Formatting Media For Use With FreeBSD

A Tutorial

Doug White

dwhite@resnet.uoregon.edu

March 1997

This document describes how to slice, partition, and format hard disk drives and similar media for use with FreeBSD. The examples given have been tested under FreeBSD 2.2 and should work for other releases. The text has been updated for FreeBSD version 4.

1 Introduction & Definitions

1.1 Overview

Successfully adding disks to an existing system is the mark of an experienced system administrator. Slicing, partitioning, and adding disks requires a careful dance of proper command and name syntax. One slipped finger and an entire disk could disappear in seconds. This document is written in an attempt to simplify this process and avoid accidents. Thankfully, enhancements to existing tools (notably sysinstall) have greatly improved this process in recent releases of FreeBSD.

There are two possible modes of disk formatting:

- *compatibility mode*: Arranging a disk so that it has a slice table for use with other operating systems.

- *dedicated mode*, sometimes called *dangerously dedicated mode*: Formatting a disk with no slice table. This makes the process of adding disks easier, however non-FreeBSD operating systems may not accept the disk. The term *dangerously* refers to the danger that the system may not recognize a disk formatted in this manner.

For most cases, dedicated mode is the easiest to set up and use in existing systems, as a new disk is usually dedicated entirely to FreeBSD. However, compatibility mode insures optimum interoperability with future installations at a cost of increased complexity.

In addition to selecting the mode, two methods of slicing the disk are available. One is using the system installation tool `/stand/sysinstall`. 2.1.7-RELEASE and later versions of `sysinstall` contain code to ease setup of disks during normal system operation, mainly allowing access to the Label and Partition editors and a Write feature which will update just the selected disk and slice without affecting other disks. The other method is running the tools manually from a root command line. For dedicated mode, only three or four commands are involved while `sysinstall` requires some manipulation.

1.2 Definitions

UNIX® disk management over the centuries has invented many new definitions for old words. The following glossary covers the definitions used in this document and (hopefully) for FreeBSD in general.

- compatibility mode: Arranging a disk so that it has a slice table for use with other operating systems. Oppose dedicated mode.

- (dangerously) dedicated mode: Formatting a disk with no slice table. This makes the process of adding disks easier, however non-FreeBSD operating systems may not accept the disk. Oppose compatibility mode.

- disk: Hard disks, CDROMs, magneto-optical devices and Zip®/Jaz® removable media are example of storage devices commonly used today. The basic principle of the way these work is that one or more spinning disks spin by a motor, while a head, moving on a radial path close to the disks, reads from or writes data to the disk. Writing is done by modifying some physical properties of the disk (magnetic flow, reflectivity, etc.) while reading is done by "detecting" changes to the same physical properties of the disk.

- slice: A division of a disk. Up to four slices are permitted on one disk in the PC standard. Slices are composed of contiguous sectors. Slices are recorded in a "slice table" used by the system BIOS to locate bootable partitions. The slice table is usually called the "partition table" in DOS parlance. Maintained by the fdisk utility.

- partition: A division of a slice. Usually used in reference to divisions of the FreeBSD slice of a disk. Each filesystem and swap area on a disk resides in a partition. Maintained using the disklabel utility.

- sector: Smallest subdivision of a disk. One sector usually represents 512 bytes of data.

1.3 Warnings & Pitfalls

Building disks is not something to take lightly. It is quite possible to destroy the contents of other disks in your system if the proper precautions are not taken.

Check your work carefully. It is very simple to destroy the incorrect disk when working with these commands. When in doubt consult the kernel boot output for the proper device.

Needless to say, we are not responsible for any damage to any data or hardware that you may experience. You work at your own risk!

1.4 Zip, Jaz, and Other Removables

Removable disks can be formatted in the same way as normal hard disks. It is essential to have the disk drive connected to the system and a disk placed in the drive during startup, so the kernel can determine the drive's geometry. Check the `dmesg` output and make sure your device and the disk's size is listed. If the kernel reports

```
Can't get the size
```

then the disk was not in the drive. In this case, you will need to restart the machine before attempting to format disks.

