The Linux+FreeBSD mini−HOWTO

Table of Contents

The Linux+FreeBSD mini−HOWTO

Niels Kristian Bech Jensen `nkbj@image.dk`

v1.11, 30 March 2000

This document describes how to use Linux and FreeBSD on the same system. It introduces FreeBSD and discusses how the two operating systems can cooperate, e.g. by sharing swap space. You should probably have some experience with Linux or FreeBSD and hard drive partitioning (`fdisk`) before you read this document. The tips herein are tested using FreeBSD 2.2.2, but they should be valid for newer versions as well. Do not hesitate to mail me if you have comments, questions or suggestions about this document. I would also like to hear from people who have experience using Linux together with NetBSD or OpenBSD.

---

8. Credits and legal stuff

---

1. What is FreeBSD?

FreeBSD is a free Unix−like operating system much like Linux. The main difference is that, while the Linux kernel has been written from scratch, FreeBSD is based on the freely redistributable parts of 4.4BSD (Berkeley Software Distribution) known as 4.4BSD−lite. This fact might lead some people to suggest that FreeBSD is closer to being ``real'' UNIX® than Linux. FreeBSD runs only on the Intel PC platform (i386 and higher); ports to the DEC Alpha and Sun Sparc platforms are being worked on at the moment. NetBSD and OpenBSD are similar to FreeBSD, and both run on several platforms. Hardware requirements for all these *BSD systems are similar to those for Linux.

The development of FreeBSD is managed in a different way than the Linux development. A core team of developers serve as arbitrators and provide leadership for the project. Big changes are discussed in advance on the mailing lists. The FreeBSD project has two development trees (just like Linux): **``−CURRENT''** and **``−STABLE''**. The ``−CURRENT'' development tree is where the development of new features happens. Development to the ``−STABLE'' tree are restricted to bug fixes and some thoroughly tested new features.

FreeBSD can be used and (re−)distributed freely just as Linux. Most parts of the system are released under the BSD copyright; the rest is under the GNU GPL or other open−source licences.

---

2. The FreeBSD way of labelling hard drives

Linux and FreeBSD label hard drives and partitions after two differents schemes. This section explains the main differences between the two schemes. In fact the FreeBSD labelling scheme is an adaption of the traditional BSD labelling style ported to live within the PC's fdisk partitions. Thus it is very similar to other BSD−based Unix systems such as NetBSD, OpenBSD, Ultrix, Digital Unix, SunOS, and Solaris.

2.1 FreeBSD ``slices'' and ``partitions''

FreeBSD needs one of the four entries in the partition table on your PC's hard drive. This primary partition is called a **``slice''** in FreeBSD terminology. It then uses the `disklabel` program to make up to eight partitions in this primary partition. These logical partitions are called **``partitions''** in FreeBSD terminology. This concept is similar to the way Linux (and DOS) handles logical partitions in an extended partition. You cannot install FreeBSD in an extended partition made by Linux (or DOS). Note that the Linux `fdisk` program doesn't display the BSD partitions in a FreeBSD slice from the main menu, but it can display BSD disklabel information if you give the command `b'. The output is something like this (`/dev/hda4` is the FreeBSD slice):

```
bash# fdisk /dev/hda

Command (m for help): p

Disk /dev/hda: 64 heads, 63 sectors, 621 cylinders
Units = cylinders of 4032 * 512 bytes
```

```
   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda1   *        1        1       27    54400+  83  Linux native
/dev/hda2           28       28       55    56448   83  Linux native
/dev/hda3           56       56      403   701568   83  Linux native
/dev/hda4          404      404      621   439488   a5  BSD/386

Command (m for help): b
Reading disklabel of /dev/hda4 at sector 1624897.

BSD disklabel command (m for help): p

8 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  a:    64512  1624896    4.2BSD        0     0     0   # (Cyl.  404 - 419)
  b:   104832  1689408      swap                        # (Cyl.  420 - 445)
  c:   878976  1624896    unused        0     0         # (Cyl.  404 - 621)
  e:    64512  1794240    4.2BSD        0     0     0   # (Cyl.  446 - 461)
  f:   645120  1858752    4.2BSD        0     0     0   # (Cyl.  462 - 621)

BSD disklabel command (m for help): q
bash#
```

The letters `a'...`f' in the first column are the same labels as shown below in the example for a FreeBSD slice. There are three special partitions in BSD parlace. The letter `a' designates the root partition, `b' designates the swap partition, while `c' designates the whole slice. See the FreeBSD documentation for more information on the ``standard'' way of assigning these letters to different partition types.

