Óõ÷íÝò ÅñùôÞóåéò ãéá ôï FreeBSD 6.*x* êáé 7.*x*

Ç ÏìÜäá Ôåêìçñßùóçò ôïõ FreeBSD

Óõ÷íÝò ÅñùôÞóåéò ãéá ôï FreeBSD 6.*x* êáé 7.*x*

áðü Ç ÏìÜäá Ôåêìçñßùóçò ôïõ FreeBSD

ÄçìïóéåõìÝíï $FreeBSD: doc/el_GR.ISO8859-7/books/faq/book.sgml,v 1.16 2009/02/09 18:04:38 keramida Exp $

ÐíåõìáôéêÜ Äéêáéþìáôá © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Ç ÏìÜäá Ôåêìçñßùóçò ôïõ FreeBSD

Ôï êåßìåíï áõôü ðåñéÝ÷åé ôéò Óõ÷íÝò ÅñùôÞóåéò ãéá ôéò åêäüóåéò 6.*x* êáé 7.*x* ôïõ FreeBSD. ¼ëåò ïé êáôá÷ùñÞóåéò èåùñåßôáé üôé éó÷ýïõí ãéá ôçí Ýêäïóç 6.*x* êáé ôéò ìåôáãåíÝóôåñåò åêäüóåéò, åêôüò áí áíáöÝñåôáé äéáöïñåôéêÜ. Áí åíäéáöÝñåóôå íá ìáò âïçèÞóåôå óå áõôü ôï Ýñãï, óôåßëôå Ýíá email óôçí çëåêôñïíéêÞ ëßóôá ïìÜäáò ôåêìçñßùóçò ôïõ FreeBSD (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc). Ç ôåëåõôáßá Ýêäïóç áõôïý ôïõ êåéìÝíïõ åßíáé ðÜíôïôå äéáèÝóéìç óôçí äéêôõáêÞ ôïðïèåóßá ôïõ FreeBSD (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/faq/index.html). Ìðïñåßôå åðßóçò íá ôçí êáôåâÜóåôå ùò Ýíá ìåãÜëï áñ÷åßï HTML (book.html) ìÝóù HTTP Þ áêüìá êáé ùò áðëü êåßìåíï, PostScript®, PDF, êëð. áðü ôïí åîõðçñåôçôÞ FTP ôïõ FreeBSD (ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/). Ìðïñåßôå áêüìá íá øÜîåôå óôéò Óõ÷íÝò ÅñùôÞóåéò (http://www.FreeBSD.org/search/index.html).

Ç äéáíïìÞ êáé ÷ñÞóç óå ìïñöÞ ðçãáßïõ êþäéêá (SGML DocBook) Þ óå ’ìåôáãëùôôéóìÝíç’ ìïñöÞ (SGML, HTML, PDF, PostScript, RTF êïê) ìå Þ ÷ùñßò áëëáãÝò, åðéôñÝðåôáé åöüóïí ïé ðáñáêÜôù ðñïûðïèÝóåéò ôçñïýíôáé:

1. Ç äéáíïìÞ óå ìïñöÞ ðçãáßïõ êþäéêá (SGML DocBook) ðñÝðåé íá äéáôçñåß ôçí ðáñáðÜíù äÞëùóç ðíåõìáôéêþí äéêáéùìÜôùí, áõôÞ ôç ëßóôá ìå ðñïûðïèÝóåéò êáé ôçí åðüìåíç ðáñÜãñáöï óôéò ðñþôåò ãñáììÝò ôïõ áñ÷åßïõ, áìåôÜâëçôåò.
2. Ç äéáíïìÞ óå ìåôáãëùôôéóìÝíåò ìïñöÝò (ìåôÜöñáóç óå Üëëá DTD, ìåôáôñïðÞ óå PDF, PostScript, RTF Þ Üëëåò ìïñöÝò) ðñÝðåé íá áíáðáñÜãåé ôçí ðáñáðÜíù äÞëùóç ðíåõìáôéêþí äéêáéùìÜôùí, áõôÞ ôç ëßóôá ìå ðñïûðïèÝóåéò, êáé ôçí ðáñáêÜôù ðáñÜãñáöï ôüóï óôçí ôåêìçñßùóç üóï êáé óå Üëëï õëéêü ðïõ ðáñÝ÷åôáé ìáæß ìå ôçí äéáíïìÞ.

Óçìáíôéêü: ÁÕÔÇ Ç ÔÅÊÌÇÑÉÙÓÇ ÄÉÍÅÔÁÉ ÁÐÏ ÔÇÍ ÏÌÁÄÁ ÔÅÊÌÇÑÉÙÓÇÓ ÔÏÕ FREEBSD "ÙÓ Å×ÅÉ" ÊÁÉ ÄÅÍ ÐÁÑÅ×ÅÔÁÉ ÊÁÌÉÁ ÁÌÅÓÇ Ç ÅÌÌÅÓÇ ÅÃÃÕÇÓÇ, ÓÕÌÐÅÑÉËÁÌÂÁÍÏÌÅÍÙÍ, ÁËËÁ ×ÙÑÉÓ ÍÁ ÐÅÑÉÏÑÉÆÅÔÁÉ ÌÏÍÏ ÓÅ ÁÕÔÅÓ, ÊÁÉ ÔÙÍ ÅÌÌÅÓÙÍ ÅÃÃÕÇÓÅÙÍ ÃÉÁ ÅÌÐÏÑÅÕÓÉÌÏÔÇÔÁ Ç ÊÁÔÁËËÇËÏÔÇÔÁ ÃÉÁ ÏÐÏÉÏÍÄÇÐÏÔÅ ÓÕÃÊÅÊÑÉÌÅÍÏ ÓÊÏÐÏ. ÓÅ ÊÁÌÉÁ ÐÅÑÉÐÔÙÓÇ ÄÅÍ ÅÕÈÕÍÅÔÁÉ Ç ÏÌÁÄÁ ÔÅÊÌÇÑÉÙÓÇÓ ÔÏÕ FREEBSD ÃÉÁ ÏÐÏÉÅÓÄÇÐÏÔÅ ÁÌÅÓÅÓ, ÅÌÌÅÓÅÓ, ÔÕ×ÁÉÅÓ, ÅÉÄÉÊÅÓ, ÓÇÌÁÍÔÉÊÅÓ, Ç ÊÁÔÁ ÐÅÑÉÐÔÙÓÇ ÂËÁÂÅÓ (ÓÕÌÐÅÑÉËÁÌÂÁÍÏÌÅÍÙÍ, ÁËËÁ ×ÙÑÉÓ ÍÁ ÐÅÑÉÏÑÉÆÅÔÁÉ ÌÏÍÏ ÓÅ ÁÕÔÅÓ, ÊÁÉ ÔÇÍ ÁÄÕÍÁÌÉÁ ÐÑÏÓÂÁÓÇÓ ÓÅ ÅÍÁËËÁÊÔÉÊÅÓ ÐÇÃÅÓ Ç ÕÐÇÑÅÓÉÅÓ, ÔÇÍ ÁÄÕÍÁÌÉÁ ×ÑÇÓÇÓ, ÔÇÍ ÁÐÙËÅÉÁ ÄÅÄÏÌÅÍÙÍ Ç ÊÅÑÄÏÕÓ, ÊÁÉ ÔÇÍ ÄÉÁÊÏÐÇ ÅÐÉ×ÅÉÑÇÌÁÔÉÊÙÍ ËÅÉÔÏÕÑÃÉÙÍ), ÐÏÕ ÐÑÏÊÁËÏÕÍÔÁÉ ÌÅ ÏÐÏÉÏÄÇÐÏÔÅ ÔÑÏÐÏ ÁÐÏ ÔÇ ×ÑÇÓÇ ÁÕÔÇÓ ÔÇÓ ÔÅÊÌÇÑÉÙÓÇÓ.

Ôï FreeBSD åßíáé Ýíá êáôï÷õñùìÝíï åìðïñéêü óýìâïëï ôïõ FreeBSD Foundation.

Ïé ëÝîåéò 3Com êáé HomeConnect åßíáé êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá ôçò 3Com Corporation.

Ïé ëÝîåéò Þ öñÜóåéò Adobe, Acrobat, Acrobat Reader, êáé PostScript åßíáé åßôå êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá Þ åìðïñéêÜ óýìâïëá ôçò Adobe Systems Incorporated óôéò ÇíùìÝíåò Ðïëéôåßåò Þ/êáé óå Üëëåò ÷þñåò.

Ç öñÜóç Sound Blaster åßíáé åìðïñéêü óýìâïëï ôçò Creative Technology Ltd. óôéò ÇíùìÝíåò Ðïëéôåßåò Þ/êáé óå Üëëåò ÷þñåò.

Ç ëÝîç CVSup åßíáé êáôï÷õñùìÝíï åìðïñéêü óýìâïëï ôïõ John D. Polstra.

Ïé ëÝîåéò Þ öñÜóåéò IBM, AIX, EtherJet, Netfinity, OS/2, PowerPC, PS/2, S/390, êáé ThinkPad åßíáé åìðïñéêÜ óýìâïëá ôçò International Business Machines Corporation óôéò ÇíùìÝíåò Ðïëéôåßåò, Üëëåò ÷þñåò, Þ êáé óôá äýï ôáõôü÷ñïíá.

Ïé ëÝîåéò IEEE, POSIX, êáé 802 åßíáé êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá ôïõ Institute of Electrical and Electronics Engineers, Inc. óôéò ÇíùìÝíåò Ðïëéôåßåò.

Ïé ëÝîåéò Intel, Celeron, EtherExpress, i386, i486, Itanium, Pentium, êáé Xeon åßíáé åìðïñéêÜ óýìâïëá Þ êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá ôçò Intel Corporation êáé ôùí èõãáôñéêþí ôçò óôéò ÇíùìÝíåò Ðïëéôåßåò êáé óå Üëëåò ÷þñåò.

Ïé ëÝîåéò Iomega, Zip, êáé Jaz åßíáé åßôå êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá Þ åìðïñéêÜ óýìâïëá ôçò Iomega Corporation óôéò ÇíùìÝíåò Ðïëéôåßåò Þ/êáé óå Üëëåò ÷þñåò.

Ôï Linux åßíáé Ýíá êáôï÷õñùìÝíá åìðïñéêü óýìâïëï ôïõ Linus Torvalds óôéò ÇíùìÝíåò Ðïëéôåßåò.

Ïé ëÝîåéò Microsoft, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media, êáé Windows NT åßíáé åßôå êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá Þ åìðïñéêÜ óýìâïëá ôçò Microsoft Corporation óôéò ÇíùìÝíåò Ðïëéôåßåò êáé/Þ óå Üëëåò ÷þñåò.

Ïé ëÝîåéò MIPS êáé R4000 åßíáé êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá ôçò MIPS Technologies, Inc. óôéò ÇíùìÝíåò Ðïëéôåßåò êáé óå Üëëåò ÷þñåò.

Ïé ëÝîåéò Netscape êáé Netscape Navigator åßíáé êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá ôçò Netscape Communications Corporation óôéò Ç.Ð.Á êáé Üëëåò ÷þñåò.

Ïé ëÝîåéò Motif, OSF/1, êáé UNIX åßíáé êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá êáé ïé ëÝîåéò Þ öñÜóåéò IT DialTone êáé The Open Group åßíáé åìðïñéêÜ óýìâïëá ôïõ The Open Group óôéò ÇíùìÝíåò Ðïëéôåßåò êáé óå Üëëåò ÷þñåò.

Ç ëÝîç Oracle åßíáé êáôï÷õñùìÝíï åìðïñéêü óýìâïëï ôçò Oracle Corporation.

Ïé ëÝîåéò ïé öñÜóåéò Silicon Graphics, SGI, êáé OpenGL åßíáé êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá ôçò Silicon Graphics, Inc. óôéò ÇíùìÝíåò Ðïëéôåßåò êáé/Þ óå Üëëåò ÷þñåò óôïí êüóìï.

Ïé ëÝîåéò Þ öñÜóåéò Sparc, Sparc64, SPARCEngine, êáé UltraSPARC åßíáé åìðïñéêÜ óýìâïëá ôçò SPARC International, Inc. óôéò ÇíùìÝíåò Ðïëéôåßåò êáé óå Üëëåò ÷þñåò. Ôá ðñïúüíôá ðïõ öÝñïõí ôá åìðïñéêÜ óýìâïëá SPARC åßíáé âáóéóìÝíá óôçí áñ÷éôåêôïíéêÞ ðïõ áíÝðôõîå ç Sun Microsystems, Inc.

Ïé ëÝîåéò Þ öñÜóåéò Sun, Sun Microsystems, Java, Java Virtual Machine, JavaServer Pages, JDK, JRE, JSP, JVM, Netra, Solaris, StarOffice, Sun Blade, Sun Enterprise, Sun Fire, SunOS, Ultra êáé VirtualBox åßíáé åìðïñéêÜ óýìâïëá Þ êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá ôçò Sun Microsystems, Inc. óôéò ÇíùìÝíåò Ðïëéôåßåò êáé óå Üëëåò ÷þñåò.

Ïé ëÝîåéò Þ öñÜóåéò U.S. Robotics êáé Sportster åßíáé êáôï÷õñùìÝíá åìðïñéêÜ óýìâïëá ôçò U.S. Robotics Corporation.

Ç ëÝîç XFree86 åßíáé Ýíá åìðïñéêü óýìâïëï ôïõ The XFree86 Project, Inc.

ÐïëëÝò áðü ôéò ëÝîåéò Þ öñÜóåéò ïé ïðïßåò ÷ñçóéìïðïéïýíôáé áðü ôïõò êáôáóêåõáóôÝò Þ ôïõò ðùëçôÝò ôïõò ãéá íá äéáêñßíïõí ôá ðñïúüíôá ôïõò èåùñïýíôáé åìðïñéêÜ óýìâïëá. ¼ðïõ áõôÝò åìöáíßæïíôáé óå áõôü ôï êåßìåíï êáé ãéá üóåò áðü áõôÝò ãíùñßæåé ç ÏìÜäá ÁíÜðôõîçò ôïõ FreeBSD üôé åßíáé ðéèáíüí íá åßíáé åìðïñéêÜ óýìâïëá, èá äåßôå Ýíá áðü ôá óýìâïëá: “™” Þ “®”.

Ðßíáêáò Ðåñéå÷ïìÝíùí

1 ÅéóáãùãÞ 1
2 Ôåêìçñßùóç êáé ÕðïóôÞñéîç 7
3 ÅãêáôÜóôáóç 12
4 Óõìâáôüôçôá Õëéêïý 22
  4.1 ÃåíéêÜ 22
  4.2 Áñ÷éôåêôïíéêÝò êáé ÅðåîåñãáóôÝò 22
  4.3 Óêëçñïß äßóêïé, ìïíÜäåò ôáéíßáò, ïäçãïß CD êáé DVD 23
  4.4 Ðëçêôñïëüãéá êáé ðïíôßêéá 24
  4.5 ÓõóêåõÝò óåéñéáêÞò åðéêïéíùíßáò êáé Äéêôýùóçò 27
  4.6 ÓõóêåõÝò Þ÷ïõ 28
  4.7 ¶ëëï õëéêü 29
5 Áíôéìåôþðéóç ÐñïâëçìÜôùí 31
6 ÅìðïñéêÝò ÅöáñìïãÝò 46
7 ÅöáñìïãÝò Ôåëéêïý ×ñÞóôç 49
8 Ñýèìéóç ÐõñÞíá 53
9 Äßóêïé, ÓõóôÞìáôá Áñ÷åßùí êáé ÖïñôùôÝò Åêêßíçóçò 56
10 Äéá÷åßñéóç ÓõóôÞìáôïò 67
11 Ôï Óýóôçìá X Windows êáé ïé ÅéêïíéêÝò Êïíóüëåò 75
12 Äéêôýùóç 84
13 ÁóöÜëåéá 90
14 PPP 93
15 ÓåéñéáêÝò Åðéêïéíùíßåò 105
16 ÄéÜöïñåò ÅñùôÞóåéò 112
17 ×éïýìïñ êáé FreeBSD 116
18 Ðñï÷ùñçìÝíá ÈÝìáôá 119
19 Åõ÷áñéóôßåò 126
Bibliography 127

ÊáôÜëïãïò ÐéíÜêùí

3-1. ÌÝãéóôá ìåãÝèç áñ÷åßùí 19
12-1. Network cards based on the DEC PCI chipset 86

ÊåöÜëáéï 1 ÅéóáãùãÞ

Êáëþò Þëèáôå óôéò Óõ÷íÝò ÅñùôÞóåéò ôïõ FreeBSD 6.*x*-7.*x*!

¼ðùò óõíçèßæåôáé óôá FAQs ôïõ Usenet, ôï êåßìåíï áõôü Ý÷åé óêïðü íá êáëýøåé ôéò ðéï óõíçèéóìÝíåò åñùôÞóåéò ðïõ áöïñïýí ôï ëåéôïõñãéêü óýóôçìá FreeBSD (êáé öõóéêÜ íá ôéò áðáíôÞóåé!). Áí êáé ï áñ÷éêüò óêïðüò ôùí FAQs Þôáí ç åîïéêïíüìçóç åýñïõò æþíçò ôïõ äéêôýïõ áðü ôçí åðáíáëáìâáíüìåíç áðÜíôçóç ôùí ßäéùí åñùôÞóåùí, ôá FAQs áíáãíùñßóôçêáí ôåëéêÜ ùò ðïëýôéìåò ðçãÝò ðëçñïöïñéþí.

¸÷åé êáôáâëçèåß êÜèå ðñïóðÜèåéá þóôå áõôü ôï FAQ íá ðåñéÝ÷åé üóï ôï äõíáôüí ðåñéóóüôåñåò ðëçñïöïñßåò. Áí èÝëåôå íá êÜíåôå êÜðïéåò õðïäåßîåéò ó÷åôéêÜ ìå ôñüðïõò âåëôßùóçò ôïõ, óáò ðáñáêáëïýìå óôåßëôå ìáò Ýíá email óôç çëåêôñïíéêÞ ëßóôá ïìÜäáò ôåêìçñßùóçò ôïõ FreeBSD (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc).

1. Ôé åßíáé ôï FreeBSD;

Åí óõíôïìßá, ôï FreeBSD åßíáé Ýíá ëåéôïõñãéêü óýóôçìá ôýðïõ UNIX® ãéá ôéò áñ÷éôåêôïíéêÝò AMD64 êáé Intel® EM64T, i386™ PC-98, IA-64, ARM®, PowerPC® êáé UltraSPARC®. Âáóßæåôáé óôçí Ýêäïóç “4.4BSD-Lite” ôïõ Ðáíåðéóôçìßïõ ôïõ Berkeley, ìå êÜðïéåò âåëôéþóåéò áðü ôï “4.4BSD-Lite2”. Âáóßæåôáé åðßóçò Ýììåóá óôçí åñãáóßá ìåôáöïñÜò (port) ôïõ “Net/2” (åðßóçò áðü ôï Berkeley) ðïõ Ýãéíå áðü ôïí William Jolitz (ç ïðïßá Ýãéíå ãíùóôÞ ìå ôï üíïìá “386BSD”), áí êáé äåí Ý÷åé ìåßíåé ó÷åäüí ôßðïôá áðü ôïí áñ÷éêü êþäéêá ôïõ 386BSD. ÐëçñÝóôåñç ðåñéãñáöÞ ãéá ôï ôé åßíáé ôï FreeBSD êáé ðùò ìðïñåß íá ëåéôïõñãÞóåé ãéá åóÜò, ìðïñåßôå íá âñåßôå óôçí äéêôõáêÞ ôïðïèåóßá ôïõ FreeBSD (http://www.FreeBSD.org/index.html).

Ôï FreeBSD ÷ñçóéìïðïéåßôáé áðü åôáéñßåò, ðáñï÷åßò õðçñåóéþí Internet (ISPs), åñåõíçôÝò, åðáããåëìáôßåò ôçò ÐëçñïöïñéêÞò, öïéôçôÝò êáé ïéêéáêïýò ÷ñÞóôåò áðü üëï ôïí êüóìï, óôç äïõëåéÜ ôïõò, óôçí åêðáßäåõóç êáé ãéá áíáøõ÷Þ.

Ãéá ðåñéóóüôåñåò ëåðôïìÝñåéåò ó÷åôéêÜ ìå ôï FreeBSD, ðáñáêáëïýìå äéáâÜóôå ôï Åã÷åéñßäéï ôïõ FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/index.html).

2. Ðïéïò åßíáé ï óêïðüò ôïõ FreeBSD Project;

Ï óêïðüò ôïõ FreeBSD Project åßíáé íá ðáñÝ÷åé ëïãéóìéêü ôï ïðïßï íá ìðïñåß íá ÷ñçóéìïðïéçèåß ãéá ïðïéïõóäÞðïôå óêïðïýò êáé ÷ùñßò ðåñéïñéóìïýò. Ðïëëïß áðü åìÜò Ý÷ïõìå åðåíäýóåé óçìáíôéêÜ óôïí êþäéêá (êáé óôï ßäéï ôï Ýñãï) êáé óßãïõñá äåí èá ìáò ðåñßóáå êáé êÜðïéá ÷ñçìáôéêÞ áðïæçìßùóç ðïõ êáé ðïõ, áëëÜ óßãïõñá äåí åðéìÝíïõìå óå áõôü. Ðéóôåýïõìå üôé ç ðñþôç êáé âáóéêüôåñç “áðïóôïëÞ” ìáò åßíáé íá ðáñÝ÷ïõìå êþäéêá óå üëïõò, ãéá ïðïéïäÞðïôå ðéèáíü óêïðü, Ýôóé þóôå ï êþäéêáò ìáò íá Ý÷åé ôçí åõñýôåñç äõíáôÞ ÷ñÞóç êáé åðéôõã÷Üíåé ôï ìåãáëýôåñï äõíáôü üöåëïò. Ðéóôåýïõìå üôé áõôüò åßíáé Ýíáò áðü ôïõò óçìáíôéêüôåñïõò óêïðïýò ôïõ Åëåýèåñïõ Ëïãéóìéêïý êáé ôïí õðïóôçñßæïõìå èåñìÜ.

Ï êþäéêáò, óôï äÝíôñï ôïõ ðçãáßïõ ìáò êþäéêá, ï ïðïßïò åìðßðôåé óôçí Üäåéá ÷ñÞóçò GNU General Public License (GPL) (http://www.FreeBSD.org/copyright/COPYING) Þ GNU Library General Public License (LGPL) (http://www.FreeBSD.org/copyright/COPYING.LIB) Ý÷åé ðåñéóóüôåñïõò ðåñéïñéóìïýò, áëëÜ áõôïß êëßíïõí, åõôõ÷þò, ðñïò ôçí õðï÷ñÝùóç åëåýèåñçò ðñüóâáóçò áíôß ãéá ôï áíôßèåôï ðïõ åßíáé êáé ôï ðëÝïí óõíçèéóìÝíï. Ëüãù ôçò áõîçìÝíçò ðïëõðëïêüôçôáò ðïõ ìðïñåß íá ðñïêýøåé áðü ôçí åìðïñéêÞ ÷ñÞóç ëïãéóìéêïý GPL, ðñïóðáèïýìå ãåíéêÜ íá áíôéêáôáóôÞóïõìå áõôü ôï ëïãéóìéêü ìå áíôßóôïé÷ï õðü ôçí ðéï ÷áëáñÞ Üäåéá FreeBSD (http://www.FreeBSD.org/copyright/freebsd-license.html), üðïõ áõôü åßíáé äõíáôü.

3. ÕðÜñ÷ïõí êÜðïéïé ðåñéïñéóìïß óôçí Üäåéá ôïõ FreeBSD;

Íáé. Ïé ðåñéïñéóìïß áõôïß äåí åëÝã÷ïõí ðùò ÷ñçóéìïðïéåßôå ôïí êþäéêá, áëëÜ ìüíï ðùò ìåôá÷åéñßæåóôå ôï ßäéï ôï FreeBSD Project. Áí óáò åíäéáöÝñåé óïâáñÜ ç Üäåéá, äéáâÜóôå ôçí åäþ (http://www.FreeBSD.org/copyright/freebsd-license.html). Ãéá ôïõò áðëþò ðåñßåñãïõò, ç Üäåéá ðåñéëçðôéêÜ ãñÜöåé:

- Ìçí éó÷õñéóèåßôå üôé ãñÜøáôå áõôü ôï ëïãéóìéêü.
- Ìç ìáò ìçíýóåôå áí ÷áëÜóåé.

4. Ìðïñåß ôï FreeBSD íá áíôéêáôáóôÞóåé ôï ôñÝ÷ïí ëåéôïõñãéêü ìïõ óýóôçìá;

Ãéá ôïõò ðåñéóóüôåñïõò áíèñþðïõò, íáé. ÁëëÜ áõôÞ ç áðÜíôçóç äåí ìðïñåß íá äïèåß ôüóï îåñÜ.

Ïé ðåñéóóüôåñïé Üíèñùðïé äåí ÷ñçóéìïðïéïýí óôçí ðñáãìáôéêüôçôá êÜðïéï ëåéôïõñãéêü óýóôçìá. ×ñçóéìïðïéïýí åöáñìïãÝò. Ïé åöáñìïãÝò åßíáé áõôÝò ðïõ ÷ñçóéìïðïéïýí ôï ëåéôïõñãéêü óýóôçìá. Ôï FreeBSD åßíáé ó÷åäéáóìÝíï ãéá íá ðáñÝ÷åé Ýíá óôéâáñü êáé ïëïêëçñùìÝíï ðåñéâÜëëïí ãéá åöáñìïãÝò. Õðïóôçñßæåé ìåãÜëç ðïéêéëßá áðü öõëëïìåôñçôÝò, óïõßôåò ãñáöåßïõ, ðñïãñÜììáôá çëåêôñïíéêïý ôá÷õäñïìåßïõ, ðñïãñÜììáôá ãñáöéêþí, ðåñéâÜëëïíôá ðñïãñáììáôéóìïý, åîõðçñåôçôÝò äéêôýïõ, êáé ïõóéáóôéêÜ ïôéäÞðïôå Üëëï ìðïñåß íá èåëÞóåôå. Ìðïñåßôå íá äéá÷åéñéóôåßôå ôéò ðåñéóóüôåñåò áðü áõôÝò ôéò åöáñìïãÝò ìÝóù ôçò ÓõëëïãÞò ôùí Ports (http://www.FreeBSD.org/ports/).

Áí ÷ñåéÜæåóôå íá ÷ñçóéìïðïéÞóåôå ìéá åöáñìïãÞ ðïõ åßíáé äéáèÝóéìç óå Ýíá ìüíï ëåéôïõñãéêü óýóôçìá, ôüôå áðëÜ äåí ìðïñåßôå íá áíôéêáôáóôÞóåôå áõôü ôï ëåéôïõñãéêü óýóôçìá. Åßíáé, üìùò, ðïëý ðéèáíü üôé èá âñåßôå ìéá áíôßóôïé÷ç åöáñìïãÞ óôï FreeBSD. Áí èÝëåôå Ýíáí óôáèåñü åîõðçñåôçôÞ ãéá ôï ãñáöåßï óáò Þ ãéá ôï Internet, Ýíá áîéüðéóôï óôáèìü åñãáóßáò, Þ áðëþò ôçí éêáíüôçôá íá êÜíåôå ôçí åñãáóßá óáò ÷ùñßò äéáêïðÝò, åßíáé ó÷åäüí óßãïõñï üôé ôï FreeBSD ìðïñåß íá êÜíåé üôé ÷ñåéÜæåóôå. Ðïëëïß ÷ñÞóôåò õðïëïãéóôþí óå üëï ôïí êüóìï — ôüóï áñ÷Üñéïé üóï êáé Ýìðåéñïé äéá÷åéñéóôÝò óõóôçìÜôùí UNIX — ÷ñçóéìïðïéïýí ôï FreeBSD ùò ôï âáóéêü ëåéôïõñãéêü óýóôçìá óôïõò óôáèìïýò åñãáóßáò ôïõò.

Áí Ýñ÷åóôå óôï FreeBSD áðü êÜðïéï Üëëï ðåñéâÜëëïí ôýðïõ UNIX, îÝñåôå Þäç ôá ðåñéóóüôåñá áðü áõôÜ ðïõ ÷ñåéÜæåóôå. Áí üìùò ç åìðåéñßá óáò ðñïÝñ÷åôáé áðü ëåéôïõñãéêÜ óõóôÞìáôá ìå ãñáöéêü ðåñéâÜëëïí, üðùò ôá Windows® êáé ïé ðáëéüôåñåò åêäüóåéò ôïõ Mac OS®, èá ðñÝðåé íá åðåíäýóåôå, üðùò åßíáé áíáìåíüìåíï, åðéðëÝïí ÷ñüíï ãéá íá åîïéêåéùèåßôå ìå ôïí ôñüðï åñãáóßáò óå Ýíá óýóôçìá UNIX. Áõôü ôï FAQ, êáèþò êáé ôï Åã÷åéñßäéï ôïõ FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/index.html) áðïôåëïýí åîáéñåôéêÜ áíáãíþóìáôá ãéá íá îåêéíÞóåôå.

5. Ãéáôß ïíïìÜæåôáé FreeBSD;

- Ìðïñåß íá ÷ñçóéìïðïéçèåß ÷ùñßò ÷ñÝùóç, áêüìá êáé ãéá åìðïñéêïýò óêïðïýò.
- Äéáôßèåôáé ï ðëÞñçò ðçãáßïò êþäéêáò ãéá ôï ëåéôïõñãéêü óýóôçìá, êáé ìå ôïõò åëÜ÷éóôïõò äõíáôïýò ðåñéïñéóìïýò ó÷åôéêÜ ìå ôç ÷ñÞóç ôïõ, ôç äéáíïìÞ ôïõ êáé ôçí åíóùìÜôùóç ôïõ óå Üëëá Ýñãá (åìðïñéêÜ Þ ìç).
- ÏðïéïóäÞðïôå Ý÷åé íá ðñïôåßíåé ìéá âåëôßùóç Þ äéüñèùóç, åßíáé åëåýèåñïò íá õðïâÜëëåé ôïí êþäéêá ôïõ, ï ïðïßïò êáé èá ðñïóôåèåß óôï äÝíôñï ðçãáßïõ êþäéêá (áñêåß íá ðëçñïýíôáé ìéá-äõï âáóéêÝò ðñïûðïèÝóåéò).

Áîßæåé íá áíáöÝñïõìå üôé ç ëÝîç “åëåýèåñï” ÷ñçóéìïðïéåßôáé åäþ ìå äýï ôñüðïõò, ï Ýíáò óçìáßíåé “äùñåÜí”, êáé ï Üëëïò óçìáßíåé “ìðïñåßôå íá ôï êÜíåôå üôé èÝëåôå”. Åêôüò áðü Ýíá-äõï ðñÜãìáôá ðïõ *äåí* ìðïñåßôå íá êÜíåôå ìå ôïí êþäéêá ôïõ FreeBSD, ãéá ðáñÜäåéãìá íá éó÷õñéóèåßôå üôé ôïí ãñÜøáôå, ìðïñåßôå óôá áëÞèåéá íá ôïí êÜíåôå üôé èÝëåôå.

6. Ðïéåò åßíáé ïé äéáöïñÝò ìåôáîý ôïõ FreeBSD êáé ôïõ NetBSD, OpenBSD êáé ôùí Üëëùí BSD óõóôçìÜôùí áíïéêôïý êþäéêá;

Ï James Howard Ý÷åé ãñÜøåé ìéá êáëÞ åîÞãçóç ôçò éóôïñßáò êáé ôùí äéáöïñþí ìåôáîý ôùí äéÜöïñùí Ýñãùí ãéá ôï DaemonNews (http://www.daemonnews.org/), óôï Üñèñï Ç ÏéêïãÝíåéá Ëåéôïõñãéêþí BSD (http://ezine.daemonnews.org/200104/bsd_family.html).

7. Ðïéá åßíáé ç ôåëåõôáßá Ýêäïóç ôïõ FreeBSD;

ÁõôÞ ôç óôéãìÞ õðÜñ÷ïõí äýï ðáñÜëëçëïé êëÜäïé óôçí áíÜðôõîç ôïõ FreeBSD. Ïé åðßóçìåò åêäüóåéò ôïõ FreeBSD ðáñÜãïíôáé ôáõôü÷ñïíá êáé áðü ôïõò äýï áõôïýò êëÜäïõò. Ç óåéñÜ ôùí åêäüóåùí 6.*x* äçìéïõñãåßôáé áðü ôïí êëÜäï *6-STABLE* êáé ç óåéñÜ åêäüóåùí 7.*x* äçìéïõñãåßôáé áðü ôïí êëÜäï *7-STABLE*.

ÌÝ÷ñé êáé ôçí Ýêäïóç 7.0, ç óåéñÜ åêäüóåùí 6.*x* Þôáí ãíùóôÞ ùò *-STABLE*. Ùóôüóï áðü ôçí 7.0 êáé ìåôÜ, ç óåéñÜ 6.*x* èåùñåßôáé üôé åéóÜãåôáé óôç öÜóç ôçò “åêôåôáìÝíçò õðïóôÞñéîçò” êáé èá ëáìâÜíåé ìüíï äéïñèþóåéò ãéá óïâáñÜ ðñïâëÞìáôá, üðùò ð.÷. áõôÜ ðïõ áíáöÝñïíôáé óå êåíÜ áóöáëåßáò. Èá õðÜñîïõí ðåñéóóüôåñåò åêäüóåéò áðü ôç óåéñÜ *6-STABLE*, áëëÜ áõôÞ èåùñåßôáé ðëÝïí “îåðåñáóìÝíç” (legacy) êáé ïé ðéï ðñüóöáôåò âåëôéþóåéò èá åíóùìáôùèïýí ìüíï óôïí êëÜäï *7-STABLE*.

Version 8.0 (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.0-RELEASE/) is the latest release from the *7-STABLE* branch; it was released in ÌÜéï 2009. Version 7.2 (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/7.2-RELEASE/) is the latest release from the *6-STABLE* branch; it was released in Íïå 2008.

Ç Ýêäïóç 8.0 (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.0-RELEASE/) åßíáé ç ðëÝïí ôåëåõôáßá ôçò óåéñÜò *7-STABLE* êáé äçìéïõñãÞèçêå óôéò ÌÜéï 2009. Ç Ýêäïóç 7.2 (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/7.2-RELEASE/) åßíáé ç ðëÝïí ôåëåõôáßá ôçò óåéñÜò *6-STABLE* êáé äçìéïõñãÞèçêå óôéò Íïå 2008.

Åí óõíôïìßá, ç óåéñÜ *-STABLE* ðñïïñßæåôáé ãéá ôïõò ISPs, ôïõò åôáéñéêïýò ÷ñÞóôåò êáèþò êáé ïðïéïäÞðïôå ÷ñÞóôç èÝëåé óôáèåñüôçôá êáé ìéêñü áñéèìü áëëáãþí óå ó÷Ýóç ìå ôá íÝá (êáé ðéèáíþò áóôáèÞ) ÷áñáêôçñéóôéêÜ ðïõ åìöáíßæïíôáé óôç óåéñÜ *-CURRENT*. Åêäüóåéò ìðïñåß íá ãßíïíôáé áðü ïðïéïäÞðïôå êëÜäï, áëëÜ ç Ýêäïóç *-CURRENT* èá ðñÝðåé íá ÷ñçóéìïðïéåßôáé ìüíï áí åßóôå ðñïåôïéìáóìÝíïé íá ÷åéñéóôåßôå ôá åíäå÷ïìÝíùò ðéï áóôáèÞ ÷áñáêôçñéóôéêÜ ôçò (óå ó÷Ýóç ðÜíôá ìå ôçí áíôßóôïé÷ç *-STABLE*).

ÍÝåò åêäüóåéò åìöáíßæïíôáé êÜèå ìåñéêïýò ìÞíåò. Áí êáé ðïëëïß åðéëÝãïõí íá äéáôçñïýíôáé åíçìåñùìÝíïé êáé ðéï óõ÷íÜ ìÝóù ôïõ ðçãáßïõ êþäéêá ôïõ FreeBSD (äåßôå ôéò åñùôÞóåéò óôï FreeBSD-CURRENT êáé FreeBSD-STABLE), ïé åêäüóåéò åßíáé êÜôé ðáñáðÜíù áðü õðï÷ñÝùóç, êáèþò ï ðçãáßïò êþäéêáò åßíáé ðåñéóóüôåñï Ýíáò êéíïýìåíïò óôü÷ïò.

Ðåñéóóüôåñåò ðëçñïöïñßåò ãéá ôéò åêäüóåéò ôïõ FreeBSD ìðïñåßôå íá âñåßôå óôç Óåëßäá Ðñïåôïéìáóßáò Åêäüóåùí (http://www.FreeBSD.org/releng/index.html) óôçí äéêôõáêÞ ôïðïèåóßá ôïõ FreeBSD.

8. Ôé åßíáé ôï FreeBSD-CURRENT;

Ôï FreeBSD-CURRENT (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#CURRENT) åßíáé ç õðü åîÝëéîç Ýêäïóç ôïõ ëåéôïõñãéêïý óõóôÞìáôïò, ôï ïðïßï ìå ôïí êáéñü èá åîåëé÷èåß óôï íÝï êëÜäï FreeBSD-STABLE. Ãéá ôï ëüãï áõôü, ðáñïõóéÜæåé óõíÞèùò åíäéáöÝñïí ìüíï óå üóïõò áó÷ïëïýíôáé ìå ôçí áíÜðôõîç êþäéêá ôïõ óõóôÞìáôïò êáé óå óêëçñïðõñçíéêïýò ÷ïìðßóôåò. Äåßôå ôï ó÷åôéêü ôìÞìá (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#CURRENT) óôï åã÷åéñßäéï (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/index.html) ãéá ëåðôïìÝñåéåò ó÷åôéêÜ ìå ôï -CURRENT.

Áí äåí åßóôå åîïéêåéùìÝíïò ìå ôï ëåéôïõñãéêü óýóôçìá, Þ äåí åßóôå éêáíüò íá áíáãíùñßóåôå ôç äéáöïñÜ ìåôáîý åíüò ðñáãìáôéêïý êáé åíüò ðñïóùñéíïý ðñïâëÞìáôïò, ìÜëëïí äåí èá ðñÝðåé íá ÷ñçóéìïðïéÞóåôå ôï FreeBSD-CURRENT. Ï êëÜäïò áõôüò ïñéóìÝíåò öïñÝò åîåëßóóåôáé ðïëý ãñÞãïñá, êáé ìðïñåß áêüìá êáé ç ìåôáãëþôôéóç ôïõ íá ìçí åßíáé äõíáôÞ ãéá áñêåôÝò óõíå÷üìåíåò ìÝñåò. ¼óïé ÷ñçóéìïðïéïýí ôï FreeBSD-CURRENT áíáìÝíåôáé íá åßíáé éêáíïß íá áíáëýïõí ôá üðïéá ðñïâëÞìáôá êáé íá ôá áíáöÝñïõí ìüíï áí èåùñïýí üôé ðñüêåéôáé ãéá óçìáíôéêÜ ëÜèç êáé ü÷é ãéá "ìéêñïðñïâëÞìáôá". ÅñùôÞóåéò ôïõ ôýðïõ "ôï make world ðáñÜãåé êÜðïéá óöÜëìáôá ó÷åôéêÜ ìå ôá groups" óôç ëßóôá ôá÷õäñïìåßïõ -CURRENT, ìðïñåß íá áíôéìåôùðéóôïýí ðåñéöñïíçôéêÜ.

ÊÜèå ìÞíá, ðáñÜãïíôáé óôéãìéüôõðá (http://www.FreeBSD.org/snapshots/) åêäüóåùí ðïõ âáóßæïíôáé óôçí ôñÝ÷ïõóá êáôÜóôáóç ôùí êëÜäùí *-CURRENT* êáé *-STABLE*. Ïé óôü÷ïé ðßóù áðü êÜèå Ýêäïóç óôéãìéüôõðïõ åßíáé:

- Ï Ýëåã÷ïò ôçò ôåëåõôáßáò Ýêäïóçò ôïõ ëïãéóìéêïý åãêáôÜóôáóçò.
- Íá äþóåé ôç äõíáôüôçôá åýêïëçò åãêáôÜóôáóçò óå üóïõò åðéèõìïýí íá åãêáôáóôÞóïõí ôï *-CURRENT* Þ ôï *-STABLE* áëëÜ äåí Ý÷ïõí ôï ÷ñüíï Þ ôï åýñïò æþíçò íá ôï ðáñáêïëïõèïýí ìÝñá ìå ôç ìÝñá.
- Ç äéáôÞñçóç åíüò óôáèåñïý óçìåßïõ áíáöïñÜò ó÷åôéêÜ ìå ôïí ðçãáßï êþäéêá, óå ðåñßðôùóç ðïõ ÷áëÜóïõìå êÜôé ðïëý Üó÷çìá áñãüôåñá. (Áí êáé ëüãù ôçò ÷ñÞóçò ôïõ CVS åßíáé äýóêïëï íá óõìâåß êÜôé ðñáãìáôéêÜ ôüóï öñéêôü.)
- Íá åîáóöáëéóôåß üôé êÜèå íÝï ÷áñáêôçñéóôéêü êáé äéüñèùóç ðïõ ÷ñåéÜæåôáé Ýëåã÷ï, èá Ý÷åé ôï ìåãáëýôåñï äõíáôü êïéíü ðéèáíþí äïêéìáóôþí.

