FreeBSD 常見問答集

摘要

這份文件是 FreeBSD 常見問答集 ((FAQ))。我們盡可能地讓這份 FAQ 提供有用的資訊。

目錄

1. 前言 . . . 3
2. 文件與技術支援 . . . 7
3. 安裝 . . . 10
4. 硬體相容性 . . . 13
5. Troubleshooting . . . 17
6. User Applications . . . 22
7. Kernel Configuration . . . 24
8. Disks, File Systems, and Boot Loaders . . . 26
9. ZFS . . . 33
10. System Administration . . . 35
11. The X Window System and Virtual Consoles . . . 42
12. Networking . . . 48
13. Security . . . 52
14. PPP . . . 55
15. Serial Communications . . . 63
16. Miscellaneous Questions . . . 66
17. The FreeBSD Funnies . . . 69
18. Advanced Topics . . . 72
19. Acknowledgments . . . 76

Chapter 1. 前言

1.1. 什麼是 FreeBSD？

FreeBSD 是一個使用於桌機、筆電、伺服器與嵌入式系統平台的現代作業系統，支援多種平台。

它是根據 U.C. Berkeley 所開發出來的 "4.4BSD-Lite" ，並加上了許多 "4.4BSD-Lite2" 的增強功能。它同時也間接使用了 U.C. Berkeley 所開發出來並由 William Jolitz 移植到 i386™ 的 "Net/2"，也就是 "386BSD"，不過現在 386BSD 的程式碼只剩下極少數還留 存在 FreeBSD 中。

FreeBSD 已被廣泛地被世界各地的公司行號、ISP、研究人員、電腦專家、學生，以及家庭用戶所使用，用在工作、教育以及娛樂上。

如果想看關於 FreeBSD 更深入的資料，請看 FreeBSD 使用手冊。

1.2. 發展 FreeBSD 計畫的目的是什麼？

FreeBSD 計畫的目的是提供可以任意使用且沒有限制的穩定快速與一般用途的作業系統。

1.3. FreeBSD 版權有任何限制嗎？

有的。但是這並不是限制你怎麼去使用這些程式碼，而是你怎麼看待 FreeBSD 這個計畫。可以在此閱讀版權本文，簡單來說總結如下:

- 請勿宣稱是您寫了這個程式。
- 如果它出問題了，不要控告我們。
- 不要移除和修改版權

我們許多人在這個計畫投入很多心血，並不會介意獲得一些財務上的報酬，但是我們並沒有堅持一定要有。我們相信我們首要的"任務"是將程式碼提供給所有使用者，無論他們有任何的目的，這麼一來，這些程式碼才能被用在最多地方，也才能發揮它們最大的利益。我們相信這就是自由軟體最基本的目標之一，而且我們會盡全力去支持它。

在我們 source tree 中有部份的程式碼是採用所謂的GNU General Public License (GPL) 或 GNU Library General Public License (LGPL)版權宣告，雖然這些版權宣告是用來保障而非限制使用者的權利，畢竟是不那麼自由了些。由於這些 GPL 的軟體在商業使用上會引起非常複雜的版權問題，因此只要有機會，我們會盡量以採用比較鬆的 FreeBSD 版權 的軟體來取代這些 GPL 版權宣告的軟體。

1.4. FreeBSD 可以取代我現在在用的作業系統嗎？

對大部份的人來說是這樣沒錯，但事實上這問題並沒有這麼好回答。

大部份的人並不是真正在使用一個作業系統。他們使用的是應用程式；而那些應用程式才是真正用到作業系統的東西。FreeBSD 是設計用來提供一個強韌且功能完整的作業環境給應用程式來執行。它支援了多種瀏覽器，辦公室套件軟體，電子郵件閱讀軟體，繪圖程式，程式設計環境，網路伺服器軟體，以及幾乎所有你想要的東西。大部份的程式都可以靠 Ports Collection 來管理。

但是如果你想要使用的應用程式只能在某個特定的作業系統上面執行的話，你就不能輕易地把它換掉，或者指望在 FreeBSD 上有很相似的應用程式才有機會。如果你想要的是一個強健的辦公室或是網路伺服器，或是一部穩定的工作站，FreeBSD 無疑是您的最佳選擇。世界各地有很多使用者，包括初學或資深的 UNIX™ 管理人員都選用 FreeBSD 當他們唯一的桌上作業系統。

如果你是從其他的 UNIX™-like 環境轉換到 FreeBSD 的話會很熟悉。 Windows™ 或是 Mac OS™ 的使用者可能會對 GhostBSD 、 MidnightBSD 、 NomadBSD 有興趣，他是基於 FreeBSD 的一個桌面環境發行版，非UNIX™ 使用者可能就要多花一點時間來學習怎麼用 UNIX™ 的

方法來做事。你可以從這份 FAQ 和 FreeBSD 使用手冊 來入門。

1.5. 為什麼要叫做 FreeBSD？

- 您可以免費使用它，即使是用於商業用途。
- 整個 FreeBSD
  作業系統完整的原始程式都可以免費取得，而且不管是在使用，散佈或是整合進其他程式等各方面也只受到最小的限制 (不論是否用於商業用途)。
- 任何人都可以自由地把他對系統的改良或錯誤修正的程式碼加入 source tree 之中
  (當然要符合幾個先決條件)。

特別值得注意的是這裡的 “free” 出現了兩次，而且它們 的意思是不一樣的：一種代表 “免費”，另一種代表 “自由”。您可以拿 FreeBSD 去做任何您想要做的事，除了一些例外，例如您宣稱 FreeBSD 是您寫的。

1.6. FreeBSD 及 NetBSD, OpenBSD 以及其他 open source BSD 作業系統之間有何不同之處呢？

James Howard 寫了一篇關於不同計畫的差異和歷史淵源的好文章叫 The BSD Family Tree 可以回答這個問題。雖然有些資訊有點過時，但是關於歷史淵源的部份仍是相當正確的。

時至今日，大部分的 BSD 家族仍是共用修補和程式碼的。這些 BSD 家族有著共同的祖先。

FreeBSD 的設計目的如 發展 FreeBSD 計畫的目的是什麼 所述。其他 BSD 家族的設計目的如下所述：

- OpenBSD 目標在作業系統的安全性。OpenBSD團隊寫的 ssh(1) 和 pf(4) 都移植到了 FreeBSD。
- NetBSD 目標在易於移植到其他的硬體平台。
- DragonFly BSD 是 FreeBSD 4.8 的一個分支，發展出許多有趣的特色，包括 HAMMER 檔案系統和支援 "vkernels" 使用者模式。

1.7. 最新版的 FreeBSD 是那一版？

在 FreeBSD 開發的任何時間點，都有多個平行的分支。13.X releases 是從 13-STABLE 分支而來，而 12.X releases 是從 12-STABLE 分支而來。

在 12.0 之前，12.X 系列仍屬 -STABLE分支。 然而從14.X 發行開始，12.X 將只著重在重大問題上(比如：漏洞修補、安全維護)以及所謂的 "extended support" 。

Version 12.0 is the latest release from the 12-STABLE branch; it was released in December 2018.
Version 10.4 is the latest release from the 11-STABLE branch; it was released in October 2017.

Releases 版 每幾個月 才會發行一次。 雖然如此，有很多人和 FreeBSD 原始碼同步更新 (詳見 FreeBSD-CURRENT 和 FreeBSD-STABLE的相關問題)
，但因為原始碼是一直不斷地在變動的，所以如果要這麼做的話得要花上更多的精力。

其他更多相關 FreeBSD 發行情報，可由 FreeBSD 網站上的 Release Engineering 頁面 和 release(7)得知。

1.8. 什麼是 FreeBSD-CURRENT?

FreeBSD-CURRENT 指的是正在發展中的作業系統版本，它終將在適當的時機成為 FreeBSD-STABLE 分支。它實在是只適合給系統發展者以及有毅力的業餘愛好者使用 。如果想要得到有關如何使用-CURRENT的深入資訊，請參考使用手冊的相關部份。

如果您對 FreeBSD 本身並不是很熟悉那麼您就不應該使用FreeBSD-CURRENT。
這個分支的程式碼有時候變動得很快，而且可能會因此
而使您有好幾天的時間無法更新您的系統。我們假設使用 FreeBSD-CURRENT
的使用者都有能力去分析他們所遇到的問題，除錯，並且回報問題。

1.9. 什麼是 FreeBSD-STABLE？

FreeBSD-STABLE
是由主要發行版這個開發分支來的。他的變動較慢，而且一般來說假設他們都已經先在FreeBSD-CURRENT測試過了。然而在任何時候，FreeBSD-STABLE的原始碼仍有可能不一定適合一般用途，因為他可能包含在 FreeBSD-CURRENT 沒有發現到的錯誤。沒有能力和資源的使用者應該改使用 FreeBSD 正式發行版。另一方面，FreeBSD-CURRENT 從2.0開始就是另一個分支。

更多關於開發分支的資訊請見 "FreeBSD Release Engineering: Creating the Release Branch"，分支的開發狀態和接下來的發行計畫時間表可以在 Release Engineering 資訊 找到。

12.3 版本是 12-STABLE 分支最新的正式發行版本，在 2021年12月7日發行。13.1 版本是 13-stable 分支最新的正式發行版本，在 2022年5月16日發行。

1.10. 每次新的 FreeBSD 將於什麼時候推出？

一般而言，Release Engineering Team <re@FreeBSD.org>
約每18個月發行一次主要發行版本，約平均每8個月發行一次次要發行版本。每次新版本的發表時程都會事先公告， 相關的開發人員就會知道，什麼時候該先把手邊的計劃完成並且測試過，
此外，這些更動都已經完整地測試過，確保新功能不會影響系統的穩定度。 雖然，等這些好東西進入 -STABLE 的時間令人等得有些不耐煩， 但是大多數的使用者都認為這種謹慎的態度是 FreeBSD 最好的優點之一。

有關發行情報的更多細節部分(包括 release 的行程表、進度)，都可在 FreeBSD 網站上的 發行情報 上面獲得。

為了滿足那些需要 (或想要) 新鮮刺激感的使用者，上面已經提到我們每周都會發行 snapshots 版可供使用。

1.11. FreeBSD snapshots 版本什麼時候推出?

我們會根據目前-CURRENT 和 -STABLE 的狀況發行 snapshot 版。發表這些 snapshot 的目的在於:

- 測試最新版的安裝程式。
- 提供一個簡單的方法給那些喜歡使用 -CURRENT 或是 -STABLE 但是沒有時間和頻寬去每天升級的使用者。
- 為了替我們發展中的程式保留一個固定的參考點，以防止我們未來不幸搞砸了。(雖然一般而言 Subversion 可以防止類似者種的可怕事件)。
- 為了確保所有需要測試的新功能或修正都可以得到最多的測試。

我們不對 -CURRENT snapshot 做任何的 "品質保證"。如果你想要的是一個穩定且經過充分測試過的系統的話，最好選擇使用完整 releases。

您可以從 snapshot 取得 snapshot release。

對每個有在活動的分支，都會定期產生官方的 snapshots。

1.12. 誰負責 FreeBSD 的發展？

如果是一些有關 FreeBSD 計畫的關鍵性決定，像是整個計畫的走向 或是決定誰可以改 source tree 裡的程式碼這類的事，是由一個由 9 個 人所組成的 core team 來決定。而有另一群超過 350 個人的 committers 有權利可以直接修改 FreeBSD 的 source tree。

無論如何，大多數的改變都會事前在 郵件論壇先討論過，而且不分角色，每個人都可以參與討論。

1.13. 我要如何取得 FreeBSD？

所有受到支援的版本都在 FreeBSD release locator page:

- 最新的 13-STABLE 和 13.1-RELEASE 正式發行版本在 the appropriate architecture and installation mode for 13.1-RELEASE。
- 最新的 11-STABLE 和 10.4-RELEASE 正式發行版本在 the appropriate architecture and installation mode for 12.3-RELEASE。
- -CURRENT 和 -STABLE 分支的Snapshot版本通常每個月會做一次，主要是為了提供給那些熱心的測試者和開發人員。

FreeBSD 的 CD、DVD，還有其他取得方式可以在 the Handbook 中找到解答。

1.14. 我要如何去查詢、提交問題回報(Problem Report，簡稱PR)資料庫呢？

所有使用者的變更要求都可以經由網頁版的 PR 查詢 界面來察看。

可以使用瀏覽器經由網頁版的 PR 界面 來傳送問題回報

然而，在您回報問題之前，請先閱讀 如何撰寫 FreeBSD 的問題回報單，這是一篇告訴你怎樣才能寫出一篇真正有用的問題回報單。

Chapter 2. 文件與技術支援

2.1. 有哪些 FreeBSD 相關的好書呢？

FreeBSD 文件計畫已陸續發表了相當廣泛範圍的文件，可在 https://www.FreeBSD.org/docs/ 取得。除此之外，也可以參閱使用手冊的 參考書目建議的其他書籍。

2.2. 這些文件有其他格式的嗎？像是：純文字(ASCII)或 PostScript 之類的格式？

有的。這些文件都分別以不同格式儲存以及壓縮處理並放在 FTP 上面，可以從各個 FreeBSD FTP 站的 /pub/FreeBSD/doc/ 目錄內找到你要的。

文件以幾種不同的方式分類。包括：

- 文件名稱，例如：faq (常見問答集)或是 handbook (FreeBSD 使用手冊)等等。
- 文件的語言與編碼。他們是基於 FreeBSD 系統中 /usr/shared/locale 裡所見到的語系名稱。目前包含的語言與編碼如下：

語系名稱	說明
en_US.ISO8859-1	英文 (美國)
bn_BD.ISO10646-1	孟加拉文 (孟加拉)
da_DK.ISO8859-1	丹麥文 (丹麥)
de_DE.ISO8859-1	德文 (德國)
el_GR.ISO8859-7	希臘文 (希臘)
es_ES.ISO8859-1	西班牙文 (西班牙)
fr_FR.ISO8859-1	法文 (法國)
hu_HU.ISO8859-2	匈牙利文 (匈牙利)
it_IT.ISO8859-15	義大利文 (義大利)
ja_JP.eucJP	日文 (日本, EUC 編碼)
ko_KR.UTF-8	韓文 (韓國, UTF-8 編碼)
mn_MN.UTF-8	蒙古文 (蒙古, UTF-8 編碼)
nl_NL.ISO8859-1	荷蘭文 (荷蘭)
pl_PL.ISO8859-2	波蘭文 (波蘭)
pt_BR.ISO8859-1	葡萄牙文 (巴西)
ru_RU.KOI8-R	俄文 (俄羅斯, KOI8-R 編碼)
tr_TR.ISO8859-9	土耳其文 (土耳其)
zh_CN.UTF-8	簡體中文 (中國, UTF-8 編碼)
zh_TW.UTF-8	正體中文 (台灣, UTF-8 編碼)

> 上列的各國翻譯語系文件中，並非所有文件都有翻譯。

- 文件的格式。我們的每份文件都提供許多不同的格式，每種格式各有利弊，有些格式適合線上閱讀，有些則適合列印出美觀的文件。這些不同格式的文件能夠確保我們的讀者們，無論是在螢幕上閱讀或是列印成紙本，都能夠閱讀他們感興趣的內容，目前有提供的格式如下：

格式	說明
html-split	依章節區分成多個小的、互相連結的 HTML 檔案
html	所有內容包含在單一個 HTML 檔案
pdf	Adobe’s PDF 格式
txt	純文字

- 壓縮和打包方式
  a. 當採用 html-split 格式時，檔案先透過 tar(1) 工具來進行打包。接著再將產生出來的 .tar 檔接透過第二點所述的壓縮方式壓縮。
  b. 其他的格式都是單一個檔案。例如 article.pdf、book.html ，以此類推。

     這些檔案接著透過 zip 或 bz2 來壓縮。 tar(1) 工具可用來解壓縮這些檔案。

     因此 PDF 版本的手冊經過 bzip2 壓縮後會存成一個叫做 book.ps.bz2 的檔案，並位於 handbook/資料夾。

在選取格式與壓縮方式後，下載壓縮後的檔案並解壓縮，再把文件複製到想要的地方。

舉例來說，透過 bzip2(1) 壓縮的英文問與答的章節分割 HTML 版本，可以在 doc/en_US.ISO8859-1/books/faq/book.html-split.tar.bz2 中找到。若要下載並解壓縮這個檔案，請輸入

```
# fetch https://download.freebsd.org/doc/en_US.ISO8859-1/books/faq/book.html-split.tar.bz2
# tar xvf book.html-split.tar.bz2
```

如果檔案被壓縮過的話，tar 會自動偵測正確的格式並解壓縮出一堆 .html 檔案。主要的檔案是 index.html，包含了主目錄跟介紹以及連接到文件其他部份的連結。

2.3. 哪裡有關於 FreeBSD 的郵遞論壇(mailing lists)呢？有哪些可以使用的 FreeBSD 新聞群組(news groups)呢?

