

# FreeBSD 5.X、6.X 常見問答集

FreeBSD 文件計畫

## FreeBSD 5.X、6.X 常見問答集

由FreeBSD 文件計畫

版權 © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 FreeBSD 文件計畫

這份文件是FreeBSD 5.X 及6.X 的常見問答集。除非有特別加註，否則這些項目都適用於FreeBSD 5.0 及以後的版本。(如果條目內容中有<XXX> 則是尚未完成中譯的部份。) 如果您對協助本文件/翻譯計畫的進行有興趣的話，請寄e-mail 到FreeBSD documentation project 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc>)。此外，隨時可從 FreeBSD 網站 ([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/books/faq/index.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/faq/index.html)) 拿到這份文件的最新版本。也可以利用HTTP 來下載一份龐大的HTML (book.html) 文件，或是經由 FreeBSD FTP 站 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/>) 下載純文字、PostScript®、或PDF 版本的檔案。您也可以在這裡使用搜尋資料 (<http://www.FreeBSD.org/search/search.html>) 的功能。

Redistribution and use in source (XML DocBook) and 'compiled' forms (XML, HTML, PDF, PostScript, RTF and so forth) with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code (XML DocBook) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.
2. Redistributions in compiled form (transformed to other DTDs, converted to PDF, PostScript, RTF and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

**重要:** THIS DOCUMENTATION IS PROVIDED BY THE FREEBSD DOCUMENTATION PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD DOCUMENTATION PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

FreeBSD 是FreeBSD基金會的註冊商標

3Com 和HomeConnect 是3Com Corporation 的註冊商標。

Adobe, Acrobat, Acrobat Reader, 以及PostScript 是Adobe Systems Incorporated 在美國和/或其他國家的商標或註冊商標。

Sound Blaster 是Creative Technology Ltd. 在美國和/或其他國家的註冊商標。

CVSup 是John D. Polstra 的註冊商標。

IBM, AIX, EtherJet, Netfinity, OS/2, PowerPC, PS/2, S/390, 和ThinkPad 是國際商用機器公司在美國和其他國家的註冊商標或商標。

IEEE, POSIX, 和802 是Institute of Electrical and Electronics Engineers, Inc. 在美國的註冊商標。

Intel, Celeron, EtherExpress, i386, i486, Itanium, Pentium, 和Xeon 是Intel Corporation 及其分支機構在美國和其他國家的商標或註冊商標。

Iomega, Zip, 和Jaz 是Iomega Corporation 在美國和/或其他國家的商標或註冊商標。

Linux 是Linus Torvalds 的註冊商標。

Microsoft, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media, 和Windows NT 是Microsoft Corporation 在美國和/或其他國家的商標或註冊商標。

MIPS 和R4000 是MIPS Technologies, Inc. 在美國以及其他國家的註冊商標。

Netscape 以及Netscape Navigator 是Netscape Communications Corporation 在美國和其他國家的註冊商標。

Motif, OSF/1, 和UNIX 是The Open Group 在美國和其他國家的註冊商標；IT DialTone 和The Open Group 是其商標。

Oracle 是Oracle Corporation 的註冊商標。

Silicon Graphics, SGI, 和OpenGL 是Silicon Graphics, Inc. 在美國和/或其他國家的註冊商標。

Sparc, Sparc64, SPARCEngine, 以及UltraSPARC 是SPARC International, Inc 在美國和其他國家的商標。為Sun Microsystems, Inc. 所開發的架構，包含SPARC 商標的產品。

Sun, Sun Microsystems, Java, Java Virtual Machine, JavaServer Pages, JDK, JSP, JVM, Netra, Solaris, StarOffice, Sun Blade, Sun Enterprise, Sun Fire, SunOS, 和Ultra 是Sun Microsystems, Inc. 在美國和其他國家的商標或註冊商標。

U.S. Robotics 和Sportster 是U.S. Robotics Corporation 的註冊商標。

XFree86 是The XFree86 Project, Inc 的商標。

許多製造商和經銷商使用一些稱為商標的圖案或文字設計來彰顯自己的產品。本文中出現的衆多商標，以及FreeBSD Project 本身廣所人知的商標，後面將以‘TM’ 或‘®’ 符號來標註。

# 內容目錄

1. 前言、一般問題.....	1
2. 文件與技術支援.....	6
3. 安裝.....	10
4. 硬體支援方面.....	19
4.1. 一般問題.....	19
4.2. 硬體架構及CPU.....	19
4.3. 硬碟、磁帶機以及光碟、DVD、燒錄機.....	19
4.4. 鍵盤、滑鼠.....	21
4.5. 網路跟serial 設備.....	23
4.6. 音效卡.....	24
4.7. 其他怪異問題(ACPI、重開機後掛了..等).....	25
5. 常見問題解決.....	27
6. 商業軟體.....	39
7. 一般應用程式.....	42
8. kernel 設定.....	46
9. 硬碟、檔案系統、Boot Loader.....	48
10. 系統管理.....	58
11. X Window System 及Virtual Consoles.....	68
12. Networking.....	78
13. 系統安全篇.....	84
14. PPP.....	87
15. Serial Communications.....	99
16. 其它各式各樣的問題.....	106
17. FreeBSD 冷笑話集.....	112
18. 進階主題.....	115
19. 感謝.....	121
20. Ports and Packages 常見問題.....	122
Bibliography.....	133

# 附表目錄

3-1. 檔案大小的最大限制 .....	16
12-1. Network cards based on the DEC PCI chipset .....	80

# 章1. 前言、一般問題

歡迎使用FreeBSD 4.X-6.X FAQ!

跟其他Usenet 上的FAQ 一樣，這份文件涵蓋了有關FreeBSD 這套作業系統最常被問到的問題(當然包括了回答!)。雖然說我們本來的目的是為了減少網路頻寬的浪費，以及避免同樣的問題一再出現，但事實上FAQ 已被公認為是值得閱讀的文件資源。

我們已經儘可能地使這份FAQ 更豐富了。如果您對如何改善、進步方面有任何建議，請隨時寄電子郵件給FreeBSD documentation project 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc>)。

## 1. 什麼是FreeBSD?

簡單地來說，FreeBSD 是一套可以在Alpha/AXP, AMD64 及Intel® EM64T, i386™ IA-64, PC-98, UltraSPARC® 上執行的UN\*X-like 作業系統，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。您可以在FreeBSD 首頁 (<http://www.FreeBSD.org/index.html>) 找到FreeBSD 以及它可以幫您做些什麼的相關資訊。

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 ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/index.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/index.html))。

## 2. 發展FreeBSD 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軟體。我們在程式碼(以及計畫本身) 上付出了大量心血，當然不會介意來點金錢上的回饋，不過我們絕對不會如此堅持。我們相信我們首要的“任務” 就是提供程式碼給每一個使用者，不管他們打算用來幹嘛；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效益。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GPL (<http://www.FreeBSD.org/copyright/COPYING>) 或是LGPL (<http://www.FreeBSD.org/copyright/COPYING.LIB>) 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寬鬆的FreeBSD 版權 (<http://www.FreeBSD.org/copyright/freebsd-license.html>)的軟體來取代這些GPL 版權宣告的軟體。

## 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。如果你有版權焦慮症的話，請閱讀 版權本文 (<http://www.FreeBSD.org/copyright/freebsd-license.html>)。簡單地來說，這份版權的重點可以條列如下。

- 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。

#### 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 (<http://www.freebsd.org/ports/>) 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，或是想在不被中斷的環境下工作的話，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 環境轉換到FreeBSD 的話，基本上是大同小異的。但是如果你之前用的是圖形界面的作業系統，例如說是Windows® 或是比較古老的Mac OS® 的話，可能就要多花一點時間來學習怎麼用UNIX 的方式來做。你可以從這份FAQ 和 FreeBSD 使用手冊 ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/index.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/index.html)) 來入門。

#### 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。

#### 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 在 DaemonNews (<http://www.daemonnews.org/>) 上面寫了 The BSD Family Tree ([http://ezine.daemonnews.org/200104/bsd\\_family.html](http://ezine.daemonnews.org/200104/bsd_family.html)) 的文件，裡面說明了這些歷史淵源及這些\*BSD 家族計畫之間的差異。

#### 7. 最新版的FreeBSD 是那一版？

就FreeBSD 目前的發展而言，有兩個主要發展分支：由5-STABLE 所發行(release)的5.X 系列、由6-STABLE 所發行(release)的6.X 系列這兩個分支。

在5.3 release 之前，4.X 系列仍屬-STABLE 分支。自從5.3 開始，5.X 系列開始規劃新的-STABLE 分支發展重點，而4.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”，不再會有新的突破性發展。另一方面，5-STABLE 分支雖然仍將繼續發行，但是由於它只是“legacy” 過渡期分支，所以大多數主力都已轉移到6-STABLE 繼續開發。

於2006 年5 月所發行的9.1 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/9.1-RELEASE>) 版是目前最新的6-STABLE 分支；而於2006 年5 月所發行的8.4 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.4-RELEASE/>) 版則是目前最新的5-STABLE 分支。

簡單地來說，*-STABLE* 的主要訴求對象是對於穩定性及低變異性的需求遠勝過對最新-*CURRENT* snapshot 中特別新功能的需求，例如ISP 或公司行號的使用者。這兩個branch 都有可能產生release 版，但只有當你能接受-*CURRENT* 遠比-*STABLE* 容易更動這一點，才應該用-*CURRENT*。

Release 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 (<http://www.FreeBSD.org/releng/index.html>) 得知。

## 8. 什麼是FreeBSD-CURRENT ?

FreeBSD-*CURRENT* ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/cutting-edge.html#CURRENT](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/cutting-edge.html#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 ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/index.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/index.html)) 的相關部份 ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/cutting-edge.html#CURRENT](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/cutting-edge.html#CURRENT))。

如果您對作業系統本身並不是很熟悉，或是您沒辦法分辨您遇到的問題是真的發生了問題亦或是暫時性的小狀況，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並且只回報真正的問題而非“小狀況”。如果您在-*CURRENT* mailing list 中提到類似“make world 造成一些有關groups 的錯誤”之類的問題的話，也許會被其他人輕視。

我們每天都會根據目前-*CURRENT* 和-*STABLE* 的狀況對這兩個分支各發行一個snapshot (<http://www.FreeBSD.org/snapshots/>) 版。有的時候甚至還會發行可供取得的版本。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昇級的使用者。
- 為了替我們發展中的程式保留一個固定的參考點，以防止我們未來造成不幸。(雖然一般而言CVS 可以防止類似這種的可怕事件:)
- 為了確保所有需要測試的新功能都可以得到最多的測試。

我們不對-*CURRENT* snapshot 做任何形式的“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release 的版本，或是使用-*STABLE* snapshots。

您可以直接從 snapshots (<http://www.FreeBSD.org/snapshots/>) 處取得-*CURRENT* 的snapshot release。

對每個有在活動的分支而言，平均每天都會產生一次snapshots。

## 9. 什麼是FreeBSD-STABLE ?

回溯到FreeBSD 2.0.5 剛發表的時候，我們決定把FreeBSD 的發展分成兩支。一支叫做-*STABLE* ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/current-stable.html#STABLE](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/current-stable.html#STABLE))，我們只對它做錯誤修正及小幅度的修改(這是給ISP 和商業公司等，對實驗中功能不感興趣的單位所使用的)。另外一支叫做-*CURRENT* ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/current-stable.html#CURRENT](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/current-stable.html#CURRENT))，從2.0 版發行以後，就不斷地朝著6.2-*RELEASE* (含後續的版本)前進著。

6-STABLE 分支是從6.0-RELEASE 開始(5-STABLE 分支算是5.3-RELEASE 之後才開始的), 然後原本的FreeBSD-CURRENT 就會成為7-CURRENT。

2.2-STABLE 這個分支隨著2.2.8 的發表而功成身退。3-STABLE 這個分支則是結束在3.5.1 發表之後, 它也是3.X 的最後一次發表。之後除了安全漏洞的後續修正之外, 這些分支就幾乎沒有再更動過。而5-STABLE 分支的支援將仍持續一段時間, 但主要焦點僅在於安全方面的漏洞、臭蟲及其他嚴重問題的修補。

6-STABLE 是目前正在發展中的-STABLE 分支。6-STABLE 的最新的一次發表是在2006 年5 月發行的9.1-RELEASE。

7-CURRENT 這個分支是FreeBSD 的-CURRENT 分支, 仍然不斷地在發展當中。如果想要知道更多關於這個分支的資訊的話, 請參考什麼是FreeBSD-CURRENT?。

## 10. FreeBSD 版本命名原則是什麼呢?

您可能會看到以下幾種FreeBSD 的版本名稱:

- 1) 4.4.2-RELEASE、4.4-BETA: 最前面的數字A.B.C 裡, A 表示主要的作業系統版本(Major), B 表示次要的作業系統版本(Minor), C 表示些微修正版本(Patch)。後面的英文表示是否為正式版, 通常是測試版(ALPHA、BETA、GAMMA)、正式版(RELEASE)。
- 2) 6.0-STABLE、7.0-CURRENT: 通常FreeBSD 會有一個以發展新功能為主的版本, 稱為-CURRENT, 目前CURRENT 版本是7.0; FreeBSD 也會有一個以維護穩定性及系統安全為主的版本, 稱為-STABLE, 目前STABLE 的版本是6.0。雖然如此, 並不代表-STABLE 就沒有新功能, 也不代表\_CURRENT 就不穩定不安全, 這兩個版本是相輔相成的, 而且終有一天7.0-CURRENT 會變成7.0-STABLE, 而開始8.0-CURRENT 的發展。不過一般來說, 由於-CURRENT 系統開發的速度相當快, 跟-STABLE 相比較不穩定, 而且最好是有相當經驗的使用者才來使用。如果是商業環境或是伺服器站台, 最好還是跑-STABLE 與-RELEASE 比較好。由於-STABLE 與-CURRENT 都是一直在開發維護中的版本, 因此沒有一個特定的數字版本可以稱呼, 因此通常我們會以編譯日期來代表是哪個時候的-STABLE 或-CURRENT, 例如「4.4-STABLE、編譯日期2001/10/08」。
- 3) 5.0-011025-SNAP、4.2-010816-RELENG: SNAP 與RELENG 並不是正式發行的版本, 其中3.0-970625-SNAP 表示是在2001 年10 月25 日發行的5.0 測試版(SNAPshot), 通常是指CURRENT 的版本。而4.2-010816-RELENG 表示是在2001 年8 月16 日發行的4.2 非正式的穩定版, 通常是指STABLE 的版本。通常FreeBSD 核心小組會不定時釋出SNAP 以供測試, 然後有一天SNAP 會變成ALPHA->BETA->GAMMA, 再來是RELEASE, 而後可能會有不定時的RELENG 以供測試使用, 但是RELENG 的穩定性通常又比前期的RELEASE 來的好。

您要是習慣微軟的術語, 這樣說好了, SNAP 是開發期間的內部流出版, 那個日期(011025)就是Build Number, 而ALPHA 與BETA 是搶鮮版, RELEASE(RELENG 勉強也算)是正式版, 這樣應該了解了吧。

## 11. 每次新的FreeBSD 將於什麼時候推出?

一般而言, Release Engineering Team <re@FreeBSD.org> 平均每四個月發行一次release, 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, 且不會影響系統穩定度。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(<http://www.FreeBSD.org/releng/index.html>) 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(-CURRENT的部分)已經提到我們每天都會發行snapshots 版可供使用。

## 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

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/articles/contributors/article.html#STAFF-CORE](http://www.FreeBSD.org/doc/zh_TW.Big5/articles/contributors/article.html#STAFF-CORE))來決定。而有另一群超過300個人的 committers

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/articles/contributors/article.html#STAFF-COMMITTERS](http://www.FreeBSD.org/doc/zh_TW.Big5/articles/contributors/article.html#STAFF-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先在 mailing lists 先討論過，而且不分角色，每個人都可以參與討論。

## 13. 我要如何取得FreeBSD？

每個FreeBSD的重要版本都可以經由匿名ftp從FreeBSD FTP站 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/>)取得：

- 如果需要6-STABLE的最新版，也就是9.1-RELEASE，請到9.1-RELEASE (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/9.1-RELEASE/>)這個目錄
- 7-CURRENT Snapshot (<ftp://current.FreeBSD.org/pub/FreeBSD/>) 通常也是每天都會做一份，這是從-CURRENT分支做出來的，主要是為了提供給那些熱心的測試者和開發人員。
- 如果需要5-STABLE的最新版，也就是8.4-RELEASE，請到8.4-RELEASE (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.4-RELEASE/>)這個目錄
- 5.X、6X snapshots (<ftp://current.FreeBSD.org/pub/FreeBSD/snapshots/>) 通常每天都會做一份。

FreeBSD的CD、DVD，還有其他取得方式可以在使用手冊

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/mirrors.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/mirrors.html)) 中找到解答。

## 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介面的PR查詢介面

(<http://www.FreeBSD.org/cgi/query-pr-summary.cgi?query>)來察看(或是回報)我們的錯誤回報資料庫。

也可以使用send-pr(1)這個指令透過電子郵件來回報問題、要求變更。或者是經由網頁介面的PR (<http://www.FreeBSD.org/send-pr.html>)來送出問題回報。

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

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/articles/problem-reports/article.html](http://www.FreeBSD.org/doc/zh_TW.Big5/articles/problem-reports/article.html))，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。

## 15. 還有其他有關FreeBSD的資訊嗎？

詳見FreeBSD (<http://www.FreeBSD.org>) 網站上的文件 (<http://www.FreeBSD.org/docs.html>) 列表。

## 章2. 文件與技術支援

### 1. 關於FreeBSD 有哪些好書可以推薦閱讀的嗎？

FreeBSD 文件計畫已陸續發表了相當廣泛範圍的文件，可在<http://www.FreeBSD.org/docs.html> 取得。另外，FreeBSD 本身的manual(一般通稱的man)、doc也如同套件軟體一樣，可以輕鬆地裝在您系統上。

此外，也建議參閱本份FAQ 最後所列的參考書目表(Bibliography)與FreeBSD 使用手冊。

### 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，放在FTP 上面，可以從各FreeBSD FTP 站的/pub/FreeBSD/doc/ (<ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/>) 目錄內找到你要的。

文件的分類方面主要是一些不同性質所組成：

- 文件名稱，比如：faq(常見問答集)或是handbook (FreeBSD 使用手冊) 等等。
- 各國翻譯的文件：這主要是由locale 名稱來決定的(不清楚的話，可參考您的FreeBSD 作業系統上的/usr/share/locale) 目前文件總共有下列幾種語言(及編碼)有翻譯：

Locale 名稱	說明(所代表的語系、編碼)
en_US.ISO8859-1	美式英文(US English)
de_DE.ISO8859-1	德文(German)
es_ES.ISO8859-1	西班牙文(Spanish)
fr_FR.ISO8859-1	法文(French)
it_IT.ISO8859-15	義大利文(Italian)
ja_JP.eucJP	日文(Japanese，編碼方式：EUC)
ru_RU.KOI8-R	俄文(Russian，編碼方式：KOI8-R)
zh_CN.GB2312	簡體中文(Simplified Chinese，編碼方式：GB2312)
zh_TW.Big5	正體中文(Traditional Chinese，編碼方式：Big5)

**注：** 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：每份文件都以各種不同格式儲存，每種格式都各有好壞，有些格式適合線上閱讀，而有些則適合列印出美觀的文件。我們都提供這些不同格式的文件，來確保無論是螢幕上、列印紙本，每個人都可以正常地閱讀內容，目前可供使用的格式如下：

格式	說明
html-split	章節模式
html	完整模式
pdb	Palm Pilot 資料格式，使用iSilo ( <a href="http://www.iSilo.com/">http://www.iSilo.com/</a> ) 程式來閱讀

格式	說明
pdf	Adobe's PDF 格式
ps	PostScript 格式
rtf	Microsoft's RTF格式。
txt	純文字(ASCII)

注釋:

- a. 當使用MS Word 來開啓RTF 格式的話，頁數顯示並不會自動更新。(在開啓文件後，要按**CTRL+A**，**CTRL+END**，**F9**，這樣子才會更新頁數的顯示。)

· 文件的壓縮、打包方式：目前有三種方式：

1. 當採用章節模式(html-split)，章節模式所產生的各檔案會先使用tar(1) 來壓縮。檔名結尾有.tar 的檔案就是tar 格式。接著，會再以下列方式再壓縮。
2. 其他格式的檔案都會是單一檔案，檔名通常會是：book. 格式(舉例：book.pdb，book.html 等等..後面通常加上『.格式』)。

而這些檔案會分別以兩種壓縮型態進行壓縮，而存成兩種壓縮型態。

格式	說明
zip	Zip 格式，若要在FreeBSD 上解壓zip 檔，則必須先安裝chinese/unzip 或archivers/unzip 。
bz2	BZip2 格式，雖然不如zip 格式的廣泛使用，但是好處在於可壓縮成更小的檔案。要解壓bz2 格式的話，需先安裝archivers/bzip2 。

所以像是Handbook 的PostScript 版格式，會以BZip2 格式壓縮，存放在handbook/ 目錄內，檔名就是book.ps.bz2 。

選擇想要下載的文件格式與壓縮型態之後，則要決定是否以FreeBSD 套件(package) 型態來下載。

下載、安裝『package』的好處在於：可以透過一般FreeBSD 套件管理方式來進行管理，比如pkg\_add(1) 及pkg\_delete(1) 。

若決定好要下載、安裝『package』的話，必須要確認所要下載的檔名。文件計畫的套件(package)通常是放在是packages 的目錄內，每個文件計畫的套件檔名通常是：文件名稱.語系.編碼.格式.tgz 。

舉個例子，英文版的FAQ (格式選擇PDF)在package 就叫做faq.en\_US.ISO8859-1.pdf.tgz 。

再舉個例子，中文版的FAQ (格式選擇PDF)在package 就叫做faq.zh\_TW.Big5.pdf.tgz 。

知道這點之後，就可以用下面指令來安裝中文版FAQ 套件：

```
# pkg_add ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/packages/faq.zh_TW.Big5.pdf.tgz
```

完成之後，可以用pkg\_info(1) 來找出檔案裝在哪邊：

```
# pkg_info -f faq.zh_TW.Big5.pdf
Information for faq.zh_TW.Big5.pdf:
```

```
Packing list:
```

```

Package name: faq.zh_TW.Big5.pdf
CWD to /usr/share/doc/zh_TW.Big5/books/faq
File: book.pdf
CWD to .
File: +COMMENT (ignored)
File: +DESC (ignored)

```

如同您所看到的book.pdf 會被安裝到/usr/share/doc/zh\_TW.Big5/books/faq 內。  
若不想用package 方式安裝，那麼就需手動下載、解壓縮、複製到你想要擺放的位置去。

舉例，章節模式(split HTML)版的英文FAQ (壓縮為bzip2(1))會放在doc/en\_US.ISO8859-1/books/faq/book.html-split.tar.bz2 要下載、解壓的話，則要打：

```

# fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/en_US.ISO8859-1/books/faq/book.html-split.tar.bz2
# bzip2 -d book.html-split.tar.bz2
# tar xvf book.html-split.tar

```

這時你會看到一堆.html 的檔案，主要的目錄檔為index.html 內含主目錄及連結到其他文件。(若有需要的話，也可以複製或搬移這些檔案到同一目錄下)

### 3. 哪裡有關於FreeBSD 的郵遞論壇(mailing lists)呢？

這個問題，可以從FreeBSD 使用手冊上面的郵遞論壇(mailing-lists) ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/eresources.html#ERESOURCES-MAIL](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/eresources.html#ERESOURCES-MAIL)) 部分獲得答案。

### 4. 有哪些可以使用的FreeBSD 新聞群組(news groups)呢？

這答案可以從FreeBSD 使用手冊上面的新聞群組(newsgroups) ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/eresources-news.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/eresources-news.html)) 部分獲得答案。

### 5. 在文件中常看到類似"pf(4)"、"pfctl(8)" 等這些字樣代表什麼意思呢？

這表示man 的章節，系統上一般有八個章節：

- 1->general commands
- 2->system calls and error numbers
- 3->C libraries
- 4->devices and device drivers
- 5->file formats
- 6->games
- 7->miscellaneous information pages
- 8->system maintenance and operation commands

比如：pf(4) 就是指man 4 pf。

## 6. 有哪些FreeBSD IRC (Internet Relay Chat)頻道呢？

有的，大部分的IRC 主機都有FreeBSD 聊天頻道：

- EFNet (<http://www.efnet.org/index.php>) 的#FreeBSD 頻道是個FreeBSD 論壇，但可不適合那些想不勞而獲或者搬救兵用的。這裡是聊天用的頻道，話題範圍甚至涉及『性、運動、核武』等，請注意：我們已經警告過你了！本頻道可經由irc.chat.org 進入。
- EFNet (<http://www.efnet.org/index.php>) 的#FreeBSDhelp 頻道乃是給FreeBSD 使用者之間交流的，來這裡提問會比#FreeBSD 好一些，當然請不要一股腦隨便亂問。
- DALNET (<http://www.dal.net/>) 的#FreeBSD 頻道，可由irc.dal.net(位於美國)及irc.eu.dal.net(位於歐洲)進入。
- DALNET (<http://www.dal.net/>) 的#FreeBSDhelp 頻道，可由irc.dal.net(位於美國)及irc.eu.dal.net(位於歐洲)進入。
- UNDERNET (<http://www.undernet.org/>) 的#FreeBSD 頻道，可由us.undernet.org(位於美國)及eu.undernet.org(位於歐洲)進入。由於這是個輔助新手用的頻道，請記得閱讀別人向你提及的連結或檔案。
- RUSNET (<http://www.rusnet.org.ru/>) 的#FreeBSD 頻道是俄語國家的FreeBSD 使用者頻道。這裡同時也是一般交流的討論好去處。
- freebsd-gnome ([http://freenode.net/irc\\_servers.shtml](http://freenode.net/irc_servers.shtml)) 的#FreeBSD 頻道，可由irc.freenode.net 進入，這是Gnome 的FreeBSD 使用者頻道。
- freenode ([http://freenode.net/irc\\_servers.shtml](http://freenode.net/irc_servers.shtml)) 的#bsdchat 頻道，可由irc.freenode.net 進入，這是台灣的FreeBSD 使用者頻道。(UTF-8 編碼)

上述每個頻道都不一樣，風格迥異而各具特色，且並沒有相連，因此，你得多方嘗試才能找到適合自己的頻道。而有些地方與所有的IRC 文化類似，就是請注意自己言行是否恰當，另外可能跟頻道內一些年輕/老一輩的會有些代溝需要適應，總之請多保持禮貌。

## 7. 可以從哪邊獲得FreeBSD 的教育課程訓練及技術支援呢？

DaemonNews 有專門提供FreeBSD 的教育課程訓練及技術支援。詳情請到BSD Mall (<http://www.bsdmall.com/>) 察看，謝謝。

FreeBSD Mall 有提供BSD 技術支援付費服務，詳情請到FreeBSD Mall (<http://www.freebsdmall.com/>) 察看，謝謝。

其他任何有提供教育課程訓練及技術支援的組織、單位，若也想列表於此的話，請與FreeBSD documentation project 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc>) 聯絡，謝謝。

## 章3. 安裝

1. 若要用軟碟片開機來安裝FreeBSD 的話，要下載哪些檔案呢？

FreeBSD 4.X 的話，需要兩個image 檔：floppies/kernel.flp 及floppies/mfsroot.flp。image 檔必須用工具像是fdimage 或dd(1) 來傳送到磁片上。若是在FreeBSD 5.3 (及之後版本)有重新規劃開機片架構，所以要抓的是floppies/boot.flp 以及floppies/kernX 檔案(目前X 為1 跟2 兩個，加上floppies/boot.flp，總共是3 個檔案)。

若想自己下載distributions 的話(比如以MS-DOS® 檔案系統格式安裝)，以下是建議要抓的distributions：

- base/ (4.X 版本則為bin/)
- manpages/
- compat\*/
- doc/
- src/ssys.\*

完整安裝步驟以及大部分的安裝問題，請參閱FreeBSD 使用手冊的安裝FreeBSD ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/install.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/install.html)) 章節

2. 若磁片裝不下image 檔的話，該怎麼辦呢？

一張3.5 英吋(1.44MB) 的磁碟片是可以裝上1474560 bytes 的資料，而開機片的image 檔案大小實際上也是1474560 bytes。

在製作開機片時，常見錯誤有：

- 使用FTP 來下載檔案時，未選擇binary 傳輸模式來下載。

有些FTP client 端程式，是預設將傳輸模式設定為ascii 模式，而且會修改接收到的檔案行尾字串為client 端的作業系統方式，比如newline(UNIX格式) 到了作業系統為Windows 的client 端會被改為CR-LF(MS-DOS格式)，這會使得image 檔本身遭到修改而無法正常使用。因此，如果下載的image 檔案大小若與FTP 主機上面的檔案『不一致』的話，請重新使用binary 傳輸模式下載即可。

FTP 指令: 進入FTP 之後，打binary 指令，即可切換到binary 傳輸模式，然後再下載相關image 檔案。

- 直接用MS-DOS 的copy 指令(或類似的GUI 程式、或是視窗上直接複製)來複製開機用的image 檔到磁片上。

不可以用像是copy 這類程式直接將image 檔複製到磁片上，因為image 檔本身包含了完整的磁軌資料，所以不能單純用複製方式，而必須使用低階工具程式(像是fdimage 或rawrite)，以“raw” 方式傳送到磁片上。(這部分可參閱FreeBSD 使用手冊上的安裝FreeBSD ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/install.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/install.html)))

3. 可以在哪邊找到安裝FreeBSD 的解說步驟呢？

安裝步驟的解說，請參閱FreeBSD 使用手冊上的安裝FreeBSD

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/install.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/install.html)) 章節部分。

#### 4. 要跑FreeBSD 需要什麼的配備呢？

FreeBSD 4.X 之前的版本，硬體需求為386 或更高級的PC，記憶體(RAM)至少要5 MB 或更多，硬碟空間至少要60 MB 或更多。不過，FreeBSD 『系統安裝程式』的記憶體(RAM)需求為至少16 MB。

從FreeBSD 5.X 起，硬體需求為486 或更高級的PC，記憶體(RAM)至少要24 MB 或更多，硬碟空間至少要150 MB 或更多。

FreeBSD 的所有版本都可以只用低階的MDA 規格顯示卡，不過...要跑X11R6 視窗的話，還是至少用VGA 或更好規格的顯示卡來用吧。

這部分也可參閱μÚ 4 章。

#### 5. 我電腦RAM 只有4MB 而已，可以裝FreeBSD 嗎？

安裝FreeBSD 4.X 的記憶體需求為至少5 MB，而安裝FreeBSD 5.X (含之後版本) 則是至少要8 MB。

在5.X 之前的所有FreeBSD 版本，都可以只用4 MB 的記憶體來『運作』，不過，前面那節我們說過了『系統安裝程式』的話，則無法只用4 MB 的記憶體來執行。因此，你可以先在『系統安裝程式』這步驟之前，先將記憶體加到16 MB 以上，安裝完FreeBSD 之後，就可以把多餘的記憶體拿下來。或者是，先把要安裝的硬碟拿到有足夠記憶體的機器上先裝好，然後再把硬碟放回原機器。

此外，只用4 MB 的記憶體來運作的話，必須要自製kernel(拿掉不必要的以及犧牲一些東西)。也有人成功只用2 MB 的記憶體以FreeBSD 開機(雖然這樣的系統幾乎等於廢了..)

#### 6. 要怎樣才能自行打造專用的開機、安裝磁片呢？

目前，還沒有辦法『只』自製專用的開機、安裝磁片。必須透過自行打造完整作業系統的release(發行)，這樣裡面才會包括自己的開機、安裝磁片。

若想自行打造、發行(release)一個完整的作業系統，請參閱這篇Release Engineering ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/articles/releng/article.html](http://www.FreeBSD.org/doc/zh_TW.Big5/articles/releng/article.html)) 文章。

#### 7. Windows 可以與FreeBSD 共存於電腦上嗎？

先裝Windows 再裝FreeBSD。那麼FreeBSD 多重開機管理員(boot manager)就會出現選單讓你選擇要以Windows 或FreeBSD 來開機。不過，若你是先裝FreeBSD 再裝Windows 的話，那麼Windows 將會不問先宰，把FreeBSD 的多重開機管理員(boot manager)蓋掉，當你遇上這種情況時，請參考下一節說明。

#### 8. 嗚.. Windows 把我的多重開機管理員(boot manager)拿掉了！要怎麼救回來呢？

可以用以下三種方式之一，來救回你的FreeBSD 多重開機管理員(boot manager)：

- 可以從各FreeBSD FTP 站的/pub/FreeBSD/tools/ (<ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/>) 找到bootinst.exe 及boot.bin 這兩個檔，以binary 傳輸模式下載後，複製到磁片上，再用DOS 開機片開機，接著打類似下面的指令：

```
> bootinst.exe boot.bin
```

這樣，多重開機管理員(boot manager)就會重裝完畢了。

- 用FreeBSD 開機片開機，然後選單那邊選Custom installation(自訂安裝)，再選Partition，接著選擇你要裝多重開機管理員(boot manager)的硬碟(通常是第一顆)，然後會出現partition editor 的畫面，這時請不要做任何修改，直接按W 儲存，這時程式就會問是否要確定Write，最後出現Boot Manager 選擇畫面，記得要

選“Boot Manager”，這樣就會重新將多重開機管理員(*boot manager*)安裝到硬碟上。現在，就大功告成可以離開安裝選單並重開機了。

- 用FreeBSD開機片或是開機光碟開機，然後選單那邊選“Fixit”，或是以Fixit開機片或是光碟安裝的第二片(選擇“live”*filesystem*選項)然後就會進入fixit shell了，接著打下列指令：

```
Fixit# fdisk -B -b /boot/boot0 bootdevice
```

請將上面的*bootdevice*修改為您實際的開機硬碟代號比如ad0(第一顆IDE硬碟)，或是ad4(first IDE disk on auxiliary controller), da0(第一顆SCSI硬碟)等等。

### 9. IBM Thinkpad A系列、T系列或X系列的筆記型電腦裝完FreeBSD之後重開機，就當了，該怎麼辦呢？

(本問題主要是發生在2000~2001四月間時)這些IBM機器上的BIOS早期版本有個臭蟲(bug)會把FreeBSD分割區誤認為是FAT格式分割區，然後當BIOS試著偵測FreeBSD分割區時，就會當了。

根據IBM方面的說法<sup>1</sup>，以下型號/BIOS版本的機種，已經都有修正：

型號	BIOS 版本
T20	IYET49WW(含之後)
T21	KZET22WW(含之後)
A20p	IVET62WW(含之後)
A20m	IWET54WW(含之後)
A21p	KYET27WW(含之後)
A21m	KXET24WW(含之後)
A21e	KUET30WW

這些後期的IBM BIOS修訂版大多已經修正此一臭蟲。Jacques Vidrine寫給FreeBSD laptop computer 郵遞論壇(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile>)的這封信(<http://docs.FreeBSD.org/cgi/mid.cgi?20010427133759.A71732>)上面說明了若你新的IBM筆記型電腦無法順利以FreeBSD開機的解法步驟(假設可以升級或降級BIOS版本的話)。

如果機器用的是較古早版本的BIOS，而且不適合更新BIOS的話，那麼以下我們會介紹暫時解法，教你如何修改FreeBSD分割區所使用的ID，並安裝相關修補程式。

First, you will need to restore the machine to a state where it can get through its self-test screen. Doing this requires powering up the machine without letting it find a FreeBSD partition on its primary disk. One way is to remove the hard disk and temporarily move it to an older ThinkPad (such as a ThinkPad 600) or a desktop PC with an appropriate conversion cable. Once it is there, you can delete the FreeBSD partition and move the hard disk back. The ThinkPad should now be in a bootable state again.

With the machine functional again, you can use the workaround procedure described here to get a working FreeBSD installation.