Äåí ðáñÝ÷åôáé êáìßá åããýçóç "ôåëéêÞò ðïéüôçôáò" ãéá ôá óôéãìéüôõðá ôïõ êëÜäïõ *-CURRENT*. Áí ÷ñåéÜæåóôå Ýíá óôáèåñü êáé äïêéìáóìÝíï óýóôçìá, èá ðñÝðåé íá ðñïôéìÜôå ôéò åðßóçìåò ðëÞñåéò åêäüóåéò Þ ôá óôéãìéüôõðá ôïõ êëÜäïõ *-STABLE*.

Ðåñéóóüôåñåò ðëçñïöïñßåò ãéá ôá óôéãìéüôõðá ìðïñåßôå íá âñåßôå óôç óåëßäá óôéãìéüôõðùí åêäüóåùí (http://www.FreeBSD.org/snapshots/).

Åðßóçìá óôéãìéüôõðá åêäüóåùí ðáñÜãïíôáé ìéá öïñÜ ôï ìÞíá ãéá üëïõò ôïõò åíåñãïýò êëÜäïõò áíÜðôõîçò. ÇìåñÞóéá óôéãìéüôõðá ãéá ôéò ðéï äçìïöéëåßò áñ÷éôåêôïíéêÝò (i386 êáé amd64) ìðïñåßôå íá âñåßôå óôç óåëßäá http://snapshots.us.freebsd.org/.

9. Ôé åßíáé ôï FreeBSD-STABLE;

¼ôáí êõêëïöüñçóå ôï FreeBSD 2.0.5, ç áíÜðôõîç ôïõ FreeBSD ÷ùñßóôçêå óå äýï êëÜäïõò. Ï Ýíáò êëÜäïò ïíïìÜóôçêå -STABLE (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#STABLE), êáé ï Üëëïò -CURRENT (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#CURRENT). Ôï *FreeBSD-STABLE* áðåõèýíåôáé óå Ðáñï÷åßò Õðçñåóéþí Internet (ISPs) êáé Üëëåò åìðïñéêÝò ÷ñÞóåéò, üðïõ ïé áðüôïìåò áëëáãÝò êáé ôá ôõ÷üí ðåéñáìáôéêÜ ÷áñáêôçñéóôéêÜ åßíáé ãåíéêÜ áíåðéèýìçôá. Óôïí êëÜäï áõôü åíóùìáôþíïíôáé ìüíï êáëÜ äïêéìáóìÝíåò äéïñèþóåéò êáé Üëëåò ìéêñÝò áëëáãÝò. Áðü ôçí Üëëç ìåñéÜ, ôï *FreeBSD-CURRENT* âñßóêåôáé óå ìéá ìïíáäéêÞ áäéÜêïðç ãñáììÞ áíÜðôõîçò áðü ôçí åðï÷Þ ôçò êõêëïöïñßáò ôçò Ýêäïóçò 2.0, ïäçãþíôáò ðñïò ôçí Ýêäïóç 8.0-RELEASE áëëÜ êáé ìåôÜ áðü áõôÞ. Ãéá ðåñéóóüôåñåò ðëçñïöïñßåò äåßôå ôï Üñèñï "Ðñïåôïéìáóßá Åêäüóåùí ôïõ FreeBSD: Äçìéïõñãßá ôïõ ÊëÜäïõ ¸êäïóçò (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/releng/release-proc.html#REL-BRANCH)".

Ï êëÜäïò 2.2-STABLE åãêáôáëåßöèçêå ìå ôçí êõêëïöïñßá ôçò Ýêäïóçò 2.2.8. Ï êëÜäïò 3-STABLE ôåëåßùóå ìå ôçí êõêëïöïñßá ôçò Ýêäïóçò 3.5.1, ðïõ Þôáí êáé ç ôåëåõôáßá ôçò óåéñÜò åêäüóåùí 3.*x*. Ï êëÜäïò 4-STABLE ôåëåßùóå ìå ôçí êõêëïöïñßá ôçò Ýêäïóçò 4.11, ôçò ôåëåõôáßáò ôçò óåéñÜò 4.*x*. Ïé ìüíåò áëëáãÝò ðïõ ãßíïíôáé êáôÜ âÜóç óå êÜèå Ýíá áðü áõôïýò ôïõò êëÜäïõò, Ý÷ïõí ó÷Ýóç ìå äéïñèþóåéò óå êåíÜ áóöáëåßáò. Ç õðïóôÞñéîç ôïõ êëÜäïõ 5-STABLE óôáìÜôçóå ìå ôçí Ýêäïóç 5.5, ôçí ôåëåõôáßá Ýêäïóç ôçò óåéñÜò 5.*X*. Ç õðïóôÞñéîç ãéá ôç óåéñÜ åêäüóåùí 6-STABLE èá óõíå÷éóôåß ãéá ëßãï êáéñü áêüìá, áëëÜ èá åðéêåíôñùèåß ðåñéóóüôåñï óå äéïñèþóåéò êåíþí áóöáëåßáò êáé Üëëùí óïâáñþí ðñïâëçìÜôùí.

Ï êëÜäïò 8.0-STABLE åßíáé ï -*STABLE* êëÜäïò ðïõ âñßóêåôáé õðü åíåñãÞ áíÜðôõîç. Ç ôåëåõôáßá êõêëïöïñßá ðïõ âáóßæåôáé óôïí êëÜäï 8.0-STABLE åßíáé ç 8.0-RELEASE, êáé Ýãéíå óôéò ÌÜéï 2009.

Ï êëÜäïò 8-CURRENT åßíáé ï êëÜäïò -*CURRENT* ðïõ áíáðôýóóåôáé áõôÞ ôç óôéãìÞ åíåñãÜ þóôå íá äçìéïõñãçèåß ç íÝá ãåíéÜ ôïõ FreeBSD. Äåßôå ôï ôìÞìá Ôé åßíáé ôï FreeBSD-CURRENT; ãéá ðåñéóóüôåñåò ðëçñïöïñßåò ó÷åôéêÜ ìå áõôü ôïí êëÜäï.

10. ÊÜèå ðüôå ãßíïíôáé åðßóçìåò êõêëïöïñßåò ôïõ FreeBSD;

H ÏìÜäá ÏñãÜíùóçò ôùí Åêäüóåùí <`re@FreeBSD.org`> äßíåé óôçí êõêëïöïñßá ìéá êáéíïýñéá Ýêäïóç ôïõ FreeBSD êÜèå ðåñßðïõ ôÝóóåñéò ìÞíåò, êáôÜ ìÝóï üñï. Ïé çìåñïìçíßåò êõêëïöïñßáò áíáêïéíþíïíôáé áñêåôÜ íùñßò, þóôå üóïé äïõëåýïõí ðÜíù óôï óýóôçìá íá îÝñïõí ðüôå ïé åñãáóßåò ôïõò ðñÝðåé íá Ý÷ïõí ïëïêëçñùèåß êáé äïêéìáóôåß. Ðñéí áðü êÜèå êõêëïöïñßá, ðñïçãåßôáé ìéá ðåñßïäïò äïêéìþí, þóôå íá åîáóöáëéóôåß üôé ç ðñïóèÞêç íÝùí ÷áñáêôçñéóôéêþí äåí Ý÷åé áñíçôéêÝò åðéðôþóåéò óôçí óôáèåñüôçôá ôçò Ýêäïóçò. Ðïëëïß ÷ñÞóôåò èåùñïýí áõôü ôï åßäïò ôçò ðñïóï÷Þò óáí Ýíá áðü ôá êáëýôåñá ðñÜãìáôá ó÷åôéêÜ ìå ôï FreeBSD, áí êáé ç áíáìïíÞ ùò üôïõ öôÜóïõí üëá ôá ùñáßá íÝá ÷áñáêôçñéóôéêÜ óôï -STABLE ìðïñåß íá åßíáé ëßãï åêíåõñéóôéêÞ.

Ðåñéóóüôåñåò ðëçñïöïñßåò ó÷åôéêÜ ìå ôç äéáäéêáóßá êõêëïöïñßáò (ðåñéëáìâáíïìÝíïõ êáé åíüò ÷ñïíïäéáãñÜììáôïò åðéêåßìåíùí êõêëïöïñéþí) ìðïñïýí íá âñåèïýí óôéò óåëßäåò ðñïåôïéìáóßáò åêäüóåùí (http://www.FreeBSD.org/releng/index.html) óôç äéêôõáêÞ ôïðïèåóßá ôïõ FreeBSD.

Ãéá ôïõò ðéï áíõðüìïíïõò, õðÜñ÷ïõí êáèçìåñéíÜ óôéãìéüôõðá (binary snapshots) üðùò áíáöÝñèçêå ðéï ðÜíù.

11. Ðïéïò åßíáé õðåýèõíïò ãéá ôï FreeBSD

Ïé âáóéêÝò áðïöÜóåéò ðïõ áöïñïýí ôï FreeBSD Project, üðùò ç óõíïëéêÞ êáôåýèõíóç ôïõ Ýñãïõ êáé ðïéïò åðéôñÝðåôáé íá ðñïóèÝóåé ðçãáßï êþäéêá óôï äÝíôñï, ðáßñíïíôáé áðü ôçí âáóéêÞ ïìÜäá (core team) (http://www.FreeBSD.org/administration.html#t-core) ðïõ áðïôåëåßôáé áðü 9 Üôïìá. ÕðÜñ÷åé ìéá áñêåôÜ ìåãáëýôåñç ïìÜäá ìå ðåñéóóüôåñá áðü 350 Üôïìá ðïõ ïíïìÜæïíôáé äéáðñÜêôåò (committers) (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/contributors/article.html#STAFF-COMMITTERS) êáé ïé ïðïßïé Ý÷ïõí ôçí Ýãêñéóç íá êÜíïõí áðåõèåßáò áëëáãÝò óôïí ðçãáßï êþäéêá ôïõ FreeBSD.

Ùóôüóï ïé ðåñéóóüôåñåò óçìáíôéêÝò áëëáãÝò óõæçôïýíôáé áðü ðñéí óôéò ëßóôåò çëåêôñïíéêïý ôá÷õäñïìåßïõ, êáé äåí õðÜñ÷ïõí ðåñéïñéóìïß ó÷åôéêÜ ìå ôï ðïéïé ëáìâÜíïõí ìÝñïò óå áõôÝò ôéò óõæçôÞóåéò.

12. Ðïõ ìðïñþ íá ðñïìçèåõôþ ôï FreeBSD;

ÊÜèå óçìáíôéêÞ Ýêäïóç ôïõ FreeBSD åßíáé äéáèÝóéìç ìÝóù áíþíõìïõ FTP áðü ôïí äéáêïìéóôÞ FTP ôïõ FreeBSD (ftp://ftp.FreeBSD.org/pub/FreeBSD/):

- Ç ôåëåõôáßá êõêëïöïñßá ðïõ âáóßæåôáé óôï 6-STABLE, 8.0-RELEASE ìðïñåß íá âñåèåß óôïí êáôÜëïãï 8.0-RELEASE (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.0-RELEASE/).

- Åêäüóåéò ðïõ âáóßæïíôáé óåÓôéãìéüôõðá (http://www.FreeBSD.org/snapshots/) ãßíïíôáé êáèçìåñéíÜ áðü ôïõò êëÜäïõò -CURRENT êáé -STABLE, êáé åîõðçñåôïýí êáôÜ âÜóç üóïõò áó÷ïëïýíôáé ìå ôçí áíÜðôõîç êáé ôïí Ýëåã÷ï ôùí ôåëåõôáßáò ãåíéÜò ðñïãñáììÜôùí.
- Ç ôåëåõôáßá êõêëïöïñßá ðïõ âáóßæåôáé óôïí êëÜäï 5-STABLE, ç 7.2-RELEASE, ìðïñåß íá âñåèåß óôïí êáôÜëïãï 7.2-RELEASE (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/7.2-RELEASE/).

Ðëçñïöïñßåò ó÷åôéêÜ ìå ôç äéÜèåóç ôïõ FreeBSD óå CD, DVD êáé Üëëá ìÝóá ìðïñïýí íá âñåèïýí óôï Åã÷åéñßäéï (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/mirrors.html).

13. Ðùò ìðïñþ íá Ý÷ù ðñüóâáóç óôç ÂÜóç ÄåäïìÝíùí ìå ôéò ÁíáöïñÝò ÐñïâëçìÜôùí;

Ìðïñåßôå íá êÜíåôå áíáæçôÞóåéò óôç âÜóç äåäïìÝíùí ìå ôéò áíáöïñÝò ðñïâëçìÜôùí ìÝóù ôçò äéåðáöÞò áíáæçôÞóåùí óôï Web (http://www.FreeBSD.org/cgi/query-pr-summary.cgi?query).

Ç åíôïëÞ send-pr(1) ìðïñåß íá ÷ñçóéìïðïéçèåß ãéá ôçí õðïâïëÞ áíáöïñþí ðñïâëçìÜôùí, êáèþò êáé áéôÞóåùí ãéá áëëáãÝò, ìÝóù çëåêôñïíéêïý ôá÷õäñïìåßïõ. ÅíáëëáêôéêÜ, ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå ôçí áíôßóôïé÷ç äõíáôüôçôá õðïâïëÞò áíáöïñþí ðñïâëçìÜôùí ìÝóù ôçò äéåðáöÞò web (http://www.freebsd.org/send-pr.html) ìå ôç âïÞèåéá åíüò ðñïãñÜììáôïò öõëëïìåôñçôÞ.

Ðñéí õðïâÜëåôå ìéá áíáöïñÜ ðñïâëÞìáôïò, ðáñáêáëïýìå äéáâÜóôå ôï ÃñÜöïíôáò ÁíáöïñÝò ÐñïâëçìÜôùí ãéá ôï FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/problem-reports/article.html), Ýíá Üñèñï ãéá ôï ðùò íá ãñÜöåôå êáëÝò áíáöïñÝò ðñïâëçìÜôùí.

14. Ôé Üëëåò ðçãÝò ðëçñïöïñéþí õðÜñ÷ïõí;

Ðáñáêáëïýìå äåßôå ôç ëßóôá ôçò Ôåêìçñßùóçò (http://www.FreeBSD.org/docs.html) óôçí êýñéá äéêôõáêÞ ôïðïèåóßá ôïõ FreeBSD (http://www.FreeBSD.org).

ÊåöÜëáéï 2 Ôåêìçñßùóç êáé ÕðïóôÞñéîç

1. Ðïéá êáëÜ âéâëßá õðÜñ÷ïõí ó÷åôéêÜ ìå ôï FreeBSD;

Ôï Project ðáñÜãåé ìéá ìåãÜëç ãêÜìá ôåêìçñßùóçò ðïõ äéáôßèåôáé online áðü ôïí ðáñáêÜôù óýíäåóìï: http://www.FreeBSD.org/docs.html. Ôá ßäéá áõôÜ Ýããñáöá åßíáé äéáèÝóéìá êáé ùò ðáêÝôá ôá ïðïßá ìðïñåßôå íá åãêáôáóôÞóåôå åýêïëá óôï FreeBSD óýóôçìá óáò. Ðåñéóóüôåñåò ëåðôïìÝñåéåò ó÷åôéêÜ ìå ôá ðáêÝôá ôåêìçñßùóçò, èá âñåßôå óôéò áêüëïõèåò ðáñáãñÜöïõò.

Åðéðñüóèåôá, èá âñåßôå êáé Üëëá óõíéóôþìåíá âéâëßá óôçí Âéâëéïãñáößá óôï ôÝëïò áõôïý ôïõ FAQ, êáé ôïõ Åã÷åéñéäßïõ.

2. Åßíáé äéáèÝóéìç ç ôåêìçñßùóç êáé óå Üëëåò ìïñöÝò, üðùò áðëü êåßìåíï (ASCII) Þ PostScript;

Íáé. Ç ôåêìçñßùóç åßíáé äéáèÝóéìç óå ðëÞèïò äéáöïñåôéêþí ìïñöþí êáé ôñüðùí óõìðßåóçò, óôï äéáêïìéóôÞ FTP ôïõ FreeBSD, óôïí êáôÜëïãï /pub/FreeBSD/doc/ (ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/).

Ç ôåêìçñßùóç åßíáé êáôçãïñéïðïéçìÝíç ìå äéÜöïñïõò ôñüðïõò. Áõôïß ðåñéëáìâÜíïõí:

- Ôï üíïìá ôïõ åããñÜöïõ, üðùò ð.÷. `faq`, Þ `handbook`.
- H ãëþóóá êáé ç êùäéêïðïßçóç ôïõ êåéìÝíïõ. ÁõôÜ âáóßæïíôáé óôá ïíüìáôá ðïõ Ý÷ïõí äïèåß óôéò ôïðéêÝò ñõèìßóåéò êáé ðïõ ìðïñåßôå íá âñåßôå óôïí êáôÜëïãï `/usr/share/locale` óôï FreeBSD óýóôçìÜ óáò. Ïé ôñÝ÷ïõóåò ãëþóóåò êáé êùäéêïðïéÞóåéò ðïõ Ý÷ïõìå äéáèÝóéìåò áõôÞ ôç óôéãìÞ óôçí ôåêìçñßùóç åßíáé ïé ðáñáêÜôù:

¼íïìá	Åñìçíåßá
`en_US.ISO8859-1`	ÁããëéêÜ ÇÐÁ
`bn_BD.ISO10646-1`	Bengali (Þ Bangla)
`da_DK.ISO8859-1`	ÄáíÝæéêá
`de_DE.ISO8859-1`	ÃåñìáíéêÜ
`es_ES.ISO8859-1`	ÉóðáíéêÜ
`fr_FR.ISO8859-1`	ÃáëëéêÜ
`hu_HU.ISO8859-2`	ÏõããáñÝæéêá
`it_IT.ISO8859-15`	ÉôáëéêÜ
`ja_JP.eucJP`	ÃéáðùíÝæéêá (êùäéêïðïßçóç EUC)
`mn_MN.UTF-8`	ÌïããïëéêÜ (êùäéêïðïßçóç UTF-8)
`nl_NL.ISO8859-1`	ÏëëáíäéêÜ
`pl_PL.ISO8859-2`	ÐïëùíéêÜ
`pt_BR.ISO8859-1`	ÐïñôïãáëéêÜ (Âñáæéëßá)
`ru_RU.KOI8-R`	Ñþóéêá (êùäéêïðïßçóç KOI8-R)
`sr_YU.ISO8859-2`	ÓÝñâéêá
`tr_TR.ISO8859-9`	Ôïýñêéêá
`zh_CN.GB2312`	ÁðëïðïéçìÝíá ÊéíÝæéêá (êùäéêïðïßçóç GB2312)
`zh_TW.Big5`	ÐáñáäïóéáêÜ ÊéíÝæéêá (êùäéêïðïßçóç Big5)

Óçìåßùóç: ÊÜðïéá Ýãããñáöá ìðïñåß íá ìçí åßíáé äéáèÝóéìá óå üëåò ôéò ãëþóóåò.

- Ôç ìïñöÞ ôïõ åããñÜöïõ. ÐáñÜãïõìå ôåêìçñßùóç óå ðëÞèïò äéáöïñåôéêþí ìïñöþí åîüäïõ. ÊÜèå ìïñöÞ Ý÷åé ôá äéêÜ ôçò ðëåïíåêôÞìáôá êáé ìåéïíåêôÞìáôá. ÊÜðïéåò ìïñöÝò åßíáé ðåñéóóüôåñï êáôÜëëçëåò ãéá áíÜãíùóç online, åíþ Üëëåò äßíïõí ðéï êáëáßóèçôï áðïôÝëåóìá üôáí åêôõðùèïýí. Ç äéÜèåóç ôçò ôåêìçñßùóçò óå üëåò áõôÝò ôéò ìïñöÝò åîáóöáëßæåé üôé ïé áíáãíþóôåò ìáò èá ìðïñïýí íá äéáâÜóïõí ôá ôìÞìáôá ðïõ ôïõò åíäéáöÝñïõí, åßôå óôçí ïèüíç ôïõò, åßôå áöïý ôá åêôõðþóïõí. Ïé äéáèÝóéìåò áõôÞ ôç óôéãìÞ ìïñöÝò åßíáé:

ÌïñöÞ	Åñìçíåßá
`html-split`	ÓõëëïãÞ ìéêñþí, óõíäåìÝíùí ìåôáîý ôïõò, áñ÷åßùí HTML.
`html`	¸íá ìåãÜëï HTML áñ÷åßï ðïõ ðåñéÝ÷åé ïëüêëçñï ôï Ýããñáöï.
`pdb`	ÌïñöÞ âÜóçò äåäïìÝíùí ãéá ôï Palm Pilot, ãéá ÷ñÞóç ìå ôï ðñüãñáììá áíÜãíùóçò iSilo (http://www.iSilo.com/).
`pdf`	ÌïñöÞ êåéìÝíïõ PDF ôçò Adobe.
`ps`	PostScript
`rtf`	ÌïñöÞ ÅìðëïõôéóìÝíïõ ÊåéìÝíïõ ôçò Microsoft[a]
`txt`	Áðëü êåßìåíï

Ðßíáêáò Óçìåéþóåùí:

[a] Ïé áñéèìïß óåëßäùí äåí áíáíåþíïíôáé áõôüìáôá üôáí öïñôþíåôå áõôÞ ôç ìïñöÞ åããñÜöïõ óôï Word. ÐéÝóôå **CTRL+A**, **CTRL+END**, **F9** ìåôÜ ôç öüñôùóç ôïõ êåéìÝíïõ ãéá ôçí áíáíÝùóç ôùí áñéèìþí óåëßäùí.

- Ï ôñüðïò óõìðßåóçò êáé ðáêåôáñßóìáôïò. ÕðÜñ÷ïõí ôñåéò ôñüðïé ðïõ ÷ñçóéìïðïéïýíôáé áõôÞ ôç óôéãìÞ.

  1. ¼ôáí ç ìïñöÞ åßíáé `html-split`, ôá áñ÷åßá óõìðéÝæïíôáé ìå ÷ñÞóç ôçò tar(1). Ôï áñ÷åßï `.tar` ðïõ ðñïêýðôåé, óõìðéÝæåôáé Ýðåéôá ìå ôïõò ôñüðïõò óõìðßåóçò ðïõ ðåñéãñÜöïíôáé ðáñáêÜôù.
  2. ¼ëåò ïé Üëëåò ìïñöÝò äçìéïõñãïýí Ýíá áñ÷åßï ðïõ ïíïìÜæåôáé `book.`*`ìïñöÞ`* (ð.÷., `book.pdb`, `book.html`, ê.ï.ê.).

  Ôá áñ÷åßá áõôÜ óõìðéÝæïíôáé êáôüðéí ìå äýï ôñüðïõò óõìðßåóçò.

Ôñüðïò	ÐåñéãñáöÞ
`zip`	ÌïñöÞ óõìðßåóçò Zip. Áí èÝëåôå íá ôï áðïóõìðéÝóåôå óôï FreeBSD èá ðñÝðåé íá åãêáôáóôÞóåôå ðñþôá ôï port `archivers/unzip`.
`bz2`	Ç ìïñöÞ BZip2. Åßíáé ëéãüôåñï äéáäåäïìÝíç áðü ôï Zip, áëëÜ ãåíéêÜ äçìéïõñãåß ìéêñüôåñá áñ÷åßá. ÅãêáôáóôÞóôå ôï port `archivers/bzip2` ãéá íá áðïóõìðéÝóåôå áñ÷åßá áõôïý ôïõ ôýðïõ.

  Ìå ôïí ôñüðï áõôü, ç ìïñöÞ PostScript ôïõ Åã÷åéñéäßïõ, óõìðéåóìÝíç ìå ÷ñÞóç ôïõ BZip2 èá áðïèçêåõôåß óå Ýíá áñ÷åßï ìå üíïìá `book.ps.bz2` óôïí êáôÜëïãï `handbook/`.

Áöïý åðéëÝîåôå ôç ìïñöÞ êáé ôï ìç÷áíéóìü óõìðßåóçò ðïõ åðéèõìåßôå íá êáôåâÜóåôå, èá ðñÝðåé Ýðåéôá íá áðïöáóßóåôå áí èÝëåôå Þ ü÷é íá êáôåâÜóåôå ôï Ýããñáöï ìå ôç ìïñöÞ *ðáêÝôïõ* ôïõ FreeBSD.

Ôï ðëåïíÝêôçìá óôï íá êáôåâÜóåôå êáé íá åãêáôáóôÞóåôå ôï ðáêÝôï åßíáé üôé Ýðåéôá ìðïñåßôå íá äéá÷åéñéóôåßôå ôçí ôåêìçñßùóç ÷ñçóéìïðïéþíôáò ôá óõíçèéóìÝíá åñãáëåßá äéá÷åßñéóçò ðáêÝôùí ôïõ FreeBSD üðùò ôçí pkg_add(1) êáé ôçí pkg_delete(1).

Áí áðïöáóßóåôå íá êáôåâÜóåôå êáé íá åãêáôáóôÞóåôå ôçí ôåêìçñßùóç ùò ðáêÝôï, èá ðñÝðåé íá îÝñåôå ôï áêñéâÝò üíïìá áñ÷åßïõ ðïõ èá êáôåâÜóåôå. Ôá áñ÷åßá ôåêìçñßùóçò-ùò-ðáêÝôï áðïèçêåýïíôáé óå Ýíá êáôÜëïãï ìå ôï üíïìá `packages`. ÊÜèå ðáêÝôï ìïéÜæåé ìå *`üíïìá-êåéìÝíïõ.ãëþóóá.êùäéêïðïßçóç.ìïñöÞ`*`.tgz`.

Ãéá ðáñÜäåéãìá, ôï FAQ, óôá ÁããëéêÜ, óå ìïñöÞ PDF, åßíáé óôï ðáêÝôï ìå üíïìá `faq.en_US.ISO8859-1.pdf.tgz`.

Áí ôï îÝñåôå áõôü ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå ôçí áêüëïõèç åíôïëÞ ãéá íá åãêáôáóôÞóåôå ôï ðáêÝôï ôïõ Áããëéêïý PDF FAQ:

```
# pkg_add ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/packages/faq.en_US.ISO8859-1.pdf.tgz
```

Áöïý ôï êÜíåôå áõôü, ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå ôçí åíôïëÞ pkg_info(1) ãéá íá âñåßôå ðïõ Ý÷åé åãêáôáóôáèåß ôï áñ÷åßï.

```
# pkg_info -f faq.en_US.ISO8859-1.pdf
Information for faq.en_US.ISO8859-1.pdf:

Packing list:
 Package name: faq.en_US.ISO8859-1.pdf
 CWD to /usr/share/doc/en_US.ISO8859-1/books/faq
File: book.pdf
 CWD to .
File: +COMMENT (ignored)
File: +DESC (ignored)
```

¼ðùò ìðïñåßôå íá äåßôå, ôï `book.pdf` èá Ý÷åé åãêáôáóôáèåß óôïí êáôÜëïãï `/usr/share/doc/en_US.ISO8859-1/books/faq`.

Áí äåí èÝëåôå íá ÷ñçóéìïðïéÞóåôå ôá ðáêÝôá, èá ðñÝðåé íá êáôåâÜóåôå ìüíï óáò ôá óõìðéåóìÝíá áñ÷åßá, íá ôá áðïóõìðéÝóåôå êáé Ýðåéôá íá áíôéãñÜøåôå ôá áíôßóôïé÷á Ýããñáöá óôç èÝóç ôïõò.

Ãéá ðáñÜäåéãìá, ç Ýêäïóç ôïõ FAQ óå óõíäåäåìÝíá áñ÷åßá HTML, óõìðéåóìÝíç ìå ÷ñÞóç ôïõ bzip2(1), ìðïñåß íá âñåèåß óôï áñ÷åßï `doc/en_US.ISO8859-1/books/faq/book.html-split.tar.bz2` . Ãéá íá êáôåâÜóåôå êáé íá áðïóõìðéÝóåôå áõôü ôï áñ÷åßï èá ðñÝðåé íá êÜíåôå ôï ðáñáêÜôù:

```
# 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
```

Èá êáôáëÞîåôå ìå ìéá óõëëïãÞ áðü áñ÷åßá `.html`. Ôï âáóéêü ïíïìÜæåôáé `index.html`, êáé èá ðåñéÝ÷åé ôïí ðßíáêá ðåñéå÷ïìÝíùí, åéóáãùãéêü õëéêü, êáé äåóìïýò ðñïò ôá Üëëá ôìÞìáôá ôïõ åããñÜöïõ. Ìðïñåßôå Ýðåéôá íá áíôéãñÜøåôå Þ íá ìåôáêéíÞóåôå ôá áñ÷åßá áõôÜ óôéò ôåëéêÝò ôïõò èÝóåéò, üðùò áðáéôåßôáé.

3. Ðïõ ìðïñþ íá âñù ðëçñïöïñßåò ãéá ëßóôåò çëåêôñïíéêïý ôá÷õäñïìåßïõ ôïõ FreeBSD;

Ìðïñåßôå íá âñåßôå ðëÞñåéò ðëçñïöïñßåò óôçí áíôßóôïé÷ç êáôá÷þñçóç ôïõ Åã÷åéñéäßïõ ãéá ôéò ëßóôåò çëåêôñïíéêïý ôá÷õäñïìåßïõ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/eresources.html#ERESOURCES-MAIL).

4. Ôé ïìÜäåò óõæÞôçóçò åßíáé äéáèÝóéìåò ãéá ôï FreeBSD;

Ìðïñåßôå íá âñåßôå ðëÞñåéò ðëçñïöïñßåò óôçí êáôá÷þñçóç ôïõ Åã÷åéñéäßïõ ãéá ôéò ïìÜäåò óõæçôÞóåùí (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/eresources-news.html).

5. ÕðÜñ÷ïõí êáíÜëéá óôï IRC (Internet Relay Chat) ãéá ôï FreeBSD;

Íáé, ôá ðåñéóóüôåñá IRC äßêôõá äéáèÝôïõí êáíÜëé óõæÞôçóçò ãéá ôï FreeBSD:

- Ôï êáíÜëé `#FreeBSD` óôï EFNet (http://www.efnet.org/index.php) áó÷ïëåßôáé ìå ôï FreeBSD, áëëÜ ìçí ðÜôå åêåß ãéá ôå÷íéêÞ õðïóôÞñéîç Þ ãéá íá ðñïóðáèÞóåôå íá ðåßóåôå ôïõò èáìþíåò íá óáò âïçèÞóïõí ãéá íá áðáëëáãåßôå áðü ôïí êüðï íá äéáâÜóåôå ôéò óåëßäåò âïÞèåéáò Þ íá åñåõíÞóåôå ìüíïé óáò. Åßíáé áðëþò Ýíá êáíÜëé óõæÞôçóçò, ðñþôá áðü üëá, êáé ôá èÝìáôá åêåß ìðïñåß íá êõìáßíïíôáé áíÜìåóá óôï óåî, ôïí áèëçôéóìü, ôá ðõñçíéêÜ üðëá ùò êáé ôï FreeBSD. Óáò Ý÷ïõìå ðñïåéäïðïéÞóåé! Åßíáé äéáèÝóéìï óôïí åîõðçñåôçôÞ `irc.chat.org`.
- Ôï êáíÜëé `#FreeBSDhelp` óôï EFNet (http://www.efnet.org/index.php) åßíáé áöéåñùìÝíï óôç âïÞèåéá ôùí ÷ñçóôþí ôïõ FreeBSD. ¸÷åôå ðïëý ìåãáëýôåñç ðéèáíüôçôá íá âñåßôå áðáíôÞóåéò óôéò åñùôÞóåéò óáò óå áõôü, ðáñÜ óôï `#FreeBSD`.
- Ôï êáíÜëé `##FreeBSD` óôï Freenode (http://freenode.net/) åßíáé Ýíá ãåíéêü êáíÜëé âïÞèåéáò ìå ðåñßðïõ 300 ÷ñÞóôåò êÜèå öïñÜ. Ïé óõæçôÞóåéò ìåñéêÝò öïñÝò îåöåýãïõí áðü ôï èÝìá, áëëÜ äßíåôáé ðñïôåñáéüôçôá óôïõò ÷ñÞóôåò ìå åñùôÞóåéò ãéá ôï FreeBSD. Åßìáóôå êáëïß óôï íá óáò âïçèÞóïõìå íá êáôáëÜâåôå ôá âáóéêÜ, íá óáò äåßîïõìå ðïõ íá áíáôñÝîåôå óôï Åã÷åéñßäéï üðïôå ÷ñåéÜæåôáé, êáé íá óáò êáôåõèýíïõìå åêåß üðïõ ìðïñåßôå íá ìÜèåôå ðåñéóóüôåñá ãéá èÝìá ðïõ óáò áðáó÷ïëåß. Åßìáóôå êáôÜ âÜóç Áããëüöùíï êáíÜëé, áí êáé Ý÷ïõìå ÷ñÞóôåò áðü üëï ôïí êüóìï. Áí èÝëåôå íá ñùôÞóåôå óôç ìçôñéêÞ óáò ãëþóóá, ðñïóðáèÞóôå íá êÜíåôå ôçí åñþôçóç óôá ÁããëéêÜ êáé Ýðåéôá ìåôáêéíçèåßôå óå êÜðïéï Üëëï êáíÜëé ##freebsd-*lang* áí ÷ñåéÜæåôáé.
- Ôï êáíÜëé `#FreeBSD` óôï DALNET (http://www.dal.net/) åßíáé äéáèÝóéìï óôï `irc.dal.net` óôéò ÇÐÁ êáé óôï `irc.eu.dal.net` óôçí Åõñþðç.
- Ôï êáíÜëé `#FreeBSDHelp` óôï DALNET (http://www.dal.net/) åßíáé äéáèÝóéìï óôï `irc.dal.net` óôéò ÇÐÁ êáé óôï `irc.eu.dal.net` óôçí Åõñþðç.
- Ôï êáíÜëé `#FreeBSD` óôï UNDERNET (http://www.undernet.org/) åßíáé äéáèÝóéìï óôï `us.undernet.org` óôéò ÇÐÁ êáé óôï `eu.undernet.org` óôçí Åõñþðç. Êáèþò åßíáé êáíÜëé âïÞèåéáò, èá ðñÝðåé íá åßóôå ðñïåôïéìáóìÝíïé íá äéáâÜóåôå ôá Ýããñáöá óôá ïðïßá èá óáò ðáñáðÝìøïõí.
- Ôï êáíÜëé `#FreeBSD` óôï RUSNET (http://www.rusnet.org.ru/) åßíáé ðñïóáíáôïëéóìÝíï óôç âïÞèåéá ÷ñçóôþí ôïõ FreeBSD ðïõ ìéëïýí ôç ÑùóéêÞ ãëþóóá. Åßíáé åðßóçò Ýíá êáëü ìÝñïò ãéá ìç-ôå÷íéêÝò óõæçôÞóåéò.
- Ôï êáíÜëé `#BSDChat` óôï IRCNET (http://www.ircnet.net/) åßíáé Ýíá êáíÜëé óå ðáñáäïóéáêÞ ÊéíåæéêÞ ãëþóóá, áöéåñùìÝíï óôç âïÞèåéá ÷ñçóôþí ôïõ FreeBSD. Åßíáé åðßóçò Ýíá êáëü ìÝñïò ãéá ìç-ôå÷íéêÝò óõæçôÞóåéò.

ÊÜèå Ýíá áðü áõôÜ ôá êáíÜëéá, åßíáé ÷ùñéóôü, êáé äåí äéáóõíäÝïíôáé ìå ôá õðüëïéðá. Ôá óôõë ôçò óõæÞôçóçò äéáöÝñïõí, êáé ßóùò èá èÝëåôå íá äïêéìÜóåôå êáèÝíá ãéá íá âñåßôå êÜðïéï ðïõ íá ôáéñéÜæåé óôï äéêü óáò. ¼ðùò ìå *êÜèå* ôýðï óõæÞôçóçò óôï IRC, áí åßóôå áðü áõôïýò ðïõ ðñïóâÜëëïíôáé åýêïëá, Þ äåí ìðïñåßôå íá áíôéìåôùðßóåôå ðïëëïýò íÝïõò áíèñþðïõò (êáé áñêåôïýò ðéï ìåãÜëïõò) íá ðåéñÜæïíôáé óôï ëåêôéêü áíôßóôïé÷ï ôçò ðõãìá÷ßáò, ìçí áó÷ïëçèåßôå êáèüëïõ.

6. Ðïõ ìðïñþ íá âñù åðß ðëçñùìÞ õðïóôÞñéîç êáé åêðáßäåõóç ãéá ôï FreeBSD;

Ôï DaemonNews ðáñÝ÷åé åðß ðëçñùìÞ åêðáßäåõóç êáé õðïóôÞñéîç ãéá ôï FreeBSD. Ðåñéóóüôåñåò ðëçñïöïñßåò ìðïñåßôå íá âñåßôå óôçí äéêôõáêÞ ôïõò ôïðïèåóßá BSD Mall (http://www.bsdmall.com/).

Ôï FreeBSD Mall ðáñÝ÷åé åðßóçò åðß ðëçñùìÞ õðïóôÞñéîç ãéá ôï FreeBSD. Ìðïñåßôå íá âñåßôå ðåñéóóüôåñåò ðëçñïöïñßåò óôçí äéêôõáêÞ ôïõò ôïðïèåóßá (http://www.freebsdmall.com/).

ÏðïéïóäÞðïôå Üëëïò ïñãáíéóìüò ðáñÝ÷åé åêðáßäåõóç êáé õðïóôÞñéîç, èá ðñÝðåé íá åðéêïéíùíÞóåé ìå ôï Project ãéá íá êáôá÷ùñçèåß óå áõôÞ ôç ëßóôá.

ÊåöÜëáéï 3 ÅãêáôÜóôáóç

1. Ðïéï áñ÷åßï ðñÝðåé íá êáôåâÜóù ãéá íá ðÜñù ôï FreeBSD;

×ñåéÜæåóôå ôñßá images äéóêåôþí: `floppies/boot.flp`, `floppies/kern1.flp`, êáé `floppies/kern2.flp`. Ôá images áõôÜ ðñÝðåé íá íá ôá ãñÜøåôå óå äéóêÝôåò ÷ñçóéìïðïéþíôáò êÜðïéï åñãáëåßï üðùò ôï `fdimage` Þ ôï dd(1).

Áí ðñÝðåé íá êáôåâÜóåôå ìüíïé óáò ôá óåô äéáíïìþí (distributions — ãéá ðáñÜäåéãìá ãéá ìéá åãêáôÜóôáóç ìÝóù óõóôÞìáôïò áñ÷åßùí DOS), èá âñåßôå ðáñáêÜôù ìåñéêÝò óõóôÜóåéò ó÷åôéêÜ ìå ôï ðïéáò äéáíïìÝò íá êáôåâÜóåôå:

- base/
- manpages/
- compat*/
- doc/
- src/ssys.*

ÐëÞñåéò ïäçãßåò ó÷åôéêÜ ìå ôç äéáäéêáóßá êáèþò êáé ðåñéóóüôåñåò ðëçñïöïñßåò ó÷åôéêÜ ìå ãåíéêÜ ðñïâëÞìáôá êáôÜ ôçí åãêáôÜóôáóç ìðïñåßôå íá âñåßôå óôï ÊåöÜëáéï ôïõ Åã÷åéñéäßïõ ó÷åôéêÜ ìå ôçí åãêáôÜóôáóç ôïõ FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html).

2. Ôé ìðïñþ íá êÜíù áí ôï image ôçò äéóêÝôáò äåí ÷ùñÜåé óå ìéá ìüíï äéóêÝôá;

Ìéá äéóêÝôá ôùí 3.5 éíôóþí (1.44MB) ìðïñåß íá ÷ùñÝóåé ùò 1474560 bytes äåäïìÝíùí. Ôï áñ÷åßï image ôçò äéóêÝôáò åêêßíçóçò åßíáé áêñéâþò áõôü ôï ìÝãåèïò.

ÓõíçèéóìÝíá ëÜèç êáôÜ ôçí ðñïåôïéìáóßá ôçò äéóêÝôáò åêêßíçóçò:

- ÊáôÝâáóìá ôçò äéóêÝôáò ÷ùñßò ôç ÷ñÞóç ôïõ *äõáäéêïý (binary)* ôñüðïõ üôáí ÷ñçóéìïðïéåßôáé åîõðçñåôçôÞò FTP.

