4p2      /               ufs       rw              1    1
/dev/ad4p4      /tmp            ufs       rw              2    2
/dev/ad4p6      /usr            ufs       rw              2    2
/dev/ad4p5      /var            ufs       rw              2    2
/dev/acd0       /cdrom          cd9660    ro,noauto       0    0
```

Выключаю сервер командой **shutdown -p now** и отключаю старый диск.

## 7. Заключение

После включения у меня всё заработало как надо, кроме **mysql**, потому что его сокет лежал в **/tmp**, а во время переноса на директорию **/tmp** изменились права. Починил командой

```
chmod 1777 /tmp
```

Ну и всё. Удачи!