請參考 FreeBSD 使用者手冊上的 郵件論壇 (mailing-lists) 和 新聞群組 (news-groups) 。

2.4. 有 FreeBSD IRC (Internet Relay Chat)頻道嗎？

有的，大部分的 IRC 主機都有 FreeBSD 聊天頻道， FreeBSD wiki 有最新的 list of IRC channels 。

每個頻道都是不同且互相獨立的。因為他們的聊天風格不同，您可以每個都試試看來找到適合您的頻道。

2.5. 有沒有任何網頁形式的 FreeBSD 論壇呢？

官方的 FreeBSD 論壇位於 https://forums.FreeBSD.org/。

2.6. 可以從哪邊獲得商業化的 FreeBSD 的教育課程及技術支援呢？

iXsystems, Inc.， FreeBSD 商城的母公司，提供 FreeBSD 開發與調校解決方案與 FreeBSD 與 TrueOS 的軟體 支援。

BSD Certification Group, Inc. 提供 DragonFly BSD、FreeBSD、NetBSD 與 OpenBSD 的系統管理認證。請參閱 他們的網站 來獲得更多資訊。

如果有其他組織提供技術訓練或技術支援，請聯絡 FreeBSD 計畫來加入以上清單。

Chapter 3. 安裝

3.1. Which platform should I download? I have a 64 bit capable Intel CPU, but I only see amd64.

amd64 is the term FreeBSD uses for 64-bit compatible x86 architectures (also known as "x86-64" or "x64"). Most modern computers should use amd64. Older hardware should use i386. When installing on a non-x86-compatible architecture, select the platform which best matches the hardware.

3.2. Which file do I download to get FreeBSD?

On the Getting FreeBSD page, select [iso] next to the architecture that matches the hardware.

Any of the following can be used:

檔案	描述
disc1.iso	Contains enough to install FreeBSD and a minimal set of packages.
dvd1.iso	Similar to disc1.iso but with additional packages.
memstick.img	A bootable image sufficient for writing to a USB stick.
bootonly.iso	A minimal image that requires network access during installation to completely install FreeBSD.

Full instructions on this procedure and a little bit more about installation issues in general can be found in the Handbook entry on installing FreeBSD.

3.3. What do I do if the install image does not boot?

This can be caused by not downloading the image in binary mode when using FTP.

Some FTP clients default their transfer mode to ascii and attempt to change any end-of-line characters received to match the conventions used by the client’s system. This will almost invariably corrupt the boot image. Check the SHA-256 checksum of the downloaded boot image: if it is not exactly that on the server, then the download process is suspect.

When using a command line FTP client, type binary at the FTP command prompt after getting connected to the server and before starting the download of the image.

3.4. 可以在哪邊找到安裝 FreeBSD 的解說步驟呢？

安裝說明可以在 使用手冊的安裝 FreeBSD 找到。

3.5. 要跑 FreeBSD 至少需要什麼樣的配備呢？

FreeBSD 需要 486 以上的 PC，64 MB 以上的 RAM，和至少 1.1 GB 的硬碟空間。

3.6. 要怎樣才能自行打造專用的安裝磁片呢？

可以透過編譯客製化發行版本來建立客製化的 FreeBSD 安裝媒體。請參閱 Release Engineering 文章的說明。

3.7. Windows 可以與 FreeBSD 共存嗎？

如果 Windows™ 先安裝，那就可以。 FreeBSD 的開機管理程式將會管理 Windows™ 和 FreeBSD 的開機啟動。 如果 Windows™ 後安裝，它將覆蓋開機管理程式。如果發生這種情況，請見下一小節。

3.8. Another operating system destroyed my Boot Manager. How do I get it back?

This depends upon the boot manager. The FreeBSD boot selection menu can be reinstalled using boot0cfg(8). For example, to restore the boot menu onto the disk ada0:

```
# boot0cfg -B ada0
```

The non-interactive MBR bootloader can be installed using gpart(8):

```
# gpart bootcode -b /boot/mbr ada0
```

For more complex situations, including GPT disks, see gpart(8).

3.9. 我需要安裝完整的原始碼嗎？

In general, no. There is nothing in the base system which requires the presence of the source to operate. Some ports, like sysutils/lsof, will not build unless the source is installed. In particular, if the port builds a kernel module or directly operates on kernel structures, the source must be installed.

3.10. 需要重新 build kernel 嗎？

Usually not. The supplied GENERIC kernel contains the drivers an ordinary computer will need. freebsd-update(8), the FreeBSD binary upgrade tool, cannot upgrade custom kernels, another reason to stick with the GENERIC kernel when possible. For computers with very limited RAM, such as embedded systems, it may be worthwhile to build a smaller custom kernel containing just the required drivers.

3.11. Should I use DES, Blowfish, or MD5 passwords and how do I specify which form my users receive?

FreeBSD uses SHA512 by default. DES passwords are still available for backwards compatibility with operating systems that still use the less secure password format. FreeBSD also supports the Blowfish and MD5 password formats. Which password format to use for new passwords is controlled by the passwd_format login capability in /etc/login.conf, which takes values of des, blf (if these are available) or md5. See the login.conf(5) manual page for more information about login capabilities.

3.12. What are the limits for FFS file systems?

For FFS file systems, the largest file system is practically limited by the amount of memory required to fsck(8) the file system. fsck(8) requires one bit per fragment, which with the default fragment size of 4 KB equates to 32 MB of memory per TB of disk. This does mean that on architectures which limit userland processes to 2 GB (e.g., i386™), the maximum fsck(8)'able filesystem is ~60 TB.

If there was not a fsck(8) memory limit the maximum filesystem size would be 2 ^ 64 (blocks) * 32 KB ⇒ 16 Exa * 32 KB ⇒ 512 ZettaBytes.

The maximum size of a single FFS file is approximately 2 PB with the default block size of 32 KB. Each 32 KB block can point to 4096 blocks. With triple indirect blocks, the calculation is 32 KB * 12 + 32 KB * 4096 + 32 KB * 4096^2 + 32 KB * 4096^3. Increasing the block size to 64 KB will increase the max file size by a factor of 16.

3.13. Why do I get an error message, readin failed after compiling and booting a new kernel?

The world and kernel are out of sync. This is not supported. Be sure to use make buildworld and make buildkernel to update the kernel.

Boot the system by specifying the kernel directly at the second stage, pressing any key when the | shows up before loader is started.

3.14. 是否有工具可以執行安裝後的設定工作嗎？

是的。bsdconfig 提供很棒的介面來進行 FreeBSD 安裝後設定。

Chapter 4. 硬體相容性

4.1. 一般問題

4.1.1. I want to get a piece of hardware for my FreeBSD system. Which model/brand/type is best?

This is discussed continually on the FreeBSD mailing lists but is to be expected since hardware changes so quickly. Read through the Hardware Notes for FreeBSD 12.0 or 10.4 and search the mailing list archives before asking about the latest and greatest hardware. Chances are a discussion about that type of hardware took place just last week.

Before purchasing a laptop, check the archives for FreeBSD laptop computer mailing list and FreeBSD general questions mailing list, or possibly a specific mailing list for a particular hardware type.

4.1.2. What are the limits for memory? Does FreeBSD support more than 4 GB of memory (RAM)? More than 16 GB? More than 48 GB?

FreeBSD as an operating system generally supports as much physical memory (RAM) as the platform it is running on does. Keep in mind that different platforms have different limits for memory; for example i386™ without PAE supports at most 4 GB of memory (and usually less than that because of PCI address space) and i386™ with PAE supports at most 64 GB memory. As of FreeBSD 10, AMD64 platforms support up to 4 TB of physical memory.

4.1.3. Why does FreeBSD report less than 4 GB memory when installed on an i386 machine?

The total address space on i386™ machines is 32-bit, meaning that at most 4 GB of memory is addressable (can be accessed). Furthermore, some addresses in this range are reserved by hardware for different purposes, for example for using and controlling PCI devices, for accessing video memory, and so on. Therefore, the total amount of memory usable by the operating system for its kernel and applications is limited to significantly less than 4 GB. Usually, 3.2 GB to 3.7 GB is the maximum usable physical memory in this configuration.

To access more than 3.2 GB to 3.7 GB of installed memory (meaning up to 4 GB but also more than 4 GB), a special tweak called PAE must be used. PAE stands for Physical Address Extension and is a way for 32-bit x86 CPUs to address more than 4 GB of memory. It remaps the memory that would otherwise be overlaid by address reservations for hardware devices above the 4 GB range and uses it as additional physical memory (see pae(4)). Using PAE has some drawbacks; this mode of memory access is a little bit slower than the normal (without PAE) mode and loadable modules (see kld(4)) are not supported. This means all drivers must be compiled into the kernel.

The most common way to enable PAE is to build a new kernel with the special ready-provided kernel configuration file called PAE, which is already configured to build a safe kernel. Note that some entries in this kernel configuration file are too conservative and some drivers marked as unready to be used with PAE are actually usable. A rule of thumb is that if the driver is usable on 64-bit architectures (like AMD64), it is also usable with PAE. When creating a custom kernel configuration file, PAE can be enabled by adding the following line:

```
options     PAE
```

PAE is not much used nowadays because most new x86 hardware also supports running in 64-bit mode, known as AMD64 or Intel™ 64. It has a much larger address space and does not need such tweaks. FreeBSD supports AMD64 and it is recommended that this version of FreeBSD be used instead of the i386™ version if 4 GB or more memory is required.

4.2. Architectures and Processors

4.2.1. Does FreeBSD support architectures other than the x86?

Yes. FreeBSD divides support into multiple tiers. Tier 1 architectures, such as i386 or amd64; are fully supported. Tiers 2 and 3 are supported on a best-effort basis. A full explanation of the tier system is available in the Committer' s Guide.

A complete list of supported architectures can be found on the platforms page.

4.2.2. Does FreeBSD support Symmetric Multiprocessing (SMP)?

FreeBSD supports symmetric multi-processor (SMP) on all non-embedded platforms (e.g, i386, amd64, etc.). SMP is also supported in arm and MIPS kernels, although some CPUs may not support this. FreeBSD' s SMP implementation uses fine-grained locking, and performance scales nearly linearly with number of CPUs.

smp(4) has more details.

4.2.3. What is microcode? How do I install Intel CPU microcode updates?

Microcode is a method of programmatically implementing hardware level instructions. This allows for CPU bugs to be fixed without replacing the on board chip.

Install sysutils/devcpu-data, then add:

```
microcode_update_enable="YES"
```

to /etc/rc.conf

4.3. Hard Drives, Tape Drives, and CD and DVD Drives

4.3.1. What kind of hard drives does FreeBSD support?

FreeBSD supports EIDE, SATA, SCSI, and SAS drives (with a compatible controller; see the next section), and all drives using the original "Western Digital" interface (MFM, RLL, ESDI, and of course IDE). A few ESDI controllers that use proprietary interfaces may not work: stick to WD1002/3/6/7 interfaces and clones.

4.3.2. Which SCSI or SAS controllers are supported?

See the complete list in the Hardware Notes for FreeBSD 12.0 or 10.4.

4.3.3. What types of tape drives are supported?

FreeBSD supports all standard SCSI tape interfaces.

4.3.4. Does FreeBSD support tape changers?

FreeBSD supports SCSI changers using the ch(4) device and the chio(1) command. The details of how to control the changer can be found in chio(1).

While AMANDA and some other products already understands changers, other applications only know how to move a tape from one point to another. In this case, keep track of which slot a tape is in and which slot the tape currently in the drive needs to go back to.

4.3.5. Which CD-ROM and CD-RW drives are supported by FreeBSD?

Any SCSI drive connected to a supported controller is supported. Most ATAPI compatible IDE CD-ROMs are supported.

FreeBSD supports any ATAPI-compatible IDE CD-R or CD-RW drive.

FreeBSD also supports any SCSI CD-R or CD-RW drives. Install the sysutils/cdrtools port or package, then use cdrecord.

4.4. Keyboards and Mice

4.4.1. Is it possible to use a mouse outside the X Window system?

The default console driver, syscons(4), provides the ability to use a mouse pointer in text consoles to cut & paste text. Run the mouse daemon, moused(8), and turn on the mouse pointer in the virtual console:

```
# moused -p /dev/xxxx -t yyyy
# vidcontrol -m on
```

Where xxxx is the mouse device name and yyyy is a protocol type for the mouse. The mouse daemon can automatically determine the protocol type of most mice, except old serial mice. Specify the auto protocol to invoke automatic detection. If automatic detection does not work, see the moused(8) manual page for a list of supported protocol types.

For a PS/2 mouse, add moused_enable="YES" to /etc/rc.conf to start the mouse daemon at boot time. Additionally, to use the mouse daemon on all virtual terminals instead of just the console, add allscreens_flags="-m on" to /etc/rc.conf.

When the mouse daemon is running, access to the mouse must be coordinated between the mouse daemon and other programs such as X Windows. Refer to the FAQWhy does my mouse not work with X? for more details on this issue.

4.4.2. How do I cut and paste text with a mouse in the text console?

It is not possible to remove data using the mouse. However, it is possible to copy and paste. Once the mouse daemon is running as described in the previous question, hold down button 1 (left button) and move the mouse to select a region of text. Then, press button 2 (middle button) to paste it at the text cursor. Pressing button 3 (right button) will "extend" the selected region of text.

If the mouse does not have a middle button, it is possible to emulate one or remap buttons using mouse daemon options. See the moused(8) manual page for details.

4.5. My mouse has a fancy wheel and buttons. Can I use them in FreeBSD?

The answer is, unfortunately, "It depends". These mice with additional features require specialized driver in most cases. Unless the mouse device driver or the user program has specific support for the mouse, it will act just like a standard two, or three button mouse.

For the possible usage of wheels in the X Window environment, refer to that section.

4.5.1. How do I use my delete key in sh and csh?

For the Bourne Shell, add the following lines to ~/.shrc. See sh(1) and editrc(5).

```
bind ^? ed-delete-next-char # for console
bind ^[[3~ ed-delete-next-char # for xterm
```

For the C Shell, add the following lines to ~/.cshrc. See csh(1).

```
bindkey ^? delete-char # for console
bindkey ^[[3~ delete-char # for xterm
```

For more information, see this page.

4.6. Other Hardware

4.6.1. Workarounds for no sound from my pcm4 sound card?

Some sound cards set their output volume to 0 at every boot. Run the following command every time the machine boots:

```
# mixer pcm 100 vol 100 cd 100
```

4.6.2. Does FreeBSD support power management on my laptop?

FreeBSD supports the ACPI features found in modern hardware. Further information can be found in acpi(4).

Chapter 5. Troubleshooting

5.1. Why is FreeBSD finding the wrong amount of memory on i386 hardware?

The most likely reason is the difference between physical memory addresses and virtual addresses.

The convention for most PC hardware is to use the memory area between 3.5 GB and 4 GB for a special purpose (usually for PCI). This address space is used to access PCI hardware. As a result real, physical memory cannot be accessed by that address space.

What happens to the memory that should appear in that location is hardware dependent. Unfortunately, some hardware does nothing and the ability to use that last 500 MB of RAM is entirely lost.

Luckily, most hardware remaps the memory to a higher location so that it can still be used. However, this can cause some confusion when watching the boot messages.

On a 32-bit version of FreeBSD, the memory appears lost, since it will be remapped above 4 GB, which a 32-bit kernel is unable to access. In this case, the solution is to build a PAE enabled kernel. See the entry on memory limits for more information.

On a 64-bit version of FreeBSD, or when running a PAE-enabled kernel, FreeBSD will correctly detect and remap the memory so it is usable. During boot, however, it may seem as if FreeBSD is detecting more memory than the system really has, due to the described remapping. This is normal and the available memory will be corrected as the boot process completes.

5.2. Why do my programs occasionally die with Signal 11 errors?

Signal 11 errors are caused when a process has attempted to access memory which the operating system has not granted it access to. If something like this is happening at seemingly random intervals, start investigating the cause.

These problems can usually be attributed to either:

1. If the problem is occurring only in a specific custom application, it is probably a bug in the code.
2. If it is a problem with part of the base FreeBSD system, it may also be buggy code, but more often than not these problems are found and fixed long before us general FAQ readers get to use these bits of code (that is what -CURRENT is for).

It is probably not a FreeBSD bug if the problem occurs compiling a program, but the activity that the compiler is carrying out changes each time.

For example, if make buildworld fails while trying to compile ls.c into ls.o and, when run again, it fails in the same place, this is a broken build. Try updating source and try again. If the compile fails elsewhere, it is almost certainly due to hardware.

In the first case, use a debugger such as gdb(1) to find the point in the program which is attempting to access a bogus address and fix it.

In the second case, verify which piece of hardware is at fault.

Common causes of this include:

1. The hard disks might be overheating: Check that the fans are still working, as the disk and other hardware might be overheating.
2. The processor running is overheating: This might be because the processor has been

overclocked, or the fan on the processor might have died. In either case, ensure that the hardware is running at what it is specified to run at, at least while trying to solve this problem. If it is not, clock it back to the default settings.)

Regarding overclocking, it is far cheaper to have a slow system than a fried system that needs replacing! Also the community is not sympathetic to problems on overclocked systems.

3. Dodgy memory: if multiple memory SIMMS/DIMMS are installed, pull them all out and try running the machine with each SIMM or DIMM individually to narrow the problem down to either the problematic DIMM/SIMM or perhaps even a combination.
4. Over-optimistic motherboard settings: the BIOS settings, and some motherboard jumpers, provide options to set various timings. The defaults are often sufficient, but sometimes setting the wait states on RAM too low, or setting the "RAM Speed: Turbo" option will cause strange behavior. A possible idea is to set to BIOS defaults, after noting the current settings first.
5. Unclean or insufficient power to the motherboard. Remove any unused I/O boards, hard disks, or CD-ROMs, or disconnect the power cable from them, to see if the power supply can manage a smaller load. Or try another power supply, preferably one with a little more power. For instance, if the current power supply is rated at 250 Watts, try one rated at 300 Watts.