  ÊÜðïéá ðñïãñÜììáôá-ðåëÜôåò ãéá FTP, ÷ñçóéìïðïéïýí áðü ðñïåðéëïãÞ ôïí *ascii (êåéìÝíïõ)* ôñüðï ìåôáöïñÜò äåäïìÝíùí êáé ðñïóðáèïýí íá áëëÜîïõí ôïõò ÷áñáêôÞñåò ôÝëïõò ãñáììÞò þóôå íá ôáéñéÜæïõí ìå ôéò óõìâÜóåéò ðïõ ÷ñçóéìïðïéïýíôáé áðü ôï óýóôçìá óôï ïðïßï ãßíåôáé ç ëÞøç. Áõôü ó÷åäüí óßãïõñá èá êáôáóôñÝøåé ôï image åêêßíçóçò. ÅëÝãîôå ôï ìÝãåèïò ôïõ image åêêßíçóçò ðïõ êáôåâÜóáôå: Áí äåí åßíáé *áêñéâþò* áõôü ðïõ öáßíåôáé óôïí åîõðçñåôçôÞ, ç äéáäéêáóßá ôïõ êáôåâÜóìáôïò èá ðñÝðåé íá èåùñçèåß ýðïðôç.

  Ãéá íá ðáñáêÜìøåôå ôï ðñüâëçìá: ãñÜøôå *binary* óôçí ãñáììÞ åíôïëþí ôïõ FTP, ìåôÜ ðïõ èá óõíäåèåßôå óôïí åîõðçñåôçôÞ, êáé ðñéí îåêéíÞóåôå íá êáôåâÜæåôå ôï image.

- ×ñÞóç ôçò åíôïëÞò `copy` ôïõ DOS (ç áíôßóôïé÷ï ãñáöéêïý åñãáëåßïõ) ãéá ôç ìåôáöïñÜ ôïõ áñ÷åßïõ image óôç äéóêÝôá.

  ÐñïãñÜììáôá üðùò ôï `copy`, äåí èá ëåéôïõñãÞóïõí óùóôÜ êáèþò ôï image åêêßíçóçò Ý÷åé öôéá÷ôåß þóôå íá åêêéíåß Üìåóá. Ôï image Ý÷åé ôá ðëÞñç ðåñéå÷üìåíá ôçò äéóêÝôáò, ôñï÷éÜ ðñïò ôñï÷éÜ, êáé äåí ðñÝðåé íá ãñáöåß óôç äéóêÝôá ùò êáíïíéêü áñ÷åßï. Èá ðñÝðåé íá ôï ìåôáöÝñåôå óôç äéóêÝôá ìå “ùìü” ôñüðï, ÷ñçóéìïðïéþíôáò åñãáëåßá ÷áìçëïý åðéðÝäïõ (low-level) (üðùò ð.÷. ôï `fdimage` Þ ôï `rawrite`) ðïõ ðåñéãñÜöïíôáé óôïí ïäçãü åãêáôÜóôáóçò ôïõ FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html).

3. Ðïõ âñßóêïíôáé ïé ïäçãßåò ãéá ôçí åãêáôÜóôáóç ôïõ FreeBSD;

Ìðïñåßôå íá âñåßôå ôéò ïäçãßåò óôçí êáôá÷þñçóç ôïõ Åã÷åéñéäßïõ ó÷åôéêÜ ìå ôçí åãêáôÜóôáóç ôïõ FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html).

4. Ôé ÷ñåéÜæïìáé ãéá íá åêôåëÝóù ôï FreeBSD;

Ãéá åêäüóåéò ôïõ FreeBSD áðü ôï 5.X êáé ìåôÜ ÷ñåéÜæåóôå Ýíá PC ìå åðåîåñãáóôÞ 486 Þ êáëýôåñï, ìå 24 MB Þ ðåñéóóüôåñç RAM, êáé ôïõëÜ÷éóôïí 150 MB ÷þñïõ óôï óêëçñü óáò äßóêï.

Ìðïñåßôå íá åêôåëÝóåôå üëåò ôéò åêäüóåéò ôïõ FreeBSD ìå ìéá ÷áìçëþí ðñïäéáãñáöþí, ìïíü÷ñùìç (MDA) êÜñôá ãñáöéêþí, áëëÜ ãéá íá ÷ñçóéìïðïéÞóåôå ôï Xorg, èá ÷ñåéáóôåßôå êÜñôá VGA Þ êáëýôåñç.

Äåßôå åðßóçò êáé ôï ÊåöÜëáéï 4.

5. Ðùò ìðïñþ íá äçìéïõñãÞóù ôç äéêÞ ìïõ, ðñïóáñìïóìÝíç äéóêÝôá åãêáôÜóôáóçò;

Ôç äåäïìÝíç óôéãìÞ, äåí õðÜñ÷åé ôñüðïò íá öôéÜîåôå *áðëþò* ìéá ðñïóáñìïóìÝíç äéóêÝôá åãêáôÜóôáóçò. Èá ðñÝðåé íá äçìéïõñãÞóåôå ïëüêëçñç íÝá Ýêäïóç, ç ïðïßá èá ðåñéëáìâÜíåé êáé ôçí ðñïóáñìïóìÝíç äéóêÝôá åãêáôÜóôáóçò óáò.

Ãéá íá äçìéïõñãÞóåôå ìéá ðñïóáñìïóìÝíç Ýêäïóç, áêïëïõèÞóôå ôéò ïäçãßåò óôï Üñèñï ó÷åôéêÜ ìå ôçí ÏñãÜíùóç Åêäüóåùí (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/releng/article.html).

6. Ìðïñþ íá Ý÷ù ðåñéóóüôåñá áðü Ýíá ëåéôïõñãéêÜ óõóôÞìáôá óôïí õðïëïãéóôÞ ìïõ;

Ñßîôå ìéá ìáôéÜ óôçí óåëßäá ðïëëáðëþí ëåéôïõñãéêþí (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/multi-os/index.html).

7. Ìðïñïýí íá óõíõðÜñîïõí ôá Windows ìå ôï FreeBSD;

ÅãêáôáóôÞóôå ðñþôá ôá Windows, êáé ìåôÜ ôï FreeBSD. Ï äéá÷åéñéóôÞò åêêßíçóçò ôïõ FreeBSD èá óáò åðéôñÝðåé Ýðåéôá íá åðéëÝîåôå ôçí åêêßíçóç åßôå ôùí Windows åßôå ôïõ FreeBSD. Áí åãêáôáóôÞóåôå ôá Windows ìåôÜ ôï FreeBSD, èá óáò óâÞóïõí ôïí äéá÷åéñéóôÞ åêêßíçóçò, ÷ùñßò êáí íá óáò ñùôÞóïõí. Áí óáò óõìâåß, äåßôå ôï åðüìåíï ôìÞìá.

8. Ôá Windows êáôÝóôñåøáí ôï äéá÷åéñéóôÞ åêêßíçóçò ìïõ! Ðùò èá ôïí åðáíáöÝñù;

Ìðïñåßôå íá åðáíåãêáôáóôÞóåôå ôï äéá÷åéñéóôÞ åêêßíçóçò ôïõ FreeBSD ÷ñçóéìïðïéþíôáò Ýíáí áðü ôïõò ôñåéò ðáñáêÜôù ôñüðïõò:

- ÅêôåëþíôáòôïDOS, ìåôáêéíçèåßôå óôïí êáôÜëïãï tools/ ôçò FreeBSD äéáíïìÞò óáò, êáé øÜîôå ãéá ôï áñ÷åßï `bootinst.exe`. ÅêôåëÝóôå ôï üðùò öáßíåôáé ðáñáêÜôù:

  ```
  ...\TOOLS> bootinst.exe boot.bin
  ```

  êáé ï äéá÷åéñéóôÞò åêêßíçóçò èá åðáíåãêáôáóôáèåß.
- ÎåêéíÞóôå îáíÜ ÷ñçóéìïðïéþíôáò ôç äéóêÝôá åêêßíçóçò ôïõ FreeBSD êáé ðçãáßíåôå óôçí åðéëïãÞ Custom Installation ôïõ ìåíïý. ÅðéëÝîôå Partition. ÅðéëÝîôå ôïí ïäçãü ðïõ êáíïíéêÜ èá ðåñéåß÷å ôïí äéá÷åéñéóôÞ åêêßíçóçò óáò (óõíÞèùò ôïí ðñþôï) êáé üôáí öôÜóåôå óôïí åðåîåñãáóôÞ êáôáôìÞóåùí (partition editor) ãéá ôïí

ïäçãü áõôü, ôï ðñþôï ðñÜãìá ðïõ èá êÜíåôå (÷ùñßò Üëëåò áëëáãÝò), åßíáé íá åðéëÝîåôå (W)rite. ÅðéëÝîôå yes óôçí åðéâåâáßùóç ðïõ èá åìöáíéóôåß êáé üôáí öôÜóåôå óôçí ðñïôñïðÞ åðéëïãÞò Äéá÷åéñéóôÞ Åêêßíçóçò, åðéëÝîôå “Boot Manager”. Ìå ôïí ôñüðï áõôü ï äéá÷åéñéóôÞò åêêßíçóçò èá ãñáöåß îáíÜ óôï äßóêï. Ìðïñåßôå ôþñá íá âãåßôå áðü ôï ìåíïý ôçò åãêáôÜóôáóçò êáé íá åðáíåêêéíÞóåôå áðü ôï óêëçñü äßóêï, üðùò óõíÞèùò.

- ÎåêéíÞóôå ìå ôç âïÞèåéá ôçò äéóêÝôáò åêêßíçóçò (Þ ôïõ CD) ôïõ FreeBSD êáé åðéëÝîôå “Fixit” áðü ôï ìåíïý. ÅðéëÝîôå åßôå ôç äéóêÝôá Fixit åßôå ôï CD #2 (ôï “live” óýóôçìá áñ÷åßùí) êáé èá åéóÝëèåôå óôï êÝëõöïò fixit. ÅêôåëÝóôå Ýðåéôá ôçí áêüëïõèç åíôïëÞ:

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

  áíôéêáèéóôþíôáò ôï `bootdevice` ìå ôïí ðñáãìáôéêü óáò äßóêï åêêßíçóçò, üðùò ð.÷. `ad0` (ðñþôïò äßóêïò IDE), `ad4` (ðñþôïò äßóêïò IDE óôïí âïçèçôéêü åëåãêôÞ), `da0` (ðñþôïò äßóêïò SCSI), ê.ë.ð.

9. Ï öïñçôüò ìïõ õðïëïãéóôÞò óåéñÜò IBM Thinkpad, óåéñÜò Á, Ô Þ ×, óôáìáôÜåé íá áðïêñßíåôáé üôáí ðñïóðáèþ íá åêêéíÞóù ôï FreeBSD ìåôÜ ôçí åãêáôÜóôáóç. Ðùò ìðïñþ íá åðéëýóù áõôü ôï ðñüâëçìá;

Ðñüêåéôáé ãéá Ýíá ðñüâëçìá óôéò áñ÷éêÝò åêäüóåéò ôïõ BIOS ôçò ÉÂÌ óôá óõãêåêñéìÝíá ìç÷áíÞìáôá, ôï ïðïßï áíáãíùñßæåé ôçí êáôÜôìçóç ôïõ FreeBSD ùò ðéèáíü äéáìÝñéóìá FAT ãéá ôç ëåéôïõñãßá äéáñáíïðïßçóçò óôï äßóêï (suspend-to-disk). Ôï BIOS óôáìáôÜåé íá áðïêñßíåôáé êáèþò ðñïóðáèåß íá áíáëýóåé ôï äéáìÝñéóìá ôïõ FreeBSD.

Óýìöùíá ìå ôçí IBM[1], ôá áêüëïõèá ìïíôÝëá êáé åêäüóåéò BIOS ðåñéëáìâÜíïõí ôéò áðáñáßôçôåò äéïñèþóåéò.

ÌïíôÝëï	**¸êäïóç BIOS**
T20	IYET49WW Þ ìåôáãåíÝóôåñï
T21	KZET22WW Þ ìåôáãåíÝóôåñï
A20p	IVET62WW Þ ìåôáãåíÝóôåñï
A20m	IWET54WW Þ ìåôáãåíÝóôåñï
A21p	KYET27WW Þ ìåôáãåíÝóôåñï
A21m	KXET24WW Þ ìåôáãåíÝóôåñï
A21e	KUET30WW

¸÷åé áíáöåñèåß üôé ìåôáãåíÝóôåñåò åêäüóåéò ôïõ BIOS ôçò ÉÂÌ, ßóùò ðáñïõóéÜæïõí îáíÜ ôï ðñüâëçìá. To ìÞíõìá áõôü (http://docs.FreeBSD.org/cgi/mid.cgi?20010427133759.A71732) áðü ôïí Jacques Vidrine óôçí çëåêôñïíéêÞ ëßóôá ôïõ FreeBSD ãéá öïñçôïýò õðïëïãéóôÝò (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile) ðåñéãñÜöåé ìéá äéáäéêáóßá ç ïðïßá ßóùò äïõëÝøåé óå íåüôåñá ìïíôÝëá öïñçôþí ôçò IBM ôá ïðïßá äåí åêêéíïýí êáíïíéêÜ ôï FreeBSD, êáé óôá ïðïßá ìðïñåßôå íá áíáâáèìßóåôå ôï BIOS Þ êáé íá åðéóôñÝøåôå óå ðñïçãïýìåíç Ýêäïóç ôïõ.

Áí Ý÷åôå ðáëéüôåñï BIOS, áëëÜ äåí Ý÷åôå ôçí åðéëïãÞ íá ôï áíáâáèìßóåôå, Ýíáò ôñüðïò íá îåðåñÜóåôå ôï ðñüâëçìá åßíáé íá åãêáôáóôÞóåôå ôï FreeBSD, íá áëëÜîåôå ôïí áíáãíùñéóôéêü áñéèìü (partition id) ôçò êáôÜôìçóçò, êáé íá åãêáôáóôÞóåôå íÝá boot blocks ôá ïðïßá íá ìðïñïýí íá ÷åéñéóôïýí ôï äéáöïñåôéêü áíáãíùñéóôéêü ôçò êáôÜôìçóçò.

Áñ÷éêÜ, èá ðñÝðåé íá åðáíáöÝñåôå ôï ìç÷Üíçìá óå ìéá êáôÜóôáóç þóôå íá ðåñíÜåé áðü ôçí áñ÷éêÞ äéáãíùóôéêÞ ïèüíç. Áõôü áðáéôåß íá áíïñãïðïéÞóåôå ôï ìç÷Üíçìá ÷ùñßò íá ôï áöÞóåôå íá âñåé ôçí êáôÜôìçóç ôïõ FreeBSD óôïí êýñéï äßóêï ôïõ. ¸íáò ôñüðïò åßíáé íá áðïìáêñýíåôå ôïí óêëçñü äßóêï êáé íá ôïí ìåôáêéíÞóåôå ðñïóùñéíÜ óå Ýíá ðáëéüôåñï ThinkPad (üðùò ôï ThinkPad 600) Þ êÜðïéï åðéôñáðÝæéï ìç÷Üíçìá, ÷ñçóéìïðïéþíôáò ôï êáôÜëëçëï êáëþäéï ìåôáôñïðÞò. Óôï ìç÷Üíçìá áõôü ìðïñåßôå ðëÝïí íá äéáãñÜøåôå ôçí êáôÜôìçóç ôïõ FreeBSD êáé Ýðåéôá íá ìåôáêéíÞóåôå ôïí äßóêï óôï áñ÷éêü ìç÷Üíçìá. Ôï ThinkPad èá ðñÝðåé ôþñá íá åêêéíåß êáíïíéêÜ.

Ìå ôï ìç÷Üíçìá óå ëåéôïõñãéêÞ êáôÜóôáóç, ìðïñåßôå ôþñá íá ÷ñçóéìïðïéÞóåôå ôçí äéáäéêáóßá ðïõ ðåñéãñÜöåôáé åäþ þóôå íá Ý÷åôå ôåëéêÜ ìéá åãêáôÜóôáóç ôïõ FreeBSD ðïõ íá ëåéôïõñãåß.

1. ÊáôåâÜóôå ôéò äéóêÝôåò `boot1` êáé `boot2` áðü http://people.FreeBSD.org/~bmah/ThinkPad/. Áðïèçêåýóôå ôá áñ÷åßá áõôÜ êÜðïõ ðïõ èá ìðïñåßôå íá ôá îáíáâñåßôå áñãüôåñá.
2. ÅãêáôáóôÞóôå êáíïíéêÜ ôï FreeBSD óôï ThinkPad. *Ìçí* ÷ñçóéìïðïéÞóåôå ôçí êáôÜóôáóç `Dangerously Dedicated`. *Ìçí* êÜíåôå åðáíåêêßíçóç üôáí ôåëåéþóåé ç åãêáôÜóôáóç.
3. ×ñçóéìïðïéÞóôå åßôå ôï “Emergency Holographic Shell” (**ALT+F4**) åßôå ôï êÝëõöïò “fixit”.
4. ×ñçóéìïðïéÞóôå ôçí fdisk(8) ãéá íá áëëÜîåôå ôïí áíáãíùñéóôéêü áñéèìü ôçò êáôÜôìçóçò ôïõ FreeBSD áðü `165` óå `166` (áõôüò åßíáé ï ôýðïò ðïõ ÷ñçóéìïðïéåßôáé áðü ôï OpenBSD).
5. ÃñÜøôå ôá áñ÷åßá `boot1` êáé `boot2` óôï ôïðéêü óýóôçìá áñ÷åßùí.
6. ×ñçóéìïðïéÞóôå ôï disklabel(8) ãéá íá ãñÜøåôå ôá `boot1` êáé `boot2` óôï slice ôïõ FreeBSD.

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

   Ôï *n* åßíáé ï áñéèìüò ôïõ slice ðïõ Ý÷åôå åãêáôáóôÞóåé ôï FreeBSD.
7. ÊÜíôå åðáíåêêßíçóç. Óôçí ðñïôñïðÞ åêêßíçóçò èá äåßôå ôçí åðéëïãÞ íá îåêéíÞóåôå ôï `OpenBSD`. Óôçí ðñáãìáôéêüôçôá, ìå áõôü ôïí ôñüðï èá îåêéíÞóåôå ôï FreeBSD.

Ôï íá êÜíåôå ôï ðáñáðÜíù íá ëåéôïõñãÞóåé óôçí ðåñßðôùóç ðïõ èÝëåôå íá Ý÷åôå äéðëÞ åêêßíçóç OpenBSD êáé FreeBSD óôïí ßäéï öïñçôü õðïëïãéóôÞ, ôï áöÞíïõìå ùò Üóêçóç óôïí áíáãíþóôç.

10. Ìðïñþ íá êÜíù åãêáôÜóôáóç óå Ýíá äßóêï ìå ÷áëáóìÝíïõò ôïìåßò;

Ìðïñåßôå, áëëÜ åßíáé êáêÞ éäÝá.

Áí äåßôå ÷áëáóìÝíïõò ôïìåßò óå Ýíá óýã÷ñïíï ïäçãü IDE, õðÜñ÷åé ìåãÜëç ðéèáíüôçôá ï ïäçãüò áõôüò, ðïëý óýíôïìá, íá óôáìáôÞóåé íá ëåéôïõñãåß åíôåëþò (ï ïäçãüò äåí Ý÷åé Üëëïõò åíáëëáêôéêïýò êõëßíäñïõò ðïõ íá ìðïñïýí íá áíôéêáôáóôÞóïõí ôïõò ÷áëáóìÝíïõò ìÝóù ôçò åóùôåñéêÞò ëåéôïõñãßáò åðáíáôïðïèÝôçóçò ôùí äåäïìÝíùí, êÜôé ôï ïðïßï óçìáßíåé üôé ï äßóêïò Ý÷åé óçìáíôéêÞ öèïñÜ). Óáò óõíéóôïýìå íá áãïñÜóåôå êáéíïýñéï äßóêï.

Áí Ý÷åôå ïäçãü SCSI ìå ÷áëáóìÝíïõò ôïìåßò äåßôå áõôÞ ôçí áðÜíôçóç.

11. ÄéÜöïñá ðåñßåñãá ðñÜãìáôá óõìâáßíïõí üôáí åêêéíþ ìå ôç äéóêÝôá åêêßíçóçò! Ôé óõìâáßíåé;

Áí âëÝðåôå êáôáóôÜóåéò üðùò ôï ìç÷Üíçìá íá ìçí áðïêñßíåôáé Þ íá êÜíåé åðáíåêêßíçóç ìüíï ôïõ üôáí ðñïóðáèåßôå íá åêêéíÞóåôå ìå ôçí äéóêÝôá åêêßíçóçò, èá ðñÝðåé íá êÜíåôå óôïí åáõôü óáò ôñåéò åñùôÞóåéò:

1. ×ñçóéìïðïéÞóáôå êáéíïýñéåò, öñåóêï-äéáìïñöùìÝíåò, êáé ÷ùñßò ëÜèç äéóêÝôåò (êáôÜ ðñïôßìçóç êáëÞò åôáéñßáò ôéò ïðïßåò ìüëéò âãÜëáôå êáéíïýñéåò áðü ôï êïõôß ôïõò, óå áíôßèåóç ìå ôç äéóêÝôá ðïõ ðÞñáôå ìáæß ìå êÜðïéï ðåñéïäéêü êáé ç ïðïßá âñßóêïíôáí êÜôù áðü ôï êñåâÜôé ôá ôñßá ôåëåõôáßá ÷ñüíéá);
2. ÊáôåâÜóáôå ôï image ôçò äéóêÝôáò ìå äõáäéêü (binary Þ image) ôñüðï ìåôáöïñÜò; (ìç íôñÝðåóôå, áêüìá êáé ïé êáëýôåñïé áðü ìáò Ý÷ïõí êáôÜ ëÜèïò êáôåâÜóåé êÜðïéï äõáäéêü áñ÷åßï óå êáôÜóôáóç ASCII (êåéìÝíïõ), ôïõëÜ÷éóôïí ìéá öïñÜ!)
3. Áí ÷ñçóéìïðïéåßôå Windows 95 Þ 98, åêôåëÝóôå ôï `fdimage` Þ ôï `rawrite` óå êáèáñÞ êáôÜóôáóç DOS; Ôá ëåéôïõñãéêÜ áõôÜ ìðïñåß íá ðáñåìâëçèïýí óå ðñïãñÜììáôá ôá ïðïßá ãñÜöïõí áðåõèåßáò óôï õëéêü, êÜôé ôï ïðïßï óõìâáßíåé êáé ìå ôá ðñïãñÜììáôá äçìéïõñãßáò ôùí äéóêåôþí. Ìðïñåß íá äçìéïõñãçèåß ðñüâëçìá, áêüìá êáé áí ôá åêôåëåßôå óå ðáñÜèõñï DOS ìÝóá áðü ôï ãñáöéêü ðåñéâÜëëïí.

¸÷ïõí åðßóçò áíáöåñèåß ðåñéðôþóåéò üðïõ ôï Netscape® äçìéïõñãåß ðñïâëÞìáôá óôï êáôÝâáóìá ôçò äéóêÝôáò åêêßíçóçò, Ýôóé åßíáé êáëýôåñá íá ÷ñçóéìïðïéÞóåôå êÜðïéï Üëëï ðñüãñáììá FTP, áí áõôü åßíáé äõíáôüí.

12. Îåêßíçóá áðü ôï ATAPI CDROM ìïõ, áëëÜ ôï ðñüãñáììá åãêáôÜóôáóçò ëÝåé üôé äåí âñßóêåé CDROM. Ðïõ ðÞãå;

Ç óõíÞèçò áéôßá áõôïý ôïõ ðñïâëÞìáôïò åßíáé Ýíáò êáêïñõèìéóìÝíïò ïäçãüò CDROM. ÐïëëÜ PC Ýñ÷ïíôáé ðëÝïí ìå ôï CDROM ùò slave óõóêåõÞ óôï äåõôåñåýïíôá åëåãêôÞ, ÷ùñßò íá õðÜñ÷åé óôï ßäéï êáíÜëé óõóêåõÞ master. Áõôü, óýìöùíá ìå ôéò ðñïäéáãñáöÝò ôïõ ATAPI, äåí åßíáé Ýãêõñï, áëëÜ ôá Windows ôçñïýí ôéò ðñïäéáãñáöÝò ÷áëáñÜ, åíþ ôï BIOS ôï áãíïåß êáôÜ ôçí åêêßíçóç. Áõôüò åßíáé êáé ï ëüãïò ðïõ ôï BIOS êáôÜöåñå íá äåé ôï CDROM êáé íá ôï ÷ñçóéìïðïéÞóåé, áëëÜ êáé ï ëüãïò ðïõ ôï FreeBSD äåí ìðüñåóå íá ôï äåé ãéá íá óõíå÷ßóåé ôçí åãêáôÜóôáóç.

Ñõèìßóôå îáíÜ ôï óýóôçìá óáò, þóôå ôï CDROM åßôå íá åßíáé ç master óõóêåõÞ óôïí åëåãêôÞ ðïõ åßíáé óõíäåìÝíç, Þ âåâáéùèåßôå üôé åßíáé slave óå Ýíá åëåãêôÞ IDE ï ïðïßïò üìùò Ý÷åé Üëëç óõóêåõÞ óõíäåìÝíç ùò master.

13. Ìðïñþ íá åãêáôáóôÞóù ôï FreeBSD óôï öïñçôü ìïõ ÷ñçóéìïðïéþíôáò PLIP (Parallel Line IP, IP ìÝóù ÐáñÜëëçëçò Èýñáò);

Íáé. ×ñçóéìïðïéÞóôå ôõðïðïéçìÝíï êáëþäéï ôýðïõ Laplink. Áí ÷ñåéÜæåôáé, äéáâÜóôå ôï ôìÞìá PLIP ôïõ Åã÷åéñéäßïõ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-plip.html) ãéá ëåðôïìÝñåéåò ó÷åôéêÝò ìå äéêôýùóç ìÝóù ðáñÜëëçëçò èýñáò.

14. Ôé ãåùìåôñßá íá ÷ñçóéìïðïéÞóù ãéá ôï óêëçñü ìïõ äßóêï;

Óçìåßùóç: Ìå ôïí üñï "ãåùìåôñßá", åííïïýìå ôïí áñéèìü ôùí êõëßíäñùí, êåöáëþí êáé ôïìÝùí áíÜ ôñï÷éÜ åíüò äßóêïõ. Ãéá åõêïëßá, èá áíáöåñüìáóôå óôïí üñï áõôü ùò C/H/S (Cylinders / Heads / Sectors). Áõôüò åßíáé êáé ï ôñüðïò ìå ôïí ïðïßï ôï BIOS âñßóêåé óå ðïéá ðåñéï÷Þ ôïõ äßóêïõ íá ãñÜøåé.

Ôï ðáñáðÜíù ðñïêáëåß óýã÷õóç óôïõò íÝïõò äéá÷åéñéóôÝò óõóôçìÜôùí. Êáôáñ÷Þí, ç *öõóéêÞ* ãåùìåôñßá åíüò ïäçãïý SCSI åßíáé óõíïëéêÜ Üó÷åôç, áöïý ôï FreeBSD ëåéôïõñãåß ìå âÜóç ôá ìðëïê äßóêïõ. Óôçí ðñáãìáôéêüôçôá, äåí õðÜñ÷åé êáí "áëçèéíÞ" öõóéêÞ ãåùìåôñßá, êáèþò ç ðõêíüôçôá ôùí ôïìÝùí ìåôáâÜëëåôáé áðü ðåñéï÷Þ óå ðåñéï÷Þ ôïõ äßóêïõ. Áõôü ðïõ ïé êáôáóêåõáóôÝò áíáöÝñïõí óõ÷íÜ ùò "öõóéêÞ ãåùìåôñßá" åßíáé óõíÞèùò ç ãåùìåôñßá ìå ôçí üðïéá Ý÷ïõí áíáêáëýøåé üôé ãßíåôáé ç ìéêñüôåñç óðáôÜëç åëåýèåñïõ ÷þñïõ. Ãéá äßóêïõò IDE, ôï FreeBSD ÷ñçóéìïðïéåß ðñÜãìáôé ôï C/H/S, áëëÜ üëïé ïé ìïíôÝñíïé ïäçãïß ìåôáôñÝðïõí åóùôåñéêÜ áõôÝò ôéò áíáöïñÝò óå áíôßóôïé÷á ìðëïê.

Ôï ìüíï ðïõ Ý÷åé ðñáãìáôéêÞ óçìáóßá åßíáé ç *ëïãéêÞ* ãåùìåôñßá. ÁõôÞ åßíáé êáé ç áðÜíôçóç ðïõ äÝ÷åôáé ôï BIOS üôáí ñùôÜåé ôï äßóêï "ðïéá åßíáé ç ãåùìåôñßá óïõ;" ¸ðåéôá, ÷ñçóéìïðïéåß áõôÞ ôç ãåùìåôñßá ãéá íá áðïêôÞóåé ðñüóâáóç óôï äßóêï. Êáèþò ôï FreeBSD ÷ñçóéìïðïéåß ôï BIOS êáôÜ ôçí åêêßíçóç, åßíáé ðïëý óçìáíôéêü ç ãåùìåôñßá áõôÞ íá åßíáé óùóôÞ. Åéäéêüôåñá, áí Ý÷åôå ðåñéóóüôåñá áðü Ýíá ëåéôïõñãéêÜ óõóôÞìáôá óôï äßóêï, ðñÝðåé üëá íá óõìöùíïýí üóï áöïñÜ ôç ãåùìåôñßá. Äéáöïñåôéêü èá Ý÷åôå óïâáñÜ ðñïâëÞìáôá êáôÜ ôçí åêêßíçóç!

Ãéá äßóêïõò SCSI, ç ãåùìåôñßá ðïõ ðñÝðåé íá ÷ñçóéìïðïéçèåß, åîáñôÜôáé áðü ôï áí åßíáé åíåñãïðïéçìÝíç ç õðïóôÞñéîç åêôåôáìÝíçò ìåôÜöñáóçò (áõôü óõ÷íÜ áíáöÝñåôáé ùò "õðïóôÞñéîç ãéá äßóêïõò DOS >1GB" Þ êÜôé áíôßóôïé÷ï). Áí åßíáé áðåíåñãïðïéçìÝíç, ÷ñçóéìïðïéÞóôå *N* êõëßíäñïõò, 64 êåöáëÝò êáé 32 ôïìåßò/ôñï÷éÜ, üðïõ ôï *N* åßíáé ç ÷ùñçôéêüôçôá ôïõ äßóêïõ óå ÌÂ. Ãéá ðáñÜäåéãìá, ãéá Ýíá äßóêï 2GB, èá ðñÝðåé íá Ý÷åôå 2048 êõëßíäñïõò, 64 êåöáëÝò, êáé 32 ôïìåßò/ôñï÷éÜ.

Áí *åßíáé* åíåñãïðïéçìÝíç (êáé óõ÷íÜ ðáñÝ÷åôáé Ýôóé þóôå íá îåðåñíéïýíôáé êÜðïéïé ðåñéïñéóìïß óôï MS-DOS®) êáé ç ÷ùñçôéêüôçôá ôïõ äßóêïõ åßíáé ìåãáëýôåñç áðü 1GB, ÷ñçóéìïðïéÞóôå M êõëßíäñïõò, 63 ôïìåßò áíÜ ôñï÷éÜ (*ü÷é* 64) êáé 255 êåöáëÝò, üðïõ M åßíáé ç ÷ùñçôéêüôçôá ôïõ äßóêïõ óå MB, äéáéñåìÝíç ìå ôï 7.844238 (!). ¸ôóé, óôï ðáñÜäåéãìá ìáò, ï äßóêïò ôùí 2GB èá åß÷å 261 êõëßíäñïõò, 63 ôïìåßò áíÜ ôñï÷éÜ êáé 255 êåöáëÝò.

Áí äåí åßóôå óßãïõñïò ãéá ôï ðáñáðÜíù, Þ áí ôï FreeBSD áðïôý÷åé óôçí áíß÷íåõóç ôçò óùóôÞò ãåùìåôñßáò êáôÜ ôçí åãêáôÜóôáóç, ï áðëïýóôåñïò ôñüðïò ãéá íá ôï ðáñáêÜìøåôå, åßíáé óõíÞèùò íá äçìéïõñãÞóåôå Ýíá ìéêñü äéáìÝñéóìá DOS óôï äßóêï. Ôï BIOS Ýðåéôá èá áíé÷íåýóåé ôç óùóôÞ ãåùìåôñßá, êáé ìðïñåßôå ðÜíôá íá äéáãñÜøåôå ôï äéáìÝñéóìá DOS ìÝóá áðü ôïí åðåîåñãáóôÞ êáôáôìÞóåùí, áí äå èÝëåôå íá ôï êñáôÞóåôå. Ìðïñåß ùóôüóï íá èåëÞóåôå íá ôï áöÞóåôå, ãéá íá ðñïãñáììáôßæåôå êÜñôåò äéêôýïõ êáé ãéá Üëëåò, áíôßóôïé÷åò, åñãáóßåò.

Åíáëëáêôéêü, õðÜñ÷åé Ýíá åëåýèåñá äéáèÝóéìï âïçèçôéêü ðñüãñáììá ôï ïðïßï äéáíÝìåôáé ìå ôï FreeBSD êáé ëÝãåôáé `pfdisk.exe`. Ìðïñåßôå íá ôï âñåßôå óôïí õðïêáôÜëïãï `tools` óôï CDROM ôïõ FreeBSD Þ óôéò äéêôõáêÝò ôïðïèåóßåò FTP ôïõ FreeBSD. Ôï ðñüãñáììá áõôü ìðïñåß íá ÷ñçóéìïðïéçèåß ãéá íá áíáêáëýøåé ôé ãåùìåôñßá ÷ñçóéìïðïéåßôáé áðü ôá Üëëá ëåéôïõñãéêÜ óõóôÞìáôá óôï äßóêï ðïõ ÷ñçóéìïðïéåßôáé. Ìðïñåßôå íá åéóÜãåôå áðåõèåßáò áõôÞ ôç ãåùìåôñßá óôïí åðåîåñãáóôÞ êáôáôìÞóåùí.

15. ÕðÜñ÷ïõí êÜðïéïé ðåñéïñéóìïß óôï ðùò ðñÝðåé íá ÷ùñßóù ôï äßóêï;

Íáé. ÐñÝðåé íá âåâáéùèåßôå üôé ç ñéæéêÞ (root) êáôÜôìçóç âñßóêåôáé êÜôù áðü ôïõò 1024 êõëßíäñïõò, þóôå ôï BIOS íá ìðïñåß íá åêêéíÞóåé ôïí ðõñÞíá áðü áõôÞ. (Óçìåéþóôå üôé áõôüò åßíáé Ýíáò ðåñéïñéóìüò óôï BIOS ôïõ PC, êáé ü÷é óôï FreeBSD).

Ãéá Ýíá äßóêï SCSI, óõíÞèùò áõôü óçìáßíåé üôé ç ñéæéêÞ êáôÜôìçóç èá âñßóêåôáé óôá ðñþôá 1024MB (Þ óôá ðñþôá 4096ÌÂ áí ÷ñçóéìïðïéåßôáé ç åêôåôáìÝíç ìåôÜöñáóç - äåßôå ôçí ðñïçãïýìåíç åñþôçóç). Ãéá äßóêïõò IDE, ôï áíôßóôïé÷ï üñéï åßíáé ôá 504MB.

16. Åßíáé óõìâáôü ôï FreeBSD ìå ðñïãñÜììáôá äéá÷åßñéóçò äßóêùí (disk managers);

To FreeBSD áíáãíùñßæåé êáé åðéôñÝðåé ôç ÷ñÞóç ôïõ Ontrack Disk Manager. Äåí õðïóôçñßæïíôáé Üëëïé äéá÷åéñéóôÝò äßóêùí.

Áí èÝëåôå áðëþò íá ÷ñçóéìïðïéÞóåôå ôï äßóêï ìå ôï FreeBSD, äåí ÷ñåéÜæåóôå äéá÷åéñéóôÞ äßóêïõ. Áðëþò ñõèìßóôå ôï äßóêï ãéá üóï ðåñéóóüôåñï ÷þñï ìðïñåß íá äåé ôï BIOS (óõíÞèùò 504ÌÂ), êáé ôï FreeBSD èá áíáêáëýøåé ðüóï åëåýèåñï ÷þñï Ý÷åôå óôçí ðñáãìáôéêüôçôá. Áí ÷ñçóéìïðïéåßôå êÜðïéï ðáëéü äßóêï óå åëåãêôÞ MFM, ßóùò íá ðñÝðåé íá ðåßôå óôï FreeBSD ðüóïõò êõëßíäñïõò íá ÷ñçóéìïðïéÞóåé.

Áí èÝëåôå íá ÷ñçóéìïðïéÞóåôå ôï äßóêï ôüóï ìå ôï FreeBSD üóï êáé ìå êÜðïéï Üëëï ëåéôïõñãéêü óýóôçìá, èá ðñÝðåé íá ìðïñåßôå íá ôï êÜíåôå ÷ùñßò äéá÷åéñéóôÞ äßóêïõ: áðëþò âåâáéùèåßôå üôé ôï äéáìÝñéóìá åêêßíçóçò ôïõ FreeBSD êáèþò êáé ç êáôÜôìçóç ôïõ Üëëïõ ëåéôïõñãéêïý óõóôÞìáôïò âñßóêïíôáé ìÝóá óôïõò ðñþôïõò 1024 êõëßíäñïõò. Áí åßóôå áñêåôÜ ðñïóåêôéêüò, Ýíá äéáìÝñéóìá åêêßíçóçò (boot) ìåãÝèïõò 20MB èá åßíáé áñêåôü.

17. ¼ôáí åêêéíþ ôï FreeBSD ãéá ðñþôç öïñÜ, ðáßñíù ôï ìÞíõìá "`Missing Operating System`". Ôé óõìâáßíåé;

ÁõôÞ åßíáé ìéá êëáóéêÞ ðåñßðôùóç äéÝíåîçò ìåôáîý ôïõ FreeBSD êáé ôïõ DOS Þ êÜðïéïõ Üëëïõ ëåéôïõñãéêïý ó÷åôéêÜ ìå ôçí éäÝá ðïõ Ý÷åé ôï êáèÝíá ãéá ôçí ãåùìåôñßá ôïõ äßóêïõ. Èá ðñÝðåé íá åðáíåãêáôáóôÞóåôå ôï FreeBSD áëëÜ ôçñþíôáò ðñïóåêôéêÜ ôéò ïäçãßåò ðïõ äþóáìå ðéï ðÜíù, åßíáé ó÷åäüí óßãïõñï üôé èá ôá êáôáöÝñåôå.

18. Ãéáôß äåí ìðïñþ íá óõíå÷ßóù ðÝñá áðü ôçí ðñïôñïðÞ `F?` ôïõ äéá÷åéñéóôÞ åêêßíçóçò;

Áõôü åßíáé Üëëï Ýíá óýìðôùìá ôïõ ðñïâëÞìáôïò ðïõ ðåñéãñÜöåôáé óôçí ðñïçãïýìåíç åñþôçóç. Äåí óõìðßðôåé ç ãåùìåôñßá ôïõ BIOS ìå áõôÞ ôïõ FreeBSD! Áí ï åëåãêôÞò Þ ôï BIOS óáò õðïóôçñßæåé ìåôÜöñáóç êõëßíäñùí (óõ÷íÜ áíáöÝñåôáé ùò ">1GB drive support"), äïêéìÜóôå íá áëëÜîåôå áõôÞ ôç ôéìÞ êáé íá åðáíåãêáôáóôÞóåôå ôï FreeBSD.

19. ÐñÝðåé íá åãêáôáóôÞóù üëï ôïí ðçãáßï êþäéêá;

ÃåíéêÜ, ü÷é. Ùóôüóï óáò óõíéóôïýìå íá åãêáôáóôÞóåôå, ùò åëÜ÷éóôï, ôïí ðçãáßï êþäéêá ôçò äéáíïìÞò `base`, ï ïðïßïò ðåñéëáìâÜíåé áñêåôÜ áðü ôá áñ÷åßá ðïõ áíáöÝñïíôáé åäþ, êáèþò êáé ôïí ðçãáßï êþäéêá ôçò äéáíïìÞò `sys`, ç ïðïßá ðåñéëáìâÜíåé ôïí ðçãáßï êþäéêá ôïõ ðõñÞíá. Äåí õðÜñ÷åé ùóôüóï êÜôé óôï óýóôçìá ôï ïðïßï íá áðáéôåß ôçí ýðáñîç ôïõ ðçãáßïõ êþäéêá ãéá íá ëåéôïõñãÞóåé, åêôüò áðü ôï ðñüãñáììá ñýèìéóçò ôïõ ðõñÞíá config(8). Ìå åîáßñåóç ôïí ðçãáßï êþäéêá ôïõ ðõñÞíá, ç äïìÞ ôïõ óõóôÞìáôïò ìåôáãëþôôéóçò ìáò åßíáé ôÝôïéá, þóôå ìðïñåßôå ðÜíôá íá ðñïóáñôÞóåôå ôï äÝíôñï ðçãáßïõ êþäéêá ìÝóù NFS (êáé ìå äéêáéþìáôá ìüíï áíÜãíùóçò) êáé ðÜëé íá ìðïñåßôå íá äçìéïõñãÞóåôå íÝá åêôåëÝóéìá (ëüãù ôïõ ðåñéïñéóìïý ðïõ õðÜñ÷åé óôïí ðçãáßï êþäéêá ôïõ ðõñÞíá, óáò óõíéóôïýìå íá ìçí êÜíåôå ôçí ðñïóÜñôçóç áðåõèåßáò óôïí êáôÜëïãï `/usr/src`, áëëÜ íá ÷ñçóéìïðïéÞóåôå êÜðïéá Üëëç ôïðïèåóßá êáé íá äçìéïõñãÞóåôå ôïõò êáôÜëëçëïõò óõìâïëéêïýò äåóìïýò ðïõ íá áíôéãñÜöïõí ôç äïìÞ ôçò êïñõöáßáò éåñáñ÷ßáò ôïõ äÝíôñïõ ðçãáßïõ êþäéêá).