1. 從<http://people.FreeBSD.org/~bmah/ThinkPad/>來下載boot1及boot2這兩個檔。把這兩個檔案放在磁片、光碟或其他硬碟上。
2. 以一般安裝FreeBSD裝到ThinkPad上，記得『*不要用*』Dangerously Dedicated模式。此外，『*不要*』在完成安裝之後就重開機。
3. 進入shell：(按ALT+F4)切換到“Emergency Holographic Shell”或是選單上面選擇“fixit” shell。

4. 用fdisk(8)把FreeBSD分割區ID由165改為166(OpenBSD所使用的ID)。
5. 把剛剛提的boot1跟boot2這兩個檔案放到目前的硬碟檔案系統上。
6. 以disklabel(8)把boot1及boot2存到你的FreeBSD slice上面。

```
# disklabel -B -b boot1 -s boot2 ad0sn
```

「n」是你裝FreeBSD的slice，請將「n」改為符合你系統現況的slice。

7. 重開機，在boot prompt會看到有OpenBSD的選項，選它，這樣就會以FreeBSD開機。

另外，至於如何讓OpenBSD及FreeBSD並存在同一台IBM ThinkPad上...這個問題就交給各位看倌們去嘗試了:p

#### 10. 有壞軌的硬碟可以拿來裝FreeBSD嘛？

(如果很堅持的話)也是可以，不過這想法顯然不太明智。:(

如果在一般較新的IDE硬碟上看到有壞軌，很有可能代表：這顆硬碟即將掛點了。(因為目前所有較新的IDE硬碟，內部都有自動remapping壞軌的能力。如果看到有壞軌，則表示它內部自動remapping功能失效，無法處理壞軌，也就是說這顆硬碟已經是嚴重損壞程度了。)我們建議買顆新硬碟比較乾脆些唷。

如果是SCSI硬碟有壞軌的話，請試著參考這個解法。

#### 11. 用安裝磁片開機時，卻有些怪現象發生！這是什麼情況呢？

若看到一些怪異現象，像是開機片開機開到一半就當了，磁碟機完全沒任何動作、或是不斷反覆重開機，請先檢查以下幾個線索：

1. 請確定是否為全新、沒有磁軌錯誤的磁片？(最好使用新買的，而非雜誌、書本附贈的，甚至還放在床底下三年了...=\_="")
2. 請確定是否有用binary(或image)傳輸模式來下載image檔？(不用覺得不好意思，即使是我們也曾意外以ASCII傳輸模式來下載binary檔案！)
3. 若你是Windows 95/98/ME/NT/2000/XP/2003來下載、製作開機磁片的話，請確定是否有在DOS模式使用fdimage或rawrite這兩個工具程式？剛講的這些作業系統，都會影響程式去直接寫入硬體，像是製作開機片之類的動作。有時候，在GUI介面上的DOS shell也可能會發生這樣的問題。

此外，直接透過Netscape®瀏覽器下載image檔的話，也有類似現象。所以，如果可以的話，請改用其他可以調整設定的FTP client端程式來進行下載。(當然，要記得調binary傳輸模式)

#### 12. 用光碟開機片來安裝，但光碟開機後，安裝程式說找不到光碟...這是怎麼了？

通常問題在於光碟機設定錯誤。目前很多電腦的出廠標準配備都有光碟機，並且會預先設定為IDE通道上面Secondary的Slave設備，而Secondary上面的卻沒有Master設備。以ATAPI的規格而言，這是錯誤的設定，然而Windows的作法是不理會這些規格上的設定問題，而且開機時BIOS偵測也會略過這點。這也就是為什麼BIOS可以看到光碟，並且可用光碟開機，但FreeBSD無法正常抓到光碟以順利進行安裝。

解法：重新設定系統，讓光碟成為它所連接那條IDE通道的Master，或者只有一條IDE通道的話，那就讓光碟機成為Slave，當然該IDE通道上至少要有Master設備。

### 13. 我可以用PLIP(Parallel Line IP)方式來安裝FreeBSD 到筆記型電腦上嗎？

可以，用一條普通的Lapl原因線就可以囉。若有這方面需求的話，請參閱FreeBSD 使用手冊中的PLIP 章節 ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/network-plip.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/network-plip.html)) 的細部設定

### 14. 該用哪一種硬碟設定參數(geometry)呢？

**注：**這裡的『硬碟設定參數(geometry)』，指的是硬碟上的「cylinders」、「heads」、「sectors/track」這三個設定參數。接下來的文章內，為了方便介紹，將簡稱為『C/H/S』。這些設定參數是讓PC 上的BIOS 能順利地正常判別硬碟，與硬碟本身讀寫的重要因素。

對剛接手的系統管理者新手而言，這些設定參數常造成一些困擾。首先，SCSI 硬碟上的*physical geometry* 跟FreeBSD 上的disk blocks 是完全無關的。事實上，就硬碟上磁區密度的變化而言，並沒有所謂『physical geometry』這種東西。硬碟製造商所說的『physical geometry』通常是指：硬碟上所使用最小空間來存放資料的設定參數(geometry)。以IDE 硬碟而言，FreeBSD 用以存取硬碟設定的方式是C/H/S，然而，目前市面上的硬碟早就在內部運作時，就自動轉換為block 方式了。

真正關鍵的地方，其實是在於『logical geometry』—這是BIOS 偵測硬碟時所得到的設定，並且用來決定硬碟存取方式。由於FreeBSD 是採用BIOS 的偵測設定值，所以如何來讓BIOS 偵測到的設定值保持正確，是十分重要。尤其是同一顆硬碟上有多個作業系統的情況，它們都必須採用一致的硬碟設定參數(geometry)，否則就會有開機進不去作業系統的嚴重問題了。

以SCSI 硬碟而言，硬碟設定參數(geometry)是由SCSI 卡上的extended translation(通常指的是有標示“support >1GB” (支援1GB 以上容量，或類似名詞) 支援與否來作決定。如果不支援，那麼就會採用*N cylinders*、64 heads、32 sectors/track 作為硬碟設定參數(geometry)，這裡講的『*N*』是指硬碟的容量(單位：MB)。舉個例子來說，一顆2GB 硬碟應該是2048 cylinders、64 heads、32 sectors/track。

如果該SCSI 有支援使用extended translation 的話，(通常這個方式在MS-DOS 使用上有某些限制)，並且硬碟容量大於1GB，那麼硬碟設定參數(geometry)就會使用像是：*M cylinders*、255 heads、每磁軌63 sectors(請注意：不是『64』哦)，這裡講的『*M*』是指硬碟的容量(單位：MB)再除以7.844238 所得出的數值喔！所以，這個例子的話，同樣是2GB 硬碟應該是261 cylinders、255 heads、每磁軌63 sectors。

若對上面講的不瞭解，或是FreeBSD 在安裝時所偵測到的硬碟設定參數(geometry) 有問題的話，最簡單的解法通常是在硬碟上建立一塊小小的DOS 分割區(partition)。這樣一來，就可以偵測到正確的硬碟設定參數了，而且，如果不想繼續留著那小塊DOS 分割區的話，可以隨時用partition editor 來拿掉它。或者把它留著當作網路卡驅動程式使用，或隨你高興怎麼用它。

此外呢，有個免費好用的工具程式叫做『*pfdisk.exe*』，這個程式放在各FreeBSD FTP 站或光碟的tools 目錄下，它可以用來找出硬碟上其他作業系統所使用的硬碟設定參數，然後就可以在partition editor 內輸入剛剛找到的那些設定參數就可以了。

### 15. 分割磁碟機時有任何限制嗎？

有，你必須確認你的root 分割區是在1024 cylinders 之內，讓BIOS 可以從其中啓動核心。(注意：這是PC 的BIOS 功能限制，而不是FreeBSD 的)

以SCSI 硬碟而言，通常是把root (/) 分割區放到硬碟最前面的1024MB (如果有支援extended translation 的話，那麼是最前面的4096MB —這點請參考上一小節)。而IDE 硬碟的話，相對應的則是504MB。

### 16. 可以使用哪些磁碟管理程式(disk managers)呢？

FreeBSD 可以用Ontrack Disk Manager 並且運作正常，至於其他的disk manager 則不在正式支援之列。

若整顆硬碟只裝FreeBSD，那麼就不用再裝disk manager了。只要把硬碟設定為BIOS所能抓到的最大空間，那麼FreeBSD就可算出實際上可使用的空間了。如果，正在使用的是古早MFM控制卡的舊式硬碟，那麼就需要在FreeBSD內作cylinders相關設定了。

如果想在磁碟上使用FreeBSD和另外的作業系統，也可以不裝disk manager，只要確定FreeBSD的啟動分割區跟其他作業系統的slice都位於開始的1024 cylinders內就可以了。如果你相當地高明的話，一個20MB的啟動分割區應該就夠用了。

### 17. FreeBSD 安裝完畢後重開機，但是電腦卻說Missing Operating System這是怎麼了？

通常原因出在FreeBSD及DOS或其他作業系統在硬碟的設定參數(geometry)上的規劃有相衝。解法是重裝，但是請照上述的相關章節步驟來做。

### 18. 為什麼機器上多重開機管理員(boot manager)出現了F? 這個選單畫面，但卻不會自動跳過而繼續開機呢？

這個症狀是本文上面所提的另外一個問題了，原因在於BIOS上跟FreeBSD上面兩邊的硬碟的設定參數(geometry)並不一致。若你硬碟或BIOS支援cylinder translation(通常會被標為“support >1GB(支援1 GB以上容量)”)的話，試試看更改相關設定，並重裝FreeBSD。

### 19. 安裝FreeBSD時，需要安裝完整的sources嗎？

一般來說，這是不用的。然而，我們強烈建議您的source至少要裝base(包含了本文中所提的一些檔案)、以及sys(kernel的source檔)。雖然，作業系統本身運是不需要裝source檔，但是config(8)這個kernel設定程式則需要src。若沒裝kernel的source檔，仍然可以用其他地方透過NFS掛載的唯讀檔案系統，來編譯程式。但由於kernel-source本身的限制，我們建議不要直接mount在/usr/src，最好是用symbolic link(參閱ln(1))將掛載的路徑，設定連結到/usr/src目錄。

在機器上直接裝有source並且瞭解相關編譯過程，這樣子日後升級FreeBSD會來得方便多。

若忘了裝相關source的話，可以事後用sysinstall(FreeBSD 5.2之前版本則是/stand/sysinstall)來補裝，選單：Configure → Distributions → src。

### 20. 需要重新build kernel嗎？

在很久很久以前，原本重新build kernel在FreeBSD安裝過程中，是絕對必需的步驟之一。但目前早就不用這麼麻煩了，目前主要常見的版本都使用更友善的kernel設定指令。FreeBSD 4.X(含之前版本)，在FreeBSD啟動提示號(boot:)時，使用“-c”flag就會進入設定畫面，來對kernel作常見的ISA卡細節設定。而FreeBSD 5.X(含之後版本)的話，則是以更具彈性的“hints”設定方式。

如果想更節省RAM的使用、縮短開機流程，那麼建議：新的kernel設定檔只要包含你需要的驅動程式，然後重新編譯、安裝kernel並重開機。然而呢，這點對大多數的系統來說，這不一定是必要的。

### 21. 密碼編碼該採用DES、Blowfish或MD5的哪一種？該怎麼設定呢？

FreeBSD預設的密碼編碼方式是採MD5。就密碼編碼方式而言，MD5方式一般被視為比傳統UNIX的DES方式較為安全。然而，若有需要在使用舊環境(較不安全的密碼編碼方式的)，DES密碼編碼方式一樣可以使用。(若使用sysinstall時有選擇“crypto”套件，或是從source內編譯安裝)有裝crypto的話，crypto libraries也支援更安全的Blowfish編碼方式。

密碼編碼的方式是由/etc/login.conf 內的“passwd\_format”欄位來決定的。該欄設定值，(若有裝crypto的話)可以是“des”或“blf”或是原本的“md5”。詳情請參閱login.conf(5)說明。

## 22. 為什麼開機磁片開到一半，就出現Probing Devices...的畫面訊息，然後就停住了？

若機器上有裝IDE 介面的Iomega Zip® 或是Jaz®的話，因為這些設備可能跟開機片有相衝，請先拿掉這些設備再重試。當整個作業系統裝好後，就可以把這些設備接回去使用了。希望這點在日後release 的FreeBSD 可以獲得徹底解決。

## 23. 系統裝完並重開機後，為什麼卻出現panic: can't mount root?

這個錯誤是因為啓動磁區跟kernel 兩者對磁碟裝置的認知不相同。通常這問題會發生在有兩顆IDE 硬碟的系統，比如：硬碟Jumper 設定為Master、或是兩條IDE 排線各只連接一顆硬碟，但是裝FreeBSD 那顆開機硬碟，卻接在Secondary IDE 排線上。如此一來，在開機時當kernel 指定第二個IDE 控制器的第一個磁碟機ad2，啓動磁區卻認為系統是裝在ad0(BIOS偵測的第二顆硬碟)！偵測完裝置後，kernel 試著把啓動磁區所認為的開機硬碟(ad0) mount 起來，事實上應該是ad2 才對，所以就出現上面的錯誤訊息了。

解法如下，請選擇其中一種方式就好：

1. 重開機並在出現『Booting kernel in 10 seconds; hit [Enter] to interrupt』提示的時候，按下**Enter**鍵。這樣子就會進入boot loader 畫面：

請輸入 `set root_disk_unit="disk_number"` 這裡的『disk\_number』請依據情況換成相關代號：若FreeBSD 硬碟是設定裝在Primary IDE 接線的Master，就設為0；若是Primary IDE 接線的Slave，就設為1；若是Secondary IDE 接線的Master，就設為2；若是Secondary IDE 接線的Slave，就設為3。

接著請再輸入boot，然後系統應該就可以正常開機了。

若要每次開機都自動設定，而不必每次都打一次，那麼就在/boot/loader.conf.local 檔案內加上 `root_disk_unit="disk_number"` 這行(當然，『disk\_number』要改成相關代號)

2. 把FreeBSD 硬碟改接到Primary IDE 接線上，如此一來就可順利使用。

## 24. 記憶體最大限制為多少？

一般i386 機器上最多可支援到4 GB(gigabytes)。而自FreeBSD 4.9 及5.1 可以開始使用pae(4) 來支援更多的記憶體。有關這點，需要在kernel 設定檔內加入下列內容並重新編譯kernel，才能使用PAE。

```
options          PAE
```

FreeBSD 在pc98 機器上最多則只支援4 GB，而且不能使用PAE。在Alpha 機器上，記憶體限制方面要視所使用的型號支援程度而定，這點請參閱所使用的Alpha 硬體規格表。至於其他架構的FreeBSD 理論上最多可使用的記憶體，有較多的限制。

## 25. 檔案系統最大限制為多少？

就檔案系統方面，理論上的限制是最多到8TB(2G blocks)，或是使用內定8K block 大小時，限制是16TB。實際上，目前軟體使用上限制只能用到1 TB，然而如果是有另行改造過檔案系統，那麼達到4TB 的目標是可行的(也有人成功過)。

單一檔案的大小方面，假如block 以4K 作規劃的話，則最大是趨近1G blocks(4TB)。

表格3-1. 檔案大小的最大限制

檔案系統block 大小	works	should work
4K	4T-1	>4T
8K	>32G	32T-1
16K	>128G	32T-1
32K	>512G	64T-1
64K	>2048G	128T-1

When the fs block size is 4K, triple indirect blocks work and everything should be limited by the maximum fs block number that can be represented using triple indirect blocks (approx.  $1K^3 + 1K^2 + 1K$ ), but everything is limited by a (wrong) limit of 1G-1 on fs block numbers. The limit on fs block numbers should be 2G-1. There are some bugs for fs block numbers near 2G-1, but such block numbers are unreachable when the fs block size is 4K.

block 大小如果是8K 或更大，檔案系統block 數目會被限制在2G-1，但實際上應該說限制是1G-1 才對，因為採用2G-1 block 的檔案系統會導致一些問題。

26. 為何在啓動新的kernel 時，看到archsw.readin.failed 錯誤訊息？

原因出在你的world 以及kernel 並不同步，舉例：kernel 用4.11，而world 卻是4.8，這樣是會有問題的。請再次確認，是否有以make buildworld 及make buildkernel 來正常更新kernel。

在啓動loader 之前，會看到"|" 這個符號在轉動，這時可以按任何鍵中斷，然後再指定要載入哪個kernel 來開機。

27. “security profiles” 是指什麼？

A “security profile” is a set of configuration options that attempts to achieve the desired ratio of security to convenience by enabling and disabling certain programs and other settings. For full details, see the Security Profile ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/install-post.html#SECURITYPROFILE](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/install-post.html#SECURITYPROFILE)) section of the Handbook’s post-install chapter ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/install-post.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/install-post.html)).

28. 在開機時，選擇使用ACPI 則在安裝過程就掛了，該怎麼辦？

試試看關閉ACPI support。當在載入bootloader時，按下空白鍵。系統會顯示

```
OK
```

這時輸入

```
unset acpi_load
```

接著打

```
boot
```

以繼續開機，這樣子應該就可以了。

## 注

1. 一封來自 Keith Frechette 的e-mail <kfrechet@us.ibm.com>。

## 章4. 硬體支援方面

### 4.1. 一般問題

1. 我想組裝自己的FreeBSD 機器，有哪些型號、品牌、規格是支援程度最好的呢？

有關這點，在FreeBSD 討論區上時常有人討論。雖然硬體汰換速度很快，可能隨時有新規格、新產品出現，然而這些都在我們意料之中，我們仍然強烈建議：在詢問有關最新規格硬體的支援問題之前，請先參閱FreeBSD 9.1 (<http://www.FreeBSD.org/releases/9.1R/hardware.html>) 或8.4 (<http://www.FreeBSD.org/releases/8.4R/hardware.html>)的支援硬體列表，或是搜尋討論區的舊文章 (<http://www.FreeBSD.org/search/#mailinglists>)，也許，上週才剛恰巧有人討論過你所要問的硬體。

如果要找有關筆記型電腦方面，請到FreeBSD-mobile 筆記型電腦討論區。不然，就到FreeBSD-questions 討論區，或是特定硬體規格(比如pc98, Alpha)的專屬討論區吧。

### 4.2. 硬體架構及CPU

1. FreeBSD 有支援x86 之外的硬體架構平台嗎？

有的，FreeBSD 目前可以在Intel x86 and DEC (現在的HP-Compaq) Alpha 架構上面運作。自FreeBSD 5.0 之後的版本，則可支援AMD64 及Intel EM64T, IA-64 以及Sparc64® 架構。未來平台支援上還會有MIPS® 及PowerPC®, 細節請分別參閱FreeBSD MIPS porting 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mips>) 或FreeBSD PowerPC porting 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ppc>)。一般而言，新的硬體架構平台方面，都是到FreeBSD non-Intel platforms porting 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms>) 討論。

若你機器不是以上架構或是比較奇特的，而想立刻試試看BSD 的魔力，我們建議你可以考慮使用NetBSD (<http://www.netbsd.org/>) 或OpenBSD (<http://www.openbsd.org/>)。

2. FreeBSD 支援CPU 對稱多工處理(SMP, Symmetric Multiprocessing)嗎？

有的。SMP 在FreeBSD 5.2 預設的kernel(*GENERIC*)已有啟動。

在FreeBSD 5.3 要release 時，SMP相關設定也是預設就有啟動。然而，在一些較新型的機器(像是emt64)上卻又有些問題，所以還是決定在相關問題、安全議題未獲解決前，先關閉SMP 的相關啟動。這點，正是FreeBSD 5.4 所優先考慮的方向。

FreeBSD 4.X 的話，預設的kernel 並沒有啟動SMP，因此，必須要把options SMP 加入kernel 設定檔並重新編譯才能啟動。至於還有哪些相關設定要放入kernel 設定檔，請參閱/sys/i386/conf/LINT。

### 4.3. 硬碟、磁帶機以及光碟、DVD、燒錄機

#### 1. FreeBSD 可支援哪些種類的硬碟呢？

FreeBSD 都支援EIDE 及SCSI 介面的硬碟(以及SCSI 卡，請看下一節說明) 以及“Western Digital” 介面的硬碟(MFM、RLL、ESDI，當然包含IDE)，不過有一些少數的ESDI 晶片組的(型號：WD1002/3/6/7) 可能無法正常運作。

#### 2. 支援哪些SCSI 卡、設備呢？

請參閱FreeBSD 的硬體支援表( 9.1 (<http://www.FreeBSD.org/releases/9.1R/hardware.html>) 或8.4 (<http://www.FreeBSD.org/releases/8.4R/hardware.html>))

#### 3. 支援哪些磁帶機呢？

FreeBSD 支援SCSI 及QIC-36 (QIC-02 介面) 規格的磁帶機。同時包含了8-mm (也就是Exabyte) 及DAT 磁帶機。

有些早期版本的8-mm 磁帶機並不是完全相容於SCSI-2 規格，所以可能在FreeBSD 上表現不是很好。

#### 4. FreeBSD 支援磁帶自動換帶機嗎？

FreeBSD 可以用ch(4) 上面所列的機種，搭配chio(1) 指令，來使用SCSI 種類的自動換帶機，細節部分請參閱chio(1) 說明。

If you are not using **AMANDA** or some other product that already understands changers, remember that they only know how to move a tape from one point to another, so you need to keep track of which slot a tape is in, and which slot the tape currently in the drive needs to go back to.

#### 5. FreeBSD 可支援哪些種類的碟機呢？

只要有支援的SCSI 卡上所接的任一SCSI 光碟機都有支援。

此外，也支援下列的光碟機：

- Mitsumi LU002 (8bit), LU005 (16bit) 及FX001D (16bit 2x Speed)
- Sony CDU 31/33A
- Sound Blaster 非SCSI 介面的光碟機
- Matsushita/Panasonic 光碟機
- 相容ATAPI 規格的IDE CDROMs

相對於SCSI 機種而言，其他非SCSI 的光碟機都是比較慢，此外，有些ATAPI 種類的光碟機可能無法順利運作

Daemon News 以及FreeBSD Mall 所發行的正式FreeBSD 光碟以及燒錄用的影像檔(ISO)，都可以直接用於開機光碟使用。

## 6. FreeBSD 支援哪些光碟燒錄機的驅動程式呢？

FreeBSD 支援任何相容ATAPI 標準的IDE CD-R 或CD-RW 光碟燒錄機，細節請參閱burncd(8) 說明。

FreeBSD 也支援任何SCSI CD-R 或CD-RW 光碟燒錄機。請用port 或packag 機制來安裝、使用cdrecord ，並確定您的kernel 內有將pass設備一併編譯在內。(預設的kernel.GENERIC 都會有device pass 這段)

## 7. FreeBSD 支援Iomega Zip 嗎？

FreeBSD 支援外接式的SCSI 及ATAPI(IDE) 介面的Iomega Zip。不過SCSI ZIP 只能被設為SCSI ID 5 或是6 才可以運作，但如果SCSI 卡上的BIOS 支援它，你甚至可以用它來開機。我們不曉得哪一塊卡可以把卡的ID 設在除了0 或1 以外的地方而開機成功，因此，如果想改SCSI ID 的話，請務必參閱該型號的說明手冊。

FreeBSD 同時也支援Parallel Port Zip磁碟機。請檢查kernel 設定檔是否有：scbus0、da0、ppbus0，以及vp0 這些驅動程式(預設的GENERIC kernel 除了vp0 沒包進去，其他三者都會有)。加了這幾個驅動程式之後，Parallel Port Zip 就會成為/dev/da0s4。

這時，就可以用像是mount /dev/da0s4 /mnt 或(DOS 檔案系統)mount\_msdos /dev/da0s4 /mnt 之類的指令來掛載、讀寫。

也可以參閱下面有關隨身磁片部分，以及抽取碟、隨身碟的『格式化』討論的部分

## 8. FreeBSD 有支援Jaz、EZ 及其他類似的隨身磁片嗎？

可以啊，除了IDE 的EZ drive 外，其他的應該都是SCSI 介面，所以在FreeBSD 上都會以SCSI 硬碟來處理。

當然，你必須確定在開機時，這些設備的電源是打開的，以便讓FreeBSD 可以偵測到。

如果在磁碟運中狀態中，要更換磁片的話，記得先看一下mount(8)、umount(8)、以及(SCSI的話)camcontrol(8) 或atacontrol(8) 還有FAQ 後面有關使用抽取碟、隨身碟的討論。

# 4.4. 鍵盤、滑鼠

## 1. FreeBSD 有支援USB 鍵盤嗎？

FreeBSD (尤其是有支援USB keyboards。Enable USB support in /etc/rc.conf.

若有關USB 鍵盤支援而且同時接上AT 跟USB 鍵盤的話，那麼AT 鍵盤會變成/dev/kbd0，而USB 鍵盤則是/dev/kbd1。如果只接USB 鍵盤，那麼它就是/dev/ukbd0 囉。

如果想在console 上使用USB 鍵盤的話，那麼必須設定console 指定用USB 鍵盤。可以在系統開機程序時，加上下列指令：

```
# kbdcontrol -k /dev/kbd1 < /dev/ttyv0 > /dev/null
```

注意：若只有USB 鍵盤的話，也就是/dev/ukbd0，那麼請改用下列指令：

```
# kbdcontrol -k /dev/ukbd0 < /dev/ttyv0 > /dev/null
```

建議：可以把上述指令放入/etc/rc.i386。

設定成功之後，USB 鍵盤不用作任何特別設定，就可以在X 視窗環境上正常運作囉。

USB 鍵盤的熱插拔(Hot-plugging and unplugging)在FreeBSD 可能還無法完全正常運作，建議：在系統開機前就先接上鍵盤，直到關機為止，以避免不必要的困擾。

相關細節請參閱ukbd(4) 的說明。

## 2. 古早的bus 滑鼠，要怎麼設定呢？

FreeBSD 支援一些廠商(像是：Microsoft、Logitech、ATI)所做的bus 及InPort bus 介面的滑鼠。然而，預設的kernel(GENERIC)已經不內含它們的驅動程式。因此，要加入下列到kernel 設定檔並重新編譯、安裝，才能啟用：

```
device mse0 at isa? port 0x23c irq5
```

Bus 滑鼠通常要搭配專用的介面卡才能使用。這些卡可以設定port address 及IRQ 值，這些細節請參閱你的滑鼠說明手冊及mse(4) 說明。

## 3. PS/2 ( “mouse port” 或 “keyboard” )的滑鼠要怎麼設定才好呢？

PS/2 滑鼠都有支援，所需要用到的驅動程式psm 在預設的kernel(GENERIC)已有內含了。

若你自訂的kernel 內漏了psm 的話，那麼就再把下列內容加到kernel 設定檔並重新編譯、安裝：

```
device psm0 at atkbd? irq 12
```

當開機時kernel 有正確偵測到psm0，請務必確認在/dev 內有psm0。如果沒有的話，那麼就用root 來打下列指令來建立吧：

```
# cd /dev; sh MAKEDEV psm0
```

**注：**如果是FreeBSD 5.0-RELEASE(含之後版本)的話，因為採用devfs(5) 機制的因素，所以會自動在/dev 下建立相關設備的節點，因此就可以略過上面這一步。

## 4. 如果不用X Window 環境的話，也可以用滑鼠嗎？

若使用console 的預設驅動程式(也就是syscons(4))，那麼就可以在文字介面的console 上面用滑鼠來剪貼文字了。那麼要啟動moused(8) 並開啓游標顯示，請打下列指令：

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

其中『xxxx』是滑鼠的設備名稱，而『yyyy』則是滑鼠所使用的protocol 種類。目前的moused 可以自動偵測(除了舊式的serial 滑鼠之外)大多數滑鼠所使用的protocol 種類，而不用刻意去指定。『protocol 種類』設定用auto 就會自動偵測了。若自動偵測失敗的話，請參閱moused(8) 裡面的type 那段說明。

如果用的是PS/2 滑鼠，只要把moused\_enable="YES" 加到/etc/rc.conf，這樣每次開機就會自動啓動了。此外，如果要在所有virtual terminals 上也能使用滑鼠，而不限定只有console 的話，那麼請再把allscreens\_flags="-m on" 加到/etc/rc.conf 裡面即可。

`moused` 在執行中的時候，如果要使用滑鼠相關功能，都必須透過`moused` 或其他程式像是X 視窗來進行。請參閱FAQ 中有關『為什麼不能在X 視窗裡使用滑鼠？』以瞭解相關細節。

#### 5. 在文字模式的console 環境要怎麼用滑鼠來剪貼文字呢？

當執行`moused` 後，(參閱前一節) 按住左鍵，接著移動滑鼠來選擇一個區域之後放開，這樣就完成『複製』。要『貼上』的話，按滑鼠中鍵就可以了。要『延伸選取區』的話，按滑鼠右鍵

如果你的滑鼠沒有中鍵，你可以用模擬的方式，或是重新定義滑鼠按鍵的方式，來達成「延伸」的功能。詳情請參閱`moused(8)` 說明。

#### 6. 我滑鼠上面的滾輪、滾輪按鈕，可以在console 上使用嗎？

這個答案嘛...，很不幸地，在大多數的情況下不行。這些有滾輪的滑鼠需要用特殊驅動程式才行，除非，滑鼠驅動程式或使用者自己的應用程式有支援，不然，這些滑鼠只能夠當成是普通的兩鍵或三鍵的滑鼠來用而已。

如果要在X 視窗環境上使用滾輪的話，請參閱 X 視窗上的滾輪使用說明。

#### 7. 要怎麼在筆記型電腦上使用滑鼠、軌跡球、觸控板呢？

請參閱前面的PS/2 滑鼠的問答。

## 4.5. 網路跟serial 設備

### 1. FreeBSD 支援哪些網路卡呢？

請參考FreeBSD 各版本的硬體支援列表。

### 2. 為什麼FreeBSD 找不到PnP(隨插隨用，Plug & Play)規格的modem?

原因在於：需要把modem 的PnP ID 加到serial 驅動程式的PnP ID 表，作法如下：

1. 首先，在kernel 設定檔內加入`controller pnp0`，並重新編譯、安裝kernel，最後重開機就會啟動PnP 支援。
2. 然後，kernel 會把偵測到所有設備上的PnP ID 都列出。這時，修改`/usr/src/sys/isa/sio.c`(大約第752行左右的地方)，可以搜尋`SUP1310` 當關鍵字(位於`sio_ids[]` 表內)，請將剛才kernel 顯示的modem 的PnP ID 複製到相關位置。
3. 這時，再重新編譯、安裝kernel，最後重開機應該就會正確偵測到modem 了。

此外，也可以在開機時以`pnp` 指令來手動設定PnP 設備，來讓kernel 得以正確偵測，舉例：

```
pnp 1 0 enable os irq0 3 drq0 0 port0 0x2f8
```

### 3. FreeBSD 支援像是Winmodems 之類的軟體modem 嗎？

FreeBSD 可以安裝額外的軟體來支援軟體modem。像是comms/ltmdm 可支援常見的Lucent LT 晶片，comms/mwavem 則可支援IBM Thinkpad 600 及700 筆記型電腦上面的modem。

然而，並不能用軟體modem 來安裝FreeBSD，因為：這類軟體必須在作業系統安裝完畢之後，才能安裝。

### 4. 有Broadcom 43xx 無線網卡的原生驅動程式(Native driver)嗎？

沒有，而且也不太可能會有。

Broadcom 拒絕公開有關無線網卡晶片的驅動程式相關說明，主因大概是他們用軟體來控制無線傳輸方式。事實上，因為要能通過美國聯邦電信委員會(FCC)檢磁安規的話，必須確保產品不能讓使用者不能隨意更動相關設定，比如：電磁波頻率、相關模組參數、輸出電源等。但是，如果我們不知道如何去控制晶片的話，那麼撰寫驅動程式之路恐怕不太可行。

### 5. FreeBSD 支援哪些multi-port serial卡呢？

請參閱使用手冊上的安裝篇—其他硬體

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/install.html#INSTALL-MISC](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/install.html#INSTALL-MISC)) 列表。

雖然有些卡是沒牌的(尤其是有標明：相容AST 規格)但也可以正常使用。

至於卡的設定方面，請參閱sio(4) 的說明。

### 6. 在serial console 上要如何才會出現boot: 提示呢？

1. kernel 設定檔加入options COMCONSOLE
2. 建立/boot.config 檔，並且該檔裡面內容只填上-p
3. 把鍵盤從機器上拔掉

細節請看/usr/src/sys/i386/boot/biosboot/README.serial

## 4.6. 音效卡

### 1. FreeBSD 支援哪些音效卡？

FreeBSD 支援各種音效卡，包括了SoundBlaster®、SoundBlaster Pro、SoundBlaster 16、Pro Audio Spectrum 16、AdLib、及Gravis UltraSound sound cards (細節請參閱FreeBSD 發行情報 (<http://www.FreeBSD.org/releases/>) 以及snd(4) 的說明)。此外，對MPU-401 及MIDI 相容規格的也有一定程度的支援，而Microsoft® Sound System 規格也有支援。

**注：**驅動程式僅適用於『音效』部分！除了SoundBlaster 之外，目前音效驅動程式並不支援這些音效卡上的光碟機、SCSI設備或搖桿。SoundBlaster 的SCSI 介面及某些非SCSI 的光碟機是有支援，但無法用來開機。

## 2. pcm(4) 所支援的音效卡沒有聲音，有什麼暫時解決方式嗎？

因為有些像是es1370 晶片的音效卡會在每次開機時把音量調為零。暫時解法是在每次開機時執行下面指令，或是加到/etc/rc.local 內：

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

## 4.7. 其他怪異問題(ACPI、重開機後掛了..等)

### 1. FreeBSD 還支援其他哪些硬體呢？

請參閱使用手冊上的安裝篇—其他硬體

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/install.html#INSTALL-MISC](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/install.html#INSTALL-MISC))。

### 2. FreeBSD 支援筆記型電腦的省電管理功能嗎？

FreeBSD 4.X(含之後版本)在某些機種上都有支援APM。細節請參閱apm(4) 的說明。

FreeBSD 5.X(含之後版本)支援在目前大部分機種上都有的ACPI 功能。細節請參閱acpi(4) 的說明。若機器上同時都有APM 及ACPI 功能的話，我們建議你可以兩者都試試看，看看哪一種比較符合你的需求。

### 3. 該如何關閉ACPI？

把

```
hint.acpi.0.disabled="1"
```

這段加到/boot/device.hints 即可。

### 4. Micron 電腦總是在FreeBSD 啟動時就掛掉，該怎麼辦呢？

有些Micron 主機板上的BIOS 在PCI 方面會有問題，這會導致PCI 設備會被BIOS 偵測為不正確設定，而進入FreeBSD 就掛掉。

暫時解決方式：關閉BIOS 內“Plug and Play Operating System” 的設定。

### 5. 3Com® PCI 介面網路卡無法在Micron 電腦上使用，該怎麼辦？

這問題跟前面的問題因素一樣，總之，就是關閉BIOS 中有關OS PnP 的設定。

暫時解決方式：關閉BIOS 內“Plug and Play Operating System” 的設定。

### 6. 主機板是用華碩(ASUS) K7V，可是用開機片開到一半就當了，怎麼辦呢？

進入BIOS 設定，並關閉“boot virus protection” 設定即可。

**7. PCMCIA 卡無法使用，並出現錯誤訊息 “cbb0: unsupported card type detected.” 該怎麼辦？**

可以試試看改用舊的方式，請先修改kernel 設定檔，拿掉下面這幾行：

```
device cbb
device pccard
device cardbus
```

然後加上：

```
device pcic
device card 1
```

最後請參閱Handbook 中調整FreeBSD Kernel

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/kernelconfig.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/kernelconfig.html)) 章節，以重新編譯、安裝新的kernel。

**8. 為什麼Dell™ PowerEdge™ 2850 裝完FreeBSD 之後，重開機接著鍵盤就掛了？**

(本題由cdsheen 提供)嚐試在Dell PowerEdge 2850 上面安裝FreeBSD 6.0，不過安裝完成、並重新開機之後，發現console 的鍵盤不能動了，同樣的情況似乎也存在於FreeBSD 5.3 及FreeBSD 5.4，經過一番搜尋，發覺是因為這台機器上面有一個“Dell Remote Access Controller (DRAC)”，這個裝置會被系統辨識成一個USB Keyboard，所以導致開完機之後，正常的PS/2 鍵盤反而不能動了！

暫時解決方式如下：

1. 先以Single User Mode 進入系統
2. 在命令列模式下，先執行下列命令：

```
# fsck -y /
# mount -u /
```

3. 然後編輯/etc/devd.conf，把對於USB Keyboard 的支援暫時拿掉，也就是把下面幾行開頭加上#

```
# When a USB keyboard arrives, attach it as the console keyboard.
#attach 100 {
#    device-name "ukbd0";
#    action "kbdcontrol -k /dev/ukbd0 < /dev/console && /etc/rc.d/syscons restart";
#};
#detach 100 {
#    device-name "ukbd0";
#    action "kbdcontrol -k /dev/kbd0 < /dev/console";
#};
```

4. 然後輸入exit離開Single User Mode 之後，就可以順利進入系統、而且鍵盤也可以正常運作，下次開機也不會有問題！

另外，Dell 的DRAC/BMC 看起來有蠻多不錯的遠端存取功能，有興趣的人可以玩玩看...

## 章5. 常見問題解決

### 1. 為什麼FreeBSD 抓到錯誤的記憶體容量？

抓到錯誤的記憶體容量是因為物理位址跟虛擬位址兩者是不同的。

The convention for most PC hardware is to use the memory area between 3.5G and 4G for a special purpose (usually for PCI). This address space is used to access PCI hardware. As a result real, physical memory can not appear in that address space.

What happens to the memory that should appear in that location is dependent on your hardware. Unfortunately, some hardware does nothing and the ability to use that last 500M 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 if you watch the boot messages.

On a 32 bit version of FreeBSD, the memory appears lost, since it will be remapped above 4G, which a 32 bit kernel is unable to access. In this case, the solution is to build a PAE enabled kernel. See this FAQ entry 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. This is normal and the available memory will be corrected as the boot process completes.

### 2. 硬碟有壞軌時該怎麼辦？

若是SCSI 硬碟的話，那麼磁碟機應該有能力自動作re-mapping 的動作。然而，因為一些未知的因素，在出廠時，很多硬碟的這項功能是關閉的...

要將其重新開啓，您需要編輯裝置的第一個page 模式（first device page mode），在FreeBSD 上可以用下面的指令辦到（以root身分執行）

```
# scsi -f /dev/rsd0c -m 1 -e -P 3
```

然後將AWRE 和ARRE 的數值從0 變成1:-

```
AWRE (Auto Write Reallocation Enbld) : 1  
ARRE (Auto Read Reallocation Enbld) : 1
```

以下這段是由Ted Mittelstaedt <tedm@toybox.placo.com>所提供：

若為IDE 硬碟，任何的壞軌通常都是麻煩的預兆。目前所有較新的IDE 硬碟，內部都有自動remapping 壞軌的能力。目前所有IDE 硬碟製造商，都提供了更久的保證，而且會幫您更換出現壞軌的硬碟。

如果您仍想要修復產生壞軌的IDE 硬碟，您仍可以試著去下載IDE 硬碟製造商所提供的檢測程式，並用它來檢查您的硬碟。有時這些軟體可以強迫重新檢查硬碟的壞軌，並將它們標示出來。

對ESDI, RLL 及MFM 的硬碟來說，通常壞軌是正常現象，也不是什麼麻煩的前兆。在PC 上，磁碟控制卡和BIOS 負責標示壞軌的任務。這對一些使用BIOS來存取磁碟的作業環境（如DOS）是沒有問題的。然而，FreeBSD 的磁碟驅動程式並不經過BIOS 來存取磁碟，所以，有個bad144 的機制用來取代這項功能。bad144 只能用在wd 這個磁碟驅動程式上（這個代表了FreeBSD 4.0 並不支援它），它也無法用在SCSI 硬碟上。bad144的工作方法是將所有找到的壞軌資料存到一個特別的檔案☹。

使用bad144 的警告- 存著壞軌資料的特別檔案是放在硬碟的最後一軌上。因為這個檔案儲存的壞軌資料中，有可能有些資料是指向硬碟最前端所發生的壞軌情形，就是可能儲存/kernel 這個檔的地方，所以它一定要能被開機程式所讀取，而開機程式是透過BIOS 來讀取kernel 檔。這表示了使用bad144 的硬碟絕不能擁有超過1024 個cylinder，16 個head 及63 個sector。而這使得欲使用bad144 的硬碟的大小不能大於500 MB。

要使用bad144 很簡單，只要在開始安裝時，在FreeBSD fdisk 畫面把“Bad Block” 掃描設為ON 即可。在FreeBSD 2.2.7 以後都可以使用此方法。但這個硬碟的cylinder 一定要在1024 以下。使用前，我們建議這個硬碟要至少先使用四個小時，以便熱膨脹與磁軌偏移達一般狀態。

如果這個硬碟擁有超過1024 個cylinder（像大容量的ESDI 硬碟），ESDI 控制卡利用一個特別的轉換模式使它能在DOS 下工作。而如果您在fdisk 的“set geometry” 中輸入“轉換過” 的geometry，wd 這個驅動程式能了解這些轉換模式。您也絕對不能使用dangerously dedicated 模式來建立FreeBSD 的分割區，因為它會忽略geometry 這個參數。此外，就算fdisk 使用您所輸入的geometry 參數，它依然會去讀取這硬碟的真正資料，而會嘗試去建立一個過大的FreeBSD 分割區。如果磁碟的geometry 已經被“轉換” 過了，那麼這個分割區“必須” 以手動輸入block 數目的方法來建立。

一個快速的小技巧是利用ESDI 控制卡來設定大容量的ESDI 硬碟，用DOS 開機片開機，再將它format 為DOS 的分割區。然後重開機進入FreeBSD 安裝程序，在fdisk 畫面，把DOS 分割區的blocksize 和block number 抄下來。然後重新設定geometry 使其跟DOS 使用的一樣。刪除DOS 分割區，然後使用您剛剛抄下的blocksize 來建立一個“cooperative” FreeBSD 分割區。然後設定這個分割區為可開機，再打開壞軌掃描。在真正的安裝過程中，bad144 會在任何檔案系統被建立前被執行。（您可以按Alt-F2 來監看這一切）如果在建立壞軌資料檔時發生了問題，您會需要設定一個較大的磁碟geometry - 這表示您需要重開機，然後全部再重新開始（包括重新分割以及在DOS 下重新format）。

如果remapping 的功能已經啟動了，而您依然一直看到壞軌產生，那麼考慮換一台硬碟吧。壞軌的情形只會隨時間增加而更為嚴重。

### 3. 為什麼FreeBSD 抓不到HP Netserver 的SCSI 控制卡？

基本上這個是一個已知的問題。在HP Netserver 機器上的on-board EISA 介面SCSI 控制卡占據了定址為第11 的EISA 槽，因此所有的“真實” EISA 槽都在它之前。可是，在EISA 定址空間 $\geq 10$  時，會與指定給PCI 用的定址空間相衝突，且FreeBSD 的auto-configuration 無法正確的處理這個情形。

因此，現在你能做的最好事情就是在kernel 設定EISA\_SLOTS 這個選項為12，然後當作沒有這個問題:)。請依照 Handbook 中有關kernel 的設定 (../handbook/kernelconfig.html) 所說的方法來設定與編譯您的kernel。

當然，在安裝FreeBSD 到這種機器上時，這是一個雞生蛋蛋生雞的問題。為了解決這個問題，在UserConfig 中有一個特別的方法，安裝時不要進入“visual” 介面，相反的，在命令列模式中，鍵入

```
eisa 12
quit
```

然後就如以往一樣安裝您的系統。不過我們建議您編譯與安裝一個屬於自己的kernel，但希望在未來的版本中能對這個問題有一個好的解決方法。

**注:** 您無法在HP Netserver 上使用dangerously dedicated 磁碟模式。您可以參考這份註解 以獲得更多資訊。

#### 4. 一直看到類似`ed1: timeout`的訊息。它們是什麼意思呢？

這個通常是由於中斷衝突(**interrupt conflict**)所造成的(例如：兩塊卡使用到了相同的**IRQ**)。FreeBSD 在2.0.5版以前都容許這個情形，就算有**IRQ**衝突情形，網路卡也應該仍可正常運作。然而，在2.0.5版及其以後，已不再容許有**IRQ**衝突的情形了。請於開機時使用`-c`這個選項，然後更改`ed0/de0/..`等的設定，使其和您網路卡本身的設定一致。

如果您是使用您網路卡上的**BNC**接頭，您或許也會因不良的終端電阻設定，而發生裝置(**device**) **timeout**的情形。要檢查是否有這種情形，您可以在網路卡上直接接上終端電阻(不要接網路線)，然後，看看這個錯誤訊息是不是就消失了。

有些**NE2000**的相容卡，如果它的**UTP**埠沒有接網路線，或是該網路線並沒被使用的話，也會出現這個錯誤訊息。

#### 5. 3Com 3C509 網路卡莫名其妙罷工？

這塊卡有個不好的地方在於它常常會遺失本身的設定資料。請使用該卡的**DOS**工具`3c5x9.exe`來更新卡上設定。

#### 6. 平行埠印表速度破天荒的慢，該怎麼辦？

如果唯一的問題就是速度很慢的話，試著改變您的印表機連接埠設定(`../handbook/printing-intro-setup.html#PRINTING-PARALLEL-PORT-MODE`)這個在手冊中的印表機設定(`../handbook/printing-intro-setup.html`)這個章節有加以討論。