Read the section on Signal 11 for a further explanation and a discussion on how memory testing software or hardware can still pass faulty memory. There is an extensive FAQ on this at the SIG11 problem FAQ.

Finally, if none of this has helped, it is possibly a bug in FreeBSD. Follow these instructions to send a problem report.

5.3. My system crashes with either Fatal trap 12: page fault in kernel mode, or panic:, and spits out a bunch of information. What should I do?

The FreeBSD developers are interested in these errors, but need more information than just the error message. Copy the full crash message. Then consult the FAQ section on kernel panics, build a debugging kernel, and get a backtrace. This might sound difficult, but does not require any programming skills. Just follow the instructions.

5.4. What is the meaning of the error maxproc limit exceeded by uid %i, please see tuning(7) and login.conf(5)?

The FreeBSD kernel will only allow a certain number of processes to exist at one time. The number is based on the kern.maxusers sysctl(8) variable. kern.maxusers also affects various other in-kernel limits, such as network buffers. If the machine is heavily loaded, increase kern.maxusers. This will increase these other system limits in addition to the maximum number of processes.

To adjust the kern.maxusers value, see the File/Process Limits section of the Handbook. While that section refers to open files, the same limits apply to processes.

If the machine is lightly loaded but running a very large number of processes, adjust the kern.maxproc tunable by defining it in /boot/loader.conf. The tunable will not get adjusted until the system is rebooted. For more information about tuning tunables, see loader.conf(5). If these processes are being run by a single user, adjust kern.maxprocperuid to be one less than the new kern.maxproc value. It must be at least one less because one system program, init(8), must always be running.

5.5. Why do full screen applications on remote machines misbehave?

The remote machine may be setting the terminal type to something other than xterm which is required by the FreeBSD console. Alternatively the kernel may have the wrong values for the width and height of the terminal.

Check the value of the TERM environment variable is xterm. If the remote machine does not support that try vt100.

Run stty -a to check what the kernel thinks the terminal dimensions are. If they are incorrect, they can be changed by running stty rows RR cols CC.

Alternatively, if the client machine has x11/xterm installed, then running resize will query the terminal for the correct dimensions and set them.

5.6. Why does it take so long to connect to my computer via ssh or telnet?

The symptom: there is a long delay between the time the TCP connection is established and the time when the client software asks for a password (or, in telnet(1)'s case, when a login prompt appears).

The problem: more likely than not, the delay is caused by the server software trying to resolve the client' s IP address into a hostname. Many servers, including the Telnet and SSH servers that come with FreeBSD, do this to store the hostname in a log file for future reference by the administrator.

The remedy: if the problem occurs whenever connecting the client computer to any server, the problem is with the client. If the problem only occurs when someone connects to the server computer, the problem is with the server.

If the problem is with the client, the only remedy is to fix the DNS so the server can resolve it. If this is on a local network, consider it a server problem and keep reading. If this is on the Internet, contact your ISP.

If the problem is with the server on a local network, configure the server to resolve address-to-hostname queries for the local address range. See hosts(5) and named(8) for more information. If this is on the Internet, the problem may be that the local server' s resolver is not functioning correctly. To check, try to look up another host such as www.yahoo.com. If it does not work, that is the problem.

Following a fresh install of FreeBSD, it is also possible that domain and name server information is missing from /etc/resolv.conf. This will often cause a delay in SSH, as the option UseDNS is set to yes by default in /etc/ssh/sshd_config. If this is causing the problem, either fill in the missing information in /etc/resolv.conf or set UseDNS to no in sshd_config as a temporary workaround.

5.7. Why does file: table is full show up repeatedly in dmesg8?

This error message indicates that the number of available file descriptors have been exhausted on the system. Refer to the kern.maxfiles section of the Tuning Kernel Limits section of the Handbook for a discussion and solution.

5.8. Why does the clock on my computer keep incorrect time?

The computer has two or more clocks, and FreeBSD has chosen to use the wrong one.

Run dmesg(8), and check for lines that contain Timecounter. The one with the highest quality value that FreeBSD chose.

```
# dmesg | grep Timecounter
Timecounter "i8254" frequency 1193182 Hz quality 0
Timecounter "ACPI-fast" frequency 3579545 Hz quality 1000
Timecounter "TSC" frequency 2998570050 Hz quality 800
Timecounters tick every 1.000 msec
```

Confirm this by checking the kern.timecounter.hardware sysctl(3).

```
# sysctl kern.timecounter.hardware
kern.timecounter.hardware: ACPI-fast
```

It may be a broken ACPI timer. The simplest solution is to disable the ACPI timer in /boot/loader.conf:

```
debug.acpi.disabled="timer"
```

Or the BIOS may modify the TSC clock—perhaps to change the speed of the processor when running from batteries, or going into a power saving mode, but FreeBSD is unaware of these adjustments, and appears to gain or lose time.

In this example, the i8254 clock is also available, and can be selected by writing its name to the kern.timecounter.hardware sysctl(3).

```
# sysctl kern.timecounter.hardware=i8254
kern.timecounter.hardware: TSC -> i8254
```

The computer should now start keeping more accurate time.

To have this change automatically run at boot time, add the following line to /etc/sysctl.conf:

```
kern.timecounter.hardware=i8254
```

5.9. What does the error swap_pager: indefinite wait buffer: mean?

This means that a process is trying to page memory from disk, and the page attempt has hung trying to access the disk for more than 20 seconds. It might be caused by bad blocks on the disk drive, disk wiring, cables, or any other disk I/O-related hardware. If the drive itself is bad, disk errors will appear in /var/log/messages and in the output of dmesg. Otherwise, check the cables and connections.

5.10. What is a lock order reversal?

The FreeBSD kernel uses a number of resource locks to arbitrate contention for certain resources. When multiple kernel threads try to obtain multiple resource locks, there’s always the potential

for a deadlock, where two threads have each obtained one of the locks and blocks forever waiting for the other thread to release one of the other locks. This sort of locking problem can be avoided if all threads obtain the locks in the same order.

A run-time lock diagnostic system called witness(4), enabled in FreeBSD-CURRENT and disabled by default for stable branches and releases, detects the potential for deadlocks due to locking errors, including errors caused by obtaining multiple resource locks with a different order from different parts of the kernel. The witness(4) framework tries to detect this problem as it happens, and reports it by printing a message to the system console about a lock order reversal (often referred to also as LOR).

It is possible to get false positives, as witness(4) is conservative. A true positive report does not mean that a system is dead-locked; instead it should be understood as a warning that a deadlock could have happened here.

Problematic LORs tend to get fixed quickly, so check the FreeBSD-CURRENT mailing list before posting to it.

5.11. What does Called … with the following non-sleepable locks held mean?

This means that a function that may sleep was called while a mutex (or other unsleepable) lock was held.

The reason this is an error is because mutexes are not intended to be held for long periods of time; they are supposed to only be held to maintain short periods of synchronization. This programming contract allows device drivers to use mutexes to synchronize with the rest of the kernel during interrupts. Interrupts (under FreeBSD) may not sleep. Hence it is imperative that no subsystem in the kernel block for an extended period while holding a mutex.

To catch such errors, assertions may be added to the kernel that interact with the witness(4) subsystem to emit a warning or fatal error (depending on the system configuration) when a potentially blocking call is made while holding a mutex.

In summary, such warnings are non-fatal, however with unfortunate timing they could cause undesirable effects ranging from a minor blip in the system’s responsiveness to a complete system lockup.

For additional information about locking in FreeBSD see locking(9).

5.12. Why does buildworld/installworld die with the message touch: not found?

This error does not mean that the touch(1) utility is missing. The error is instead probably due to the dates of the files being set sometime in the future. If the CMOS clock is set to local time, run adjkerntz -i to adjust the kernel clock when booting into single-user mode.

Chapter 6. User Applications

6.1. Where are all the user applications?

Refer to the ports page for info on software packages ported to FreeBSD. The list currently tops 24,000 and is growing daily, so come back to check often or subscribe to the FreeBSD announcements mailing list for periodic updates on new entries.

Most ports should work on all supported versions of FreeBSD. Those that do not are specifically marked as such. Each time a FreeBSD release is made, a snapshot of the ports tree at the time of release in also included in the ports/ directory.

FreeBSD supports compressed binary packages to easily install and uninstall ports. Use pkg(7) to control the installation of packages.

6.2. How do I download the Ports tree? Should I be using Subversion?

Any of the methods listed here work:

- Use portsnap for most use cases. Refer to Using the Ports Collection for instructions on how to use this tool.
- Use Subversion if custom patches to the ports tree are needed. Refer to Using Subversion for details.

6.3. Does FreeBSD support Java?

Yes. Refer to https://www.FreeBSD.org/java/ for more information.

6.4. Why can I not build this port on my 11.X -, or 12.X -STABLE machine?

If the installed FreeBSD version lags significantly behind -CURRENT or -STABLE, update the Ports Collection using the instructions in Using the Ports Collection. If the system is up-to-date, someone might have committed a change to the port which works for -CURRENT but which broke the port for -STABLE. Submit a bug report, since the Ports Collection is supposed to work for both the -CURRENT and -STABLE branches.

6.5. I just tried to build INDEX using make index, and it failed. Why?

First, make sure that the Ports Collection is up-to-date. Errors that affect building INDEX from an up-to-date copy of the Ports Collection are high-visibility and are thus almost always fixed immediately.

There are rare cases where INDEX will not build due to odd cases involving OPTIONS_SET being set in make.conf. If you suspect that this is the case, try to make INDEX with those variables turned off before reporting it to FreeBSD ports mailing list.

6.6. I updated the sources, now how do I update my installed ports?

FreeBSD does not include a port upgrading tool, but it does have some tools to make the upgrade process somewhat easier. Additional tools are available to simplify port handling and are described the Upgrading Ports section in the FreeBSD Handbook.

6.7. Do I need to recompile every port each time I perform a major version update?

Yes! While a recent system will run with software compiled under an older release, things will randomly crash and fail to work once other ports are installed or updated.

When the system is upgraded, various shared libraries, loadable modules, and other parts of the system will be replaced with newer versions. Applications linked against the older versions may fail to start or, in other cases, fail to function properly.

For more information, see the section on upgrades in the FreeBSD Handbook.

6.8. Do I need to recompile every port each time I perform a minor version update?

In general, no. FreeBSD developers do their utmost to guarantee binary compatibility across all releases with the same major version number. Any exceptions will be documented in the Release Notes, and advice given there should be followed.

6.9. Why is /bin/sh so minimal? Why does FreeBSD not use bash or another shell?

Many people need to write shell scripts which will be portable across many systems. That is why POSIX™ specifies the shell and utility commands in great detail. Most scripts are written in Bourne shell (sh(1)), and because several important programming interfaces (make(1), system(3), popen(3), and analogues in higher-level scripting languages like Perl and Tcl) are specified to use the Bourne shell to interpret commands. Because the Bourne shell is so often and widely used, it is important for it to be quick to start, be deterministic in its behavior, and have a small memory footprint.

The existing implementation is our best effort at meeting as many of these requirements simultaneously as we can. To keep /bin/sh small, we have not provided many of the convenience features that other shells have. That is why other more featureful shells like bash, scsh, tcsh(1), and zsh are available. Compare the memory utilization of these shells by looking at the "VSZ" and "RSS" columns in a ps -u listing.

6.10. How do I create audio CDs from my MIDI files?

To create audio CDs from MIDI files, first install audio/timidity from ports then install manually the GUS patches set by Eric A. Welsh, available at http://alleg.sourceforge.net/digmid.html. After TiMidity++ has been installed properly, MIDI files may be converted to WAV files with the following command line:

```
% timidity -Ow -s 44100 -o /tmp/juke/01.wav 01.mid
```

The WAV files can then be converted to other formats or burned onto audio CDs, as described in the FreeBSD Handbook.

Chapter 7. Kernel Configuration

7.1. I would like to customize my kernel. Is it difficult?

Not at all! Check out the kernel config section of the Handbook.

> The new kernel will be installed to the /boot/kernel directory along with its modules, while the old kernel and its modules will be moved to the /boot/kernel.old directory. If a mistake is made in the configuration, simply boot the previous version of the kernel.

7.2. Why is my kernel so big?

GENERIC kernels shipped with FreeBSD are compiled in debug mode. Kernels built in debug mode contain debug data in separate files that are used for debugging. FreeBSD releases prior to 11.0 store these debug files in the same directory as the kernel itself, /boot/kernel/. In FreeBSD 11.0 and later the debug files are stored in /usr/lib/debug/boot/kernel/. Note that there will be little or no performance loss from running a debug kernel, and it is useful to keep one around in case of a system panic.

When running low on disk space, there are different options to reduce the size of /boot/kernel/ and /usr/lib/debug/.

To not install the symbol files, make sure the following line exists in /etc/src.conf:

```
WITHOUT_KERNEL_SYMBOLS=yes
```

For more information see src.conf(5).

If you want to avoid building debug files altogether, make sure that both of the following are true:

- This line does not exist in the kernel configuration file:

  ```
  makeoptions DEBUG=-g
  ```

- Do not run config(8) with -g.

Either of the above settings will cause the kernel to be built in debug mode.

To build and install only the specified modules, list them in /etc/make.conf:

```
MODULES_OVERRIDE= accf_http ipfw
```

Replace accf_httpd ipfw with a list of needed modules. Only the listed modules will be built. This reduces the size of the kernel directory and decreases the amount of time needed to build the kernel. For more information, read /usr/shared/examples/etc/make.conf.

Unneeded devices can be removed from the kernel to further reduce the size. See I would like to customize my kernel. Is it difficult? for more information.

To put any of these options into effect, follow the instructions to build and install the new kernel.

For reference, the FreeBSD 11 amd64 kernel (/boot/kernel/kernel) is approximately 25 MB.

7.3. Why does every kernel I try to build fail to compile, even GENERIC?

There are a number of possible causes for this problem:

- The source tree is different from the one used to build the currently running system. When attempting an upgrade, read /usr/src/UPDATING, paying particular attention to the "COMMON ITEMS" section at the end.
- The make buildkernel did not complete successfully. The make buildkernel target relies on files generated by the make buildworld target to complete its job correctly.
- Even when building FreeBSD-STABLE, it is possible that the source tree was fetched at a time when it was either being modified or it was broken. Only releases are guaranteed to be buildable, although FreeBSD-STABLE builds fine the majority of the time. Try re-fetching the source tree and see if the problem goes away. Try using a different mirror in case the previous one is having problems.

7.4. Which scheduler is in use on a running system?

The name of the scheduler currently being used is directly available as the value of the kern.sched.name sysctl:

```
% sysctl kern.sched.name
kern.sched.name: ULE
```

7.5. What is kern.sched.quantum?

kern.sched.quantum is the maximum number of ticks a process can run without being preempted in the 4BSD scheduler.

Chapter 8. Disks, File Systems, and Boot Loaders

8.1. How can I add my new hard disk to my FreeBSD system?

See the Adding Disks section in the FreeBSD Handbook.

8.2. How do I move my system over to my huge new disk?

The best way is to reinstall the operating system on the new disk, then move the user data over. This is highly recommended when tracking -STABLE for more than one release or when updating a release instead of installing a new one. Install booteasy on both disks with boot0cfg(8) and dual boot until you are happy with the new configuration. Skip the next paragraph to find out how to move the data after doing this.

Alternatively, partition and label the new disk with either sade(8) or gpart(8). If the disks are MBR-formatted, booteasy can be installed on both disks with boot0cfg(8) so that the computer can dual boot to the old or new system after the copying is done.

Once the new disk set up, the data cannot just be copied. Instead, use tools that understand device files and system flags, such as dump(8). Although it is recommended to move the data while in single-user mode, it is not required.

When the disks are formatted with UFS, never use anything but dump(8) and restore(8) to move the root file system. These commands should also be used when moving a single partition to another empty partition. The sequence of steps to use dump to move the data from one UFS partitions to a new partition is:

1. newfs the new partition.
2. mount it on a temporary mount point.
3. cd to that directory.
4. dump the old partition, piping output to the new one.

For example, to move /dev/ada1s1a with /mnt as the temporary mount point, type:

```
# newfs /dev/ada1s1a
# mount /dev/ada1s1a /mnt
# cd /mnt
# dump 0af - / | restore rf -
```

Rearranging partitions with dump takes a bit more work. To merge a partition like /var into its parent, create the new partition large enough for both, move the parent partition as described above, then move the child partition into the empty directory that the first move created:

```
# newfs /dev/ada1s1a
# mount /dev/ada1s1a /mnt
# cd /mnt
# dump 0af - / | restore rf -
```

```
# cd var
# dump 0af - /var | restore rf -
```

To split a directory from its parent, say putting /var on its own partition when it was not before, create both partitions, then mount the child partition on the appropriate directory in the temporary mount point, then move the old single partition:

```
# newfs /dev/ada1s1a
# newfs /dev/ada1s1d
# mount /dev/ada1s1a /mnt
# mkdir /mnt/var
# mount /dev/ada1s1d /mnt/var
# cd /mnt
# dump 0af - / | restore rf -
```

The cpio(1) and pax(1) utilities are also available for moving user data. These are known to lose file flag information, so use them with caution.