Áí Ý÷åôå Üìåóá äéáèÝóéìï ôïí ðçãáßï êþäéêá, êáé ãíùñßæåôå ðùò íá ìåôáãëùôôßóåôå Ýíá ïëüêëçñï óýóôçìá áðü áõôüí, èá äéåõêïëõíèåßôå ðÜñá ðïëý üôáí áíáâáèìßæåôå ôï óýóôçìá óáò óå ìåëëïíôéêÝò åêäüóåéò ôïõ FreeBSD.

Ãéá íá åðéëÝîåôå Ýíá õðïóýíïëï ôïõ ðçãáßïõ êþäéêá, áðü ôçí åðéëïãÞ Distributions ôïõ åñãáëåßïõ åãêáôÜóôáóçò óõóôÞìáôïò, åðéëÝîôå ôï ìåíïý Custom.

20. ×ñåéÜæåôáé íá öôéÜîù ðñïóáñìïóìÝíï ðõñÞíá;

Ç äçìéïõñãßá åíüò íÝïõ ðõñÞíá Þôáí áñ÷éêÜ ó÷åäüí õðï÷ñåùôéêü âÞìá óå ìéá åãêáôÜóôáóç FreeBSD, áëëÜ ïé ðéï ðñüóöáôåò åêäüóåéò Ý÷ïõí ùöåëçèåß áðü ôçí åéóáãùãÞ áéóèçôÜ öéëéêüôåñùí ðñïãñáììÜôùí ñýèìéóçò ôïõ ðõñÞíá. Áðü ôï FreeBSD 5.X êáé ìåôÜ, åßíáé áñêåôÜ åýêïëï íá ñõèìßóåôå ôïí ðõñÞíá ÷ñçóéìïðïéþíôáò ôï ðïëý ðéï åõÝëéêôï óýóôçìá ôùí "hints" ôá ïðïßá ìðïñåßôå íá ñõèìßóåôå óôçí ðñïôñïðÞ ôïõ loader.

Åíäå÷ïìÝíùò íá áîßæåé áêüìá íá äçìéïõñãÞóåôå Ýíá íÝï ðõñÞíá ï ïðïßïò íá ðåñéÝ÷åé ìüíï ôá ðñïãñÜììáôá ïäÞãçóçò ðïõ ÷ñåéÜæåóôå, ãéá íá ãëõôþóåôå êÜðïéá ìéêñÞ ðïóüôçôá ìíÞìçò RAM, áëëÜ áõôü äåí åßíáé ðëÝïí áðáñáßôçôï ãéá ôá ðåñéóóüôåñá óõóôÞìáôá.

21. Ãéá ôïõò êùäéêïýò ðñüóâáóçò ôùí ÷ñçóôþí, íá ÷ñçóéìïðïéÞóù DES, Blowfish, Þ MD5 êùäéêïðïßçóç, êáé ðùò èá êáèïñßóù ôé èá ÷ñçóéìïðïéïýí ïé ÷ñÞóôåò ìïõ;

Ç ðñïåðéëåãìÝíç ìïñöÞ êñõðôïãñÜöçóçò ãéá êùäéêïýò óôï FreeBSD åßíáé ôï *MD5*. Ç ãåíéêÞ áíôßëçøç åßíáé üôé ðáñÝ÷ïõí êáëýôåñç áóöÜëåéá óå ó÷Ýóç ìå ôçí ðáñáäïóéáêÞ ìïñöÞ ôïõ UNIX ðïõ âáóßæåôáé óôïí áëãüñéèìï *DES*. Ïé êùäéêïß DES åßíáé áêüìá äéáèÝóéìïé, áí ÷ñåéÜæåóôå íá äéáìïéñÜóåôå ôï áñ÷åßï ôùí êùäéêþí óáò ìå ðáëéüôåñá ëåéôïõñãéêÜ óõóôÞìáôá, ôá ïðïßá ÷ñçóéìïðïéïýí áêüìá ôï ðáëéüôåñï êáé ëéãüôåñï áóöáëÝò óýóôçìá (åßíáé äéáèÝóéìá áí åãêáôáóôÞóåôå ôç äéáíïìÞ “crypto” ìÝóù ôïõ sysinstall Þ åãêáèéóôþíôáò ôïí áíôßóôïé÷ï ðçãáßï êþäéêá áí êÜíåôå åãêáôÜóôáóç ìÝóù ðçãáßïõ êþäéêá). Áí åãêáôáóôÞóåôå ôéò âéâëéïèÞêåò crypto èá ìðïñÝóåôå åðßóçò íá ÷ñçóéìïðïéÞóåôå êñõðôïãñÜöçóç Blowfish ç ïðïßá åßíáé áêüìá ðéï áóöáëÞò. Ôï ðïéá ìïñöÞ êùäéêþí ÷ñçóéìïðïéåßôáé ãéá ôïõò íÝïõò êùäéêïýò, åëÝã÷åôáé áðü ôçí äõíáôüôçôá åéóüäïõ “passwd_format” óôï `/etc/login.conf`, ôï ïðïßï ðáßñíåé ôéò ôéìÝò “des”, “blf” (áí åßíáé äéáèÝóéìç) Þ “md5”. Äåßôå ôç óåëßäá manual ôïõ login.conf(5) ãéá ðåñéóóüôåñåò ðëçñïöïñßåò ó÷åôéêÜ ìå ôéò äõíáôüôçôåò åéóüäïõ.

22. Ãéáôß åíþ ç äéóêÝôá åêêßíçóçò îåêéíÜåé êáíïíéêÜ, êñåìÜåé óôçí ïèüíç `Probing Devices...`;

Áí Ý÷åôå åãêáôáóôçìÝíï ïäçãü IDE Zip® Þ Jaz®, áöáéñÝóôå ôïí êáé îáíáðñïóðáèÞóôå. Ç äéóêÝôá åêêßíçóçò ìðïñåß íá ìðåñäåõôåß áðü áõôïýò ôïõò ïäçãïýò. ÌåôÜ ôçí åãêáôÜóôáóç ôïõ óõóôÞìáôïò, ìðïñåßôå íá îáíáóõíäÝóåôå ôïí ïäçãü. Åõåëðéóôïýìå üôé ôï ðñüâëçìá áõôü èá äéïñèùèåß óå åðüìåíç Ýêäïóç.

23. Ãéáôß ðáßñíù ôï ìÞíõìá ëÜèïõò “`panic: can't mount root`” üôáí åêêéíþ ôï óýóôçìá ãéá ðñþôç öïñÜ ìåôÜ ôçí åãêáôÜóôáóç;

Ôï óöÜëìá áõôü ðñïÝñ÷åôáé áðü ôçí óýã÷õóç ðïõ ðñïêáëåßôáé åîáéôßáò ôïõ äéáöïñåôéêïý ôñüðïõ ìå ôïí ïðïßï ôï BIOS êáé ôï boot block áíôéëáìâÜíïíôáé ôïõò óêëçñïýò äßóêïõò. Ôï ðñüâëçìá óõíÞèùò åìöáíßæåôáé óå óõóôÞìáôá ìå äýï äßóêïõò IDE, åéäéêÜ üôáí ïé äßóêïé åßíáé master (Þ ìüíïé ôïõò) ï êáèÝíáò óôï äéêü ôïõ åëåãêôÞ IDE êáé ìå ôï FreeBSD íá åßíáé åãêáôåóôçìÝíïò óôï äßóêï ðïõ âñßóêåôáé óôïí äåõôåñåýïíôá åëåãêôÞ. Ôï boot block íïìßæåé üôé ôï óýóôçìá åßíáé åãêáôåóôçìÝíï óôïí ad0 (ôï äåýôåñï äßóêï ôïõ BIOS) åíþ ï ðõñÞíáò áíáèÝôåé ôïí ðñþôï äßóêï óôïí äåõôåñåýïíôá åëåãêôÞ, ad2. ÌåôÜ ôçí áíß÷íåõóç ôùí óõóêåõþí, ï ðõñÞíáò ðñïóðáèåß íá ðñïóáñôÞóåé áõôü ðïõ ôï boot block ðéóôåýåé üôé åßíáé ï äßóêïò åêêßíçóçò, ad0 åíþ óôçí ðñáãìáôéêüôçôá åßíáé ï ad2 êáé öõóéêÜ áðïôõã÷Üíåé.

Ãéá íá äéïñèþóåôå ôï ðñüâëçìá, êÜíôå Ýíá áðü ôá ðáñáêÜôù:

1. ÅðáíåêêéíÞóôå ôï óýóôçìá êáé ðéÝóôå **Enter** óôçí ðñïôñïðÞ `Booting kernel in 10 seconds; hit [Enter] to interrupt`. Ìå ôïí ôñüðï áõôü èá âñåßôå óôï ðñüãñáììá ôïõ öïñôùôÞ åêêßíçóçò.

   Êáôüðéí ãñÜøôå `set root_disk_unit="`*`disk_number`*`"` . Ôï *`disk_number`* èá åßíáé `0` áí ôï FreeBSD åßíáé åãêáôåóôçìÝíï óôïí master äßóêï ôïõ ðñþôïõ åëåãêôÞ IDE, `1` áí åßíáé åãêáôåóôçìÝíï óôïí slave äßóêï ôïõ ðñþôïõ åëåãêôÞ, `2` áí åßíáé åãêáôåóôçìÝíïò óôïí master äßóêï ôïõ äåõôåñåýïíôïò IDE êáíáëéïý êáé ôÝëïò, `3` áí åßíáé åãêáôåóôçìÝíï óôïí slave äßóêï ôïõ äåõôåñåýïíôïò IDE êáíáëéïý.

   ¸ðåéôá ãñÜøôå `boot`, êáé ôï óýóôçìá óáò èá ðñÝðåé íá åêêéíÞóåé êáíïíéêÜ.

   Ãéá íá êÜíåôå ìüíéìç áõôÞ ôçí áëëáãÞ (þóôå íá ìçí ÷ñåéÜæåôáé íá êÜíåôå ôï ðáñáðÜíù êÜèå öïñÜ ðïõ åðáíåêêéíåßôå Þ åíåñãïðïéåßôå ôï FreeBSD ìç÷Üíçìá óáò), âÜëôå ôç ãñáììÞ `root_disk_unit="`*`disk_number`*`"` óôï áñ÷åßï `/boot/loader.conf.local`.

2. ÌåôáêéíÞóôå ôï äßóêï ôïõ FreeBSD óôïí ðñùôåýïíôá åëåãêôÞ IDE, þóôå ïé óêëçñïß äßóêïé íá åßíáé óõíå÷üìåíïé.

24. Ðïéá åßíáé ôá üñéá ôçò ìíÞìçò;

Ôï üñéï åßíáé ôá 4 gigabytes óå ìéá óõíçèéóìÝíç åãêáôÜóôáóç óå áñ÷éôåêôïíéêÞ i386. Îåêéíþíôáò áðü ôéò åêäüóåéò FreeBSD 4.9 êáé 5.1, õðïóôçñßæåôáé êáé ðåñéóóüôåñç ìíÞìç ìÝóù ôïõ pae(4). ×ñåéÜæåôáé ùóôüóï íá ìåôáãëùôôéóôåß îáíÜ ï ðõñÞíáò ðåñéëáìâÜíïíôáò êáé ìéá Ýîôñá åðéëïãÞ ãéá ôçí åíåñãïðïßçóç ôïõ PAE:

```
options         PAE
```

Ôï FreeBSD/pc98 Ý÷åé üñéï ôá 4 GB ìíÞìçò, êáé äåí ìðïñåß íá ÷ñçóéìïðïéçèåß PAE óå áõôÞ ôçí áñ÷éôåêôïíéêÞ. Óôï FreeBSD/alpha, ôï üñéï ôçò ìíÞìçò åîáñôÜôáé áðü ôïí ôýðï ôïõ õëéêïý ðïõ ÷ñçóéìïðïéåßôáé - ãéá ëåðôïìÝñåéåò äåßôå ôéò Óçìåéþóåéò ¸êäïóçò Õëéêïý ãéá Alpha. ¶ëëåò áñ÷éôåêôïíéêÝò ðïõ õðïóôçñßæïíôáé áðü ôï FreeBSD, Ý÷ïõí áñêåôÜ ìåãáëýôåñá èåùñçôéêÜ üñéá ó÷åôéêÜ ìå ôç ìÝãéóôç ðïóüôçôá ìíÞìçò (ðïëëÜ terabytes).

25. Ðïéá åßíáé ôá üñéá ôïõ óõóôÞìáôïò áñ÷åßùí ffs;

Ãéá óõóôÞìáôá áñ÷åßùí ffs, ôï ìÝãéóôï èåùñçôéêü üñéï åßíáé ôá 8 terabytes (2G blocks), Þ 16ÔÂ ãéá ðñïåðéëåãìÝíï ìÝãåèïò block ôùí 8Ê. Óôçí ðñáãìáôéêüôçôá, õðÜñ÷åé Ýíá áñ÷éêü üñéï 1 terabyte, áëëÜ ìå êÜðïéåò ìåôáôñïðÝò, åßíáé äõíáôüí íá ÷ñçóéìïðïéçèïýí (êáé õðÜñ÷ïõí) óõóôÞìáôá áñ÷åßùí ìåãÝèïõò 4 terabytes.

Ôï ìÝãéóôï ìÝãåèïò åíüò áñ÷åßïõ óå Ýíá óýóôçìá ffs åßíáé ðåñßðïõ 1G blocks, Þ 4TB ìå ìÝãåèïò block ôùí 4K.

Ðßíáêáò 3-1. ÌÝãéóôá ìåãÝèç áñ÷åßùí

ÌÝãåèïò block fs	ëåéôïõñãåß	ðñÝðåé íá ëåéôïõñãåß
4K	4T-1	>4T
8K	>32G	32T-1
16K	>128G	32T-1
32K	>512G	64T-1
64K	>2048G	128T-1

¼ôáí ôï ìÝãåèïò block ôïõ fs åßíáé 4K, ëåéôïõñãïýí ôá ôñéðëÜ Ýììåóá blocks (triple indirect blocks) êáé ôá ðÜíôá èá Ýðñåðå íá ðåñéïñßæïíôáé ìüíï áðü ôï ìÝãéóôï áñéèìü block ðïõ ìðïñåß íá áíáðáñáóôáèåß ìå ôç ÷ñÞóç ôñéðëþí Ýììåóùí blocks (ðåñßðïõ 1K^3 + 1K^2 + 1K), áëëÜ ôåëéêÜ ï ðåñéïñéóìüò ïöåßëåôáé óå Ýíá (ëÜèïò) üñéï 1G-1 óôïõò áñéèìïýò ôùí blocks. Ôï üñéï óôïõò áñéèìïýò ôùí block èá Ýðñåðå íá åßíáé 2G-1. ÕðÜñ÷ïõí êÜðïéá ðñïâëÞìáôá üôáí ïé áñéèìïß ôùí block ôïõ fs ðëçóéÜæïõí ôï 2G-1, áëëÜ ôÝôïéïé áñéèìïß block äåí ìðïñïýí íá ðñïóåããéóôïýí üôáí ôï ìÝãåèïò block fs åßíáé 4Ê.

Ãéá ìåãÝèç block 8Ê êáé ìåãáëýôåñá, ôá ðÜíôá èá Ýðñåðå íá ðåñéïñßæïíôáé áðü ôï üñéï 2G-1 óôïõò áñéèìïýò ôùí block ôïõ fs, áëëÜ óôçí ðñáãìáôéêüôçôá ï ðåñéïñéóìüò ïöåßëåôáé óôï ëÜèïò üñéï 1G-1 óôïõò áñéèìïýò ôùí block ôïõ fs. H ÷ñÞóç ôïõ óùóôïý ïñßïõ ôùí 2G-1 blocks, äçìéïõñãåß ðñÜãìáôé ðñïâëÞìáôá.

26. Ãéáôß ðáßñíù ôï ìÞíõìá ëÜèïò, “`archsw.readin.failed`” ìåôÜ ôçí ìåôáãëþôôéóç êáé åêêßíçóç íÝïõ ðõñÞíá;

Ãéáôß ï ðõñÞíáò óáò êáé ôï õðüëïéðï ôïõ âáóéêïý óõóôÞìáôïò (world) åßíáé åêôüò óõã÷ñïíéóìïý. Ç ëåéôïõñãßá óå áõôÞ ôçí êáôÜóôáóç äåí õðïóôçñßæåôáé. Âåâáéùèåßôå üôé ÷ñçóéìïðïéåßôå ôéò åíôïëÝò `make buildworld` êáé `make buildkernel` ãéá íá áíáâáèìßóåôå ôïí ðõñÞíá óáò.

Ìðïñåßôå íá åêêéíÞóåôå ïñßæïíôáò ôïí ðõñÞíá áðåõèåßáò áðü ôï äåýôåñï óôÜäéï, ðéÝæïíôáò ïðïéïäÞðïôå ðëÞêôñï ìüëéò äåßôå ôï | êáé ðñéí îåêéíÞóåé ï loader.

27. Ç åãêáôÜóôáóç êáôáññÝåé êáôÜ ôçí åêêßíçóç. Ôé ìðïñþ íá êÜíù;

ÄïêéìÜóôå íá áðåíåñãïðïéÞóåôå ôçí õðïóôÞñéîç ACPI. Ìüëéò îåêéíÞóåé ï öïñôùôÞò åêêßíçóçò, ðéÝóôå ôï ðëÞêôñï space. To óýóôçìá óáò èá åìöáíßóåé

`OK`

. ÃñÜøôå

`unset acpi_load`

êáé êáôüðéí

`boot`

.

Óçìåéþóåéò

1. Óå Ýíá e-mail áðü ôïí Keith Frechette <`kfrechet@us.ibm.com`>.

ÊåöÜëáéï 4 Óõìâáôüôçôá Õëéêïý

4.1 ÃåíéêÜ

1. ÈÝëù íá áãïñÜóù õëéêü ãéá ôï FreeBSD óýóôçìá ìïõ. Ðïéï ìïíôÝëï / ìÜñêá / ôýðïò åßíáé ôï êáëýôåñï;

ÕðÜñ÷ïõí óõíÝ÷åéá óõæçôÞóåéò ãéá ôï èÝìá áõôü óôéò ëßóôåò çëåêôñïíéêïý ôá÷õäñïìåßïõ ôïõ FreeBSD. Áõôü ùóôüóï åßíáé áíáìåíüìåíï, êáèþò ôï õëéêü ôùí õðïëïãéóôþí áëëÜæåé ðïëý ãñÞãïñá. Åìåßò *åîáêïëïõèïýìå* íá óõíéóôïýìå ìå Ýìöáóç, íá äéáâÜóåôå ôéò Óçìåéþóåéò Õëéêïý ôïõ FreeBSD 8.0 (http://www.FreeBSD.org/releases/8.0R/hardware.html) Þ 7.2 (http://www.FreeBSD.org/releases/7.2R/hardware.html) êáé íá øÜîåôå óôá áñ÷åßá (http://www.FreeBSD.org/search/#mailinglists) ôùí ëéóôþí çëåêôñïíéêïý ôá÷õäñïìåßïõ, ðñéí áñ÷ßóåôå íá ñùôÜôå ó÷åôéêÜ ìå ôï ôåëåõôáßï êáé êáëýôåñï õëéêü. Åßíáé áñêåôÜ ðéèáíü íá äéáðéóôþóåôå, üôé ãéá ôï õëéêü ðïõ áíáæçôÜôå, õðÞñîå ó÷åôéêÞ óõæÞôçóç ìüëéò ðñéí ìéá åâäïìÜäá.

Áí øÜ÷íåôå ãéá öïñçôü õðïëïãéóôÞ, åëÝãîôå ôá áñ÷åßá ôÞò ëßóôáò çëåêôñïíéêïý ôá÷õäñïìåßïõ freebsd-mobile (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile). ÄéáöïñåôéêÜ, ìÜëëïí èá èÝëåôå íá äåßôå ôá áñ÷åßá ôçò freebsd-questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) Þ ðéèáíüí ìéá ëßóôá ðïõ íá åîåéäéêåýåôáé óôïí ôýðï ôïõ õëéêïý ðïõ øÜ÷íåôå.

4.2 Áñ÷éôåêôïíéêÝò êáé ÅðåîåñãáóôÝò

1. Õðïóôçñßæåé ôï FreeBSD áñ÷éôåêôïíéêÝò äéáöïñåôéêÝò áðü ôçí x86;

Íáé. ÁõôÞ ôç óôéãìÞ ôï FreeBSD ìðïñåß íá åêôåëåóôåß óå áñ÷éôåêôïíéêÝò x86 êáé DEC (ôþñá ðëÝïí Compaq) Alpha. Áðü ôï FreeBSD 5.0 êáé ìåôÜ, õðïóôçñßæïíôáé åðßóçò ïé áñ÷éôåêôïíéêÝò AMD64 êáé Intel EM64T, ç IA-64 êáèþò êáé ç Sparc64®. Áñ÷éôåêôïíéêÝò ðïõ èá õðïóôçñßæïíôáé ìåëëïíôéêÜ, ðåñéëáìâÜíïõí ôçí MIPS® êáé PowerPC. Ìðïñåßôå íá åíçìåñùèåßôå óôéò ëßóôåò ôá÷õäñïìåßïõ freebsd-ppc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ppc) êáé freebsd-mips (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mips) áíôßóôïé÷á ãéá ðëçñïöïñßåò ó÷åôéêÜ ìå ôçí ðñüïäï ôùí åñãáóéþí óå áõôÝò ôéò áñ÷éôåêôïíéêÝò. Ãéá ãåíéêÝò ðëçñïöïñßåò ó÷åôéêÜ ìå íÝåò áñ÷éôåêôïíéêÝò, åããñáöåßôå óôçí çëåêôñïíéêÞ ëßóôá ôïõ FreeBSD ãéá ôéò ìç-Intel ðëáôöüñìåò (http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms).

Áí ôï ìç÷Üíçìá óáò åßíáé äéáöïñåôéêÞò áñ÷éôåêôïíéêÞò êáé ÷ñåéÜæåóôå ëåéôïõñãéêü Üìåóá, óáò óõíéóôïýìå íá ñßîåôå ìéá ìáôéÜ óôï NetBSD (http://www.netbsd.org/) Þ OpenBSD (http://www.openbsd.org/).

2. Õðïóôçñßæåé ôï FreeBSD ÓõììåôñéêÞ Ðïëõåðåîåñãáóßá (SMP);

Íáé. Ôï SMP Þôáí åíåñãïðïéçìÝíï áðü ðñïåðéëïãÞ óôïí ðõñÞíá *GENERIC* Þäç áðü ôï FreeBSD 5.2.

Ç áñ÷éêÞ ðñüèåóç Þôáí íá åßíáé åíåñãïðïéçìÝíï åðßóçò áðü ðñïåðéëïãÞ êáé óôïí ðõñÞíá ôçò Ýêäïóçò 5.3 ôïõ FreeBSD, áëëÜ ëüãù êÜðïéùí ðñïâëçìÜôùí óôçí åêôÝëåóç ôïõ ðõñÞíá SMP óå ìç÷áíÞìáôá ÷ùñßò ðïëëáðëïýò åðåîåñãáóôÝò, áðïöáóßóôçêå íá ìåßíåé áíåíåñãü ìÝ÷ñé ôçí áíôéìåôþðéóç ôïõò. Áõôü ùóôüóï áðïôåëåß ðñïôåñáéüôçôá ãéá ôçí Ýêäïóç 5.4 ôïõ FreeBSD.

4.3 Óêëçñïß äßóêïé, ìïíÜäåò ôáéíßáò, ïäçãïß CD êáé DVD

1. Ðïéá åßäç óêëçñþí äßóêùí õðïóôçñßæïíôáé áðü ôï FreeBSD;

Ôï FreeBSD õðïóôçñßæåé ìïíÜäåò äßóêïõ ôýðïõ EIDE, SATA, SCSI, êáé SAS (ìå ôïí êáôÜëëçëï óõìâáôü åëåãêôÞ — äåßôå ôçí åðüìåíç åíüôçôá), êáèþò êáé üëïõò ôïõò ïäçãïýò ðïõ ÷ñçóéìïðïéïýí ôï áñ÷éêü interface ôçò “Western Digital” (äçë. MFM, RLL, ESDI êáé öõóéêÜ IDE). ºóùò íá ìç ëåéôïõñãÞóïõí êÜðïéïé åëåãêôÝò ESDI ðïõ ÷ñçóéìïðïéïýí ìç-ôõðïðïéçìÝíï interface. Êáëýôåñá íá ðñïóêïëëçèåßôå óå interfaces ôýðïõ WD1002/3/6/7 êáé áíôßóôïé÷á ôïõò.

2. Ðïéïé åëåãêôÝò SCSI Þ SAS õðïóôçñßæïíôáé;

Äåßôå ôçí ðëÞñç ëßóôá óôéò Óçìåéþóåéò Õëéêïý ôïõ FreeBSD 8.0
(http://www.FreeBSD.org/releases/8.0R/hardware.html) Þ 7.2
(http://www.FreeBSD.org/releases/7.2R/hardware.html).

3. Ôé ôýðïé ïäçãþí ôáéíßáò õðïóôçñßæïíôáé;

Ôï FreeBSD õðïóôçñßæåé ïäçãïýò SCSI êáé QIC-36 (ìå interface QIC-02). Ðåñéëáìâáíïíôáé ïäçãïß 8-mm (ãíùóôïß ùò Exabyte) êáèþò êáé ïäçãïß DAT.

ÏñéóìÝíïé áðü ôïõò ðñþôïõò ïäçãïýò 8-mm äåí åßíáé éäéáßôåñá óõìâáôïß ìå SCSI-2, êáé ìðïñåß íá ìçí ëåéôïõñãïýí éêáíïðïéçôéêÜ ìå ôï FreeBSD.

4. Õðïóôçñßæåé ôï FreeBSD ìïíÜäåò åíáëëáãÞò ôáéíéþí (tape changers);

Ôï FreeBSD õðïóôçñßæåé ìïíÜäåò åíáëëáãÞò ôýðïõ SCSI ìå ôçí ÷ñÞóç ôçò óõóêåõÞò ch(4) êáé ôçò åíôïëÞò chio(1). Ìðïñåßôå íá âñåßôå ôéò ëåðôïìÝñåéåò ó÷åôéêÜ ìå ôïí ôñüðï åëÝã÷ïõ ôçò ìïíÜäáò åíáëëáãÞò óôç óåëßäá manual ôïõ chio(1).

Áí äåí ÷ñçóéìïðïéåßôå ôï **AMANDA** Þ êÜðïéï Üëëï ðñüãñáììá ðïõ íá ãíùñßæåé ðùò íá ÷åéñéóôåß ôçí ìïíÜäá åíáëëáãÞò ôáéíéþí, èá ðñÝðåé íá èõìÜóôå üôé ãíùñßæïõí ìüíï ðùò íá ìåôáêéíÞóïõí ìéá ôáéíßá áðü Ýíá óçìåßï óå Ýíá Üëëï, êáé èá ðñÝðåé åóåßò íá óçìåéþóåôå óå ðïéá èÝóç (slot) âñßóêåôáé ç ôáéíßá, êáé óå ðïéá èÝóç ðñÝðåé íá ðÜåé ç ôáéíßá ðïõ âñßóêåôáé áõôÞ ôç óôéãìÞ ìÝóá óôïí ïäçãü.

5. Ðïéïé ïäçãïß CDROM õðïóôçñßæïíôáé áðü ôï FreeBSD;

Õðïóôçñßæåôáé ïðïéïóäÞðïôå ïäçãüò SCSI ðïõ åßíáé óõíäåìÝíïò óå áíôßóôïé÷á õðïóôçñéæüìåíï åëåãêôÞ.

Õðïóôçñßæïíôáé åðßóçò ôá áêüëïõèá ìç-ôõðïðïéçìÝíá CDROM interfaces:

- Mitsumi LU002 (8bit), LU005 (16bit) and FX001D (16bit ôá÷ýôçôáò 2x).
- Sony CDU 31/33A
- Sound Blaster ìç-SCSI CDROM
- Matsushita/Panasonic CDROM

- IDE CDROM óõìâáôÜ ìå ATAPI

¼ëåò ïé êÜñôåò ðïõ äåí åßíáé SCSI åßíáé ãåíéêÜ åîáéñåôéêÜ ðéï áñãÝò üôáí óõãêñßíïíôáé ìå áíôßóôïé÷ïõò SCSI ïäçãïýò, êáé êÜðïéïé ïäçãïß ôýðïõ ATAPI ßóùò íá ìç ëåéôïõñãÞóïõí.

Ôá åðßóçìá FreeBSD CDROM ISO, êáèþò êáé ôá CDROM áðü ôï Daemon News êáé ôï FreeBSD Mall, õðïóôçñßæïõí åêêßíçóç áðåõèåßáò áðü ôï CD.

6. Ðïéïé ïäçãïß CD-RW õðïóôçñßæïíôáé áðü ôï FreeBSD;

Ôï FreeBSD õðïóôçñßæåé ïðïéïäÞðïôå ïäçãü IDE CD-R Þ CD-RW óõìâáôü ìå ATAPI. Äåßôå ôï burncd(8) ãéá ëåðôïìÝñåéåò.

Ôï FreeBSD õðïóôçñßæåé åðßóçò ïðïéïäÞðïôå ïäçãü SCSI CD-R Þ CD-RW. ÅãêáôáóôÞóôå êáé ÷ñçóéìïðïéÞóôå ôçí åíôïëÞ `cdrecord` áðü ôçí óõëëïãÞ ôùí ports Þ áðü ðáêÝôï, êáé âåâáéùèåßôå üôé Ý÷åôå åíóùìáôþóåé óôïí ðõñÞíá óáò ôç óõóêåõÞ `pass`.

7. Õðïóôçñßæåé ôï FreeBSD ïäçãïýò Zip;

Ôï FreeBSD õðïóôçñßæåé åããåíþò ïäçãïýò Zip SCSI êáé ATAPI. Ïé ïäçãïß SCSI ZIP ìðïñïýí íá ëåéôïõñãÞóïõí ìüíï áí Ý÷ïõí ñõèìéóôåß óå SCSI IDs 5 Þ 6, áëëÜ áí ôï õðïóôçñßæåé ï åëåãêôÞò SCSI ðïõ ÷ñçóéìïðïéåßôå, ìðïñåßôå áêüìá êáé íá åêêéíÞóåôå áðü áõôïýò. Äåí åßíáé îåêÜèáñï ðïéïé åëåãêôÝò SCSI õðïóôçñßæïõí åêêßíçóç áðü óõóêåõÝò ìå ID äéáöïñåôéêü áðü 0 Þ 1, Ýôóé èá ðñÝðåé íá óõìâïõëåõèåßôå ôçí ôåêìçñßùóç ôïõ åëåãêôÞ óáò, áí èÝëåôå íá ÷ñçóéìïðïéÞóåôå áõôÞ ôç äõíáôüôçôá.

Ôï FreeBSD õðïóôçñßæåé åðßóçò ïäçãïýò Zip ðáñÜëëçëçò èýñáò. Âåâáéùèåßôå üôé ï ðõñÞíáò óáò ðåñéÝ÷åé ôá ðñïãñÜììáôá ïäÞãçóçò ãéá ôéò óõóêåõÝò `scbus0`, `da0`, `ppbus0`, êáé `vp0` (ï ðõñÞíáò GENERIC ðåñéÝ÷åé ôá ðÜíôá åêôüò áðü ôï `vp0`). Ìå ÷ñÞóç áõôþí ôùí ðñïãñáììÜôùí ïäÞãçóçò, ï ïäçãüò ôçò ðáñÜëëçëçò èýñáò èá ðñÝðåé íá åßíáé äéáèÝóéìïò ùò óõóêåõÞ `/dev/da0s4`. Ìðïñåßôå íá ðñïóáñôÞóåôå äßóêïõò ÷ñçóéìïðïéþíôáò ôçí åíôïëÞ `mount /dev/da0s4 /mnt` Þ (ãéá äßóêïõò ðïõ Ý÷ïõí äéáìïñöùèåß ìÝóù dos) ôçí `mount_msdos /dev/da0s4 /mnt`.

ÄéáâÜóôå åðßóçò ôï FAQ ó÷åôéêÜ ìå áöáéñïýìåíïõò ïäçãïýò óå åðüìåíï óçìåßï áõôïý ôïõ êåöáëáßïõ, êáèþò êáé ôéò óçìåéþóåéò ó÷åôéêÜ ìå ôçí “äéáìüñöùóç” óôï êåöÜëáéï Äéá÷åßñéóçò.

8. Õðïóôçñßæåé ôï FreeBSD Jaz, EZ êáé Üëëïõò áöáéñïýìåíïõò ïäçãïýò;

Áðëþò ëåéôïõñãïýí. Ïé ðåñéóóüôåñïé áðü áõôïýò åßíáé SCSI, êáé Ýôóé äåß÷íïõí óáí SCSI äßóêïé óôï FreeBSD. Ôï IDE EZ åìöáíßæåôáé óáí ïäçãüò IDE.

Âåâáéùèåßôå üôé Ý÷åôå åíåñãïðïéÞóåé ôõ÷üí åîùôåñéêÝò óõóêåõÝò ðñéí åêêéíÞóåôå ôï óýóôçìá óáò.

Ãéá íá áëëÜîåôå ìÝóá áðïèÞêåõóçò åí þñá ëåéôïõñãßáò, åëÝãîôå ôéò mount(8), umount(8), êáé ôçí camcontrol(8) (ãéá óõóêåõÝò SCSI) Þ atacontrol(8) (ãéá óõóêåõÝò IDE), êáé åðßóçò ôéò óçìåéþóåéò ó÷åôéêÜ ìå ôç ÷ñÞóç áöáéñïýìåíùí ïäçãþí óå åðüìåíï ôìÞìá ôïõ FAQ.

4.4 Ðëçêôñïëüãéá êáé ðïíôßêéá

1. Õðïóôçñßæåé ôï FreeBSD ôï USB ðëçêôñïëüãéï ìïõ;

Ôï FreeBSD õðïóôçñßæåé åããåíþò USB ðëçêôñïëüãéá. ÅíåñãïðïéÞóôå ôçí õðïóôÞñéîç USB óôï `/etc/rc.conf`.

Ìüëéò åíåñãïðïéçèåß ç õðïóôÞñéîç USB ðëçêôñïëïãßïõ óôï óýóôçìá óáò, ôï ðëçêôñïëüãéï ôýðïõ AT áíáãíùñßæåôáé ùò `/dev/kbd0` êáé ôï USB ðëçêôñïëüãéï ãßíåôáé `/dev/kbd1`, áí åßíáé êáé ôá äýï óõíäåìÝíá óôï óýóôçìá. Áí õðÜñ÷åé ìüíï ôï USB ðëçêôñïëüãéï, èá áíáãíùñéóôåß ùò `/dev/ukbd0`.

Áí èÝëåôå íá ÷ñçóéìïðïéÞóåôå ôï USB ðëçêôñïëüãéï óôçí êïíóüëá, èá ðñÝðåé íá äçëþóåôå óõãêåêñéìÝíá óôïí ïäçãü ôçò êïíóüëáò íá ÷ñçóéìïðïéÞóåé ôï õðÜñ÷ïí USB ðëçêôñïëüãéï. Áõôü ìðïñåß íá ãßíåé åêôåëþíôáò ôçí áêüëïõèç åíôïëÞ ùò ìÝñïò ôçò äéáäéêáóßáò áñ÷éêïðïßçóçò ôïõ óõóôÞìáôïò:

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

ÐáñáôçñÞóôå üôé áí ôï ðëçêôñïëüãéï USB åßíáé ôï ìïíáäéêü ðëçêôñïëüãéï, èá åßíáé äéáèÝóéìï ùò `/dev/ukbd0`, êáé ç åíôïëÞ èá äåß÷íåé üðùò ðáñáêÜôù:

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

¸íá êáëü ìÝñïò ãéá íá ðñïóèÝóåôå ôçí ðáñáðÜíù åíôïëÞ, åßíáé ôï áñ÷åßï `/etc/rc.i386`.

Ìüëéò ãßíåé áõôü, ôï USB ðëçêôñïëüãéï èá ðñÝðåé íá ëåéôïõñãåß êáé óôï × ðåñéâÜëëïí, ÷ùñßò áíÜãêç åéäéêþí ñõèìßóåùí.

Ç åí èåñìþ óýíäåóç êáé áðïóýíäåóç USB ðëçêôñïëïãßùí, ßóùò íá ìç ëåéôïõñãåß áêüìá óùóôÜ. Óáò óõíéóôïýìå íá óõíäÝóåôå ôï ðëçêôñïëüãéï ðñéí ôçí åêêßíçóç ôïõ óõóôÞìáôïò, êáé íá ôï áöÞóåôå óõíäåìÝíï ìÝ÷ñé ôïí ôåñìáôéóìü, ãéá íá áðïöýãåôå ôõ÷üí ðñïâëÞìáôá.

Äåßôå ôç óåëßäá manual ukbd(4) ãéá ðåñéóóüôåñåò ðëçñïöïñßåò.

2. ¸÷ù Ýíá ìç-ôõðéêü ðïíôßêé ôýðïõ bus. Ðùò èá ôï ñõèìßóù;

Ôï FreeBSD õðïóôçñßæåé ðïíôßêéá ôýðïõ bus êáé ôïí ôýðï InPort bus áðü êáôáóêåõáóôÝò üðùò Microsoft, Logitech êáé ATI. Ï ðõñÞíáò GENERIC äåí ðåñéÝ÷åé ôïí áðáñáßôçôï ïäçãü óõóêåõÞò. Ãéá íá ðåñéëÜâåôå ôï ðñüãñáììá ïäÞãçóçò óôïí äéêü óáò ðñïóáñìïóìÝíï ðõñÞíá, ðñïóèÝóôå ôçí áêüëïõèç ãñáììÞ óôï áñ÷åßï ñõèìßóåùí ðõñÞíá:

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

Ôá ðïíôßêéá ôýðïõ bus óõíÞèùò Ýñ÷ïíôáé ìå äéêÝò ôïõò êÜñôåò åðÝêôáóçò. Åíäå÷ïìÝíùò íá Ý÷åôå äõíáôüôçôá íá ñõèìßóåôå ôçí êÜñôá óå äéáöïñåôéêÞ äéåýèõíóç èýñáò êáé IRQ áðü áõôÜ ðïõ öáßíïíôáé ðáñáðÜíù. Óõìâïõëåõèåßôå ôï åã÷åéñßäéï ôïõ ðïíôéêéïý óáò êáé ôç óåëßäá manual mse(4) ãéá ðåñéóóüôåñåò ðëçñïöïñßåò.

3. Ðùò ìðïñþ íá ÷ñçóéìïðïéÞóù ôï ðïíôßêé ìïõ ôýðïõ PS/2 (“ðüñôáò ðïíôéêéïý” Þ “ðëçêôñïëïãßïõ”);

Ôï ðïíôßêé ôýðïõ PS/2 õðïóôçñßæåôáé åããåíþò. Ôï áðáñáßôçôï ðñüãñáììá ïäÞãçóçò, `psm`, ðåñéëáìâÜíåôáé óôïí ðõñÞíá.

Áí ï ðñïóáñìïóìÝíïò ðõñÞíáò óáò äåí ôïí ðåñéÝ÷åé, ðñïóèÝóôå ôçí áêüëïõèç ãñáììÞ óôï áñ÷åßï ñõèìßóåùí ðõñÞíá, êáé ìåôáãëùôôßóôå îáíÜ ôïí ðõñÞíá óáò.

```
device psm0 at atkbdc? irq 12
```

Ìüëéò ï ðõñÞíáò áíé÷íåýóåé óùóôÜ ôç óõóêåõÞ `psm0` êáôÜ ôçí åêêßíçóç, âåâáéùèåßôå üôé õðÜñ÷åé ç áíôßóôïé÷ç êáôá÷þñçóç ãéá ôï `psm0` óôïí êáôÜëïãï `/dev`. Ìðïñåßôå íá ôï äçìéïõñãÞóåôå ãñÜöïíôáò:

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

üôáí Ý÷åôå åéóÝëèåé ùò ÷ñÞóôçò `root`.