2 Formatting Disks in Dedicated Mode

2.1 Introduction

This section details how to make disks that are totally dedicated to FreeBSD. Remember, dedicated mode disks sometimes cannot be booted by the PC architecture.

2.2 Making Dedicated Mode Disks using Sysinstall

`/stand/sysinstall`, the system installation utility, has been expanded in recent versions to make the process of dividing disks properly a less tiring affair. The fdisk and disklabel editors built into sysinstall are GUI tools that remove much of the confusion from slicing disks. For FreeBSD versions 2.1.7 and later, this is perhaps the simplest way to slice disks.

1. Start sysinstall as root by typing

   # **/stand/sysinstall**

   from the command prompt.
2. Select `Index`.
3. Select `Partition`.
4. Select the disk to edit with arrow keys and **SPACE**.
5. If you are using this entire disk for FreeBSD, select `A`.
6. When asked if you still want to do this, answer `Yes`.
7. Select `Write`.
8. When warned about writing on installed systems, answer `Yes`.
9. When asked about installing a boot loader, select `None` to leave the Master Boot Record untouched. This is only needed on a new install of FreeBSD unless the disk will be placed into another machine.
10. Press **ENTER** on the message stating "Wrote FDISK partition information out successfully".
11. `Quit` the FDISK Editor and **ESCAPE** back to the Index menu.
12. Select `Label` from the Index menu.

13. Label as desired. For a single partition, enter `C` to Create a partition, accept the default size, partition type Filesystem, and a mountpoint (which is not used).

14. Enter `W` when done and confirm to continue. The filesystem will be newfs'd for you, unless you select otherwise (for new partitions you will want to do this!). You will get the error:

```
Error mounting /mnt/dev/ad2s1e on /mnt/blah : No such file or directory
```

Ignore.

15. Exit out by repeatedly pressing **ESCAPE**.

2.3 Making Dedicated Mode Disks Using the Command Line

Execute the following commands, replacing `ad2` with the disk name.

```
# dd if=/dev/zero of=/dev/ad2 count=2
# disklabel /dev/ad2 | disklabel –B –R –r ad2 /dev/stdin
```
We only want one partition, so using slice 'c' should be fine:
```
# newfs /dev/ad2c
```

If you need to edit the disklabel to create multiple partitions (such as swap), use the following:

```
# dd if=/dev/zero of=/dev/ad2 count=2
# disklabel /dev/ad2 > /tmp/label
```
Edit disklabel to add partitions:
```
# vi /tmp/label
# disklabel –B –R –r ad2 /tmp/label
```
newfs partitions appropriately

Your disk is now ready for use.

3 Making Compatibility Mode Disks

3.1 Introduction

The command line is the easiest way to make dedicated disks, and the worst way to make compatibility disks. The command-line `fdisk` utility requires higher math skills and an in-depth understanding of the slice table, which is more than most people want to deal with. Use sysinstall for compatibility disks, as described below.

3.2 Making Compatibility Mode Disks Using Sysinstall

1. Start sysinstall as root by typing

```
# /stand/sysinstall
```

from the command prompt.

2. Select `Index`.

3. Select `Partition`.

4. Select the disk to edit with arrow keys and **SPACE**.

5. If you are using this entire disk for FreeBSD, select `A`.

6. When asked:

   ```
   Do you want to do this with a true partition entry so as to remain
   cooperative with any future possible operating systems on the
   drive(s)?
   ```

   answer `yes`.

7. Select `Write`.

8. When asked to install the boot manager, select None with **SPACE** then hit **ENTER** for OK.

9. `Quit` the FDISK Editor.

10. You will be asked about the boot manager, select `None` again.

11. Select `Label` from the Index menu.

12. Label as desired. For a single partition, accept the default size, type filesystem, and a mountpoint (which is not used).

13. The filesystem will be newfs'd for you, unless you select otherwise (for new partitions you will want to do this!). You will get the error:

    ```
    Error mounting /mnt/dev/ad2s1e on /mnt/blah : No such file or directory
    ```

    Ignore.