8.3. Which partitions can safely use Soft Updates? I have heard that Soft Updates on / can cause problems. What about Journaled Soft Updates?

Short answer: Soft Updates can usually be safely used on all partitions.

Long answer: Soft Updates has two characteristics that may be undesirable on certain partitions. First, a Soft Updates partition has a small chance of losing data during a system crash. The partition will not be corrupted as the data will simply be lost. Second, Soft Updates can cause temporary space shortages.

When using Soft Updates, the kernel can take up to thirty seconds to write changes to the physical disk. When a large file is deleted the file still resides on disk until the kernel actually performs the deletion. This can cause a very simple race condition. Suppose one large file is deleted and another large file is immediately created. The first large file is not yet actually removed from the physical disk, so the disk might not have enough room for the second large file. This will produce an error that the partition does not have enough space, even though a large chunk of space has just been released. A few seconds later, the file creation works as expected.

If a system should crash after the kernel accepts a chunk of data for writing to disk, but before that data is actually written out, data could be lost. This risk is extremely small, but generally manageable.

These issues affect all partitions using Soft Updates. So, what does this mean for the root partition?

Vital information on the root partition changes very rarely. If the system crashed during the thirty-second window after such a change is made, it is possible that data could be lost. This risk is negligible for most applications, but be aware that it exists. If the system cannot tolerate this much risk, do not use Soft Updates on the root file system!

/ is traditionally one of the smallest partitions. If /tmp is on /, there may be intermittent space problems. Symlinking /tmp to /var/tmp will solve this problem.

Finally, dump(8) does not work in live mode (-L) on a filesystem, with Journaled Soft Updates (SU+J).

8.4. Can I mount other foreign file systems under FreeBSD?

FreeBSD supports a variety of other file systems.

UFS
: UFS CD-ROMs can be mounted directly on FreeBSD. Mounting disk partitions from Digital UNIX and other systems that support UFS may be more complex, depending on the details of the disk partitioning for the operating system in question.

ext2/ext3
: FreeBSD supports ext2fs and ext3fs partitions. See ext2fs(5) for more information.

NTFS
: FUSE based NTFS support is available as a port (sysutils/fusefs-ntfs). For more information see ntfs-3g.

FAT
: FreeBSD includes a read-write FAT driver. For more information, see mount_msdosfs(8).

ZFS
: FreeBSD 包含由 Sun™ 移植過來的 ZFS 驅動程式。 目前的建議是僅在記憶體充足的 amd64 平臺上使用它。有關更詳細資訊, 請參閱 zfs(8)。

FreeBSD includes the Network File System NFS and the FreeBSD Ports Collection provides several FUSE applications to support many other file systems.

8.5. How do I mount a secondary DOS partition?

The secondary DOS partitions are found after all the primary partitions. For example, if E is the second DOS partition on the second SCSI drive, there will be a device file for "slice 5" in /dev. To mount it:

```
# mount -t msdosfs /dev/da1s5 /dos/e
```

8.6. Is there a cryptographic file system for FreeBSD?

Yes, gbde(8) and geli(8). See the Encrypting Disk Partitions section of the FreeBSD Handbook.

8.7. How do I boot FreeBSD and Linux using GRUB?

To boot FreeBSD using GRUB, add the following to either /boot/grub/menu.lst or /boot/grub/grub.conf, depending upon which is used by the Linux™ distribution.

```
title FreeBSD 9.1
  root (hd0,a)
  kernel /boot/loader
```

Where hd0,a points to the root partition on the first disk. To specify the slice number, use something like this (hd0,2,a). By default, if the slice number is omitted, GRUB searches the first slice which has the a partition.

8.8. How do I boot FreeBSD and Linux using BootEasy?

Install LILO at the start of the Linux™ boot partition instead of in the Master Boot Record. You can then boot LILO from BootEasy.

This is recommended when running Windows™ and Linux™ as it makes it simpler to get Linux™ booting again if Windows™ is reinstalled.

8.9. How do I change the boot prompt from ??? to something more meaningful?

This cannot be accomplished with the standard boot manager without rewriting it. There are a number of other boot managers in the sysutils category of the Ports Collection.

8.10. How do I use a new removable drive?

If the drive already has a file system on it, use a command like this:

```
# mount -t msdosfs /dev/da0s1 /mnt
```

If the drive will only be used with FreeBSD systems, partition it with UFS or ZFS. This will provide long filename support, improvement in performance, and stability. If the drive will be used by other operating systems, a more portable choice, such as msdosfs, is better.

```
# dd if=/dev/zero of=/dev/da0 count=2
# gpart create -s GPT /dev/da0
# gpart add -t freebsd-ufs /dev/da0
```

Finally, create a new file system:

```
# newfs /dev/da0p1
```

and mount it:

```
# mount /dev/da0s1 /mnt
```

It is a good idea to add a line to /etc/fstab (see fstab(5)) so you can just type mount /mnt in the future:

```
/dev/da0p1 /mnt ufs rw,noauto 0 0
```

8.11. Why do I get Incorrect super block when mounting a CD?

The type of device to mount must be specified. This is described in the Handbook section on Using Data CDs.

8.12. Why do I get Device not configured when mounting a CD?

This generally means that there is no CD in the drive, or the drive is not visible on the bus. Refer to the Using Data CDs section of the Handbook for a detailed discussion of this issue.

8.13. Why do all non-English characters in filenames show up as ? on my CDs when mounted in FreeBSD?

The CD probably uses the "Joliet" extension for storing information about files and directories. This is discussed in the Handbook section on Using Data CD-ROMs.

8.14. A CD burned under FreeBSD cannot be read under any other operating system. Why?

This means a raw file was burned to the CD, rather than creating an ISO 9660 file system. Take a look at the Handbook section on Using Data CDs.

8.15. How can I create an image of a data CD?

This is discussed in the Handbook section on Writing Data to an ISO File System. For more on working with CD-ROMs, see the Creating CDs Section in the Storage chapter in the Handbook.

8.16. Why can I not mount an audio CD?

Trying to mount an audio CD will produce an error like cd9660: /dev/cd0: Invalid argument. This is because mount only works on file systems. Audio CDs do not have file systems; they just have data. Instead, use a program that reads audio CDs, such as the audio/xmcd package or port.

8.17. How do I mount a multi-session CD?

By default, mount(8) will attempt to mount the last data track (session) of a CD. To load an earlier session, use the -s command line argument. Refer to mount_cd9660(8) for specific examples.

8.18. How do I let ordinary users mount CD-ROMs, DVDs, USB drives, and other removable media?

As root set the sysctl variable vfs.usermount to 1.

```
# sysctl vfs.usermount=1
```

To make this persist across reboots, add the line vfs.usermount=1 to /etc/sysctl.conf so that it is reset at system boot time.

Users can only mount devices they have read permissions to. To allow users to mount a device permissions must be set in /etc/devfs.conf.

For example, to allow users to mount the first USB drive add:

```
# Allow all users to mount a USB drive.
	own	/dev/da0	root:operator
```

```
perm    /dev/da0    0666
```

All users can now mount devices they could read onto a directory that they own:

```
% mkdir ~/my-mount-point
% mount -t msdosfs /dev/da0 ~/my-mount-point
```

Unmounting the device is simple:

```
% umount ~/my-mount-point
```

Enabling vfs.usermount, however, has negative security implications. A better way to access MS-DOS™ formatted media is to use the emulators/mtools package in the Ports Collection.

The device name used in the previous examples must be changed according to the configuration.

8.19. The du and df commands show different amounts of disk space available. What is going on?

This is due to how these commands actually work. du goes through the directory tree, measures how large each file is, and presents the totals. df just asks the file system how much space it has left. They seem to be the same thing, but a file without a directory entry will affect df but not du.

When a program is using a file, and the file is deleted, the file is not really removed from the file system until the program stops using it. The file is immediately deleted from the directory listing, however. As an example, consider a file large enough to affect the output of du and df. A file being viewed with more can be deleted wihout causing an error. The entry is removed from the directory so no other program or user can access it. However, du shows that it is gone as it has walked the directory tree and the file is not listed. df shows that it is still there, as the file system knows that more is still using that space. Once the more session ends, du and df will agree.

This situation is common on web servers. Many people set up a FreeBSD web server and forget to rotate the log files. The access log fills up /var. The new administrator deletes the file, but the system still complains that the partition is full. Stopping and restarting the web server program would free the file, allowing the system to release the disk space. To prevent this from happening, set up newsyslog(8).

Note that Soft Updates can delay the freeing of disk space and it can take up to 30 seconds for the change to be visible.

8.20. How can I add more swap space?

This section of the Handbook describes how to do this.

8.21. Why does FreeBSD see my disk as smaller than the manufacturer says it is?

Disk manufacturers calculate gigabytes as a billion bytes each, whereas FreeBSD calculates them as 1,073,741,824 bytes each. This explains why, for example, FreeBSD’s boot messages will report a disk that supposedly has 80 GB as holding 76,319 MB.

Also note that FreeBSD will (by default) reserve 8% of the disk space.

8.22. How is it possible for a partition to be more than 100% full?

A portion of each UFS partition (8%, by default) is reserved for use by the operating system and the root user. df(1) does not count that space when calculating the Capacity column, so it can exceed 100%. Notice that the Blocks column is always greater than the sum of the Used and Avail columns, usually by a factor of 8%.

For more details, look up -m in tunefs(8).

Chapter 9. ZFS

9.1. 使用 ZFS 最少需要多少記憶體？

至少需要 4GB 的記憶體才能跑得順，但不同的工作負載可能會造成相當大的差異。

9.2. ZIL 是什麼而又何時會被使用？

The ZIL (ZFS 動向日誌) 是一個紀錄日誌，用以實現系統當機時 POSIX 寫入保證的語義，多個正常 ZFS 寫入動作會被分成多個交易處理群組，並在交易處理群組被填滿時寫入磁碟 ("Transaction Group Commit")。然而像 fsync(2) 這樣的系統呼叫，會要求該系統呼叫在返回前，能承諾已將資料寫入磁碟，ZIL 就是用來紀錄確認為已執行寫入的資料，但其實尚未存在於磁碟上，即尚未完成交易處理，交易處理群組具有時間戳記，在系統當機後，找到 ZIL 最後一個有效的時間戳記，即將遺失的資料再舍併至磁碟上。

9.3. 我需要用固態硬碟 (SSD) 來存 ZIL 嗎？

ZFS 預設將 ZIL 儲存在包含所有資料的 zpool 中，如果應用程式的寫入負載很重，將 ZIL 儲存在同步速度非常快的獨立設備中，藉由循序寫入效能的提高可以改善整個系統的效能，對於其他類型的工作負載， 固態硬碟就不會有太大的助益。

9.4. L2ARC 是什麼？

The L2ARC (Second Level Adaptive Replacement Cache) 是存於快速儲存設備 SSD 上的讀取快取，此快取在重新開機後會消失，請注意記憶體是第一層的快取，只有在記憶體不足的情況下才需要 L2ARC。

L2ARC 需要 ARC 的空間來為其製作索引，因此，有一種反常的情況，如果有一種工作集 (working set) 可以完美地剛好放入 ARC，一旦系統使用 L2ARC，該工作集的運作將不再完美，因為 ARC 需要用一部分空間來保存 L2ARC 的索引，以至於必須將工作集的一部分存入比記憶體慢的 L2ARC。

9.5. 建議啟用去冗餘 (deduplication) 嗎？

一般而言，不建議這麼做。

去冗餘需要相當多的記憶體，而且會讓讀寫磁碟所需的時間變長，除非磁碟上儲存了非常多重複的資料，例如：虛擬機的映像檔或者是使用者的備份資料，否則開啟去冗餘可能弊大於利。另一個需要考量的狀況是：啟用去冗餘功能之後再將其關閉，無法將磁碟上去冗餘的狀態立即逆轉，必須等到下次修改了之前被去冗餘的資料，變更的區塊才會再被複製一份。

去冗餘也可能會導致某些非預期的情況，特別是刪除檔案時可能會慢很多。

9.6. 在我建立的 ZFS pool 中無法刪除和新增檔案，應該怎麼修復？

這很有可能是該 pool 的空間使用率已達 100% 滿了，因 ZFS 需要儲存空間以將紀錄交易處理的輔助資料 (metadata) 寫入，為了讓該 pool 回復至可用狀態，必須用檔案切除的方法 (truncate 命令) 刪除不重要的檔案：

```
% truncate -s 0 unimportant-file
```

因為檔案切除不需要建立交易處理紀錄，並能釋放出可使用的磁碟區塊。

> 如果系統曾進行過額外的 ZFS dataset 調校，例如：去冗餘，釋放出來的空間也許不會立即可得。

9.7. ZFS 支援固態硬碟 (SSD) 的 TRIM 功能嗎？

自 FreeBSD 10-CURRENT 修定 rr240868 開始，就支援 ZFS TRIM。ZFS TRIM 的支援分別已在 rr252162 和 rr251419 的修訂，加進所有 FreeBSD-STABLE 分支。

ZFS TRIM 預設就已開啟，也可以將其關閉，只要加入一行設定到 /etc/sysctl.conf:

```
vfs.zfs.trim.enabled=0
```

ZFS TRIM 也可能某些設定中會無效，例如：在採用 GELI 裝置上的 ZFS 檔案系統。

Chapter 10. System Administration

10.1. Where are the system start-up configuration files?

The primary configuration file is /etc/defaults/rc.conf which is described in rc.conf(5). System startup scripts such as /etc/rc and /etc/rc.d, which are described in rc(8), include this file. Do not edit this file! Instead, to edit an entry in /etc/defaults/rc.conf, copy the line into /etc/rc.conf and change it there.

For example, if to start named(8), the included DNS server:

```
# echo 'named_enable="YES"' >> /etc/rc.conf
```

To start up local services, place shell scripts in the /usr/local/etc/rc.d directory. These shell scripts should be set executable, the default file mode is 555.

10.2. How do I add a user easily?

Use the adduser(8) command, or the pw(8) command for more complicated situations.

To remove the user, use the rmuser(8) command or, if necessary, pw(8).

10.3. Why do I keep getting messages like root: not found after editing /etc/crontab?

This is normally caused by editing the system crontab. This is not the correct way to do things as the system crontab has a different format to the per-user crontabs. The system crontab has an extra field, specifying which user to run the command as. cron(8) assumes this user is the first word of the command to execute. Since no such command exists, this error message is displayed.

To delete the extra, incorrect crontab:

```
# crontab -r
```

10.4. Why do I get the error, you are not in the correct group to su root when I try to su to root?

This is a security feature. In order to su to root, or any other account with superuser privileges, the user account must be a member of the wheel group. If this feature were not there, anybody with an account on a system who also found out root's password would be able to gain superuser level access to the system.

To allow someone to su to root, put them in the wheel group using pw:

```
# pw groupmod wheel -m lisa
```

The above example will add user lisa to the group wheel.

10.5. I made a mistake in rc.conf, or another startup file, and now I cannot edit it because the file system is read-only. What should I do?

Restart the system using boot -s at the loader prompt to enter single-user mode. When prompted for a shell pathname, press Enter and run mount -urw / to re-mount the root file system in read/write mode. You may also need to run mount -a -t ufs to mount the file system where your favorite editor is defined. If that editor is on a network file system, either configure the network manually before mounting the network file systems, or use an editor which resides on a local file system, such as ed(1).

In order to use a full screen editor such as vi(1) or emacs(1), run export TERM=xterm so that these editors can load the correct data from the termcap(5) database.

After performing these steps, edit /etc/rc.conf to fix the syntax error. The error message displayed immediately after the kernel boot messages should indicate the number of the line in the file which is at fault.

10.6. Why am I having trouble setting up my printer?

See the Handbook entry on printing for troubleshooting tips.

10.7. How can I correct the keyboard mappings for my system?

Refer to the Handbook section on using localization, specifically the section on console setup.

10.8. Why can I not get user quotas to work properly?

1. It is possible that the kernel is not configured to use quotas. In this case, add the following line to the kernel configuration file and recompile the kernel:

   ```
   options QUOTA
   ```

   Refer to theHandbook entry on quotas for full details.

2. Do not turn on quotas on /.
3. Put the quota file on the file system that the quotas are to be enforced on:

File System	Quota file
/usr	/usr/admin/quotas
/home	/home/admin/quotas
...	...

10.9. Does FreeBSD support System V IPC primitives?

Yes, FreeBSD supports System V-style IPC, including shared memory, messages and semaphores, in the GENERIC kernel. With a custom kernel, support may be loaded with the sysvshm.ko, sysvsem.ko and sysvmsg.ko kernel modules, or enabled in the custom kernel by adding the following lines to the kernel configuration file:

```
options   SYSVSHM      # enable shared memory
options   SYSVSEM      # enable for semaphores
options   SYSVMSG      # enable for messaging
```

Recompile and install the kernel.

10.10. What other mail-server software can I use instead of Sendmail?

The Sendmail server is the default mail-server software for FreeBSD, but it can be replaced with another MTA installed from the Ports Collection. Available ports include mail/exim, mail/postfix, and mail/qmail. Search the mailing lists for discussions regarding the advantages and disadvantages of the available MTAs.

10.11. I have forgotten the root password! What do I do?

Do not panic! Restart the system, type boot -s at the Boot: prompt to enter single-user mode. At the question about the shell to use, hit Enter which will display a # prompt. Enter mount -urw / to remount the root file system read/write, then run mount -a to remount all the file systems. Run passwd root to change the root password then run exit(1) to continue booting.