> **Óçìåßùóç:** Ìðïñåßôå íá ðáñáëåßøåôå áõôü ôï âÞìá áí ÷ñçóéìïðïéåßôå FreeBSD 5.0-RELEASE Þ íåþôåñï ìå åíåñãïðïéçìÝíï ôï devfs(5), êáèþò ôá áðáñáßôçôá áñ÷åßá óõóêåõþí èá äçìéïõñãçèïýí áõôüìáôá êÜôù áðü ôïí êáôÜëïãï `/dev`.

4. Åßíáé äõíáôüí íá ÷ñçóéìïðïéçèåß ôï ðïíôßêé ìå êÜðïéï ôñüðï Ýîù áðü ôï ðåñéâÜëëïí ôïõ óõóôÞìáôïò X Window;

Áí ÷ñçóéìïðïéåßôå ôï ðñïåðéëåãìÝíï ðñüãñáììá ïäÞãçóçò êïíóüëáò, syscons(4), ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå ôï äåßêôç ôïõ ðïíôéêéïý óáò óå êïíóüëåò êåéìÝíïõ ãéá íá êÜíåôå áðïêïðÞ êáé åðéêüëëçóç êåéìÝíïõ. ÅêôåëÝóôå ôïí äáßìïíá ôïõ ðïíôéêéïý, moused(8), êáé åíåñãïðïéÞóôå ôï äåßêôç ôïõ ðïíôéêéïý óôçí åéêïíéêÞ êïíóüëá:

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

¼ðïõ ôï *xxxx* åßíáé ôï üíïìá óõóêåõÞò ôïõ ðïíôéêéïý êáé ôï *yyyy* åßíáé ï ôýðïò ôïõ ðñùôïêüëëïõ ôïõ. Ï äáßìïíáò ôïõ ðïíôéêéïý ìðïñåß íá áíáãíùñßóåé áõôüìáôá ôï åßäïò ôïõ ðñùôïêüëëïõ ãéá ôá ðåñéóóüôåñá ðïíôßêéá, åêôüò áðü ðáëéÜ óåéñéáêÜ ìïíôÝëá. Êáèïñßóôå ôï ðñùôüêïëëï `auto` ãéá íá ÷ñçóéìïðïéÞóåôå ôçí áõôüìáôç áíß÷íåõóç. Áí áõôÞ äåí äïõëÝøåé, äåßôå ôç óåëßäá manual moused(8) ãéá ìéá ëßóôá ìå õðïóôçñéæüìåíïõò ôýðïõò ðñùôïêüëëùí.

Áí Ý÷åôå ðïíôßêé ôýðïõ PS/2, áðëþò ðñïóèÝóôå `moused_enable="YES"` óôï áñ÷åßï `/etc/rc.conf` ãéá íá åêêéíÜåé ï äáßìïíáò ôïõ ðïíôéêéïý êáôÜ ôçí åêêßíçóç. Åðéðñüóèåôá, áí åðéèõìåßôå íá ÷ñçóéìïðïéåßôå ôï äáßìïíá ôïõ ðïíôéêéïý óå üëåò ôéò åéêïíéêÝò êïíóüëåò, êáé ü÷é ìüíï óôçí êïíóüëá óõóôÞìáôïò, ðñïóèÝóôå ôç ãñáììÞ `allscreens_flags="-m on"` óôï `/etc/rc.conf`.

¼ôáí åêôåëåßôáé ï äáßìïíáò ôïõ ðïíôéêéïý, ç ðñüóâáóç óôï ðïíôßêé ðñÝðåé íá óõíôïíßæåôáé ìåôáîý ôïõ äáßìïíá êáé Üëëùí ðñïãñáììÜôùí, üðùò ôá X Windows. ÊïéôÜîôå óôï FAQ ôçí åñþôçóç Ãéáôß ôï ðïíôßêé ìïõ äåí äïõëåýåé óôá ×; ãéá ðåñéóóüôåñåò ëåðôïìÝñåéåò ó÷åôéêÜ ìå áõôü ôï ðñüâëçìá.

5. Ðùò ìðïñþ íá êÜíù áðïêïðÞ êáé åðéêüëëçóç êåéìÝíïõ ìå ôï ðïíôßêé óå ìéá êïíóüëá êåéìÝíïõ;

Ìüëéò åíåñãïðïéÞóåôå ôï äáßìïíá ôïõ ðïíôéêéïý (äåßôå ôçí ðñïçãïýìåíç åíüôçôá), êñáôÞóôå ðáôçìÝíï ôï ðëÞêôñï 1 (ôï áñéóôåñü ðëÞêôñï) êáé êéíÞóôå ôï ðïíôßêé ãéá íá åðéëÝîåôå ìéá ðåñéï÷Þ êåéìÝíïõ. Êáôüðéí, ðéÝóôå ôï ðëÞêôñï 2 (ôï ìåóáßï ðëÞêôñï) ãéá íá ôï åðéêïëëÞóåôå óôçí ðåñéï÷Þ ôïõ äñïìÝá. Ìå ôçí ðßåóç ôïõ ðëÞêôñïõ 3 (äåîéïý ðëÞêôñïõ) ìðïñåßôå íá "åðåêôåßíåôå" ôçí åðéëåãìÝíç ðåñéï÷Þ êåéìÝíïõ.

Áí ôï ðïíôßêé óáò äåí Ý÷åé ìåóáßï ðëÞêôñï, ìðïñåß íá èÝëåôå íá ôï åîïìïéþóåôå Þ íá áëëÜîåôå ôéò ëåéôïõñãßåò ôùí ðëÞêôñùí ÷ñçóéìïðïéþíôáò ôéò åðéëïãÝò ôïõ ðáñÝ÷ïíôáé áðü ôïí äáßìïíá ôïõ ðïíôéêéïý Äåßôå ôç óåëßäá manual moused(8) ãéá ôéò ëåðôïìÝñåéåò.

6. Ôï ðïíôßêé ìïõ Ý÷åé äéÜöïñá Ýîõðíá ðëÞêôñá êáé ñïäÝëá êýëéóçò. Ìðïñþ íá ôá ÷ñçóéìïðïéÞóù óôï FreeBSD;

Ç áðÜíôçóç, äõóôõ÷þò, åßíáé "âåáéùò ü÷é". Ôá ðïíôßêéá ìå Ýîôñá äõíáôüôçôåò óõíÞèùò áðáéôïýí åîåéäéêåõìÝíá ðñïãñÜììáôá ïäÞãçóçò. Áí ôï ðñüãñáììá ïäÞãçóçò ôïõ ðïíôéêéïý Þ ôï áíôßóôïé÷ï ðñüãñáììá ôïõ ÷ñÞóôç äåí ðáñÝ÷ïõí óõãêåêñéìÝíç õðïóôÞñéîç ãéá ôï ðïíôßêé, èá ëåéôïõñãåß ùò Ýíá áðëü ðïíôßêé äýï Þ ôñéþí ðëÞêôñùí.

Ãéá ðéèáíÞ ÷ñÞóç ôçò ñïäÝëáò óå ðåñéâÜëëïí X Window, äåßôå ôçí áíôßóôïé÷ç åíüôçôá.

7. Ðùò ìðïñþ íá ÷ñçóéìïðïéÞóù ôï ðïíôßêé / trackball / touchpad óôïí öïñçôü ìïõ õðïëïãéóôÞ;

Äåßôå ôçí áðÜíôçóç óôçí ðñïçãïýìåíç åñþôçóç.

8. Ðùò ìðïñþ íá ÷ñçóéìïðïéÞóù ôï ðëÞêôñï delete óôï `sh` êáé `csh`;

Ãéá ôï **ÊÝëõöïò Bourne**, ðñïóèÝóôå ôéò áêüëïõèåò ãñáììÝò óôï áñ÷åßï óáò `.shrc`. Äåßôå åðßóçò ôéò óåëßäåò manual sh(1) êáé editrc(5).

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

Ãéá ôï **ÊÝëõöïò C**, ðñïóèÝóôå ôéò áêüëïõèåò ãñáììÝò óôï áñ÷åßï óáò `.cshrc`. Äåßôå åðßóçò ôç óåëßäá manual ôïõ csh(1).

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

Ãéá ðåñéóóüôåñåò ðëçñïöïñßåò, äåßôå áõôÞ ôç óåëßäá (http://www.ibb.net/~anne/keyboard.html).

4.5 ÓõóêåõÝò óåéñéáêÞò åðéêïéíùíßáò êáé Äéêôýùóçò

1. Ðïéåò êÜñôåò äéêôýïõ õðïóôçñßæåé ôï FreeBSD;

Ãéá ôçí ðëÞñç ëßóôá, äåßôå ôéò Óçìåéþóåéò Õëéêïý ðïõ ðáñÝ÷ïíôáé ìå êÜèå Ýêäïóç ôïõ FreeBSD.

2. Õðïóôçñßæåé ôï FreeBSD modems ðïõ ëåéôïõñãïýí ìå ôç âïÞèåéá ëïãéóìéêïý üðùò ôá Winmodems;

To FreeBSD õðïóôçñßæåé áñêåôÜ software modems ìå ôçí âïÞèåéá åðéðñüóèåôïõ ëïãéóìéêïý. Ôï port `comms/ltmdm` ðñïóèÝôåé õðïóôÞñéîç ãéá modems ðïõ âáóßæïíôáé óôï äçìïöéëÝò êýêëùìá Lucent LT. Ôï port `comms/mwavem` õðïóôçñßæåé ôï modem ðïõ äéáèÝôïõí ïé öïñçôïß õðïëïãéóôÝò Thinkpad 600 êáé 700 ôçò IBM.

Äåí ìðïñåßôå íá åãêáôáóôÞóåôå ôï FreeBSD ìÝóù software modem. Ôï ëïãéóìéêü áõôü ðñÝðåé íá åãêáôáóôáèåß ìåôÜ ôçí åãêáôÜóôáóç ôïõ FreeBSD.

3. ÕðÜñ÷åé åããåíÝò ðñüãñáììá ïäÞãçóçò ãéá ôéò êÜñôåò Broadcom 43xx;

¼÷é, êáé ìÜëëïí äåí èá õðÜñîåé.

Ç Broadcom áñíåßôáé íá äþóåé äçìüóéá ðëçñïöïñßåò ó÷åôéêÜ ìå ôïí ðñïãñáììáôéóìü ôùí ïëïêëçñùìÝíùí ôçò ðïõ ÷ñçóéìïðïéïýíôáé óå åöáñìïãÝò áóýñìáôùí äéêôýùí, ðéèáíüí åðåéäÞ êáé ôï ôìÞìá ôïõ ðïìðïÝêôç ôçò êÜñôáò åëÝã÷åôáé ìå ôç âïÞèåéá ëïãéóìéêïý. Ãéá íá áðïêôÞóïõí Ýãêñéóç ãéá ôéò êÜñôåò ôïõò áðü ôï FCC, ðñÝðåé íá åîáóöáëßóïõí üôé ïé ôåëéêïß ÷ñÞóôåò äåí èá åßíáé óå èÝóç íá êÜíïõí ñõèìßóåéò üðùò áëëáãÞ ôçò óõ÷íüôçôáò ëåéôïõñãßáò, ôùí ðáñáìÝôñùí äéáìüñöùóçò êáé ôçò éó÷ýïò åêðïìðÞò. ÁëëÜ ÷ùñßò ôéò ðëçñïöïñßåò ðñïãñáììáôéóìïý, åßíáé ó÷åäüí áäýíáôï íá ãñáöåß ðñüãñáììá ïäÞãçóçò.

4. Ðïéåò êÜñôåò ðïëëáðëþí óåéñéáêþí èõñþí õðïóôçñßæïíôáé áðü ôï FreeBSD;

ÕðÜñ÷åé ìéá ëßóôá ãéá áõôÝò óôçí åíüôçôá äéÜöïñùí óõóêåõþí (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html#INSTALL-MISC) ôïõ Åã÷åéñéäßïõ.

Áêüìá öáßíåôáé üôé ëåéôïõñãïýí êáé êÜðïéåò êÜñôåò ðïõ åßíáé áíôéãñáöÝò åðþíõìùí ìïíôÝëùí, åéäéêÜ üóåò õðïóôçñßæïõí üôé åßíáé óõìâáôÝò ìå ôéò áíôßóôïé÷åò ôéò AST.

Äåßôå ôç óåëßäá manual sio(4) ãéá ðåñéóóüôåñåò ðëçñïöïñßåò ó÷åôéêÜ ìå ôç ñýèìéóç ôÝôïéùí êáñôþí.

5. Ðùò ìðïñþ íá åìöáíßóù ôçí ðñïôñïðÞ boot: óå ìéá óåéñéáêÞ êïíóüëá;

1. ÄçìéïõñãÞóôå ðõñÞíá ðïõ íá ðåñéÝ÷åé ôçí åðéëïãÞ `options COMCONSOLE`.
2. ÄçìéïõñãÞóôå ôï `/boot.config` êáé ãñÜøôå ìÝóá óå áõôü ìüíï ôçí åðéëïãÞ `-P`.
3. ÁðïóõíäÝóôå ôï ðëçêôñïëüãéï áðü ôï óýóôçìá.

Äåßôå ôï áñ÷åßï `/usr/src/sys/i386/boot/biosboot/README.serial` ãéá ðåñéóóüôåñåò ðëçñïöïñßåò.

4.6 ÓõóêåõÝò Þ÷ïõ

1. Ðïéåò êÜñôåò Þ÷ïõ õðïóôçñßæïíôáé áðü ôï FreeBSD;

Ôï FreeBSD õðïóôçñßæåé äéÜöïñåò êÜñôåò Þ÷ïõ, óõìðåñéëáìâáíïìÝíùí ôùí SoundBlaster®, SoundBlaster Pro, SoundBlaster 16, Pro Audio Spectrum 16, AdLib, êáé Gravis UltraSound (ãéá ðåñéóóüôåñåò ðëçñïöïñßåò, äåßôå ôéò Ðëçñïöïñßåò ¸êäïóçò ôïõ FreeBSD (http://www.FreeBSD.org/releases/) êáé ôç óåëßäá manual snd(4)). ÕðÜñ÷åé åðßóçò ðåñéïñéóìÝíç õðïóôÞñéîç ãéá êÜñôåò MIDI ðïõ åßíáé óõìâáôÝò ìå ôï ðñüôõðï MPU-401. Åðßóçò õðïóôçñßæïíôáé ïé êÜñôåò ðïõ åßíáé óõìâáôÝò ìå ôï ðñüôõðï Microsoft® Sound System.

> **Óçìåßùóç:** Ôï ðáñáðÜíù éó÷ýåé ìüíï ãéá ôïí Þ÷ï! Ôï ðñüãñáììá ïäÞãçóçò äåí õðïóôçñßæåé ôõ÷üí CDROM, SCSI Þ joysticks ðïõ óõíäÝïíôáé ðÜíù óå áõôÝò ôéò êÜñôåò, åêôüò áðü ôçí SoundBlaster. Áí êáé ç äéåðáöÞ SCSI ôçò SoundBlaster êáèþò êáé êÜðïéá ìç-SCSI CDROM õðïóôçñßæïíôáé, äåí ìðïñïýí ùóôüóï íá ÷ñçóéìïðïéçèïýí ãéá ôç äéáäéêáóßá åêêßíçóçò.

2. ÕðÜñ÷åé êÜðïéá ëýóç ãéá ôï ðñüâëçìá ôïõ Þ÷ïõ óôçí êÜñôá ìïõ ðïõ õðïóôçñßæåôáé áðü ôï pcm(4);

ÊÜðïéåò êÜñôåò Þ÷ïõ, üðùò ç es1370, ìçäåíßæïõí ôçí Ýíôáóç ôïõ Þ÷ïõ óå êÜèå åêêßíçóç. ÐñÝðåé íá åêôåëåßôå ôçí áêüëïõèç åíôïëÞ êÜèå öïñÜ ðïõ îåêéíÜ ôï ìç÷Üíçìá:

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

4.7 ¶ëëï õëéêü

1. Ðïéåò Üëëåò óõóêåõÝò õðïóôçñßæïíôáé áðü ôï FreeBSD;

Äåßôå ôï Åã÷åéñßäéï (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html#INSTALL-MISC) ãéá ôç ëßóôá ôùí õðüëïéðùí óõóêåõþí ðïõ õðïóôçñßæïíôáé.

2. Õðïóôçñßæåé ôï FreeBSD äéá÷åßñéóç åíÝñãåéáò ãéá ôï öïñçôü ìïõ õðïëïãéóôÞ;

Áðü ôï FreeBSD 4.X êáé ìåôÜ, õðïóôçñßæåôáé ôï APM óå óõãêåêñéìÝíá ìç÷áíÞìáôá. Ðåñéóóüôåñåò ðëçñïöïñßåò ìðïñåßôå íá âñåßôå óôï apm(4).

Áðü ôï FreeBSD 5.X êáé ìåôÜ, õðïóôçñßæåôáé ç äõíáôüôçôá ACPI ç ïðïßá õðÜñ÷åé óå üëïõò ôïõò óýã÷ñïíïõò õðïëïãéóôÝò. Ìðïñåßôå íá âñåßôå ðåñéóóüôåñåò ðëçñïöïñßåò óôï acpi(4). Áí Ýíá óýóôçìá õðïóôçñßæåé ôüóï APM üóï êáé ACPI, ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå üðïéï èÝëåôå. Óáò óõíéóôïýìå íá äïêéìÜóåôå êáé ôá äýï êáé íá åðéëÝîåôå áõôü ðïõ êáëýðôåé êáëýôåñá ôéò áíÜãêåò óáò.

3. Ðùò ìðïñþ íá áðåíåñãïðïéÞóù ôï ACPI;

ÐñïóèÝóôå ôç ãñáììÞ

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

óôï áñ÷åßï `/boot/device.hints`.

4. Ãéáôß ôï Micron óýóôçìá ìïõ êñåìÜåé êáôÜ ôçí åêêßíçóç;

ÏñéóìÝíåò ìçôñéêÝò Micron õëïðïéïýí ôï PCI BIOS ìå ìç ôõðïðïéçìÝíï ôñüðï, ðñïêáëþíôáò ðñïâëÞìáôá óôçí åêêßíçóç ôïõ FreeBSD, êáèþò ïé PCI óõóêåõÝò äåí ñõèìßæïíôáé óôéò äéåõèýíóåéò ðïõ áíáöÝñïíôáé.

Ãéá íá ðáñáêÜìøåôå ôï ðñüâëçìá, áðåíåñãïðïéÞóôå ôçí åðéëïãÞ “Plug and Play Operating System” áðü ôï BIOS.

5. Ç äéóêÝôá åêêßíçóçò êñåìÜåé óôç ìçôñéêÞ ASUS K7V. Ðùò ìðïñþ íá ôï äéïñèþóù áõôü;

Óôéò ñõèìßóåéò ôïõ BIOS, áðåíåñãïðïéÞóôå ôçí åðéëïãÞ “boot virus protection”.

6. Ãéáôß ç PCI êÜñôá äéêôýïõ ìïõ ôçò 3Com® äåí ëåéôïõñãåß ìå ôï Micron õðïëïãéóôÞ ìïõ;

ÏñéóìÝíåò ìçôñéêÝò ôçò Micron Ý÷ïõí ìç-ôõðïðïéçìÝíï PCI BIOS ôï ïðïßï äåí ñõèìßæåé ôéò óõóêåõÝò PCI óôéò äéåõèýíóåéò ðïõ áíáöÝñïíôáé. Áõôü äçìéïõñãåß ðñïâëÞìáôá êáôÜ ôçí åêêßíçóç ôïõ FreeBSD.

Ãéá íá ðáñáêÜìøåôå ôï ðñüâëçìá, áðåíåñãïðïéÞóôå ôçí åðéëïãÞ “Plug and Play Operating System” áðü ôï BIOS.

7. Ç PCMCIA êÜñôá ìïõ äåí ëåéôïõñãåß. ÂëÝðù ôï åîÞò ìÞíõìá: “cbb0: unsupported card type detected.” Ôé ìðïñþ íá êÜíù;

Ìðïñåßôå íá äïêéìÜóåôå íá ÷ñçóéìïðïéÞóåôå ôçí áñ÷éêÞ õëïðïßçóç OLDCARD. ÔñïðïðïéÞóôå ôï áñ÷åßï ñýèìéóçò ôïõ ðõñÞíá óáò, êáé áöáéñÝóôå ôéò áêüëïõèåò ãñáììÝò:

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

ÌåôÜ ðñïóèÝóôå:

```
device pcic
device card 1
```

Ìåôáãëùôôßóôå îáíÜ êáé åãêáôáóôÞóôå ôï íÝï ðõñÞíá üðùò ðåñéãñÜöåôáé óôçí Ñýèìéóç ÐõñÞíá ôïõ FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/kernelconfig.html).

ÊåöÜëáéï 5 Áíôéìåôþðéóç ÐñïâëçìÜôùí

1. Ãéáôß ôï FreeBSD âñßóêåé ëÜèïò ðïóüôçôá ìíÞìçò;

Áõôü ïöåßëåôáé óôç äéáöïñÜ ìåôáîý öõóéêþí êáé åéêïíéêþí äéåõèýíóåùí ìíÞìçò.

Ç óýìâáóç ðïõ êáôÜ âÜóç áêïëïõèåßôáé óôï ðëåßóôï ôùí PC, åßíáé íá ÷ñçóéìïðïéåßôáé ç ìíÞìç ìåôáîý 3.5G êáé 4G ãéá åéäéêü óêïðü, óõíÞèùò ãéá ôçí ðñüóâáóç óå êÜñôåò PCI. Áõôü Ý÷åé ùò áðïôÝëåóìá íá ìçí ìðïñåß íá áíôéóôïé÷çèåß öõóéêÞ ìíÞìç óå áõôÞ ôçí ðåñéï÷Þ äéåõèýíóåùí.

Ôï õëéêü ôïõ õðïëïãéóôÞ óáò èá êáèïñßóåé ôé ãßíåôáé ìå ôçí ìíÞìç ðïõ êáíïíéêÜ åìöáíßæåôáé óå áõôÞ ôç èÝóç. Äõóôõ÷þò, óå êÜðïéåò ðåñéðôþóåéò ôï õëéêü äåí êÜíåé ôßðïôá, êáé ÷Üíåôáé ç äõíáôüôçôá ÷ñÞóçò ôùí ôåëåõôáßùí 500Ì ìíÞìçò RAM.

Åõôõ÷þò, óôéò ðåñéóóüôåñåò ðåñéðôþóåéò ôï õëéêü áíáêáôåõèýíåé ôç ìíÞìç óå øçëüôåñç èÝóç, þóôå íá åßíáé áêüìá äõíáôÞ ç ÷ñÞóç ôçò. Áõôü ìðïñåß ùóôüóï íá óáò ðñïêáëÝóåé êÜðïéá óýã÷õóç áí ðáñáêïëïõèåßôå ôá ìçíýìáôá åêêßíçóçò.

Óôçí 32 bit Ýêäïóç ôïõ FreeBSD, ç ìíÞìç öáßíåôáé íá Ý÷åé ÷áèåß êáèþò áíáêáôåõèýíåôáé ðÜíù áðü ôá 4G, ôá ïðïßá äåí åßíáé ðñïóâÜóéìá áðü 32 bit ðõñÞíá. Óôçí ðåñßðôùóç áõôÞ ç ëýóç åßíáé íá öôéÜîåôå Ýíá ðõñÞíá ôýðïõ PAE. Äåßôå áõôÞí ôçí êáôá÷þñçóç óôï FAQ ãéá ðåñéóóüôåñåò ðëçñïöïñßåò.

Óôçí 64 bit Ýêäïóç ôïõ FreeBSD, Þ üôáí ÷ñçóéìïðïéåßôáé ðõñÞíáò ôýðïõ PAE, ôï FreeBSD èá áíé÷íåýóåé êáé èá áíáêáôåõèýíåé óùóôÜ ôç ìíÞìç þóôå íá åßíáé ÷ñçóéìïðïéÞóéìç. ÊáôÜ ôçí åêêßíçóç ùóôüóï, ìðïñåß íá öáßíåôáé üôé ôï FreeBSD áíé÷íåýåé ðåñéóóüôåñç ìíÞìç áðü áõôÞ ðïõ Ý÷åé óôçí ðñáãìáôéêüôçôá ôï óýóôçìá. Áõôü åßíáé öõóéïëïãéêü êáé ç äéáèÝóéìç ìíÞìç èá äéïñèùèåß êáèþò ïëïêëçñþíåôáé ç äéáäéêáóßá ôçò åêêßíçóçò.

2. Ï óêëçñüò ìïõ äßóêïò Ý÷åé ÷áëáóìÝíïõò ôïìåßò. Ôé ìðïñþ íá êÜíù;

Óôïõò äßóêïõò SCSI, ï ïäçãüò ìðïñåß óõíÞèùò íá åðáíáôïðïèåôÞóåé áõôüìáôá ôá äåäïìÝíá óå åíáëëáêôéêïýò ôïìåßò. Ùóôüóï ïé ðåñéóóüôåñïé äßóêïé Ýñ÷ïíôáé ìå ôçí äõíáôüôçôá áõôÞ áðåíåñãïðïéçìÝíç.

Ãéá íá åíåñãïðïéÞóåôå ôçí åðáíáôïðïèÝôçóç ÷áëáóìÝíùí ôïìÝùí, åðåîåñãáóôåßôå ôçí ðñþôç óåëßäá êáôÜóôáóçò ôçò óõóêåõÞò (modepage), äßíïíôáò ôçí ðáñáêÜôù åíôïëÞ (ùò `root`):

```
# camcontrol modepage sd0 -m 1 -e -P 3
```

êáé áëëÜîôå ôéò ôéìÝò ôùí AWRE êáé ARRE áðü 0 óå 1:

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

Ïé óýã÷ñïíïé ïäçãïß ôýðïõ IDE Ý÷ïõí åðßóçò åíåñãïðïéçìÝíç áðü ôï åñãïóôÜóéï ôç äõíáôüôçôá åðáíáôïðïèÝôçóçò ÷áëáóìÝíùí ôïìÝùí.

Áí äåßôå ðñïåéäïðïéÞóåéò ó÷åôéêÜ ìå ÷áëáóìÝíïõò ôïìåßò (óå ïðïéïäÞðïôå åßäïò äßóêïõ), åßíáé þñá íá óêåöôåßôå íá áëëÜîåôå ôïí ïäçãü. ºóùò ìðïñÝóåôå íá ÷ñçóéìïðïéÞóåôå ôï äéáãíùóôéêü ðñüãñáììá ðïõ äßíåé ï êáôáóêåõáóôÞò ôïõ äßóêïõ ãéá íá áðïìïíþóåôå ôïõò ÷áëáóìÝíïõò ôïìåßò, áëëÜ óôçí êáëýôåñç ðåñßðôùóç áðëþò èá êåñäßóåôå ëßãï ðåñéóóüôåñï ÷ñüíï.

3. Ãéáôé ôï FreeBSD äåí áíé÷íåýåé ôïí åëåãêôÞ SCSI óôïí HP Netserver;

Ôï ðñüâëçìá áõôü åßíáé ãíùóôü. Ï åíóùìáôùìÝíïò óôç ìçôñéêÞ åëåãêôÞò SCSI ôïõ HP Netserver, ÷ñçóéìïðïéåß óýíäåóç ôýðïõ EISA êáé êáôáëáìâÜíåé ôç èÝóç EISA ìå áñéèìü 11. Ìå ôïí ôñüðï áõôü, üëåò ïé “ðñáãìáôéêÝò” õðïäï÷Ýò ôýðïõ EISA âñßóêïíôáé ðñéí áðü áõôÞ. Ùóôüóï, ç ðåñéï÷Þ äéåõèýíóåùí ôùí õðïäï÷þí EISA ìå áñéèìü >= 10, óõãêñïýåôáé ìå ôçí ðåñéï÷Þ äéåõèýíóåùí ôïõ PCI, êáé ôï FreeBSD óôç óçìåñéíÞ ôïõ ìïñöÞ, äåí ìðïñåß íá ÷åéñéóôåß óùóôÜ áõôÞ ôçí êáôÜóôáóç.

¸ôóé, ãéá ôçí þñá, ôï êáëýôåñï ðïõ ìðïñåßôå íá êÜíåôå åßíáé íá ðáñéóôÜíåôå üôé äåí õðÜñ÷åé óýãêñïõóç äéåõèýíóåùí :) êáé íá áíåâÜóåôå ôçí åðéëïãÞ `EISA_SLOTS` ôïõ ðõñÞíá óôçí ôéìÞ 12. Ìåôáãëùôôßóôå Ýðåéôá îáíÜ ôïí ðõñÞíá, üðùò ðåñéãñÜöåôáé óôçí ó÷åôéêÞ êáôá÷þñçóç ôïõ Åã÷åéñéäßïõ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/kernelconfig.html).

ÖõóéêÜ áõôü åßíáé Ýíá ðñüâëçìá áíôßóôïé÷ï ìå ôï áõãü êáé ôçí êüôá, üóï áöïñÜ ôçí åãêáôÜóôáóç åíüò ôÝôïéïõ ìç÷áíÞìáôïò. Ãéá íá ðñïóðåñÜóåôå ôï ðñüâëçìá, õðÜñ÷åé åéäéêÞ ðñüâëåøç óôï *UserConfig*. Ìç ÷ñçóéìïðïéÞóåôå ôï “visual” interface, áëëÜ ôçí ãñáììÞ åíôïëþí. Áðëþò ãñÜøôå:

```
eisa 12
quit
```

óôçí ðñïôñïðÞ, êáé åãêáôáóôÞóôå ôï óýóôçìá óáò üðùò óõíÞèùò. Óáò óõíéóôïýìå ùóôüóï íá ìåôáãëùôôßóåôå êáé íá åãêáôáóôÞóåôå ôï äéêü óáò ðñïóáñìïóìÝíï ðõñÞíá.

Åõåëðéóôïýìå üôé óå ìåëëïíôéêÝò åêäüóåéò, èá õðÜñ÷åé êáëýôåñç äéüñèùóç ãéá ôï ðñüâëçìá áõôü.

> **Óçìåßùóç:** Äåí ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå äßóêï óå êáôÜóôáóç `dangerously dedicated (åðéêßíäõíá áöïóéùìÝíç)` ìå ôïí HP Netserver. Äåßôå áõôÞ ôç óçìåßùóç ãéá ðåñéóóüôåñåò ðëçñïöïñßåò.

4. ÂëÝðù óõíÝ÷åéá ìçíýìáôá ôïõ ôýðïõ “`ed1: timeout`”. Ôé óçìáßíïõí;

Ôá ìçíýìáôá áõôÜ ðñïêáëïýíôáé óõíÞèùò áðü äéåíÝîåéò óôá interrupts (ð.÷. äýï êÜñôåò ðïõ ÷ñçóéìïðïéïýí ôï ßäéï IRQ). ÅêêéíÞóôå ìå ôçí åðéëïãÞ -c êáé áëëÜîôå ôçí êáôá÷þñçóç ed0/de0/... þóôå íá óõìâáäßæåé ìå ôï õëéêü óáò.

Áí ÷ñçóéìïðïéåßôå ôçí óýíäåóç BNC ôçò êÜñôáò äéêôýïõ óáò, ßóùò íá äåßôå åðßóçò áíôßóôïé÷á ìçíýìáôá óå ðåñßðôùóç ðñïâëçìáôéêïý ôåñìáôéóìïý. Ãéá íá åëÝãîåôå ôçí ðåñßðôùóç áõôÞ, óõíäÝóôå Ýíá ôåñìáôéóôÞ áðåõèåßáò óôçí êÜñôá (÷ùñßò êáëþäéï) êáé äåßôå áí óôáìáôÞóïõí ôá ìçíýìáôá.

ÊÜðïéåò êÜñôåò óõìâáôÝò ìå NE2000, äßíïõí áõôü ôï ìÞíõìá áí äåí õðÜñ÷åé óýíäåóç óôç èýñá UTP Þ áí ôï êáëþäéï åßíáé áðïóõíäåìÝíï.

5. Ãéáôß óôáìÜôçóå íá ëåéôïõñãåß ç êÜñôá ìïõ 3Com 3C509 ÷ùñßò íá õðÜñ÷åé åìöáíÞò ëüãïò;

Ç êÜñôá áõôÞ Ý÷åé ôçí êáêÞ óõíÞèåéá íá ÷Üíåé ôéò ñõèìßóåéò ôçò. Áíáíåþóôå ôéò, ÷ñçóéìïðïéþíôáò ôï âïçèçôéêü ðñüãñáììá DOS `3c5x9.exe`.

6. Ï åêôõðùôÞò ìïõ óôçí ðáñÜëëçëç èýñá åßíáé áðåëðéóôéêÜ áñãüò. Ôé ìðïñþ íá êÜíù;

Áí ôï ìüíï ðñüâëçìá åßíáé ï õðåñâïëéêÜ áñãüò åêôõðùôÞò, ìðïñåßôå íá äïêéìÜóåôå íá áëëÜîåôå ôçí êáôÜóôáóç ëåéôïõñãßáò ôçò ðáñÜëëçëçò èýñáò (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/printing-intro-setup.html#PRINTING-PARALLEL-PORT-MODE) üðùò ðåñéãñÜöåôáé óôï êåöÜëáéï ôïõ Åã÷åéñéäßïõ

ó÷åôéêÜ ìå ôçí ÅãêáôÜóôáóç ÅêôõðùôÞ
(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/printing-intro-setup.html).

7. Ãéáôß ôá ðñïãñÜììáôá ìïõ ðåñéóôáóéáêÜ ôåñìáôßæïõí ìå óöÜëìá “`Signal 11`”;

Ôá óöÜëìáôá ôýðïõ Signal 11 äçìéïõñãïýíôáé üôáí ìéá äéåñãáóßá ðñïóðáèåß íá ðñïóðåëÜóåé ðåñéï÷Þ ìíÞìçò ãéá ôçí ïðïßá äåí Ý÷åé ðÜñåé Üäåéá áðü ôï ëåéôïõñãéêü óýóôçìá. Áí óõìâáßíåé êÜôé ôÝôïéï óå öáéíïìåíéêÜ ôõ÷áßá ÷ñïíéêÜ äéáóôÞìáôá, èá ðñÝðåé íá áñ÷ßóåôå íá ôï åñåõíÜôå ðïëý ðñïóåêôéêÜ.

Ôá ðñïâëÞìáôá áõôÜ óõíÞèùò ïöåßëïíôáé óå êÜðïéï áðü ôïõò ðáñáêÜôù ëüãïõò:

1. Áí ôï ðñüâëçìá åìöáíßæåôáé ìüíï óå ìéá óõãêåêñéìÝíç åöáñìïãÞ ôçí ïðïßá áíáðôýóóåôå åóåßò, åßíáé ðéèáíþò ëÜèïò óôïí äéêü óáò êþäéêá.
2. Áí ôï ðñüâëçìá âñßóêåôáé óå ôìÞìá ôïõ âáóéêïý óõóôÞìáôïò ôïõ FreeBSD, ìðïñåß åðßóçò íá åßíáé ðñïâëçìáôéêüò êþäéêáò, áëëÜ ôéò ðåñéóóüôåñåò öïñÝò, ôá ðñïâëÞìáôá áõôÜ âñßóêïíôáé êáé äéïñèþíïíôáé ðñéí äéáíåìçèïýí óôïõò ðåñéóóüôåñïõò áðü åóÜò ðïõ äéáâÜæåôå ôï FAQ (ãéá ôï ëüãï áõôü Üëëùóôå õðÜñ÷åé êáé ç ãñáììÞ áíÜðôõîçò -current).

Ãéá ðáñÜäåéãìá, Ýíáò ãñÞãïñïò ôñüðïò íá äéáðéóôþóåôå üôé *äåí* ðñüêåéôáé ãéá ðñüâëçìá ôïõ FreeBSD, åßíáé áí ôï ðñüâëçìá åìöáíßæåôáé êáôÜ ôç ìåôáãëþôôéóç êÜðïéïõ ðñïãñÜììáôïò, áëëÜ êÜèå öïñÜ êáé óå äéáöïñåôéêü óçìåßï.

Ãéá ðáñÜäåéãìá, õðïèÝóôå üôé åêôåëåßôå Ýíá “make buildworld”, êáé ç ìåôáãëþôôéóç áðïôõã÷Üíåé êáôÜ ôçí åðåîåñãáóßá ôïõ áñ÷åßïõ `ls.c` óå `ls.o`. Áí åêôåëÝóåôå îáíÜ “make buildworld”, êáé ç ìåôáãëþôôéóç óôáìáôÞóåé óôï ßäéï óçìåßï, ðñüêåéôáé ðñÜãìáôé ãéá ðñüâëçìá óôá áñ÷åßá ôïõ build -- äïêéìÜóôå íá áíáíåþóåôå ôïí ðçãáßï êþäéêá êáé íá îáíáðñïóðáèÞóåôå. Áí ç ìåôáãëþôôéóç áðïôõã÷Üíåé áëëïý, áõôü ó÷åäüí óßãïõñá ïöåßëåôáé óå ðñïâëçìáôéêü õëéêü.

Ôé ðñÝðåé íá êÜíåôå:

Óôçí ðñþôç ðåñßðôùóç ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå êÜðïéï debugger üðùò ôï gdb ãéá íá âñåßôå ôï óçìåßï óôï ðñüãñáììá ìå ôçí ðñïâëçìáôéêÞ äéåýèõíóç êáé íá ôï äéïñèþóåôå.

Óôç äåýôåñç ðåñßðôùóç, èá ðñÝðåé íá åðáëçèåýóåôå üôé äåí öôáßåé ôï õëéêü óáò.

Óôéò óõíçèéóìÝíåò áéôßåò áõôïý ôïõ ðñïâëÞìáôïò, ðåñéëáìâÜíïíôáé:

1. Ïé óêëçñïß óáò äßóêïé ìðïñåß íá õðåñèåñìáßíïíôáé. ÅëÝãîôå üôé ëåéôïõñãïýí ïé áíåìéóôÞñåò óôï êïõôß óáò. Áí äåí ëåéôïõñãïýí, åßíáé ðéèáíü ïé äßóêïé óáò (êáé ßóùò êáé Üëëá åîáñôÞìáôá) íá õðåñèåñìáßíïíôáé.
2. Ï åðåîåñãáóôÞò óáò Ý÷åé õðåñèåñìáíèåß: Áõôü ìðïñåß íá óõìâåß óå ðåñßðôùóç ðïõ ôïí ëåéôïõñãåßôå óå ìåãáëýôåñç óõ÷íüôçôá áðü ôçí êáíïíéêÞ (overclocking) Þ áí ôï áíåìéóôçñÜêé ôïõ åðåîåñãáóôÞ Ý÷åé óôáìáôÞóåé íá ëåéôïõñãåß. Óå êÜèå ðåñßðôùóç, èá ðñÝðåé íá åîáóöáëßóåôå üôé ÷ñçóéìïðïéåßôå ôï õëéêü óáò óýìöùíá ìå ôéò ðñïäéáãñáöÝò ôïõ, ôïõëÜ÷éóôïí ãéá üóï äéÜóôçìá ÷ñåéÜæåôáé ãéá íá åðéëýóåôå ôï ðñüâëçìá. Ãéá ðáñÜäåéãìá, áí Ý÷åôå êÜíåé overclocking, åðéóôñÝøôå ôïí åðåîåñãáóôÞ óôçí êáíïíéêÞ ôïõ óõ÷íüôçôá.

   Ó÷åôéêÜ ìå ôï overclocking, óçìåéþóôå åðßóçò üôé åßíáé öôçíüôåñï íá Ý÷åôå Ýíá ðéï áñãü óýóôçìá áðü Ýíá êáôåóôñáììÝíï ðïõ ÷ñåéÜæåôáé áíôéêáôÜóôáóç! Åðßóçò ç êïéíüôçôá ãåíéêÜ äåí èá óáò áíôéìåôùðßóåé ìå êáôáíüçóç áí áíáöÝñåôå ðñïâëÞìáôá ðïõ ðáñïõóéÜæïíôáé óå óõóôÞìáôá ðïõ ëåéôïõñãïýí åêôüò ðñïäéáãñáöþí, åßôå åóåßò ðéóôåýåôå üôé ç ëåéôïõñãßá ôïõò åßíáé áóöáëÞò, åßôå ü÷é.
3. ÐñïâëçìáôéêÞ ìíÞìç: Áí Ý÷åôå åãêáôåóôçìÝíá ðåñéóóüôåñá áðü Ýíá SIMMS / DIMMS, áöáéñÝóôå ôá êáé ðñïóðáèÞóôå íá ëåéôïõñãÞóåôå ôï ìç÷Üíçìá ìå Ýíá-Ýíá ÷ùñéóôÜ þóôå íá åíôïðßóåôå ôï ðñüâëçìá óå åðßðåäï åíüò SIMM / DIMM, Þ ßóùò óå Ýíá óõíäõáóìü ôïõò.

