Notes on FreeBSD

FreeBSD as serial console

```
# tip cisco
```

This connects you to the serial line labelled as 'cisco' in `/etc/remote`. To get out of `tip`, type

```
[Enter] ~ .
```

Configure `/etc/remote` with the serial port you wish to use: /dev/cuaa0 is "COM1" and /dev/cuaa1 is "COM2"

```
cisco:dv=/dev/cuaa0:br#9600:pa=none:
```

Line editing

To get line editing (hit cursor up to edit previous line), you may need to run 'bash' by hand. You can use 'vipw' to change your shell to /usr/local/bin/bash to make this permanent.

```
# bash
bash-2.03#
```

Using 'vi'

If you are unfamiliar with the 'vi' text editor, here is the absolute minimum set of commands you need to know. Beware: when you start 'vi' it is in command mode, so anything you type is interpreted as a command, not text to be entered!

```
# vi <filename>              -- edit a file

i  ....   [ESC]            -- insert text before cursor position
A  ....   [ESC]            -- append text to end of current line
x                          -- delete character under cursor
dd                         -- delete whole line
:wq                        -- save and exit
:q!                        -- exit without saving

h j k l                    -- left|down|up|right (if no cursor keys available)
```

Why did we choose FreeBSD rather than Linux?

We are using FreeBSD 4.0. It has a number of features which make it more appropriate than Linux for use in an ISP environment:

- Very stable, especially under load
- Heavily used and tested in large service providers
- Scalability features as standard: e.g. pwd.db (indexed password database)
- Optional "softupdates" filesystem combines crash-safety of BSD filesystem with speed of Linux filesystem - see `/usr/src/sys/contrib/softupdates/README`
- Similarities to BSDI and other "industrial strength" Unixes

Linux users will find some annoyances: for example, that 'bash' is not installed as standard, and new package management tools to learn. However, we feel that the robustness of FreeBSD is the most overriding factor for ISPs, and it's worth you getting to know and use FreeBSD.

Installation notes

Slices and partitions

What MS-DOS calls partitions are called "slices" by FreeBSD. FreeBSD almost always sits in a single slice.

The FreeBSD slice is then divided into "partitions". Example:

```
/dev/ad0      -- first ATA/ATAPI (IDE) hard drive
/dev/ad0s1    -- first slice (MSDOS "partition") on first IDE hard drive
/dev/ad0s1a   -- first partition in this FreeBSD slice
/dev/ad0s1b   -- second partition in this FreeBSD slice
/dev/ad0s1e   -- third (usable) partition
```

For historical reasons, partitions c and d are not used. We strongly recommend you configure your partitions as:

```
a:        root filesystem (/)
b:        swap space
e,f...:   other filesystems
```

All "large" parts of the filesystem should be separate from the root, so that the root itself remains small (less likely to get corrupted). This means at least /usr and /var, and possibly also /home if you have user accounts. The convention we have used is to put all remaining disk space in a partition called /u, and put home directories under that (/u/home/name)

Distribution sets

FreeBSD comes in several parts:

- "distribution sets" are the parts of the base system. The "bin" distribution is the minimum you can install. However there are additional distributions such as "manpages" and "XF86336".
- "ports" and "packages" are third-party software, not really part of FreeBSD itself, but supplied alongside it. "ports" build from source code, whilst "packages" are ports which have already been compiled and are ready to install.

Quick installation guide (using CD-ROM)

For your reference, this is the sequence of operations we followed to install the classroom workstations. Obviously you should adjust this as required to suit your needs (for example, most servers do not need to have X installed)

```
Insert boot floppy, change to root floppy when prompted
Skip kernel config
Express install
Delete any existing partitions, then select "Entire disk"
Say Yes to standard partition entry
Select BootMgr
```

```
Create partition; ctrl-U to delete number presented, enter "100m" instead
  FS   /
Create partition; ctrl-U; 100m
  Swap
Create partition; ctrl-U; 400m
  FS   /var
Create partition; ctrl-U; 400m
  FS   /usr
Create partition; hit enter to accept number given (i.e. rest of disk)
  FS   /u

X-User  (must hit SPACEBAR, not Enter, to select it)
No crypto
Not US resident
Yes install ports collection

Default answers to remaining questions (i.e. just hit Enter)
WAIT for install to complete
No extra options after install
Exit install
Reboot (remember to remove floppy and CD)
Login as root
halt

Label machine as being successfully installed
```

Post-installation configuration

There is a menu-based system you can use for configuring the system - setting IP address, configuring hostname, adding further distribution sets and packages, adding users etc:

```
# /stand/sysinstall
```

However, you may find that some of this is quicker to do from the command line, some of which is explained below.

rc.conf

The main system configuration file is `/etc/rc.conf`. This file is edited by /stand/sysinstall, but it's perfectly OK to edit this by hand. It is in this file that you configure the hostname, IP address for each interface, and so on. Changes you make in here won't take effect until you reboot.

```
ifconfig_ed0="inet 137.158.217.1 netmask 255.255.255.248"
defaultrouter="137.158.217.6"
hostname="pc1.t2.ws.afnog.org"

# On hosts where you don't want sendmail to accept incoming port 25
# (but you still want daemons to be able to send outgoing mail):
sendmail_flags="-q30m"
```

The full list of options, and their default values, can be found in /etc/defaults/rc.conf - but don't edit this file, edit /etc/rc.conf instead. This makes it easier to upgrade your system to a later version of FreeBSD.

User accounts

You can use the "adduser" script to add individual users to the system. Note: **Do not** edit the /etc/passwd file directly! Instead, use `vipw` to edit the password file. This actually edits

/etc/master.passwd and automatically generates all the other files from this (/etc/passwd, pwd.db, spwd.db)

Adding packages or ports by hand

A lot of third-party software is supplied in ready-to-use form, but is not in the base system. This is (a) to keep the base system smaller, and (b) because of different licencing terms. (The FreeBSD licence is actually *less* restrictive than the GNU/GPL licence under which a lot of open-source software is distributed)

All this third-party software installs under `/usr/local`.

You can use `/stand/sysinstall` to add packages, but it is quicker to use 'pkg_add' from the command line. For example, to add 'bash' and 'less':

```
# cd /cdrom/packages/All
# ls
# pkg_add bash-2.03.tgz
# pkg_add less-352.tgz
```

Note that the configuration files for third-party software are in `/usr/local/etc`, and scripts to start daemons are installed under `/usr/local/etc/rc.d`

You can also compile packages directly from the source code, if you have the "ports" distribution installed. The ports system automatically fetches the source file via FTP or anonCVS, applies any FreeBSD-specific patches, and compiles and installs the code. A "package" is really just a "port" which has been compiled.

```
# cd /usr/ports/shells/bash
# make
# make install
# make clean
```

Sometimes you will find that a "port" exists, but no corresponding binary "package". This is usually because of licencing or export restrictions. The "port" is always distributable because it does not include any software, only instructions on how to fetch and compile the software from somewhere else.

Differences from Linux

Other key differences between Linux and FreeBSD centre around the naming of devices and some of the configuration files:

```
Linux:   eth0 = first ethernet device (of any type)
FreeBSD: ed0  = first NE2000 device, ep0 = first 3Com 3c509, etc.

Linux:   COM1 serial port = /dev/ttyS0
FreeBSD: COM1 serial port = /dev/cuaa0 (call out) or /dev/ttyd0 (call in)

Linux:   /etc/inittab configures incoming serial connections
FreeBSD: /etc/ttys configures incoming serial connections
```

See the FreeBSD handbook and FAQ for more information, at http://www.freebsd.org/

Last updated 2000-04-28