If you are still prompted to give the root password when entering the single-user mode, it means that the console has been marked as insecure in /etc/ttys. In this case, it will be required to boot from a FreeBSD installation disk, choose the Live CD or Shell at the beginning of the install process and issue the commands mentioned above. Mount the specific partition in this case and then chroot to it. For example, replace mount -urw / with mount /dev/ada0p1 /mnt; chroot /mnt for a system on ada0p1.

If the root partition cannot be mounted from single-user mode, it is possible that the partitions are encrypted and it is impossible to mount them without the access keys. For more information see the section about encrypted disks in the FreeBSD Handbook.

10.12. How do I keep ControlAltDelete from rebooting the system?

When using syscons(4), the default console driver, build and install a new kernel with this line in the configuration file:

```
options SC_DISABLE_REBOOT
```

This can also be done by setting the following sysctl(8) which does not require a reboot or kernel recompile:

```
# sysctl hw.syscons.kbd_reboot=0
```

The above two methods are exclusive: The sysctl(8) does not exist if the kernel is compiled with SC_DISABLE_REBOOT.

10.13. How do I reformat DOS text files to UNIX ones?

Use this perl(1) command:

```
% perl -i.bak -npe 's/\r\n/\n/g' file(s)
```

where file(s) is one or more files to process. The modification is done in-place, with the original file stored with a .bak extension.

Alternatively, use tr(1):

```
% tr -d '\r' < dos-text-file > unix-file
```

dos-text-file is the file containing DOS text while unix-file will contain the converted output. This can be quite a bit faster than using perl.

Yet another way to reformat DOS text files is to use the converters/dosunix port from the Ports Collection. Consult its documentation about the details.

10.14. How do I re-read /etc/rc.conf and re-start /etc/rc without a reboot?

Go into single-user mode and then back to multi-user mode:

```
# shutdown now
# return
# exit
```

10.15. I tried to update my system to the latest -STABLE, but got -BETAx, -RC or -PRERELEASE! What is going on?

Short answer: it is just a name. RC stands for "Release Candidate". It signifies that a release is imminent. In FreeBSD, -PRERELEASE is typically synonymous with the code freeze before a release. (For some releases, the -BETA label was used in the same way as -PRERELEASE.)

Long answer: FreeBSD derives its releases from one of two places. Major, dot-zero, releases, such as 9.0-RELEASE are branched from the head of the development stream, commonly referred to as -CURRENT. Minor releases, such as 6.3-RELEASE or 5.2-RELEASE, have been snapshots of the active -STABLE branch. Starting with 4.3-RELEASE, each release also now has its own branch which can be tracked by people requiring an extremely conservative rate of development (typically only security advisories).

When a release is about to be made, the branch from which it will be derived from has to undergo a certain process. Part of this process is a code freeze. When a code freeze is initiated, the name of the branch is changed to reflect that it is about to become a release. For example, if the branch used to be called 6.2-STABLE, its name will be changed to 6.3-PRERELEASE to signify the code freeze and signify that extra pre-release testing should be happening. Bug fixes can still be committed to be part of the release. When the source code is in shape for the release the name will be changed to 6.3-RC to signify that a release is about to be made from it. Once in the RC stage, only the most critical bugs found can be fixed. Once the release (6.3-RELEASE in this example) and release branch have been made, the branch will be renamed to 6.3-STABLE.

For more information on version numbers and the various Subversion branches, refer to the

Release Engineering article.

10.16. I tried to install a new kernel, and the chflags1 failed. How do I get around this?

Short answer: the security level is greater than 0. Reboot directly to single-user mode to install the kernel.

Long answer: FreeBSD disallows changing system flags at security levels greater than 0. To check the current security level:

```
# sysctl kern.securelevel
```

The security level cannot be lowered in multi-user mode, so boot to single-user mode to install the kernel, or change the security level in /etc/rc.conf then reboot. See the init(8) manual page for details on securelevel, and see /etc/defaults/rc.conf and the rc.conf(5) manual page for more information on rc.conf.

10.17. I cannot change the time on my system by more than one second! How do I get around this?

Short answer: the system is at a security level greater than 1. Reboot directly to single-user mode to change the date.

Long answer: FreeBSD disallows changing the time by more that one second at security levels greater than 1. To check the security level:

```
# sysctl kern.securelevel
```

The security level cannot be lowered in multi-user mode. Either boot to single-user mode to change the date or change the security level in /etc/rc.conf and reboot. See the init(8) manual page for details on securelevel, and see /etc/defaults/rc.conf and the rc.conf(5) manual page for more information on rc.conf.

10.18. Why is rpc.statd using 256 MB of memory?

No, there is no memory leak, and it is not using 256 MB of memory. For convenience, rpc.statd maps an obscene amount of memory into its address space. There is nothing terribly wrong with this from a technical standpoint; it just throws off things like top(1) and ps(1).

rpc.statd(8) maps its status file (resident on /var) into its address space; to save worrying about remapping the status file later when it needs to grow, it maps the status file with a generous size. This is very evident from the source code, where one can see that the length argument to mmap(2) is 0x10000000, or one sixteenth of the address space on an IA32, or exactly 256 MB.

10.19. Why can I not unset the schg file flag?

The system is running at securelevel greater than 0. Lower the securelevel and try again. For more information, see the FAQ entry on securelevel and the init(8) manual page.

10.20. What is vnlru?

vnlru flushes and frees vnodes when the system hits the kern.maxvnodes limit. This kernel thread sits mostly idle, and only activates when there is a huge amount of RAM and users are accessing

tens of thousands of tiny files.

10.21. What do the various memory states displayed by top mean?

- Active: pages recently statistically used.
- Inactive: pages recently statistically unused.
- Cache: (most often) pages that have percolated from inactive to a status where they maintain their data, but can often be immediately reused (either with their old association, or reused with a new association). There can be certain immediate transitions from active to cache state if the page is known to be clean (unmodified), but that transition is a matter of policy, depending upon the algorithm choice of the VM system maintainer.
- Free: pages without data content, and can be immediately used in certain circumstances where cache pages might be ineligible. Free pages can be reused at interrupt or process state.
- Wired: pages that are fixed into memory, usually for kernel purposes, but also sometimes for special use in processes.

Pages are most often written to disk (sort of a VM sync) when they are in the inactive state, but active pages can also be synced. This depends upon the CPU tracking of the modified bit being available, and in certain situations there can be an advantage for a block of VM pages to be synced, whether they are active or inactive. In most common cases, it is best to think of the inactive queue to be a queue of relatively unused pages that might or might not be in the process of being written to disk. Cached pages are already synced, not mapped, but available for immediate process use with their old association or with a new association. Free pages are available at interrupt level, but cached or free pages can be used at process state for reuse. Cache pages are not adequately locked to be available at interrupt level.

There are some other flags (e.g., busy flag or busy count) that might modify some of the described rules.

10.22. How much free memory is available?

There are a couple of kinds of "free memory". One kind is the amount of memory immediately available without paging anything else out. That is approximately the size of cache queue + size of free queue (with a derating factor, depending upon system tuning). Another kind of "free memory" is the total amount of VM space. That can be complex, but is dependent upon the amount of swap space and memory. Other kinds of "free memory" descriptions are also possible, but it is relatively useless to define these, but rather it is important to make sure that the paging rate is kept low, and to avoid running out of swap space.

10.23. What is /var/empty?

/var/empty is a directory that the sshd(8) program uses when performing privilege separation. The /var/empty directory is empty, owned by root and has the schg flag set. This directory should not be deleted.

10.24. I just changed /etc/newsyslog.conf. How can I check if it does what I expect?

To see what newsyslog(8) will do, use the following:

```
% newsyslog -nrvv
```

10.25. My time is wrong, how can I change the timezone?

Use tzsetup(8).

Chapter 11. The X Window System and Virtual Consoles

11.1. What is the X Window System?

The X Window System (commonly X11) is the most widely available windowing system capable of running on UNIX™ or UNIX™ like systems, including FreeBSD. The X.Org Foundation administers the X protocol standards, with the current reference implementation, version 11 release 7.7, so references are often shortened to X11.

Many implementations are available for different architectures and operating systems. An implementation of the server-side code is properly known as an X server.

11.2. I want to run Xorg, how do I go about it?

To install Xorg do one of the following:

Use the x11/xorg meta-port, which builds and installs every Xorg component.

Use x11/xorg-minimal, which builds and installs only the necessary Xorg components.

Install Xorg from FreeBSD packages:

```
# pkg install xorg
```

After the installation of Xorg, follow the instructions from the X11 Configuration section of the FreeBSD Handbook.

11.3. I tried to run X, but I get a No devices detected. error when I type startx. What do I do now?

The system is probably running at a raised securelevel. It is not possible to start X at a raised securelevel because X requires write access to io(4). For more information, see at the init(8) manual page.

There are two solutions to the problem: set the securelevel back down to zero or run xdm(1) (or an alternative display manager) at boot time before the securelevel is raised.

See How do I start XDM on boot? for more information about running xdm(1) at boot time.

11.4. Why does my mouse not work with X?

When using syscons(4), the default console driver, FreeBSD can be configured to support a mouse pointer on each virtual screen. To avoid conflicting with X, syscons(4) supports a virtual device called /dev/sysmouse. All mouse events received from the real mouse device are written to the sysmouse(4) device via moused(8). To use the mouse on one or more virtual consoles, and use X, see Is it possible to use a mouse outside the X Window system? and set up moused(8).

Then edit /etc/X11/xorg.conf and make sure the following lines exist:

```
Section "InputDevice"
  Option      "Protocol" "SysMouse"
  Option      "Device" "/dev/sysmouse"
```

```
.....
```

Starting with Xorg version 7.4, the InputDevice sections in xorg.conf are ignored in favor of autodetected devices. To restore the old behavior, add the following line to the ServerLayout or ServerFlags section:

```
Option "AutoAddDevices" "false"
```

Some people prefer to use /dev/mouse under X. To make this work, /dev/mouse should be linked to /dev/sysmouse (see sysmouse(4)) by adding the following line to /etc/devfs.conf (see devfs.conf(5)):

```
link   sysmouse   mouse
```

This link can be created by restarting devfs(5) with the following command (as root):

```
# service devfs restart
```

11.5. My mouse has a fancy wheel. Can I use it in X?

Yes, if X is configured for a 5 button mouse. To do this, add the lines Buttons 5 and ZAxisMapping 4 5 to the "InputDevice" section of /etc/X11/xorg.conf, as seen in this example:

```
Section "InputDevice"
  Identifier    "Mouse1"
  Driver       "mouse"
  Option       "Protocol" "auto"
  Option       "Device" "/dev/sysmouse"
  Option       "Buttons" "5"
  Option       "ZAxisMapping" "4 5"
EndSection
```

The mouse can be enabled in Emacsby adding these lines to ~/.emacs:

```
;; wheel mouse
(global-set-key [mouse-4] 'scroll-down)
(global-set-key [mouse-5] 'scroll-up)
```

11.6. My laptop has a Synaptics touchpad. Can I use it in X?

Yes, after configuring a few things to make it work.

In order to use the Xorg synaptics driver, first remove moused_enable from rc.conf.

To enable synaptics, add the following line to /boot/loader.conf:

```
hw.psm.synaptics_support="1"
```

Add the following to /etc/X11/xorg.conf:

```
Section "InputDevice"
Identifier  "Touchpad0"
Driver      "synaptics"
Option      "Protocol" "psm"
Option      "Device" "/dev/psm0"
EndSection
```

And be sure to add the following into the "ServerLayout" section:

```
InputDevice   "Touchpad0" "SendCoreEvents"
```

11.7. How do I use remote X displays?

For security reasons, the default setting is to not allow a machine to remotely open a window.

To enable this feature, start X with the optional -listen_tcp argument:

```
% startx -listen_tcp
```

11.8. What is a virtual console and how do I make more?

Virtual consoles provide several simultaneous sessions on the same machine without doing anything complicated like setting up a network or running X.

When the system starts, it will display a login prompt on the monitor after displaying all the boot messages. Type in your login name and password to start working on the first virtual console.

To start another session, perhaps to look at documentation for a program or to read mail while waiting for an FTP transfer to finish, hold down Alt and press F2. This will display the login prompt for the second virtual console. To go back to the original session, press Alt + F1.

The default FreeBSD installation has eight virtual consoles enabled. Alt + F1, Alt + F2, Alt + F3, and so on will switch between these virtual consoles.

To enable more of virtual consoles, edit /etc/ttys (see ttys(5)) and add entries for ttyv8 to ttyvc, after the comment on "Virtual terminals":

```
# Edit the existing entry for ttyv8 in /etc/ttys and change
# "off" to "on".
ttyv8   "/usr/libexec/getty Pc"       xterm  on secure
ttyv9   "/usr/libexec/getty Pc"       xterm  on secure
ttyva   "/usr/libexec/getty Pc"       xterm  on secure
```

```
ttyvb  "/usr/libexec/getty Pc"      xterm  on secure
```

The more virtual terminals, the more resources that are used. This can be problematic on systems with 8 MB RAM or less. Consider changing secure to insecure.

> In order to run an X server, at least one virtual terminal must be left to off for it to use. This means that only eleven of the Alt-function keys can be used as virtual consoles so that one is left for the X server.

For example, to run X and eleven virtual consoles, the setting for virtual terminal 12 should be:

```
ttyvb  "/usr/libexec/getty Pc"      xterm  off secure
```

The easiest way to activate the virtual consoles is to reboot.

11.9. How do I access the virtual consoles from X?

Use Ctrl + Alt + Fn to switch back to a virtual console. Press Ctrl + Alt + F1 to return to the first virtual console.

Once at a text console, use Alt + Fn to move between them.

To return to the X session, switch to the virtual console running X. If X was started from the command line using startx, the X session will attach to the next unused virtual console, not the text console from which it was invoked. For eight active virtual terminals, X will run on the ninth, so use Alt + F9.

11.10. How do I start XDM on boot?

There are two schools of thought on how to start xdm(1). One school starts xdm from /etc/ttys (see ttys(5)) using the supplied example, while the other runs xdm from rc.local (see rc(8)) or from an X script in /usr/local/etc/rc.d. Both are equally valid, and one may work in situations where the other does not. In both cases the result is the same: X will pop up a graphical login prompt.

The ttys(5) method has the advantage of documenting which vty X will start on and passing the responsibility of restarting the X server on logout to init(8). The rc(8) method makes it easy to kill xdm if there is a problem starting the X server.

If loaded from rc(8), xdm should be started without any arguments. xdm must start after getty(8) runs, or else getty and xdm will conflict, locking out the console. The best way around this is to have the script sleep 10 seconds or so then launch xdm.

When starting xdm from /etc/ttys, there still is a chance of conflict between xdm and getty(8). One way to avoid this is to add the vt number in /usr/local/lib/X11/xdm/Xservers:

```
:0 local /usr/local/bin/X vt4
```

The above example will direct the X server to run in /dev/ttyv3. Note the number is offset by one. The X server counts the vty from one, whereas the FreeBSD kernel numbers the vty from zero.

11.11. Why do I get Couldn’t open console when I run xconsole?

When X is started with startx, the permissions on /dev/console will not get changed, resulting in things like xterm -C and xconsole not working.

This is because of the way console permissions are set by default. On a multi-user system, one does not necessarily want just any user to be able to write on the system console. For users who are logging directly onto a machine with a VTY, the fbtab(5) file exists to solve such problems.

In a nutshell, make sure an uncommented line of the form is in /etc/fbtab (see fbtab(5)):

```
/dev/ttyv0 0600 /dev/console
```

It will ensure that whomever logs in on /dev/ttyv0 will own the console.

11.12. Why does my PS/2 mouse misbehave under X?

The mouse and the mouse driver may have become out of synchronization. In rare cases, the driver may also erroneously report synchronization errors:

```
psmintr: out of sync (xxxx != yyyy)
```

If this happens, disable the synchronization check code by setting the driver flags for the PS/2 mouse driver to 0x100. This can be easiest achieved by adding hint.psm.0.flags="0x100" to /boot/loader.conf and rebooting.

11.13. How do I reverse the mouse buttons?

Type xmodmap -e "pointer = 3 2 1". Add this command to ~/.xinitrc or ~/.xsession to make it happen automatically.

11.14. How do I install a splash screen and where do I find them?

The detailed answer for this question can be found in the Boot Time Splash Screens section of the FreeBSD Handbook.

11.15. Can I use the Windows keys on my keyboard in X?

Yes. Use xmodmap(1) to define which functions the keys should perform.

Assuming all Windows keyboards are standard, the keycodes for these three keys are the following:

- 115 — `Windows` key, between the left-hand `Ctrl` and `Alt` keys
- 116 — `Windows` key, to the right of `AltGr`
- 117 — `Menu`, to the left of the right-hand `Ctrl`

To have the left `Windows` key print a comma, try this.

```
# xmodmap -e "keycode 115 = comma"
```

To have the `Windows` key-mappings enabled automatically every time X is started, either put the xmodmap commands in ~/.xinitrc or, preferably, create a ~/.xmodmaprc and include the xmodmap options, one per line, then add the following line to ~/.xinitrc:

```
xmodmap $HOME/.xmodmaprc
```

For example, to map the 3 keys to be `F13`, `F14`, and `F15`, respectively. This would make it easy to map them to useful functions within applications or the window manager.