#### 7. 程式有時會因**Signal 11**錯誤而停止？

**Signal 11**這個錯誤是因為你的**process**嘗試要存取一塊記憶體，而你的作業系統並不允許它做這個動作而發生的。如果這種情形常常不定時發生，那麼你應該要開始看看是不是哪裡出問題了。

這些問題可能是與下列情形有關：

1. 如果這個問題只在某一個您自己寫的某個特定程式發生，那麼很有可能是您的程式碼有問題。
2. 如果這個問題是在FreeBSD的某些系統檔案發生，有可能是因為程式有問題，但通常在我們這群讀**FAQ**的使用者去跑這些有問題的程式碼前，它們早就已經被解決了（這是**-current**在做的事）。

尤其如果你在編譯一個程式，但是每次編譯器跑出來的結果都不一樣的話，這是一個無解的問題，而不是FreeBSD臭蟲。

舉例來說，假設您正在跑“`make buildworld`”，而**compiler**在將`ls.c`編譯成`ls.o`時發生錯誤，這時請再跑一次“`make buildworld`”，如果**compiler**依然同樣的地方發生問題，那麼就是程式碼有問題——請更新原始碼然後再試試看。而如果**compiler**是在其他的地方發生錯誤，那麼幾乎可以確定是硬體的問題了。

您這時應該做什麼：

如果是第一種情形，可以使用一些**debugger**，如：`gdb`，來找出程式是在那兒會去嘗試存取錯誤的記憶體位址，然後再修正它。

如果是第二種情形，就需要檢查看看是不是硬體的問題了。

一些造成硬體不穩的原因包括：

1. 可能是硬碟過熱：請檢查機殼內的風扇是否運作正常，因為您的硬碟（或者還有其他的硬體裝置）過熱了。

2. 處理器過熱：這個有可能是因為超頻，或者是處理器的風扇掛了。不論是哪種原因，您都需要將所有的元件回復到它們原先設定的工作狀態，這樣才能解決這個問題。舉個例子來說：將處理器調回原先的工作頻率。

如果您還是堅持要超頻的話，請謹記，與其燒壞而需要換新的一台主機，不如將速度調慢一點！除此之外，不管你覺得它安不安全，一般人對於您因為超頻而發生的問題，是不會有什麼同情心的。

3. 不穩定的記憶體：如果主機上有安裝數根SIMM/DIMM 記憶體，試著把它們全拆下來，然後一根一根插上去做測試，藉此縮小範圍，以便找出有問題的某根記憶體或是某種記憶體組合。
4. 最佳化過頭的主機板設定：在BIOS 或是有些主機板的jumper 上，有時可以更改一些timing，但在大多數的情形，使用預設值就已經足夠了。況且有時候把RAM 的wait states 設太小，或是在BIOS 把“RAM Speed: Turbo” 這個或是其他類似的選項打開都有可能造成一些不正常的現象。一個解決的方法是把BIOS 設回預設值，不過在這之前記得先記下目前的設定！
5. 供給主機板的電力不乾淨或是不足。試著把系統內沒有用到的I/O 卡、硬碟或是CDROM 暫時拆掉或是拔掉電源線，看看你的電源供應器是不是能夠在小一點的負荷下正常工作。不然就是換上另一個新的電源供應器，最好是瓦數高一點的（打個比方來說，如果原先的電源供應器是250 瓦的，那麼就換上300 瓦的試試）。

請順便參閱SIG11 FAQ（連結在下面），雖然它是站在Linux 的角度寫的，可是裡面對這些問題有許多很棒的解說。它裡面也有討論為什麼有問題的記憶體能通過軟體或硬體的測試的原因。

最後，如果上面這些原因都排除了，那麼有可能是遇到了FreeBSD 的一隻臭蟲，請參閱指示做一個問題回報。

這兒有一個更詳細的FAQ — the SIG11 problem FAQ (<http://www.bitwizard.nl/sig11/>)

8. 當機時出現：Fatal trap 12: page fault in kernel mode，或是panic: 以及一堆錯誤訊息，該怎麼辦？

FreeBSD 的開發者對於這些錯誤訊息相當的有興趣，但是他們需要更詳細的一些細節。請把您的當機的訊息全部複製下來，接著查閱FAQ kernel panics 這節，依說明編譯一個含除錯碼的kernel，以取得函式呼叫順序（backtrace）。這個聽起來很難，但實際上並不需要任何程式設計的能力，您只需要依照指示做即可。

9. 為什麼當我開機時，螢幕變黑，且不停閃動？

這個問題，已知是由ATI Mach 64 顯示卡所引起的。因為這塊卡使用到2e8 這個位址，而這與第四個序列埠（serial port）所使用的位址相同。而在sio(4) 這個驅動程式，不知道是bug 或是功能（feature），就算您沒有第四個序列埠，或是已經將sio3（第四個序列埠）取消了，它依然會去嘗試驅動它。

直到這個問題被解決以前，您可以使用這個方法：

1. 在看到開機提示時輸入-c（這會讓kernel 進入設定模式）。
2. 取消sio0，sio1，sio2 和sio3（全部）。這可以讓sio 驅動程式不動作->於是問題解決。
3. 輸入exit 以繼續啟動程序。

如果您想要使用您的序列埠，您需要修改/usr/src/sys/i386/isa/sio.c，在該檔中找出0x2e8 這個字串，移除這個字串及它前面的逗號（保留後面的），然後重新編譯一個新的kernel。

就算使用了上面這些方法，X Window 仍然有可能無法順利執行。如果發生了這種情形，請確定你用的XFree86的版本是最新的XFree86 3.3.3 或是其後的版本。它們有內建支援Mach 64 這張卡，甚至為了這些卡還附有一個特別的X Server

#### 10. 為什麼我的系統裝有128 MB 的RAM，而FreeBSD 只用了其中的64MB？

因為FreeBSD 是使用呼叫BIOS 來取得記憶體大小的方法，因此它只能偵測到16 bits 位元長度的KByte 大小（65535 KBytes = 64MB）（或者更少..。有些BIOS 將最高記憶體大小限為只有16MB）如果您擁有64MB 以上的RAM，FreeBSD 會嘗試去偵測出它，但是有可能會失敗。

要解決這個問題，您需要使用下面所提的kernel 設定選項。雖然有方法可以從BIOS 中取得記憶體的完整資訊，但是目前我們在開機區中並沒有多餘的空間來做這件事。當某天開機區空間不足的情形獲得解決時，我們將會使用BIOS 的延伸功能來取得記憶體的完整資訊...但現在我們將它放在kernel 設定選項中。

```
options "MAXMEM=n"
```

*n* 是指您的記憶體大小，以KB 為單位。以一台有128MB RAM 的機器來說，您可使用131072這個數字。

#### 11. 機器上的RAM 有1GB 以上，可是為何卻收到“kmem\_map too small” 的panic 錯誤訊息？

通常FreeBSD 會依據機器狀況來自動調整kernel 相關參數設定，比如：根據機器所裝的RAM 大小來決定同時可開啓的檔案數量多寡。然而，在1GB RAM(含以上) 的機器上，這個『自動調整』的機制可能有時會高估：比方說..開機時，kernel 會先配置各種不同用途的表格及其他架構放到記憶體上，然後，當整個作業系統都開始運作之後，kernel 就會開始不夠空間來做記憶體配置的動態調整，於是就panic 掛了。

解法是：把VM\_KMEM\_SIZE\_MAX 加到kernel 設定檔內，並重新編譯kernel，比如：

```
options VM_KMEM_SIZE_MAX=419430400
```

這樣會設定400 MB 來給kernel 使用，而且採用400 MB 的話，目前在6GB RAM 的機器上都可被有效運用。

#### 12. 機器上RAM 不到1GB，但仍會出現kmem\_map too small! 的panic 錯誤訊息

之所以panic 的原因在於系統用光了給network buffer 用途的virtual memory (尤其是mbuf clusters)。解法是增加給mbuf clusters 用的virtual memory 數量，這步驟請參閱FreeBSD 使用手冊的網路限制篇 ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/configtuning-kernel-limits.html#NMBCLUSTERS](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/configtuning-kernel-limits.html#NMBCLUSTERS))。

#### 13. 為什麼我一直看到/kernel: proc: table is full 這個錯誤訊息？

FreeBSD kernel 只允許一定數量的process 在同一時間裡同時運作。而這個數目是根據kernel 設定檔裡面的MAXUSERS 值來決定的。MAXUSERS 這個值也會影響其他的kernel 內定值，比如說network buffer (請參閱這個之前討論過的問題)。如果機器負荷(load)很重，您可能需要增加MAXUSERS 這個值。這麼作會一併提高系統的其他內定值，包括最大可擁有的process 數等。

若要調整MAXUSERS，請參閱FreeBSD 使用手冊中的 檔案/Process的限制

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/configtuning-kernel-limits.html#KERN-MAXFILES](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/configtuning-kernel-limits.html#KERN-MAXFILES)) 章節。(雖然該處指的是『檔案的開啓數量限制』，但也適用於process部分。)

在FreeBSD 4.4 之後，MAXUSERS 已經變成可以靠著更改/boot/loader.conf 的kern.maxusers 這個值而調整的變數了。而在之前的FreeBSD 版本中，這個值只能在kernel 設定檔調整。

如果機器負荷並不重，而您只是需要同時跑很多很多process，那麼也可以直接用sysctl 調整kern.maxproc 值。假如這些process 都是屬於某個使用者的，那麼您還需要另外調整kern.maxprocperuid 這個值，使它比新的kern.maxproc 這個值少1 (一定要少1，因為init(8) 這個系統程式絕對要保持在運作狀態)。

如果在每次開機都要更改sysctl 的值，而且您的FreeBSD 是最近的版本的話，請在/etc/sysctl.conf 這個檔中設定，而如果是舊的版本，可以在/etc/rc.local 中作設定。

**14. 為什麼用新kernel 開機時，出現CMAP busy 這個錯誤訊息？**

用來偵測/var/db/kvm\_\*.db 過時檔案的機制偶爾會發生問題，而使用到了一個不協調(mismatch)的檔案有時就會導致panic。

如果發生了這個問題，請重新開機，進入single 使用者模式，然後執行：

```
# rm /var/db/kvm_*.db
```

**15. 請問這個訊息：ahc0: brkadrint, Illegal Host Access at seqaddr 0x0 是什麼意思？**

這是一個和Ultrastor SCSI 控制卡有關的衝突 (conflict)。

在開機時，進入kernel 設定選單取消uha0，它是造成這個問題的原因。

**16. 開機時，看到這個錯誤訊息ahc0: illegal cable configuration。我的排線確定有接對。是出了什麼問題呢？**

您的主機板可能不支援自動終端電阻設定。請進到SCSI 的BIOS 裡面手動指定正確的終端電阻順序，而不要使用自動設定。AIC7XXX 的驅動程式並無法知道有沒有這些排線偵測(以及自動終端電阻設定)的電路(external logic) 存在。如果EEPROM 裡面的設定是"automatic termination" 時，它只會單純假定這些電路當然是存在的。若缺少了這個電路，驅動程式在設定終端電阻時就常常出問題。而這種問題將導致SCSI 匯流排的可靠性降低。

**17. 為什麼Sendmail 一直出現 "mail loops back to myself" 這個錯誤訊息？**

這個問題在sendmail 的FAQ 中是這樣回答的:-

\* 我一直收到有關 "Local configuration error" 的信件，例如：

```
553 relay.domain.net config error: mail loops back to myself
554 <user@domain.net>... Local configuration error
```

我要如何解決這個問題？

您利用 MX 設定，讓要寄到某 domain (如: domain.net) 的信件，寄到您所指定的機器 (在這個例子中為 relay.domain.net)，但是這部機器並未被設定接受 domain.net 的信件。請把 domain.net 加到/etc/sendmail.cw 中 (如果您有使用 FEATURE(use\_cw\_file)) 或是在 sendmail.cf 中加入 "Cw domain.net"

最新版本的sendmail FAQ (<ftp://rtfm.mit.edu/pub/usenet/news.answers/mail/sendmail-faq>) 現在已不再隨著sendmail “出貨”。它目前是被定期的發表在comp.mail.sendmail (news:comp.mail.sendmail) , comp.mail.misc (news:comp.mail.misc) , comp.mail.smail (news:comp.mail.smail) , comp.answers (news:comp.answers) , 和news.answers (news:news.answers). 您也可以寄一封Email 到<[mail-server@rtfm.mit.edu](mailto:mail-server@rtfm.mit.edu)> , 然後在信件內文中寫上send usenet/news.answers/mail/sendmail-faq 以取得這份FAQ 文件。

### 18. 為什麼執行遠端機器 (remote machine) 的全螢幕的軟體時, 有不正常的情形?

或許遠端機器並非將您的終端機模式設為FreeBSD console 所用的cons25, 而是設為其它模式。

這兒有幾個解決這個問題的方法:

- 在logging 進遠端機器後, 更改您的shell 變數TERM 為ansi 或是sco (如果遠端機器支援這些模式的話)。
- 使用支援VT100 的模擬軟體, 如FreeBSD console 下的screen 軟體。screen 提供您在一個terminal 同時跑好幾個session 的能力, 而且它本身也是一個相當好的軟體。每個screen 都像是一個VT100 的終端機, 所以遠端機器的TERM 變數應該設為vt100。
- 在遠端機器的終端機資料庫 (terminal database) 中加入cons25 的資料。加入的方法視遠端機器的作業系統不同而有所差異。請參閱遠端機器給系統管理員的說明書, 應該會有所幫助。
- 啟動FreeBSD 的X 伺服器, 然後使用一些X Window 下的終端機模擬器來登入遠端機器, 例如xterm 或rxvt。而遠端機器的TERM 變數應該要設為xterm 或vt100。

### 19. 為什麼我的機器一直顯示calcr: negative time... ?

跟中斷 (interrupt) 有關的不同硬體與/或軟體的搭配都有可能造成這個問題。這有可能是bug 或是某個裝置本身的問題。在平行埠上使用大的MTU 來作TCP/IP 傳輸可以重現這個問題。若是圖形加速卡造成這個問題的話, 您應該先檢查卡的中斷設定。

這個問題的邊際效應是會造成有些process 出現 “SIGXCPU exceeded cpu time limit” 的訊息, 而不正常停止。

若是FreeBSD 3.0 或是1998 年11 月29 日以後其他版本, 萬一這個問題一直無法以其他方法解決, 就只能設定sysctl 變數:

```
# sysctl -w kern.timecounter.method=1
```

這樣會對效能有些影響, 但是若考慮到這個問題帶來的後果, 這樣做是值得的。如果這個問題還是存在的話, 讓sysctl 那個值依然設為1, 然後增加kernel 設定檔中NTIMECOUNTER 這個選項的數值。如果您將NTIMECOUNTER 增加到20 依然無法解決這個問題, 那麼您機器上的中斷已經多到無法讓計數器維持在可靠的狀態了。

### 20. 出現pcm0 not found 這個訊息, 或者是我的音效卡變成了pcm1, 但在kernel 設定檔我是設device pcm0 啊。這是怎麼回事呢?

如果您在FreeBSD 3.x 上使用PCI 音效卡就會發生這種問題。因為pcm0 這個device 是內定保留給ISA 的音效卡的, 所以如果您有一張PCI 的音效卡, 您就會遇到這個問題, 而您的卡會變成pcm1。

**注:** 如果您只把kernel 設定檔中的設定改成device pcm1 是無法除去這個警告訊息的，這樣會造成pcm1 被保留給ISA 音效卡，而PCI 音效卡則會變成pcm2（外加pcm1 not found 的警告訊息）。

如果您有一張PCI 的音效卡，您需要make snd1 這個device，而不是snd0：

```
# cd /dev
# ./MAKEDEV snd1
```

這個問題在FreeBSD 4.x 上並不會發生，因為很多人投下了許多心力讓它更PnP 導向，而且pcm0 這個device 也不再是只保留給ISA 的音效卡了。

**21.** 為什麼在更新到FreeBSD 4.X 後會抓不到我的PnP 卡（或者是抓成unknown）？

FreeBSD 4.X 現在已經更PnP 導向了，而邊際效應就是會發生有些在FreeBSD 3.X 可以用的PnP 裝置（如音效卡或是內插式數據機）變成無法使用。

這個原因可以用一封由Peter Wemm 發到freebsd-questions 這個mailing list 上的信來解釋，它原本是解釋為什麼有一個內插式數據機，在系統升級到FreeBSD 4.x 後，就沒法被抓到了（在[] \*的是另外加的註解，讓內容更易懂）。

The PNP bios preconfigured it [the modem] and left it laying around in port space , so [in 3.x] the old-style ISA probes “found” it there.

Under 4.0 , the ISA code is much more PnP-centric. It was possible [in 3.x] for an ISA probe to find a “stray” device and then for the PNP device id to match and then fail due to resource conflicts. So , it disables the programmable cards first so this double probing cannot happen. It also means that it needs to know the PnP id’s for supported PnP hardware. Making this more user tweakable is on the TODO list.

如果要讓裝置能再度運作，我們需要找出它的PnP id，然後再將它加入一份在偵測ISA 裝置時會使用的表中。可以執行pnpinfo(8) 來偵測這個裝置，舉例來說，下面是pnpinfo(8) 抓到的一個內插式數據機的資料：

```
# pnpinfo
Checking for Plug-n-Play devices...

Card assigned CSN #1
Vendor ID PMC2430 (0x3024a341) ,Serial Number 0xffffffff
PnP Version 1.0 ,Vendor Version 0
Device Description: Pace 56 Voice Internal Plug & Play Modem

Logical Device ID: PMC2430 0x3024a341 #0
    Device supports I/O Range Check
TAG Start DF
    I/O Range 0x3f8 .. 0x3f8 ,alignment 0x8 ,len 0x8
    [16-bit addr]
    IRQ: 4 - only one type (true/edge)

[more TAG lines elided]

TAG End DF
End Tag

Successfully got 31 resources ,1 logical fdevs
```

```
-- card select # 0x0001

CSN PMC2430 (0x3024a341), Serial Number 0xffffffff

Logical device #0
IO: 0x03e8 0x03e8 0x03e8 0x03e8 0x03e8 0x03e8 0x03e8 0x03e8
IRQ 5 0
DMA 4 0
IO range check 0x00 activate 0x01
```

您所需要的資訊是一開始看到的“Vendor ID”這一行。括號中的十六位元碼（這個例子中是0x3024a341）就是PnP id，而在這之前的字串（PMC2430）則是一個獨一無二的ASCII id。而這些資料需要被加到/usr/src/sys/isa/sio.c這個檔案。

為了防止任何東西出錯，您應該要先備份目前的sio.c。而且您要submit PR時也需要這個原始檔案來做出patch（您應該會將它submit PR吧..：）..。接著就編輯sio.c找尋下面這行

```
static struct isa_pnp_id sio_ids[] = {
```

接著往下捲動，找個正確的位置來插入您的裝置資訊。您看到的就下面這個樣子，它們是照右邊註解裡面的ASCII這個Vendor ID做排序的，或是pnpinfo(8)所找到的一部分裝置描述：

```
{0x0f804f3f, NULL}, /* OZ0800f - Zoom 2812 (56k Modem) */
{0x39804f3f, NULL}, /* OZ08039 - Zoom 56k flex */
{0x3024a341, NULL}, /* PMC2430 - Pace 56 Voice Internal Modem */
{0x1000eb49, NULL}, /* ROK0010 - Rockwell ? */
{0x5002734a, NULL}, /* RSS0250 - 5614Jx3(G) Internal Modem */
```

把您這個裝置的十六進位的Vendor ID加到正確的地方，存檔，然後重新編一個kernel，再重開機。之後這個裝置應該就會像在FreeBSD 3.X下，被偵測為sio裝置了。

## 22. 為什麼我常常在跑一些程式（例如top或sysstat）的時候出現nlist failed這個錯誤訊息？

這個問題是因為您跑的程式需要一個特別的kernel symbol，可是不知道什麼原因而找不到；而會發生這個問題可能是因為下面兩個原因：

- 您的kernel和userland的檔案版本並不一致（例如說，您編了一個新的kernel，但是並沒有執行對應的installworld，或是其他類似情形），因此symbol table的內容就和應用程式編譯時的不太一樣了。如果是這種情形，請執行完整的升級步驟（請參閱/usr/src/UPDATING以得知正確的流程）。
- 您沒有用/boot/loader來載入您的kernel，而是直接由boot2開機（請參閱boot(8)）。雖然說跳過/boot/loader並沒有什麼錯，但是它在kernel symbols跟應用程式的溝通方面佔了很重的份量。

## 23. 為什麼我用ssh或telnet連到我的電腦時，會等待很長的一段時間才能連上？

症狀：TCP連線建立之後和詢問密碼之前（如果是在說telnet(1)的話，則是login提示符號跳出來之前），要等待很長的一段時間。

問題所在：這種延遲情形常常是因為伺服器軟體（server software）嘗試要將客戶端（client）的IP位址轉換成主機名稱。因為很多伺服器軟體，包括FreeBSD內建的Telnet和SSH，為了將主機名稱寫入紀錄檔中以便管理者作參考，而會做這項動作。

解決方法：如果這個問題在您連接不同的伺服器時都會發生，那麼問題是在您客戶端這一方；同樣的，如果別人只有在連到您的伺服器上才會發生這個情形，那麼問題就是在伺服器這邊了。

如果是客戶端這方有問題，唯一的方法就是將DNS 伺服器修好，這樣對方伺服器才能正確的轉換名稱。如果問題是在內部區域網路發生的，這應該是伺服器有問題，請詳細檢查一下；相反的，如果您是在Internet 時發生的，那麼您需要跟您的ISP 聯絡，請他們解決這個問題。

如果是伺服器這邊的問題，而且是發生在內部區域網路，那麼您需要設定這個伺服器，使它能正確將內部網路的IP 位址轉換為主機名稱。請參閱hosts(5) 和named(8) 的說明以獲得更多資訊。如果是在Internet 上的伺服器發生這個問題，那麼有可能是您伺服器的轉換功能出問題。您可以試試查詢另一個主機名稱，比如：`www.yahoo.com`。如果查不到，那麼可以確定是您這邊出問題了。

#### 24. stray IRQ 這個錯誤訊息是什麼意思？

Stray IRQs 是硬體IRQ 有點小問題的現象，大多是因為硬體本身在發出中斷需求後，又取消了它自己的中斷要求。

有三個方法可以應付這個問題：

- 不理會這個警告。反正一個irq 出現五次警告後系統就不會再顯示了。
- 把isa\_strayintr() 的值，由5 改成0，這樣所有的警告訊息都不會出現。
- 安裝使用irq 7 的平行埠硬體設備，以及它的PPP 驅動程式（這個大部分系統都有做），接著安裝ide 硬碟或是其他會使用irq 15 的硬體設備以及它的驅動程式。

#### 25. 為什麼file: table is full 這個訊息一直在dmesg 重複出現？

這個錯誤訊息代表了您系統的file descriptors 已經使用光了。請參閱手冊內 Tuning Kernel Limits (`./handbook/configtuning-kernel-limits.html`) 裡面的 `kern.maxfiles` (`./handbook/configtuning-kernel-limits.html#KERN-MAXFILES`) 這個章節，裡面有一些討論及解決方法。

#### 26. 為什麼我筆記型電腦上的時鐘一直顯示錯誤的時間？

您的筆記型電腦裡有兩個以上的時鐘，而FreeBSD 選到了錯的那個。

執行dmesg(8)，檢查一下有Timecounter 字串的那幾行。最後一行是FreeBSD 選用的，通常是TSC。

```
# dmesg | grep Timecounter
Timecounter "i8254" frequency 1193182 Hz
Timecounter "TSC" frequency 595573479 Hz
```

您可以執行sysctl(3) 看一下`kern.timecounter.hardware` 這個值做確認。

```
# sysctl kern.timecounter.hardware
kern.timecounter.hardware: TSC
```

BIOS 可能在一些情形下會更改TSC 的時脈——有時候是因為在使用電池工作時會更改處理器的速度，另外也有可能是進入了省電模式，可是FreeBSD 並不會察覺到這些調整，而會發生時間增加或是減少的情形。

在上面的例子當中，我們看到還有i8254 這個時鐘可以選擇，執行sysctl(3) 用手動的方式將這個值寫入`kern.timecounter.hardware` 中。

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

這樣您的筆記型電腦應該就可以保持正確的時間了。

如果要讓這個更改的動作再每次開機時自動執行，在/etc/sysctl.conf 加入下面這行。

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

## 27. 為什麼我的筆記型電腦無法正確的偵測到PC card ？

這個問題常常發生在灌了多個作業系統的筆記型電腦上。有些非BSD 的作業系統會讓PC card 的硬體裝置處在一個不一致的狀態下 (inconsistent state) 。使得pccardd 在偵測這片卡時，無法抓到正確的型號，而是 "(null) "(null) " 。

您需要移除PC card 插槽的電源以重置這個硬體裝置。一個方法是將您的筆記型電腦關機 (不是休眠模式，也不是待命模式；要完全的關機) 。等個幾秒鐘再重開機。這樣您的PC card 應該就正常了。

有時有些筆記型電腦雖然看起來已經關機了，但實際上並沒有。如果您發現上面那個方法沒有用，請關機，移除電池，等個幾秒鐘，把電池裝上去然後重開機。

## 28. 為什麼在BIOS 畫面之後，FreeBSD 的boot loader 顯示Read error 然後就停止不動了？

這是因為FreeBSD 的boot loader 無法正確的找出硬碟的geometry 。這樣的話，就需要在用fdisk 分割或是修改FreeBSD 的slice 時，手動將正確的值輸入進去了。

正確的硬碟geometry 值在BIOS 裡面可以查的到。注意該硬碟的cylinders , heads 以及sectors 這些數值。

在執行sysinstall(8)的fdisk 時，按下G 以便手動設定硬碟的geometry 。

這時會有一個對話框跳出來，詢問您有關cylinders , heads 以及sectors 這些東西的值。請將剛剛在BIOS 查到的數字，以/ 作分隔輸入進去。

舉例來說，如果是5000 cylinders , 250 sectors 和60 sectors 就輸入 **5000/250/60**

輸入完後請按enter 鍵確認，最後按下W 鍵把新的分割區表寫入硬碟當中。

## 29. 另一個作業系統摧毀了我的Boot Manager 。我要怎麼樣才能把它還原回來？

執行sysinstall(8) 接著選Configure , 然後選Fdisk 。再來用空白鍵選擇原先Boot Manager 所在的硬碟。按下W 鍵來作寫入的動作。這時會跳出一個提示訊息，詢問您要安裝哪一個boot loader 。請選擇Boot Manager , 這樣就可以將它還原了。

## 30. 這個錯誤訊息：swap\_pager: indefinite wait buffer: 是什麼意思呢？

這個訊息是說有一個執行程序正在嘗試將分頁記憶體 (page memory) 寫入硬碟中，而這個動作嘗試了20 秒鐘仍然無法成功。這個有可能是因為硬碟有壞軌、電路或排線有問題、以及其他跟硬碟讀出寫入有關的硬體設備。如果真的是硬碟壞軌的問題，您應該會在/var/log/messages 這個檔案中，或是在執行dmesg 這個指令後，看到有關磁碟錯誤的訊息。如果沒有，那麼請檢查您的排線還有接頭連接是否良好。

**31.** 為何在buildworld/installworld 時，會臥在touch: not found 的錯誤訊息?

這錯誤訊息並不是指touch(1) 程式不見了，事實上可能是該檔檔案時間被設為未來的時間。若機器上的CMOS-clock 時鐘設定為當地時間(非格林威治時間，比如台灣時間為GMT +08:00 ，也就是CST 中原標準時間)，那麼請在開機時，先選single user 模式進入，然後打adjkerntz -i 來調整kernel clock 與機器上的CMOS-clock 來同步。

## 章6. 商業軟體

注：這一節的內容還是相當少，不過我們當然希望各個公司能為它加點內容:) FreeBSD 組織和列在這裡的任何一家公司都沒有金錢上的利害關係，列出來純粹只是對大眾公開介紹(同時也認為在FreeBSD 上的商機若興旺，會對FreeBSD 可長可久有極正面的效益)。我們鼓勵商業軟體的廠商把他們的產品包括在下面的名單中，在Vendors page ([../././././commercial/index.html](http://www.freebsd.org/commercial/index.html)) 可以看到更長的列表。

### 1. 在哪邊找到給FreeBSD 用的Office 套件呢？

OpenOffice (<http://www.openoffice.org>) 這套open-source 性質的office 可以在FreeBSD 上正常運用自如，而Linux® 版的StarOffice (<http://www.sun.com/staroffice/>), 這套closed-source 的OpenOffice 加值版，也可以在FreeBSD 上正常使用。

FreeBSD 上還有許多編排軟體、試算表(Spreadsheet)以及繪圖軟體都可用Ports Collection 來安裝喔。

### 2. 在哪邊可找到給FreeBSD 用的Motif ？

Open Group 釋出了Motif 2.1.30 的原始碼，可以透過open-motif package 安裝，或是由ports 自行編譯。相關的資訊，請參考handbook 中的ports ([../handbook/ports.html](http://www.freebsd.org/handbook/ports.html)) 章節。

注：Open Motif 只能在同樣也是open source (<http://www.opensource.org/>) 的作業系統或計劃中使用。

另外，也是有商業版本的Motif 存在。也許這種版本的Motif 不是免費的，但是絕對允許用在closed-source 的環境下。Apps2go 提供了最便宜的FreeBSD (包括i386 跟alpha)版本的ELF Motif 2.1.20 套件。

目前提供兩種不同環境的版本，“發展用版本”及“runtime 版本”。這兩種套件都包括：

- OSF/Motif manager, xmbind, panner, wsm.
- Development kit with uil, mrm, xm, xmcxx, include and Imake files.
- Static and dynamic ELF libraries (for use with FreeBSD 3.0 and above).
- Demonstration applets.

因為Apps2go 也有提供NetBSD 和OpenBSD 的版本，所以在訂購時請特別指定是要FreeBSD 版本的Motif ! 他們目前只提供以FTP 的方式取得這份套件。

更多資訊

Apps2go WWW page (<http://www.apps2go.com/>)

或

<sales@apps2go.com> 或 <support@apps2go.com>

或

phone (817) 431 8775 or +1 817 431-8775

也可以聯絡Xi Graphics，他們提供了一個FreeBSD a.out 格式的Motif 2.0 套件。

在這套件中包括了：

- OSF/Motif manager, xmbind, panner, wsm.
- Development kit with uil, mrm, xm, xmcxx, include and Imake files.
- Static and dynamic libraries (for use with FreeBSD 2.2.8 and earlier).
- Demonstration applets.
- Preformatted man pages.

在你跟他們訂購Motif時，請一定註明你要的是FreeBSD的版本！因為Xi Graphics也同時提供了BSDI跟Linux版本的Motif。目前發行的版本是放在四塊磁片中，將來他們會將所有的東西都放到光碟裡，就像他們所發行的CDE一樣。

### 3. 在哪邊可找到給FreeBSD用的CDE？

Xi Graphics 以前有賣FreeBSD用的CDE，不過現已停止發售了。

就許多方面而言，KDE (<http://www.kde.org/>) 這個open source 的桌面環境與CDE相當類似。此外，你可能會喜歡使用xfce (<http://www.xfce.org/>)。KDE及xfce都可由ports機制 (<http://www.FreeBSD.org/ports/index.html>)來安裝。

### 4. 有沒有要錢，但是高效率的X servers？

有，Xi Graphics (<http://www.xig.com/>) 有提供給FreeBSD（或其他Intel平台上）用的X視窗加速產品。

Xi Graphics所提供的高效能X Server有非常簡單的設定方式，並且支援了目前市面上當紅的各大廠牌的顯示卡。它只給你Binary檔案，是用磁片的方式發行，FreeBSD跟Linux版本都相同。Xi Graphics同時也提供了專門給筆記型電腦用的高效能X Server。

### 5.0 版有提供免費的相容“demo”版本

Xi Graphics 也有在賣FreeBSD用的Motif跟CDE（往上面看看）。

更多的資訊

Xi Graphics WWW page (<http://www.xig.com/>)

或

<[sales@xig.com](mailto:sales@xig.com)> 或<[support@xig.com](mailto:support@xig.com)>

或

phone (800) 946 7433 or +1 303 298-7478.

**5.** 在FreeBSD 上有任何的資料庫嗎？

有! 請看FreeBSD 網站上 商業軟體公司 ([../../../commercial/software\\_bycat.html#CATEGORY\\_DATABASE](http://www.freebsd.org/commercial/software_bycat.html#CATEGORY_DATABASE)) 這一部份。

還有請參考ports 中 Databases ([../../../ports/databases.html](http://www.freebsd.org/ports/databases.html)) 相關的收集。

**6.** 可以在FreeBSD 上執行Oracle 嗎？

可以，下面這個網頁會說明如何在FreeBSD 上執行Linux 版的Oracle：

- <http://www.scc.nl/~marcel/howto-oracle.html> (<http://www.scc.nl/~marcel/howto-oracle.html>)
- <http://www.lf.net/lf/pi/oracle/install-linux-oracle-on-freebsd>  
(<http://www.lf.net/lf/pi/oracle/install-linux-oracle-on-freebsd>)

# 章7. 一般應用程式

## 1. 嗯..我要在哪找到我要的程式呢?

請看看ports 目錄 ([.././../ports/index.html](http://www.FreeBSD.org/ports/index.html)) 吧。這邊有份已經port 到FreeBSD 的軟體列表。目前有超過24,000 個軟體已經被port 到FreeBSD 上，並且每天都在增加中。所以有空就多看看這份列表，不然你也可以訂閱 `freebsd-announce` 這份mailing list，會有人將每個星期最新的軟體列表貼在上面。

大部份的ports 應該都可以在4.X、5.X 跟6.X 的系統上使用。每次當FreeBSD release 新版時，都會有一份ports tree 被放在這一個release cd 裡面的ports/ 目錄裡。

我們也支援一種叫“package” 的概念，基本上就是gzip 壓縮、可用來發行的binary 檔案，但是裡面藏了一些相當有用的資訊，可以給各種自訂安裝來使用。使用者不必知道某個package 裡究竟有包括哪些檔案，就可以很方便地重複將它安裝/反安裝。

你可以執行 `sysinstall` (FreeBSD 5.2 之前版本則是 `/stand/sysinstall`) 後，在post-configuration 選單下選擇package 這個安裝選項；或是對某個有興趣的package 檔案執行 `pkg_add(1)` 把它裝起來。Package 檔案通常以 `.tgz` 或 `.tbz` 為副檔名，手上有FreeBSD CDROM 的人可以在 `packages/All` 這個目錄下找到這類檔案。對不同的FreeBSD 版本，也可以從下列位址由網路上取得：

給4.X-RELEASE/4-STABLE 用的

```
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-4-stable/  
(ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-4-stable/)
```

給5.X-RELEASE/5-STABLE 用的

```
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-5-stable  
(ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-5-stable/)
```

給6.X-RELEASE/6-STABLE 用的

```
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-6-stable  
(ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-6-stable/)
```

給7-CURRENT 用的

```
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-current  
(ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-current/)
```

或是離你最近的mirror 站。

要注意的是，因為新的port 一直在增加中，所以並不是所有port 都有相對應的package。最好定時檢查 `ftp.FreeBSD.org` (`ftp://ftp.FreeBSD.org/pub/FreeBSD/`)，看看有哪些package 可以用。

## 2. 該怎麼設定INN(Internet News) 來當news server ?

以package 或port 方式來裝好 `news/inn` 之後，Dave Barr's INN Page (<http://www.visi.com/~barr/INN.html>) 是個非常好的INN 入門處，你可以在那邊找到INN 的FAQ。

## 3. FreeBSD 有支援Java™ 嗎？

有啊，請看 <http://www.FreeBSD.org/java/> (<http://www.FreeBSD.org/java/index.html>)

**4. 我可以在哪邊找到libc.so.3.0?**

你可能在一台2.1.x 的機器上，跑著給2.2/3.x/4.0 的軟體。請再往上面一個章節看，正確的取得給你機器用的port/package。

**5. 為何我得到了這個訊息?Error: can't find libc.so.4.0?**

你不小心抓了給4.X 及5.X 系統用的package，並且嘗試著去裝在你的2.X 或3.X 的系統上面。請下載正確版本的package。

**6. ghostscript 在我的386/486SX 上有點問題呢！**

你沒有浮點運算器，對吧？你必須在你的kernel 中加入數學運算模擬器，你可以跟著下面的步驟做，並在更改過你的kernel 設定檔後，重新編譯過一次。

```
options GPL_MATH_EMULATE
```

**注:** 當你加入上一行的同時，你必須將MATH\_EMULATE 移除掉。

**7. 為什麼當我執行SCO/iBCS2 的程式時，它在socksys 這個地方出了問題？(FreeBSD 3.0 以及更早的版本才有此問題。)**

你必須先修改/etc/sysconfig (或是/etc/rc.conf, 請讀rc.conf(5)) 這檔案最後一個章節，將下面所講到的變數設成YES：

```
# Set to YES if you want ibcs2 (SCO) emulation
    loaded at startup ibcs2=NO
```

這會在開機時將ibcs2 這一個kernel 模組載入。

你還要將你的/compat/ibcs2/dev 改成下面這樣：

```
lrwxr-xr-x  1 root  wheel           9 Oct 15 22:20 X0R@ -> /dev/null
lrwxr-xr-x  1 root  wheel           7 Oct 15 22:20 nfsd@ -> socksys
-rw-rw-r--  1 root  wheel           0 Oct 28 12:02 null
lrwxr-xr-x  1 root  wheel           9 Oct 15 22:20 socksys@ -> /dev/null
crw-rw-rw-  1 root  wheel    41,   1 Oct 15 22:14 spx
```

你只需要將socksys 轉向到/dev/null (請讀null(4)) 去騙過open & close 的動作。在-current 裡面的ibcs2 相關程式碼將會處理其餘的部份，這種作法比以前的方式乾淨太多了。假如你想要使用spx 方面的程式，在你的kernel 設定檔裡面加上SPX\_HACK。

**8. 我該使用那個版本的Microsoft FrontPage？**

Use the Port, Luke！在ports tree 中已經有一個包含FrontPage 的Apache 版本了。

### 9. 為什麼我無法在3.X-STABLE 機器上順利編好這個port?

如果你的FreeBSD 版本相較-CURRENT 或-STABLE 之下是很古早的話，或許你會需要一個升級ports 的工具，在 <http://www.FreeBSD.org/ports/../../ports/index.html>。如果你以將其更新卻仍無用，那麼一定是某人更動之後造成-CURRENT 才能用，-STABLE 無法用的情況。由於ports 內所收集的軟體在-CURRENT 或是-STABLE 上都要能用，所以請儘速送出關於此問題的蟲報告；請使用send-pr(1) 這個指令來送蟲報告。

### 10. 那裡可以找得到ld.so ?

有些a.out 格式的應用程式會需要a.out 格式的函式庫，Netscape Navigator 就是一個例子。不過用ELF 函式庫編起來的FreeBSD 預設並不會安裝舊的a.out 函式庫，所以您可能會得到類似找不到/usr/libexec/ld.so 的抱怨訊息。如果說您的系統有這安裝a.out 函式庫的必要，這些函式庫(compat22) 也能夠利用sysinstall(8) 來安裝。或者利用FreeBSD 原始碼來安裝：

```
# cd /usr/src/lib/compat/compat22# make install clean
```

如果你希望每次make world 時會自動更新compat22 函式庫，那麼修改/etc/make.conf，加入COMPAT22=YES。這些相容於古老版本的函式庫已經沒什麼在更新了，所以一般說來是不需要這樣的。

同時也請您看一下3.1-RELEASE 和3.2-RELEASE 的勘誤表(ERRATA)。

### 11. 我更新了系統原始碼，現在我要怎樣升級某個已經安裝上的ports ?

FreeBSD 本身並沒有自動升級ports 的工具，但有一些可以讓升級簡化一些小程式。你也可以自己裝上額外的工具來處理。

pkg\_version(1) 指令可以自動產生用來達到自動升級到ports tree 最新版本的script。

```
# pkg_version -c > /tmp/myscript
```

一定要在手動修改一下產生出來的script。目前的pkg\_version(1) 在script 最前面加入exit(1) 強迫你去修改它。

你應將執行script 所產生的輸出記錄下來，因為裡面會有記載某些尚未升級但已經更新的ports。不過你不一定要去升級它們。通常是因為有某個共用的函式庫已經改變版本號了，才要去重編一次那些使用到該函式庫的ports。

如果你的硬碟空間很夠，那麼可以用portupgrade 這個工具來做全自動處理。portupgrade 裡面也有一些小程式來簡化package 升級，它在sysutils/portupgrade。這個工具是用Ruby 這個語言寫的，所以並不適合加入到FreeBSD 的原始碼中，不過並不會因此讓某些人不用它。