2.2 Drive and partition labelling in Linux and FreeBSD

The hard drives are labelled in the following way in Linux and FreeBSD:

```
                        Linux           FreeBSD
First IDE drive         /dev/hda        /dev/wd0
Second IDE drive        /dev/hdb        /dev/wd1
First SCSI drive        /dev/sda        /dev/sd0
Second SCSI drive       /dev/sdb        /dev/sd1
```

The partitions (FreeBSD slices) on an IDE drive are labelled in the following way (`/dev/hda` is used as an example):

```
                                Linux           FreeBSD
First primary partition         /dev/hda1       /dev/wd0s1
Second primary partition        /dev/hda2       /dev/wd0s2
Third primary partition         /dev/hda3       /dev/wd0s3
Fourth primary partition        /dev/hda4       /dev/wd0s4
```

The partitions in my FreeBSD slice is labelled in the following way. It is the labelling you get by default. It is possible to change the labelling if you do a custom installation of FreeBSD (`/dev/hda4` is the FreeBSD slice in the example):

```
Linux label     FreeBSD label   FreeBSD mount point
/dev/hda5       /dev/wd0s4a     /
/dev/hda6       /dev/wd0s4b     swap
/dev/hda7       /dev/wd0s4e     /var
/dev/hda8       /dev/wd0s4f     /usr
```

If you run `dmesg` in Linux you will see this as (The linux kernel must be build with **UFS filesystem support** for this to work. See section Installing and preparing Linux):

```
Partition check:
 hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 >
```

If you have installed FreeBSD in the `/dev/sd1s3` slice (`/dev/sdb3` in Linux parlace), and `/dev/sdb2` is a Linux extended partition containing two logical partitions (`/dev/sdb5` and `/dev/sdb6`), the previous example would look like this:

```
Linux label      FreeBSD label    FreeBSD mount point
/dev/sdb7        /dev/sd1s3a      /
/dev/sdb8        /dev/sd1s3b      swap
/dev/sdb9        /dev/sd1s3e      /var
/dev/sdb10       /dev/sd1s3f      /usr
```

This will be shown as

```
Partition check:
 sdb: sdb1 sdb2 < sdb5 sdb6 > sdb3 < sdb7 sdb8 sdb9 sdb10 >
```

in the output from `dmesg`.

If you have a Linux extended partition *after* your FreeBSD slice you're in for trouble, because most Linux kernels installation floppies are build without UFS support, they will not recognise the FreeBSD partitions inside the slice. What should have have been seen as (`/dev/hda3` is the FreeBSD slice and `/dev/hda4` is the Linux extended partition)

```
Partition check:
 hda: hda1 hda2 hda3 < hda5 hda6 hda7 hda8 > hda4 < hda9 hda10 >
```

is seen as:

```
Partition check:
 hda: hda1 hda2 hda3 hda4 < hda5 hda6 >
```

This can give you the wrong device assignment and cause the loss of data. My advice is to *always put your FreeBSD slice after any Linux extended partitions, and do not change any logical partitions in your Linux extended partitions after installing FreeBSD!*

---

3. Sharing swap space between Linux and FreeBSD

This section describes how I got Linux and FreeBSD to share a swap partition. There may be other ways to get the same result. You can install FreeBSD before Linux if you want to, just pay attention to the order of the partitions in the FreeBSD slice.

3.1 Installing and preparing Linux

The first step is to install Linux as normal. You have to leave space for the FreeBSD slice at your hard drive. You don't have to make a Linux swap partition, but if you want one, put it in the space you want to allocate for FreeBSD. That way you can delete the Linux swap partition later and use the space for FreeBSD.

When you have installed Linux you have to build a new kernel. Read **The Linux Kernel HOWTO** if this is new to you. You *have* to include both **UFS filesystem support (read only)** and **BSD disklabel (FreeBSD partition tables) support**:

```
UFS filesystem support (read only) (CONFIG_UFS_FS) [N/y/m/?] y
BSD disklabel (FreeBSD partition tables) support (CONFIG_BSD_DISKLABEL) [N/y/?]
(NEW) y
```

Install the new kernel and reboot. Remove any line including the word *swap* from your `/etc/fstab` file if you have made a Linux swap partition. *Make sure you have a working Linux boot floppy with the new kernel.* Now you are ready to install FreeBSD.