To do this, put the following in ~/.xmodmaprc.

```
keycode 115 = F13
keycode 116 = F14
keycode 117 = F15
```

For the x11-wm/fvwm2 desktop manager, one could map the keys so that `F13` iconifies or de-iconifies the window the cursor is in, `F14` brings the window the cursor is in to the front or, if it is already at the front, pushes it to the back, and `F15` pops up the main Workplace menu even if the cursor is not on the desktop, which is useful when no part of the desktop is visible.

The following entries in ~/.fvwmrc implement the aforementioned setup:

```
Key F13     FTIWS   A     Iconify
Key F14     FTIWS   A     RaiseLower
Key F15     A     A     Menu Workplace Nop
```

11.16. How can I get 3D hardware acceleration for OpenGL?

The availability of 3D acceleration depends on the version of Xorg and the type of video chip. For an nVidia chip, use the binary drivers provided for FreeBSD by installing one of the following ports:

The latest versions of nVidia cards are supported by the x11/nvidia-driver port.

Older drivers are available as x11/nvidia-driver-#

nVidia provides detailed information on which card is supported by which driver on their web site: http://www.nvidia.com/object/IO_32667.html.

For Matrox G200/G400, check the x11-drivers/xf86-video-mga port.

For ATI Rage 128 and Radeon see ati(4), r128(4) and radeon(4).

Chapter 12. Networking

12.1. Where can I get information on diskless booting?

"Diskless booting" means that the FreeBSD box is booted over a network, and reads the necessary files from a server instead of its hard disk. For full details, see the Handbook entry on diskless booting.

12.2. Can a FreeBSD box be used as a dedicated network router?

Yes. Refer to the Handbook entry on advanced networking, specifically the section on routing and gateways.

12.3. Can I connect my Windows box to the Internet via FreeBSD?

Typically, people who ask this question have two PCs at home, one with FreeBSD and one with some version of Windows™ the idea is to use the FreeBSD box to connect to the Internet and then be able to access the Internet from the Windows™ box through the FreeBSD box. This is really just a special case of the previous question and works perfectly well.

Dialup users must use -nat and set gateway_enable to YES in /etc/rc.conf. For more information, refer to ppp(8) or the Handbook entry on user PPP.

If the connection to the Internet is over Ethernet, use natd(8). A tutorial can be found in the natd section of the Handbook.

12.4. Does FreeBSD support PPP?

Yes. ppp(8) provides support for both incoming and outgoing connections.

For more information on how to use this, refer to the Handbook chapter on PPP.

12.5. Does FreeBSD support NAT or Masquerading?

Yes. For instructions on how to use NAT over a PPP connection, see the Handbook entry on PPP. To use NAT over some other sort of network connection, look at the natd section of the Handbook.

12.6. How can I set up Ethernet aliases?

If the alias is on the same subnet as an address already configured on the interface, add netmask 0xffffffff to this command:

```
# ifconfig ed0 alias 192.0.2.2 netmask 0xffffffff
```

Otherwise, specify the network address and netmask as usual:

```
# ifconfig ed0 alias 172.16.141.5 netmask 0xffffff00
```

More information can be found in the FreeBSD Handbook.

12.7. Why can I not NFS-mount from a Linux box?

Some versions of the Linux™ NFS code only accept mount requests from a privileged port; try to issue the following command:

```
# mount -o -P linuxbox:/blah /mnt
```

12.8. Why does mountd keep telling me it can’t change attributes and that I have a bad exports list on my FreeBSD NFS server?

The most frequent problem is not understanding the correct format of /etc/exports. Review exports(5) and the NFS entry in the Handbook, especially the section on configuring NFS.

12.9. How do I enable IP multicast support?

Install the net/mrouted package or port and add mrouted_enable="YES" to /etc/rc.conf start this service at boot time.

12.10. Why do I have to use the FQDN for hosts on my site?

See the answer in the FreeBSD Handbook.

12.11. Why do I get an error, Permission denied, for all networking operations?

If the kernel is compiled with the IPFIREWALL option, be aware that the default policy is to deny all packets that are not explicitly allowed.

If the firewall is unintentionally misconfigured, restore network operability by typing the following as root:

```
# ipfw add 65534 allow all from any to any
```

Consider setting firewall_type="open" in /etc/rc.conf.

For further information on configuring this firewall, see the Handbook chapter.

12.12. Why is my ipfw fwd rule to redirect a service to another machine not working?

Possibly because network address translation (NAT) is needed instead of just forwarding packets. A "fwd" rule only forwards packets, it does not actually change the data inside the packet. Consider this rule:

```
01000 fwd 10.0.0.1 from any to foo 21
```

When a packet with a destination address of foo arrives at the machine with this rule, the packet is forwarded to 10.0.0.1, but it still has the destination address of foo. The destination address of the packet is not changed to 10.0.0.1. Most machines would probably drop a packet that they receive

with a destination address that is not their own. Therefore, using a "fwd" rule does not often work the way the user expects. This behavior is a feature and not a bug.

See the FAQ about redirecting services, the natd(8) manual, or one of the several port redirecting utilities in the Ports Collection for a correct way to do this.

12.13. How can I redirect service requests from one machine to another?

FTP and other service requests can be redirected with the sysutils/socket package or port. Replace the entry for the service in /etc/inetd.conf to call socket, as seen in this example for ftpd:

```
ftp stream tcp nowait nobody /usr/local/bin/socket socket ftp.example.com ftp
```

where ftp.example.com and ftp are the host and port to redirect to, respectively.

12.14. Where can I get a bandwidth management tool?

There are three bandwidth management tools available for FreeBSD. dummynet(4) is integrated into FreeBSD as part of ipfw(4). ALTQ has been integrated into FreeBSD as part of pf(4). Bandwidth Manager from Emerging Technologies is a commercial product.

12.15. Why do I get /dev/bpf0: device not configured?

The running application requires the Berkeley Packet Filter (bpf(4)), but it was removed from a custom kernel. Add this to the kernel config file and build a new kernel:

```
device bpf      # Berkeley Packet Filter
```

12.16. How do I mount a disk from a Windows machine that is on my network, like smbmount in Linux?

Use the SMBFS toolset. It includes a set of kernel modifications and a set of userland programs. The programs and information are available as mount_smbfs(8) in the base system.

12.17. What are these messages about: Limiting icmp/open port/closed port response in my log files?

This kernel message indicates that some activity is provoking it to send a large amount of ICMP or TCP reset (RST) responses. ICMP responses are often generated as a result of attempted connections to unused UDP ports. TCP resets are generated as a result of attempted connections to unopened TCP ports. Among others, these are the kinds of activities which may cause these messages:

- Brute-force denial of service (DoS) attacks (as opposed to single-packet attacks which exploit a specific vulnerability).
- Port scans which attempt to connect to a large number of ports (as opposed to only trying a few well-known ports).

The first number in the message indicates how many packets the kernel would have sent if the limit was not in place, and the second indicates the limit. This limit is controlled using net.inet.icmp.icmplim. This example sets the limit to 300 packets per second:

```
# sysctl net.inet.icmp.icmplim=300
```

To disable these messages without disabling response limiting, use net.inet.icmp.icmplim_output to disable the output:

```
# sysctl net.inet.icmp.icmplim_output=0
```

Finally, to disable response limiting completely, set net.inet.icmp.icmplim to 0. Disabling response limiting is discouraged for the reasons listed above.

12.18. What are these arp: unknown hardware address format error messages?

This means that some device on the local Ethernet is using a MAC address in a format that FreeBSD does not recognize. This is probably caused by someone experimenting with an Ethernet card somewhere else on the network. This is most commonly seen on cable modem networks. It is harmless, and should not affect the performance of the FreeBSD system.

12.19. Why do I keep seeing messages like: 192.168.0.10 is on fxp1 but got reply from 00:15:17:67:cf:82 on rl0, and how do I disable it?

Because a packet is coming from outside the network unexpectedly. To disable them, set net.link.ether.inet.log_arp_wrong_iface to 0.

12.20. How do I compile an IPv6 only kernel?

Configure your kernel with these settings:

```
include GENERIC
ident GENERIC-IPV6ONLY
makeoptions MKMODULESENV+="WITHOUT_INET_SUPPORT="
nooptions INET
nodevice gre
```

Chapter 13. Security

13.1. What is a sandbox?

"Sandbox" is a security term. It can mean two things:

- A process which is placed inside a set of virtual walls that are designed to prevent someone who breaks into the process from being able to break into the wider system.

  The process is only able to run inside the walls. Since nothing the process does in regards to executing code is supposed to be able to breach the walls, a detailed audit of its code is not needed in order to be able to say certain things about its security.

  The walls might be a user ID, for example. This is the definition used in the security(7) and named(8) man pages.

  Take the ntalk service, for example (see inetd(8)). This service used to run as user ID root. Now it runs as user ID tty. The tty user is a sandbox designed to make it more difficult for someone who has successfully hacked into the system via ntalk from being able to hack beyond that user ID.

- A process which is placed inside a simulation of the machine. It means that someone who is able to break into the process may believe that he can break into the wider machine but is, in fact, only breaking into a simulation of that machine and not modifying any real data.

  The most common way to accomplish this is to build a simulated environment in a subdirectory and then run the processes in that directory chrooted so that / for that process is this directory, not the real / of the system).

  Another common use is to mount an underlying file system read-only and then create a file system layer on top of it that gives a process a seemingly writeable view into that file system. The process may believe it is able to write to those files, but only the process sees the effects — other processes in the system do not, necessarily.

  An attempt is made to make this sort of sandbox so transparent that the user (or hacker) does not realize that he is sitting in it.

UNIX™ implements two core sandboxes. One is at the process level, and one is at the userid level.

Every UNIX™ process is completely firewalled off from every other UNIX™ process. One process cannot modify the address space of another.

A UNIX™ process is owned by a particular userid. If the user ID is not the root user, it serves to firewall the process off from processes owned by other users. The user ID is also used to firewall off on-disk data.

13.2. What is securelevel?

securelevel is a security mechanism implemented in the kernel. When the securelevel is positive, the kernel restricts certain tasks; not even the superuser (root) is allowed to do them. The securelevel mechanism limits the ability to:

- Unset certain file flags, such as schg (the system immutable flag).
- Write to kernel memory via /dev/mem and /dev/kmem.
- Load kernel modules.
- Alter firewall rules.

To check the status of the securelevel on a running system:

```
# sysctl -n kern.securelevel
```

The output contains the current value of the securelevel. If it is greater than 0, at least some of the securelevel’ s protections are enabled.

The securelevel of a running system cannot be lowered as this would defeat its purpose. If a task requires that the securelevel be non-positive, change the kern_securelevel and kern_securelevel_enable variables in /etc/rc.conf and reboot.

For more information on securelevel and the specific things all the levels do, consult init(8).

> Securelevel is not a silver bullet; it has many known deficiencies. More often than not, it provides a false sense of security.
>
> One of its biggest problems is that in order for it to be at all effective, all files used in the boot process up until the securelevel is set must be protected. If an attacker can get the system to execute their code prior to the securelevel being set (which happens quite late in the boot process since some things the system must do at start-up cannot be done at an elevated securelevel), its protections are invalidated. While this task of protecting all files used in the boot process is not technically impossible, if it is achieved, system maintenance will become a nightmare since one would have to take the system down, at least to single-user mode, to modify a configuration file.
>
> This point and others are often discussed on the mailing lists, particularly the FreeBSD security mailing list. Search the archives here for an extensive discussion. A more fine-grained mechanism is preferred.

13.3. BIND9 (named) is listening on some high-numbered ports. What is going on?

BIND uses a random high-numbered port for outgoing queries. Recent versions of it choose a new, random UDP port for each query. This may cause problems for some network configurations, especially if a firewall blocks incoming UDP packets on particular ports. To get past that firewall, try the avoid-v4-udp-ports and avoid-v6-udp-ports options to avoid selecting random port numbers within a blocked range.

> If a port number (like 53) is specified via the query-source or query-source-v6 options in /usr/local/etc/namedb/named.conf, randomized port selection will not be used. It is strongly recommended that these options not be used to specify fixed port numbers.

Congratulations, by the way. It is good practice to read sockstat(1) output and notice odd things!

13.4. The Sendmail daemon is listening on port 587 as well as the standard port 25! What is going on?

Recent versions of Sendmail support a mail submission feature that runs over port 587. This is not yet widely supported, but is growing in popularity.

13.5. What is this UID 0 toor account? Have I been compromised?

Do not worry. toor is an "alternative" superuser account, where toor is root spelled backwards. It is intended to be used with a non-standard shell so the default shell for root does not need to change.

This is important as shells which are not part of the base distribution, but are instead installed from ports or packages, are installed in /usr/local/bin which, by default, resides on a different file system. If root's shell is located in /usr/local/bin and the file system containing /usr/local/bin) is not mounted, root will not be able to log in to fix a problem and will have to reboot into single-user mode in order to enter the path to a shell.

Some people use toor for day-to-day root tasks with a non-standard shell, leaving root, with a standard shell, for single-user mode or emergencies. By default, a user cannot log in using toor as it does not have a password, so log in as root and set a password for toor before using it to login.

Chapter 14. PPP

14.1. I cannot make ppp8 work. What am I doing wrong?

First, read ppp(8) and the PPP section of the Handbook. To assist in troubleshooting, enable logging with the following command:

```
set log Phase Chat Connect Carrier lcp ipcp ccp command
```

This command may be typed at the ppp(8) command prompt or it may be entered at the start of the default section in /etc/ppp/ppp.conf. Make sure that /etc/syslog.conf contains the lines below and the file /var/log/ppp.log exists:

```
!ppp
*.*     /var/log/ppp.log
```

A lot about what is going can be learned from the log file. Do not worry if it does not all make sense as it may make sense to someone else.

14.2. Why does ppp8 hang when I run it?

This is usually because the hostname will not resolve. The best way to fix this is to make sure that /etc/hosts is read first by the by ensuring that the hosts line is listed first in /etc/host.conf. Then, put an entry in /etc/hosts for the local machine. If there is no local network, change the localhost line:

```
127.0.0.1     foo.example.com foo localhost
```

Otherwise, add another entry for the host. Consult the relevant manual pages for more details.

When finished, verify that this command is successful: ping -c1 hostname.

14.3. Why will ppp8 not dial in -auto mode?

First, check that a default route exists. This command should display two entries:

```
Destination     Gateway       Flags    Refs    Use    Netif Expire
default         10.0.0.2      UGSc      0       0     tun0
10.0.0.2        10.0.0.1      UH        0       0     tun0
```

If a default route is not listed, make sure that the HISADDR line has been added to /etc/ppp/ppp.conf.

Another reason for the default route line being missing is that a default route has been added to /etc/rc.conf and this line is missing from /etc/ppp/ppp.conf:

```
delete ALL
```

If this is the case, go back to the Final System Configuration section of the Handbook.

14.4. What does No route to host mean?

This error is usually because the following section is missing in /etc/ppp/ppp.linkup:

```
MYADDR:
 delete ALL
 add 0 0 HISADDR
```

This is only necessary for a dynamic IP address or when the address of the default gateway is unknown. When using interactive mode, the following can be typed in after entering packet mode. Packet mode is indicated by the capitalized PPP in the prompt:

```
delete ALL
add 0 0 HISADDR
```

Refer to the PPP and Dynamic IP addresses section of the Handbook for further details.

14.5. Why does my connection drop after about 3 minutes?

The default PPP timeout is 3 minutes. This can be adjusted with the following line:

```
set timeout NNN
```

where NNN is the number of seconds of inactivity before the connection is closed. If NNN is zero, the connection is never closed due to a timeout. It is possible to put this command in ppp.conf, or to type it at the prompt in interactive mode. It is also possible to adjust it on the fly while the line is active by connecting to ppp' s server socket using telnet(1) or pppctl(8). Refer to the ppp(8) man page for further details.

14.6. Why does my connection drop under heavy load?

If Link Quality Reporting (LQR) is configured, it is possible that too many LQR packets are lost between the FreeBSD system and the peer. ppp(8) deduces that the line must therefore be bad, and disconnects. LQR is disabled by default and can be enabled with the following line:

```
enable lqr
```

14.7. Why does my connection drop after a random amount of time?

Sometimes, on a noisy phone line or even on a line with call waiting enabled, the modem may hang up because it incorrectly thinks that it lost carrier.

There is a setting on most modems for determining how tolerant it should be to temporary losses of carrier. Refer to the modem manual for details.

14.8. Why does my connection hang after a random amount of time?

Many people experience hung connections with no apparent explanation. The first thing to establish is which side of the link is hung.

When using an external modem, try using ping(8) to see if the TD light is flashing when data is transmitted. If it flashes but the RD light does not, the problem is with the remote end. If TD does not flash, the problem is local. With an internal modem, use the set server command in ppp.conf. When the hang occurs, connect to ppp(8) using pppctl(8). If the network connection suddenly revives due to the activity on the diagnostic socket, or if it will not connect but the set socket command succeeded at startup time, the problem is local. If it can connect but things are still hung, enable local logging with set log local async and use ping(8) from another window or terminal to make use of the link. The async logging will show the data being transmitted and received on the link. If data is going out and not coming back, the problem is remote.