14. Exit out by repeatedly pressing **ESCAPE**.

Your new disk is now ready for use.

4 Other Disk Operations

4.1 Adding Swap Space

As a system grows, its need for swap space can also grow. Although adding swap space to existing disks is very difficult, a new disk can be partitioned with additional swap space.

To add swap space when adding a disk to a system:

1. When partitioning the disk, edit the disklabel and allocate the amount of swap space to add in partition 'b' and the remainder in another partition, such as 'a' or 'e'. The size is given in 512 byte blocks.

2. When newfsing the drive, do NOT newfs the 'c' partition. Instead, newfs the partition where the non-swap space lies.

3. Add an entry to `/etc/fstab` as follows:

```
/dev/ad0b                               none            swap    sw 0 0
```

Change `/dev/ad0b` to the device of the newly added space.

4. To make the new space immediately available, use the `swapon` command.

```
# swapon /dev/da0b
swapon:  added /dev/da0b as swap space
```

4.2 Copying the Contents of Disks

Submitted By: Renaud Waldura (<`renaud@softway.com`>)

To move file from your original base disk to the fresh new one, do:

```
# mount /dev/ad2 /mnt
# pax -r -w -p e /usr/home /mnt
# umount /mnt
# rm -rf /usr/home/*
# mount /dev/ad2 /usr/home
```

4.3 Creating Striped Disks using CCD

Commands Submitted By: Stan Brown (<`stanb@awod.com`>)

The Concatenated Disk Driver, or CCD, allows you to treat several identical disks as a single disk. Striping can result in increased disk performance by distributing reads and writes across the disks. See the ccd(4) and ccdconfig(8) manual pages or the CCD Homepage (http://stampede.cs.berkeley.edu/ccd/) for further details.

You no longer need to build a special kernel to run ccd. When you run `ccdconfig`, it will load the KLD for you if the kernel does not contain CCD support.

You build CCDs on disk partitions of type *4.2BSD*. If you want to use the entire disk, you still need to create a new partition. For example, **disklabel -e** might show:

```
#        size    offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0    0    0   # (Cyl.   0 - 59597)
```

You should not use partition *c* for the CCD, since it is of type *unused*. Instead, create a new partition of exactly the same size, but with type *4.2BSD*:

```
#        size    offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0    0    0   # (Cyl.   0 - 59597)
  e: 60074784        0    4.2BSD        0    0    0   # (Cyl.   0 - 59597)
```

To create a new CCD, execute the following commands. This describes how to add three disks together; simply add or remove devices as necessary. Remember that the disks to be striped must be *identical.*

```
# cd /dev ; sh MAKEDEV ccd0

# disklabel -r -w da0 auto
# disklabel -r -w da1 auto
# disklabel -r -w da2 auto

# disklabel -e da0
```
Add partition e with type 4.2BSD
```
# disklabel -e da1
```
Add partition e with type 4.2BSD
```
# disklabel -e da2
```
Add partition e with type 4.2BSD
```
# ccdconfig ccd0 273 0 /dev/da0e /dev/da1e /dev/da2e

# newfs /dev/ccd0c
```

The value 273 is the stripe size. This is the number of disk sectors (of 512 bytes each) in each block of data on the CCD. It should be at least 128 kB, and it should not be not be a power of 2.

Now you can mount and use your CCD by referencing device /dev/ccd0c.

A more powerful and flexible alternative to CCD is Vinum. See the Vinum Project home page (http://www.vinumvm.org/) for further details.

5 Credits

The author would like to thank the following individuals for their contributions to this project:

- Darryl Okahata (<darrylo@hpnmhjw.sr.hp.com>) for his simple dedicated mode setup documentation which I have used repeatedly on FreeBSD-questions.

- Jordan K. Hubbard <jkh@FreeBSD.org> for making sysinstall useful for this type of task.

- John Fieber (<jfieber@indiana.edu>) for making information and examples of the DocBook DTD on which this document is based.

- Greg Lehey <grog@FreeBSD.org> for checking my work and pointing out inaccuracies, as well as miscellaneous support.