4. Õðåñ-áéóéüäïîåò ñõèìßóåéò ìçôñéêÞò: Óôéò ñõèìßóåéò ôïõ BIOS, êáé óå êÜðïéåò ðåñéðôþóåéò óå ñõèìßóåéò óôç ìçôñéêÞ ìÝóù âñá÷õêõêëùôÞñùí (jumpers), õðÜñ÷åé ç äõíáôüôçôá ìåôáâïëÞò äéÜöïñùí ÷ñïíéóìþí. Óôéò ðåñéóóüôåñåò ðåñéðôþóåéò ïé ðñïåðéëåãìÝíåò ñõèìßóåéò åßíáé åðáñêåßò, êáé ßóùò äçìéïõñãÞóåôå ðñïâëÞìáôá áí ñõèìßóåôå ðïëý ÷áìçëÜ ôéò êáèõóôåñÞóåéò áíáìïíÞò (wait states) ôçò RAM Þ èÝóåôå óôï BIOS ôçí åðéëïãÞ "RAM Speed: Turbo". Ìéá êáëÞ éäÝá åßíáé íá åðéóôñÝøåôå ôéò ñõèìßóåéò ôïõ BIOS óôéò ðñïåðéëåãìÝíåò, áëëÜ ðñéí ôï êÜíåôå, óçìåéþóôå êÜðïõ ôéò áëëáãÝò óáò.
5. ÁíåðáñêÞò Þ êáêÞò ðïéüôçôáò ôñïöïäïôéêü óôç ìçôñéêÞ. Áí Ý÷åôå êÜñôåò I/O, óêëçñïýò äßóêïõò Þ CDROM óôï óýóôçìá óáò ðïõ äåí ÷ñçóéìïðïéåßôå, äïêéìÜóôå íá ôá áöáéñÝóåôå Þ íá áðïóõíäÝóåôå ðñïóùñéíÜ ôçí ðáñï÷Þ ôñïöïäïóßáò ôïõò, ãéá íá äéáðéóôþóåôå áí ôï ôñïöïäïôéêü óáò ìðïñåß íá äéá÷åéñéóôåß ìéêñüôåñï öïñôßï. [1] áðëþò äïêéìÜóôå Ýíá Üëëï ôñïöïäïôéêü, êáôÜ ðñïôßìçóç Ýíá ìå ëßãï ìåãáëýôåñç éó÷ý (ãéá ðáñÜäåéãìá áí ôï ôñÝ÷ïí óáò ôñïöïäïôéêü åßíáé ïíïìáóôéêÞò éó÷ýïò 250W, äïêéìÜóôå Ýíá éó÷ýïò 300W).

Èá ðñÝðåé åðßóçò íá äéáâÜóåôå ôï SIG11 FAQ (ôï ïðïßï öáßíåôáé ðáñáêÜôù) ôï ïðïßï ðåñéëáìâÜíåé åîáéñåôéêÝò åðåîçãÞóåéò ãéá üëá áõôÜ ôá ðñïâëÞìáôá, áí êáé ðïëëÝò áðü áõôÝò åßíáé ãñáììÝíåò áðü ôçí óêïðéÜ ôïõ Linux®. ¸íá åíäéáöÝñïí ôìÞìá ôïõ SIG11 FAQ åßíáé êáé áõôü ðïõ áíáöÝñåôáé óôçí ðéèáíüôçôá íá ìçí áíé÷íåýåôáé ðñïâëçìáôéêÞ ìíÞìç áðü äéáãíùóôéêÜ ðñïãñÜììáôá Þ óõóêåõÝò åëÝã÷ïõ.

ÔÝëïò, áí ôßðïôá áðü ôá ðáñáðÜíù äåí âïçèÞóåé, åßíáé ðéèáíüí íá Ý÷åôå åíôïðßóåé Ýíá ðñüâëçìá (bug) óôï FreeBSD êáé èá ðñÝðåé íá áêïëïõèÞóåôå ôéò ïäçãßåò ãéá íá óôåßëåôå áíáöïñÜ ðñïâëÞìáôïò.

Ìðïñåßôå íá âñåßôå åêôåôáìÝíç áíÜëõóç óôï FAQ ó÷åôéêÜ ìå ôï ðñüâëçìá SIG11. (http://www.bitwizard.nl/sig11/)

8. Ôï óýóôçìá ìïõ óôáìáôÜåé åßôå ìå "`Fatal trap 12: page fault in kernel mode`", Þ ìå "`panic:`", äåß÷íïíôáò êáé ìéá óåéñÜ áðü ðëçñïöïñßåò. Ôé ðñÝðåé íá êÜíù;

Ç ïìÜäá áíÜðôõîçò ôïõ FreeBSD åíäéáöÝñåôáé éäéáßôåñá ãéá áõôÜ ôá ëÜèç, áëëÜ ÷ñåéÜæåôáé ðåñéóóüôåñåò ðëçñïöïñßåò åêôüò áðü ôï ìÞíõìá ëÜèïõò ðïõ âëÝðåôå. ÁíôéãñÜøôå ôï ðëÞñåò ìÞíõìá êáé Ýðåéôá óõìâïõëåõèåßôå ôçí åíüôçôá ôïõ FAQ ó÷åôéêÜ ìå ôá kernel panics, äçìéïõñãÞóôå Ýíá ðõñÞíá ìå äõíáôüôçôá áðïóöáëìÜôùóçò (debugging kernel) êáé åêôåëÝóôå Ýíá backtrace. Áõôü ìðïñåß íá áêïýãåôáé äýóêïëï, áëëÜ äåí ÷ñåéÜæåóôå óôçí ðñáãìáôéêüôçôá ãíþóåéò ðñïãñáììáôéóìïý. Áñêåß íá áêïëïõèÞóåôå ôéò ïäçãßåò.

9. Ãéáôß ç ïèüíç ìïõ ìáõñßæåé êáé ÷Üíåé ôï óõã÷ñïíéóìü ôçò êáôÜ ôçí åêêßíçóç;

Ðñüêåéôáé ãéá ãíùóôü ðñüâëçìá ìå ôçí êÜñôá ãñáöéêþí ATI Mach64. Ôï ðñüâëçìá åßíáé üôé ç êÜñôá áõôÞ ÷ñçóéìïðïéåß ôçí äéåýèõíóç `2e8`, ç ïðïßá ÷ñçóéìïðïéåßôáé åðßóçò êáé áðü ôçí ôÝôáñôç óåéñéáêÞ èýñá. Ëüãù êÜðïéïõ ðñïâëÞìáôïò (Þ ôçò ó÷åäßáóçò) ôïõ ðñïãñÜììáôïò ïäÞãçóçò sio(4), ôï ðñüãñáììá ü÷é ìüíï èá ðñïóðáèÞóåé íá áíé÷íåýóåé áõôÞ ôç äéåýèõíóç áêüìá êáé áí äåí Ý÷åôå ôÝôáñôç óåéñéáêÞ èýñá, áëëÜ *áêüìá* êáé óôçí ðåñßðôùóç ðïõ Ý÷åôå áðåíåñãïðïéÞóåé ôç óåéñéáêÞ èýñá sio3 (äçë. ôçí ôÝôáñôç) ç ïðïßá öõóéïëïãéêÜ ÷ñçóéìïðïéåß áõôÞ ôç äéåýèõíóç.

ÌÝ÷ñé íá äéïñèùèåß ôï ðñüâëçìá áõôü, ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå ôï ðáñáêÜôù ôÝ÷íáóìá ãéá íá ôï ðáñáêÜìøåôå:

1. ÃñÜøôå `-c` óôçí ðñïôñïðÞ åêêßíçóçò. (Ìå ôïí ôñüðï áõôü èá âÜëåôå ôïí ðõñÞíá óå êáôÜóôáóç ñýèìéóçò).
2. ÁðåíåñãïðïéÞóôå ôéò `sio0`, `sio1`, `sio2` êáé `sio3` (üëåò). Ìå ôïí ôñüðï áõôü ôï ðñüãñáììá ïäÞãçóçò äåí åíåñãïðïéåßôáé êáí, Üñá äåí äçìéïõñãåßôáé ðñüâëçìá.
3. ÃñÜøôå exit ãéá íá óõíå÷ßóåôå ôçí åêêßíçóç.

Áí èÝëåôå íá ÷ñçóéìïðïéÞóåôå ôéò óåéñéáêÝò èýñåò, èá ðñÝðåé íá äçìéïõñãÞóåôå íÝï ðõñÞíá, ìå ôçí áêüëïõèç ìåôáôñïðÞ: Óôï áñ÷åßï `/usr/src/sys/i386/isa/sio.c` âñåßôå ôï ðñþôï óçìåßï ðïõ åìöáíßæåôáé ôï áëöáñéèìçôéêü `0x2e8` êáé áöáéñÝóôå áõôü ôï áëöáñéèìçôéêü êáé ôï êüììá ðïõ âñßóêåôáé ðñéí áðü áõôü (êñáôÞóôå ôï êüììá ðïõ âñßóêåôáé ìåôÜ). ÁêïëïõèÞóôå ôþñá ôç óõíçèéóìÝíç äéáäéêáóßá äçìéïõñãßáò íÝïõ ðõñÞíá.

Áêüìá êáé ìåôÜ ôçí åöáñìïãÞ áõôþí ôùí äéïñèþóåùí, ßóùò áíáêáëýøåôå üôé ôï óýóôçìá X Window äåí ëåéôïõñãåß óùóôÜ. Áí óõìâáßíåé áõôü, âåâáéùèåßôå üôé ÷ñçóéìïðïéåßôå Ýêäïóç 3.3.3 Þ ìåãáëýôåñç ôïõ XFree86™. Áðü ôçí Ýêäïóç áõôÞ êáé ìåôÜ, õðÜñ÷åé åíóùìáôùìÝíç õðïóôÞñéîç ãéá êÜñôåò Mach64 êáé åðßóçò äéáôßèåôáé åîåéäéêåõìÝíïò åîõðçñåôçôÞò X ãéá ôçí êÜñôá áõôÞ.

10. Ãéáôß ôï FreeBSD óýóôçìá ìïõ ÷ñçóéìïðïéåß ìüíï 64MB RAM, åíþ ï õðïëïãéóôÞò ìïõ Ý÷åé åãêáôåóôçìÝíá 128MB;

Åîáéôßáò ôïõ ôñüðïõ ìå ôïí ïðïßï ôï FreeBSD äéáâÜæåé ôï ìÝãåèïò ôçò ìíÞìçò áðü ôï BIOS, ìðïñåß íá áíé÷íåýóåé ìüíï 16 bits ìÝãåèïò óå Kbytes (65536 Kbytes = 64MB) (Þ êáé ëéãüôåñï... ïñéóìÝíá BIOS äßíïõí ðñïêáèïñéóìÝíï ìÝãåèïò ìíÞìçò 16Ì). Áí Ý÷åôå ðåñéóóüôåñá áðü 64MB, ôï FreeBSD èá ðñïóðáèÞóåé íá ôá áíé÷íåýóåé. Ç áíß÷íåõóç ùóôüóï ìðïñåß íá áðïôý÷åé.

Ãéá íá ðáñáêÜìøåôå ôï ðñüâëçìá, èá ðñÝðåé íá ÷ñçóéìïðïéÞóåôå ôçí åðéëïãÞ ôïõ ðõñÞíá ðïõ öáßíåôáé ðáñáêÜôù. ÕðÜñ÷åé ôñüðïò íá ëçöèïýí ðëÞñåéò ðëçñïöïñßåò ó÷åôéêÜ ìå ôç ìíÞìç áðü ôï BIOS, áëëÜ óôï bootblock äåí õðÜñ÷åé áñêåôüò ÷þñïò ãéá íá ãßíåé áõôü. ÊÜðïéá ìÝñá, üôáí äéïñèùèåß ôï ðñüâëçìá ôçò Ýëëåéøçò ÷þñïõ óôá bootblocks, èá ÷ñçóéìïðïéÞóïõìå ôéò åêôåôáìÝíåò ëåéôïõñãßåò ôïõ BIOS ãéá íá áíáêôÞóïõìå ðëÞñåéò ðëçñïöïñßåò ó÷åôéêÜ ìå ôç ìíÞìç. Ãéá ôçí þñá, ðñÝðåé íá ðåñéïñéóôïýìå óôçí ñýèìéóç ôçò áíôßóôïé÷çò åðéëïãÞò ôïõ ðõñÞíá.

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

¼ðïõ ôï *n* åßíáé ôï ìÝãåèïò ôçò ìíÞìçò óå kilobytes. Ãéá ìç÷Üíçìá ìå 128 MB, èá ðñÝðåé íá ÷ñçóéìïðïéÞóåôå ôï `131072`.

11. Ôï óýóôçìá ìïõ Ý÷åé ðåñéóóüôåñï áðü 1 GB RAM, êáé ðáßñíù panics ìå ìçíýìáôá “kmem_map too small”. Ðïõ åßíáé ôï ðñüâëçìá;

Öõóéïëïãéêá, ôï FreeBSD ÷ñçóéìïðïéåß ôï ìÝãåèïò ôçò åãêáôåóôçìÝíçò ìíÞìçò ãéá íá êáèïñßóåé ìéá óåéñÜ áðü ðáñáìÝôñïõò ôïõ ðõñÞíá, üðùò ôï ìÝãéóôï áñéèìü áñ÷åßùí ðïõ ìðïñåß íá åßíáé ôáõôü÷ñïíá áíïé÷ôÜ. Óå óõóôÞìáôá ìå ðåñéóóüôåñç áðü 1GB ìíÞìç, áõôüò ï ìç÷áíéóìüò “áõôüìáôçò ñýèìéóçò ìåãåèþí” ßóùò åðéëÝîåé ôéìÝò ïé ïðïßåò íá åßíáé ðïëý õøçëÝò. ÊáôÜ ôçí åêêßíçóç, ï ðõñÞíáò åê÷ùñåß äéÜöïñïõò ðßíáêåò êáé Üëëåò äïìÝò, ïé ïðïßåò êáôáëáìâÜíïõí ôïí ðåñéóóüôåñï äéáèÝóéìï ÷þñï ôïõ. Áñãüôåñá, êáèþò ôï óýóôçìá ëåéôïõñãåß, ï ðõñÞíáò äåí Ý÷åé Üëëï ÷þñï ãéá äõíáìéêÝò åê÷ùñÞóåéò ìíÞìçò, êáé äçìéïõñãåßôáé panic.

ÄçìéïõñãÞóôå ôï äéêü óáò ðõñÞíá, êáé ðñïóèÝóôå ôçí åðéëïãÞ `VM_KMEM_SIZE_MAX` óôï áñ÷åßï ñõèìßóåùí ôïõ, þóôå íá áõîÞóåôå ôï ìÝãéóôï ìÝãåèïò óå 400 MB (`options VM_KMEM_SIZE_MAX=419430400`). Ôá 400 MB öáßíåôáé íá åðáñêïýí ãéá ìç÷áíÞìáôá ìå ìÝãåèïò ìíÞìçò ùò 6 GB.

12. Ôï óýóôçìá ìïõ äåí Ý÷åé 1GB RAM, êáé ðÜëé üìùò ôï FreeBSD äçìéïõñãåß panic ìå ôï ìÞíõìá “`kmem_map too small!`”

Ôï panic äåß÷íåé üôé ôï óýóôçìá Ý÷åé ìåßíåé áðü åéêïíéêÞ ìíÞìç ãéá ðñïóùñéíÞ áðïèÞêåõóç äåäïìÝíùí äéêôýïõ (network buffers, êáé åéäéêüôåñá mbuf clusters). Ìðïñåßôå íá áõîÞóåôå ôï ìÝãåèïò ôçò åéêïíéêÞò ìíÞìçò ðïõ äéáôßèåôáé ãéá mbuf clusters, áêïëïõèþíôáò ôéò ïäçãßåò óôçí åíüôçôá ¼ñéá Äéêôýïõ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html#NMBCLUSTERS) ôïõ Åã÷åéñéäßïõ.

13. Ãéáôß ðáßñíù ôï ìÞíõìá ëÜèïõò “`/kernel: proc: table is full`”;

Ï ðõñÞíáò ôïõ FreeBSD åðéôñÝðåé êÜèå ÷ñïíéêÞ óôéãìÞ ôçí ýðáñîç åíüò óõãêåêñéìÝíïõ áñéèìïý äéåñãáóéþí. Ï áñéèìüò áõôüò âáóßæåôáé óôçí åðéëïãÞ `MAXUSERS` ôïõ ðõñÞíá. Ôï `MAXUSERS` åðçñåÜæåé åðßóçò êáé Üëëá üñéá ìÝóá óôïí ðõñÞíá, üðùò ç ðñïóùñéíÞ ìíÞìç ôïõ äéêôýïõ (network buffers) (äåßôå ôçí ðñïçãïýìåíç åñþôçóç). Áí ôï ìç÷Üíçìá óáò ëåéôïõñãåß óå õøçëü öïñôßï, ßóùò Ý÷åé íüçìá íá áõîÞóåôå ôçí åðéëïãÞ `MAXUSERS`. Ìå ôïí ôñüðï áõôü, ìáæß ìå ôï ìÝãéóôï áñéèìü äéåñãáóéþí, èá áõîçèïýí êáé Üëëá üñéá ôïõ óõóôÞìáôïò.

Ãéá íá ñõèìßóåôå ôçí ôéìÞ ôïõ `MAXUSERS`, äåßôå ôçí åíüôçôá ¼ñéá Áñ÷åßùí/Äéåñãáóéþí (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html#KERN-MAXFILES) ôïõ Åã÷åéñéäßïõ. (Áí êáé ç åíüôçôá áõôÞ áíáöÝñåôáé óå áíïé÷ôÜ áñ÷åßá, ôá ßäéá üñéá éó÷ýïõí êáé ãéá ôéò äéåñãáóßåò.)

Áí ôï ìç÷Üíçìá óáò ëåéôïõñãåß óå ÷áìçëü öïñôßï, áëëÜ åêôåëåß ìåãÜëï áñéèìü äéåñãáóéþí, ìðïñåßôå áðëþò íá ñõèìßóåôå ôïí áñéèìü ôïõò áëëÜæïíôáò ôçí ôéìÞ ôçò ìåôáâëçôÞò `kern.maxproc`. Áí ðñÝðåé íá ñõèìßóåôå áõôÞ ôç ìåôáâëçôÞ, èá ðñÝðåé íá ôçí ïñßóåôå óôï áñ÷åßï `/boot/loader.conf`. Ç ñýèìéóç äåí èá éó÷ýóåé ìÝ÷ñé íá åðáíåêêéíÞóåôå ôï óýóôçìá. Ãéá ðåñéóóüôåñåò ðëçñïöïñßåò ó÷åôéêÜ ìå ôéò ìåôáâëçôÝò ôïõ ðõñÞíá, äåßôå ôéò óåëßäåò manual loader.conf(5) êáé sysctl.conf(5). Áí üëåò áõôÝò ïé äéåñãáóßåò åêôåëïýíôáé áðü Ýíá ìüíï ÷ñÞóôç, èá ðñÝðåé åðßóçò íá ñõèìßóåôå ôçí ôéìÞ ôçò ìåôáâëçôÞò `kern.maxprocperuid` þóôå íá åßíáé êáôÜ Ýíá ìéêñüôåñç áðü ôçí íÝá ôéìÞ ôçò `kern.maxproc`. (ÐñÝðåé íá åßíáé êáôÜ Ýíá ìéêñüôåñç, ãéáôß õðÜñ÷åé ðÜíôá Ýíá ðñüãñáììá óõóôÞìáôïò, ôï init(8), ðïõ ðñÝðåé íá åêôåëåßôáé óõíÝ÷åéá.).

Ãéá íá ãßíåé ìüíéìç ìéá áëëáãÞ åíüò sysctl, ôïðïèåôÞóôå ôçí êáôÜëëçëç ôéìÞ óôï áñ÷åßï `/etc/sysctl.conf`. Ðåñéóóüôåñåò ðëçñïöïñßåò ãéá ôç ñýèìéóç ôïõ óõóôÞìáôïò ìå ôçí ÷ñÞóç ôïõ sysctl(8), ìðïñåßôå íá âñåßôå óôçí åíüôçôá Ñõèìßóåéò ìÝóù sysctl (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-sysctl.html) ôïõ Åã÷åéñéäßïõ.

14. Ãéáôß ðáßñíù ôï ìÞíõìá ëÜèïõò “`CMAP busy`” üôáí åðáíåêêéíþ ìå íÝï ðõñÞíá;

Ç ëïãéêÞ ôïõ óõóôÞìáôïò ðïõ ðñïóðáèåß íá áíé÷íåýóåé ôõ÷üí ðáëéÝò åêäüóåéò ôùí áñ÷åßùí `/var/db/kvm_*.db` êÜðïéåò öïñÝò áðïôõã÷Üíåé, êáé ç ÷ñÞóç áóýìâáôùí åêäüóåùí ìðïñåß óå ïñéóìÝíåò ðåñéðôþóåéò íá ïäçãÞóåé óå panic.

Áí óáò óõìâåß áõôü, åðáíåêêéíÞóôå óå êáôÜóôáóç åíüò ÷ñÞóôç (single user) êáé ãñÜøôå:

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

15. Ôé óçìáßíåé ôï ìÞíõìá “`ahc0: brkadrint, Illegal Host Access at seqaddr 0x0`”;

ÕðÜñ÷åé ìéá äéÝíåîç ìå ôçí êÜñôá Ultrastor SCSI Host Adapter.

ÊáôÜ ôç äéÜñêåéá ôçò äéáäéêáóßáò åêêßíçóçò, åéóÝëèåôå óôï ìåíïý ñõèìßóåùí ôïõ ðõñÞíá êáé áðåíåñãïðïéÞóôå ôç óõóêåõÞ `uha0`, ç ïðïßá åßíáé áõôÞ ðïõ ðñïêáëåß ôï ðñüâëçìá.

16. ¼ôáí îåêéíþ ôï óýóôçìá ìïõ ðáßñíù ôï ëÜèïò “`ahc0: illegal cable configuration`”. Ç êáëùäßùóç ìïõ åßíáé óùóôÞ. Ôé óõìâáßíåé;

Ç ìçôñéêÞ ðëáêÝôá óáò äåí Ý÷åé ôá áðáéôïýìåíá åîùôåñéêÜ êõêëþìáôá þóôå íá õðïóôçñßæåé áõôüìáôï ôåñìáôéóìü ôïõ äéáýëïõ SCSI. Áíôß íá âáóßæåóôå óôïí áõôüìáôï ôåñìáôéóìü, äçëþóôå óôï SCSI BIOS ôïí óùóôü ôåñìáôéóìü ãéá ôç äéÜôáîç óõóêåõþí ðïõ Ý÷åôå. Ôï ðñüãñáììá ïäÞãçóçò ôïõ AIC7XXX äåí ìðïñåß íá êáèïñßóåé áí åßíáé äéáèÝóéìï ôï êýêëùìá ðïõ ÷ñçóéìïðïéåßôáé ãéá ôçí áíß÷íåõóç ôïõ êáëùäßïõ (Üñá êáé ôïõ áõôüìáôïõ ôåñìáôéóìïý). Ôï ðñüãñáììá ïäÞãçóçò õðïèÝôåé üôé õðÜñ÷åé õðïóôÞñéîç, åöüóïí ïé ñõèìßóåéò ðïõ ðåñéÝ÷ïíôáé óôç óåéñéáêÞ EEPROM áíáöÝñïõí "áõôüìáôï ôåñìáôéóìü". Óõ÷íÜ, ÷ùñßò ôï åîùôåñéêü êýêëùìá áíß÷íåõóçò ôïõ êáëùäßïõ, ôï ðñüãñáììá ïäÞãçóçò èá ñõèìßæåé ëáíèáóìÝíá ôïí ôåñìáôéóìü, êÜôé ðïõ ìðïñåß íá äçìéïõñãÞóåé ðñüâëçìá óôçí áîéïðéóôßá ôïõ äéáýëïõ SCSI.

17. Ãéáôß ôï Sendmail äßíåé ôï ìÞíõìá ëÜèïõò ““`mail loops back to myself`””;

Áõôü ðåñéãñÜöåôáé óôï sendmail FAQ üðùò öáßíåôáé ðáñáêÜôù:

> * Ðáßñíù ìçíýìáôá ëÜèïõò "Local configuration error" üðùò ôï:
>
> 553 relay.domain.net config error: mail loops back to myself
> 554 <user@domain.net>... Local configuration error
>
> Ðùò ìðïñþ íá åðéëýóù ôï ðñüâëçìá;
>
> ¸÷åôå æçôÞóåé íá êáôåõèýíåôå ôï mail ðñïò ôï domain (ð.÷. domain.net)
> ðñïò êÜðïéï óõãêåêñéìÝíï ìç÷Üíçìá (óôçí ðåñßðôùóç áõôÞ, ôï
> relay.domain.net) ÷ñçóéìïðïéþíôáò ìéá åããñáöÞ MX, áëëÜ ôï ìç÷Üíçìá
> ðïõ êÜíåé ôçí áíáêáôåýèõíóç äåí áíáãíùñßæåé ôïí åáõôü ôïõ ùò
> domain.net. ÐñïóèÝóôå ôï domain.net óôï /etc/mail/local-host-names
> (áí ÷ñçóéìïðïéåßôå ôï FEATURE(use_cw_file)) Þ ðñïóèÝóôå
> "Cw domain.net" óôï /etc/mail/sendmail.cf.

Ç ôñÝ÷ïõóá Ýêäïóç ôïõ sendmail FAQ (ftp://rtfm.mit.edu/pub/usenet/news.answers/mail/sendmail-faq) äåí óõíôçñåßôáé ðëÝïí ìå êÜèå Ýêäïóç ôïõ sendmail. Ùóôüóï, äçìïóéåýåôáé áíÜ ôáêôÜ äéáóôÞìáôá óôéò ëßóôåò 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), êáé news.answers (news:news.answers). Ìðïñåßôå åðßóçò íá ëÜâåôå áíôßãñáöï ìÝóù email, óôÝëíïíôáò Ýíá ìÞíõìá óôï <`mail-server@rtfm.mit.edu`> ìå ôçí åíôïëÞ `send usenet/news.answers/mail/sendmail-faq` óôï êýñéï ìÝñïò ôïõ ìçíýìáôïò.

18. Ãéáôß äåí óõìðåñéöÝñïíôáé óùóôÜ ïé åöáñìïãÝò ðëÞñïõò ïèüíçò óå áðïìáêñõóìÝíá ìç÷áíÞìáôá;

Åßíáé ðéèáíüí ôï áðïìáêñõóìÝíï ìç÷Üíçìá íá ñõèìßæåé ôïí ôýðï ôïõ ôåñìáôéêïý óáò óå êÜôé äéáöïñåôéêü áðü ôïí ôýðï `cons25` ðïõ áðáéôåßôáé áðü ôçí êïíóüëá ôïõ FreeBSD.

ÕðÜñ÷ïõí äéÜöïñïé ôñüðïé ãéá íá ðáñáêÜìøåôå áõôü ôï ðñüâëçìá:

- ÌåôÜ ôçí åßóïäï óáò óôï áðïìáêñõóìÝíï ìç÷Üíçìá, ïñßóôå ôçí ìåôáâëçôÞ TERM ôïõ êåëýöïõò óå `ansi` Þ `sco`, åöüóïí ôï áðïìáêñõóìÝíï ìç÷Üíçìá ìðïñåß íá ëåéôïõñãÞóåé ìå áõôÜ ôá åßäç ôåñìáôéêþí.
- Óôçí êïíóüëá ôïõ FreeBSD, ÷ñçóéìïðïéÞóôå êÜðïéï åîïìïéùôÞ ôåñìáôéêïý VT100, üðùò ôï **screen**. Ôï **screen** óáò äßíåé ôç äõíáôüôçôá íá Ý÷åôå ðïëëáðëÝò óõíåäñßåò áðü Ýíá ìüíï ôåñìáôéêü, êáé åßíáé Ýôóé êáé áëëéþò ÷ñÞóéìï

ðñüãñáììá. ÊÜèå ðáñÜèõñï ôïõ **screen** óõìðåñéöÝñåôáé ùò ôåñìáôéêü ôïõ VT100, Ýôóé ç ìåôáâëçôÞ TERM óôïí áðïìáêñõóìÝíï õðïëïãéóôÞ èá ðñÝðåé íá ñõèìéóôåß óå `vt100`.

- Åãêáôáóôåßóôå ôçí êáôá÷þñçóç `cons25` óôç âÜóç äåäïìÝíùí ôåñìáôéêþí ôïõ áðïìáêñõóìÝíïõ õðïëïãéóôÞ. Ï ôñüðïò ãéá íá ãßíåé áõôü, åîáñôÜôáé áðü ôï ëåéôïõñãéêü óýóôçìá ôïõ áðïìáêñõóìÝíïõ õðïëïãéóôÞ. Ôõóéïëïãéêü, èá âñåßôå áõôÝò ôéò ðëçñïöïñßåò óôá åã÷åéñßäéá äéá÷åßñéóçò óõóôÞìáôïò ôïõ áðïìáêñõóìÝíïõ ìç÷áíÞìáôïò.
- Óôï ôïðéêü óáò FreeBSD ìç÷Üíçìá, ÷ñçóéìïðïéÞóôå ôïí X server êáé êÜíôå login óôïí áðïìáêñõóìÝíï ìç÷Üíçìá ÷ñçóéìïðïéþíôáò êÜðïéï åîïìïéùôÞ ôåñìáôéêïý üðùò ôï `xterm` Þ ôï `rxvt`. Óôçí ðåñßðôùóç áõôÞ, èá ðñÝðåé óôï áðïìáêñõóìÝíï ìç÷Üíçìá íá ñõèìßóåôå ôçí ìåôáâëçôÞ TERM óå `xterm` Þ `vt100`.

19. Ãéáôß ôï ìç÷Üíçìá ìïõ äåß÷íåé ôï ìÞíõìá “`calcru: negative time...`”;

Áõôü ìðïñåß íá óõìâåß áðü äéÜöïñåò áéôßåò ðïõ ó÷åôßæïíôáé ìå interrupts, ôüóï óôï õëéêü üóï êáé óôï ëïãéóìéêü. Ìðïñåß íá ïöåßëåôáé óå ðñïâëÞìáôá (bugs) áëëÜ ìðïñåß åðßóçò íá ðñïêëçèåß åîáéôßáò ôçò öýóçò êÜðïéùí óõóêåõþí. ¸íáò óõíçèéóìÝíïò ôñüðïò ðñüêëçóçò ôïõ ðñïâëÞìáôïò, åßíáé ç åêôÝëåóç åöáñìïãþí TCP/IP ìå ìåãÜëï MTU ìÝóù ôçò ðáñÜëëçëçò èýñáò. Ìðïñåß åðßóçò íá ðñïêëçèåß áðü êÜðïéïõò åðéôá÷õíôÝò ãñáöéêþí, êáé óôçí ðåñßðôùóç áõôÞ ôï ðñþôï ðñÜãìá ðïõ èá ðñÝðåé íá åëÝãîåôå åßíáé ç ñýèìéóç interrupt ôçò áíôßóôïé÷çò êÜñôáò.

ÐáñåíÝñãåéá áõôïý ôïõ ðñïâëÞìáôïò åßíáé ï áðüôïìïò ôåñìáôéóìüò äéåñãáóéþí ìå ôï ìÞíõìá “SIGXCPU exceeded cpu time limit”.

Áí ôï ðñüâëçìá äåí ìðïñåß íá ëõèåß ìå äéáöïñåôéêü ôñüðï, ç ëýóç åßíáé íá ïñßóåôå ôçí ðáñáêÜôù ìåôáâëçôÞ ôïõ sysctl:

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

> **Óçìåßùóç:** Ç åðéëïãÞ `-w` ôïõ sysctl(8) èåùñåßôáé ðáñù÷çìÝíç êáé áãíïåßôáé óéùðçëÜ áðü ôï FreeBSD 4.4-RELEASE êáé ìåôÜ. Ìðïñåßôå ìå áóöÜëåéá íá ôï ðáñáëåßøåôå êáôÜ ôç ñýèìéóç ôùí åðéëïãþí ìå ôçí `sysctl` üðùò öáßíåôáé ðáñáðÜíù.

Ôï ðáñáðÜíù èá Ý÷åé åðßäñáóç óôçí áðüäïóç, áëëÜ óå ó÷Ýóç ìå ôçí áéôßá ôïõ ðñïâëÞìáôïò, ìÜëëïí äåí èá ôï ðáñáôçñÞóåôå. Áí ôï ðñüâëçìá åðéìÝíåé, äéáôçñÞóôå ôçí ôéìÞ ôïõ sysctl óôï Ýíá, êáé ñõèìßóôå ôçí åðéëïãÞ `NTIMECOUNTER` óôïí ðõñÞíá óáò, óå ïëïÝíá áõîáíüìåíåò ôéìÝò. Áí öôÜóåôå ôçí ôéìÞ `NTIMECOUNTER=20` êáé ôï ðñüâëçìá äåí Ý÷åé ëõèåß, ôá interrupts óôï ìç÷Üíçìá óáò åßíáé ðïëý ðñïâëçìáôéêÜ êáé áêáôÜëëçëá ãéá áêñéâÞ ñýèìéóç ôçò þñáò.

20. Ãéáôß ç PnP êÜñôá ìïõ äåí áíé÷íåýåôáé ðëÝïí (Þ áíé÷íåýåôáé ùò `unknown`) ìåôÜ ôçí áíáâÜèìéóç óå FreeBSD 4.X;

Ôï FreeBSD 4.X áêïëïõèåß ðëÝïí áñêåôÜ ðéï ðéóôÜ ôï ðñüôõðï *PnP* êáé áõôü äçìéïõñãåß ïñéóìÝíåò öïñÝò ôçí ðáñåíÝñãåéá íá ìç ëåéôïõñãïýí êÜðïéåò óõóêåõÝò PnP (ð.÷. êÜñôåò Þ÷ïõ êáé åóùôåñéêÜ modems) ïé ïðïßåò ùóôüóï ëåéôïõñãïýóáí óôï FreeBSD 3.×.

Ïé ëüãïé ãéá ôçí óõìðåñéöïñÜ áõôÞ, åîçãïýíôáé óôï áêüëïõèï e-mail, ôï ïðïßï óôÜëèçêå óôç ëßóôá freebsd-questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) áðü ôïí Peter Wemm, ùò áðÜíôçóç óå åñþôçóç ó÷åôéêÜ ìå Ýíá åóùôåñéêü modem ôï ïðïßï äåí Þôáí áíé÷íåýóéìï áðü ôï óýóôçìá ìåôÜ áðü áíáâÜèìéóç óå FreeBSD 4.X (ôá ó÷üëéá ìÝóá óå `[]` Ý÷ïõí ðñïóôåèåß ãéá íá ãßíåé ðéï êáôáíïçôü ôï áíôéêåßìåíï ôçò óõæÞôçóçò).

> **Óçìåßùóç:** Ôï ðåñéå÷üìåíï áõôÞò ôçò ðáñÜèåóçò Ý÷åé áíáíåùèåß óå ó÷Ýóç ìå ôï áñ÷éêü êåßìåíï.

> Ôï PNP bios ôï ðñï-ñýèìéóå [ôï modem] êáé ôï Üöçóå óôçí ðåñéï÷Þ äéåõèýíóåùí ôùí èõñþí, êáé Ýôóé [óôçí Ýêäïóç 3.×] ç ðáëáéÜ ôýðïõ áíß÷íåõóç ISA ôï "âñÞêå" åêåß.
>
> Óôçí Ýêäïóç 4.0, ï êþäéêáò äéá÷åßñéóçò ôïõ ISA, åßíáé ðïëý ðåñéóóüôåñï ðñïóáíáôïëéóìÝíïò óôï PnP ìïíôÝëï. Óôï 3.× Þôáí äõíáôüí ç áíß÷íåõóç ISA íá åíôïðßóåé ìéá "÷áìÝíç" óõóêåõÞ êáé Ýðåéôá ç PNP óõóêåõÞ íá ôáéñéÜîåé êáé íá áðïôý÷åé ç ñýèìéóç ôçò ëüãù äéÝíåîçò ðüñùí. ¸ôóé, áðåíåñãïðïéïýíôáé áñ÷éêÜ ïé ðñïãñáììáôéæüìåíåò êÜñôåò, þóôå íá ìç óõìâåß áõôÞ ç äéðëÞ áíß÷íåõóç. Áõôü åðßóçò óçìáßíåé üôé ç áíß÷íåõóç ðñÝðåé íá ãíùñßæåé ôá PnP ids ôùí õðïóôçñéæüìåíùí óõóêåõþí. Åßíáé óôéò ðñïèÝóåéò ìáò íá êÜíïõìå ôç äéáäéêáóßá áõôÞ ðåñéóóüôåñç ðñïóâÜóéìç óôïõò ÷ñÞóôåò.

Ãéá íá ëåéôïõñãÞóåé îáíÜ ç óõóêåõÞ, ðñÝðåé íá âñåèåß ôï PNP id ôçò êáé íá ðñïóôåèåß óôç ëßóôá ôùí áíé÷íåýóåùí ISA ðïõ ÷ñçóéìïðïéïýíôáé ãéá ôçí áíáãíþñéóç PnP óõóêåõþí. Áõôü ìðïñåß íá ãßíåé ìå ôç ÷ñÞóç ôçò pnpinfo(8) ãéá ôçí áíß÷íåõóç ôçò óõóêåõÞò, ãéá ðáñÜäåéãìá áõôÞ åßíáé ç Ýîïäïò ôçò pnpinfo(8) ãéá Ýíá åóùôåñéêü modem:

```
# 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)
```

[ðáñáëåßðïíôáé ïé õðüëïéðåò ãñáììÝò TAG]

```
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
```

Ïé ðëçñïöïñßåò ðïõ áðáéôïýíôáé, âñßóêïíôáé óôç ãñáììÞ "Vendor ID", óôçí áñ÷Þ ôçò åîüäïõ. Ï äåêáåîáäéêüò áñéèìüò óôéò ðáñåíèÝóåéò (óôï ðáñÜäåéãìá ìáò 0x3024a341) åßíáé ôï PnP id åíþ ôï áëöáñéèìçôéêü ðïõ âñßóêåôáé áêñéâþò ðñéí áðü áõôüí åßíáé Ýíá ìïíáäéêü ASCII áíáãíùñéóôéêü.

ÅíáëëáêôéêÜ, áí ôï pnpinfo(8) äåí äåß÷íåé ôçí æçôïýìåíç êÜñôá, ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå ôï pciconf(8). ÐáñáêÜôù öáßíåôáé Ýíá ìÝñïò ôçò åîüäïõ ôçò `pciconf -vl` ãéá Ýíá êýêëùìá Þ÷ïõ åíóùìáôùìÝíïõ óôç ìçôñéêÞ:

```
# pciconf -vl
```

```
chip1@pci0:31:5:        class=0x040100 card=0x00931028 chip=0x24158086 rev=0x02 hdr=0x00
    vendor   = 'Intel Corporation'
    device   = '82801AA 8xx Chipset AC'97 Audio Controller'
    class    = multimedia
    subclass = audio
```

Åäþ, èá ÷ñçóéìïðïéïýóáìå ôçí ôéìÞ ôïõ `chip`, “0x24158086”.

Ç ðëçñïöïñßá áõôÞ (Vendor ID Þ ôéìÞ chip) èá ðñÝðåé íá ðñïóôåèåß óôï áñ÷åßï `/usr/src/sys/isa/sio.c`.