如果你的系統一直都處於開機狀態，可利用periodic(8) 系統，每個星期產生一張需要升級的清單。只要在/etc/periodic.conf 加入weekly\_status\_pkg\_enable="YES" 就可以了。

### 12. 為什麼/bin/sh這麼的小？為什麼FreeBSD 不改用bash 或者是其他比較強悍的shell ?

因為POSIX 說，該要有這麼樣的一個shell 在才行。

比較繁瑣的答案：許多人需要寫可以跨很多平台的shell script。這也是為何POSIX 將shell 以及工具命稱都定義的非常詳細的緣故。大部份的script 都適用於Bourne shell，又因為有幾個重要的寫程式所用到的程式或者函式(make(1), system(3), popen(3)，還有在Perl 或者Tcl 裡面呼叫系統程式的地方) 都指定用Bourne shell。那麼因為Bourne Shell 如此的廣泛常用，那麼它的執行效率便很重要，快速是它決定性的要點之一，還要不佔太多記憶體。

目前的/bin/sh 已是我們嘔心瀝血之作，它已經盡量地符合標準規定。為了讓它非常小，我們拿掉了一些其他shell 有的方便功能。這也是為什麼ports 裡面還有很多强悍的shell，像是bash, scsh, tcsh 以及zsh。(你可以自己比較一下這些shell 執行時所佔的記憶體大小，去看看ps -u 列出來的“VSZ”和“RSS”這兩個欄位就知道了。)

**13. 為什麼Netscape 和Opera 要花好久的時間才能啓動？**

通常是因為你的DNS 沒有設定好。Netscape 跟Opera 在啓動的時候都會去檢查一下DNS。直到DNS 有回應，或者是斷定網路目前是斷線之後，它們才會顯示畫面出來。

## 章8. kernel 設定

### 1. 我想自訂kernel，這會很困難嗎？

不會！請查閱 使用手冊中的kernel 設定一節 ([../handbook/kernelconfig.html](#))。

**注：**我會建議你在你讓核心能正常工作後，做一個kernel.YYMMDD 日期形式的備份，同時也備份/module這個目錄至/modules.YYMMDD。這樣下次如果你很不幸的玩壞了設定，至少可以不需要使用最原始的kernel.GENERIC。如你正從一個GENERIC kernel 裡面不支援的控制器裡啟動時，這就顯得特別重要。

### 2. 我的核心因為\_hw\_float遺失而編譯失敗。該怎麼修正呢？

讓我猜看看，你把npx0 (詳見npx(4)) 從你的kernel 設定檔移除了，因為你沒有數學運算器，對嗎？錯了！:-) 這個npx0是必須要有的。就算你沒有數學運算器，你還是必須引入npx0 裝置。

### 3. 為什麼造出來的kernel 這麼大(10MB 以上) ？

這很有可能是因為，你把kernel 編成偵錯模式了。偵錯模式之下的kernel 裡面會存著偵錯用的許多符號，因此會大幅增加kernel 的大小。如果說你的FreeBSD 是3.0 以後的版本，這對於效能來說影響並不大，幾乎是沒有。而在系統會因某些原因panic 時，有個偵錯模式的kernel 在也挺有用的。

不過呢，如果你的磁碟空間很小，或者你就是不想用偵錯模式的kernel 的話，請確認以下事情：

- kernel 設定檔裡面沒有這一行：

```
makeoptions DEBUGS=-g
```

- 執行config(8) 時沒有加上-g 這個選項。

以上兩件事情都會讓你編出一個偵錯模式的kernel。但只要避免之，就可以編出一個正常的kernel，而你也會注意到，kernel 明顯的變小了；大部份的kernel 都差不多在1.5MB 到2MB 之間。

### 4. 為何出現了multi-port serial code 的中斷衝突？

當我編譯一個multi-port serial code 的核心時，它告訴我只有第一個被偵測到，其他的則因中斷衝突而跳過了，我該怎麼修正它？

這個問題是因為FreeBSD 使用內建程式碼避免因為硬體或軟體衝突導致kernel 過於肥大或無用。要修正這種情形的方法是除了一個port 外把其他所有的IRQ 設定都做保留。這裡有一個範例：

```
#
# Multiport high-speed serial line - 16550 UARTS
#
device sio2 at isa? port 0x2a0 tty irq 5 flags 0x501 vector siointr
device sio3 at isa? port 0x2a8 tty flags 0x501 vector siointr
device sio4 at isa? port 0x2b0 tty flags 0x501 vector siointr
device sio5 at isa? port 0x2b8 tty flags 0x501 vector siointr
```

## 5. 為什麼我一個kernel 都編不起來？甚至GENERIC 也不行？

這有很多種可能的原因：

- 你沒有用新的`make buildkernel` 與 `make installkernel` 這兩個方法來編，而正好你的系統原始碼的版本和正在執行的系統核心版本不一樣(像是，在跑4.0-RELEASE 的系統上嘗試著編4.3-RELEASE)。如果說你要升級系統的話，請務必去看看 `/usr/src/UPDATING` 這個檔案，特別注意最後面的“COMMON ITEMS” 這個小節。
- 你已經用上 `make buildkernel` 以及 `make installkernel` 了，但是在 `make buildworld` 時失敗了。可惜的是，`make buildkernel` 要成功，需要依賴 `make buildworld` 後造出來的一些檔案。
- 就算是在編 `FreeBSD-STABLE`，還是有可能你抓到了正在修改中，或著因為某些緣故而根本還沒改好的原始碼；雖然說 `FreeBSD-STABLE` 大部份的時候都是可以編的，但只有 `RELEASE` 才是保證可以編的。碰到這個問題時，再次更新原始碼並且再試試看。也有可能是放原始碼的伺服器出現某些問題，所以更新原始碼時也試試從不同伺服器來更新看看。

## 章9. 硬碟、檔案系統、Boot Loader

### 1. 我要怎麼把我的系統搬到新硬碟上面去？

理想的方式是先在新硬碟上重裝好作業系統，然後把使用者相關程式、資料搬過去就好。This is highly recommended if you have been tracking -STABLE for more than one release, or have updated a release instead of installing a new one. You can install booteasy on both disks with boot0cfg(8), and dual boot them until you are happy with the new configuration. Skip the next paragraph to find out how to move the data after doing this.

Should you decide not to do a fresh install, you need to partition and label the new disk with either `sysinstall`(FreeBSD 5.2 之前版本則是 `/stand/sysinstall`), or `fdisk(8)` and `disklabel(8)`. You should also install booteasy on both disks with boot0cfg(8), so that you can dual boot to the old or new system after the copying is done.

Now you have the new disk set up, and are ready to move the data. Unfortunately, you cannot just blindly copy the data. Things like device files (in `/dev`), flags, and links tend to screw that up. You need to use tools that understand these things, which means `dump(8)`. Although it is suggested that you move the data in single user mode, it is not required.

You should never use anything but `dump(8)` and `restore(8)` to move the root filesystem. The `tar(1)` command may work - then again, it may not. You should also use `dump(8)` and `restore(8)` if you are moving a single partition to another empty partition. The sequence of steps to use `dump` to move a partitions data 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, if you are going to move `root` to `/dev/ad1s1a`, with `/mnt` as the temporary mount point, it is:

```
# newfs /dev/ad1s1a
# mount /dev/ad1s1a /mnt
# cd /mnt
# dump 0af - / | restore xf -
```

Rearranging your 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/ad1s1a
# mount /dev/ad1s1a /mnt
# cd /mnt
# dump 0af - / | restore xf -
# cd var
# dump 0af - /var | restore xf -
```

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/ad1s1a
# newfs /dev/ad1s1d
# mount /dev/ad1s1a /mnt
# mkdir /mnt/var
# mount /dev/ad1s1d /mnt/var
# cd /mnt
# dump 0af - / | restore xf -
```

You might prefer `cpio(1)`, `pax(1)`, `tar(1)` to `dump(8)` for user data. At the time of this writing, these are known to lose file flag information, so use them with caution.

## 2. Will a “dangerously dedicated” disk endanger my health?

The installation procedure allows you to choose two different methods in partitioning your hard disk(s). The default way makes it compatible with other operating systems on the same machine, by using `fdisk` table entries (called “slices” in FreeBSD), with a FreeBSD slice that employs partitions of its own. Optionally, one can choose to install a boot-selector to switch between the possible operating systems on the disk(s). The alternative uses the entire disk for FreeBSD, and makes no attempt to be compatible with other operating systems.

So why is it called “dangerous”? A disk in this mode does not contain what normal PC utilities would consider a valid `fdisk` table. Depending on how well they have been designed, they might complain at you once they are getting in contact with such a disk, or even worse, they might damage the BSD bootstrap without even asking or notifying you. In addition, the “dangerously dedicated” disk’s layout is known to confuse many BIOSes, including those from AWARD (e.g. as found in HP Netserver and Micronics systems as well as many others) and Symbios/NCR (for the popular 53C8xx range of SCSI controllers). This is not a complete list, there are more. Symptoms of this confusion include the `read error` message printed by the FreeBSD bootstrap when it cannot find itself, as well as system lockups when booting.

Why have this mode at all then? It only saves a few kbytes of disk space, and it can cause real problems for a new installation. “Dangerously dedicated” mode’s origins lie in a desire to avoid one of the most common problems plaguing new FreeBSD installers - matching the BIOS “geometry” numbers for a disk to the disk itself.

“Geometry” is an outdated concept, but one still at the heart of the PC’s BIOS and its interaction with disks. When the FreeBSD installer creates slices, it has to record the location of these slices on the disk in a fashion that corresponds with the way the BIOS expects to find them. If it gets it wrong, you will not be able to boot.

“Dangerously dedicated” mode tries to work around this by making the problem simpler. In some cases, it gets it right. But it is meant to be used as a last-ditch alternative - there are better ways to solve the problem 99 times out of 100.

So, how do you avoid the need for “DD” mode when you are installing? Start by making a note of the geometry that your BIOS claims to be using for your disks. You can arrange to have the kernel print this as it boots by specifying `-v` at the `boot:` prompt, or using `boot -v` in the loader. Just before the installer starts, the kernel will print a list of BIOS geometries. Do not panic - wait for the installer to start and then use scrollback to read the numbers. Typically the BIOS disk units will be in the same order that FreeBSD lists your disks, first IDE, then SCSI.

When you are slicing up your disk, check that the disk geometry displayed in the `FDISK` screen is correct (ie. it matches the BIOS numbers); if it is wrong, use the `g` key to fix it. You may have to do this if there is absolutely nothing on the disk, or if the disk has been moved from another system. Note that this is only an issue with the disk that you are going to boot from; FreeBSD will sort itself out just fine with any other disks you may have.

Once you have got the BIOS and FreeBSD agreeing about the geometry of the disk, your problems are almost guaranteed to be over, and with no need for “DD” mode at all. If, however, you are still greeted with the dreaded `read error` message when you try to boot, it is time to cross your fingers and go for it - there is nothing left to lose. To return a “dangerously dedicated” disk for normal PC use, there are basically two options. The first is, you write enough NULL bytes over the MBR to make any subsequent installation believe this to be a blank disk. You can do this for example with

```
# dd if=/dev/zero of=/dev/rda0 count=15
```

Alternatively, the undocumented DOS “feature”

```
C:\> fdisk /mbr
```

will to install a new master boot record as well, thus clobbering the BSD bootstrap.

### 3. Which partitions can safely use Soft Updates? I have heard that Soft Updates on / can cause problems.

Short answer: you can usually use Soft Updates safely on all partitions.

Long answer: There used to be some concern over using Soft Updates on the root partition. Soft Updates has two characteristics that caused this. First, a Soft Updates partition has a small chance of losing data during a system crash. (The partition will not be corrupted; the data will simply be lost.) Also, Soft Updates can cause temporary space shortages.

When using Soft Updates, the kernel can take up to thirty seconds to actually write changes to the physical disk. If you delete a large file, the file still resides on disk until the kernel actually performs the deletion. This can cause a very simple race condition. Suppose you delete one large file and immediately create another large file. 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. You get an error that the partition does not have enough space, although you know perfectly well that you just released a large chunk of space! When you try again mere seconds later, the file creation works as you expect. This has left more than one user scratching his head and doubting his sanity, the FreeBSD filesystem, or both.

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 or corrupted. This risk is extremely small, but generally manageable. Use of IDE write caching greatly increases this risk; it is strongly recommended that you disable IDE write caching when using Soft Updates.

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. Files such as `/kernel` and the contents of `/etc` only change during system maintenance, or when users change their passwords. 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 you should be aware that it exists. If your system cannot tolerate this much risk, do not use Soft Updates on the root filesystem!

`/` is traditionally one of the smallest partitions. By default, FreeBSD puts the `/tmp` directory on `/`. If you have a busy `/tmp`, you might see intermittent space problems. Symlinking `/tmp` to `/var/tmp` will solve this problem.

### 4. What is inappropriate about my ccd?

The symptom of this is:

```
# ccdconfig -C
ccdconfig: ioctl (CCDIOCSET): /dev/ccd0c: Inappropriate file type or format
```

This usually happens when you are trying to concatenate the `c` partitions, which default to type `unused`. The `ccd` driver requires the underlying partition type to be `FS_BSDFFS`. Edit the `disklabel` of the disks you are trying to concatenate and change the types of partitions to `4.2BSD`.

### 5. Why can I not edit the `disklabel` on my `ccd`?

The symptom of this is:

```
# disklabel ccd0
(it prints something sensible here, so let us try to edit it)
# disklabel -e ccd0
(edit, save, quit)
disklabel: ioctl DIOCWDINFO: No disk label on disk;
use "disklabel -r" to install initial label
```

This is because the `disklabel` returned by `ccd` is actually a “fake” one that is not really on the disk. You can solve this problem by writing it back explicitly, as in:

```
# disklabel ccd0 > /tmp/disklabel.tmp
# disklabel -Rr ccd0 /tmp/disklabel.tmp
# disklabel -e ccd0
(this will work now)
```

### 6. Can I mount other foreign filesystems under FreeBSD?

FreeBSD supports a variety of other filesystems.

#### Digital UNIX

UFS CDRoms 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.

#### Linux

FreeBSD supports `ext2fs` partitions. See `mount_ext2fs(8)` for more information.

#### Windows NT®

FreeBSD includes a read-only NTFS driver. For more information, see `mount_ntfs(8)`.

#### FAT

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

FreeBSD also supports network filesystems such as NFS (see `mount_nfs(8)`), NetWare (see `mount_nwfs(8)`), and Microsoft-style SMB filesystems (see `mount_smbfs(8)`).

### 7. How do I mount a secondary DOS partition?

The secondary DOS partitions are found after ALL the primary partitions. For example, if you have an “E” partition as the second DOS partition on the second SCSI drive, you need to create the special files for “slice 5” in /dev, then mount /dev/dal5:

```
# cd /dev
# sh MAKEDEV dal5
# mount -t msdos /dev/dal5 /dos/e
```

注: You can omit this step if you are running FreeBSD 5.0-RELEASE or newer with devfs(5) enabled.

### 8. FreeBSD 有檔案加密系統嗎？

有啊！FreeBSD 5.0 起內建gbde(8)，而FreeBSD 6.0 又加上geli(8)。而較早期的版本，請多利用security/cfs port，謝謝。

### 9. How can I use the Windows NT loader to boot FreeBSD?

The general idea is that you copy the first sector of your native root FreeBSD partition into a file in the DOS/Windows NT partition. Assuming you name that file something like c:\bootsect.bsd (inspired by c:\bootsect.dos), you can then edit the c:\boot.ini file to come up with something like this:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows NT"
C:\BOOTSECT.BSD="FreeBSD"
C:\="DOS"
```

If FreeBSD is installed on the same disk as the Windows NT boot partition simply copy /boot/boot1 to C:\BOOTSECT.BSD. However, if FreeBSD is installed on a different disk /boot/boot1 will not work, /boot/boot0 is needed.

/boot/boot0 needs to be installed using sysinstall(FreeBSD 5.2 之前版本則是/stand/sysinstall) by selecting the FreeBSD boot manager on the screen which asks if you wish to use a boot manager. This is because /boot/boot0 has the partition table area filled with NULL characters but sysinstall copies the partition table before copying /boot/boot0 to the MBR.

**警告** Do not simply copy /boot/boot0 instead of /boot/boot1; you will overwrite your partition table and render your computer un-bootable!

When the FreeBSD boot manager runs it records the last OS booted by setting the active flag on the partition table entry for that OS and then writes the whole 512-bytes of itself back to the MBR so if you just copy /boot/boot0 to C:\BOOTSECT.BSD then it writes an empty partition table, with the active flag set on one entry, to the MBR.

**10. How do I boot FreeBSD and Linux from LILO?**

If you have FreeBSD and Linux on the same disk, just follow LILO's installation instructions for booting a non-Linux operating system. Very briefly, these are:

Boot Linux, and add the following lines to `/etc/lilo.conf`:

```
other=/dev/hda2
    table=/dev/hda
    label=FreeBSD
```

(the above assumes that your FreeBSD slice is known to Linux as `/dev/hda2`; tailor to suit your setup). Then, run `lilo` as `root` and you should be done.

If FreeBSD resides on another disk, you need to add `loader=/boot/chain.b` to the LILO entry. For example:

```
other=/dev/dab4
    table=/dev/dab
    loader=/boot/chain.b
    label=FreeBSD
```

In some cases you may need to specify the BIOS drive number to the FreeBSD boot loader to successfully boot off the second disk. For example, if your FreeBSD SCSI disk is probed by BIOS as BIOS disk 1, at the FreeBSD boot loader prompt you need to specify:

```
Boot: 1:da(0,a)/kernel
```

You can configure `boot(8)` to automatically do this for you at boot time.

The `Linux+FreeBSD mini-HOWTO` (<http://sunsite.unc.edu/LDP/HOWTO/mini/Linux+FreeBSD.html>) is a good reference for FreeBSD and Linux interoperability issues.

**11. How do I boot FreeBSD and Linux using BootEasy?**

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

If you are running Windows 95 and Linux this is recommended anyway, to make it simpler to get Linux booting again if you should need to reinstall Windows 95 (which is a Jealous Operating System, and will bear no other Operating Systems in the Master Boot Record).

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

You can not do that with the standard boot manager without rewriting it. There are a number of other boot managers in the `sysutils` ports category that provide this functionality.

**13. I have a new removable drive, how do I use it?**

Whether it is a removable drive like a Zip or an EZ drive (or even a floppy, if you want to use it that way), or a new hard disk, once it is installed and recognized by the system, and you have your cartridge/floppy/whatever slotted in, things are pretty much the same for all devices.

(this section is based on Mark Mayo's ZIP FAQ (<http://www.vmunix.com/mark/FreeBSD/ZIP-FAQ.html>))

If it is a ZIP drive or a floppy, you have already got a DOS filesystem on it, you can use a command like this:

```
# mount -t msdos /dev/fd0c /floppy
```

if it is a floppy, or this:

```
# mount -t msdos /dev/da2s4 /zip
```

for a ZIP disk with the factory configuration.

For other disks, see how they are laid out using `fdisk(8)` or `sysinstall(8)`.

The rest of the examples will be for a ZIP drive on `da2`, the third SCSI disk.

Unless it is a floppy, or a removable you plan on sharing with other people, it is probably a better idea to stick a BSD filesystem on it. You will get long filename support, at least a 2X improvement in performance, and a lot more stability. First, you need to redo the DOS-level partitions/filesystems. You can either use `fdisk(8)` or `sysinstall`(FreeBSD 5.2 之前版本則是 `/stand/sysinstall`), or for a small drive that you do not want to bother with multiple operating system support on, just blow away the whole FAT partition table (slices) and just use the BSD partitioning:

```
# dd if=/dev/zero of=/dev/rda2 count=2
# disklabel -Brw da2 auto
```

You can use `disklabel` or `sysinstall` to create multiple BSD partitions. You will certainly want to do this if you are adding swap space on a fixed disk, but it is probably irrelevant on a removable drive like a ZIP.

Finally, create a new filesystem, this one is on our ZIP drive using the whole disk:

```
# newfs /dev/rda2c
```

and mount it:

```
# mount /dev/da2c /zip
```

and it is probably a good idea to add a line like this to `/etc/fstab` (see `fstab(5)`) so you can just type `mount /zip` in the future:

```
/dev/da2c /zip ffs rw,noauto 0 0
```

#### 14. Why do I get `Incorrect super block` when mounting a CDROM?

You have to tell `mount(8)` the type of the device that you want to mount. This is described in the Handbook section on optical media ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html)), specifically the section Using Data CDs ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html#MOUNTING-CD](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html#MOUNTING-CD)).

#### 15. Why do I get `Device not configured` when mounting a CDROM?

This generally means that there is no CDROM in the CDROM drive, or the drive is not visible on the bus. Please see the Using Data CDs ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html#MOUNTING-CD](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html#MOUNTING-CD)) section of the Handbook for a detailed discussion of this issue.

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

Your CDROM probably uses the “Joliet” extension for storing information about files and directories. This is discussed in the Handbook chapter on creating and using CDROMs ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html)), specifically the section on Using Data CDROMs ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html#MOUNTING-CD](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html#MOUNTING-CD)).

**17. I burned a CD under FreeBSD and now I can not read it under any other operating system. Why?**

You most likely burned a raw file to your CD, rather than creating an ISO 9660 filesystem. Take a look at the Handbook chapter on creating CDROMs ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html)), particularly the section on burning raw data CDs ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html#RAWDATA-CD](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html#RAWDATA-CD)).

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

This is discussed in the Handbook section on duplicating data CDs ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html#IMAGING-CD](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html#IMAGING-CD)). For more on working with CDROMs, see the Creating CDs Section ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/creating-cds.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/creating-cds.html)) in the Storage chapter in the Handbook.

**19. Why can I not mount an audio CD?**

If you try to mount an audio CD, you will get an error like `cd9660: /dev/acd0c: Invalid argument`. This is because `mount` only works on filesystems. Audio CDs do not have filesystems; they just have data. You need a program that reads audio CDs, such as the `audio/xmcd` port.

**20. How do I mount a multi-session CD?**

By default, `mount(8)` will attempt to mount the last data track (session) of a CD. If you would like to load an earlier session, you must use the `-s` command line argument. Please see `mount_cd9660(8)` for specific examples.

**21. How do I let ordinary users mount floppies, CDROMs and other removable media?**

Ordinary users can be permitted to mount devices. Here is how:

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

```
# sysctl -w vfs.usermount=1
```
2. As `root` assign the appropriate permissions to the block device associated with the removable media.

For example, to allow users to mount the first floppy drive, use:

```
# chmod 666 /dev/fd0
```

To allow users in the group `operator` to mount the CDROM drive, use:

```
# chgrp operator /dev/acd0c
# chmod 640 /dev/acd0c
```

3. If you are running FreeBSD 5.X or later, you will need to alter `/etc/devfs.conf` to make these changes permanent across reboots.

As `root`, add the necessary lines to `/etc/devfs.conf`. For example, to allow users to mount the first floppy drive add:

```
# Allow all users to mount the floppy disk.
own      /dev/fd0      root:operator
perm     /dev/fd0      0666
```

To allow users in the group `operator` to mount the CD-ROM drive add:

```
# Allow members of the group operator to mount CD-ROMs.
own      /dev/acd0      root:operator
perm     /dev/acd0      0660
```

4. Finally, add the line `vfs.usermount=1` to the file `/etc/sysctl.conf` so that it is reset at system boot time.

All users can now mount the floppy `/dev/fd0` onto a directory that they own:

```
% mkdir ~/my-mount-point
% mount -t msdos /dev/fd0 ~/my-mount-point
```

Users in group `operator` can now mount the CDRom `/dev/acd0c` onto a directory that they own:

```
% mkdir ~/my-mount-point
% mount -t cd9660 /dev/acd0c ~/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.

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

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

You need to understand what `du` and `df` really do. `du` goes through the directory tree, measures how large each file is, and presents the totals. `df` just asks the filesystem 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 you delete the file, the file is not really removed from the filesystem until the program stops using it. The file is immediately deleted from the directory listing, however. You can see this easily enough with a program such as `more`. Assume you have a file large enough that its presence affects the output of `du` and `df`. (Since disks can be so large today, this might be a *very* large file!) If you delete this file while using `more` on it, `more` does not immediately choke and complain that it cannot view the file. The entry is simply removed from the directory so no other program or user can access it. `du` shows that it is gone — it has walked the directory tree and the file is not listed. `df` shows that it is still there, as the filesystem knows that `more` is still using that space. Once you end the `more` session, `du` and `df` will agree.

Note that Soft Updates can delay the freeing of disk space; you might need to wait up to 30 seconds for the change to be visible!

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)`.

**23.** How can I add more swap space?

In the Configuration and Tuning ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/config-tuning.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/config-tuning.html)) section of the Handbook, you will find a section ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/adding-swap-space.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/adding-swap-space.html)) describing how to do this.

**24.** 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 80GB as holding 76319MB.

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

**25.** 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%. Also, you will 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 the `-m` option in `tunefs(8)`.

# 章10. 系統管理

## 1. 系統起始設定檔在哪？

從2.0.5R 到2.2.1R，主要的設定檔是`/etc/sysconfig`。所有的選項都被指定在這個檔，而其他像`/etc/rc` (參見`rc(8)`) 和`/etc/netstart` 只是引用它。

觀察`/etc/sysconfig` 這個檔並修正其值以適合你的系統。這個檔用註解填滿以表示何處該放置什麼設定。

在post-2.2.1 以後及3.0，`/etc/sysconfig` 亦更名為一個更容易描述的檔名叫`rc.conf(5)`，並且語法簡化了些。`/etc/netstart` 亦更名為`/etc/rc.network` 因此所有的檔案都可以用`cp /usr/src/etc/rc* /etc` 來拷貝。

在3.1 以及，`/etc/rc.conf` 被移到`/etc/defaults/rc.conf`。千萬不要編輯這個檔！如果`/etc/defaults/rc.conf` 內有想要更動的項目，你應該將那一行的內容拷貝到`/etc/rc.conf`，然後再修改它。

例如FreeBSD 3.1 及以後的版本內，有一個DNS 伺服器`named`，而你想要啓動它。你所需要作的事就是：

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

想要在FreeBSD 3.1 及以後的版本中，啓動本地端服務的話，將shell script 置於`/usr/local/etc/rc.d` 目錄下。這些shell script 應該設定成可執行，並且檔名以`.sh` 結束。在FreeBSD 3.0 及更早的版本中，你應該直接編輯`/etc/rc.local` 檔。

`/etc/rc.serial` 用來初始化序列埠（像是鎖定埠的特性等）。

`/etc/rc.i386` 是Intel 專用設定，像是iBCS2 模擬或是PC 系統主控台設定。

## 2. 該如何簡單地新增帳號？

使用`adduser(8)` 指令。如果需要更複雜的使用方式，請用`pw(8)` 這個指令。

要再次移除使用者，使用`rmuser(8)` 指令。還有，`pw(8)` 也可以使用。

## 3. 在改完`crontab` 檔案後，為什麼老是收到這樣的訊息：`root: not found`？

通常都是因為編輯了系統的`crontab` (`/etc/crontab`) 然後就用`crontab(1)` 去安裝它：

```
# crontab /etc/crontab
```

這樣作是不對的。系統的`crontab` 和`crontab(1)` 所更新的使用者的`crontab` 格式並不一樣(`crontab(5)` 說明文件針對差異處有詳細的說明)。

如果你已經用這種方法，額外多出的`crontab` 只就是`/etc/crontab` 的拷貝，只是格式是錯誤的。可用以下的命令刪除：

```
# crontab -r
```

下次你編輯`/etc/crontab` 檔案的時候，你不用作任何動作去通知`cron(8)`，它自動會去偵測是否有更動。

如果你想要每天、每週、或是每月固定執行某些動作一次，也許加個shell script

在`/usr/local/etc/periodic`目錄下會更好，系統的cron會固定執行`periodic(8)`命令，它可將你的程式和其它的系統週期性工作一起執行。

這個錯誤的真正原因，是因為系統的`crontab`有一個額外的欄位，說明該命令要以什麼使用者身份執行。在FreeBSD的預設系統`crontab`中，所有的項目都是`root`。當這個`crontab`被當作是`root`的使用者`crontab`（它和系統的`crontab`是不一樣的），`cron(8)`會以為`root`字串是欲執行的命令的第一個字，但是實際上並沒有這樣的命令存在。

4. 為什麼我想要用`su`成為`root`時，會得到`you are not in the correct group to su root`的錯誤訊息？

這是一個安全特性。想要利用`su`成為`root`（或其它有`superuser`權限的帳號），你一定要在`wheel`群組內。如果沒有這個特性的話，任何人只要在系統裡有帳號，並且恰巧知道`root`的密碼，就可以取得`superuser`等級的權限以存取系統。有了這個特性，這樣的情況就不會發生；如果使用者不在`wheel`群組內的話，`su(1)`會讓他們連試著鍵入密碼的機會都沒有。

要讓某人可以利用`su`成為`root`的話，只要把他們放入`wheel`群組內即可。

5. 我在`rc.conf`還是某個起動檔案裡犯了錯誤，因為檔案系統變成唯讀的，我無法去編輯它。我該怎麼辦？

當電腦問你shell完整路徑名時，只要按`ENTER`，然後執行`mount /`以讀寫模式重新掛載根檔案系統。你也許需要執行`mount -a -t ufs`，將你慣用的文字編輯器所在的檔案系統掛載上來。如果你慣用的文字編輯器在網路檔案系統上的話，你必須先手動將網路設定起來，以便將網路檔案系統掛載上來，或是使用本地端檔案系統上的編輯器，例如`ed(1)`。

如果你想要使用像`vi(1)`或是`emacs(1)`等的全螢幕文字編輯器的話，你也需要執行`export TERM=cons25`，以便讓這些編輯器能夠從`termcap(5)`資料庫裡讀取正確的資料。

當你已經完成了這些步驟後，你可以照你平常修改文法錯誤的方式去編輯`/etc/rc.conf`檔案。在核心(kernel)啟動時所顯示的錯誤訊息，能夠告訴你檔案中哪一行有錯誤。

6. 為什麼我沒辦法設定我的印表機？

請參考一下Handbook中，有關列印的部份。它應該能夠解決你大部份的問題。請參考Handbook中的列印部份(`../handbook/printing.html`)。

有些印表機需要主機支援的驅動程式(host-based driver)才能執行任何列印功能。FreeBSD本身並不支援這些所謂的“WinPrinters”。如果你的印表機無法在DOS或Windows NT 4.0下執行，那它大概就是一台WinPrinter。你唯一能使用這樣的印表機的希望，就是試試`print/pnm2ppa`支不支援它了。

7. 我要怎麼樣修正我的系統所使用的鍵盤對映(keyboard mapping)？

請參考Handbook中的using localization (`../handbook/using-localization.html`) 章節，尤其是console setup (`../handbook/using-localization.html#SETTING-CONSOLE`) 章節。

## 8. 為什麼我在系統啟動時，得到unknown: <PNP0303> can't assign resources 的訊息？

以下是從FreeBSD-CURRENT 通信論壇的一篇文章中節錄出來的。

“無法指派資源(can't assign resources)” 訊息表示那些裝置是傳統的ISA 裝置，而核心中已經編入不認得PNP 的驅動程式。這些裝置包括鍵盤控制器，可程式化岔斷控制晶片，還有幾個標準設備。資源無法指派給這些裝置，是因為早已有驅動程式使用那些位址了。

—Garrett Wollman <wollman@FreeBSD.org>, 2001 年四月24 日

## 9. 為什麼user quotas 無法正常運作？

1. 可能你kernel 設定未加入quotas 支援(預設是無)。如果是這樣子的話，那麼請把下面這行加到kernel 設定檔內並重新編譯、安裝：

```
options QUOTA
```

細節部分，請參閱Handbook 內的 quotas

([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/quotas.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/quotas.html)) 章節。

2. 請不要直接在/ 打開quotas
3. 把quotas 檔放在它必須強迫置入的檔案系統內，舉例：

檔案系統	Quota 檔
/usr	/usr/admin/quotas
/home	/home/admin/quotas
...	...

## 10. FreeBSD 支援System V IPC 格式指令集？

是的，FreeBSD 支援System V-style IPC。這包括共享記憶體，訊息跟信號。你需要在你的kernel 設定檔內加入下列幾行以啟動它們。

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

注：在FreeBSD 3.2 以及之後的版本，這些選項已經是 *GENERIC* 核心的一部份，也就是說它們已經編進了你的系統中。

重新編譯並安裝。

## 11. 我該如何讓sendmail 透過UUCP 來遞送郵件？

跟隨FreeBSD 套裝而來的sendmail 設定是適合那些直接連上網際網路的站台。想透過UUCP 交換郵件的站台必須另外安裝sendmail 的設定檔案。

手動修改/etc/sendmail.cf 是絕對必要的。第8 版的sendmail 提供一個全新的入口以透過一些像m4(1) 的處理就能產生設定檔，這事實上是一個高層概念等級的技巧性設定。你應該可以在/usr/src/usr.sbin/sendmail/cf 以下裡使用它：

假如你不是用full sources 方式安裝系統，那麼sendmail 設定項目可能已經分散成好幾個來源分布檔在等著你，假設你已經mount 光碟機，做以下動作：

```
# cd /cdrom/src
# cat scontrib.?? | tar xzf - -C /usr/src contrib/sendmail
```

別驚慌，這只有數十萬個位元組的大小。在cf 目錄裡的README 可以提供一個m4 設定法的基本介紹。

以UUCP 遞送來說，建議你最好使用mailertable 特點。建構一個資料庫讓sendmail 可以使用它自己的路徑決策。

首先，你必須建立自己的.mc 檔。/usr/src/usr.sbin/sendmail/cf/cf 目錄是這些檔案的家。查看一下，已經有好幾個範例檔，假設你已經命名自己的檔叫foo.mc，你要做的只是把它轉換成一個有效的sendmail.cf：

```
# cd /usr/src/usr.sbin/sendmail/cf/cf
# make foo.cf
# cp foo.cf /etc/mail/sendmail.cf
```

一個典型的.mc 檔看起來可能像這樣：

```
VERSIONID(`Your version number')
OSTYPE(bsd4.4)

FEATURE(accept_unresolvable_domains)
FEATURE(nocanonify)
FEATURE(mailertable, `hash -o /etc/mail/mailertable')

define(`UUCP_RELAY', your.uucp.relay)
define(`UUCP_MAX_SIZE', 200000)
define(`confDONT_PROBE_INTERFACES')

MAILER(local)
MAILER(smtp)
MAILER(uucp)

Cw    your.alias.host.name
Cw    youruucpnodename.UUCP
```

accept\_unresolvable\_domains，nocanonify，和confDONT\_PROBE\_INTERFACES 特性將避免任何在遞送郵件時會用到DNS 的機會。UUCP\_RELAY 項目的出現理由很奇怪，就不要問為何了。簡單的放入一個網際網路上可以處理UUCP 虛擬網域位址的主機名稱；通常，你只需要在這裡填入你ISP 的信件回覆處(mail replay)。

你已經做到這裡了，你還需要這個叫/etc/mail/mailertable。如果你只有一個用來傳遞所有郵件的對外通道的話，以下的檔案就足夠了：

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#                                     uucp-dom:your.uucp.relay
```

另一個更複雜的例子看起來像這樣：

```
#
```

```
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#
horus.interface-business.de    uucp-dom:horus
interface-business.de         uucp-dom:if-bus
interface-business.de         uucp-dom:if-bus
heep.sax.de                   smtp8:%1
horus.UUCP                     uucp-dom:horus
if-bus.UUCP                    uucp-dom:if-bus
                               uucp-dom:
```

如你所見，這是某個真實檔案裡的一部份。首三行處理網域定址信件不應該被送出到內定路徑，而由某些UUCP鄰居(UUCP neighbor)取代的特殊情形，這是為了“縮短”遞送的路徑。下一行處理到本地乙太網路網域的信件讓它可以使SMTP來遞送。最後，UUCP鄰居提到UUCP虛擬網域的記載，允許一個uucp-neighbor !recipient推翻內定規則。最後一行則以一個單獨的句點做結束，以UUCP遞送到提供當你全世界性郵件閘門的UUCP鄰居。所有在uucp-dom: 關鍵字裡的節點名稱必須都是有效的UUCP鄰居，你可以用uuname命令去確認。

提醒你這個檔案在使用前必須被轉換成DBM資料庫檔案，最好在mailertable最上面用註解寫出命令列來完成這個工作。當你每次更換你的mailertable後你總是需要執行這個命令。

最後提示：如果你不確定某些特定的信件路徑可用，記得把-bt選項加到sendmail。這會將sendmail啓動在address test mode；只要按下0，接著輸入你希望測試的信件路徑位址。最後一行告訴你使用內部的信件代理程式，代理程式的會通知目的主機，以及(可能轉換的)位址。要離開此模式請按Control-D。

```
% sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 foo@example.com
canonify          input: foo @ example . com
..
parse            returns: $# uucp-dom $# your.uucp.relay $: foo < @ example . com . >
> ^D
```

## 12. 當我用撥接連上網路時該怎麼設定信件遞送？

如果你已經有一個固定的IP數字，你不需要調整任何內定值。設好你要指定的網路名稱，其他的sendmail都會幫你做完。

如果你拿到的是動態配置的IP數字而使用撥接ppp連接到網際網路，你可能已經在你的ISP信件主機上有一個信箱。假設你的ISP網域是example.net，你的使用者名稱是user。亦假設你稱自己的主機名稱是bsd.home而你的ISP告訴你可以使用relay.example.net當作信件回覆處。

為了從你的信箱接收信件，你將需要安裝取信程式以便從信箱取回信件。Fetchmail是一個不錯的選擇，因為它支援許多不同的通訊協定，通常你的ISP會提供POP3。如果你選擇使用user-ppp，你可以在連線到網路成功後自動抓取你的信件，只要在/etc/ppp/ppp.linkup裡面設定以下這項：

```
MYADDR:
!bg su user -c fetchmail
```

假使你正使用sendmail(如下所示)傳送信件到非本地帳號，置入以下命令：

```
!bg su user -c "sendmail -q"
```

在上面那項命令之後。這會強迫**sendmail** 在連接上網路後馬上開始處理mailqueue。

我假設你在bsd.home 機器上有一個user 的帳號。在bsd.home 機器上user 的家目錄裡建立一個.fetchmailrc 的檔案：

```
poll example.net protocol pop3 fetchall pass MySecret
```

無須贅言，這個檔除了user 外不應該被任何人讀取，因為它包含MySecret 這個密碼。

為了在寄信時有正確的抬頭from:，你必須告訴**sendmail** 使用user@example.net 而非user@bsd.home。你可能會希望告訴**sendmail** 從relay.example.net 送出所有信件，加快信件傳送。

以下的.mc 檔應能滿足你的要求：

```
VERSIONID('bsd.home.mc version 1.0')
OSTYPE(bsd4.4) dnl
FEATURE(nouucp) dnl
MAILER(local) dnl
MAILER(smtp) dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS('example.net') dnl
FEATURE(allmasquerade) dnl
FEATURE(masquerade_envelope) dnl
FEATURE(nocanonify) dnl
FEATURE(nodns) dnl
define('SMART_HOST', 'relay.example.net')
Dmbsd.home
define('confDOMAIN_NAME', 'bsd.home') dnl
define('confDELIVERY_MODE', 'deferred') dnl
```

如何轉換這個.mc 檔案到sendmail.cf 檔的詳細細節，請參考上一節。另外，在更新sendmail.cf 以後不要忘記重新啟動**sendmail**。

### 13. 除了Sendmail 外，還有哪些郵件伺服器可以使用呢？

Sendmail (<http://www.sendmail.org/>) 是FreeBSD 預設使用的郵件伺服器，但是你還是可以很容易地以其它郵件伺服器(例如，從port 安裝的郵件伺服器) 取代之。

port 裡有很多可供選擇的郵件伺服器，像mail/exim、mail/postfix、mail/qmail、mail/zmailer 等，就是幾個很受歡迎的選擇。

多樣選擇是好事，而且大家有許多郵件伺服器可以使用也被認為是好事；所以請避免在通信論壇裡問像“Sendmail 有比Qmail 好嗎？”這樣的問題。如果你真的很想問的話，請先到通信論壇archive 裡找一下。每一個郵件伺服器的優點與缺點，以前大概就已經討論好幾次了。

### 14. 我忘了root 密碼了！怎麼辦？

不要驚慌！只要重新啟動系統，在看到Boot: 時輸入boot -s 即可進入單使用者模式(在3.2-RELEASE 之前的版本請改用-s)。在問要使用哪個shell 時，按下ENTER。你會看到一個# 的提示號，輸入mount -u / 以重新掛上(mount) 你的根檔案系統可供讀/寫。執行passwd root 以更換root 密碼，然後執行exit(1) 繼續啟動程序。

**15. 我該怎麼讓Control-Alt-Delete 不會重新啓動系統？**

如果你是使用FreeBSD 2.2.7-RELEASE 或之後版本的syscons(系統內定的主控台驅動程式)，把下列這行放到kernel 設定檔內，然後重做一個新的核心：