Having established whether the problem is local or remote, there are now two possibilities:

- If the problem is remote, read on entry The remote end is not responding. What can I do?.
- If the problem is local, read on entry ppp(8) has hung. What can I do?.

14.9. The remote end is not responding. What can I do?

There is very little that can be done about this. Many ISPs will refuse to help users not running a Microsoft™ OS. Add enable lqr to /etc/ppp/ppp.conf, allowing ppp(8) to detect the remote failure and hang up. This detection is relatively slow and therefore not that useful.

First, try disabling all local compression by adding the following to the configuration:

```
disable pred1 deflate deflate24 protocomp acfcomp shortseq vj
deny pred1 deflate deflate24 protocomp acfcomp shortseq vj
```

Then reconnect to ensure that this makes no difference. If things improve or if the problem is solved completely, determine which setting makes the difference through trial and error. This is good information for the ISP, although it may make it apparent that it is not a Microsoft™ system.

Before contacting the ISP, enable async logging locally and wait until the connection hangs again. This may use up quite a bit of disk space. The last data read from the port may be of interest. It is usually ASCII data, and may even describe the problem (Memory fault, Core dumped).

If the ISP is helpful, they should be able to enable logging on their end, then when the next link drop occurs, they may be able to tell why their side is having a problem.

14.10. ppp(8) has hung. What can I do?

In this case, rebuild ppp(8) with debugging information, and then use gdb(1) to grab a stack trace from the ppp process that is stuck. To rebuild the ppp utility with debugging information, type:

```
# cd /usr/src/usr.sbin/ppp
# env DEBUG_FLAGS='-g' make clean
# env DEBUG_FLAGS='-g' make install
```

Then, restart ppp and wait until it hangs again. When the debug build of ppp hangs, start gdb on the stuck process by typing:

```
# gdb ppp `pgrep ppp`
```

At the gdb prompt, use the bt or where commands to get a stack trace. Save the output of the gdb session, and "detach" from the running process by typing quit.

14.11. I keep seeing errors about magic being the same. What does it mean?

Occasionally, just after connecting, there may be messages in the log that say Magic is same. Sometimes, these messages are harmless, and sometimes one side or the other exits. Most PPP implementations cannot survive this problem, and even if the link seems to come up, there will be repeated configure requests and configure acknowledgments in the log file until ppp(8) eventually gives up and closes the connection.

This normally happens on server machines with slow disks that are spawning a getty(8) on the port, and executing ppp(8) from a login script or program after login. There were reports of it happening consistently when using slirp. The reason is that in the time taken between getty(8) exiting and ppp(8) starting, the client-side ppp(8) starts sending Line Control Protocol (LCP) packets. Because ECHO is still switched on for the port on the server, the client ppp(8) sees these packets "reflect" back.

One part of the LCP negotiation is to establish a magic number for each side of the link so that "reflections" can be detected. The protocol says that when the peer tries to negotiate the same magic number, a NAK should be sent and a new magic number should be chosen. During the period that the server port has ECHO turned on, the client ppp(8) sends LCP packets, sees the same magic in the reflected packet and NAKs it. It also sees the NAK reflect (which also means ppp(8) must change its magic). This produces a potentially enormous number of magic number changes, all of which are happily piling into the server' s tty buffer. As soon as ppp(8) starts on the server, it is flooded with magic number changes and almost immediately decides it has tried enough to negotiate LCP and gives up. Meanwhile, the client, who no longer sees the reflections, becomes happy just in time to see a hangup from the server.

This can be avoided by allowing the peer to start negotiating with the following line in ppp.conf:

```
set openmode passive
```

This tells ppp(8) to wait for the server to initiate LCP negotiations. Some servers however may never initiate negotiations. In this case, try something like:

```
set openmode active 3
```

This tells ppp(8) to be passive for 3 seconds, and then to start sending LCP requests. If the peer starts sending requests during this period, ppp(8) will immediately respond rather than waiting for the full 3 second period.

14.12. LCP negotiations continue until the connection is closed. What is wrong?

There is currently an implementation mis-feature in ppp(8) where it does not associate LCP, CCP & IPCP responses with their original requests. As a result, if one PPP implementation is more than 6 seconds slower than the other side, the other side will send two additional LCP configuration requests. This is fatal.

Consider two implementations, A and B. A starts sending LCP requests immediately after

connecting and B takes 7 seconds to start. When B starts, A has sent 3 LCP REQs. We are assuming the line has ECHO switched off, otherwise we would see magic number problems as described in the previous section. B sends a REQ, then an ACK to the first of A' s REQs. This results in `A entering the OPENED state and sending and ACK (the first) back to B. In the meantime, B sends back two more ACKs in response to the two additional REQs sent by A before B started up. B then receives the first ACK from A and enters the OPENED state. A receives the second ACK from B and goes back to the REQ-SENT state, sending another (forth) REQ as per the RFC. It then receives the third ACK and enters the OPENED state. In the meantime, B receives the forth REQ from A, resulting in it reverting to the ACK-SENT state and sending another (second) REQ and (forth) ACK as per the RFC. A gets the REQ, goes into REQ-SENT and sends another REQ. It immediately receives the following ACK and enters OPENED.

This goes on until one side figures out that they are getting nowhere and gives up.

The best way to avoid this is to configure one side to be passive — that is, make one side wait for the other to start negotiating. This can be done with the following command:

```
set openmode passive
```

Care should be taken with this option. This command can also be used to limit the amount of time that ppp(8) waits for the peer to begin negotiations:

```
set stopped N
```

Alternatively, the following command (where N is the number of seconds to wait before starting negotiations) can be used:

```
set openmode active N
```

Check the manual page for details.

14.13. Why does ppp8 lock up when I shell out to test it?

When using shell or !, ppp(8) executes a shell or the passed arguments. The ppp program will wait for the command to complete before continuing. Any attempt to use the PPP link while running the command will appear as a frozen link. This is because ppp(8) is waiting for the command to complete.

To execute commands like this, use !bg instead. This will execute the given command in the background, and ppp(8) can continue to service the link.

14.14. Why does ppp8 over a null-modem cable never exit?

There is no way for ppp(8) to automatically determine that a direct connection has been dropped. This is due to the lines that are used in a null-modem serial cable. When using this sort of connection, LQR should always be enabled with the following line:

```
enable lqr
```

LQR is accepted by default if negotiated by the peer.

14.15. Why does ppp8 dial for no reason in -auto mode?

If ppp(8) is dialing unexpectedly, determine the cause, and set up dial filters to prevent such dialing.

To determine the cause, use the following line:

```
set log +tcp/ip
```

This will log all traffic through the connection. The next time the line comes up unexpectedly, the reason will be logged with a convenient timestamp next to it.

Next, disable dialing under these circumstances. Usually, this sort of problem arises due to DNS lookups. To prevent DNS lookups from establishing a connection (this will not prevent ppp(8) from passing the packets through an established connection), use the following:

```
set dfilter 1 deny udp src eq 53
set dfilter 2 deny udp dst eq 53
set dfilter 3 permit 0/0 0/0
```

This is not always suitable, as it will effectively break demand-dial capabilities. Most programs will need a DNS lookup before doing any other network related things.

In the DNS case, try to determine what is actually trying to resolve a host name. A lot of the time, Sendmail is the culprit. Make sure to configure Sendmail not to do any DNS lookups in its configuration file. See the section on using email with a dialup connection in the FreeBSD Handbook for details. You may also want to add the following line to .mc:

```
define(`confDELIVERY_MODE', `d')dnl
```

This will make Sendmail queue everything until the queue is run, usually, every 30 minutes, or until a sendmail -q is done, perhaps from /etc/ppp/ppp.linkup.

14.16. What do these CCP errors mean?

I keep seeing the following errors in my log file:

```
CCP: CcpSendConfigReq
CCP: Received Terminate Ack (1) state = Req-Sent (6)
```

This is because ppp(8) is trying to negotiate Predictor1 compression, but the peer does not want to negotiate any compression at all. The messages are harmless, but can be silenced by disabling the compression:

```
disable pred1
```

14.17. Why does ppp8 not log my connection speed?

To log all lines of the modem conversation, enable the following:

```
set log +connect
```

This will make ppp(8) log everything up until the last requested "expect" string.

To see the connect speed when using PAP or CHAP, make sure to configure ppp(8) to expect the whole CONNECT line, using something like this:

```
set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 4 \
 \"\" ATZ OK-ATZ-OK ATDT\\T TIMEOUT 60 CONNECT \\c \\n"
```

This gets the CONNECT, sends nothing, then expects a line-feed, forcing ppp(8) to read the whole CONNECT response.

14.18. Why does ppp8 ignore the \ character in my chat script?

The ppp utility parses each line in its configuration files so that it can interpret strings such as set phone "123 456 789" correctly and realize that the number is actually only one argument. To specify a " character, escape it using a backslash (\).

When the chat interpreter parses each argument, it re-interprets the argument to find any special escape sequences such as \P or \T. As a result of this double-parsing, remember to use the correct number of escapes.

To actually send a \ character, do something like:

```
set dial "\"\" ATZ OK-ATZ-OK AT\\\\X OK"
```

It will result in the following sequence:

```
ATZ
OK
AT\X
OK
```

Or:

```
set phone 1234567
set dial "\"\" ATZ OK ATDT\\T"
```

It will result in the following sequence:

```
ATZ
OK
ATDT1234567
```

14.19. What are FCS errors?

FCS stands for Frame Check Sequence. Each PPP packet has a checksum attached to ensure that the data being received is the data being sent. If the FCS of an incoming packet is incorrect, the packet is dropped and the HDLC FCS count is increased. The HDLC error values can be displayed using the show hdlc command.

If the link is bad or if the serial driver is dropping packets, it will produce the occasional FCS error. This is not usually worth worrying about although it does slow down the compression protocols substantially.

If the link freezes as soon as it connects and produces a large number of FCS errors, make sure the modem is not using software flow control (XON/XOFF). If the link must use software flow control, use set accmap 0x000a0000 to tell ppp(8) to escape the ^Q and ^S characters.

Another reason for too many FCS errors may be that the remote end has stopped talking PPP. In this case, enable async logging to determine if the incoming data is actually a login or shell prompt. If it is a shell prompt at the remote end, it is possible to terminate ppp(8) without dropping the line by using close lcp followed by term) to reconnect to the shell on the remote machine.

If nothing in the log file indicates why the link was terminated, ask the remote administrator or ISP why the session was terminated.

14.20. None of this helps — I am desperate! What can I do?

If all else fails, send the details of the error, the configuration files, how ppp(8) is being started, the relevant parts of the log file, and the output of netstat -rn, before and after connecting, to the FreeBSD general questions mailing list.

Chapter 15. Serial Communications

This section answers common questions about serial communications with FreeBSD. PPP is covered in the Networking section.

15.1. Which multi-port serial cards are supported by FreeBSD?

There is a list of these in the Serial Communications chapter of the Handbook.

Most multi-port PCI cards that are based on 16550 or clones are supported with no extra effort.

Some unnamed clone cards have also been known to work, especially those that claim to be AST compatible.

Check uart(4) and sio(4) to get more information on configuring such cards.

15.2. How do I get the boot: prompt to show on the serial console?

See this section of the Handbook.

15.3. How do I tell if FreeBSD found my serial ports or modem cards?

As the FreeBSD kernel boots, it will probe for the serial ports for which the kernel is configured. Either watch the boot messages closely or run this command after the system is up and running:

```
% grep -E '^(sio|uart)[0-9]' < /var/run/dmesg.boot
sio0: <16550A-compatible COM port> port 0x3f8-0x3ff irq 4 flags 0x10 on acpi0
sio0: type 16550A
sio1: <16550A-compatible COM port> port 0x2f8-0x2ff irq 3 on acpi0
sio1: type 16550A
```

This example shows two serial ports. The first is on IRQ4, port address 0x3f8, and has a 16550A-type UART chip. The second uses the same kind of chip but is on IRQ3 and is at port address 0x2f8. Internal modem cards are treated just like serial ports, except that they always have a modem attached to the port.

The GENERIC kernel includes support for two serial ports using the same IRQ and port address settings in the above example. If these settings are not right for the system, or if there are more modem cards or serial ports than the kernel is configured for, reconfigure using the instructions in building a kernel for more details.

15.4. How do I access the serial ports on FreeBSD?

The third serial port, sio2, or COM3, is on /dev/cuad2 for dial-out devices, and on /dev/ttyd2 for dial-in devices. What is the difference between these two classes of devices?

When opening /dev/ttydX in blocking mode, a process will wait for the corresponding cuadX device to become inactive, and then wait for the carrier detect line to go active. When the cuadX device is opened, it makes sure the serial port is not already in use by the ttydX device. If the port is available, it steals it from the ttydX device. Also, the cuadX device does not care about carrier detect. With this scheme and an auto-answer modem, remote users can log in and local users can

still dial out with the same modem and the system will take care of all the conflicts.

15.5. How do I enable support for a multi-port serial card?

The section on kernel configuration provides information about configuring the kernel. For a multi-port serial card, place an sio(4) line for each serial port on the card in the device.hints(5) file. But place the IRQ specifiers on only one of the entries. All of the ports on the card should share one IRQ. For consistency, use the last serial port to specify the IRQ. Also, specify the following option in the kernel configuration file:

```
options COM_MULTIPORT
```

The following /boot/device.hints example is for an AST 4-port serial card on IRQ 12:

```
hint.sio.4.at="isa"
hint.sio.4.port="0x2a0"
hint.sio.4.flags="0x701"
hint.sio.5.at="isa"
hint.sio.5.port="0x2a8"
hint.sio.5.flags="0x701"
hint.sio.6.at="isa"
hint.sio.6.port="0x2b0"
hint.sio.6.flags="0x701"
hint.sio.7.at="isa"
hint.sio.7.port="0x2b8"
hint.sio.7.flags="0x701"
hint.sio.7.irq="12"
```

The flags indicate that the master port has minor number 7 (0x700), and all the ports share an IRQ (0x001).

15.6. Can I set the default serial parameters for a port?

See the Serial Communications section in the FreeBSD Handbook.

15.7. How can I enable dialup logins on my modem?

Refer to the section about Dial-in Services in the FreeBSD Handbook.

15.8. How can I connect a dumb terminal to my FreeBSD box?

This information is in the Terminals section of the FreeBSD Handbook.

15.9. Why can I not run tip or cu?

The built-in tip(1) and cu(1) utilities can only access the /var/spool/lock directory via user uucp and group dialer. Use the dialer group to control who has access to the modem or remote systems by adding user accounts to dialer.

Alternatively, everyone can be configured to run tip(1) and cu(1) by typing:

```
# chmod 4511 /usr/bin/cu
# chmod 4511 /usr/bin/tip
```

Chapter 16. Miscellaneous Questions

16.1. FreeBSD uses a lot of swap space even when the computer has free memory left. Why?

FreeBSD will proactively move entirely idle, unused pages of main memory into swap in order to make more main memory available for active use. This heavy use of swap is balanced by using the extra free memory for caching.

Note that while FreeBSD is proactive in this regard, it does not arbitrarily decide to swap pages when the system is truly idle. Thus, the system will not be all paged out after leaving it idle overnight.

16.2. Why does top show very little free memory even when I have very few programs running?

The simple answer is that free memory is wasted memory. Any memory that programs do not actively allocate is used within the FreeBSD kernel as disk cache. The values shown by top(1) labeled as Inact and Laundry are cached data at different aging levels. This cached data means the system does not have to access a slow disk again for data it has accessed recently, thus increasing overall performance. In general, a low value shown for Free memory in top(1) is good, provided it is not very low.

16.3. Why will chmod not change the permissions on symlinks?

Symlinks do not have permissions, and by default, chmod(1) will follow symlinks to change the permissions on the source file, if possible. For the file, foo with a symlink named bar, this command will always succeed.

```
% chmod g-w bar
```

However, the permissions on bar will not have changed.

When changing modes of the file hierarchies rooted in the files instead of the files themselves, use either -H or -L together with -R to make this work. See chmod(1) and symlink(7) for more information.

> -R does a recursive chmod(1). Be careful about specifying directories or symlinks to directories to chmod(1). To change the permissions of a directory referenced by a symlink, use chmod(1) without any options and follow the symlink with a trailing slash (/). For example, if foo is a symlink to directory bar, to change the permissions of foo (actually bar), do something like:
>
> ```
> % chmod 555 foo/
> ```
>
> With the trailing slash, chmod(1) will follow the symlink, foo, to change the permissions of the directory, bar.

16.4. Can I run DOS binaries under FreeBSD?

Yes. A DOS emulation program, emulators/doscmd, is available in the FreeBSD Ports Collection.

If doscmd will not suffice, emulators/pcemu emulates an 8088 and enough BIOS services to run many DOS text-mode applications. It requires the X Window System.

The Ports Collection also has emulators/dosbox. The main focus of this application is emulating old DOS games using the local file system for files.

16.5. What do I need to do to translate a FreeBSD document into my native language?

See the Translation FAQ in the FreeBSD Documentation Project Primer.

16.6. Why does my email to any address at FreeBSD.org bounce?

The FreeBSD.org mail system implements some Postfix checks on incoming mail and rejects mail that is either from misconfigured relays or otherwise appears likely to be spam. Some of the specific requirements are:

- The IP address of the SMTP client must "reverse-resolve" to a forward confirmed hostname.
- The fully-qualified hostname given in the SMTP conversation (either HELO or EHLO) must resolve to the IP address of the client.