3.2 Installing FreeBSD

Install FreeBSD as described in the FreeBSD documentation. Remove the Linux swap partition if you have made one (you can use the FreeBSD `fdisk` program.) Pay attention to the order of the partitions in the FreeBSD slice. If you use the default labelling the second partition will be the swap partition. Complete the installation of FreeBSD and reboot into Linux *using the new Linux boot floppy*.

3.3 Setting up the FreeBSD swap partition in Linux

Run `dmesg` when you have booted into Linux. In the output you should see something like this:

```
Partition check:
 hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 >
```

This means that `/dev/hda4` is your FreeBSD slice, while `/dev/hda5`, `/dev/hda6`, `/dev/hda7` and `/dev/hda8` are the FreeBSD partitions. If your swap partition is the second partition in the slice, it will be `/dev/hda6`.

You have to put the following line into your Linux `/etc/fstab` file to enable the swap partition:

```
/dev/hda6       none            swap            sw              0       0
```

While FreeBSD can use any type of partition as swap space, Linux needs a special signature in the swap partition. This signature is made by `mkswap`. FreeBSD ruins this signature when it uses the shared swap partition, so you will have to run `mkswap` each time you boot into Linux. To do this automagically you have to find the script that runs `swapon` at boot time. In Red Hat Linux it is `/etc/rc.d/rc.sysinit`. Put the following line into that file just *before* `swapon -a`:

```
awk -- '/swap/ && ($1 !~ /#/) { system("mkswap "$1"") }' /etc/fstab
```

This will run `mkswap` on any swap partitions in `/etc/fstab` every time you boot except if they are commented out (having ``#" as the first character in the line.)

Run `free` to check out the size of the swap space when you have rebooted into Linux. You should also reboot into FreeBSD to make sure everything works as expected. If it does not, you have probably used the wrong partition as swap partition. The only solution to that problem is to reinstall FreeBSD and try again. Experience is a great teacher. :−)

4. Booting FreeBSD using LILO

You can easily boot FreeBSD with LILO. Do not install the FreeBSD boot selector (`Booteasy`) if you want to use LILO. Append the following lines to your `/etc/lilo.conf` file and run `lilo` (assuming the FreeBSD slice is `/dev/hda4`):

```
other=/dev/hda4
        table=/dev/hda
        label=FreeBSD
```

If you have installed FreeBSD on the second SCSI drive, use something like this (the FreeBSD slice being `/dev/sdb2`):

```
other=/dev/sdb2
        table=/dev/sdb
        loader=/boot/chain.b
        label=FreeBSD
```

---

5. Mounting filesystems

5.1 Mounting UFS filesystems under Linux

Unfortunately the UFS driver in the Linux 2.0.xx kernels do not include support for FreeBSD. When you try to mount a FreeBSD filesystem, you just get some error messages (the filesystem actually gets mounted, but you cannot do anything with it.) This problem has been solved in the newer Linux kernels (version 2.1.87 and higher.)

There is another version of the UFS driver for Linux 2.0.xx kernels (xx <= 30) on metalab.unc.edu (the former sunsite.unc.edu). It is called **U2FS** and the current version is `u2fs−0.4.3.tar.gz`. A version of U2FS (`ufs−0.4.4.tar.gz`) for Linux 2.0.31 and higher (2.0.xx; not 2.1.xx) can be found at this site along with further information about U2FS (and UFS.)

Now you have to build a new kernel with support for the U2FS filesystem and BSD disklabels. See section Installing and preparing Linux for more information on this. You can leave out **UFS filesystem support** from the kernel when you use U2FS.

When you have installed the new kernel, you can mount your UFS filesystems (all the partitions in the FreeBSD slice except the swap partition) with a command like this:

```
mount −t u2fs /dev/hda8 /mnt
```

You should use a command like

```
mount −t ufs /dev/hda8 /mnt
```

if you use a Linux kernel version 2.1.87 or higher. From Linux kernel version 2.1.112 you must add `−o ufstype=44bsd` to the command like this:

```
mount −t ufs −o ufstype=44bsd /dev/hda8 /mnt
```

The UFS (and U2FS) driver is read−only. That is; you can read from the UFS filesystems but you cannot write to them. An experimental read−write UFS driver has replaced the read−only driver in Linux kernels version 2.1.112 and higher; writing to FreeBSD partitions is supported from version 2.1.127.