```
options SC_DISABLE_REBOOT
```

若是使用FreeBSD 2.2.5-RELEASE 或之後版本的PCVT 主控台驅動程式，則以下列選項代替：

```
options PCVT_CTRL_ALT_DEL
```

其他更早期的FreeBSD 版本，請修改你正在使用的主控台鍵盤對應，並將所有boot 關鍵字以nop 取代。內定的鍵盤對應是在/usr/share/syscons/keymaps/us.iso.kbd。你可能需要明白的吩咐/etc/rc.conf 去讀取這個鍵盤對應以確保更動生效。當然如果你正在用適合你國籍的鍵盤對應，你應該編輯那一個。

**16. 我該怎麼把DOS 文字檔案重新格式化成UNIX 的？**

只要使用這個perl 命令：

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

file 就是要處理的檔案。這個修改是在內部完成，原始的檔案會儲存成副檔名為.bak 的檔案。

或者你可以使用tr(1) 這個命令：

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

dos-text-file 是包含DOS 文字的檔案，而unix-text-file 則是包含轉換的輸出結果。這比使用perl 還要快上一點點。

**17. 我該怎麼用名稱砍掉process？**

使用killall(1)。

**18. 為何在su 一直說我不在root 的ACL 裡？**

這個錯誤是因為Kerberos 分散認證系統。這個問題並不是很嚴重但是令人厭煩。你可以用-K 選項去執行su，或是像下個問題所描述的移除Kerberos。

**19. 我該怎麼移除Kerberos？**

要從系統裡移除Kerberos，重裝你正在執行的release 版本的bin distribution。如果你有CDROM，你可以mount cd(假設在/cdrom) 並執行：

```
# cd /cdrom/bin
# ./install.sh
```

或者你也可以將/etc/make.conf 裡的"MAKE\_KERBEROS" 選項全都拿掉，然後再build world。

**20. 我該怎麼增加系統的虛擬終端機？**

如果你有許多telnet，ssh，X 或是screen 使用者，你或許會用完虛擬終端機，這能教你怎麼加更多：

1. 建立並安裝一個新的kernel 並且把這一行

```
pseudo-device pty 256
```

加入到設定檔裡。

2. 執行這個命令：

```
# cd /dev
# sh MAKEDEV pty{1,2,3,4,5,6,7}
```

會造出256 個虛擬終端機的裝置節點。

3. 編輯/etc/ttys 並加入符合256 個終端機的行數。它們應該符合已經存在單項的格式，舉例來說，它們看起來像：

```
ttyqc none network
```

字母設計的順序是tty[pqrsPQRS][0-9a-v]，使用正規表示式。

4. 用新的kernel 重新啓動電腦就可以了。

**21. /dev/snd0 這個裝置做不出來！**

並沒snd 這個裝置的存在。這個名字是用來當作各個組成FreeBSD 聲音驅動程式組，諸如mixer，sequencer，以及dsp 的簡稱。

可以用以下的命令作出這些裝置：

```
# cd /dev
# sh MAKEDEV snd0
```

**22. 可以不用開機，重新讀取/etc/rc.conf、再次啓動/etc/rc 嗎？**

先進入單人使用者模式，然後再回到多使用者模式。

在主控台執行：

```
# shutdown now
(Note: without -r or -h)
```

```
# return
# exit
```

**23. 我想要把我的系統昇級到最新的-STABLE，但是得到的是-RC 或-PRERELEASE！怎麼了？**

簡單地說：那只是名字而已。RC 的意思是“Release Candiate，發行候選版本”，它表示新版本快要發行了。在FreeBSD 中，-PRERELEASE 通常是發行前的程式碼凍結的代名詞。(有些發行版本中，-BETA 標籤跟-PRERELEASE 是相同意思的。)

詳細地說：FreeBSD 從兩個地方分支出它的發行版本。主版號、點零、release (例如3.0-RELEASE 及4.0-RELEASE) 的，是從發展過程開始時分支出來的，通常稱為-CURRENT。有副版號的版本(例如3.1-RELEASE 或4.2-RELEASE)，是活躍的-STABLE 分支中的發行版本快照。從4.3-RELEASE 開始，每一個發行版本有它自己的分支，可為偏好極度保守的發展速度(通常只會作安全方面的更新)的人所用。

準備要製作發行版本時，其所在的分支會經過一定的程序。有一個是程式碼凍結。當程式碼凍結開始時，分支名稱會更名，以反映它快要成為一個發行版本了。舉個例子，如果原來的分支叫4.5-STABLE，它的名字會變成4.6-PRERELEASE 以表示程式碼已凍結，並且額外的發行前測試將要開始了。臭蟲更正仍可回報，以成為發行版本的一部份。當程式碼有了可成為發行版本的雛形時，它的名字就會變成4.6-RC，以表示發行版本快好了。進入RC 階段後，只有找到的最有影響的臭蟲才會被修正。當發行版本(本例中為4.6-RELEASE) 產生後，發行版本會有自己的分支，原分支會被更名為4.6-STABLE。

想要得知更多有關版本號碼與各CVS 分支的資訊，請參考Release Engineering ([../articles/releeng/article.html](http://www.freebsd.org/articles/releeng/article.html)) 一文。

#### 24. 我試著要安裝一個新的核心，但是無法chflags。我該怎麼解決？

簡單地說：你的securelevel 可能大於零。直接重新開機到單人模式，再安裝核心。

詳細地說：FreeBSD 在securelevel 大於零情況下，不允許變更系統旗標(system flags)。你可以用這個指令檢查你的securelevel：

```
# sysctl kern.securelevel
```

你沒有辦法降低securelevel；你必須啓動系統到單人模式以安裝核心，或是修改/etc/rc.conf 內的securelevel 再重新開機。請參考init(8) 說明文件，以取得更多有關securelevel 的資訊，還有/etc/defaults/rc.conf 和rc.conf(5) 說明文件，以取得更多有關rc.conf 的資訊。

#### 25. 在我的系統上，我無法變更時間超過一秒以上的範圍！我該怎麼辦？

簡單地講：你系統的securelevel 也許大於1。直接重新開機至單人模式，然後再修改時間。

詳細地說：在securelevel 大於1 的情況下，FreeBSD 不允許時間變動大於一秒。你可以用以下的命令來檢查目前的securelevel：

```
# sysctl kern.securelevel
```

你無法降低securelevel；你必須啓動電腦至單人模式下以修改時間，或是修改/etc/rc.conf 再重新開機。請參考init(8) 說明文件，以取得更多有關securelevel 的資訊，還有/etc/defaults/rc.conf 和rc.conf(5) 說明文件，以取得更多有關rc.conf 的資訊。

#### 26. 為什麼rpc.statd 用了256 megabytes 的記憶體？

不，那不是memory leak，而且它也不是真的用了256 Mbyte 的記憶體。它只是喜歡(意思就是總會這樣作) 將一狗票的記憶體映謝到它自己的位址空間，以方便作事。就技術而言，這樣並沒有什麼不對；這樣只是會讓top(1) 和ps(1) 嚇一大跳而已。

rpc.statd(8) 會將它的狀態檔案(位於/var ) 映射至它的位址空間裡；為了防止需要的時候再增大所導致的重新映射，它一次會使用相當大的大小。從程式碼來看的話就更明顯了，可以看到mmap(2) 的長度參數為0x10000000，它是IA32 架構上的十六分之一的定址空間，也就是256MB。

**27. 為什麼我沒辦法取消`schg` 檔案旗標？**

你正在一個提高了`securelevel` (也就是大於0) 的系統運作。降低`securelevel` 再試試看。請參考 FAQ 中對`securelevel` 的說明 和`init(8)` 說明文件。

**28. 為什麼近來的新版FreeBSD 預設無法利用`.shosts` 完成SSH 認證？**

為什麼近來新版FreeBSD `.shosts` 認證預設為取消的原因，是因為`ssh(1)` 預設不安裝為`suid` 成 `root`。要“修正”這點，你可以作下列的任何一件事：

- 要一勞永逸解決，請將`/etc/make.conf` 裡的`ENABLE_SUID_SSH` 設成`true`，然後再重新`build ssh` (或是執行`make world`)。
- 只作一時的修正的話，可以`root` 身份執行`chmod 4755 /usr/bin/ssh` 將`/usr/bin/ssh` 設成`4555`。然後將`ENABLE_SUID_SSH= true` 加入`/etc/make.conf` 裡，這樣下次`make world` 執行就會生效了。

**29. 什麼是`vnlru`?**

當系統達到上限`kern.maxvnodes` 時，`vnlru` 會清除並釋放`vnode`。這個核心執行緒大部份的時間都沒事作，只有當你有很大的記憶體，而且正在存取上萬個小檔案時，才會被啟動。

# 章11. X Window System 及 Virtual Consoles

## 1. 我想要執行X，我該怎麼做？

最簡單的方法就是在安裝系統的時候一併安裝。

然後看看xorgconfig(1)的文件，這個程式可以幫您設定Xorg 相關設定，使它能夠正確運用您的顯示卡、滑鼠等週邊。此外，還可以用xorgcfg(1)的圖形介面來做相關設定。

您或許也想試試看Xaccel server。詳情請看Xi Graphics 這一段。

## 2. 我試著要執行X, 但是當我鍵入startx時，得到KDENABIO failed (Operation not permitted) 錯誤。我該怎麼辦？

你的系統一定提高了securelevel，對不對？在一個提高了securelevel的系統上，是絕對無法起動X的。想知道為什麼，請參考init(8)說明文件。

所以這個問題變成：你還能怎麼辦。基本上你有兩種選擇：將你的securelevel設回零(通常在/etc/rc.conf裡面設定)，或是在啟動時執行xdm(1)(在securelevel被昇高前)。

請參考問：8. 以取得更多有關啟動時執行xdm(1)的資訊。

## 3. 為什麼我不能在X裡使用滑鼠？

如果您用的是syscons (內定的console 驅動程式)的話，您可以經由設定FreeBSD來讓它支援在每個virtual console 使用滑鼠。為了避免和X產生衝突，syscons使用了一個叫做/dev/sysmouse的虛擬裝置。所有滑鼠產生的event都會利用moused來寫到sysmouse這個裝置。如果您希望在一個或以上的virtual console上使用滑鼠，並且能夠使用X的話，請參考問：4. 並且設定好moused。

然後編輯/etc/XF86Config這個檔案，並且確認你有以下這幾行的設定。

```
Section          Pointer
Protocol         "SysMouse"
Device           "/dev/sysmouse"
....
```

以上的例子，適用於XFree86 3.3.2及其後的版本。用於更早的版本的，其Protocol應為MouseSystems。

有些人比較喜歡在設定X的時候用/dev/mouse這個裝置。如果您要讓它能夠正常工作的話，您就必須把/dev/mouse連結到/dev/sysmouse(請參考sysmouse(4)):

```
# cd /dev
# rm -f mouse
# ln -s sysmouse mouse
```

## 4. 滑鼠的滾輪，能在X裡面使用嗎？

可以。不過需要設定X client 端程式。請參考 Colas Nahaboo 的網頁(<http://www.inria.fr/koala/colas/mouse-wheel-scroll/>) (<http://www.inria.fr/koala/colas/mouse-wheel-scroll/>).

如果你要使用**imwheel** 程式，只要跟著下列步驟作即可。

### 1. 轉換滾輪event Translate the Wheel Events

**imwheel** 程式的運作原理，是將滑鼠的第四鍵與第五鍵轉換成按鍵event。因為如此，所以你必須讓滑鼠驅動程式將滑輪事件轉換成第四鍵與第五鍵event。有兩種方法可以達到目的，一是讓**moused(8)** 作轉換，二是讓X 本身去作event 轉換。

#### a. 使用**moused(8)** 來轉換滾輪Event

要讓**moused(8)** 來作event 轉換，只要在執行**moused(8)** 的命令列中加上`-z 4` 即可。舉個例子，如果你一般都是以**moused -p /dev/psm0** 來起動**moused(8)** 的話，只要改成**moused -p /dev/psm0 -z 4** 即可。如果你是在開機過程中利用**/etc/rc.conf** 來起動**moused(8)**，你可以在**/etc/rc.conf** 中將**moused\_flags** 上加`-z 4` 即可。

你現在需要讓X 知道你的滑鼠有五個按鍵，只要在**/etc/XF86Config** 中的“**Pointer**” 區塊中加上**Buttons 5** 這一行即可。例如，你可能在**/etc/XF86Config** 中有以下的“**Pointer**” 區塊：

**範例11-1.** 在**XFree86 3.3.x** 系列的**XF86Config** 設定檔的“**Pointer**” 區塊中，以**moused** 作轉換的滾輪鼠的設定範例

```
Section "Pointer"
    Protocol      "SysMouse"
    Device        "/dev/sysmouse"
    Buttons       5
EndSection
```

**範例11-2.** 在**XFree86 4.x** 系列的**XF86Config** 設定檔的“**InputDevice**” 區塊中，以**X Server** 作轉換的滾輪鼠的設定範例

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

**範例11-3.** 在“**.emacs**” 中，設定滾輪鼠的原生頁面滾動支援範例

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

#### b. 利用你的X Server 來作滾輪Event 轉換

如果你沒有執行**moused(8)**，或是你不想利用**moused(8)** 去作滾輪event 轉換，你可以改用X server 來作這樣的event 轉換。你得在 **/etc/XF86Config** 檔案中作幾個更動。第一，你要為你的滑鼠選擇適當的通訊協定。大多數的滾輪鼠都使用“**IntelliMouse**” 協定，不過**XFree86** 也支援其它的通訊協定，例如羅技的**MouseMan+** 滑鼠所用的“**MouseManPlusPS/2**”。當你選好之後，只要加進一行“**Pointer**” 區塊的 **Protocol** 變數即可。

第二，你要告訴X server 將捲動事件重新對映至滑鼠的第四和第五鍵。這可以利用ZAxisMapping 選項辦到。

舉個例子，如果你沒有使用moused(8)，而你有一個IntelliMouse 安裝在PS/2 滑鼠埠的話，你可以在/etc/XF86Config 裡使用以下的設定。

**範例11-4.** 在XF86Config 設定檔的“Pointer” 區塊中，以X Server 作轉換的滾輪鼠的設定範例

```
Section "Pointer"
    Protocol      "IntelliMouse"
    Device        "/dev/psm0"
    ZAxisMapping  4 5
EndSection
```

**範例11-5.** 在XFree86 4.x 系列的XF86Config 設定檔的“InputDevice” 區塊中，以X Server 作轉換的滾輪鼠的設定範例

```
Section "InputDevice"
    Identifier    "Mouse1"
    Driver        "mouse"
    Option        "Protocol" "auto"
    Option        "Device" "/dev/psm0"
    Option        "ZAxisMapping" "4 5"
EndSection
```

**範例11-6.** 在“.emacs” 中，設定滾輪鼠的原生頁面滾動支援範例

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

## 2. 安裝imwheel

接下來，從Ports 裡安裝imwheel。在x11 類別裡可以找到它，它可以將滾輪event 對映到鍵盤event。舉個例子，它可以在你將滾輪往前推時，送出一個Page Up 到你的應用程式去。Imwheel 利用一個設定檔，以便對應滾輪event 至鍵盤event，這樣它就可以在不同的應用程式中，送出不同的鍵盤按鍵。預設的imwheel 設定檔是在/usr/X11R6/etc/imwheelrc，如果你想編輯自訂的設定檔的話，可以將它複製到~/.imwheelrc，然後依你的需要修改它。設定檔的格式在imwheel(1) 裡面有詳細的說明。

## 3. 設定Emacs 與Imwheel 協同工作(選擇性)

如果你使用emacs 或是Xemacs 的話，那你需要在你的~/.emacs 檔案裡加上一小段設定。emacs 請加上這一段：

**範例11-7.** Imwheel 的Emacs 設定

```
;;; For imwheel
(setq imwheel-scroll-interval 3)
(defun imwheel-scroll-down-some-lines ()
  (interactive)
```

```
(scroll-down imwheel-scroll-interval))
(defun imwheel-scroll-up-some-lines ()
  (interactive)
  (scroll-up imwheel-scroll-interval))
(global-set-key [?\M-\C-\)] 'imwheel-scroll-up-some-lines)
(global-set-key [?\M-\C-\()] 'imwheel-scroll-down-some-lines)
;;; end imwheel section
```

**Xemacs** 則在`~/.emacs` 檔裡加上這一段：

#### 範例11-8. Imwheel 的Xemacs 設定

```
;;; For imwheel
(setq imwheel-scroll-interval 3)
(defun imwheel-scroll-down-some-lines ()
  (interactive)
  (scroll-down imwheel-scroll-interval))
(defun imwheel-scroll-up-some-lines ()
  (interactive)
  (scroll-up imwheel-scroll-interval))
(define-key global-map [(control meta \)] 'imwheel-scroll-up-some-lines)
(define-key global-map [(control meta \())] 'imwheel-scroll-down-some-lines)
;;; end imwheel section
```

#### 4. 執行Imwheel

安裝之後，你可以直接在`xterm` 裡鍵入`imwheel` 命令以起動它。它會以背景執行，並且馬上發揮效用。如果你確定要直接使用**imwheel**，只要把它加進你自己的`.xinitrc` 或`.xsession` 內檔案即可。你可以不管**imwheel** 所送出來有關PID 檔案警告。那些警告只對Linux 版的**imwheel** 有效而已。

#### 5. 為什麼X Window 的選單和對話框不能正常運作？

把Num Lock 關掉試試。

如果您的Num Lock 在開機時的預設值是開著的話，您必須把下列這行放到`XF86Config` 設定檔中的Keyboard 部份。

```
# Let the server do the NumLock processing. This should only be
# required when using pre-R6 clients
  ServerNumLock
```

#### 6. 什麼是virtual console？我要怎麼做才能用多一點？

簡單來說，**virtual console** 就是可以讓您不必做太多複雜的設定如使用網路或執行X，而在同一臺機器上同時做好幾件事的方法。

當啓動系統並顯示完所有開機訊息之後，您就會在螢幕上看到一個**login** 的提示符號。在這個時候您就可以輸入您的**login name** 以及**password**，然後就可以在第一個**virtual console** 上開始工作了(或者開始玩！)。

在某些情況下，您可能會想要作其他的工作，例如說是看看您正在執行的程式的說明文件，或是當您在FTP傳輸的等待時間中看看您的郵件。您只需要按Alt-F2 (按住Alt 鍵不放，並按下F2 鍵)，然後您就會在第二個“virtual console”上看到一個login 提示符號！當您想要回到原來的工作時，請按Alt-F1。

FreeBSD 在安裝時的預設值是使用三個virtual console (3.3-RELEASE 後為八個)，您可以用Alt-F1，Alt-F2，以及Alt-F3 在它們之間做切換。

如果您想要多一點virtual console 的話，您只需要編輯/etc/ttys 這個檔(請參考ttys(5))，在“Virtual terminals” 這個註解後面加入ttyv4 到ttyvc 的欄位：

```
# Edit the existing entry for ttyv3 in /etc/ttys and change
# "off" to "on".
ttyv3  "/usr/libexec/getty Pc"          cons25  on  secure
ttyv4  "/usr/libexec/getty Pc"          cons25  on  secure
ttyv5  "/usr/libexec/getty Pc"          cons25  on  secure
ttyv6  "/usr/libexec/getty Pc"          cons25  on  secure
ttyv7  "/usr/libexec/getty Pc"          cons25  on  secure
ttyv8  "/usr/libexec/getty Pc"          cons25  on  secure
ttyv9  "/usr/libexec/getty Pc"          cons25  on  secure
ttyva  "/usr/libexec/getty Pc"          cons25  on  secure
ttyvb  "/usr/libexec/getty Pc"          cons25  on  secure
```

您想用幾個就設幾個。您設越多virtual terminal，它們就用掉越多系統資源；如果您只有不到8MB 的記憶體的話，這影響就大了。您可能也會想把secure 換成insecure。

**重要:** 如果您想要執行X 的話，您必須為它保留(或關掉) 至少一個virtual terminal。這就是說，如果您想在按十二個Alt 功能鍵時都有login 提示符號，而且又在同一部電腦上也想執行X 的話，那麼這真是太不幸了- 您只能用十一個。

取消一個console 最簡單的方法就是把它關掉。舉例來說，如果您像上面講的一樣設定了全部的12 個terminal 並且想要執行X，您必需把virtual terminal 12 從：

```
ttyvb  "/usr/libexec/getty Pc"          cons25  on  secure
```

設成：

```
ttyvb  "/usr/libexec/getty Pc"          cons25  off secure
```

如果您的鍵盤只有10 個功能鍵的話，您就要改成這樣：

```
ttyv9  "/usr/libexec/getty Pc"          cons25  off secure
ttyva  "/usr/libexec/getty Pc"          cons25  off secure
ttyvb  "/usr/libexec/getty Pc"          cons25  off secure
```

(您也可以直接把這幾行砍掉。)

一旦您改了/etc/ttys，下一個步驟就是要確定您有足夠的virtual terminal 裝置。最簡單的方法就是：

```
# cd /dev
# sh MAKEDEV vty12
```

再過來，想要啓動這些virtual console 最簡單(也是最乾淨) 的做法就是重開機。然後，如果您不想重開機的話，您可以把X Window 關掉，然後用root 的身份執行下列指令：

```
# kill -HUP 1
```

當您執行這個命令前，您一定要完全把X Window 關掉。如果您不這麼做的話，您的系統可能會在您執行kill 命令後出現當掉或鎖死的情況。

### 7. 我要怎麼從X 切換到virtual console ？

請用**Ctrl+Alt+F<sub>n</sub>** 以切回至virtual console 。**Ctrl+Alt+F1** 可以切回至第一個virtual console 。

當你切回至文字console 後，你就可以使用一般**Alt+F<sub>n</sub>** 按鍵組合，在各console 之間切換。

要回到X 的話，你必須切回至執行X 的virtual console 。如果你是從命令列裡起動X 的話(例如使用startx 指令)，那麼X 會依附在下一個未使用的virtual console ，而不是它被起動的**文字console** 。如果你有八個使用中的virtual terminal ，那麼X 就會在第九個上執行，你就可以使用**Alt+F9** 以返回至X 中。

### 8. 我要怎麼做才能在開機時啓動XDM ？

有兩種方法可以啓動 xdm (<http://www.FreeBSD.org/cgi/man.cgi?manpath=xfree86&query=xdm>) 。一種方法是從/etc/ttys 來啓動，可以參考ttys(5) 說明文件中的範例；另一種方法是在rc.local (請參考rc(8)) 執行xdm ，或是在/usr/local/etc/rc.d 放一個x.sh 。這兩種方法都是合法的，如果您試某一種方法無效的話，您可以試試另外一種。這兩種方法的結果是一樣的：X 會顯示一個圖形化的login: 提示。

用ttys 的方法的優點，在於指明了X 在啓動時到底是用那個vty ，並且將logout 時重新啓動X server 的責任丟給init 。rc.local 的方法則是在當啓動X 出了問題時，可以很輕鬆地把xdm 殺掉來解決問題。

如果是用rc.local 的方法，在執行xdm 時您不能加任何參數（也就是跑成daemon） 。必須在getty 執行後才能啓動xdm ，否則getty 和xdm 會互相衝突而鎖住console 。最好的方式是在script 中加個sleep ，讓它暫停10 秒鐘左右，接著才執行xdm 。

如果你是從/etc/ttys 啓動xdm ，xdm 與getty(8) 仍有機會互相衝突。一個避免它的方法，就是在/usr/X11R6/lib/X11/xdm/Xservers 檔案中加入vt 數字。

```
:0 local /usr/X11R6/bin/X vt4
```

上面的例子中，會指示X server 在/dev/ttyv3 中執行。請注意數字是差一的。X server 從一開始數vty ，而FreeBSD 核心則是從零開始數vty 的。

### 9. 為什麼當我執行xconsole 時，發生了Couldn't open console 的錯誤？

如果你是用startx 來啓動startx 的話，/dev/console 的權限並不會改變，結果就是xterm -C 和xconsole 這類的程式無法正常執行。

這一切的問題，都是因為console 的權限是採用系統預設值。在一個多使用者的系統裡，我們不希望每個使用者都可以直接寫入系統console 。如果使用者是從機器的VTY 直接login 的話，那麼fctab(5) 可以解決這類的問題。

簡單地說，請確保/etc/fctab (請參考fctab(5)) 這個檔案中的這一行沒有被註解掉：

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

這一行設定的存在可以確保從/dev/ttyv0 登入的使用者可以控制console 。

**10.** 以前我可以以一般使用者執行XFree86，為什麼現在它說我一定要以root才能執行？

所有的X server 都需要以root 來執行，才能直接存取你的視訊硬體。舊版本的XFree86 (<= 3.3.6) 會自動將所附的server 以利用root 身份執行的方式安裝起來(setuid 為root)。由於X server 都是體積龐大又複雜的程式，顯而易見地，這是一個安全上的災難。因為這個原因，新版的XFree86 就不將這些server 以setuid 為root 的方式安裝。

很明顯地，我們完全無法接受將X server 以root 的身份執行。起碼就安全上不是個好主意。有兩種方法可以以一般使用者的身份使用X。第一是利用xdm 或是其它的display manager (例如kdm)；第二是利用Xwrapper。

xdm 是一個處理圖形界面登入的daemon。它通常在開機時執行起來，而且負責對使用者作身份認證，以及起動使用者的工作環境；它可說是圖形環境下的getty(8) 與login(1) 的對應程式。想得知更多有關xdm 的資訊，請參考 XFree86 文件 (<http://www.xfree86.org/support.html>)，以及其FAQ 項目。

Xwrapper 是X server 的包裝程式；它可以讓一般使用者可以手動起動X server 的小工具，而還能維持一定的安全環境。它會檢查傳入的命令列參數，如果沒問題的話，就起動適當的X server。如果你因為某種理由而不想執行display manager 的話，它是為你而設定的。如果你安裝了完整的ports，你可以在/usr/ports/x11/wrapper 中找到它。

**11.** 我的PS/2 滑鼠在X 中有點不正常？

您的滑鼠和您的滑鼠驅動程式可能存在有不同步的現象。

在2.2.5 以及更早的系統裡，從X 切到virtual terminal 然後再切回來就可以使它們重新做同步的動作。如果這個問題常常發生的話，您可以在您的kernel 設定檔中加入下面這個選項然後重新編譯：

```
options PSM_CHECKSYNC
```

如果您沒有建立kernel 的經驗，請看自訂核心 這一節。

加上這個選項以後，滑鼠和滑鼠驅動程式間的不同步問題應該就比較不會出現了。如果這個問題仍然存在的話，在移動滑鼠時按按滑鼠按鍵可以使滑鼠和滑鼠驅動程式重新做同步的動作。

該注意的是這個選項並不是對每一個系統都有效，它可能會讓接在PS/2 滑鼠位置的ALPS GlidePoint 裝置失去“tap” 這項功能。

在2.2.6 及其後的版本，同步的確認已經有了較好的解決辦法，而且這些都已經是PS/2 滑鼠驅動程式的標準了。這個方法也可以在GlidePoint 上正常工作。(因為確認的程式碼已經成為一個標準功能，所以在這些版本中我們不在提供PSM\_CHECKSYNC 的選項了。) 不過在極少數的案例中，這些驅動程式會誤報同步性錯誤，然後您就會看到這樣的核心訊息：

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

然後您就會發現您的滑鼠不能正常運作了。

如果您發生了這樣的狀況，您必須藉由把PS/2 滑鼠驅動程式的flag 設成0x100 來把同步確認的程式碼給取消掉。然後在開機提示符號時用-c 選項來進入UserConfig：

```
boot: -c
```

然後，在UserConfig 命令列中鍵入：

```
UserConfig> flags psm0 0x100
UserConfig> quit
```

## 12. 我的PS/2 滑鼠不能透過MouseSystem 來運作？

有一些報告指出某些廠牌的PS/2 滑鼠只能在“高解析度”狀態下才能運作。如果不是的話，滑鼠游標會常常跑到螢幕的左上角去。

非常不幸的，這個問題在2.0.X 和2.1.X 下是無解的。在2.2 到2.2.5 版，您可以對/sys/i386/isa/psm.c 使用下列patch 然後重建您的核心。如果您沒有建立kernel 的經驗，請看自訂核心 這節。

```
@@ -766,6 +766,8 @@
     if (verbose >= 2)
         log(LOG_DEBUG, "psm%d: SET_DEFAULTS return code:%04x\n",
             unit, i);
+   set_mouse_resolution(sc->kbd, PSMD_RES_HIGH);
+
+   #if 0
     set_mouse_scaling(sc->kbd);          /* 1:1 scaling */
     set_mouse_mode(sc->kbd);            /* stream mode */
```

在2.2.6 及以後的版本，在PS/2 滑鼠驅動程式中設定0x04 的flag 會把滑鼠設成高解析度模式。在開機提示符號時用-c 選項來進入UserConfig：

```
boot: -c
```

然後，在UserConfig 的命令列中鍵入：

```
UserConfig> flags psm0 0x04
UserConfig> quit
```

前一節有提到另一個可能導致滑鼠問題的原因。

## 13. 當我建立X 程式時，imake 說它找不到Imake.tmpl。它在哪兒？

Imake.tmpl 是Imake 套件的一部份，Imake 是標準的建立X 程式的工具。Imake.tmpl 和其他數個header file 一樣是建立X 程式的必要檔案，您可以在X prog distribution 中找到它們。您可以用sysinstall(FreeBSD 5.2 之前版本則是/stand/sysinstall) 來安裝或是直接從X distribution 中手動安裝。

## 14. 我在build 一個X 應用程式，它需要XFree86 3.3.x，但是我已經安裝XFree86 4.x 了。我該怎麼辦？

要告訴port 在編譯程式時，使用XFree86 4.x 函式庫，您可以在/etc/make.conf 裡(如果你沒有這個檔，請建立它) 加上下面這一行：

```
XFREE86_VERSION= 4
```

## 15. 我要怎麼做才能設定左撇子用的滑鼠？

在您的.xinitrc 或是.xsession 中執行xmodmap -e "pointer = 3 2 1" 的指令。

**16. 要如何安裝splash 圖形顯示程式？那裡可以找得到呢？**

就在發行FreeBSD 3.1 之前，我們加進了在開機時顯示“splash”圖形的新功能。目前用來顯示在螢幕上的檔案必須是256 色的點矩陣圖形(\*.BMP) 或ZSoft PCX(\*.PCX) 的格式。除此之外，解析度也必須在320x200 以下，才能和標準VGA 顯示卡搭配使用。如果您編譯kernel 時有加入VESA 支援，那麼最大解析度可以到1024x768。注意VESA 的支援需要加入VM86 這個kernel 選項。VESA 支援實際上可以在編譯kernel 時加入VESA 選項、或載入VESA 的kld module 來達成。

您要修改控制FreeBSD 開機步驟的設定檔，才能使用splash 顯示圖形的功能。設定檔在發行FreeBSD 3.2 前有些更動，所以現在有兩個方法可以載入splash 的功能：

- **FreeBSD 3.1**

先選出用來顯示在螢幕上的圖形，3.1 版只支援Windows 的點矩陣格式。選好了您要的圖檔後，將它拷到 /boot/splash.bmp。接著把下面幾行加入/boot/loader.rc 中：

```
load kernel
load -t splash_image_data /boot/splash.bmp
load splash_bmp
autoboot
```

- **FreeBSD 3.2+**

除了加入對PCX 檔案的支援外，FreeBSD 3.2 也改進了開機程序的設定方式。如果您願意的話，可以用上述FreeBSD 3.1 的方法，將splash\_bmp 換成splash\_pcx 來載入PCX 檔案即可。如果想用新的設定方式，您的/boot/loader.rc 必須包括這幾行：

```
include /boot/loader.4th
start
```

還需要一個包含以下幾行的/boot/loader.conf ：

```
splash_bmp_load="YES"
bitmap_load="YES"
```

這是假設您用/boot/splash.bmp 來當作splash 的螢幕顯示。如果想用PCX 的檔案，把它拷成/boot/splash.pcx，如上述做出/boot/loader.rc，再將這幾行加到/boot/loader.conf 中：

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

現在就只剩下splash 用來顯示的圖檔，您可以在<http://www.baldwin.cx/splash/> 找到各種樣品。

**17. 我能在X 裡使用鍵盤上的Windows ® 嗎？**

可以。你所要作的，就是利用xmodmap(1) 去定義你想使用的功能。

假設所有的“Windows” 都是標準的，那它們的keycode 分別為：

- 115 - Windows 鍵，在左手的Ctrl 與Alt 鍵之間
- 116 - Windows 鍵，在Alt-Gr 鍵右邊
- 117 - 選單鍵，右手的Ctrl 鍵左邊

要讓左邊的Windows 鍵印出一個逗點，試試這個：

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

你可能要重跑你的windows manager，才會有動作。

要讓Windows 鍵的對映在每次X 起動時自動設定好，你可以在你的~/.xinitrc 裡加上xmodmap，或是最好建立一個~/.xmodmaprc 檔案，裡面每一行就是一個xmodmap 的選項，然後在你的~/.xinitrc 裡加上：

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

這一行。

例如，你想要將這三個鍵各對映到F13、F14 和F15。這讓你能在你的程式或是window manager 內將其對應到便利的功能上，等一下我們會示範。

把這些放進~/.xmodmaprc 裡：

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

假如你用fvwm2 的話，你可以作這樣的對映，讓F13 能夠讓游標所在的視窗縮成小圖示(或是反過來)。F14 讓游標所在的視窗變成最上層的視窗，或是退到下層去(如果它已經是最上層了的話)。F15 則將Workplace (application) 選單叫出來，即使游標不在桌面上。當你沒有可見的桌面區域時，這個功能就相當地方便(而且按鍵上的圖案和這個功能相吻合)。

以下的~/.fvwmrc 設定可作出前述的功能。

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

### 18. 我要怎麼樣才能得到OpenGL 的3D 硬體加速功能？

3D 加速功能的有無，視你所使用的XFree86 版本與顯示晶片的型號而定。如果你的NVIDIA 晶片的話，請去FreeBSD NVIDIA Driver Initiative (<http://nvidia.netexplorer.org/>) 網頁看看，那裡有在XFree86-4 上使用NVIDIA 晶片的3D 加速功能的討論。XFree86-4 上的其它顯示卡廠牌硬體加速功能的資訊，包括Matrox G200/G400, ATI Rage 128/Radeon, 3dfx Voodoo 3, 4, 5, 以及Banshee，可在XFree86-4 Direct Rendering on FreeBSD (<http://gladstone.uoregon.edu/~eanholt/dri/>) 網頁上找到。XFree 3.3 的使用者可以使用Utah-GLX port，它可以在[graphics/utah-glx](http://graphics/utah-glx) 找到。使用它可以在Matrox Gx00, ATI Rage Pro, SiS 6326, i810, Savage, 以及舊的NVIDIA 上得到有限的OpenGL 加速。

## 章12. Networking

### 1. 我應該到哪邊找有關無磁碟開機 “diskless booting” 的資料？

“Diskless booting” 就是讓FreeBSD主機從網路上開機，並且從網路上的server上讀取其他必要的檔案，而非由主機的硬碟上取得這些檔案。詳細的資料可以參考FreeBSD手冊的無磁碟開機篇 ([../handbook/diskless.html](http://handbook/diskless.html))。

### 2. FreeBSD的主機可以當作某個網路上的路由器(router)嗎？

是的。請參考FreeBSD手冊的網路進階篇 advanced networking ([../handbook/routing.html](http://handbook/routing.html))，尤其是路由與閘道器 routing and gateways ([../handbook/routing.html#DEDICATED-ROUTER](http://handbook/routing.html#DEDICATED-ROUTER))的部分。

### 3. 我可以透過FreeBSD將我的Win95機器連上Internet嗎？

基本上，會問這種問題的人在家裡至少有兩台電腦，一台跑FreeBSD另外一台跑Win95；這個想法是將FreeBSD主機連上Internet，然後透過這台FreeBSD主機，讓跑Win95的電腦能夠上網。這個問題算是前一個問題的一個特例

... 答案是：可以的！在FreeBSD 3.x版中，使用者模式(user-mode)的ppp(8)包含了-nat選項。如果你在/etc/rc.conf使用-nat選項並設定gateway\_enable為YES，以這種設定啟動ppp(8)，並且正確的設定你的Windows主機的話，這個做法應該是可以正常使用的。

關於本主題更詳細的資料可以參考Steve Sims所撰寫的Pedantic PPP Primer ([../ppp-primer/index.html](http://ppp-primer/index.html))一文。

如果你使用的是核心模式(kernel-mode)PPP，或者你有區域連線(Ethernet connection)可通達Internet的話，你將需要使用natd(8)。請查閱FAQ中關於natd的部分。

### 4. FreeBSD支援SLIP和PPP嗎？

是的。你可以查查man pages中關於slattach(8)，sliplogin(8)，ppp(8)，以及pppd(8)的部分。ppp(8)及pppd(8)提供進出雙向連線的支援，另外sliplogin(8)專門提供進入連線的支援，而slattach(8)專門提供向外連線的支援。

如果你需要更進一步的資料的話，請查閱FreeBSD手冊中關於PPP與SLIP的說明 ([../handbook/ppp-and-slip.html](http://handbook/ppp-and-slip.html))。

如果你只能夠過“shell account”連線到Internet的話，你也許可以試試net/slirp這個套件程式。這個套件程式可以提供你的電腦直接連上某些(限定的)服務連線，如ftp及http等等。

### 5. FreeBSD支援NAT或Masquerading嗎？

如果你有一個近端的子網路(有一台以上的機器)，但是你的Internet provider卻只分配一個IP number給你(或者你只分配到一個動態的IP number)，你可以參考natd(8)這個程式。natd(8)讓你可以透過這一個IP number讓整個子網路的電腦都能連上internet。

ppp(8)這個程式也提供類似的功能，如果你指定-nat選項。alias library (libalias(3))在這兩個處理方式中都會被使用到。

## 6. 我如何將兩台FreeBSD 主機用平行埠(parallel line) 透過PLIP 連線？

請參考手冊中關於PLIP section (`../handbook/plip.html`) 的部分。

## 7. 我沒有辦法建立/dev/ed0 這個device，為什麼？

因為不需要！在Berkeley 網路架構中，只有kernel 程式碼可以直接存取網路界面卡。請參考`/etc/rc.network` 這個檔案和manual pages 取得與其他不同網路程式。更進一步的資訊：如果你覺得你完全搞混了的話，您應該找一本與其他BSD 相關作業系統網路管理有關書來參考；除了少數顯著的不同外，FreeBSD 的網路管理基本上和SunOS 4.0 和Ulrix 是一樣的。

## 8. 我如何建立Ethernet aliases？

如果你的alias 位址跟你目前網路介面的位址在同一個子網路下的話，加入一個`netmask 0xffffffff` 在你的`ifconfig(8)` command-line，範例如下：

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

不然的話，就如同加入一個新的網路位址一樣輸入你的網路位址與子網路遮罩：

```
# ifconfig ed0 alias 172.16.141.5 netmask 0xfffff00
```

## 9. 我如何指定我的3C503 使用其他不同的network port？

如果您想使用其他的port，你必須在`ifconfig(8)` 的命令中指定額外的參數。內定的port 是`link0`。要使用AUI port 代替BNC port 的話，改用`link2`。這些flags 應該改變`ifconfig_*` 的變數來指定，你可以在`/etc/rc.conf` 這個檔案裡面找到(請參考`rc.conf(5)`)。

## 10. 為什麼我在使用FreeBSD 的NFS 時出現問題？

我們用含蓄一點的說法，某些PC 的網路卡比其他的好，這種狀況在造成NFS 這種對網路敏感的程式有時會出現問題。

參考 the Handbook entry on NFS (`../handbook/nfs.html`) 以獲得這個主題的更多資訊。

## 11. 為什麼我不能NFS-mount Linux 的機器？

某些版本的Linux NFS 程式碼只接受privileged port 的mount request；試用這行指令看看

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

## 12. 為什麼我不能NFS-mount Sun 的機器？

跑SunOS 4.X 的Sun 工作站只接受來自privileged port 的mount request；試用這行指令看看

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

13. 為什麼mountd 一直鬼叫說can't change attributes 而且我一直看到bad exports list 這個訊息在我的FreeBSD NFS 伺服器上？

這個問題最常發生的原因是在於不了解/etc/exports 的正確格式。請詳讀exports(5) 以及手冊中關於NFS (../handbook/nfs.html) 的部分，特別是configuring NFS (../handbook/nfs.html#CONFIGURING-NFS)這一段。

14. 為什麼我在使用PPP 連線到NeXTStep 機器時有問題？

把TCP extensions 取消，這個設定在/etc/rc.conf 裡面(參考rc.conf(5)) 把以下這個值設成NO：