Other advice to help mail reach its destination include:

- Mail should be sent in plain text, and messages sent to mailing lists should generally be no more than 200KB in length.
- Avoid excessive cross posting. Choose one mailing list which seems most relevant and send it there.

If you still have trouble with email infrastructure at FreeBSD.org, send a note with the details to postmaster@freebsd.org; Include a date/time interval so that logs may be reviewed — and note that we only keep one week’ s worth of mail logs. (Be sure to specify the time zone or offset from UTC.)

16.7. Where can I find a free FreeBSD account?

While FreeBSD does not provide open access to any of their servers, others do provide open access UNIX™ systems. The charge varies and limited services may be available.

Arbornet, Inc, also known as M-Net, has been providing open access to UNIX™ systems since 1983. Starting on an Altos running System III, the site switched to BSD/OS in 1991. In June of 2000, the site switched again to FreeBSD. M-Net can be accessed via telnet and SSH and provides basic access to the entire FreeBSD software suite. However, network access is limited to members and patrons who donate to the system, which is run as a non-profit organization. M-Net also provides an bulletin board system and interactive chat.

16.8. What is the cute little red guy’ s name?

He does not have one, and is just called "the BSD daemon". If you insist upon using a name, call him "beastie". Note that "beastie" is pronounced "BSD".

More about the BSD daemon is available on his home page.

16.9. Can I use the BSD daemon image?

Perhaps. The BSD daemon is copyrighted by Marshall Kirk McKusick. Check his Statement on the

Use of the BSD Daemon Figure for detailed usage terms.

In summary, the image can be used in a tasteful manner, for personal use, so long as appropriate credit is given. Before using the logo commercially, contact Kirk McKusick mckusick@FreeBSD.org for permission. More details are available on the BSD Daemon’s home page.

16.10. Do you have any BSD daemon images I could use?

Xfig and eps drawings are available under /usr/shared/examples/BSD_daemon/.

16.11. I have seen an acronym or other term on the mailing lists and I do not understand what it means. Where should I look?

Refer to the FreeBSD Glossary.

16.12. Why should I care what color the bikeshed is?

The really, really short answer is that you should not. The somewhat longer answer is that just because you are capable of building a bikeshed does not mean you should stop others from building one just because you do not like the color they plan to paint it. This is a metaphor indicating that you need not argue about every little feature just because you know enough to do so. Some people have commented that the amount of noise generated by a change is inversely proportional to the complexity of the change.

The longer and more complete answer is that after a very long argument about whether sleep(1) should take fractional second arguments, Poul-Henning Kamp phk@FreeBSD.org posted a long message entitled "A bike shed (any color will do) on greener grass…". The appropriate portions of that message are quoted below.

Poul-Henning Kamp phk@FreeBSD.org on freebsd-hackers, October 2, 1999 "What is it about this bike shed?" Some of you have asked me.

It is a long story, or rather it is an old story, but it is quite short actually. C. Northcote Parkinson wrote a book in the early 1960s, called "Parkinson’s Law", which contains a lot of insight into the dynamics of management.

[snip a bit of commentary on the book]

In the specific example involving the bike shed, the other vital component is an atomic power-plant, I guess that illustrates the age of the book.

Parkinson shows how you can go into the board of directors and get approval for building a multi-million or even billion dollar atomic power plant, but if you want to build a bike shed you will be tangled up in endless discussions.

Parkinson explains that this is because an atomic plant is so vast, so expensive and so complicated that people cannot grasp it, and rather than try, they fall back on the assumption that somebody else checked all the details before it got this far. Richard P. Feynmann gives a couple of interesting, and very much to the point, examples relating to Los Alamos in his books.

A bike shed on the other hand. Anyone can build one of those over a weekend, and still have time to watch the game on TV. So no matter how well prepared, no matter how reasonable you are with your proposal, somebody will seize the chance to show that he is doing his job, that he is paying attention, that he is here.

In Denmark we call it "setting your fingerprint". It is about personal pride and prestige, it is about being able to point somewhere and say "There! I did that." It is a strong trait in politicians, but present in most people given the chance. Just think about footsteps in wet cement.

Chapter 17. The FreeBSD Funnies

17.1. How cool is FreeBSD?

1. Has anyone done any temperature testing while running FreeBSD? I know Linux™ runs cooler than DOS, but have never seen a mention of FreeBSD. It seems to run really hot.

   No, but we have done numerous taste tests on blindfolded volunteers who have also had 250 micrograms of LSD-25 administered beforehand. 35% of the volunteers said that FreeBSD tasted sort of orange, whereas Linux™ tasted like purple haze. Neither group mentioned any significant variances in temperature. We eventually had to throw the results of this survey out entirely anyway when we found that too many volunteers were wandering out of the room during the tests, thus skewing the results. We think most of the volunteers are at Apple now, working on their new "scratch and sniff" GUI. It is a funny old business we are in!

Seriously, FreeBSD uses the HLT (halt) instruction when the system is idle thus lowering its energy consumption and therefore the heat it generates. Also if you have ACPI (Advanced Configuration and Power Interface) configured, then FreeBSD can also put the CPU into a low power mode.

17.2. Who is scratching in my memory banks??

1. Is there anything "odd" that FreeBSD does when compiling the kernel which would cause the memory to make a scratchy sound? When compiling (and for a brief moment after recognizing the floppy drive upon startup, as well), a strange scratchy sound emanates from what appears to be the memory banks.

   Yes! You will see frequent references to "daemons" in the BSD documentation, and what most people do not know is that this refers to genuine, non-corporeal entities that now possess your computer. The scratchy sound coming from your memory is actually high-pitched whispering exchanged among the daemons as they best decide how to deal with various system administration tasks.

If the noise gets to you, a good fdisk /mbr from DOS will get rid of them, but do not be surprised if they react adversely and try to stop you. In fact, if at any point during the exercise you hear the satanic voice of Bill Gates coming from the built-in speaker, take off running and do not ever look back! Freed from the counterbalancing influence of the BSD daemons, the twin demons of DOS and Windows™ are often able to re-assert total control over your machine to the eternal damnation of your soul. Now that you know, given a choice you would probably prefer to get used to the scratchy noises, no?

17.3. How many FreeBSD hackers does it take to change a lightbulb?

One thousand, one hundred and sixty-nine:

Twenty-three to complain to -CURRENT about the lights being out;

Four to claim that it is a configuration problem, and that such matters really belong on -questions;

Three to submit PRs about it, one of which is misfiled under doc and consists only of "it’s dark";

One to commit an untested lightbulb which breaks buildworld, then back it out five minutes later;

Eight to flame the PR originators for not including patches in their PRs;

Five to complain about buildworld being broken;

Thirty-one to answer that it works for them, and they must have updated at a bad time;

One to post a patch for a new lightbulb to -hackers;

One to complain that he had patches for this three years ago, but when he sent them to -CURRENT they were just ignored, and he has had bad experiences with the PR system; besides, the proposed new lightbulb is non-reflexive;

Thirty-seven to scream that lightbulbs do not belong in the base system, that committers have no right to do things like this without consulting the Community, and WHAT IS -CORE DOING ABOUT IT!?

Two hundred to complain about the color of the bicycle shed;

Three to point out that the patch breaks style(9);

Seventeen to complain that the proposed new lightbulb is under GPL;

Five hundred and eighty-six to engage in a flame war about the comparative advantages of the GPL, the BSD license, the MIT license, the NPL, and the personal hygiene of unnamed FSF founders;

Seven to move various portions of the thread to -chat and -advocacy;

One to commit the suggested lightbulb, even though it shines dimmer than the old one;

Two to back it out with a furious flame of a commit message, arguing that FreeBSD is better off in the dark than with a dim lightbulb;

Forty-six to argue vociferously about the backing out of the dim lightbulb and demanding a statement from -core;

Eleven to request a smaller lightbulb so it will fit their Tamagotchi if we ever decide to port FreeBSD to that platform;

Seventy-three to complain about the SNR on -hackers and -chat and unsubscribe in protest;

Thirteen to post "unsubscribe", "How do I unsubscribe?", or "Please remove me from the list", followed by the usual footer;

One to commit a working lightbulb while everybody is too busy flaming everybody else to notice;

Thirty-one to point out that the new lightbulb would shine 0.364% brighter if compiled with TenDRA (although it will have to be reshaped into a cube), and that FreeBSD should therefore switch to TenDRA instead of GCC;

One to complain that the new lightbulb lacks fairings;

Nine (including the PR originators) to ask "what is MFC?";

Fifty-seven to complain about the lights being out two weeks after the bulb has been changed.

Nik Clayton nik@FreeBSD.org adds:

I was laughing quite hard at this.

And then I thought, "Hang on, shouldn’t there be '1 to document it.' in that list somewhere?"

And then I was enlightened :-)

Thomas Abthorpe tabthorpe@FreeBSD.org says: "None, real FreeBSD hackers are not afraid of the dark!"

17.4. Where does data written to /dev/null go?

It goes into a special data sink in the CPU where it is converted to heat which is vented through the heatsink / fan assembly. This is why CPU cooling is increasingly important; as people get used to

faster processors, they become careless with their data and more and more of it ends up in /dev/null, overheating their CPUs. If you delete /dev/null (which effectively disables the CPU data sink) your CPU may run cooler but your system will quickly become constipated with all that excess data and start to behave erratically. If you have a fast network connection you can cool down your CPU by reading data out of /dev/random and sending it off somewhere; however you run the risk of overheating your network connection and / or angering your ISP, as most of the data will end up getting converted to heat by their equipment, but they generally have good cooling, so if you do not overdo it you should be OK.

Paul Robinson adds:

There are other methods. As every good sysadmin knows, it is part of standard practice to send data to the screen of interesting variety to keep all the pixies that make up your picture happy. Screen pixies (commonly mis-typed or re-named as "pixels") are categorized by the type of hat they wear (red, green or blue) and will hide or appear (thereby showing the color of their hat) whenever they receive a little piece of food. Video cards turn data into pixie-food, and then send them to the pixies — the more expensive the card, the better the food, so the better behaved the pixies are. They also need constant stimulation — this is why screen savers exist.

To take your suggestions further, you could just throw the random data to console, thereby letting the pixies consume it. This causes no heat to be produced at all, keeps the pixies happy and gets rid of your data quite quickly, even if it does make things look a bit messy on your screen.

Incidentally, as an ex-admin of a large ISP who experienced many problems attempting to maintain a stable temperature in a server room, I would strongly discourage people sending the data they do not want out to the network. The fairies who do the packet switching and routing get annoyed by it as well.

17.5. My colleague sits at the computer too much, how can I prank her?

Install games/sl and wait for her to mistype sl for ls.

Chapter 18. Advanced Topics

18.1. How can I learn more about FreeBSD’s internals?

See the FreeBSD Architecture Handbook.

Additionally, much general UNIX™ knowledge is directly applicable to FreeBSD.

18.2. How can I contribute to FreeBSD? What can I do to help?

We accept all types of contributions: documentation, code, and even art. See the article on Contributing to FreeBSD for specific advice on how to do this.

And thanks for the thought!

18.3. What are snapshots and releases?

There are currently 2 active/semi-active branches in the FreeBSD Subversion Repository. (Earlier branches are only changed very rarely, which is why there are only 2 active branches of development):

- stable/11/ AKA 11-STABLE
- stable/12/ AKA 12-STABLE
- head/ AKA -CURRENT AKA 12-CURRENT

HEAD is not an actual branch tag. It is a symbolic constant for the current, non-branched development stream known as -CURRENT.

Right now, -CURRENT is the 13.X development stream; the 12-STABLE branch, stable/12/, forked off from -CURRENT in December 2018 and the 11-STABLE branch, stable/11/, forked off from -CURRENT in October 2016.

18.4. How can I make the most of the data I see when my kernel panics?

Here is typical kernel panic:

```
Fatal trap 12: page fault while in kernel mode
fault virtual address   = 0x40
fault code              = supervisor read, page not present
instruction pointer     = 0x8:0xf014a7e5
stack pointer           = 0x10:0xf4ed6f24
frame pointer           = 0x10:0xf4ed6f28
code segment            = base 0x0, limit 0xfffff, type 0x1b
                        = DPL 0, pres 1, def32 1, gran 1
processor eflags        = interrupt enabled, resume, IOPL = 0
current process         = 80 (mount)
interrupt mask          =
trap number             = 12
```

```
panic: page fault
```

This message is not enough. While the instruction pointer value is important, it is also configuration dependent as it varies depending on the kernel image. If it is a GENERIC kernel image from one of the snapshots, it is possible for somebody else to track down the offending function, but for a custom kernel, only you can tell us where the fault occurred.

To proceed:

1. Write down the instruction pointer value. Note that the 0x8: part at the beginning is not significant in this case: it is the 0xf0xxxxxx part that we want.
2. When the system reboots, do the following:

   ```
   % nm -n kernel.that.caused.the.panic | grep f0xxxxxx
   ```

   where f0xxxxxx is the instruction pointer value. The odds are you will not get an exact match since the symbols in the kernel symbol table are for the entry points of functions and the instruction pointer address will be somewhere inside a function, not at the start. If you do not get an exact match, omit the last digit from the instruction pointer value and try again:

   ```
   % nm -n kernel.that.caused.the.panic | grep f0xxxxx
   ```

   If that does not yield any results, chop off another digit. Repeat until there is some sort of output. The result will be a possible list of functions which caused the panic. This is a less than exact mechanism for tracking down the point of failure, but it is better than nothing.

However, the best way to track down the cause of a panic is by capturing a crash dump, then using kgdb(1) to generate a stack trace on the crash dump.

In any case, the method is this:

1. Make sure that the following line is included in the kernel configuration file:

   ```
   makeoptions    DEBUG=-g       # Build kernel with gdb(1) debug symbols
   ```

2. Change to the /usr/src directory:

   ```
   # cd /usr/src
   ```

3. Compile the kernel:

   ```
   # make buildkernel KERNCONF=MYKERNEL
   ```

4. Wait for make(1) to finish compiling.

```
# make installkernel KERNCONF=MYKERNEL
```

5. Reboot.

If KERNCONF is not included, the GENERIC kernel will instead be built and installed.

The make(1) process will have built two kernels. /usr/obj/usr/src/sys/MYKERNEL/kernel and /usr/obj/usr/src/sys/MYKERNEL/kernel.debug. kernel was installed as /boot/kernel/kernel, while kernel.debug can be used as the source of debugging symbols for kgdb(1).

To capture a crash dump, edit /etc/rc.conf and set dumpdev to point to either the swap partition or AUTO. This will cause the rc(8) scripts to use the dumpon(8) command to enable crash dumps. This command can also be run manually. After a panic, the crash dump can be recovered using savecore(8); if dumpdev is set in /etc/rc.conf, the rc(8) scripts will run savecore(8) automatically and put the crash dump in /var/crash.

FreeBSD crash dumps are usually the same size as physical RAM. Therefore, make sure there is enough space in /var/crash to hold the dump. Alternatively, run savecore(8) manually and have it recover the crash dump to another directory with more room. It is possible to limit the size of the crash dump by using options MAXMEM=N where N is the size of kernel’s memory usage in KBs. For example, for 1 GB of RAM, limit the kernel’s memory usage to 128 MB, so that the crash dump size will be 128 MB instead of 1 GB.

Once the crash dump has been recovered , get a stack trace as follows:

```
% kgdb /usr/obj/usr/src/sys/MYKERNEL/kernel.debug /var/crash/vmcore.0
(kgdb) backtrace
```

Note that there may be several screens worth of information. Ideally, use script(1) to capture all of them. Using the unstripped kernel image with all the debug symbols should show the exact line of kernel source code where the panic occurred. The stack trace is usually read from the bottom up to trace the exact sequence of events that lead to the crash. kgdb(1) can also be used to print out the contents of various variables or structures to examine the system state at the time of the crash.

If a second computer is available, kgdb(1) can be configured to do remote debugging, including setting breakpoints and single-stepping through the kernel code.

If DDB is enabled and the kernel drops into the debugger, a panic and a crash dump can be forced by typing panic at the ddb prompt. It may stop in the debugger again during the panic phase. If it does, type continue and it will finish the crash dump.

18.5. Why has dlsym() stopped working for ELF executables?

The ELF toolchain does not, by default, make the symbols defined in an executable visible to the dynamic linker. Consequently dlsym() searches on handles obtained from calls to dlopen(NULL, flags) will fail to find such symbols.

To search, using dlsym(), for symbols present in the main executable of a process, link the

executable using the --export-dynamic option to the ELF linker (ld(1)).

18.6. How can I increase or reduce the kernel address space on i386?

By default, the kernel address space is 1 GB (2 GB for PAE) for i386. When running a network-intensive server or using ZFS, this will probably not be enough.

Add the following line to the kernel configuration file to increase available space and rebuild the kernel:

```
options KVA_PAGES=N
```

To find the correct value of N, divide the desired address space size (in megabytes) by four. (For example, it is 512 for 2 GB.)

Chapter 19. Acknowledgments

This innocent little Frequently Asked Questions document has been written, rewritten, edited, folded, spindled, mutilated, eviscerated, contemplated, discombobulated, cogitated, regurgitated, rebuilt, castigated, and reinvigorated over the last decade, by a cast of hundreds if not thousands. Repeatedly.

We wish to thank every one of the people responsible, and we encourage you to join them in making this FAQ even better.