Èá ðñÝðåé ðñþôá íá êñáôÞóåôå Ýíá áíôßãñáöï áóöáëåßáò ôïõ `sio.c`, ãéá ôçí ðåñßðôùóç ðïõ êÜôé ðÜåé óôñáâÜ. Åðßóçò, èá ÷ñåéáóôåßôå ôï áíôßãñáöï ãéá íá äçìéïõñãÞóåôå Ýíá patch ôï ïðïßï èá êáôáèÝóåôå ìå ôçí áíáöïñÜ ðñïâëÞìáôïò (PR) ðïõ èá ìáò óôåßëåôå (êáé èá ìáò óôåßëåôå PR, Ýôóé;). Êáôüðéí åðåîåñãáóôåßôå ôï `sio.c` êáé øÜîôå ãéá ôç ãñáììÞ

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

Ýðåéôá ìåôáêéíçèåßôå ðñïò ôá êÜôù ãéá íá âñåßôå ôï óùóôü óçìåßï ãéá íá ðñïóèÝóåôå ôçí êáôá÷þñçóç ôçò óõóêåõÞò óáò. Ïé êáôá÷ùñÞóåéò öáßíïíôáé üðùò ðáñáêÜôù êáé åßíáé ôáîéíïìçìÝíåò êáôÜ ôï áëöáñéèìçôéêü ASCII Vendor ID ôï ïðïßï èá ðñÝðåé íá ðåñéëçöèåß óôï ó÷üëéï óôï äåîéü ìÝñïò ôçò ãñáììÞò ìáæß ìå üëç ôçí ðåñéãñáöÞ *Device Description* (áí ÷ùñÜåé, áëëéþò ìÝñïò ôçò) áðü ôçí Ýîïäï ôçò pnpinfo(8):

```
{0x0f804f3f, NULL},     /* OZO800f - Zoom 2812 (56k Modem) */
{0x39804f3f, NULL},     /* OZO8039 - Zoom 56k flex */
{0x3024a341, NULL},     /* PMC2430 - Pace 56 Voice Internal Modem */
{0x1000eb49, NULL},     /* ROK0010 - Rockwell ? */
{0x5002734a, NULL},     /* RSS0250 - 5614Jx3(G) Internal Modem */
```

ÐñïóèÝóôå ôï äåêáåîáäéêü Vendor ID ãéá ôç óõóêåõÞ óáò óôï óùóôü ìÝñïò, áðïèçêåýóôå ôï áñ÷åßï, áíáäçìéïõñãÞóôå ôïí ðõñÞíá óáò, êáé åðáíåêêéíÞóôå. Èá ðñÝðåé ôþñá ç óõóêåõÞ óáò íá âñåèåß ùò óõóêåõÞ `sio` üðùò óõíÝâáéíå êáé ìå ôï FreeBSD 3.X

21. Ãéáôß ðáßñíù ôï ëÜèïò “`nlist failed`” üôáí åêôåëþ, ãéá ðáñÜäåéãìá, ôï `top` Þ ôï `systat`;

Ôï ðñüâëçìá åßíáé üôé ç åöáñìïãÞ ðïõ ðñïóðáèåßôå íá åêôåëÝóåôå øÜ÷íåé ãéá Ýíá óõãêåêñéìÝíï óýìâïëï óôïí ðõñÞíá, áëëÜ ãéá êÜðïéï ëüãï äåí ìðïñåß íá ôï âñåß. Ôï óöÜëìá áõôü ìðïñåß íá ïöåßëåôáé óå äýï ðñïâëÞìáôá:

- Ï ðõñÞíáò óáò êáé ôá õðüëïéðá âáóéêÜ ðñïãñÜììáôá (userland) äåí åßíáé óå óõã÷ñïíéóìü (ð.÷. Ý÷åôå äçìéïõñãÞóåé íÝï ðõñÞíá, áëëÜ äåí åêôåëÝóáôå `installworld`, Þ áíôßóôñïöá), ìå áðïôÝëåóìá ï ðßíáêáò óõìâüëùí íá åßíáé äéáöïñåôéêüò áðü áõôüí ðïõ ðéóôåýåé ç åöáñìïãÞ. Áí ðñüêåéôáé ãéá áõôÞ ôçí ðåñßðôùóç, áðëþò ïëïêëçñþóôå ôç äéáäéêáóßá áíáâÜèìéóçò (äåßôå ôï `/usr/src/UPDATING` ãéá ôç óùóôÞ áêïëïõèßá åíôïëþí).
- Äåí ÷ñçóéìïðïéåßôå ôï `/boot/loader` ãéá íá öïñôþóåôå ôïí ðõñÞíá óáò, áëëÜ ôïí öïñôþíåôå áðåõèåßáò áðü ôï boot2 (äåßôå ôï boot(8)). Áí êáé äåí åßíáé ëÜèïò íá ðáñáêÜìøåôå ôïí `/boot/loader`, óå ãåíéêÝò ãñáììÝò ôï ðñüãñáììá áõôü ôá êáôáöÝñíåé êáëýôåñá óôï íá äéáèÝôåé ôá óýìâïëá ôïõ ðõñÞíá óôéò åöáñìïãÝò ÷ñÞóçò.

22. Ãéáôß ðáßñíåé ôüóï ÷ñüíï íá óõíäåèþ ìå ôïí õðïëïãéóôÞ ìïõ ìÝóù `ssh` Þ `telnet`;

Ôï óýìðôùìá: ÕðÜñ÷åé ìåãÜëç êáèõóôÝñçóç ìåôáîý ôçò óôéãìÞò ðïõ áðïêáèßóôáôáé ç TCP óýíäåóç êáé ôçò óôéãìÞò ðïõ ôï ðñüãñáììá óôç ìåñéÜ ôïõ ðåëÜôç æçôÜåé ôïí êùäéêü ðñüóâáóçò (Þ óôçí ðåñßðôùóç ôïõ telnet(1), ôçò óôéãìÞò ðïõ åìöáíßæåôáé ç ðñïôñïðÞ login).

Ôï ðñüâëçìá: Ôï ðéï ðéèáíü åßíáé üôé ç êáèõóôÝñçóç ïöåßëåôáé óôçí ðñïóðÜèåéá ðïõ êáôáâÜëëåé ôï ëïãéóìéêü óôç ìåñéÜ ôïõ åîõðçñåôçôÞ íá âñåé ôï üíïìá ôïõ ìç÷áíÞìáôïò - ðåëÜôç áðü ôçí IP äéåýèõíóç ôïõ. Ïé ðåñéóóüôåñïé åîõðçñåôçôÝò, óõìðåñéëáìâáíïìÝíùí ôïõ Telnet êáé SSH ðïõ Ýñ÷ïíôáé ìå ôï FreeBSD, ëåéôïõñãïýí ìå áõôü ôïí ôñüðï, þóôå ìåôáîý Üëëùí, íá áðïèçêåýóïõí ôï üíïìá ôïõ ìç÷áíÞìáôïò óå Ýíá áñ÷åßï êáôáãñáöÞò ãéá ìåëëïíôéêÞ áíáöïñÜ áðü ôïí äéá÷åéñéóôÞ.

Ç èåñáðåßá: Áí ôï ðñüâëçìá ðñïêýðôåé êÜèå öïñÜ ðïõ óõíäÝåóôå áðü ôïí õðïëïãéóôÞ óáò (ôïí ðåëÜôç) óå ïðïéïäÞðïôå åîõðçñåôçôÞ, ôï ðñüâëçìá âñßóêåôáé óôïí ðåëÜôç. Ìå ôïí ßäéï ôñüðï, áí ôï ðñüâëçìá óõìâáßíåé ìüíï üôáí êÜðïéïò óõíäÝåôáé óôïí õðïëïãéóôÞ óáò (ôïí åîõðçñåôçôÞ), ôï ðñüâëçìá âñßóêåôáé óôïí åîõðçñåôçôÞ.

Áí ôï ðñüâëçìá åßíáé óôïí ðåëÜôç, ç ìüíç èåñáðåßá åßíáé íá äéïñèþóåôå ôï DNS, þóôå ï åîõðçñåôçôÞò íá ìðïñåß íá ôï âñåé. Áí ôï ðñüâëçìá åìöáíßæåôáé óôï ôïðéêü óáò äßêôõï, èåùñåßóôå ôï ðñüâëçìá óôïí åîõðçñåôçôÞ êáé óõíå÷ßóôå ôçí áíÜãíùóç. Áíôßèåôá, áí ôï ðñüâëçìá åìöáíßæåôáé óå óõíäÝóåéò ìÝóù Internet, êáôÜ ðÜóá ðéèáíüôçôá èá ÷ñåéáóôåß íá åðéêïéíùíÞóåôå ìå ôïí ISP óáò êáé íá æçôÞóåôå íá óáò ôï äéïñèþóåé.

Áí ôï ðñüâëçìá åßíáé ìå ôïí åîõðçñåôçôÞ, êáé åìöáíßæåôáé óôï ôïðéêü óáò äßêôõï, èá ðñÝðåé íá ôïí ñõèìßóåôå þóôå íá ìðïñåß íá åêôåëåß áíáæçôÞóåéò ôýðïõ äéåýèõíóç óå üíïìá, ãéá ôçí ôïðéêÞ ðåñéï÷Þ äéåõèýíóåùí óáò. Äåßôå ôéò óåëßäåò manual ôùí hosts(5) êáé named(8) ãéá ðåñéóóüôåñåò ðëçñïöïñßåò. Áí ôï ðñüâëçìá åìöáíßæåôáé óôéò óõíäÝóåéò ìÝóù Internet, ìðïñåß íá ïöåßëåôáé óå êáêÞ ëåéôïõñãßá ôïõ resolver óôïí åîõðçñåôçôÞ óáò. Ãéá íá ôï åëÝãîåôå, äïêéìÜóôå íá âñåßôå êÜðïéï Üëëï ìç÷Üíçìá, ãéá ðáñÜäåéãìá ôï `www.yahoo.com`. Áí ïýôå áõôü äïõëåýåé, åêåß âñßóêåôáé ôï ðñüâëçìá óáò.

ÌåôÜ áðü ìéá íÝá åãêáôÜóôáóç ôïõ FreeBSD åßíáé åðßóçò ðéèáíü íá ëåßðïõí ïé ðëçñïöïñßåò ãéá ôïí ôïìÝá (domain) êáé ôïí åîõðçñåôçôÞ ïíïìÜôùí (nameserver) áðü ôï áñ÷åßï `/etc/resolv.conf`. Áõôü åðßóçò èá ðñïêáëÝóåé êáèõóôÝñçóç óôï **SSH**, êáèþò ç åðéëïãÞ "UseDNS" Ý÷åé ùò ðñïåðéëåãìÝíç ôçí ôéìÞ "yes" óôï áñ÷åßï ñõèìßóåùí `sshd_config` óôïí êáôÜëïãï `/etc/ssh`. Áí åßíáé áõôÞ ç áéôßá ôïõ ðñïâëÞìáôïò, èá ðñÝðåé åßôå íá óõìðëçñþóåôå ôéò áðáéôïýìåíåò ðëçñïöïñßåò óôï `/etc/resolv.conf` Þ íá èÝóåôå ôï "UseDNS" óôï "no" óôï áñ÷åßï `sshd_config` ùò ðñïóùñéíÞ ëýóç.

23. Ðïéá åßíáé ç Ýííïéá ôïõ "`stray (ðåñéðëáíþìåíïõ) IRQ`";

Ôá stray IRQs åßíáé óçìÜäéá ðñïâëçìÜôùí õëéêïý ðïõ ÷ñçóéìïðïéåß IRQs, éäéáßôåñá óå ó÷Ýóç ìå õëéêü ðïõ êáôÜ ôç ìÝóç ôïõ êýêëïõ áíáãíþñéóçò (acknowledge cycle) ôïõ interrupt, óôáìáôÜåé íá ìåôáäßäåé ôçí áíôßóôïé÷ç áßôçóç äéáêïðÞò.

¸÷åôå ôñåéò åðéëïãÝò ãéá íá áíôéìåôùðßóåôå áõôü ôï ðñüâëçìá:

- Áíå÷èåßôå ôéò ðñïåéäïðïéÞóåéò. ¸ôóé êáé áëëéþò, ìåôÜ ôéò 5 ðñþôåò, äåí èá äåßôå Üëëåò.
- ÓôáìáôÞóôå åíôåëþò ôéò ðñïåéäïðïéÞóåéò, áëëÜæïíôáò ôï 5 óå 0 óôçí `isa_strayintr()`.
- ÓôáìáôÞóôå ôéò ðñïåéäïðïéÞóåéò åãêáèéóôþíôáò õëéêü ãéá ôçí ðáñÜëëçëç ðüñôá ðïõ íá ÷ñçóéìïðïéåß ôï IRQ 7 êáé ôï áíôßóôïé÷ï ãéá áõôü ðñüãñáììá ïäÞãçóçò PPP (áõôü óõìâáßíåé óôá ðåñéóóüôåñá óõóôÞìáôá) êáé åãêáôáóôÞóôå Ýíá ïäçãü IDE Þ Üëëï õëéêü ðïõ íá ÷ñçóéìïðïéåß ôï irq 15 ìáæß ìå ôï êáôÜëëçëï ðñüãñáììá ïäÞãçóçò ôïõ.

24. Ãéáôß âëÝðù óõíÝ÷åéá ôï ìÞíõìá "`file: table is full`" óôï dmesg;

Ôï ìÞíõìá áõôü óçìáßíåé üôé Ý÷åôå åîáíôëÞóåé ôïí áñéèìü ôùí äéáèÝóéìùí ðåñéãñáöÝùí áñ÷åßùí (file descriptors) óôï óýóôçìá óáò. Ðáñáêáëïýìå äåßôå ôï kern.maxfiles (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html#KERN-MAXFILES) ôìÞìá óôï êåöÜëáéï Ñýèìéóç Ïñßùí ÐõñÞíá (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html) ôïõ Åã÷åéñéäßïõ, ãéá åñìçíåßá êáé åðßëõóç ôïõ ðñïâëÞìáôïò.

25. Ãéáôß ôï ñïëüé óôï öïñçôü ìïõ õðïëïãéóôÞ äåí êñáôÜåé ôçí óùóôÞ þñá;

Ï öïñçôüò õðïëïãéóôÞò óáò Ý÷åé äýï Þ ðåñéóóüôåñá ñïëüãéá, êáé ôï FreeBSD Ý÷åé åðéëÝîåé íá ÷ñçóéìïðïéÞóåé ôï ëÜèïò.

ÅêôåëÝóôå ôçí dmesg(8), êáé åëÝãîôå ãéá ãñáììÝò ðïõ ðåñéÝ÷ïõí ôçí ëÝîç `Timecounter`. Ç ôåëåõôáßá áðü ôéò ãñáììÝò ðïõ èá åêôõðùèåß äåß÷íåé ôï ñïëüé ðïõ åðéëÝ÷èçêå áðü ôï FreeBSD êáé ó÷åäüí óßãïõñá èá åßíáé ôï `TSC`.

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

Ìðïñåßôå íá ôï åðéâåâáéþóåôå áõôü, åëÝã÷ïíôáò ôçí ôéìÞ ôïõ `kern.timecounter.hardware` sysctl(3).

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

Ôï BIOS ßóùò íá ôñïðïðïéåß ôçí ôéìÞ ôïõ ñïëïãéïý TSC— åíäå÷ïìÝíùò ãéá íá áëëÜîåé ôçí ôá÷ýôçôá ôïõ åðåîåñãáóôÞ üôáí ëåéôïõñãåß ìå ìðáôáñßåò, Þ üôáí åéóÝñ÷åôáé óå êáôÜóôáóç ÷áìçëÞò êáôáíÜëùóçò, áëëÜ ôï FreeBSD äåí ãíùñßæåé ãéá áõôÝò ôéò áëëáãÝò êáé öáßíåôáé íá êåñäßæåé Þ íá ÷Üíåé ÷ñüíï.

Óôï ðáñÜäåéãìá ìáò, åßíáé åðßóçò äéáèÝóéìï ôï ñïëüé `i8254` êáé ìðïñåßôå íá ôï åðéëÝîåôå ãñÜöïíôáò ôï üíïìá ôïõ óôï sysctl(3) `kern.timecounter.hardware`.

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

Ï öïñçôüò õðïëïãéóôÞò óáò èá ðñÝðåé ôþñá íá åßíáé ðéï áêñéâÞò óôçí ôÞñçóç ôïõ ÷ñüíïõ.

Ãéá íá ðáñáìåßíåé ç áëëáãÞ áõôÞ óå êÜèå åêêßíçóç, ðñïóèÝóôå ôçí ðáñáêÜôù ãñáììÞ óôï `/etc/sysctl.conf`.

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

26. Ãéáôß ï öïñçôüò ìïõ õðïëïãéóôÞò äåí áíáãíùñßæåé óùóôÜ ôéò êÜñôåò ôýðïõ PC card;

Ôï ðñüâëçìá åßíáé êïéíü óå öïñçôÜ ðïõ åêêéíïýí ðåñéóóüôåñá áðü Ýíá ëåéôïõñãéêÜ óõóôÞìáôá. ÏñéóìÝíá ìç-BSD ëåéôïõñãéêÜ óõóôÞìáôá áöÞíïõí ôéò PC cards óå ìç-ðñïâëÝøéìç êáôÜóôáóç. Ç åíôïëÞ `pccardd` óå áõôÞ ôçí ðåñßðôùóç, áíé÷íåýåé ôçí êÜñôá ùò “`"(null)""(null)"`” áíôß ãéá ôï ðñáãìáôéêü ôçò ìïíôÝëï.

ÐñÝðåé íá áðïóõíäÝóåôå åíôåëþò ôçí ôñïöïäïóßá áðü ôçí èýñá PC card þóôå ôï õëéêü íá åðáíÝëèåé óôçí áñ÷éêÞ ôïõ êáôÜóôáóç. ÁðåíåñãïðïéÞóôå ðëÞñùò ôïí öïñçôü õðïëïãéóôÞ óáò. (Ìçí ôïí âÜëåôå óå êáôÜóôáóç áíáìïíÞò Þ ýðíïõ, èá ðñÝðåé íá áðåíåñãïðïéçèåß åíôåëþò.) ÐåñéìÝíåôå ãéá ëßãá ëåðôÜ êáé åðáíåêêéíÞóôå. Èá ðñÝðåé ôþñá ç PC card íá ëåéôïõñãåß êáíïíéêÜ.

Ôï õëéêü êÜðïéùí öïñçôþí õðïëïãéóôþí óôçí ðñáãìáôéêüôçôá ðáñáìÝíåé åíåñãü, áêüìá êáé üôáí õðïôßèåôáé üôé ï õðïëïãéóôÞò åßíáé áíåíåñãüò. Áí ôï ðáñáðÜíù äåí Ý÷åé ôï åðéèõìçôü áðïôÝëåóìá, ôåñìáôßóôå ôç ëåéôïõñãßá ôïõ õðïëïãéóôÞ óáò, áöáéñÝóôå ôçí ìðáôáñßá, ðåñéìÝíåôå ëßãï, óõíäÝóôå îáíÜ ôçí ìðáôáñßá êáé åðáíåêêéíÞóôå.

27. Ãéáôß ï öïñôùôÞò åêêßíçóçò ôïõ FreeBSD äåß÷íåé ôï ìÞíõìá ëÜèïõò “`Read error`” êáé óôáìáôÜåé ìåôÜ ôçí ïèüíç ôïõ BIOS;

Ï öïñôùôÞò åêêßíçóçò ôïõ FreeBSD äåí áíáãíùñßæåé óùóôÜ ôçí ãåùìåôñßá ôïõ óêëçñïý äßóêïõ. Ìðïñåßôå íá ôçí ñõèìßóåôå ÷åéñïêßíçôá ìÝóá áðü ôçí fdisk êáôÜ ôçí äçìéïõñãßá Þ ôñïðïðïßçóçò ôïõ slice ôïõ FreeBSD.

Ìðïñåßôå íá âñåßôå ôéò óùóôÝò ôéìÝò ãéá ôçí ãåùìåôñßá ôïõ ïäçãïý óôï BIOS ôïõ ìç÷áíÞìáôïò. ØÜîôå ãéá ôïí áñéèìü ôùí êõëßíäñùí, êåöáëþí êáé ôïìÝùí ãéá ôïí ïäçãü ðïõ èÝëåôå.

ÌÝóá áðü ôçí fdisk ôïõ sysinstall(8), ðéÝóôå ôï **G** ãéá íá ïñßóåôå ôçí ãåùìåôñßá ôïõ ïäçãïý.

Èá åìöáíéóôåß Ýíáò äéÜëïãïò ðïõ èá æçôÜåé ôïí áñéèìü ôùí êõëßíäñùí, êåöáëþí êáé ôïìÝùí. ÐëçêôñïëïãÞóôå ôïõò áñéèìïýò ðïõ âñÞêáôå áðü ôï BIOS, ÷ùñßæïíôáò ôïõò ìå êáèÝôïõò. Ãéá ðáñÜäåéãìá, ãéá 5000 êõëßíäñïõò, 250 êåöáëÝò êáé 60 ôïìåßò, èá ãñÜöáôå **`5000/250/60`**.

ÐéÝóôå enter ãéá íá ïñßóåôå ôéò ôéìÝò, êáé Ýðåéôá ôï **W** ãéá íá ãñÜøåôå ôï íÝï ðßíáêá êáôáôìÞóåùí óôïí ïäçãü.

28. ¸íá Üëëï ëåéôïõñãéêü óýóôçìá êáôÝóôñåøå ôïí äéá÷åéñéóôÞ åêêßíçóçò ìïõ. Ðùò ìðïñþ íá ôïí åðáíáöÝñù;

Èá ðñÝðåé íá åéóÝëèåôå óôï sysinstall(8) êáé íá åðéëÝîåôå Configure êáé êáôüðéí Fdisk. ÅðéëÝîôå ôï äßóêï óôïí ïðïßï âñßóêåôáé êáíïíéêÜ ï ÖïñôùôÞò Åêêßíçóçò ÷ñçóéìïðïéþíôáò ôï ðëÞêôñï **space**. ÐéÝóôå ôï **W** ãéá íá ãñÜøåôå ôéò áëëáãÝò óôïí ïäçãü. Èá åìöáíéóôåß ìéá ðñïôñïðÞ ðïõ èá óáò ñùôÜåé ðïéï öïñôùôÞ åêêßíçóçò íá åãêáôáóôÞóåé. ÊÜíôå ôçí áíôßóôïé÷ç åðéëïãÞ êáé ï öïñôùôÞò åêêßíçóçò èá áðïêáôáóôáèåß.

29. Ôé óçìáßíåé ôï ìÞíõìá ëÜèïõò “`swap_pager: indefinite wait buffer:`”;

Óçìáßíåé üôé ìéá äéåñãáóßá ðñïóðáèåß íá ãñÜøåé ìéá óåëßäá ìíÞìçò óôï äßóêï, êáé ç áðüðåéñá áõôÞ Ý÷åé êïëëÞóåé ðñïóðáèþíôáò íá áðïêôÞóåé ðñüóâáóç óôï äßóêï ãéá ðåñéóóüôåñï áðü 20 äåõôåñüëåðôá. Áõôü ìðïñåß íá óõìâåß áðü ÷áëáóìÝíïõò ôïìåßò óôï óêëçñü äßóêï, ðñïâëçìáôéêÜ êáëþäéá, Þ Üëëï õëéêü ôï ïðïßï íá ó÷åôßæåôáé ìå I/O. Áí ðñüêåéôáé ãéá ðñïâëçìáôéêü äßóêï, èá äåßôå åðßóçò êáé áíôßóôïé÷á ìçíýìáôá óôï `/var/log/messages` êáé óôçí Ýîïäï ôçò åíôïëÞò `dmesg`. ÄéáöïñåôéêÜ, åëÝãîôå ôéò óõíäÝóåéò êáé ôá êáëþäéá óáò.

30. Ôé åßíáé ôá óöÜëìáôá “UDMA ICRC”, êáé ðùò ìðïñþ íá ôá äéïñèþóù;

Ôï ðñüãñáììá ïäÞãçóçò ata(4) áíáöÝñåé óöÜëìáôá ôýðïõ “UDMA ICRC” üôáí åíôïðßóåé ðñüâëçìá óôçí ïñèüôçôá ôùí äåäïìÝíùí óå ìéá ìåôáöïñÜ DMA áðü Þ ðñïò ôïí ïäçãü. Ôï ðñüãñáììá ïäÞãçóçò èá ðñïóðáèÞóåé íá åðáíáëÜâåé ôç ìåôáöïñÜ ìåñéêÝò öïñÝò. Áí üëåò ïé áðüðåéñåò áðïôý÷ïõí, èá áëëÜîåé ôçí êáôÜóôáóç ëåéôïõñãßáò ôçò óõóêåõÞò áðü DMA óå PIO, ç ïðïßá åßíáé ðéï áñãÞ.

Ôï ðñüâëçìá ìðïñåß íá ðñïêëçèåß áðü ðïëëïýò ðáñÜãïíôåò, áí êáé ï ðéï óõíçèéóìÝíïò åßíáé ç ðñïâëçìáôéêÞ Þ ëáíèáóìÝíç êáëùäßùóç. ÅëÝãîôå üôé ôá êáëþäéá ÁÔÁ äåí Ý÷ïõí õðïóôåß æçìéÜ, êáé üôé åßíáé êáôÜëëçëá ðñïäéáãñáöþí ãéá ôçí êáôÜóôáóç ëåéôïõñãßáò Ultra DMA ðïõ ÷ñçóéìïðïéåßôå. Áí ÷ñçóéìïðïéåßôå áöáéñïýìåíá óõñôÜñéá äßóêùí, èá ðñÝðåé åðßóçò íá åßíáé óõìâáôÜ. Âåâáéùèåßôå üôé õðÜñ÷åé êáëÞ åðáöÞ óå üëåò ôéò óõíäÝóåéò. ¸÷ïõí åðßóçò áíáöåñèåß ðñïâëÞìáôá üôáí ãßíåôáé åãêáôÜóôáóç åíüò ðáëéïý ïäçãïý óôï ßäéï êáíÜëé DMA ìå Ýíá äßóêï Ultra DMA 66 (Þ ðéï ãñÞãïñï). ÔÝëïò, ôá ëÜèç áõôÜ ìðïñåß íá óçìáßíïõí üôé ï äßóêïò ðñüêåéôáé óýíôïìá íá ÷áëÜóåé. Ïé ðåñéóóüôåñïé êáôáóêåõáóôÝò äßóêùí ðáñÝ÷ïõí ëïãéóìéêü åëÝã÷ïõ ãéá ôïõò ïäçãïýò ôïõò, åëÝãîôå ëïéðüí ôï äßóêï óáò, êáé áí ÷ñåéÜæåôáé, ðÜñôå áíôßãñáöï ôùí äåäïìÝíùí óáò êáé áíôéêáôáóôÞóôå ôïí.

Ìðïñåßôå íá ÷ñçóéìïðïéÞóåôå ôï âïçèçôéêü ðñüãñáììá atacontrol(8) ãéá íá äåßôå êáé íá åðéëÝîåôå ôçí êáôÜóôáóç ëåéôïõñãßáò DMA Þ PIO ðïõ ÷ñçóéìïðïéåßôáé áðü êÜèå óõóêåõÞ ATA. Ðéï óõãêåêñéìÝíá, ç åíôïëÞ `atacontrol mode` *`channel`* èá óáò äåßîåé ôçí êáôÜóôáóç ëåéôïõñãßáò ôùí óõóêåõþí åíüò óõãêåêñéìÝíïõ êáíáëéïý ÁÔÁ, üðïõ ôï ðñùôåýïí êáíÜëé Ý÷åé ôçí áñßèìçóç 0 ê.ï.ê.

31. Ôé åßíáé ôï “`lock order reversal`”;

Ï Robert Watson <`rwatson@FreeBSD.org`> áðÜíôçóå ìå óáöÞíåéá áõôÞ ôçí åñþôçóç óôçí ëßóôá freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current), óå ìéá óõæÞôçóç ìå ôßôëï “lock order reversals - ôé óçìáßíïõí; (http://docs.freebsd.org/cgi/getmsg.cgi?fetch=65165+0+/usr/local/www/db/text/2003/freebsd-current/20031221.freebsd-current)”

> Ïé ðñïåéäïðïéÞóåéò áõôÝò ðñïÝñ÷ïíôáé áðü ôï Witness, Ýíá äéáãíùóôéêü óýóôçìá ãéá êëåéäþìáôá êáôÜ ôç ëåéôïõñãßá (run-time lock) ôï ïðïßï âñßóêåôáé óôïõò ðõñÞíåò -CURRENT ôïõ FreeBSD (áëëÜ áöáéñåßôáé óôéò åêäüóåéò). Ìðïñåßôå íá äéáâÜóåôå ðåñéóóüôåñá ãéá ôï Witness êáé ôéò äõíáôüôçôåò ôïõ, óôç óåëßäá manual witness(4). Ìåôáîý Üëëùí ôï Witness åðáëçèåýåé ôç óåéñÜ ôùí run-time locks ÷ñçóéìïðïéþíôáò Ýíá óõíäõáóìü áðü åíóùìáôùìÝíåò óåéñÝò êëåéäùìÜôùí êáèþò êáé áðü ôç óåéñÜ ðïõ áíé÷íåýåôáé êáôÜ ôçí åêôÝëåóç, êáé ðáñÜãåé ðñïåéäïðïéÞóåéò óôçí êïíóüëá üôáí ðáñáâéÜæïíôáé. Óêïðüò áõôÞò ôçò ëåéôïõñãßáò åßíáé íá áíé÷íåýïíôáé ðéèáíÜ deadlocks ôá ïðïßá ìðïñåß íá ïöåßëïíôáé óå ðáñáâéÜóåéò ôçò óåéñÜò ôùí êëåéäùìÜôùí. Åßíáé áîéïóçìåßùôï üôé ôï Witness åßíáé êÜðùò óõíôçñçôéêü, êáé åßíáé ðéèáíüí íá äþóåé ëÜèïò ðñïåéäïðïéÞóåéò. Óôçí ðåñßðôùóç ðïõ ôï Witness áíáöÝñåé Ýíá ðñáãìáôéêü ðñüâëçìá ìå ôçí óåéñÜ ôùí êëåéäùìÜôùí, åßíáé óáí íá ëÝåé "áí Þóáóôáí Üôõ÷ïò, èá óáò åß÷å óõìâåß deadlock óå áõôü ôï óçìåßï". ÕðÜñ÷ïõí êÜðïéåò ãíùóôÝò ðåñéðôþóåéò "ëáíèáóìÝíçò äéÜãíùóçò" ãéá ôéò ïðïßåò ÷ñåéÜæåôáé íá äçìéïõñãÞóïõìå êáëýôåñç ôåêìçñßùóç þóôå íá áðïöýãïõìå êáé ôéò ðåñéôôÝò áíáöïñÝò óöáëìÜôùí. Ïé ëéãüôåñï ãíùóôÝò ðåñéðôþóåéò ïöåßëïíôáé ðåñéóóüôåñï óå íÝá êëåéäþìáôá, êáèþò ïé áíôéóôïé÷ßåò óôç óåéñÜ ôùí êëåéäùìÜôùí äéïñèþíïíôáé ãñÞãïñá åðåéäÞ ôï Witness åßíáé ðÜíôá áðáó÷ïëçìÝíï êáé äçìéïõñãåß óõíÝ÷åéá íÝåò ðñïåéäïðïéÞóåéò :-).
>
> —Áðü ôïí Robert Watson <`rwatson@FreeBSD.org`> óôç ëßóôá freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current), óôéò 14 Äåêåìâñßïõ 2003

Óçìåßùóç: Áõôü ðïõ áðïêáëïýìå "ëáíèáóìÝíç äéÜãíùóç" äçìéïõñãåßôáé óôçí ðñáãìáôéêüôçôá üôáí ôï Witness âñßóêåé êÜðïéï ðïëý ðéï óïâáñü ëÜèïò. ÔÝôïéá ëÜèç åßíáé ôõðéêÜ ôï óöÜëìá óåëßäáò (page fault) Þ ëáíèáóìÝíá äåäïìÝíá óôç ìíÞìç ìÝóá óôïí ðõñÞíá, Þ ôÝëïò óýãêñïõóç ïíïìáóßáò ìå êÜðïéá mutexes.

Óçìåßùóç: ÄåßôÝ ôçí óåëßäá ôïõ Bjoern Zeeb ó÷åôéêÜ ìå ôéò áíôéóôñïöÝò êëåéäùìÜôùí (http://sources.zabbadoz.net/freebsd/lor.html) ãéá ôçí êáôÜóôáóç ôùí ãíùóôþí áíôéóôñïöþí.

32. Ôé óçìáßíåé ôï ìÞíõìá “`Called ... with the following non-sleepable locks held`”;

Óçìáßíåé üôé êëÞèçêå ìéá óõíÜñôçóç ìå äõíáôüôçôá sleep åíþ ôçí ßäéá óôéãìÞ Þôáí åíåñãü êÜðïéï êëåßäùìá mutex (Þ áíôßóôïé÷ï ÷ùñßò äõíáôüôçôá sleep).

Ï ëüãïò ãéá ôïí ïðïßï áõôü åßíáé ëÜèïò åßíáé åðåéäÞ ôá mutexes äåí ðñïïñßæïíôáé íá êñáôïýíôáé ãéá ìåãÜëá ÷ñïíéêÜ äéáóôÞìáôá. Åßíáé ìüíï ãéá ôç óõíôÞñçóç ìéêñþí ðåñéüäùí óõã÷ñïíéóìïý. ÁõôÞ ç ðñïãñáììáôéóôéêÞ óõìöùíßá åðéôñÝðåé óôïõò ïäçãïýò óõóêåõþí íá ÷ñçóéìïðïéïýí mutexes ãéá íá óõã÷ñïíßæïíôáé ìå ôá õðüëïéðá ðñïãñÜììáôá ôïõ ðõñÞíá êáôÜ ôçí äéÜñêåéá ôùí interrupts. Ôá interrupts (óôï FreeBSD) äåí ìðïñïýí íá ðåñéÝëèïõí óå êáôÜóôáóç sleep. Ãéá ôï ëüãï áõôü åßíáé áðáñáßôçôï íá ìçí ìðëïêÜñåôáé ï ðõñÞíáò ãéá ìåãÜëï äéÜóôçìá áðü êÜðïéï õðïóýóôçìá ðïõ êñáôÜåé Ýíá mutex.

Ãéá íá åíôïðéóôïýí áõôÜ ôá ëÜèç, ìðïñïýí íá ðñïóôåèïýí õðïèÝóåéò (assertions) óôïí ðõñÞíá ïé ïðïßåò áëëçëåðéäñïýí ìå ôï õðïóýóôçìá witness ãéá íá äþóïõí Ýíá ðñïåéäïðïéçôéêü ìÞíõìá (Þ ìÞíõìá ëÜèïõò, áíÜëïãá ìå ôéò ñõèìßóåéò ôïõ óõóôÞìáôïò) üôáí ãßíåôáé ìéá êëÞóç ç ïðïßá ðéèáíþò íá äçìéïõñãåß ìðëïêÜñéóìá ôçí óôéãìÞ ðïõ êñáôéÝôáé Ýíá mutex.

Åí óõíôïìßá, áõôïý ôïõ åßäïõò ïé ðñïåéäïðïéÞóåéò äåí åßíáé óõíÞèùò ìïéñáßåò, áëëÜ õðü ïñéóìÝíåò áôõ÷åßò ðñïûðïèÝóåéò, ìðïñåß íá ðñïêáëÝóïõí áíåðéèýìçôá öáéíüìåíá ôá ïðïßá êõìáßíïíôáé áðü ìéá óôéãìéáßá ðôþóç ôçí áðüêñéóç ôïõ óõóôÞìáôïò, ìÝ÷ñé ðëÞñçò êáôÜññåõóç.

33. Ãéáôß ç äéáäéêáóßá buildworld/installworld óôáìáôÜåé ìå ôï ìÞíõìá “`touch: not found`”;

Ôï ìÞíõìá áõôü äåí óçìáßíåé üôé óáò ëåßðåé ôï âïçèçôéêü ðñüãñáììá touch(1). Ôï ëÜèïò áõôü ðñïêáëåßôáé óõíÞèùò áðü ëáíèáóìÝíç, ìåëëïíôéêÞ, óÞìáíóç çìåñïìçíßáò ôùí áñ÷åßùí. Áí ôï ñïëüé CMOS ôïõ õðïëïãéóôÞ óáò åßíáé ñõèìéóìÝíï ãéá ôïðéêÞ þñá, ðñÝðåé íá åêôåëÝóåôå ôçí åíôïëÞ `adjkerntz -i` ãéá íá ñõèìßóåôå ôï ñïëüé ôïõ ðõñÞíá üôáí åêêéíåßôå óå êáôÜóôáóç ëåéôïõñãßáò åíüò ÷ñÞóôç.

ÊåöÜëáéï 6 ÅìðïñéêÝò ÅöáñìïãÝò

> **Óçìåßùóç:** This section is still very sparse, though we are hoping, of course, that companies will add to it! :) The FreeBSD group has no financial interest in any of the companies listed here but simply lists them as a public service (and feels that commercial interest in FreeBSD can have very positive effects on FreeBSD's long-term viability). We encourage commercial software vendors to send their entries here for inclusion. See the Vendors page (http://www.FreeBSD.org/commercial/index.html) for a longer list.

1. Where can I get an Office Suite for FreeBSD?

The open-source OpenOffice.org (http://www.openoffice.org) office suite works natively on FreeBSD. The Linux version of StarOffice (http://www.sun.com/staroffice/), the value-added closed-source version of OpenOffice.org, also works on FreeBSD.

FreeBSD also includes a variety of text editors, spreadsheets, and drawing programs in the Ports Collection.

2. Where can I get Motif® for FreeBSD?

The Open Group has released the source code to Motif 2.2.2. You can install the `open-motif` package, or compile it from ports. Refer to the ports section of the Handbook (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/ports.html) for more information on how to do this.

> **Óçìåßùóç:** The Open Motif distribution only allows redistribution if it is running on an open source (http://www.opensource.org/) operating system.

In addition, there are commercial distributions of the Motif software available. These, however, are not for free, but their license allows them to be used in closed-source software. Contact Apps2go for the least expensive ELF Motif 2.1.20 distribution for FreeBSD (either i386 or Alpha).

There are two distributions, the "development edition" and the "runtime edition" (for much less). These distributions includes:

- OSF/Motif manager, xmbind, panner, wsm.
- Development kit with uil, mrm, xm, xmcxx, include and Imake files.
- Static and dynamic ELF libraries.
- Demonstration applets.

Be sure to specify that you want the FreeBSD version of Motif when ordering (do not forget to mention the architecture you want too)! Versions for NetBSD and OpenBSD are also sold by *Apps2go*. This is currently a FTP only download.

More info

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

or

`<sales@apps2go.com>` or `<support@apps2go.com>`

or

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

Contact Xi Graphics for an a.out Motif 2.0 distribution for FreeBSD.

This distribution includes:

- 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 manual pages.

Be sure to specify that you want the FreeBSD version of Motif when ordering! Versions for BSDI and Linux are also sold by *Xi Graphics*. This is currently a 4 diskette set... in the future this will change to a unified CD distribution like their CDE.

3. Where can I get CDE for FreeBSD?

Xi Graphics used to sell CDE for FreeBSD, but no longer do.

KDE (http://www.kde.org/) is an open source X11 desktop which is similar to CDE in many respects. You might also like the look and feel of xfce (http://www.xfce.org/). KDE and xfce are both in the ports system (http://www.FreeBSD.org/ports/index.html).

4. Are there any commercial high-performance X servers?

Yes, Xi Graphics (http://www.xig.com/) sells Accelerated-X products for FreeBSD and other Intel based systems.

The Xi Graphics offering is a high performance X Server that offers easy configuration, support for multiple concurrent video boards and is distributed in binary form only, in a unified diskette distribution for FreeBSD and Linux. Xi Graphics also offers a high performance X Server tailored for laptop support.

There is a free “compatibility demo” of version 5.0 available.

Xi Graphics also sells Motif and CDE for FreeBSD (see above).

More info

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

or

`<sales@xig.com>` or `<support@xig.com>`

or

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

5. Are there any Database systems for FreeBSD?

Yes! See the Commercial Vendors (http://www.FreeBSD.org/commercial/software_bycat.html#CATEGORY_DATABASE) section of FreeBSD's Web site.

Also see the Databases (http://www.FreeBSD.org/ports/databases.html) section of the Ports collection.

6. Can I run Oracle® on FreeBSD?

Yes. The following pages tell you exactly how to set up Linux-Oracle on FreeBSD:

- http://www.unixcities.com/oracle/index.html (http://www.unixcities.com/oracle/index.html)
- http://www.shadowcom.net/freebsd-oracle9i/ (http://www.shadowcom.net/freebsd-oracle9i/)

ÊåöÜëáéï 7 ÅöáñìïãÝò Ôåëéêïý ×ñÞóôç

1. So, where are all the user applications?

Please take a look at the ports page (http://www.FreeBSD.org/ports/index.html) for info on software packages ported to FreeBSD. The list currently tops 20,000 and is growing daily, so come back to check often or subscribe to the `freebsd-announce` mailing list for periodic updates on new entries.

Most ports should work on the 4.X, 5.X, and 6.X branches. Each time a FreeBSD release is made, a snapshot of the ports tree at the time of release in also included in the `ports/` directory.

We also support the concept of a “package”, essentially no more than a compressed binary distribution with a little extra intelligence embedded in it for doing whatever custom installation work is required. A package can be installed and uninstalled again easily without having to know the gory details of which files it includes.

Use the package installation menu in `/stand/sysinstall` (under the post-configuration menu item) or invoke the pkg_add(1) command on the specific package files you are interested in installing. Package files can usually be identified by their `.tgz` or `.tbz` suffix and CDROM distribution people will have a `packages/All` directory on their CD which contains such files. They can also be downloaded over the net for various versions of FreeBSD at the following locations:

for 4.X-RELEASE/4-STABLE

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

for 5.X-RELEASE/5-STABLE

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

for 6.X-RELEASE/6-STABLE

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

for 7-CURRENT

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

or your nearest local mirror site.