```
tcp_extensions=NO
```

Xylogic 的Annex 主機也有相同的問題，您要做相同的修改才能連上這些主機。

15. 我要怎樣才能把IP multicast support 打開？

FreeBSD 2.0 以後的版本內定都有支援Multicast host 操作。如果您想將您的主機設定成multicast router 的話，您必須重新compile 您的kernel，加入MROUTING 的選項，並且執行mouted(8) FreeBSD 2.2 及之後的版本會在開機時執行mouted(8) 如果在/etc/rc.conf 中mouted\_enable 設定為"YES"

MBONE 的各種工具可以在他們ports 下所屬叫做mbone (<http://www.FreeBSD.org/ports/mbone.html>) 目錄中找到。如果您在找視訊會議的工具如vic 以及vat的話，到那邊找找吧！

16. 哪些網路卡是使用DEC PCI chipset？

以下是Glen Foster <[gfooster@driver.nsta.org](mailto:gfooster@driver.nsta.org)> 提供的清單：

表格12-1. Network cards based on the DEC PCI chipset

Vendor	Model
ASUS	PCI-L101-TB
Accton	ENI1203
Cogent	EM960PCI
Compex	ENET32-PCI
D-Link	DE-530
Dayna	DP1203, DP2100
DEC	DE435, DE450
Danpex	EN-9400P3
JCIS	Condor JC1260
Linksys	EtherPCI
Mylex	LNP101
SMC	EtherPower 10/100 (Model 9332)
SMC	EtherPower (Model 8432)
TopWare	TE-3500P
Znyx (2.2.x)	ZX312, ZX314, ZX342, ZX345, ZX346, ZX348

Vendor	Model
Znyx (3.x)	ZX345Q, ZX346Q, ZX348Q, ZX412Q, ZX414, ZX442, ZX444, ZX474, ZX478, ZX212, ZX214 (10mbps/hd)

### 17. 為什麼要用FQDN 才能連到其他機器？

你也許會發現要連的機器其實是在另一個網域。舉個例子，假設你是在foo.bar.edu 這個網域中，想要連到在一台叫mumble 的主機，他在example.org 網域下，你必須用Fully-Qualified Domain Name mumble.example.org，而不是只用mumble。

傳統的BSD BIND resolver 允許用這種方式解出機器的位址，但是FreeBSD 內附bind (see named(8)) 版本內定方式，則是除了你所在的網域以外，不支援其他非FQDN 的縮寫。所以如mumble 必須在mumble.foo.example.org，否則就會從網域的最底層開始找。

這和先前的做法不同，也就是不用mumble.example.org，和mumble.edu 繼續搜尋。看一下RFC 1535，裡面有提到為什麼之前的做法不好，甚至算是個安全漏洞。

這裡有個不錯的解法，你可以加入一行

```
search foo.example.org example.org
```

instead of the previous

```
domain foo.example.org
```

在你的/etc/resolv.conf 檔案中(請參考resolv.conf(5))。但是要確定搜尋順序不會違反RFC 1535 所謂的“boundary between local and public administration”。

### 18. 為什麼我在連線時一直出現Permission denied 的錯誤訊息？

如果在編譯kernel 時加入IPFIREWALL 選項，請注意2.1.7R 內定是拒絕所有未經核准的網路封包(但在開發2.1-STABLE 時改掉了)。

如果不小心弄錯了firewall 的設定，你可以以root 執行以下命令網路功能就會恢復正常：

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

也可以在/etc/rc.conf 加入firewall\_type="open" 的選項。

如果想知道如何設定FreeBSD firewall，請參考手冊中相關章節(../handbook/firewalls.html)。

### 19. IPFW 會造成多大的網路延遲？

請參考手冊中Firewalls (../handbook/firewalls.html) 章節，特別是IPFW Overhead & Optimization (../handbook/firewalls.html#IPFW-OVERHEAD) 這一段。

### 20. 為什麼我的ipfw “fwd” redirect 規則將服務轉向其他機器時無法正常運作？

可能是你除了轉送封包以外還額外想進行位址轉譯(network address translation, NAT)， “fwd” 規則所進行的動作就如同字面所示；僅轉送封包，它並不會去修改封包中的資料。假設我們有如下的規則：

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

當一個通往特定目標位址 *foo* 的封包送達主機時，根據這條規則，封包將被轉送至 *10.0.0.1*，但是它的目標位址卻仍然是 *foo*！封包的目標位址並沒有更改為 *10.0.0.1*。大部分的主機會將封包丟棄，因為他們並不是這個目標位址。因此，使用“fwd”規則時往往不如使用者所預期的那般順利。這種行為是系統特性，而非錯誤。

參考關於服務轉向的常見問答集，natd(8) 手冊，或者是使用 ports collection ([../ports/index.html](http://www.freebsd.org/ports/index.html)) 中許多服務轉向的工具來正確的完成你想進行的工作。

## 21. 要如何把對某台機器的網路服務要求(service request)轉向到另一台？

在 ports 目錄的“sysutils”分類中有個叫 socket 的套件，可以幫你轉向 FTP 或其他類似的網路服務。只要把該網路服務的命令改成呼叫 socket 即可，如下所示：

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

其中 *ftp.example.com* 與 *ftp* 分別是被轉到的機器和 port 名稱。

## 22. 那裡可以找到管理頻寬的工具？

FreeBSD 上有三套頻寬管理工具：dummynet(4) 已經整合進入 FreeBSD 系統(更詳細的用途, ipfw(4)); ALTQ (<http://www.csl.sony.co.jp/person/kjc/programs.html>) 可以免費使用，Emerging Technologies (<http://www.etinc.com/>) 推出的 Bandwidth Manager 則是商用軟體。

## 23. 怎麼會跑出/dev/bpf0: device not configured這個訊息？

你執行了一個需要柏克萊封包過濾器(Berkeley Packet Filter) 的程式(bpf(4))，但是你在 kernel 中沒有啟動它。把下面這一行加入 kernel 設定檔中，編譯一個新的 kernel：

```
pseudo-device bpf          # Berkeley Packet Filter
```

在重新開機之後，還要做出 device node，在/dev 下執行：

```
# sh MAKEDEV bpf0
```

如果想要更進一步知道如何做出各種 device node，請參閱 Handbook 關於週邊節點的說明 ([../handbook/kernelconfig-nodes.html](http://www.freebsd.org/handbook/kernelconfig-nodes.html))。

## 24. 我要怎樣才能將Windows 機器中的磁碟掛入系統, 就像Linux 提供的smbmount 那樣？

使用 SMBFS 工具組。這套工具組中包含了一系列的 kernel 修改還有使用者的工具程式(userland programs)。這些程式和資訊在 ports 收藏中 net/smbfs 下可以找到。在 4.5-RELEASE 之後的版本則是系統中內建。

## 25. 我在系統日誌中發現以下訊息：“icmp-response bandwidth limit 300/200 pps”，這是蝦米碗糕？

這是系統核心告訴你有些活動引發它送出比它所認為應該送出更多的 ICMP 或 TCP 重置訊息(RST)。ICMP 回應訊息常常是因為有人嘗試連接未被使用的 UDP 通訊埠。TCP 重置訊息則是有人嘗試連接未開放 TCP 通訊埠造成的結果。以下這些活動可能就是造成這些訊息的原因：

- 暴力法的服務組絕攻擊(DoS)方式(相較於針對特殊弱點使用單一封包的攻擊方式)。

· 大量的通訊埠掃描(相較於僅嘗試少數的常見服務通訊埠)。

出現的數字中第一個代表根據這些流量kernel 應該送出的封包數，第二個數字則是kernel 目前限制最大發送數。你可以利用sysctl 修改net.inet.icmp.icmplim 變數值來更改最大值。舉例來說,如果希望修改限制為300 packets per second :

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

如果你不想在系統紀錄中看到這些訊息，但是仍然希望保持回應的限制的話，你可以利用sysctl 修改net.inet.icmp.icmplim\_output 變數來取消這些訊息：

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

最後，如果你想取消這些限制的話，你可以設定net.inet.icmp.icmplim (如上例所示) 為0。基於上述理由，我們不建議你取消這些限制。

**26.** 這個錯誤訊息arp: unknown hardware address format 是什麼意思？

這代表你的區域網路連線上有一些設備使用FreeBSD 看不懂的MAC 格式。這通常是代表有人在你的區域網路上進行實驗，最常見的就是cable modem 的連線。這訊息無害，而且應該不至於影響到FreeBSD 主機的性能。

**27.** 我剛剛裝好CVSup 套件，但是在嘗試執行時發生了錯誤，要怎麼辦？

首先，看看錯誤的訊息是否如下：

```
/usr/libexec/ld-elf.so.1: Shared object "libXaw.so.6" not found
```

這種錯誤訊息代表你主機上安裝的net/cvsup 沒有包含XFree86 套件。如果你想要使用CVSup 內建的圖形介面GUI 的話，你需要安裝XFree86。此外，如果你只想以命令列方式使用CVSup 的話，你應該先移除之前安裝的套件。並安裝net/cvsup-without-gui 這套軟體。在FreeBSD 手冊中CVSup (<http://www.freebsd.org/handbook/cvsup.html>) 段落中有更詳細的說明。

# 章13. 系統安全篇

## 1. 什麼是sandbox？

“Sandbox” 是系統安全用的術語，有兩個意義：

- 放在某些虛擬防護牆裡的執行程序，這些防護牆是用來阻止某些人侵入這道程序，進而出入於更大的系統中。

這道程序可以完全在防護牆裡“動作”。也就是說，它所執行的任何程式不可能會滲透到牆的外面。所以如果您對它有安全上的顧慮，並不需要特別去監聽它的一舉一動，反正它只能在牆內活動。

舉例來說，可以用userid 來做這道防護牆，這正是security 和named 說明文件中的定義。

現在就用ntalk 這個服務作說明（見/etc/inetd.conf）。這個服務以前的userid 是root，現在執行時則是用tty。tty 這個使用者就是一個sandbox，如果有人能夠順利用ntalk 侵入系統，現在他就算進得來也只能用這個userid。

- 放在某個模擬機器裡的程式，這比上述來得更嚴密。基本上這表示能侵入該程式的人相信他能再進入所屬的機器，但事實上只會進入模擬出來的機器，無法進一步修改任何真實的資料。

達到這個目的最常用的方法，就是在某個子目錄下做出模擬的環境，然後用chroot 執行該程式，這樣該程式的根目錄便是這個子目錄，而非系統真正的根目錄。

另一個常見作法是將某個檔案系統mount 成唯讀，但在它上面另外製造出程式以為可以寫入的檔案系統。這個程式會相信它可以對其他檔案讀寫，但只有它看不到這個唯讀效應- 系統執行的一般程式都看得到。

我們試圖將這類sandbox 盡量透明化，讓使用者或侵入者無法看到他是否在某個sandbox 裡面。

UNIX 實作兩種sandbox，一個在程式層面，另一個則是由userid 來達成。

每個UNIX 執行程序會用防火牆將它和所有其他程序隔開，某個程序不可以隨意修改其他程序位址的資料。這和Windows 中，程式可以輕易修改其他位址資料，結果導致當機的情形大不相同。

每個UNIX 程序都屬於某個特定的userid。如果該userid 不是root，就會將它和其他使用者的程序隔開。Userid 同時也用於硬碟資料的存取權上。

## 2. 什麼是securelevel?

securelevel 是核心中所實作的一個安全機制。基本上當securelevel 是正值時，核心會限制某些工作；即使是superuser (也就是root) 也無法完成那些工作。在撰寫本文時，securelevel 機制在一般的限制外，還能夠限制以下的功能：

- 清除某些特定的檔案旗標，例如schg (系統唯讀標旗, the system immutable flag)
- 經由/dev/mem 與/dev/kmem，將資料寫入至核心記憶體中
- 載入核心模組
- 更動ipfirewall(4) 規則。

想要檢查在某個運作中的系統的securelevel 狀態，只要執行以下命令即可：

```
# sysctl kern.securelevel
```

輸出的結果會包含一個`sysctl(8)` 變數名稱(在這個例子中，它是`kern.securelevel`) 以及一個數字。後者即是目前的`securelevel` 值。如果它是一個正值(也就是大於0)，表示至少有一些`securelevel` 的保護機制已經開啓了。

你沒有辦法降低一個運作中的系統的`securelevel`；如果可以的話，就失去了這個機制的意義了。如果你要作一些需要`securelevel` 為非正值才可以的動作的話(例如`installworld` 或更動日期)，你需要修改`/etc/rc.conf` 內的`securelevel` 設定(找找`kern_securelevel` 和`kern_securelevel_enable` 變數)，然後重新開機。

想要知道更多有關於`securelevel` 與各個不同等級影響的細節，請參考`init(8)` 說明文件。

**警告**`securelevel` 可不是萬靈丹；它有許多已知的缺陷，往往造成一種安全的假象。

它一個最大的問題，就是要讓這個功能完全有效的話，在`securelevel` 發揮作用前的啓動過程中，所有使用到的檔案都必須被保護起來。如果一個攻擊者在`securelevel` 有效前(由於有些系統在啓動中所作的事情，無法在較高的`securelevel` 中正常運作，所以這會在啓動過程中後期才會運作)，能讓他們程式被執行的話，`securelevel` 的保護就完全無效了。保護啓動程序中所有的檔案在技術上是可行的，但是如果真的這樣作的話，系統維護將會變成一場夢魘。即使只是修改一個設定檔，也必須將整個系統關閉，至少也得到單人模式。

除了這點，還有許多其它的東西都在通信論壇上討論，尤其是`freebsd-security`。請到這裡([.././././search/index.html](http://www.freebsd.org/cgi/search/index.html)) 搜尋以前的討論。有些人希望`securelevel` 能夠儘快消失，由另一個更優秀的機制取代，不過機會有點渺茫。

風險自行承擔。

**3. BIND (named)** 除了在通訊埠53 以外也在其他高編號通訊埠(high-numbered port) 聆聽(Listen)。這是怎麼回事？

FreeBSD 3.0 後的版本使用一個特殊的BIND 版本，這個版本會使用隨機的高編號通訊埠來回應外部的查詢。如果你因為要適合防火牆的設定或是單純的想讓自己看來舒服一點而想用53 通訊埠回應外部查詢，那麼你可以嘗試更改以下檔案相關內容`/etc/namedb/named.conf`：

```
options {
    query-source address * port 53;
};
```

你也可以將\* 更改為特定IP address，藉以加強控制條件。

順便恭喜你。能夠讀取你系統上的`sockstat(1)` 報告並且注意不正常狀況是一件好事！

**4. Sendmail** 除了在標準的通訊埠25 外也在通訊埠587 聆聽！這是怎麼回事？

較新版本的Sendmail 支援mail submission 這項功能，並且使用通訊埠587。這項功能還沒有被廣泛支援但是支援的數目正在增長中。

**5. 我發現了這個UID 0 toor 帳號，這是什麼碗糕？我被黑掉了嗎？**

放心。`toor` 是一個“alternative” 管理者帳號(`toor` 是`root` 的轉向拼法)。以往是跟隨`bash(1)` 安裝而建制的，後來則成為系統內定建制的一個帳號。這個帳號將伴隨一個非標準的shell 測試使用，讓你不需要去更改到`root` 的內建shell。因為這些其他的shell 並沒有跟隨系統預設值安裝(舉例來說，某些由ports 安裝的shell package)，而被內定安裝在`/usr/local/bin` 目錄下，有可能存在不同的檔案系統中。倘若`root` 的shell 被

放在/usr/local/bin，且/usr(或是其他包含著/usr/local/bin 這個子目錄的檔案系統) 因為某些原因並沒有被正常的mount 起來的話，root 將無法正常的登入系統進行維修(雖然說你重開機成單人模式就會問你要載入哪個shell)。

有些人使用toor 帳號進行每日的root 維護工作，如此可以使用非標準的shell，而root 可以保留標準shell，以因應單一使用者模式(single user mode) 或緊急狀況處理。依照系統內定值，你將無法使用toor 登入，因為這個帳號尚未更改密碼設定。因此你如果你想啟動這個帳號，你需要使用root 登入系統並且修改toor 的密碼。

## 6. 為什麼suidperl 無法正常運作？

因為某些安全的考，suidperl 內定的安裝並沒有設定suid bit。系統管理者可以依照以下命令啟動suid 設定。

```
# chmod u+s /usr/bin/suidperl
```

如果你想要在由source 升級時suidperl 內定啟動suid 功能的話，編輯/etc/make.conf 加入ENABLE\_SUIDPERL=true 然後執行make buildworld。

# 章14. PPP

## 1. I cannot make ppp(8) work. What am I doing wrong?

You should first read the ppp(8) man page and the PPP section of the handbook ([../handbook/ppp-and-slip.html#USERPPP](#)). Enable logging with the 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 in the `/etc/ppp/ppp.conf` configuration file (the start of the `default` section is the best place to put it). Make sure that `/etc/syslog.conf` (see `syslog.conf(5)`) contains the lines

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

and that the file `/var/log/ppp.log` exists. You can now find out a lot about what is going on from the log file. Do not worry if it does not all make sense. If you need to get help from someone, it may make sense to them.

If your version of ppp(8) does not understand the `set log` command, you should download the latest version (<http://people.FreeBSD.org/~brian/>). It will build on FreeBSD version 2.1.5 and higher.

## 2. Why does ppp(8) hang when I run it?

This is usually because your hostname will not resolve. The best way to fix this is to make sure that `/etc/hosts` is consulted by your resolver first by editing `/etc/host.conf` and putting the `hosts` line first. Then, simply put an entry in `/etc/hosts` for your local machine. If you have no local network, change your `localhost` line:

```
127.0.0.1      foo.bar.com foo localhost
```

Otherwise, simply add another entry for your host. Consult the relevant man pages for more details.

You should be able to successfully `ping -c1 `hostname`` when you are done.

## 3. Why will ppp(8) not dial in `-auto` mode?

First, check that you have got a default route. By running `netstat -rn` (see `netstat(1)`), you should see two entries like this:

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	

This is assuming that you have used the addresses from the handbook, the man page or from the `ppp.conf.sample` file. If you do not have a default route, it may be because you are running an old version of ppp(8) that does not understand the word `HISADDR` in the `ppp.conf` file. If your version of ppp(8) is from before FreeBSD 2.2.5, change the

```
add 0 0 HISADDR
```

line to one saying

```
add 0 0 10.0.0.2
```

Another reason for the default route line being missing is that you have mistakenly set up a default router in your `/etc/rc.conf` (see `rc.conf(5)`) file (this file was called `/etc/sysconfig` prior to release 2.2.2), and you have omitted the line saying

```
delete ALL
```

from `ppp.conf`. If this is the case, go back to the Final system configuration ([../handbook/ppp-and-slip.html#USERPPP-FINAL](#)) section of the handbook.

#### 4. What does `No route to host` mean?

This error is usually due to a missing

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

section in your `/etc/ppp/ppp.linkup` file. This is only necessary if you have a dynamic IP address or do not know the address of your gateway. If you are using interactive mode, you can type the following 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 ([../handbook/ppp-and-slip.html#USERPPP-DYNAMICIP](#)) section of the handbook for further details.

#### 5. Why does my connection drop after about 3 minutes?

The default PPP timeout is 3 minutes. This can be adjusted with the 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 the `ppp.conf` file, 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 `ppps` server socket using `telnet(1)` or `pppctl(8)`. Refer to the `ppp(8)` man page for further details.

#### 6. Why does my connection drop under heavy load?

If you have Link Quality Reporting (LQR) configured, it is possible that too many LQR packets are lost between your machine and the peer. Ppp deduces that the line must therefore be bad, and disconnects. Prior to FreeBSD version 2.2.5, LQR was enabled by default. It is now disabled by default. LQR can be disabled with the line

```
disable lqr
```

**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, your modem may hang up because it thinks (incorrectly) that it lost carrier.

There is a setting on most modems for determining how tolerant it should be to temporary losses of carrier. On a USR Sportster for example, this is measured by the S10 register in tenths of a second. To make your modem more forgiving, you could add the following send-expect sequence to your dial string:

```
set dial "..... ATs10=10 OK ....."
```

Refer to your modem manual for details.

**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.

If you are using an external modem, you can simply try using ping(8) to see if the TD light is flashing when you transmit data. If it flashes (and 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, you will need to use the `set server` command in your `ppp.conf` file. When the hang occurs, connect to ppp(8) using pppctl(8). If your network connection suddenly revives (PPP was revived due to the activity on the diagnostic socket) or if you cannot connect (assuming the `set socket` command succeeded at startup time), the problem is local. If you can connect and things are still hung, enable local async 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 you 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, you now have two possibilities:

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

There is very little you can do about this. Most ISPs will refuse to help if you are not running a Microsoft OS. You can enable `lqr` in your `ppp.conf` file, allowing ppp(8) to detect the remote failure and hang up, but this detection is relatively slow and therefore not that useful. You may want to avoid telling your ISP that you are running user-PPP...

First, try disabling all local compression by adding the following to your 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 will provide good ammunition when you contact your ISP (although it may make it apparent that you are not running a Microsoft product).

Before contacting your 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 your 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 you why their side is having a problem. Feel free to send the details to Brian Somers <brian@FreeBSD.org>, or even to ask your ISP to contact me directly.

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

Your best bet here is to rebuild ppp(8) by adding `CFLAGS+=-g` and `STRIP=` to the end of the Makefile, then doing a `make clean && make && make install`. When ppp(8) hangs, find the ppp(8) process id with `ps ajxww | fgrep ppp` and run `gdb ppp PID`. From the gdb prompt, you can then use `bt` to get a stack trace.

Send the results to <brian@Awfulhak.org>.

#### 11. Why does nothing happen after the “Login OK!” message?

Prior to FreeBSD version 2.2.5, once the link was established, ppp(8) would wait for the peer to initiate the Line Control Protocol (LCP). Many ISPs will not initiate negotiations and expect the client to do so. To force ppp(8) to initiate the LCP, use the following line:

```
set openmode active
```

注: It usually does no harm if both sides initiate negotiation, so openmode is now active by default. However, the next section explains when it *does* do some harm.

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

Occasionally, just after connecting, you may see messages in the log that say “magic is the 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, you will see 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 on the port, and executing ppp(8) from a login script or program after login. I have also heard 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 your ppp.conf file:

```
set openmode passive
```

This tells `ppp(8)` to wait for the server to initiate LCP negotiations. Some servers however may never initiate negotiations. If this is the case, you can do 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.

### 13. 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 an 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

```
set openmode passive
```

command. Care should be taken with this option. You should also use the

```
set stopped N
```

command to limit the amount of time that `ppp(8)` waits for the peer to begin negotiations. Alternatively, the

```
set openmode active N
```

command (where `N` is the number of seconds to wait before starting negotiations) can be used. Check the manual page for details.

### 14. Why does `ppp(8)` lock up shortly after connection?

Prior to version 2.2.5 of FreeBSD, it was possible that your link was disabled shortly after connection due to `ppp(8)` mis-handling Predictor1 compression negotiation. This would only happen if both sides tried to negotiate different Compression Control Protocols (CCP). This problem is now corrected, but if you are still running an old version of `ppp(8)` the problem can be circumvented with the line

```
disable pred1
```

### 15. Why does ppp(8) lock up when I shell out to test it?

When you execute the `shell` or `!` command, ppp(8) executes a shell (or if you have passed any arguments, ppp(8) will execute those arguments). Ppp will wait for the command to complete before continuing. If you attempt to use the PPP link while running the command, the link will appear to have frozen. This is because ppp(8) is waiting for the command to complete.

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

### 16. Why does ppp(8) 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 line

```
enable lqr
```

LQR is accepted by default if negotiated by the peer.

### 17. Why does ppp(8) dial for no reason in -auto mode?

If ppp(8) is dialing unexpectedly, you must determine the cause, and set up Dial filters (dfilters) 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, you will see the reason logged with a convenient timestamp next to it.

You can now 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 your demand-dial capabilities - most programs will need a DNS lookup before doing any other network related things.

In the DNS case, you should try to determine what is actually trying to resolve a host name. A lot of the time, `sendmail(8)` is the culprit. You should make sure that you tell `sendmail` not to do any DNS lookups in its configuration file. See the section on Mail Configuration for details on how to create your own configuration file and what should go into it. You may also want to add the following line to your `.mc` file:

```
define('confDELIVERY_MODE', 'd') dn1
```

This will make sendmail queue everything until the queue is run (usually, sendmail is invoked with `-bd -q30m`, telling it to run the queue every 30 minutes) or until a `sendmail -q` is done (perhaps from your `ppp.linkup` file).

### 18. 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, and the peer does not want to negotiate any compression at all. The messages are harmless, but if you wish to remove them, you can disable Predictor1 compression locally too:

```
disable pred1
```

### 19. Why does `ppp(8)` lock up during file transfers with IO errors?

Under FreeBSD 2.2.2 and before, there was a bug in the tun driver that prevents incoming packets of a size larger than the tun interface's MTU size. Receipt of a packet greater than the MTU size results in an IO error being logged via `syslogd`.

The PPP specification says that an MRU of 1500 should *always* be accepted as a minimum, despite any LCP negotiations, therefore it is possible that should you decrease the MTU to less than 1500, your ISP will transmit packets of 1500 regardless, and you will tickle this non-feature - locking up your link.

The problem can be circumvented by never setting an MTU of less than 1500 under FreeBSD 2.2.2 or before.

### 20. Why does `ppp(8)` not log my connection speed?

In order to log all lines of your modem "conversation", you must enable the following:

```
set log +connect
```

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

If you wish to see your connect speed and are using PAP or CHAP (and therefore do not have anything to "chat" after the CONNECT in the dial script - no `set login` script), you must make sure that you instruct `ppp(8)` to "expect" the whole CONNECT line, something like this:

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

Here, we get our CONNECT, send nothing, then expect a line-feed, forcing `ppp(8)` to read the whole CONNECT response.

**21. Why does ppp(8) ignore the \ character in my chat script?**

Ppp parses each line in your config 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. In order to specify a " character, you must escape it using a backslash (\).

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

If you wish to actually send a \ character to (say) your modem, you would need something like:

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

resulting in the following sequence:

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

or

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

resulting in the following sequence:

```
ATZ
OK
ATDT1234567
```

**22. Why does ppp(8) get a seg-fault, but I see no ppp.core file?**

Ppp (or any other program for that matter) should never dump core. Because ppp(8) runs with an effective user id of 0, the operating system will not write ppp(8)'s core image to disk before terminating it. If, however ppp(8) is actually terminating due to a segmentation violation or some other signal that normally causes core to be dumped, *and* you are sure you are using the latest version (see the start of this section), then you should do the following:

```
% tar xfz ppp-*.src.tar.gz
% cd ppp*/ppp
% echo STRIP= >>Makefile
% echo CFLAGS+=-g >>Makefile
% make clean all
% su
# make install
# chmod 555 /usr/sbin/ppp
```

You will now have a debuggable version of ppp(8) installed. You will have to be `root` to run ppp(8) as all of its privileges have been revoked. When you start ppp(8), take a careful note of what your current directory was at the time.

Now, if and when ppp(8) receives the segmentation violation, it will dump a core file called `ppp.core`. You should then do the following:

```
% su
# gdb /usr/sbin/ppp ppp.core
(gdb) bt
.....
(gdb) f 0
....
(gdb) i args
....
(gdb) l
.....
```

All of this information should be given alongside your question, making it possible to diagnose the problem.

If you are familiar with gdb, you may wish to find out some other bits and pieces such as what actually caused the dump and the addresses & values of the relevant variables.

### 23. Why does the process that forces a dial in auto mode never connect?

This was a known problem with ppp(8) set up to negotiate a dynamic local IP number with the peer in auto mode. It is fixed in the latest version - search the man page for `iface`.

The problem was that when that initial program calls `connect(2)`, the IP number of the tun interface is assigned to the socket endpoint. The kernel creates the first outgoing packet and writes it to the tun device. ppp(8) then reads the packet and establishes a connection. If, as a result of ppp(8)'s dynamic IP assignment, the interface address is changed, the original socket endpoint will be invalid. Any subsequent packets sent to the peer will usually be dropped. Even if they are not, any responses will not route back to the originating machine as the IP number is no longer owned by that machine.

There are several theoretical ways to approach this problem. It would be nicest if the peer would re-assign the same IP number if possible :-). The current version of ppp(8) does this, but most other implementations do not.

The easiest method from our side would be to never change the tun interface IP number, but instead to change all outgoing packets so that the source IP number is changed from the interface IP to the negotiated IP on the fly. This is essentially what the `iface-alias` option in the latest version of ppp(8) is doing (with the help of `libalias(3)` and ppp(8)'s `-nat` switch) - it is maintaining all previous interface addresses and NATing them to the last negotiated address.

Another alternative (and probably the most reliable) would be to implement a system call that changes all bound sockets from one IP to another. ppp(8) would use this call to modify the sockets of all existing programs when a new IP number is negotiated. The same system call could be used by dhcp clients when they are forced to `re-bind()` their sockets.

Yet another possibility is to allow an interface to be brought up without an IP number. Outgoing packets would be given an IP number of `255.255.255.255` up until the first `SIOCAIFADDR` ioctl is done. This would result in fully binding the socket. It would be up to ppp(8) to change the source IP number, but only if it is set to `255.255.255.255`, and only the IP number and IP checksum would need to change. This, however is a bit of a hack as the kernel would be sending bad packets to an improperly configured interface, on the assumption that some other mechanism is capable of fixing things retrospectively.

**24. Why do most games not work with the -nat switch?**

The reason games and the like do not work when libalias is in use is that the machine on the outside will try to open a connection or send (unsolicited) UDP packets to the machine on the inside. The NAT software does not know that it should send these packets to the interior machine.

To make things work, make sure that the only thing running is the software that you are having problems with, then either run `tcpdump` on the `tun` interface of the gateway or enable `ppp(8)` tcp/ip logging (`set log +tcp/ip`) on the gateway.

When you start the offending software, you should see packets passing through the gateway machine. When something comes back from the outside, it will be dropped (that is the problem). Note the port number of these packets then shut down the offending software. Do this a few times to see if the port numbers are consistent. If they are, then the following line in the relevant section of `/etc/ppp/ppp.conf` will make the software functional:

```
nat port proto internalmachine:port port
```

where `proto` is either `tcp` or `udp`, `internalmachine` is the machine that you want the packets to be sent to and `port` is the destination port number of the packets.

You will not be able to use the software on other machines without changing the above command, and running the software on two internal machines at the same time is out of the question - after all, the outside world is seeing your entire internal network as being just a single machine.

If the port numbers are not consistent, there are three more options:

1. Submit support in libalias. Examples of “special cases” can be found in `/usr/src/lib/libalias/alias_*.c` (`alias_ftp.c` is a good prototype). This usually involves reading certain recognised outgoing packets, identifying the instruction that tells the outside machine to initiate a connection back to the internal machine on a specific (random) port and setting up a “route” in the alias table so that the subsequent packets know where to go.  
This is the most difficult solution, but it is the best and will make the software work with multiple machines.
2. Use a proxy. The application may support `socks5` for example, or (as in the “`cvsup`” case) may have a “passive” option that avoids ever requesting that the peer open connections back to the local machine.
3. Redirect everything to the internal machine using `nat addr`. This is the sledge-hammer approach.

**25. Has anybody made a list of useful port numbers?**

Not yet, but this is intended to grow into such a list (if any interest is shown). In each example, `internal` should be replaced with the IP number of the machine playing the game.

- **Asheron’s Call**

```
nat port udp internal :65000 65000
```

Manually change the port number within the game to 65000. If you have got a number of machines that you wish to play on assign a unique port number for each (i.e. 65001, 65002, etc) and add a `nat port` line for each one.

- **Half Life**

```
nat port udp internal:27005 27015
```

- **PCAnywhere 8.0**

```
nat port udp internal:5632 5632
```

```
nat port tcp internal:5631 5631
```

- **Quake**

```
nat port udp internal:6112 6112
```

Alternatively, you may want to take a look at [www.battle.net \(http://www.battle.net/support/proxy/\)](http://www.battle.net/support/proxy/) for Quake proxy support.

- **Quake 2**

```
nat port udp internal:27901 27910
```

```
nat port udp internal:60021 60021
```

```
nat port udp internal:60040 60040
```

- **Red Alert**

```
nat port udp internal:8675 8675
```

```
nat port udp internal:5009 5009
```

## 26. What are FCS errors?

FCS stands for **F**rame **C**heck **S**equence. 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 your link is bad (or if your serial driver is dropping packets), you will see the occasional FCS error. This is not usually worth worrying about although it does slow down the compression protocols substantially. If you have an external modem, make sure your cable is properly shielded from interference - this may eradicate the problem.

If your link freezes as soon as you have connected and you see a large number of FCS errors, this may be because your link is not 8 bit clean. Make sure your modem is not using software flow control (XON/XOFF). If your datalink *must* use software flow control, use the command `set accmap 0x000a0000` to tell ppp(8) to escape the ^Q and ^S characters.

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

If nothing in your log file indicates why the link might have been terminated, you should ask the remote administrator (your ISP?) why the session was terminated.

## 27. Why do MacOS and Windows 98 connections freeze when running PPPoE on the gateway?

Thanks to Michael Wozniak <mwozniak@netcom.ca> for figuring this out and Dan Flemming <danflemming@mac.com> for the Mac solution:

This is due to what is called a “Black Hole” router. MacOS and Windows 98 (and maybe other Microsoft OSs) send TCP packets with a requested segment size too big to fit into a PPPoE frame (MTU is 1500 by default for Ethernet) *and* have the “do not fragment” bit set (default of TCP) and the Telco router is not sending ICMP “must fragment” back to the www site you are trying to load. (Alternatively, the router is sending the ICMP

packet correctly, but the firewall at the www site is dropping it.) When the www server is sending you frames that do not fit into the PPPoE pipe the Telco router drops them on the floor and your page does not load (some pages/graphics do as they are smaller than a MSS.) This seems to be the default of most Telco PPPoE configurations (if only they knew how to program a router... sigh...)

One fix is to use regedit on your 95/98 boxes to add the following registry entry...

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\NetTrans\0000\MaxMTU
```

It should be a string with a value "1436", as some ADSL routers are reported to be unable to deal with packets larger than this. This registry key has been changed to `Tcpip\Parameters\Interfaces\ID for adapter\MTU` in Windows 2000 and becomes a DWORD.

Refer to the Microsoft Knowledge Base documents Q158474 - Windows TCPIP Registry Entries (<http://support.microsoft.com/support/kb/articles/Q158/4/74.asp>) and Q120642 - TCPIP & NBT Configuration Parameters for Windows NT (<http://support.microsoft.com/support/kb/articles/Q120/6/42.asp>) for more information on changing Windows MTU to work with a NAT router.

Another regedit possibility under Windows 2000 is to set the `Tcpip\Parameters\Interfaces\ID for adapter\EnablePMTUBHDetect` DWORD to 1 as mentioned in the Microsoft document 120642 mentioned above.

Unfortunately, MacOS does not provide an interface for changing TCP/IP settings. However, there is commercial software available, such as OTAdvancedTuner (OT for OpenTransport, the MacOS TCP/IP stack) by Sustainable Softworks (<http://www.softworks.com/>), that will allow users to customize TCP/IP settings. MacOS NAT users should select `ip_interface_MTU` from the drop-down menu, enter 1450 instead of 1500 in the box, click the box next to `Save as Auto Configure`, and click `Make Active`.

The latest version of ppp(8) (2.3 or greater) has an `enable tcpmssfixup` command that will automatically adjust the MSS to an appropriate value. This facility is enabled by default. If you are stuck with an older version of ppp(8), you may want to look at the **tcpmssd** port.

## 28. None of this helps - I am desperate! What can I do?

If all else fails, send as much information as you can, including your config files, how you are starting ppp(8), the relevant parts of your log file and the output of the `netstat -rn` command (before and after connecting) to the FreeBSD general questions 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>) or the `comp.unix.bsd.freebsd.misc` ([news:comp.unix.bsd.freebsd.misc](mailto:news:comp.unix.bsd.freebsd.misc)) news group, and someone should point you in the right direction.

# 章15. Serial Communications

This section answers common questions about serial communications with FreeBSD. PPP and SLIP are covered in the  $\mu\text{U}$  12 章 section.

## 1. How do I tell if FreeBSD found my serial ports?

As the FreeBSD kernel boots, it will probe for the serial ports in your system for which the kernel was configured. You can either watch your system closely for the messages it prints or run the command

```
% dmesg | grep sio
```

after your system is up and running.