5.2 Mounting ext2fs filesystems under FreeBSD

To mount ext2fs filesystems under FreeBSD, you first have to build a new kernel with ext2fs support. Read the FreeBSD handbook to learn how to do that. Put the line

```
options         "EXT2FS"
```

in your kernel configuration file for the new kernel.

When you have booted with the new kernel, you can mount an ext2fs filesystem by giving a command like:

```
mount −t ext2fs /dev/wd0s3 /mnt
```

Due to a bug in FreeBSD 2.2.8 and earlier you will have to unmount all ext2fs filesystems *before* you shut down FreeBSD if you are using these any of these versions. If you shut down FreeBSD with an ext2fs filesystem mounted, FreeBSD cannot sync the UFS filesystems. This results in `fsck` being run the next time FreeBSD is booted. You can work around this bug by putting the line:

```
umount −a −t ext2fs
```

in the `/etc/rc.shutdown` file. The bug has been fixed in FreeBSD 3.x.

---

6. Running foreign binaries

6.1 Running FreeBSD binaries under Linux

The `iBCS` package has support for running FreeBSD binaries under Linux; but it's old and unmaintained. I cannot get it to work. Please let me know if you have had better luck with this.

6.2 Running Linux binaries under FreeBSD

FreeBSD has the ability to run Linux binaries, both in a.out and ELF formats. To do this you have to take the following three steps:

1. You have to enable Linux compatibility. To do this (in FreeBSD 2.2.2 −−− details may vary in other versions) you have to edit your `/etc/rc.conf` file and change
   ```
   linux_enable="NO"
   ```
   to
   ```
   linux_enable="YES"
   ```
   and reboot. Another way to load the Linux binary support is to execute the command `/usr/bin/linux`. This way you don't have to reboot, and you don't always have the Linux binary

support loaded (i.e. you save memory.) Remember to add the line

```
options         COMPAT_LINUX
```

to the FreeBSD kernel config file if you build a new FreeBSD kernel.

2. You have to install the Linux shared libraries if your Linux binaries are dynamically linked. The libraries are included in FreeBSD 2.2.{2,5,6} as the package `linux_lib−2.4.tgz` (newer versions might be available.) Run the following command to install the package:
```
pkg_add <path_to_package>/linux_lib−2.4.tgz
```

   <path_to_package> is the directory where the package is stored. You may also load it off the net by:

```
pkg_add ftp://ftp.freebsd.org/pub/FreeBSD/packages−stable/All/linux_lib−2.4.tgz
```

   or by re−running `/stand/sysinstall`. Enter ``Configure'', ``Packages'' and use the menus. You should execute the following command if you are running statically linked Linux binaries:

```
brandelf −t Linux <name_of_statically_linked_linux_binary>
```

3. Install the Linux program(s) you want to run. The program(s) can be installed on either UFS or ext2fs filesystems. See section Mounting ext2fs filesystems under FreeBSD for more information about using ext2fs filesystems under FreeBSD.

I have successfully run the Linux versions of Applixware 4.3 and Netscape 3.01 (both ELF format) under FreeBSD 2.2.2 using this method (yes, I know there is a native FreeBSD version of Netscape 4.) The Linux versions of acroread and StarOffice 3 and 4 also work well under FreeBSD. StarOffice 5 depends on native Linux threads and currently do not work under FreeBSD. Read the FreeBSD documentation for more information on this topic.

---

7. Information resources

The latest version of this mini−HOWTO can be downloaded from this site in several formats (including SGML and PostScript.) The document has been translated into Japanese by Mr. Teruyoshi Fujiwara as part of the JF project.

You can find some articles about the difference between Linux and FreeBSD here.

You can find more informations about FreeBSD (and download the whole system) at this site. You can also buy the system on CDROMs from Walnut Creek CDROM (their servers are running FreeBSD.)

The Linux Kernel HOWTO (and this mini−HOWTO) is released as part of The Linux Documentation Project.

---

8. Credits and legal stuff

Thanks to the members of the *BSD user group in Denmark for answering the questions of a FreeBSD newbie, to Mr. Takeshi Okazaki for bringing the existence of U2FS to my attention, and to Mr. David

O'Brien for valuable suggestions.

8.1 Legal stuff