Here is some example output from the above command:

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
```

This shows two serial ports. The first is on irq 4, is using port address 0x3f8, and has a 16550A-type UART chip. The second uses the same kind of chip but is on irq 3 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 your system, or if you have added modem cards or have more serial ports than your kernel is configured for, just reconfigure your kernel. See section about building a kernel for more details.

## 2. How do I tell if FreeBSD found my modem cards?

Refer to the answer to the previous question.

## 3. I just upgraded to 2.0.5 and my `tty0x` are missing! How do I solve this problem?

Do not worry, they have been merged with the `ttidx` devices. You will have to change any old configuration files you have, though.

## 4. How do I access the serial ports on FreeBSD?

The third serial port, `sio2` (see `sio(4)`, known as COM3 in DOS), is on `/dev/cuaa2` for dial-out devices, and on `/dev/ttyd2` for dial-in devices. What is the difference between these two classes of devices?

You use `ttidx` for dial-ins. When opening `/dev/ttidx` in blocking mode, a process will wait for the corresponding `cuaax` device to become inactive, and then wait for the carrier detect line to go active. When you open the `cuaax` device, it makes sure the serial port is not already in use by the `ttidx` device. If the port is available, it “steals” it from the `ttidx` device. Also, the `cuaax` device does not care about carrier detect. With this scheme and an auto-answer modem, you can have remote users log in and you can still dial out with the same modem and the system will take care of all the conflicts.

**5. How do I enable support for a multiport serial card?**

Again, the section on kernel configuration provides information about configuring your kernel. For a multiport serial card, place an `sio(4)` line for each serial port on the card in the kernel configuration file. But place the `irq` and `vector` 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 `COM_MULTIPORT` option.

The following example is for an AST 4-port serial card on `irq 7`:

```
options "COM_MULTIPORT"
device sio4 at isa? port 0x2a0 tty flags 0x781
device sio5 at isa? port 0x2a8 tty flags 0x781
device sio6 at isa? port 0x2b0 tty flags 0x781
device sio7 at isa? port 0x2b8 tty flags 0x781 irq 7 vector siointr
```

The flags indicate that the master port has minor number 7 (0x700), diagnostics enabled during probe (0x080), and all the ports share an `irq` (0x001).

**6. Can FreeBSD handle multiport serial cards sharing irqs?**

Not yet. You will have to use a different `irq` for each card.

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

The `ttydx` (or `cuaax`) device is the regular device you will want to open for your applications. When a process opens the device, it will have a default set of terminal I/O settings. You can see these settings with the command

```
# stty -a -f /dev/ttyd1
```

When you change the settings to this device, the settings are in effect until the device is closed. When it is reopened, it goes back to the default set. To make changes to the default set, you can open and adjust the settings of the “initial state” device. For example, to turn on CLOCAL mode, 8 bits, and XON/XOFF flow control by default for `ttyd5`, do:

```
# stty -f /dev/ttyid5 clocal cs8 ixon ixoff
```

A good place to do this is in `/etc/rc.serial`. Now, an application will have these settings by default when it opens `ttyd5`. It can still change these settings to its liking, though.

You can also prevent certain settings from being changed by an application by making adjustments to the “lock state” device. For example, to lock the speed of `ttyd5` to 57600 bps, do

```
# stty -f /dev/ttyld5 57600
```

Now, an application that opens `ttyd5` and tries to change the speed of the port will be stuck with 57600 bps.

Naturally, you should make the initial state and lock state devices writable only by `root`. The `MAKEDEV(8)` script does *NOT* do this when it creates the device entries.

**8. How can I enable dialup logins on my modem?**

So you want to become an Internet service provider, eh? First, you will need one or more modems that can auto-answer. Your modem will need to assert carrier-detect when it detects a carrier and not assert it all the time. It will need to hang up the phone and reset itself when the data terminal ready (DTR) line goes from on to off. It should

probably use RTS/CTS flow control or no local flow control at all. Finally, it must use a constant speed between the computer and itself, but (to be nice to your callers) it should negotiate a speed between itself and the remote modem.

For many Hayes command-set-compatible modems, this command will make these settings and store them in nonvolatile memory:

```
AT &C1 &D3 &K3 &Q6 S0=1 &W
```

See the section on sending AT commands below for information on how to make these settings without resorting to an MS-DOS terminal program.

Next, make an entry in `/etc/ttys` (see `ttys(5)`) for the modem. This file lists all the ports on which the operating system will await logins. Add a line that looks something like this:

```
ttyd1 "/usr/libexec/getty std.57600" dialup on insecure
```

This line indicates that the second serial port (`/dev/ttyd1`) has a modem connected running at 57600 bps and no parity (`std.57600`, which comes from the file `/etc/gettytab`, see `gettytab(5)`). The terminal type for this port is `dialup`. The port is `on` and is `insecure`---meaning `root` logins on the port are not allowed. For dialin ports like this one, use the `ttydx` entry.

It is common practice to use `dialup` as the terminal type. Many users set up in their `.profile` or `.login` files a prompt for the actual terminal type if the starting type is `dialup`. The example shows the port as `insecure`. To become `root` on this port, you have to login as a regular user, then `su(1)` to become `root`. If you use `secure` then `root` can login in directly.

After making modifications to `/etc/ttys`, you need to send a hangup or HUP signal to the `init(8)` process:

```
# kill -HUP 1
```

This forces the `init(8)` process to reread `/etc/ttys`. The `init` process will then start `getty` processes on all `on` ports. You can find out if logins are available for your port by typing

```
% ps -ax | grep '[t]tyd1'
```

You should see something like:

```
747 ?? I      0:00.04 /usr/libexec/getty std.57600 ttyd1
```

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

If you are using another computer as a terminal into your FreeBSD system, get a null modem cable to go between the two serial ports. If you are using an actual terminal, see its accompanying instructions.

Then, modify `/etc/ttys` (see `ttys(5)`), like above. For example, if you are hooking up a WYSE-50 terminal to the fifth serial port, use an entry like this:

```
ttyd4 "/usr/libexec/getty std.38400" wyse50 on secure
```

This example shows that the port on `/dev/ttyd4` has a `wyse50` terminal connected at 38400 bps with no parity (`std.38400` from `/etc/gettytab`, see `gettytab(5)`) and `root` logins are allowed (`secure`).

**10. Why can I not run `tip` or `cu`?**

On your system, the programs `tip(1)` and `cu(1)` are probably executable only by `uucp` and group `dialer`. You can use the group `dialer` to control who has access to your modem or remote systems. Just add yourself to group `dialer`.

Alternatively, you can let everyone on your system run `tip(1)` and `cu(1)` by typing:

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

**11. My stock Hayes modem is not supported---what can I do?**

Actually, the man page for `tip(1)` is out of date. There is a generic Hayes dialer already built in. Just use `at=hayes` in your `/etc/remote` (see `remote(5)`) file.

The Hayes driver is not smart enough to recognize some of the advanced features of newer modems---messages like `BUSY`, `NO DIALTONE`, or `CONNECT 115200` will just confuse it. You should turn those messages off when you use `tip(1)` (using `ATX0&W`).

Also, the dial timeout for `tip(1)` is 60 seconds. Your modem should use something less, or else `tip` will think there is a communication problem. Try `ATS7=45&W`.

Actually, as shipped `tip(1)` does not yet support it fully. The solution is to edit the file `tipconf.h` in the directory `/usr/src/usr.bin/tip/tip`. Obviously you need the source distribution to do this.

Edit the line `#define HAYES 0` to `#define HAYES 1`. Then `make` and `make install`. Everything works nicely after that.

**12. How am I expected to enter these AT commands?**

Make what is called a “direct” entry in your `/etc/remote` file (see `remote(5)`). For example, if your modem is hooked up to the first serial port, `/dev/cuaa0`, then put in the following line:

```
cuaa0:dv=/dev/cuaa0:br#19200:pa=none
```

Use the highest bps rate your modem supports in the `br` capability. Then, type `tip cuaa0` (see `tip(1)`) and you will be connected to your modem.

If there is no `/dev/cuaa0` on your system, do this:

```
# cd /dev
# sh MAKEDEV cuaa0
```

Or use `cu` as `root` with the following command:

```
# cu -lline -speed
```

with `line` being the serial port (e.g. `/dev/cuaa0`) and `speed` being the speed (e.g. `57600`). When you are done entering the AT commands hit `~.` to exit.

**13. Why does the <@> sign for the pn capability not work?**

The <@> sign in the phone number capability tells tip to look in /etc/phones for a phone number. But the <@> sign is also a special character in capability files like /etc/remote. Escape it with a backslash:

```
pn=\@
```

**14. How can I dial a phone number on the command line?**

Put what is called a “generic” entry in your /etc/remote file (see remote(5)). For example:

```
tip115200|Dial any phone number at 115200 bps:\
      :dv=/dev/cuaa0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600 bps:\
      :dv=/dev/cuaa0:br#57600:at=hayes:pa=none:du:
```

Then you can do something like `tip -115200 5551234`. If you prefer `cu(1)` over `tip(1)`, use a generic `cu` entry:

```
cu115200|Use cu to dial any number at 115200bps:\
      :dv=/dev/cuaa1:br#57600:at=hayes:pa=none:du:
```

and type `cu 5551234 -s 115200`.

**15. Do I have to type in the bps rate every time I do that?**

Put in an entry for `tip1200` or `cu1200`, but go ahead and use whatever bps rate is appropriate with the `br` capability. `tip(1)` thinks a good default is 1200 bps which is why it looks for a `tip1200` entry. You do not have to use 1200 bps, though.

**16. How can I more easily access a number of hosts through a terminal server?**

Rather than waiting until you are connected and typing `CONNECT host` each time, use tip’s `cm` capability. For example, these entries in /etc/remote (see remote(5)):

```
pain|pain.deep13.com|Forrester’s machine:\
      :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank’s machine:\
      :cm=CONNECT muffin\n:tc=deep13:
deep13|Gizmonics Institute terminal server:\
      :dv=/dev/cuaa2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

will let you type `tip pain` or `tip muffin` to connect to the hosts `pain` or `muffin`; and `tip deep13` to get to the terminal server.

**17. Can tip try more than one line for each site?**

This is often a problem where a university has several modem lines and several thousand students trying to use them...

Make an entry for your university in /etc/remote (see remote(5)) and use <\@> for the `pn` capability:

```
big-university:\
      :pn=\@:tc=dialout
dialout:\
      :dv=/dev/cuaa3:br#9600:at=courier:du:pa=none:
```

Then, list the phone numbers for the university in `/etc/phones` (see `phones(5)`):

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

`tip(1)` will try each one in the listed order, then give up. If you want to keep retrying, run `tip(1)` in a while loop.

### 18. Why do I have to hit CTRL+P twice to send CTRL+P once?

CTRL+P is the default “force” character, used to tell `tip(1)` that the next character is literal data. You can set the force character to any other character with the `~s` escape, which means “set a variable”.

Type `~sforce=single-char` followed by a newline. `single-char` is any single character. If you leave out `single-char`, then the force character is the nul character, which you can get by typing CTRL+2 or CTRL+SPACE. A pretty good value for `single-char` is SHIFT+CTRL+6, which I have seen only used on some terminal servers.

You can have the force character be whatever you want by specifying the following in your `$HOME/.tiprc` file:

```
force=single-char
```

### 19. Why is everything I type suddenly in UPPER CASE?

You must have pressed CTRL+A, `tip(1)` “raise character”, specially designed for people with broken caps-lock keys. Use `~s` as above and set the variable “raisechar” to something reasonable. In fact, you can set it to the same as the force character, if you never expect to use either of these features.

Here is a sample `.tiprc` file perfect for Emacs users who need to type CTRL+2 and CTRL+A a lot:

```
force=^^
raisechar=^^
```

The ^^ is SHIFT+CTRL+6.

### 20. How can I do file transfers with `tip`?

If you are talking to another Unix system, you can send and receive files with `~p` (put) and `~t` (take). These commands run `cat(1)` and `echo(1)` on the remote system to accept and send files. The syntax is:

```
~p <local-file> [<remote-file>]
~t <remote-file> [<local-file>]
```

There is no error checking, so you probably should use another protocol, like `zmodem`.

**21.** How can I run zmodem with **tip**?

First, install one of the zmodem programs from the ports collection (such as one of the two from the comms category, **lrzsz** or **rzsz**).

To receive files, start the sending program on the remote end. Then, press enter and type `~C rz` (or `~C lrz` if you installed **lrzsz**) to begin receiving them locally.

To send files, start the receiving program on the remote end. Then, press enter and type `~C sz files` (or `~C lsz files`) to send them to the remote system.

**22.** Why does FreeBSD not find my serial ports, even when the settings are correct?

Motherboards and cards with Acer UARTs do not probe properly under the FreeBSD sio probe. Obtain a patch from [www.lemis.com](http://www.lemis.com) (<http://www.lemis.com/serial-port-patch.html>) to fix your problem.

## 章16. 其它各式各樣的問題

### 1. 為甚麼FreeBSD 用的置換(swap)空間比Linux 多？

FreeBSD僅是看起來置換空間(swap)用的比Linux多而已。在事實上，並不然。主要的差異是在於，FreeBSD積極的將閒置無用的主記憶體內容推入置換空間(swap)中，以使得主記憶體可以更為有效率的被使用。而Linux的策略是將置換空間(swap)用來作為解決記憶體問題的最終手段。較頻繁的使用置換空間(swap)。是一種更有效率的使用主記憶體的手段。

註：當一方面FreeBSD積極的使用置換空間(swap)的同時，你必需注意到，FreeBSD並不會任意的將所有的東西都推入置換空間(swap)中。如此，你才不會在一夜宿醉起床後發現，整個系統都被倒進了置換空間(swap)之中。

### 2. 即使我只有運行少數程式，為什麼top 顯示出來的剩餘記憶體還是很少？

簡單的答案是，所有未使用到的閒置記憶體都是被浪費的記憶體，任何未被你的程式所利用到的記憶體將被核心(kernel)用來當作磁碟快取(disk cache)。而這種記憶體被top(1) 標記為閒置的(Inact)，快取(Cache)，以及緩衝區(Buf)，並負責在各個不同的位置負責暫存資料。被暫存(cached)的資料代表系統不需要去存取較慢的磁碟裝置就可以得到資料，如此，可以提升系統的效能。總而言之，top(1) 顯示出較少的閒置(Free)記憶體是好的，只要顯示出來的值不是非常的低。

### 3. 為甚麼要用(甚麼是) a.out 和ELF 執行檔格式？

要了解為甚麼Freebsd使用ELF 格式，你有必要先認識一下三種在目前Unix 系統中最被廣泛應用到的執行檔格式：

注：在FreeBSD 3.x 之前，FreeBSD 使用a.out 格式。

- a.out(5)

這是最早，同是也是“最典型”的Unix目的檔格式。這種格式的檔案使用一種短且緊密的檔頭，同時，伴隨著一個魔術數字用來辨識格式。(參考a.out(5) 有更多詳細的說明)。它包含有三個節區：.text .data 及.bss 加上一個符號表及字串表。

- COFF

SVR3目的檔格式。檔頭包含了一個節區表，所以可以具備比.text .data .bss 還多的節區。

- ELF

ELF為COFF 格式の後繼者，主要的特徵為可以具有複數節區段，並可以使用32-bits或是64-bits的數值。主要的缺點為：ELF 格式是在每個系統中只會有一種ABI 的假設為前題被設計出來的。但是，在事實上，這個假設錯的離譜。因為，縱使在商用的SYSV 世界裡，也至少有SVR4，Solaris 和SCO 三種ABI。

譯註：ABI(Application Binary Interface)。如果一定要翻譯，就叫它 應用程式二進位介面好了。ABI被發展出來的用意，是為了促使在相同CPU所發展出來的應用程式，能夠在不同的系統上，作到二元檔(Binary Code)相容。比方說，Sun 所提出的Solaris ABI，保證執行檔能夠在相同CPU 的Solaris 系統上執行，另一個例子是Windows 系統。同屬於Intel x86 版本的執行檔能夠自由的在Windows 9x/me及Windows NT/2k/XP之間執行。

FreeBSD提供一個公用程式將程式所需的ABI資訊烙上，藉此試著去解決這個問題。請參考brandelf(1) 以取得更多資訊。i

FreeBSD 來自“傳統”的陣營。在傳統上，FreeBSD都使用a.out(5) 格式，這樣的技術在好幾代的BSD 都被證明是可靠的。雖然，在FreeBSD上可以建立以及正確的執行原生ELF 格式檔案(包含核心)。然而，FreeBSD在一開始反對將預設格式轉換為ELF，為什麼呢？當Linux開始痛苦的轉換至ELF 格式時，並非是為了要逃離a.out 格式。相反的，這是因為之前Linux的共享函式庫(shared libraries)採用以跳躍表格(jump-table)為基礎的技術去設計。這是一種讓發展者感到困擾，且非常難以使用，不具足夠彈性的方法。既然，已經存在的ELF 工具提供了共享函式庫(shared libraries)的解決方案，而且，那看起來是個“前衛的方法”，因此，所需的轉換代價就可接受因而轉換。

在FreeBSD的狀況中，我們的共享函式庫(shared libraries)機制和SunOS 的型式非常相近，且易於使用。然而，從3.0 開始，FreeBSD 正式將ELF 改為預設格式。雖然，a.out 格式依舊如以往般的好，但是，我們編譯工具的撰寫者，GNU 的成員，他們中止了對a.out 格式的支援與維護。在這種狀況下，迫使我們必須自行維護另一份版本的compiler 和linker，也使得我們無法從最新的GNU 發展成果中獲得好處。此外，對ISO-C++ 的需求，尤其是建構子(constructors)和解構子(destructors)，也帶動未來版本中對ELF 的原生支援。

#### 4. 是的, 但是, 為什麼會有這麼多不同格式的執行檔存在呢?

在黑暗而遙遠的過去，僅有簡陋的硬體存在。而因為硬體簡陋，當然也只能執行小而簡單的系統。a.out 格式是基於那個時代所需要，而被創造出來的(例如像PDP-11)。在這之後，許多人試著將Unix 移植到其他平台時，他們也保留了a.out 格式的執行檔。因為，這對早期的Motorola 68k, VAXen 之類的系統已經足夠使用了。

然而，人並不會滿足於現狀。一些聰明的硬體工程師想到了，如果能讓軟體多處理一些事，那CPU 的電晶體數就能少一點，並且跑得更快。要在這種新式的硬體上工作(現在稱為RISC)，a.out 這種格式就不合適了。基於這樣的現實所需，更多的執行檔格式被發展出來，以提供比簡單且受到許多限制的a.out 格式更好的效能。比方像是COFF, ECOFF, 已及一些較不為人所周知的格式紛紛被創造出來。但是，這些格式都已達到各自的極限，直到有一天ELF 的出現。

此外，當程式的體積越來越大，而磁碟空間和主記憶體相對來說都較小時，共享函式庫(shared libraries)的觀念被發展出來了。在這同時，虛擬記憶體系統(VM System)也變得越來越精巧。當每一種進步都在a.out 格式上被發展出來時，它的可用性也同時變得越來越低。另外，人們還希望程式能在執行期間動態載入，或是將已經執行過且沒有用的初始化程式碼丟棄，藉以節省更多的記憶。程式語言在這個時期也便得更精巧，人們也希望在main 之前自動的執行更多的東西。因此，許多繁雜且另人嘆為觀止的技巧被用在a.out 格式上去解決這些問題。但是，由於a.out 格式先天的限制，要解決這些問題必需付出更多的代價及時間成本，並讓程式的複雜度大為提升。而ELF 格式可以一舉解決這一切問題。但是，要將整個系統從根本轉換過去，將會有不短的陣痛期，因此，ELF格式將會有一陣子與a.out 並存。

然而，隨著時間的過去，FreeBSD的build tools 演化成平行的兩個支線(尤其是組譯器和載入器)。FreeBSD這條路加進了共享函式庫(shared libraries)並修正了一些錯誤。而原來發展這些程式的GNU 成員則為了因應現況，重寫了這些程式，以更簡單的方式對跨平台編譯(building cross compilers)，以及多種格式(plugging in different formats) 作出了支援。許多人想作出以FreeBSD 為目的平台的跨平台編譯器。但不幸的是，FreeBSD 的as 和ld 不能作這項工作。新的GNU 工具程式加入了跨平台編譯(Cross Compiler)，ELF格式支援，共享函式庫(shared libraries)，C++ 的擴充... 等等。此外，許多廠商以ELF 格式發行其產品，如果這些東西能在FreeBSD 上執行的話當然是最好的。既然，能夠執行ELF 格式的執行檔了，為什麼還須要a.out 呢？它已經是一匹垂垂老矣的馬了，在竭力盡忠的奉獻這麼多年之後，該是讓它在牧場肥沃的草地上好好休息的時候了。

ELF 格式比a.out 具有更良好的展現能力，並且在底層系統中具有更多的可擴展性。ELF 工具程式更容易被維護，且提供跨平台編譯的支援，這一點對很多人來說是很重要的。ELF 格式可能比a.out 慢一點，但是其差異非常難測量出來。這兩者間還有許多細節上的不同，比方說分頁對應的方式，程式碼初始化的方法...等等。這些並不是很重要，但是，兩者就是不同。以後，GENERIC 核心(kernel)將會移除對a.out 格式。當不在有執行傳統a.out 程式的須要時，將會從核心(kernel)中移除。

### 5. 為甚麼chmod不會改變符號連結(symmlink)的存取權限？

Symlinks 本身並沒有存取權限，同時，在預設的狀況下，chmod(1) 將不會跟隨著symlinks 去改變目標檔案的存取權限。因此，如果你有一個檔案foo，同時，有一個symlink bar 指向這個檔案，以下這個命令將永遠會成功的被執行。

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

然而，在foo 上的存取權限將不會被改變。

你必需使用-H 或是將-L 與-R 選項一起使用，參考chmod(1) 以及symlink(7) 以取得更多的資訊。

**警告**使用選項-R 會讓chmod(1) 以遞迴(RECURSIVE) 的方式工作。當你把chmod(1) 用在目錄或是連結到目錄的符號連結時更要小心。如果你要改變一個符號連結參考到的目錄之存取權限chmod(1)，且注意不要加上任何選項，並且在symlink 的結尾加上斜線(/)。舉例來說，如果foo 連結到目錄bar，而你要更改foo (實際上是bar)，那就使用：

```
% chmod 555 foo/
```

結尾的斜線會使得chmod(1) 改變foo 所指向的目錄bar 的權限。

### 6. 為什麼在FreeBSD 2.2.x 及更早的版本中，登入名稱(login names) 被限制在八個字元以下呢？

你可能認為修改UT\_NAMESIZE 後在重新編譯整個系統是很容易的事。而且在這之後，每件事都可以運作的很好。不幸的是，有許多的程式和工具(包含系統工具)把數字寫死在程式裡頭(並非總是8 或9，有時可能是古怪的15 或20)。這不僅僅是會將你的系統記錄檔弄壞而已(來自於變動長度和固定長度記錄的差異)，同時也會破壞Sun 的NIS Client 的運作。同時，和其他的Unix 系統之間也有可能產生未知的問題。

在FreeBSD 3.0 及之後的版本，帳號的最大長度增加到16個字元，同時，那些將長度寫死的程式也被找出來並作了適當的修正。正因為影響系統的範圍很廣，所以直到3.0版之後才算大致修正完成。

如果你有自信在出問題的時後能自行解決，你可以利用下面的方法讓較早期的版本支援較長的帳號。首先，修改/usr/include/utmp.h 中的UT\_NAMESIZE。然後，你必須把/usr/include/sys/param.h 中的MAXLOGNAME 改成跟UT\_NAMESIZE 相同。最後，如果你是從原始程式建立系統，別忘了/usr/include 每次都會被更新。修改/usr/src/.. 中適當的檔案。

### 7. 我能在FreeBSD下執行DOS程式嗎？

是的，自3.0版起你可以使用BSDI的doscmd DOS 模擬器，如果你對這個東西有興趣，或是想加入發展行列，請寄一封電子郵件到FreeBSD-emulation 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-emulation>) 。

對於3.0之前的系統，在ports 中有一套軟體可以模擬8088，並提供足夠的BIOS中斷服務以執行DOS文字模式的程式，這套軟體叫做pcemu，同時，運行它須要X Windows(由XFree86提供)。

8. 如果要把FreeBSD文件翻譯成我的母語，我需要作什麼？

參閱FreeBSD文件中的翻譯常見問答 (./fdp-primer/translations.html)。

9. 為什麼我寄到FreeBSD.org 相關地址的電子郵件都被退回了呢？

FreeBSD.org 的郵件系統對於進來的郵件採取嚴格的檢查，並且退回所有設定不正確，或是潛在的垃圾郵件。你的郵件被退回可能是因為下列原因所引起：

- 這封電子郵件來自自己知的垃圾郵件區域或是IP中。

FreeBSD郵件伺服器將拒絕接收已知的垃圾郵件來源的電子郵件。如果提供你網路服務的公司或是網域中有產生過垃圾郵件或是有垃圾郵件轉播站，請你換一個服務提供者，或是乾脆放棄。

- 電子郵件的本文僅有HTML。

郵件應該已純文字格式發送，請設定你的電子郵件軟體送出純文字格式。

- FreeBSD的郵件處理程式無法由IP反查送件主機的IP。

設置DNS 反查是接受一台主機郵件的一個標準要求，請為您的郵件主機設置DNS 反查。許多提供家庭網路服務(DSL, cable, dialup 等)的公司並不提供這樣的服務。在這種情況下，請透過網路服務提供者的郵件伺服器送出您的電子郵件。

- 在SMTP 使用EHLO/HELO 命令時所給予的hostname 無法被解析到一個IP 位置。

在郵件被接受以前，一個充分合格，且可被解析的主機名稱在SMTP 協定的對談中是必要的。如果你沒有在DNS 伺服器中登記你的主機名稱，請透過網路服務提供者的郵件伺服器送出您的電子郵件。

- 你的訊息中夾帶著一個message ID 以“localhost” 字串結束。

某些郵件軟體產生某些不正確的message ID，這將不被接受。你必需更改設定讓你的郵件軟體產生正確的message ID，如果這無法解決，考慮說服你的郵件軟體作者更新程式以處理這個問題。

10. 我可以在哪裡找到一個免費的FreeBSD帳號？

FreeBSD的伺服器本身不提供任何對外的服務，其他的單位中，有人提供開放的Unix 系統服務。其中有些可能要收取些許費用。

Arboret, Inc (<http://www.arboret.org/>)，也被稱為M-Net，自1983 年起就開始提供Unix 系統服務。一開始，他們使用Altos 並執行System III。他們在1991 年轉換系統成為BSD/OS。在2000 年六月，他們再度更換成為FreeBSD。M-Net 能讓使用者透過SSH 及telnet 連線到主機，並提供完整的FreeBSD 軟體以供使用。然而，M-Net 作為一個非盈利組織運行，存取權只限於成員和贊助者，M-Net 也提供BBS 系統和網路聊天服務。

GreX (<http://www.grex.org/>) 提供了非常類似M-Net 的服務，包括了BBS 系統和網路聊天。然而，機器是使用Sun 4M，並執行SunOS。

11. 什麼是sup，我該如何使用它？

SUP (<http://www.FreeBSD.org/cgi/ports.cgi?^sup>) 的意思是Software Update Protocol，由CMU發展，用來維持整個發展的同步。我們利用它保持遠端的站台和原始站台之間的同步工作。

然而，SUP在頻寬的使用上並不太友善，同時，目前也不再使用了。目前建議維持原始碼同步更新的方法是CVSup ([../handbook/synching.html#CVSUP](http://www.FreeBSD.org/handbook/synching.html#CVSUP))。

12. 這個可愛的小紅人叫作什麼？

似乎，他並沒有一個正式的名字，姑且就稱其為“BSD小惡魔”吧。如果你執意要使用一個名字。那就叫他“小動物(beastie)”吧。註：“beastie”在讀音上跟“BSD”很接近。

你可以在BSD小惡魔的主頁 (<http://www.mckusick.com/beastie/index.html>) 上取得更多的資訊。

13. 我能使用BSD小惡魔圖案嗎？

也許吧，我也不確定。BSD小惡魔圖案的版權是屬於馬歇爾蘇格蘭教會的Marshall Kirk McKusick所擁有。你可以試著去查看網頁關於BSD小惡魔肖像 (<http://www.mckusick.com/beastie/mainpage/copyright.html>) 以取得更詳細的使用細節。

總而言之，如果你純粹為了自己想要鑑賞，那麼，你可以自由的使用肖像。如果你是個人使用，只要情況適當，應該都會被許可。如果你想在商業上使用，則你必需聯繫蘇格蘭教會的Kirk McKusick以取得許可。如果你需要更進一步詳細的資訊，請參考BSD小惡魔的首頁 (<http://www.mckusick.com/beastie/index.html>)。

14. 你有任何的BSD小惡魔圖案可以讓我使用嗎？

你可以在 `/usr/share/examples/BSD_daemon/` 找到Xfig及eps兩種格式的圖檔。

15. 我在文件、郵遞論壇上，常會看到一些縮寫字、技術字彙，這些可以去哪邊查呢？

請參閱 FreeBSD 字彙表 ([http://www.FreeBSD.org/doc/zh\\_TW.Big5/books/handbook/freebsd-glossary.html](http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/freebsd-glossary.html))。

16. 為什麼我該在意腳踏車車棚的顏色？

最短最短的答案是：『不用在意』。稍微長一點的答案是：『雖然你有能力自己去建造一座腳踏車車棚，但是，這不代表因為你不喜歡現在這個車棚的顏色，就要中止他的建築。』這個比喻的意思是，你不需要去爭論每一個細項特徵，只因為你有辦法去作它。某些人的評論是：『雜音的程度，與變化的複雜性是成反比』。

更長且較完整的答案是，在經過長時間爭論關於是否該將sleep(1)的秒參數移除，Poul-Henning Kamp <[phk@FreeBSD.org](mailto:phk@FreeBSD.org)>發表了一篇長論“在青翠草地上的腳踏車車棚(任何顏色的)... (<http://www.FreeBSD.org/cgi/getmsg.cgi?fetch=506636+517178+/usr/local/www/db/text/1999/freebsd-hackers/19991003.freebsd-hackers>)”。以下，僅摘要該則文章部分內容：

“什麼是關於這個腳踏車車棚?” 部分的人這樣的詢問我。

這是一個非常長遠的故事，否則就是一個古老的故事。但是事實上，這個故事非常的短。C.諾斯科特·帕金森(C. Northcote Parkinson)在1960年代初期寫了一本書，書名為“Parkinson's Law(中文書名：升官有道—暴露上司心態之帕金森定律)”，這本書包含了很多具有卓見的動態管理學。

[引述一點在這本書上的評論]

在這個被捲入腳踏車車棚案的特殊例子，主要的要素是核能發電場，我想，這足以說明這本書的年齡。

帕金森展示了該如何在董事會中贏得贊同去建造一座數百萬或甚至十億美元的核能發電場，但是，如果你想要去建造一座腳踏車車棚，你將會被糾纏在無窮無盡的討論之中。

他(帕金森)並解釋，這是因為一個核能發電場是這樣的廣闊，這樣的昂貴，並且這樣的複雜，以至於人們無法掌握它，而並非嘗試，他們急切的希望有人能夠幫他們處理並解決所有瑣碎的細項。Richard P. Feynmann 給了一些有趣，且非常一針見血的論點，在他的書提到了Los Alamos 的例子。

另一方面，任何人都能在週末組裝一座腳踏車車棚出來，並且仍有閒聊可以觀賞電視及玩遊戲。因此，無論你作了多麼完善的準備，也不管你提出的方案是多麼的妥當，某些人仍將抓住機會跑出來告訴你，他正在作同樣的事，正在付出努力，他就在這裡。

在丹麥，我們稱這個叫作『虎死留皮』(setting your fingerprint)。這關係到你個人的驕傲和聲望，這關係到你是否可以指著某地後對著別人說：『這裡! 這是我作的。』這是政治人物很重要的一個特徵。但是，時機是大多數人民所賦與的。想想那些留在水泥地上的腳印吧。

—Poul-Henning Kamp <phk@FreeBSD.org> on freebsd-hackers, October 2, 1999

# 章17. FreeBSD 冷笑話集

## 1. How cool is FreeBSD ?

問：有人做過FreeBSD 執行時的溫度測試嗎？我知道Linux 比DOS 涼，但沒聽人提過FreeBSD，似乎很熱。

答：沒有，但是在味覺上有做過無數次測試。我們矇上自願受試者的眼睛，事先再給他們服用250 毫克的LSD-25 迷幻藥。35% 的受試者說FreeBSD 嘗起來像橘子，而Linux 則是紫色的榛樹果實。據我所知，沒有一組提到溫度上特別的差異。後來發現，有太多受試者在測試時夢遊走出房間影響到數據，最後只得放棄整個調查。我想大部份的受試者現在在Apple 工作，繼Drag and Drop 之後，研究全新的“Scratch and Sniff” 圖形界面。It's a funny old business we're in !

不開玩笑了，FreeBSD 和Linux 都使用HLT (halt) 指令，可在系統閒置時降低電力的使用也減少了熱的產生。如果有設定APM(Automatic Power Management) ，FreeBSD 也可以讓CPU 進入省電模式。

## 2. 誰在我的記憶體插槽中沙沙作響??

問：FreeBSD 編譯核心時有做甚麼“奇特” 的事讓記憶體沙沙作響嗎？當編譯時(還有開機時確認軟碟後的短暫時間)，也種似乎來自記憶體插槽的奇怪聲音。

答：是的！在BSD 的文件中你會常常看到“背後靈”，大部份的人都不知道那是一種實際存在的精神體--- 掌控著你的電腦。你聽到的聲音是這些背後靈以高音口哨在溝通怎樣做許多的系統管理工作。

如果這些聲音很困擾你，來自DOS 的fdisk /mbr 就能擺脫，但如果有相反的效果也不要驚訝。事實上，如果在儀式中聽到Bill Gates 恐怖的聲音從內建的喇叭傳來，馬上逃而且不要回頭！從BSD 背後靈不平衡的影響中解放，DOS 和Windows 背後靈通常都能重新控制整台機器並對你的靈魂詛咒。如果有選擇，我想我寧願習慣奇怪的聲音。

## 3. 要幾個FreeBSD hacker 才能換掉一個電燈泡？

一千一百七十二個；

二十三個在-current 上抱怨看不到光了；

四個宣稱這是設定上的問題，所以像這樣的email 應該放在-questions ；

三個submit PR，其中一個送錯到doc 下，並且內容只有”這裡好暗” ；

一個commit 尚未測試的電燈泡，造成不能buildworld，五分鐘後他把原來的燈泡換回來；

八個煽起flame war，責怪送出PR 的人沒有包括patch ；

五個埋怨buildworld 爛掉了；

三十一個說buildworld 可以用，不能用的人一定是cvsup 的時機不對；

一個把換成新燈泡的patch 丟到-hackers 上；

一個說他三年前就做出了patch，但送到-current 後卻被忽略掉，所以他對整個PR 系統有很不好的印象。此外，他也認為拿出的新燈泡無法反光；

三十七個咆哮說電燈泡不屬於基本系統的一部份，所以committer 不能不先諮詢整個Community 的意見就這樣做下去。還有，-core 到底和這件事有什麼關係？！

兩百人抱怨換燈泡之後，腳踏車棚的顏色變得好奇怪；

三個指出，用來換燈泡的patch 不符合style(9)的規定；

十七個埋怨拿出來的燈泡為什麼是用GPL；

五百八十六人陷入一場flame war，在GPL、BSD、MIT、NPL 各個license 和FSF 某位不具名創辦人士個人衛生之間，比較彼此的優勢；

七個將這一串討論的不同部份分別移到-chat 和-advocacy；

就算提出的新燈泡比舊的暗，還是有一個把它commit 進來；

兩個換回原先的燈泡，並且留下極為憤怒的commit 訊息。他們認為與其讓FreeBSD 用暗燈泡，還不如乾脆待在黑暗中算了；

四十六人對取消不用暗燈泡這件事大聲疾呼，要求-core 立刻提出澄清；

十一個要求換成小一點的電燈泡，以便未來FreeBSD 如果移植到電子雞上後會更為方便；

七十人抱怨-hackers 和-chat 上的SNR，藉unsubscribe 來表示抗議；

十三個送出” unsubscribe”、” 我要如何unsubscribe” 或” 拜託把我從list 名單中刪掉”，信的最後面則是一般由majordomo 加上去的footer；

當每個人忙於彼此叫罵時，有個傢伙趁沒人注意，把可以用的燈泡偷偷換上去；

三十一個指出如果用TenDRA 編譯新的燈泡，會比舊的來得亮0.364%（雖然燈泡會被編譯成正六面體），所以FreeBSD 內定的編譯器應該是TenDRA，而不是EGCS；

有個人說新燈泡缺乏美感；

九個人（包括原先送PR 的人）問” 什麼是MFC？”；

五十七個抱怨自從換了燈泡後，兩個星期都沒有光出現。

*Nik Clayton <nik@FreeBSD.org> 補注：*

*剛看到時，我快笑翻了。*

*然後想到，” 等一下，不是應該還有一個要將這些記在list 上嗎？”*

*接著終於了解我的使命:-)*

#### 4. 寫入/dev/null 的資料跑到哪裡去了？

在CPU 中有一種特別的資料散熱器，利用排出散熱片/風扇組合時，轉換成熱能，這就是為什麼CPU 冷卻日趨重要的原因；當人們使用更快的處理器時，他們變成不在乎有越來越多他們的資料都送進了/dev/null，而使得他們的CPU 過熱，如果你刪除了/dev/null (那將會有效的關閉CPU 的資料散熱器) 你的CPU 也許會降低工作溫度，但是你的系統將會很快的像是罹患了便秘伴隨著所有超出的資料開始變成行為不正常。如果你有快速的網路連線，你可以利用讀取/dev/random 並將他隨意傳送至各處，來降低你CPU 的溫度；然而你將陷入使你網路連線或/ 有過熱的風險或是惹惱你的ISP，大部分的資料最終將會在他們的設備上轉換成熱，不過他們通常都擁有好的散熱，所以如果你做的不太過分，應該是沒什麼大不了的。

*Paul Robinson 補充：*

中文版FAQ 注：以下短文屬於美式幽默，恐翻譯後造成語焉不詳，語意不通的情形，故保留原汁原味讓讀者自行品嚐。

There are other methods. As every good sysadmin knows, it is part of standard practise 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 categorised by the type of hat they wear (red, green or blue) and will hide or appear

(thereby showing the colour 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 simulation - 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.

#### 5. 1993 年就有『make world』了？

問：根據jkh的這份commit (<http://www.freebsd.org/cgi/cvsweb.cgi/src/Makefile.diff?r1=1.5;r2=1.6;f=h>)，應該是Aug 13 22:47:28 1994 UTC 之後才會有make world，怎麼會有在那之前會有呢？

答：根據pttlz.bbs@ptt.cc (工蟻) 於Apr 7 17:09:47 2008 CST 自稱：『make world 是我古早(15年前)時玩freebsd時用的』，所以1993年就有pttlz的『make world』存在了，請注意人家可是自稱『15年前』的長輩，請勿對長輩權威挑戰，謝謝！

## 章18. 進階主題

### 1. 如何能學習更多有關FreeBSD 內部的東西？

目前市面上還沒有探討作業系統內部的書是專為FreeBSD 而寫的。然而，許多一般的UNIX 知識都可以直接應用在FreeBSD 上。附加一點，仍然有相關的書是專為BSD 所寫的。

請參考Handbook 的作業系統內部之參考書目 ([../handbook/bibliography-osinternals.html](http://www.freebsd.org/handbook/bibliography-osinternals.html)) 。

### 2. 如何能為FreeBSD 出一份力？

請參考這篇文章Contributing to FreeBSD ([../articles/contributing/article.html](http://www.freebsd.org/articles/contributing/article.html)) 來提供您的建議。如果您能幫忙那就更歡迎了！

### 3. SNAP 和RELEASE 是什麼？

目前有三個活躍/半活躍的分支在FreeBSD 的CVS Repository (<http://www.FreeBSD.org/cgi/cvsweb.cgi>) (古早分支已經幾乎沒在更新了，因為通常只有三個活躍的發展分支)：

- RELENG\_5 即5-*STABLE*
- RELENG\_6 即6-*STABLE*
- HEAD 即-*CURRENT* 也就是目前的7.X-*CURRENT*

與上面其他兩個分支相比，其實HEAD 並不是真正的branch tag，它只是個symbolic constant，代表“*current* (尚未分支的發展中版本)”，通常我們會簡寫為“-CURRENT”。

就現在而言，“-CURRENT” 就是指7.X 的發展；而5-*STABLE* 分支(RELENG\_5)是在2004 年10 月從“-CURRENT” fork 出來的；6-*STABLE* 分支(RELENG\_6)是在2005 年11 月從“-CURRENT” fork 出來的。

### 4. 要怎麼作出自己的release？

請參照Release 工程 ([../articles/releng/article.html](http://www.freebsd.org/articles/releng/article.html)) 文章說明。

### 5. 為何make world 會把原來裝的binary 檔都換掉了？

沒錯，就是這樣子。如名字所示，make world 會重新編譯系統內建的每個binary 檔，這樣在結束時就可確定有個一致且乾淨的環境(所以要花上好一段時間)。

在執行make world 或make install 時，如果有設DESTDIR 這個環境變數，新產生的binary 將會裝在\${DESTDIR} 下同樣的目錄樹中。但在某些修改shared library 和重建binary 的無特定情況下，這樣做可能會使make world 失敗。

### 6. Why isn't cvsup.FreeBSD.org a round robin DNS entry to share the load amongst the various CVSup servers?

While CVSup mirrors update from the master CVSup server hourly, this update might happen at any time during the hour. This means that some servers have newer code than others, even though all servers have code that is less than an hour old. If [cvsup.FreeBSD.org](http://cvsup.FreeBSD.org) was a round robin DNS entry that simply redirected users to a random CVSup server, running CVSup twice in a row could download code older than the code already on the system.

7. 在系統開機時，出現“(bus speed defaulted)”。

Adaptec 1542 SCSI 卡允許使用者用軟體調整匯流排的存取速度。早期的1542 驅動程式試圖將它設成可用的最快速度，但後來發現在一些機器上不能用，所以現在要在kernel 設定中加TUNE\_1542 這個選項來啟動這個功能。在支援的機器上用這個選項會使硬碟存取更快，但在不支援的機器上有可能會毀掉資料。

8. 在網路頻寬有限的情況下，我也可以跟上current 的發展嗎？

是的，藉著CTM ([../handbook/synching.html#CTM](http://handbook.synching.html#CTM)) 您就可以不用下載全部的程式碼。

9. 是怎麼把發行版本中的檔案切成一個個240k 的小檔案的？

以BSD 為基礎的較新系統有個-b 選項可以把檔案以任意數目byte 切開。

這裡是/usr/src/Makefile 中的一個例子：

```
bin-tarball:
(cd ${DISTDIR}; \
tar cf - . \
gzip --no-name -9 -c | \
split -b 240640 - \
${RELEASEDIR}/tarballs/bindist/bin_tgz.)
```

10. 我在kernel 中加了新功能，我要把它寄給誰？

請參考Contributing to FreeBSD ([../articles/contributing/article.html](http://../articles/contributing/article.html)) 中的文章，以了解要如何提供您的程式碼。

同時也謝謝您的關心！

11. ISA 的隨插即用卡是如何偵測及初始化的？

由Frank Durda IV 所寫：[uhclem@nemesis.lonestar.org](mailto:uhclem@nemesis.lonestar.org)

簡單的說，當主機發出是否有PnP 卡的詢問訊號時，所有的PnP 卡會在幾個固定的I/O port 作回應。所以當偵測PnP 的程式開始時，它會先問有沒有PnP 卡在，接著所有PnP 卡會在它讀的port 以自己的型號# 作回答，這樣偵測程式就會得到一個wired-OR “yes” 的數字，其中至少會有一個bit 是打開的。然後偵測程式會要求型號(由Microsoft/Intel指定)小於X 的卡“離線”。再去看是否還有卡回答同樣的詢問，如果得到0，就表示沒有型號大於X 的卡。現在程式會問是否有型號小於x 的卡，如果有的話，程式再要型號大於X-(limit/4) 的卡離線，然後重覆上面的動作。用重複這種類似semi-binary search 的方法，在某範圍內找個幾次後，測程式最後會在機器中區分出所有的PnP 卡，搜尋次數也遠低於一個個找的 $2^{64}$  次。

一張卡的ID 由兩個32-bit(所以上面是 $2^{64}$ ) + 8bit 偵錯碼組成，第一個32 bits 是用來區分各家廠商的。這些廠商從來沒有出來澄清過，但看來應假設同一家出的不同種類的卡的廠商ID 有可能不同。用32 bits 只來表示不同廠商的想法實在有點過頭了。

第二個32 bits 則是型號#、乙太網路位址、或一些使這張卡獨特的資料。除非第一個32 bits 不同，否則廠商不可能作出第二個32 bit 相同的兩張卡。所以在一台機器中可以有同樣的好幾張卡，然而他們整個64 bits 還是會都不一樣。

這兩個32 bit 絕對不可以全為零，這才能使得最開始binary search 中的wired-OR 會得到一個非零數字。

一旦系統區分出所有卡的ID，接著會經由同樣的I/O port 一個個重新啟動每張卡，接著找出已知介面卡所需的資源、有哪些中斷可以使用等等。所有卡都會被掃描一次，來收集這些資料。

這些資訊接著和硬碟上的ECU 檔案、或MLB BIOS 裡的資料結合在一起，通常是綜合ECU 和MLB 裡的BIOS PnP 資料，這些週邊並不支援真正的PnP，然而偵測程式在檢查BIOS 和ECU 資料後，它可以避免PnP 週邊和那些偵測不到的相衝突。

接著再度拜訪這些PnP 週邊，這次會把可用的I/O、DMA、IRQ 和記憶體映射的位址都指定給它們。這些週邊就會出現在所指定的地方，直到下一次重新開機為止，不過也沒有人說不能把它們隨時移來移去。

上面有相當多的簡化，但你應該已經了解大致的過程。

Microsoft 把表示印表機狀態的幾個主要port 拿來作PnP，他們的邏輯是沒有一張卡會在這些地方解碼作相反的I/O cycles。但是我找到一款早期仍在評估PnP 提案時的IBM 原廠printer board，它的確去解對這些狀態port 的寫入資料，但是MS “說了就算”。所以它們的確有對印表機狀態port 寫入，還有讀取該位址+0x800、和另一個在0x200 及0x3ff 之間的port。

### 12. 我為某設備寫了驅動程式，能不能給它一個major number？

這要看你是否打算將這個驅動程式公開使用，如果是的話，請把它的原始碼送一份給我們，還有files.i386 修改的部份、kernel 設定檔樣本、以及用來產生設備檔的MAKEDEV(8)。如果你不打算公開、或因為版權問題而不能公開的話，我們有特地保留character major number 32 和block major number 8 給這方面的使用，直接用這兩個就好了。不論如何，我們都會很感激你能在FreeBSD technical discussions 郵遞論壇 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers>) 發表驅動程式的消息。

### 13. 關於更動目錄放置的原則？

在回答關於更動目錄放置的原則方面，我在1983 年寫好目前的作法後就沒有再改變過，這種方式是針對原先的FFS 檔案系統，後來也沒有對它作任何更動。它在避免cylinder group 被填滿這方面做得相當成功，但是就像有些人已經注意到，它和find 就配合得不好。大部份的檔案系統是由那些用depth first search(aka ftw) 產生的archive 製造出來，解出來的目錄inode 會橫跨好幾個cylinder group，如果以後要做depth first search 的話，這是最糟糕的情況之一。如果我們知道總共會產生多少目錄的話，解法是在做任何存取/寫入動作之前，在每個cylinder group 上先造出(所有目錄數/cylinder group 的數目)這麼多的目錄。很明顯的，我們必須要有根據地去猜這個數字，就算一個像10 的很小固定數目也會使效率以級數成長。區分restore (即解開上述的archive) 和一般檔案操作的方法可以是(現在用的演算法可能要更敏感)：如果一些目錄(最多10 個)都在10 秒內產生的話，那麼就把這些目錄聚集在同一個cylinder group。不管怎樣，我的經驗指出這是一個已經充份實驗過的部份。

Kirk McKusick, September 1998

### 14. 如何在kernel panics 時得到最多的資訊？

[這節是從Bill Paul <[wpaul@FreeBSD.org](mailto:wpaul@FreeBSD.org)> 在FreeBSD-CURRENT mailing list 上發表的信中節錄，Dag-Erling C. Smørgrav <[des@FreeBSD.org](mailto:des@FreeBSD.org)> 修正了打字錯誤、再加上括弧裡的注解。]

```
From: Bill Paul <wpaul@skynet.ctr.columbia.edu>
Subject: Re: the fs fun never stops
To: Ben Rosengart
Date: Sun, 20 Sep 1998 15:22:50 -0400 (EDT)
Cc: current@FreeBSD.org
```

[Ben 發表了下面的panic 訊息]

```
> 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

```

當你看到像這樣的訊息時，只把它拷一份送上來是不夠的。我在上面特地標明的instruction pointer 值相當重要，不幸的是它會因設定而不同。換句話說，這個值會跟你用的kernel image 檔而變動。如果是用某個snapshot 版本的GENERIC kernel，也許其他人可以追蹤到出問題的函式，但如果你是用自訂的kernel，那麼只有你才能告訴我們問題出在那裡。

要做的事包括這些：

1. 把instruction pointer 的值記下來。注意在前面的0x8: 在這個情況中並不重要，我們要的是0xf0xxxxxx。
2. 當系統重新開機後，執行這道命令：

```
% nm -n / (造成 panic 的 kernel 檔案) | grep f0xxxxxx
```

其中f0xxxxxx 就是記下來的instruction pointer 值。有可能不會剛好找到完整的這個字串，這是因為kernel symbol table 裡的各個symbol 只是函式的進入點，但instruction pointer 所指的位址有可能是在函式內的某一處，而不一定在開頭。所以如果找不到整個字串，那麼把instruction pointer 值的最後一個數字拿掉，再試一次：

```
% nm -n / (造成 panic 的 kernel 檔案) | grep f0xxxxx
```

如果這樣也找不到，那就把另一個數字去掉再找，一直重複到找到為止，結果是一串可能造成panic 的函式列表。這樣比直接找到出問題的函式來得差，但至少好過什麼都沒有。

我常常看到人們顯示一大片panic 訊息，但很少看到有人花一點時間把instruction pointer 和kernel symbol table 中的函式比較一下。

要追蹤出造成panic 原因的最好方法是先做出crash dump，然後用gdb(1) 在上面做stack trace。

不管是那一種，我通常是用這個方法：

1. 寫好kernel 設定檔。如果你需要用kernel debugger，在設定檔中加上options DDB 這個選項。(當我懷疑有出現無窮迴圈時，通常會用這個來設定中斷點。)
2. 用config -g KERNELCONFIG 做出用來編譯的目錄。
3. cd /sys/compile/ KERNELCONFIG; make
4. 等待kernel 編譯結束。
5. make install
6. 重新開機

`make(1)` 將會製造出兩個 `kernel`。 `kernel` 還有 `kernel.debug`。 `kernel` 將會被安裝到 `/kernel`，而 `kernel.debug` 可用來給 `gdb(1)` 當作 `debugging symbols` 的來源。

要確定能抓到 `crash dump`，先編輯 `/etc/rc.conf` 將 `dumpdev` 指到 `swap` 分割區。這樣 `rc(8)` 會用 `dumpon(8)` 來啓動 `crash dump`，你也可以手動執行 `dumpon(8)` 在 `panic` 之後，`crash dump` 可以用 `savecore(8)` 存起來；如果 `/etc/rc.conf` 裡有設 `dumpdev` 那麼重新開機後 `rc(8)` 會自動執行 `savecore(8)` 把 `crash dump` 存在 `/var/crash`。

**注：**FreeBSD 的 `crash dump` 通常和機器裡的實際記憶體一樣大，就像如果有 64MB 記憶體，`crash dump` 大小就是 64MB。所以要確定 `/var/crash` 下有足夠的空間，或是可以手動執行 `savecore(8)` 把 `crash dump` 放到另一個空間較夠的目錄下。另一種也許可以限制 `crash dump` 的方法，是在 `kernel` 設定檔中用 `options MAXMEM=(foo)`，將 `kernel` 可用的記憶體限制在合理的大小。舉例來說，如果你有 128MB 的記憶體，但是可以限制 `kernel` 只能用 16MB 的記憶體，這樣 `crash dump` 就是 16MB 而不是 128MB 了。

一旦發現有了 `crash dump`，就可以用 `gdb(1)` 來做 `stack trace`，如下所示：

```
% gdb -k /sys/compile/KERNELCONFIG/kernel.debug /var/crash/vmcore.0
(gdb) where
```

要注意可能會出現好幾個螢幕的可用資訊，你可以用 `script(1)` 把所有輸出都存起來。用包括所有 `debug symbol` 的 `kernel` 來除錯，這樣應該可以直接顯示 `panic` 是發生在那一行。通常是由下往上讀 `stack trace`，這樣才能一個個追蹤出有哪些動作引到 `crash`。也可以用 `gdb(1)` 把各種變數或結構的內容印出來，以檢查系統 `crash` 時的實際狀態。

好啦，如果你有第二台電腦而且有夠瘋狂，可以將 `gdb(1)` 設定成遠端除錯。這樣你可以在一台機器中用 `gdb(1)` 去除錯另一台裡的 `kernel`，可以執行的包括設定中斷點、在 `kernel` 原始碼中一步步執行等等，就像在一般使用者程式上除錯一樣。由於沒有什麼機會為除錯而設置兩台並鄰電腦，所以我還沒有這樣玩過。

*[Bill 補充：“我忘了提到一點：如果你有啓動 `DDB` 而 `kernel` 也已經進入除錯器，可以在 `DDB` 命令列下打 `panic`，強迫產生 `panic` (還有 `crash dump`)。也有可能是在 `panic` 階段時再進入除錯器，如果這樣的話，輸入 `continue`，接著它就會完成 `crash dump`。” -ed]*

### 15. 為什麼 `dlsym()` 不能操作 ELF 執行檔？

在 ELF 一系列的 инстру中，內定是不會讓 `dynamic linker` 看到執行檔裡定義了哪些 `symbol`。所以 `dlsym()` 沒有辦法用藉由呼叫 `dlopen(NULL, flags)` 取得的 `handle`，用它去搜尋有那些 `symbol` 一定會失敗。

如果你想要用 `dlsym()` 找出某個 `process` 的主執行檔中有哪些 `symbol`，則要在 `link` 時對 ELF linker (`ld(1)`) 加上 `-export-dynamic` 這個參數。

### 16. 我要如何增加或減少 `kernel` 能定址的空間？

預設值是，FreeBSD 3.x 的 `kernel` 可以定址的空間是 256 MB 而 FreeBSD 4.x 可以到 1 GB。如果是網路負荷相當重的伺服器(例如大型 FTP 或 HTTP 伺服器)，你也許會發現 256 MB 可能不大夠。

所以，要如何增加定址空間呢？要從兩方面著手。首先首先告訴 `kernel` 本身要保留較大空間給自己。其次，既然是在定址空間的最上面載入 `kernel`，所以還要調低載入的位址，才不會和前面定址的範圍重疊。

增加 `src/sys/i386/include/pmap.h` 裡的 `NKPDE` 就可以達成第一個目標。1 GB 的定址空間會像這樣：

```
#ifndef NKPDE
```

```

#ifdef SMP
#define NKPDE                254      /* addressable number of page tables/pde's */
#else
#define NKPDE                255      /* addressable number of page tables/pde's */
#endif /* SMP */
#endif

```

要算出NKPDE的正確值，將想要的空間大小(以megabyte為單位)除以4，接著單CPU機器減1，雙CPU則是減2。

要解決第二個問題，必須自行算出kernel被載入的位址：求出0x100100000減掉定址空間大小的值(以byte為單位)，如1GB大小就是0xc0100000。把src/sys/i386/conf/Makefile.i386裡的LOAD\_ADDRESS設成這個值；接著在src/sys/i386/conf/kernel.script中，將section列表最前面的location counter設成相同的值，如下：

```

OUTPUT_FORMAT("elf32-i386", "elf32-i386", "elf32-i386")
OUTPUT_ARCH(i386)
ENTRY(bttext)
SEARCH_DIR(/usr/lib); SEARCH_DIR(/usr/obj/elf/home/src/tmp/usr/i386-unknown-freebsdelf/lib);
SECTIONS
{
  /* Read-only sections, merged into text segment: */
  . = 0xc0100000 + SIZEOF_HEADERS;
  .interp      : { *(.interp)      }

```

然後重新編譯您的kernel。您可能會在執行ps(1)、top(1)這類的程式時碰到問題；make world應該就可以解決(或把改過的pmap.h複製到/usr/include/vm/下，再手動編譯libkvm，ps(1)還有top(1))。

注意：kernel所能定址的空間大小必須是4 megabytes的倍數。

[David Greenman <dg@FreeBSD.org> 補充：我認為kernel定址空間大小應該要是2的乘冪，但不大確定這一點。舊的啟動程式會動到high order address bits，記得它假設至少有256 MB。]

## 章19. 感謝

假如您在這份FAQ 中找到錯誤的地方，或是您想增加些甚麼，請寫封信到FAQ Maintainer <faq@FreeBSD.org> 。  
我們非常感謝您的建議，因為您的建議讓這份文件變得更好！

—FreeBSD Core Team

Jordan K. Hubbard <jkh@FreeBSD.org>

不停的更新過時的FAQ

Doug White <dwhite@FreeBSD.org>

經常在freebsd-questions 上回答問題

Jörg Wunsch <joerg@FreeBSD.org>

經常在Usenet 上回答問題

Garrett Wollman <wollman@FreeBSD.org>

Networking and formatting

Jim Lowe

Multicast information

Peter da Silva <pds@FreeBSD.org>

FreeBSD FAQ 這份文件的打字苦工

The FreeBSD Team

Kvetching, moaning, submitting data

對於那些曾經對這份FAQ 提供幫助，而我們沒提到的人們，我們由衷的感謝您！

## 章20. Ports and Packages 常見問題

### 1. 如何只抓取tarball？

如果只希望抓取tarball 下來的話，僅需輸入以下指令即可：

```
# make fetch
```

如果是要抓取單一的port，以editors/joe 為例：

```
# cd /usr/ports/editors/joe
# make fetch
```

那麼，預設會將editors/joe的tarball 下載至/usr/ports/distfiles 目錄下。

如果是希望抓取安裝此ports 所有相關相依ports 的tarball，以systuils/portupgrade 為例的話，則：

```
# cd /usr/ports/systuils/portupgrade
# make fetch-recursive
```

預設會將此port 與所有需要的其他port 的tarball，下載至/usr/ports/distfiles 目錄下。

如果是希望抓取全部所有ports 的tarball，則：

```
# cd /usr/ports
# make fetch
```

則會所將全部所有ports 的tarball 下載至/usr/ports/distfiles 目錄下。

如果是希望抓取ftp 分類下所有ports 的tarball，則：

```
# cd /usr/ports/ftp
# make fetch-recursive
```

則會所將ftp 分類下所有ports 的tarball 都下載至/usr/ports/distfiles 目錄下。

### 2. 如何僅做到解開tarball的步驟？

有時候習慣自己patch 原始碼的時候，會很常用到這個功能。以editors/joe 為例：

```
# cd /usr/ports/editors/joe
# make extract
```

會將tarball解開至/usr/ports/editors/joe/work 目錄下。

### 3. 如何僅做到解開tarball 並補上官方提供的patch ？

此法與make extract 的方法有一些類似，不同於是先補上官方提供的patch，再行patch 自己的修正。有時候習慣自己patch 原始碼的時候，則這個方式正好符合您的需求。以editors/joe 為例：

```
# cd /usr/ports/editors/joe
```

```
# make patch
```

會將tarball解開至/usr/ports/editors/joe/work 目錄下，並補上官方提供的patch。

#### 4. 如何安裝一個新的port？

如果系統上未安裝此軟體，則可以選擇安裝一個新的port。以editors/joe 為例的話，則：

```
# cd /usr/ports/editors/joe
```

```
# make install
```

如此會在系統上安裝一個新的joe 軟體。如果需要在安裝完成後，一併清除編輯時期所留下來的暫存目錄，則可配合make clean 的方法一起使用，如：

```
# cd /usr/ports/editors/joe
```

```
# make clean
```

如果想要一次清掉所有ports 產生的暫存資料，則只要回到ports 的根目錄執行即可：

```
# cd /usr/ports
```

```
# make clean
```

#### 5. 如何安裝一個新的port，並打包(package)起來？

將安裝完成的軟體打包起來，有許多便利性：包括在叢集系統中，可供其它機器使用，或將未來此軟體出問題可重新利用此package 重新快速安裝。以editors/joe 為例的話，則：

```
# cd /usr/ports/editors/joe
```

```
# make package
```

如此會在系統上安裝一個新的joe 軟體，並將此軟體打包(package)起來。package 預設會在/usr/ports/editors/joe 目錄下，如果希望集中管理的話，建議做如下的步驟：

```
# mkdir -p /usr/ports/packages
```

以後打包的packages 都會存放在此目錄下，並且系統會自動做分類，以方便管理。如果需要在安裝完成後，一併清除編輯時期所留下來的暫存目錄，則可配合make clean 一起使用，如：

```
# cd /usr/ports/editors/joe
```

```
# make package clean
```

**6. 如何打包一個port，並將其所有相依的ports 也打包起來？**

因為make package 只有打包單一套件，中間依賴的ports 並沒有一起打包，這會出現一個常遇到的問題：就是如果一個port 需要依賴其它的ports，那麼必須將其它ports 也一起打包，否則安裝packages 會有相依性的問題。以sysutils/portupgrade 為例：

```
# cd /usr/ports/sysutils/portupgrade
# make DEPENDS_TARGET=package package
```

如此會對所有portupgrade 所相依的ports 都一併打包，也包括自己本身。

**7. 如何對一個已經安裝的port 打包？**

如果安裝好一個套件，事前並未打包，事後想打包的話，則：以editors/joe 為例：

```
# cd /var/db/pkg
# pkg_create -b joe-{版本號}
```

會將已安裝的port 打包起來，放在/var/db/pkg 目錄下。

**8. 如何清理ports 編輯期間所產生的暫存資料？**

在安裝port 的時候，會有編譯期間所需要的工作目錄(work)，因此通常安裝好一個套件後，會清除此暫存目錄，以節省硬碟空間。以editors/joe 為例：

```
# cd /usr/ports/editors/joe
# make clean
```

如果是希望清除所有ports 的暫存目錄，則：

```
# cd /usr/ports
# make clean
```

如果是希望清除所有ftp 分類的暫存目錄，則：

```
# cd /usr/ports/ftp
# make clean
```

**9. 如何清理ports 編輯期間所產生的暫存資料，以及tarball 檔？**

在make clean 僅只是清除編輯期間所需要的工作目錄(work)，並沒有將編譯ports 時一併下載的tarball 刪除(相對應之tarball 預設會存放在/usr/ports/distfiles) 如果要把tarball 一併清除的話，以editors/joe 為例：

```
# cd /usr/ports/editors/joe
```

```
# make distclean
```

**make distclean** 的步驟包含了 **make clean** 的功能，也就是說除了會刪除tarball外，還會一併清除編譯時的暫存work目錄。

如果是希望清除所有ports的暫存work目錄及tarball，則：

```
# cd /usr/ports
# make distclean
```

而如果是希望清除所有ftp分類的work目錄及tarball，則：

```
# cd /usr/ports/ftp
# make distclean
```

### 10. 如何在安裝ports前查詢所依賴的相關套件？

在安裝ports前，可以查詢所依賴的相關套件。以mail/p5-Mail-SpamAssassin為例：

```
# cd /usr/ports/mail/p5-Mail-SpamAssassin
# make all-depends-list
```

**make all-depends-list** 顯示此套件所有相依的套件。

```
# cd /usr/ports/mail/p5-Mail-SpamAssassin
# make pretty-print-build-depends-list
```

**make all-depends-list** 顯示此套件在編譯期間所需要的套件。

```
# cd /usr/ports/mail/p5-Mail-SpamAssassin
# make pretty-print-run-depends-list
```

**make all-depends-list** 顯示此套件要執行時所需要的套件。

### 11. 如何移除已安裝的ports？

以editors/joe為例：

```
# cd /usr/ports/editors/joe
# make deinstall
```

或是使用**pkg\_delete**：

```
# pkg_delete joe-{version}
```

有時候套件之間的相依性會導致無法直接移除，如果要強制移除的話，則：

```
# pkg_delete -f joe-{version}
```

但請注意：很有可能導致其它相依到這軟體的套件執行起來出現問題。

至於二者的差別，請參考make deinstall 與pkg\_delete 有什麼不同

## 12. 如何一併移除所相依的ports？

以sysutils/portupgrade 為例：

```
# cd /usr/ports/sysutils/portupgrade
```

```
# make deinstall-depends
```

執行此步驟前，請注意是否會移除其他套件也有共同相依的部分。建議先參考make-depends-list 的方法來檢查。

或是使用

```
pkg_delete
```

這樣若仍有相依該套件的話，會先警告而不會移除。除非有另外加了-f 參數來強制移除。

```
# pkg_delete -r portupgrade-{version}
```

至於二者的差別，請參考make deinstall 與pkg\_delete 有什麼不同

## 13. 如何重新安裝已安裝過的ports？

重新安裝的前提是：之前有安裝過或目前已安裝。以editors/joe 為例：

```
# cd /usr/ports/editors/joe
```

```
# make deinstall clean install
```

或是

```
# cd /usr/ports/editors/joe
```

```
# make reinstall
```

## 14. 如何以關鍵字搜尋ports？

如果要從全部的ports collection 中找尋與關鍵字"ldap" 有關的ports，則：

```
# cd /usr/ports
```

```
# make search key=ldap
```

如果只要從與ftp 相關的ports 下找尋與關鍵字"ldap" 有關的ports，則：

```
# cd /usr/ports/ftp
```

```
# make search key=ldap
```

還有另一個用法，方法只是將key換成name。如果已經知道要搜尋ports的名稱，或只想找名稱相關的關鍵字"ldap"，則：

```
# cd /usr/ports
```

```
# make search name=ldap
```

### 15. 如何升級已安裝的ports？

如果已經安裝套件，事後欲升級的話，必須先移除舊版的port。以editors/joe為例：

```
# cd /usr/ports/editors/joe
```

```
# make clean reinstall
```

### 16. 如何查詢目前系統安裝了哪些套件？

查詢目前系統已安裝的全部套件：

```
# pkg_info
```

### 17. 如何查詢目前系統有沒有安裝這個關鍵字的套件？

假設要找的關鍵字是joe的話：

```
# pkg_info | grep joe
```

### 18. 如何查詢某個檔案是屬於哪些套件？

如果想查詢/usr/local/bin/joe是屬於哪個套件的話，則：

```
# pkg_info -W /usr/local/bin/joe
```

如果沒有回傳任何資訊的話，代表著這個檔案是由FreeBSD內建的。

### 19. 如何查詢某個套件安裝了哪些檔案？

如果想查詢目前系統所安裝的joe包含了哪些檔案，則：

```
# pkg_info -L /var/db/pkg/joe-{version}
```

## 20. 如何安裝舊版的ports？

有時候會因為相依性，或是新版有問題，而會想裝舊版本的套件。這裡介紹的方法是利用CVS的好處，回歸到以前舊版本存在的日子，以安裝舊版本的套件。

首先，若我們要回復到某一個套件的版本時，需要去查詢FreeBSD ports CVS repository。最常見的就是Freshports (<http://www.freshports.org/>) 網站、FreeBSD 的Mailing FreeBSD cvs (<http://lists.freebsd.org/pipermail/cvs-all/>) 或是FreeBSD ports cvsweb (<http://www.freebsd.org/cgi/cvsweb.cgi/ports/>)。

查到該套件版本所依存的日子後，就修改CVS tag。一般預設ports的CVS tag會寫在/usr/share/examples/cvsup/ports-supfile，如要回溯到2002/10/05號的話，則：

```
# vi /usr/share/examples/cvsup/ports-supfile

*default date=2002.10.05.00.00.00 #將 date 改成當日
```

然後按照一般CVSup或cvsup的時候一樣，執行CVSup或cvsup (make update)，此時的ports collections就會回到當時的情形，那麼該套件的舊版也會出現在ports collections中，只要安裝即可。

如果僅是想回溯某部份的ports，則必須加上額外的資訊，如僅希望把lang/perl5.8回溯，而我們得知此屬於lang中的一支，則：

```
# vi /usr/share/examples/cvsup/ports-supfile

#ports-all #將 ports-all 標示起來

ports-lang #加入這行
```

最後，執行CVSup或cvsup，並安裝即可。目前若希望單獨回溯單一的port，則比較麻煩。

## 21. 如何更新ports Mk？

Mk (/usr/ports/Mk) 是編譯ports時所參考的設定，有時若發生ports collections太新，而導致Mk的內容不符，此時就是應該更新Mk的時候了。

```
# cd /usr/src

# make update

# cd /usr/src/share/mk

# make install
```

## 22. 如何解決安裝ports時出現sed -i的錯誤？

因為BSD style的sed，也就是BSD本身自有的sed，與一些ports編譯期間所執行的sed不一致，所以會導致一些語法錯誤。此時先安裝sed\_inplace (textproc/sed\_inplace)，然後再安裝原本無法安裝的ports：

```
# make -DUSE_REINPLACE install
```

**23. 如何列出所有可以升級的ports？**

ports collection 的更新速度很快，在每次更新ports collections 後，往往會出現比目前現在安裝的套件還新的版本，可以令系統自行整理並提供可升級套件的列表：

```
# pkg_version -c
```

**24. 如何得知ports 所提供的編譯參數？**

所有的ports collections 中所提供的編譯參數都會在對應的Makefile 檔案內詳述，如sysutils/portupgrade 的話，則是位在/usr/ports/sysutils/portupgrade/Makefile 檔案下。

```
# cd /usr/ports/sysutils/portupgrade
```

```
# make -DNOPORTDOCS install
```

那麼安裝此ports 時，會將NOPORTDOCS 所對應的相關參數指定進去。

有時候設定較人性化的ports 會在安裝前提供參數供選擇，但是其實大部份的ports 都沒有提供，因此必須自行去搜尋可編譯的參數，在此我提供的方式如下：

```
# cd /usr/ports/sysutils/portupgrade
```

```
# grep defined Makefile
```

如此幾乎可以知道所有提供的可編譯參數，雖然有時會多出其它的資料，不會這個確實是一個不錯可參考的方式。

**25. 如何手動加入編譯ports 的參數？**

在前面所述的方法，是ports collections 有提供的前提之下，有時候並不是所有該軟體所支援的參數都會收納在ports collections 中，因此有時候會需要手動加入編譯的參數。

如ftp/pure-ftpd 中，如果不想把inetd 的支援編入的選項，並沒有被ports collections 所納入，因此必須手動加上這個編譯參數，如下：

```
# cd /usr/ports/ftp/pure-ftpd
```

```
# make CONFIGURE_ARGS+="--without-inetd" install
```

**26. 如何指定ports 的安裝路徑？**

預設ports collections 已安排安裝的路徑(/usr/local/)，如果不想將套件安裝在預設路徑的話，可以手動指定安裝路徑。以editors/joe 為例，則：

```
# cd /usr/ports/editors/joe
```

```
# make PREFIX=/usr install
```

那麼joe 就會將檔案對應在/usr 目錄下，而不是預設的/usr/local 目錄下。

**27. 安裝ports 出現FORCE\_PKG\_REGISTER 的錯誤訊息**

請參考Ohaha 的FAQ (<http://ohaha.ks.edu.tw/faq-0003.htm>)。

**28. 安裝ports 出現Shared object libintl.so.X not found 的錯誤訊息**

請參考Ohaha 的FAQ (<http://ohaha.ks.edu.tw/faq-0004.htm>)。

**29. 如何安裝packages ?**

目前FreeBSD 的packages 是由.tgz 所打包。如果想安裝一個packages ，可使用pkg\_add，如安裝一個joe 的tgz：

```
# pkg_add joe-{version}.tgz
```

**30. 如何强制安裝packages ?**

由於有些packages 會有與其他packages 相依性的關係，所以必須先行安裝那些packages 才能正常安裝。

如果須要强制安裝packages ，可以不須安裝那些有相依性packages ，但要注意的是强制安裝的結果可能會導致執行或運作的不正常。强制安裝packages 的指令如下，如强制安裝一個joe 的tgz：

```
# pkg_add -f joe-{version}.tgz
```

**31. 如何查詢packages 與其他packages 之間的相依性？**

由於有些packages 會有與其他packages 相依性的關係，所以必須先行安裝那些packages 才能正常安裝。

查詢packages 與其他packages 之間的相依性的指令如下，如查詢與portupgrade 相依的其他packages：

```
# pkg_info -r portupgrade-{version}.tgz
```

**32. 如何安裝遠端的packages ?**

有兩種設定方式，首先是PACKAGEROOT，如：

```
# setenv PACKAGEROOT ftp://ftp.tw.freebsd.org
```

另一種方式是設定PACKAGESITE，好處是若該站台的packages 倉儲並不是符合官方的設定或你想自己指定一個路徑。

以ftp.tw.freebsd.org i386 的current packages 為例：

```
# setenv PACKAGESITE ftp://ftp.tw.freebsd.org/pub/FreeBSD/ports/i386/packages-current/Latest/
```

兩種方式選擇其中一種皆可，接下來的步驟都一樣。

往後要安裝packages 的時候，如portupgrade 的話，則：

```
# pkg_add -r portupgrade
```

則系統會自動於ftp.tw.freebsd.org 抓取所有與portupgrade 相依的packages 並安裝。

### 33. 如何更新INDEX 對照表？

4.x 的INDEX 位於/usr/ports/INDEX，5.x 位於/usr/ports/INDEX-5，以此類推。

INDEX 是對照ports 所有相依等資訊的對照表，若長期未更新的話，會導致ports 對照失敗。官方有定期更新INDEX，如果想自行更新的話，則：

```
# cd /usr/ports
# make index
```

如果想直接下載官方最新的INDEX，則：

```
# cd /usr/ports
# make fetchindex
```

### 34. 如何更新INDEX HTML？

FreeBSD 提供了用網頁的方式來觀看ports collection，即可使用lynx, w3m, links 或其它可瀏覽網頁的程式來查閱。製作全部ports collection 的方式如下：

```
# cd /usr/ports
# make readmes
```

如果是僅須要做目前目錄下的資訊，或單一更新某一分類下的資訊，如/usr/ports/ftp 的話，則：

```
# cd /usr/ports/ftp
# make readme
```

則僅會更新/usr/ports/ftp 這個目錄的資訊，其上與其下的目錄皆不會更動到。執行成功後，會在相對應的目錄下產生README.html 的檔案。

### 35. 如何針對某些ports 不做ports update？

若ports 底下的某些分類完全用不到，則可以在ports update 時，不做更新，以節省網路頻寬與時間。只要修改/usr/sup/refuse 這個檔。

```
# ports/french
# ports/german
# ports/lang/perl5.8
```

則上列目錄將不會更新。

**36. make deinstall 與pkg\_delete 有什麼不同？**

簡單來說，make deinstall 會移除該port，並且不會參照其相依的相關套件。

pkg\_delete 在移除該port 前，會參照其相依的相關套件，並且pkg\_delete 有支援wild card，如要移除所有p開頭的ports，則：

```
# cd /var/db/pkg  
  
# pkg_delete p*
```

要注意的是，如果使用make deinstall，則最好確定系統目前所安裝的版本，與ports collection 中顯示的版本符合，否則有可能會出現非預期性錯誤；而pkg\_delete 是因為直接刪除系統中所安裝的版本，所以沒有此問題。

因此，換句話說，當系統所安裝的版本符合ports collections 中的版本，則可以使用make deinstall 或pkg\_delete，否則的話最好用pkg\_delete。

# Bibliography

- FreeBSD Unleashed*, Michael Urban & Brian Tiemann, Sams, 1st edition, 992 pages, October 2001, ISBN 0-67232-206-4.
- 4.4BSD System Manager's Manual*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 804 pages, ISBN 1-56592-080-5.
- 4.4BSD User's Reference Manual*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 905 pages, ISBN 1-56592-075-9.
- 4.4BSD User's Supplementary Documents*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 712 pages, ISBN 1-56592-076-7.
- 4.4BSD Programmer's Reference Manual*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 866 pages, ISBN 1-56592-078-3.
- 4.4BSD Programmer's Supplementary Documents*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 596 pages, ISBN 1-56592-079-1.
- The Design and Implementation of the 4.4BSD Operating System*, M. K. McKusick, Kirk Marshall, Keith Bostic, Michael J Karels, & John Quarterman, Addison-Wesley, Reading, 1996, ISBN 0-201-54979-4.
- The Design and Implementation of the FreeBSD Operating System*, M. K. McKusick & George V. Neville-Neil, Addison-Wesley, Boston, 2004, ISBN 0-201-70245-2.
- Unix System Administration Handbook*, Evi Nemeth, Garth Snyder, Scott Seebass, Trent R. Hein, & John Quarterman, Prentice-Hall, 3rd edition, 2000, ISBN 0-13-020601-6.
- The Complete FreeBSD*, Greg Lehey, Walnut Creek, 3rd edition, June 1999, 773 pages, ISBN 1-57176-246-9.
- [McKusick et al, 1994] *Berkeley Software Architecture Manual, 4.4BSD Edition*, M. K. McKusick, M. J. Karels, S. J. Leffler, W. N. Joy, & R. S. Faber, 5:1-42.
- FreeBSD for PC 98'ers (in Japanese)*, SHUWA System Co, LTD., ISBN 4-87966-468-5 C3055 P2900E.
- FreeBSD (in Japanese)*, CUTT, ISBN 4-906391-22-2.
- Complete Introduction to FreeBSD (in Japanese)*, Shoeisha Co., Ltd, ISBN 4-88135-473-6 P3600E.
- Personal UNIX Starter Kit FreeBSD (in Japanese)*, ASCII, ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (Japanese translation)*, ASCII, ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (in German)*, Computer und Literature Verlag/Vertrieb Hanser, 1998, ISBN 3-932311-31-0.
- FreeBSD install and Utilization Manual (in Japanese)*, Mainichi Communications Inc..
- Building Internet Server with FreeBSD (in Indonesia Language)*, Elex Media Komputindo, Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, & Widjil Widodo.

- The FreeBSD Corporate Networker's Guide*, Addison-Wesley.
- UNIX in a Nutshell*, O'Reilly & Associates, Inc., 1990, ISBN 093717520X.
- What You Need To Know When You Can't Find Your Unix System Administrator*, O'Reilly & Associates, Inc., 1995, Linda Mui, ISBN 1-56592-104-6.
- FreeBSD User's Reference Manual (Japanese translation)*, Mainichi Communications Inc., Jpman Project, Japan FreeBSD Users Group, 1998, ISBN 4-8399-0088-4 P3800E.
- Online Guide for newcomers to the UNIX environment* (<http://unixhelp.ed.ac.uk/>), Edinburgh University (<http://www.ed.ac.uk/>).
- DNS and BIND*, O'Reilly & Associates, Inc, ISBN 1-56592-512-2, Paul Albitz Albitz 卄Cricket Liu, 1998, 3rd edition.
- Sendmail*, O'Reilly & Associates, Inc, 1997, 2nd edition, Brian Costales, ISBN 1-56592-222-0.
- Essential System Administration*, Aileen Frisch, 2nd edition, O'Reilly & Associates, 1995, ISBN 1-56592-127-5.
- TCP/IP Network Administration*, Craig Hunt, 2nd edition, O'Reilly & Associates, Inc, 1997, ISBN 1-56592-322-7.
- Managing NFS and NIS*, Hal Stern, O'Reilly & Associates, Inc, 1991, ISBN 0-937175-75-7.
- FreeBSD System Administration's Manual* (<http://www.pc.mycom.co.jp/FreeBSD/sam.html>), Jpman Project, Japan FreeBSD Users Group (<http://www.jp.FreeBSD.org>), Mainichi Communications Inc. (<http://www.pc.mycom.co.jp/>), 1998, ISBN 4-8399-0109-0 P3300E.
- X Window System Toolkit*, Digital Press, Paul Asente, ISBN 1-55558-051-3.
- C: A Reference Manual*, Prentice Hall, 1995, 4th edition, Samuel P. Harbison 卄Guy L. Jr. Steele, ISBN 0-13-326224-3.
- The C Programming Language*, Prentice Hall, 1998, Brian Kernighan 卄Dennis Ritchie, ISBN 0-13-110362-9.
- Porting UNIX Software*, Greg Lehey, O'Reilly & Associates, Inc., 1995, ISBN 1-56592-126-7.
- The Standard C Library*, Prentice Hall, 1992, P. J. Plauger, ISBN 0-13-131509-9.
- Advanced Programming in the UNIX Environment*, Addison-Wesley, 1992, W. Richard Stevens, ISBN 0-201-56317-7.
- UNIX Network Programming*, W. Richard Stevens, Prentice Hall, 1998, 2nd edition, ISBN 0-13-490012-X.
- Writing Serial Drivers for UNIX*, Bill Wells, December 1994, Dr. Dobb's Journal, pp68-71, pp97-99.
- UNIX System Architecture*, Prentice-Hall, Inc, 1990, Prabhat K. Andleigh, ISBN 0-13-949843-5.
- Porting UNIX to the 386*, William Jolitz, Dr. Dobb's Journal, January 1991-July 1992.
- TCP/IP Illustrated, Volume 1: The Protocols*, W. Richard Stevens, Addison-Wesley, 1996, ISBN 0-201-63346-9.
- Unix Systems for Modern Architectures*, Addison-Wesley, Curt Schimmel, 1994, ISBN 0-201-63338-8.

- TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*, Addison-Wesley, 1996, W. Richard Stevens, ISBN 0-201-63495-3.
- UNIX Internals -- The New Frontiers*, Uresh Vahalia, Prentice Hall, 1996, ISBN 0-13-101908-2.
- TCP/IP Illustrated, Volume 2: The Implementation*, Gary R. Wright & W. Richard Stevens, 1995, Addison-Wesley, ISBN 0-201-63354-X.
- Firewalls and Internet Security: Repelling the Wily Hacker*, William R. Cheswick & Steven M. Bellovin, Addison-Wesley, 1995, ISBN 0-201-63357-4.
- Practical UNIX Security*, Simson Garfinkel & Gene Spafford, 1996, 2nd edition, O'Reilly & Associates, Inc, ISBN 1-56592-148-8.
- PGP Pretty Good Privacy*, Simson Garfinkel, O'Reilly & Associates, Inc, 1995, ISBN 1-56592-098-8.
- Pentium Processor System Architecture*, Don Anderson & Tom Shanley, Addison-Wesley, 1995, 2nd edition, ISBN 0-201-40992-5.
- Programmer's Guide to the EGA, VGA, and Super VGA Cards*, Richard F. Ferraro, 3rd edition, Addison-Wesley, 1995, ISBN 0-201-62490-7.
- 80486 System Architecture*, Tom Shanley, Addison-Wesley, 1995, 3rd edition, ISBN 0-201-40994-1.
- ISA System Architecture*, Tom Shanley, Addison-Wesley, 3rd edition, 1995, ISBN 0-201-40996-8.
- PCI System Architecture*, Tom Shanley, Addison-Wesley, 1995, 3rd edition, ISBN 0-201-40993-3.
- The Undocumented PC*, Frank Van Gilluwe, Addison-Wesley, 1994, ISBN 0-201-62277-7.
- Bell System Technical Journal, Unix Time-Sharing System*, American Telephone & Telegraph Company, July-August 1978, Vol 57, No 6, Part 2, ISSN0005-8580.
- Lion's Commentary on UNIX*, John Lion, ITP Media Group, 1996, 6th edition, ISBN 1573980137.
- The New Hacker's Dictionary*, Eric S. Raymond, MIT Press, 1996, 3rd edition, ISBN 0-262-68092-0.
- A quarter century of UNIX*, Peter H. Salus, Addison-Wesley, 1994, ISBN 0-201-54777-5.
- The UNIX-HATERS Handbook*, Steven Strassman, Daniel Weise, & Simon Garfinkel, IDG Books Worldwide, Inc, 1994, ISBN 1-56884-203-1.
- Life with UNIX —special edition*, Don Libes & Sandy Ressler, Prentice-Hall, 1989, ISBN 0-13-536657-7.
- The BSD Family Tree* (<http://svnweb.FreeBSD.org/base/head/share/misc/bsd-family-tree?view=co>), 1997.
- Absolute BSD*, Michael Lucas, No Starch Press, June 2002, ISBN 1-886411-74-3.
- The C/C++ Users Journal*, R&D Publications Inc., ISSN 1075-2838.
- Sys Admin —The Journal for UNIX System Administrators*, Miller Freeman, Inc, ISSN 1061-2688.

---