Note that all ports may not be available as packages since new ones are constantly being added. It is always a good idea to check back periodically to see which packages are available at the ftp.FreeBSD.org (ftp://ftp.FreeBSD.org/pub/FreeBSD/) master site.

2. How do I configure INN (Internet News) for my machine?

After installing the `news/inn` package or port, an excellent place to start is Dave Barr’s INN Page (http://www.visi.com/~barr/INN.html) where you will find the INN FAQ.

3. Does FreeBSD support Java™?

Yes. Please see http://www.FreeBSD.org/java/ (http://www.FreeBSD.org/java/index.html).

4. Why can I not build this port on my 4.X-STABLE machine?

If you are running a FreeBSD version that lags significantly behind -CURRENT or -STABLE, you may need to update your ports collection; see the Keeping Up (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/porters-handbook/keeping-up.html) section of the Porter's Handbook for further information on how to do this. If you are up to date, then someone might have committed a change to the port which works for -CURRENT but which broke the port for -STABLE. Please submit a bug report on this with the send-pr(1) command, since the ports collection is supposed to work for both the -CURRENT and -STABLE branches.

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

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

However, if you are up-to-date, perhaps you are seeing another problem. `make index` has a known bug in dealing with incomplete copies of the Ports Collection. It assumes that you have a local copy of every single port that every other port that you have a local copy of depends on. To explain, if you have a copy of `foo/bar` on your disk, and `foo/bar` depends on `baz/quux`, then you must also have a copy of `baz/quux` on your disk, and the ports `baz/quux` depends on, and so on. Otherwise, `make index` has insufficient information to create its dependency tree.

This is particularly a problem for FreeBSD users who utilize cvsup(1) to track the Ports Collection but choose not to install certain categories by specifying them in `refuse`. In theory, one should be able to refuse categories, but in practice there are too many ports that depend on ports in other categories. Until someone comes up with a solution for this problem, the general rule is is that if you want to build `INDEX`, you must have a complete copy of the Ports Collection.

There are rare cases where `INDEX` will not build due to odd cases involving `WITH_*` or `WITHOUT_*` variables being set in `make.conf`. If you suspect that this is the case, please try to make `INDEX` with those Makevars turned off before reporting it to çëåêôñïíéêÞ ëßóôá ôùí FreeBSD ports (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports).

6. Why is CVSup not integrated in the main FreeBSD tree?

The FreeBSD base system is designed as self-hosting - it should be possible to build the whole operating system starting with a very limited set of tools. Thus, the actual build tools needed to compile the FreeBSD sources are bundled with the sources themselves. This includes a C compiler (gcc(1)), make(1), awk(1), and similar tools.

Since CVSup is written in Modula-3, adding it to the FreeBSD base system would also require adding and maintaining a Modula-3 compiler. This would lead to both an increase in the disk space consumed by the FreeBSD sources and additional maintenance work. Thus, it is much easier for both the developers and users to keep CVSup as a separate port, which can be easily installed as a package bundled on the FreeBSD installation CDs.

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

FreeBSD does not include a port upgrading tool, but it does have some tools to make the upgrade process somewhat easier. You can also install additional tools to simplify port handling.

The pkg_version(1) command can generate a script that will update installed ports to the latest version in the ports tree.

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

The output script *must* be edited by hand before you use it. Recent versions of pkg_version(1) force this by inserting an exit(1) at the beginning of the script.

You should save the output of the script, as it will note packages that depend on the one that has been updated. These may or may not need to be updated as well. The usual case where they need to be updated is that a shared library has changed version numbers, so the ports that used that library need to be rebuilt to use the new version.

> **Óçìåßùóç:** Beginning with FreeBSD 5.0 (and higher revisions), pkg_version(1) no longer supports the `-c` option.

If you have the disk space, you can use the `portupgrade` tool to automate all of this. `portupgrade` includes various tools to simplify package handling. It is available under `ports-mgmt/portupgrade`. Since it is written in Ruby, `portupgrade` is an unlikely candidate for integration with the main FreeBSD tree. That should not stop anyone from using it, however.

If your system is up full time, the periodic(8) system can be used to generate a weekly list of ports that might need updating by setting `weekly_status_pkg_enable="YES"` in `/etc/periodic.conf`.

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

Because POSIX® says that there shall be such a shell.

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

The existing implementation is our best effort at meeting as many of these requirements simultaneously as we can. In order to keep `/bin/sh` small, we have not provided many of the convenience features that other shells have. That is why the Ports Collection includes more featureful shells like bash, scsh, tcsh, and zsh. (You can compare for yourself the memory utilization of all these shells by looking at the “VSZ” and “RSS” columns in a `ps -u` listing.)

9. Why do Netscape and Opera take so long to start?

The usual answer is that DNS on your system is misconfigured. Both Netscape and Opera perform DNS checks when starting up. The browser will not appear on your desktop until the program either gets a response or determines that the system has no network connection.

10. I updated parts of the Ports Collection using CVSup, and now many ports fail to build with mysterious error messages! What happened? Is the Ports Collection broken in some major way?

If you only update parts of the Ports Collection, using one of its CVSup subcollections and not the `ports-all` CVSup collection, you should *always* update the `ports-base` subcollection too! The reasons are described in the

Handbook
(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/cvsup.html#CVSUP-COLLEC-PBASE-WARN).

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

To create audio CDs from MIDI files, first install `audio/timidity++` from ports then install manually the GUS patches set by Eric A. Welsh, available at http://www.stardate.bc.ca/eawpatches/html/default.htm. After timidity++ has been installed properly, midi files may be converted to wav files with the following command line:

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

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

Êåö�ëáéï 8 Ñýèìéóç ÐõñÞíá

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

Not at all! Check out the kernel config section of the Handbook (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/kernelconfig.html).

> **Óçìåßùóç:** We recommend that you make a dated snapshot of your new `/kernel` called `/kernel.YYMMDD` after you get it working properly. Also back up your new `/modules` directory to `/modules.YYMMDD`. That way, if you make a mistake the next time you play with your configuration you can boot the backup kernel instead of having to fall back to `kernel.GENERIC`. This is particularly important if you are now booting from a controller that GENERIC does not support.

2. My kernel compiles fail because `_hw_float` is missing. How do I solve this problem?

You probably removed `npx0` (see npx(4)) from your kernel configuration file because you do not have a math co-processor. The `npx0` device is *MANDATORY*. Somewhere inside your hardware lies a device that provides hardware floating-point support, even if it is no longer a separate device as used in the good old 386 days. You *must* include the `npx0` device. Even if you manage to build a kernel without `npx0` support, it will not boot anyway.

3. Why is my kernel so big (over 10MB)?

Chances are, you compiled your kernel in *debug mode*. Kernels built in debug mode contain many symbols that are used for debugging, thus greatly increasing the size of the kernel. Note that there will be little or no performance decrease from running a debug kernel, and it is useful to keep one around in case of a system panic.

However, if you are running low on disk space, or you simply do not want to run a debug kernel, make sure that both of the following are true:

- You do not have a line in your kernel configuration file that reads:

  ```
  makeoptions DEBUG=-g
  ```

- You are not running config(8) with the `-g` option.

Either of the above settings will cause your kernel to be built in debug mode. As long as you make sure you follow the steps above, you can build your kernel normally, and you should notice a fairly large size decrease; most kernels tend to be around 1.5MB to 2MB.

4. Why do I get interrupt conflicts with multi-port serial code?

When I compile a kernel with multi-port serial code, it tells me that only the first port is probed and the rest skipped due to interrupt conflicts. How do I fix this?

The problem here is that FreeBSD has code built-in to keep the kernel from getting trashed due to hardware or software conflicts. The way to fix this is to leave out the IRQ settings on all but one port. Here is an example:

```
#
```

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

There are a number of possible causes for this problem. They are, in no particular order:

- You are not using the new `make buildkernel` and `make installkernel` targets, and your source tree is different from the one used to build the currently running system (e.g., you are compiling 4.3-RELEASE on a 4.0-RELEASE system). If you are attempting an upgrade, please read the `/usr/src/UPDATING` file, paying particular attention to the “COMMON ITEMS” section at the end.
- You are using the new `make buildkernel` and `make installkernel` targets, but you failed to assert the completion of the `make buildworld` target. The `make buildkernel` target relies on files generated by the `make buildworld` target to complete its job correctly.
- Even if you are trying to build FreeBSD-STABLE, it is possible that you fetched the source tree at a time when it was either being modified, or broken for other reasons; only releases are absolutely guaranteed to be buildable, although FreeBSD-STABLE builds fine the majority of the time. If you have not already done so, try re-fetching the source tree and see if the problem goes away. Try using a different server in case the one you are using is having problems.

6. How can I verify which scheduler is in use on a running system?

If you are running FreeBSD version 5.2.1 or earlier, check for the existence of the `kern.quantum` sysctl. If you have it, you should see something like this:

```
% sysctl kern.quantum
kern.sched.quantum: 99960
```

If the `kern.quantum` sysctl exists, you are using the 4BSD scheduler. If not, you will get an error printed by sysctl(8) (which you can safely ignore):

```
% sysctl kern.sched.quantum
sysctl: unknown oid 'kern.sched.quantum'
```

In FreeBSD version 5.3-RELEASE and later, the name of the scheduler currently being used is directly available as the value of the `kern.sched.name` sysctl:

```
% sysctl kern.sched.name
kern.sched.name: 4BSD
```

7. What is `kern.quantum`?

`kern.quantum` is the maximum number of ticks a process can run without being preempted. It is specific to the 4BSD scheduler, so you can use its presence or absence to determine which scheduler is in use. In FreeBSD 5.X or later `kern.quantum` has been renamed to `kern.sched.quantum`.

8. What is `kern.sched.quantum`?

See Å: 7.

ÊåöÜëáéï 9 Äßóêïé, ÓõóôÞìáôá Áñ÷åßùí êáé ÖïñôùôÝò Åêêßíçóçò

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

See the Disk Formatting Tutorial at www.FreeBSD.org (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/formatting-media/index.html).

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

The best way is to reinstall the OS on the new disk, then move the user data over. 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 `/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. See the formatting-media article (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/formatting-media/index.html) for details on this process.

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.

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

The installation procedure allows you to chose 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 chose 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 it is 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.

4. 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.

5. 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`.

6. 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)
```

7. 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).

ReiserFS

FreeBSD includes a read-only ReiserFS driver. For more information, see mount_reiserfs(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)).

8. 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/da1s5`:

```
# cd /dev
# sh MAKEDEV da1s5
# mount -t msdosfs /dev/da1s5 /dos/e
```

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

9. Is there a cryptographic filesystem for FreeBSD?

Yes. FreeBSD 5.0 includes gbde(8), and FreeBSD 6.0 added geli(8). For earlier releases, see the `security/cfs` port.

10. 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 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.

> **Ðñïåéäïðïßçóç:** *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.

11. 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.

12. How do I boot FreeBSD and Linux using GRUB

Booting FreeBSD using GRUB is very simple. Just add the following to your configuration file `/boot/grub/grub.conf`.

```
title FreeBSD 6.1
 root (hd0,a)
 kernel /boot/loader
```

Where `hd0,a` points to your root partition on the first disk. If you need to specify which slice number should be used, use something like this `(hd0,2,a)`. By default, if the slice number is omitted, GRUB searches the first slice which has `'a'` partition.

13. 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).

14. 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.

15. 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 msdosfs /dev/fd0c /floppy
```

if it is a floppy, or this:

```
# mount -t msdosfs /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 `/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 `/stand/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
```

16. 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/el_GR.ISO8859-7/books/handbook/creating-cds.html), specifically the section Using Data CDs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html#MOUNTING-CD).

17. 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/el_GR.ISO8859-7/books/handbook/creating-cds.html#MOUNTING-CD) section of the Handbook for a detailed discussion of this issue.

18. 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/el_GR.ISO8859-7/books/handbook/creating-cds.html), specifically the section on Using Data CDROMs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html#MOUNTING-CD).

19. 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/el_GR.ISO8859-7/books/handbook/creating-cds.html), particularly the section on burning raw data CDs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html#RAWDATA-CD).

20. 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/el_GR.ISO8859-7/books/handbook/creating-cds.html#IMAGING-CD). For more on working with CDROMs, see the Creating CDs Section (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html) in the Storage chapter in the Handbook.

21. 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.

22. 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.

23. 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 msdosfs /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.

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

24. 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).

25. How can I add more swap space?

In the Configuration and Tuning (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/config-tuning.html) section of the Handbook, you will find a section (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/adding-swap-space.html) describing how to do this.

26. 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.

27. 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).

ÊåöÜëáéï 10 Äéá÷åßñéóç ÓõóôÞìáôïò

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

The primary configuration file is `/etc/defaults/rc.conf` (see rc.conf(5)) System startup scripts such as `/etc/rc` and `/etc/rc.d` (see rc(8)) just include this file. *Do not edit this file!* Instead, if there is any entry in `/etc/defaults/rc.conf` that you want to change, you should copy the line into `/etc/rc.conf` and change it there.

For example, if you wish to start named, the included DNS server, all you need to do is:

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

To start up local services, place shell scripts in the `/usr/local/etc/rc.d` directory. These shell scripts should be set executable, and end with a .sh.

2. How do I add a user easily?

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

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

3. Why do I keep getting messages like “`root: not found`” after editing my crontab file?

This is normally caused by editing the system crontab (`/etc/crontab`) and then using crontab(1) to install it:

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

This is not the correct way to do things. The system crontab has a different format to the per-user crontabs which crontab(1) updates (the crontab(5) manual page explains the differences in more detail).

If this is what you did, the extra crontab is simply a copy of `/etc/crontab` in the wrong format it. Delete it with the command:

```
# crontab -r
```

Next time, when you edit `/etc/crontab`, you should not do anything to inform cron(8) of the changes, since it will notice them automatically.

If you want something to be run once per day, week, or month, it is probably better to add shell scripts `/usr/local/etc/periodic`, and let the periodic(8) command run from the system cron schedule it with the other periodic system tasks.

The actual reason for the error is that the system crontab has an extra field, specifying which user to run the command as. In the default system crontab provided with FreeBSD, this is `root` for all entries. When this crontab is used as the `root` user’s crontab (which is *not* the same as the system crontab), cron(8) assumes the string `root` is the first word of the command to execute, but no such command exists.

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

This is a security feature. In order to su to `root` (or any other account with superuser privileges), you must be in the `wheel` group. If this feature were not there, anybody with an account on a system who also found out `root`'s password would be able to gain superuser level access to the system. With this feature, this is not strictly true; su(1) will prevent them from even trying to enter the password if they are not in `wheel`.

To allow someone to su to `root`, simply put them in the `wheel` group.

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

When you get the prompt to enter the shell pathname, simply press `ENTER`, and run `mount /` to re-mount the root filesystem in read/write mode. You may also need to run `mount -a -t ufs` to mount the filesystem where your favorite editor is defined. If your favorite editor is on a network filesystem, you will need to either configure the network manually before you can mount network filesystems, or use an editor which resides on a local filesystem, such as ed(1).

If you intend to use a full screen editor such as vi(1) or emacs(1), you may also need to run `export TERM=cons25` so that these editors can load the correct data from the termcap(5) database.

Once you have performed these steps, you can edit `/etc/rc.conf` as you usually would to fix the syntax error. The error message displayed immediately after the kernel boot messages should tell you the number of the line in the file which is at fault.

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

Please have a look at the Handbook entry on printing. It should cover most of your problem. See the Handbook entry on printing (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/printing.html).

Some printers require a host-based driver to do any kind of printing. These so-called "WinPrinters" are not natively supported by FreeBSD. If your printer does not work in DOS or Windows NT 4.0, it is probably a WinPrinter. Your only hope of getting one of these to work is to check if the `print/pnm2ppa` port supports it.

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

Please see the Handbook section on using localization (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/using-localization.html), specifically the section on console setup (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/using-localization.html#SETTING-CONSOLE).

8. Why do I get messages like: "`unknown: <PNP0303> can't assign resources`" on boot?

The following is an excerpt from a post to the freebsd-current mailing list.

> The "can't assign resources" messages indicate that the devices are legacy ISA devices for which a non-PnP-aware driver is compiled into the kernel. These include devices such as keyboard controllers, the programmable interrupt controller chip, and several other bits of standard infrastructure. The resources cannot be assigned because there is already a driver using those addresses.
>
> —Garrett Wollman <`wollman@FreeBSD.org`>, 24 April 2001

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

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

   ```
   options QUOTA
   ```

   Please read the Handbook entry on quotas (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/quotas.html) for full details.
2. Do not turn on quotas on `/`.
3. Put the quota file on the filesystem that the quotas are to be enforced on, i.e.:

Filesystem	Quota file
`/usr`	`/usr/admin/quotas`
`/home`	`/home/admin/quotas`
...	...

10. Does FreeBSD support System V IPC primitives?

Yes, FreeBSD supports System V-style IPC, including shared memory, messages and semaphores, in the GENERIC kernel. In a custom kernel, enable this support by adding the following lines to your kernel config.

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

Recompile and install your kernel.

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

Sendmail (http://www.sendmail.org/) is the default mail-server software for FreeBSD, but you can easily replace it with one of the other MTA (for instance, an MTA installed from the ports).

There are various alternative MTAs in the ports tree already, with `mail/exim`, `mail/postfix`, `mail/qmail`, and `mail/zmailer` being some of the most popular choices.

Diversity is nice, and the fact that you have many different mail-servers to chose from is considered a good thing; therefore try to avoid asking questions like "Is Sendmail better than Qmail?" in the mailing lists. If you do feel like asking, first check the mailing list archives. The advantages and disadvantages of each and every one of the available MTAs have already been discussed a few times.

12. I have forgotten the `root` password! What do I do?

Do not panic! Restart the system, type **`boot -s`** at the Boot: prompt to enter Single User mode. At the question about the shell to use, hit ENTER. You will be dropped to a `#` prompt. Enter `mount -u /` to remount your root filesystem read/write, then run `mount -a` to remount all the filesystems. Run `passwd root` to change the `root` password then run exit(1) to continue booting.

13. How do I keep **Control+Alt+Delete** from rebooting the system?

If you are using syscons (the default console driver) build and install a new kernel with the line:

```
options SC_DISABLE_REBOOT
```

in the configuration file. If you use the PCVT console driver, use the following kernel configuration line instead. This can also be done by setting the following sysctl which does not require a reboot or kernel recompile:

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

```
options PCVT_CTRL_ALT_DEL
```

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

Use this perl command:

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

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

Alternatively you can use the tr(1) command:

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

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

15. How do I kill processes by name?

Use killall(1).

16. Why is su bugging me about not being in `root`'s ACL?

The error comes from the Kerberos distributed authentication system. The problem is not fatal but annoying. You can either run su with the -K option, or uninstall Kerberos as described in the next question.

17. How do I uninstall Kerberos?

To remove Kerberos from the system, reinstall the bin distribution for the release you are running. If you have the CDROM, you can mount the cd (we will assume on /cdrom) and run

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

Alternately, you can remove all `MAKE_KERBEROS` options from `/etc/make.conf` and rebuild world.

18. What happened to `/dev/MAKEDEV`?

FreeBSD 5.X and beyond use the devfs(8) device-on-demand system. Device drivers automatically create new device nodes as they are needed, obsoleting `/dev/MAKEDEV`.

If you are running FreeBSD 4.X or earlier and `/dev/MAKEDEV` is missing, then you really do have a problem. Grab a copy from the system source code, probably in `/usr/src/etc/MAKEDEV`.

19. How do I add pseudoterminals to the system?

If you have lots of telnet, ssh, X, or screen users, you will probably run out of pseudoterminals. Here is how to add more:

1. Build and install a new kernel with the line

   ```
   pseudo-device pty 256
   ```

   in the configuration file.
2. Run the commands

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

   to make 256 device nodes for the new terminals.
3. Edit `/etc/ttys` and add lines for each of the 256 terminals. They should match the form of the existing entries, i.e. they look like

   ```
   ttyqc none network
   ```

   The order of the letter designations is `tty[pqrsPQRS][0-9a-v]`, using a regular expression.
4. Reboot the system with the new kernel and you are ready to go.

20. Why can I not create the snd0 device?

There is no `snd` device. The name is used as a shorthand for the various devices that make up the FreeBSD sound driver, such as `mixer`, `sequencer`, and `dsp`.

To create these devices you should

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

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

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

Go into single user mode and then back to multi user mode.

On the console do:

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

# return
# exit
```

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

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

Long answer: FreeBSD derives its releases from one of two places. Major, dot-zero, releases, such as 4.0-RELEASE and 5.0-RELEASE, are branched from the head of the development stream, commonly referred to as -CURRENT. Minor releases, such as 4.1-RELEASE or 5.2-RELEASE, have been snapshots of the active -STABLE branch. Starting with 4.3-RELEASE, each release also now has its own branch which can be tracked by people requiring an extremely conservative rate of development (typically only security advisories).

When a release is about to be made, the branch from which it will be derived from has to undergo a certain process. Part of this process is a code freeze. When a code freeze is initiated, the name of the branch is changed to reflect that it is about to become a release. For example, if the branch used to be called 4.5-STABLE, its name will be changed to 4.6-PRERELEASE to signify the code freeze and signify that extra pre-release testing should be happening. Bug fixes can still be committed to be part of the release. When the source code is in shape for the release the name will be changed to 4.6-RC to signify that a release is about to be made from it. Once in the RC stage, only the most critical bugs found can be fixed. Once the release (4.6-RELEASE in this example) and release branch have been made, the branch will be renamed to 4.6-STABLE.

For more information on version numbers and the various CVS branches, refer to the Release Engineering (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/releng/article.html) article.

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

Short answer: You are probably at security level greater than 0. Reboot directly to single user mode to install the kernel.

Long answer: FreeBSD disallows changing system flags at security levels greater than 0. You can check your security level with the command:

```
# sysctl kern.securelevel
```

You cannot lower the security level; you have to boot to single mode to install the kernel, or change the security level in `/etc/rc.conf` then reboot. See the init(8) manual page for details on securelevel, and see `/etc/defaults/rc.conf` and the rc.conf(5) manual page for more information on rc.conf.

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

Short answer: You are probably at security level greater than 1. Reboot directly to single user mode to change the date.

Long answer: FreeBSD disallows changing the time by more that one second at security levels greater than 1. You can check your security level with the command:

```
# sysctl kern.securelevel
```

You cannot lower the security level; you have to boot to single mode to change the date, or change the security level in `/etc/rc.conf` then reboot. See the init(8) manual page for details on securelevel, and see `/etc/defaults/rc.conf` and the rc.conf(5) manual page for more information on rc.conf.

25. Why is `rpc.statd` using 256 megabytes of memory?

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

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

26. Why can I not unset the `schg` file flag?

You are running at an elevated (i.e., greater than 0) securelevel. Lower the securelevel and try again. For more information, see the FAQ entry on securelevel and the init(8) manual page.

27. Why does SSH authentication through `.shosts` not work by default in recent versions of FreeBSD?

The reason why `.shosts` authentication does not work by default in more recent versions of FreeBSD is because ssh(1) is not installed suid `root` by default. To “fix” this, you can do one of the following:

- As a permanent fix, set `ENABLE_SUID_SSH` to `true` in `/etc/make.conf` and rebuild ssh (or run `make world`).
- As a temporary fix, change the mode on `/usr/bin/ssh` to `4555` by running `chmod 4555 /usr/bin/ssh` as `root`. Then add `ENABLE_SUID_SSH= true` to `/etc/make.conf` so the change takes effect the next time `make world` is run.

28. What is `vnlru`?

`vnlru` flushes and frees vnodes when the system hits the `kern.maxvnodes` limit. This kernel thread sits mostly idle, and only activates if you have a huge amount of RAM and are accessing tens of thousands of tiny files.

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

- `Active`: pages recently statistically used.
- `Inactive`: pages recently statistically unused.

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

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

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

30. How much free memory is available?

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

31. What is `/var/empty`? I can not delete it!

`/var/empty` is a directory that the sshd(8) program uses when performing privilege separation. The `/var/empty` directory is empty, owned by `root` and has the `schg` flag set.

Although it is not recommended to delete this directory, to do so you will need to unset the `schg` flag first. See the chflags(1) manual page for more information (and bear in mind the answer to the question on unsetting the schg flag).

ÊåöÜëáéï 11 Ôï Óýóôçìá X Windows êáé ïé ÅéêïíéêÝò Êïíóüëåò

1. What is the X Window System?

The X Window System (commonly `X11`) is the most widely available windowing system capable of running on UNIX or UNIX like systems, including FreeBSD. The X.Org Foundation (http://www.x.org) administers the X protocol standards (http://en.wikipedia.org/wiki/X_Window_System_core_protocol), with the current reference implementation, version 11 release 7.4, so you will often see references shortened to `X11`.

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

2. Which X implementations are available for FreeBSD?

Historically, the default implementation of X on FreeBSD has been XFree86 which is maintained by The XFree86 Project, Inc. (http://www.xfree86.org) This software was installed by default on FreeBSD versions up until 4.10 and 5.2. Although Xorg itself maintained an implementation during that time period, it was basically only provided as a reference platform, as it had suffered greatly from bitrot over the years.

However, early in 2004, some XFree86 developers left that project over issues including the pace of code changes, future directions, and interpersonal conflicts, and are now contributing code directly to Xorg instead. At that time, Xorg updated its source tree to the last XFree86 release before its subsequent licensing change (**XFree86 version 4.3.99.903**), incorporated many changes that had previously been maintained separately, and has released that software as **X11R6.7.0**. A separate but related project, freedesktop.org (http://www.freedesktop.org) (or `fd.o` for short), is working on rearchitecting the original XFree86 code to offload more work onto the graphics cards (with the goal of increased performance) and make it more modular (with the goal of increased maintainability, and thus faster releases as well as easier configuration). Xorg intends to incorporate the freedesktop.org changes in its future releases.

As of July 2004, in FreeBSD-CURRENT, XFree86 has been replaced with Xorg as the default implementation. The XFree86 ports (`x11/XFree86-4` and subports) remain in the ports collection. But Xorg is the default X11 implementation for FreeBSD 5.3 and later.

For further information, read the X11 (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/x11.html) section of the FreeBSD Handbook.

> **Óçìåßùóç:** The above describes the default X implementation installed. It is still possible to install either implementation by following the instructions in the entry for 20040723 in `/usr/ports/UPDATING`.

> **Ðñïåéäïðïßçóç:** It is not currently possible to mix-and-match pieces of each implementation; one must choose one or the other.

3. Will my existing applications run with the Xorg suite?

The Xorg software is written to the same X11R6 specification that XFree86 is, so basic applications should work unchanged. A few lesser-used protocols have been deprecated (`XIE`, `PEX`, and `lbxproxy`), but in the first two cases, the FreeBSD port of XFree86 did not support them either.

4. Why did the X projects split, anyway?

The answer to this question is outside the scope of this FAQ. Note that there are voluminous postings in various mailing list archives on the Internet; please use your favorite search engine to investigate the history instead of asking this question on the FreeBSD mailing lists. It may even be the case that only the participants will ever know for certain.

5. Why did FreeBSD choose to go with the Xorg ports by default?

The Xorg developers claim that their goal is to release more often and incorporate new features more quickly. If they are able to do so, this will be very attractive. Also, their software still uses the traditional X license, while XFree86 is now using their modified one.

> **Óçìåßùóç:** This decision is still controversial. Only time will tell which implementation proves technically superior. Each FreeBSD user should decide which they prefer.

6. I want to run X, how do I go about it?

If you would like to add X to an existing installation, you should use the `x11/xorg` meta-port, which will build and install all the necessary components.

Then read and follow the documentation on the xorgconfig(1) tool, which assists you in configuring Xorg for your particular graphics card/mouse/etc. You may also wish to examine the xorgcfg(1) tool, which provides a graphical interface to the X configuration process.

For further information, read the X11 (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/x11.html) section of the FreeBSD Handbook.

You may also wish to investigate the Xaccel server. See the section on Xi Graphics for more details.

7. I *tried* to run X, but I get an "`KDENABIO failed (Operation not permitted)`" error when I type `startx`. What do I do now?

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

So the question is what else you should do instead, and you basically have two choices: set your securelevel back down to zero (usually from `/etc/rc.conf`), or run xdm(1) at boot time (before the securelevel is raised).

See Å: 14. for more information about running xdm(1) at boot time.

8. Why does my mouse not work with X?

If you are using syscons (the default console driver), you can configure FreeBSD to support a mouse pointer on each virtual screen. In order to avoid conflicting with X, syscons supports a virtual device called `/dev/sysmouse`. All mouse events received from the real mouse device are written to the sysmouse device via moused. If you wish to use your mouse on one or more virtual consoles, *and* use X, see Å: 4. and set up moused.

Then edit `/etc/X11/xorg.conf` and make sure you have the following lines:

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

Some people prefer to use `/dev/mouse` under X. To make this work, `/dev/mouse` should be linked to `/dev/sysmouse` (see sysmouse(4)):

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

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

Yes.

You need to tell X that you have a 5 button mouse. To do this, simply add the lines `Buttons 5` and `ZAxisMapping 4 5` to the "InputDevice" section of `/etc/X11/xorg.conf`. For example, you might have the following "InputDevice" section in `/etc/X11/xorg.conf`.

ÐáñÜäåéãìá 11-1. "InputDevice" Section for Wheeled Mouse in Xorg configuration file

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

ÐáñÜäåéãìá 11-2. ".emacs" example for naive page scrolling with Wheeled Mouse (optional)

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

10. How do I use remote X displays?

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

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

```
% startx -listen_tcp
```

11. Why do X Window menus and dialog boxes not work right?

Try turning off the **Num Lock** key.

If your **Num Lock** key is on by default at boot-time, you may add the following line in the `Keyboard` section of the `/etc/X11/xorg.conf` file.

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

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

Virtual consoles, put simply, enable you to have several simultaneous sessions on the same machine without doing anything complicated like setting up a network or running X.

When the system starts, it will display a login prompt on the monitor after displaying all the boot messages. You can then type in your login name and password and start working (or playing!) on the first virtual console.

At some point, you will probably wish to start another session, perhaps to look at documentation for a program you are running or to read your mail while waiting for an FTP transfer to finish. Just do **Alt+F2** (hold down the **Alt** key and press the **F2** key), and you will find a login prompt waiting for you on the second "virtual console"! When you want to go back to the original session, do **Alt+F1**.

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

To enable more of them, edit `/etc/ttys` (see ttys(5)) and add entries for `ttyv4` to `ttyvc` after the comment on "Virtual terminals":

```
# 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
```

Use as many or as few as you want. The more virtual terminals you have, the more resources that are used; this can be important if you have 8MB RAM or less. You may also want to change the `secure` to `insecure`.

> **Óçìáíôéêü:** If you want to run an X server you *must* leave at least one virtual terminal unused (or turned off) for it to use. That is to say that if you want to have a login prompt pop up for all twelve of your Alt-function keys, you are out of luck - you can only do this for eleven of them if you also want to run an X server on the same machine.

The easiest way to disable a console is by turning it off. For example, if you had the full 12 terminal allocation mentioned above and you wanted to run X, you would change settings for virtual terminal 12 from:

```
ttyvb   "/usr/libexec/getty Pc"         cons25  on  secure
```

to:

```
ttyvb   "/usr/libexec/getty Pc"         cons25  off secure
```

If your keyboard has only ten function keys, you would end up with:

```
ttyv9   "/usr/libexec/getty Pc"         cons25  off secure
ttyva   "/usr/libexec/getty Pc"         cons25  off secure
ttyvb   "/usr/libexec/getty Pc"         cons25  off secure
```

(You could also just delete these lines.)

Next, the easiest (and cleanest) way to activate the virtual consoles is to reboot. However, if you really do not want to reboot, you can just shut down the X Window system and execute (as `root`):

```
# kill -HUP 1
```

It is imperative that you completely shut down X Window if it is running, before running this command. If you do not, your system will probably appear to hang/lock up after executing the kill command.

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

Use **Ctrl+Alt+F*n*** to switch back to a virtual console. **Ctrl+Alt+F1** would return you to the first virtual console.

Once you are back to a text console, you can then use **Alt+F*n*** as normal to move between them.

To return to the X session, you must switch to the virtual console running X. If you invoked X from the command line, (e.g., using `startx`) then the X session will attach to the next unused virtual console, not the text console from which it was invoked. If you have eight active virtual terminals then X will be running on the ninth, and you would use **Alt+F9** to return.

14. How do I start XDM on boot?

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

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

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

If you are to start `xdm` from `/etc/ttys`, there still is a chance of conflict between `xdm` and getty(8). One way to avoid this is to add the `vt` number in the `/usr/local/lib/X11/xdm/Xservers` file.

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

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

15. Why do I get "`Couldn't open console`" when I run xconsole?

If you start **X** with `startx`, the permissions on `/dev/console` will *not* get changed, resulting in things like `xterm -C` and `xconsole` not working.

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

In a nutshell, make sure an uncommented line of the form

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

is in `/etc/fbtab` (see fbtab(5)) and it will ensure that whomever logs in on `/dev/ttyv0` will own the console.

16. Before, I was able to run XFree86 as a regular user. Why does it now say that I must be `root`?

All X servers need to be run as `root` in order to get direct access to your video hardware. Older versions of XFree86 (<= 3.3.6) installed all bundled servers to be automatically run as `root` (setuid to `root`). This is obviously a security hazard because X servers are large, complicated programs. Newer versions of XFree86 do not install the servers setuid to `root` for just this reason.

Obviously, running an X server as the `root` user is not acceptable, nor a good idea security-wise. There are two ways to be able to use X as a regular user. The first is to use `xdm` or another display manager (e.g., `kdm`); the second is to use the `Xwrapper`.

`xdm` is a daemon that handles graphical logins. It is usually started at boot time, and is responsible for authenticating users and starting their sessions; it is essentially the graphical counterpart of getty(8) and login(1). For more information on `xdm` see the XFree86 documentation (http://www.xfree86.org/sos/resources.html), and the the FAQ entry on it.

`Xwrapper` is the X server wrapper; it is a small utility to enable one to manually run an X server while maintaining reasonable safety. It performs some sanity checks on the command line arguments given, and if they pass, runs the appropriate X server. If you do not want to run a display manager for whatever reason, this is for you. If you have installed the complete ports collection, you can find the port in `/usr/ports/x11/wrapper`.

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

Your mouse and the mouse driver may have somewhat become out of synchronization.

In rare cases the driver may erroneously report synchronization problem and you may see the kernel message:

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

and notice that your mouse does not work properly.

If this happens, disable the synchronization check code by setting the driver flags for the PS/2 mouse driver to 0x100. Enter *UserConfig* by giving the `-c` option at the boot prompt:

```
boot: -c
```

Then, in the *UserConfig* command line, type:

```
UserConfig> flags psm0 0x100
UserConfig> quit
```

18. Why does my PS/2 mouse from MouseSystems not work?

There have been some reports that certain model of PS/2 mouse from MouseSystems works only if it is put into the "high resolution" mode. Otherwise, the mouse cursor may jump to the upper-left corner of the screen every so often.

Specify the flags 0x04 to the PS/2 mouse driver to put the mouse into the high resolution mode. Enter *UserConfig* by giving the `-c` option at the boot prompt:

```
boot: -c
```

Then, in the *UserConfig* command line, type:

```
UserConfig> flags psm0 0x04
UserConfig> quit
```

See the previous section for another possible cause of mouse problems.

19. I want to install different X server.

FreeBSD versions prior 5.3 will use the default **XFree86 4.X**, while latter versions will default to **Xorg**. If you want to run a different X11 implementation than the default one, add the following line to `/etc/make.conf`, (if you do not have this file, create it):

```
X_WINDOW_SYSTEM=        xorg
```

This variable may be set to `xorg`, `xfree86-4`, or `xfree86-3`.

20. How do I reverse the mouse buttons?

Run the command `xmodmap -e "pointer = 3 2 1"` from your `.xinitrc` or `.xsession`.

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

FreeBSD have a feature to allow the display of "splash" screens during the boot messages. The splash screens currently must be a 256 color bitmap (`*.BMP`) or ZSoft PCX (`*.PCX`) file. In addition, they must have a resolution of

320x200 or less to work on standard VGA adapters. If you compile VESA support into your kernel, then you can use larger bitmaps up to 1024x768. The actual VESA support can either be compiled directly into the kernel with the `VESA` kernel config option or by loading the VESA kld module during bootup.

To use a splash screen, you need to modify the startup files that control the boot process for FreeBSD.

You need to create a `/boot/loader.rc` file that contains the following lines:

```
include /boot/loader.4th
start
```

and a `/boot/loader.conf` that contains the following:

```
splash_bmp_load="YES"
bitmap_load="YES"
```

This assumes you are using `/boot/splash.bmp` for your splash screen. If you would rather use a PCX file, copy it to `/boot/splash.pcx`, create a `/boot/loader.rc` as instructed above, and create a `/boot/loader.conf` that contains:

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

Now all you need is a splash screen. For that you can surf on over to the gallery at http://www.baldwin.cx/splash/.

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

Yes. All you need to do is use xmodmap(1) to define what function you wish them to perform.

Assuming all “Windows” keyboards are standard then the keycodes for the 3 keys are

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

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

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

You will probably have to re-start your window manager to see the result.

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

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

to your `~/.xinitrc`.

For example, you could map the 3 keys to be **F13**, **F14**, and **F15**, respectively. This would make it easy to map them to useful functions within applications or your window manager, as demonstrated further down.

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

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

If you use `fvwm2`, for example, you could map the keys so that **F13** iconifies (or de-iconifies) the window the cursor is in, **F14** brings the window the cursor is in to the front or, if it is already at the front, pushes it to the back, and **F15** pops up the main Workplace (application) menu even if the cursor is not on the desktop, which is useful if you do not have any part of the desktop visible (and the logo on the key matches its functionality).

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

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

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

The availability of 3D acceleration depends on the version of XFree86 or Xorg that you are using and the type of video chip you have. If you have an NVIDIA chip, you can use the binary drivers provided for FreeBSD on the Drivers (http://www.nvidia.com/content/drivers/drivers.asp) section of their website. For other cards with XFree86-4 or Xorg, including the Matrox G200/G400, ATI Rage 128/Radeon, and 3dfx Voodoo 3, 4, 5, and Banshee, information on hardware acceleration is available on the XFree86-4 Direct Rendering on FreeBSD (http://people.FreeBSD.org/~anholt/dri/) page.

ÊåöÜëáéï 12 Äéêôýùóç

1. Where can I get information on “diskless booting”?

“Diskless booting” means that the FreeBSD box is booted over a network, and reads the necessary files from a server instead of its hard disk. For full details, please read the Handbook entry on diskless booting (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-diskless.html)

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

Yes. Please see the Handbook entry on advanced networking (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/advanced-networking.html), specifically the section on routing and gateways (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-routing.html).

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

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

If you are using dialup to connect to the Internet user-mode ppp(8) contains a `-nat` option. If you run ppp(8) with the `-nat` option, set `gateway_enable` to *YES* in `/etc/rc.conf`, and configure your Windows machine correctly, this should work fine. For more information, please see the ppp(8) manual page or the Handbook entry on user PPP (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/userppp.html).

If you are using kernel-mode PPP or have an Ethernet connection to the Internet, you need to use natd(8). Please look at the natd (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-natd.html) section of the Handbook for a tutorial.

4. Does FreeBSD support SLIP and PPP?

Yes. See the manual pages for slattach(8), sliplogin(8), ppp(8), and pppd(8). ppp(8) and pppd(8) provide support for both incoming and outgoing connections, while sliplogin(8) deals exclusively with incoming connections, and slattach(8) deals exclusively with outgoing connections.

For more information on how to use these, please see the Handbook chapter on PPP and SLIP (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/ppp-and-slip.html).

If you only have access to the Internet through a “shell account”, you may want to have a look at the `net/slirp` package. It can provide you with (limited) access to services such as ftp and http direct from your local machine.

5. Does FreeBSD support NAT or Masquerading?

Yes. If you want to use NAT over a user PPP connection, please see the Handbook entry on user PPP (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/userppp.html). If you want to use NAT over some other sort of network connection, please look at the natd (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-natd.html) section of the Handbook.

6. How do I connect two FreeBSD systems over a parallel line using PLIP?

Please see the PLIP section (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-plip.html) of the Handbook.

7. Why can I not create a `/dev/ed0` device?

Because they are not necessary. In the Berkeley networking framework, network interfaces are only directly accessible by kernel code. Please see the `/etc/rc.network` file and the manual pages for the various network programs mentioned there for more information. If this leaves you totally confused, then you should pick up a book describing network administration on another BSD-related operating system; with few significant exceptions, administering networking on FreeBSD is basically the same as on SunOS™ 4.0 or Ultrix.

8. How can I set up Ethernet aliases?

If the alias is on the same subnet as an address already configured on the interface, then add `netmask 0xffffffff` to your ifconfig(8) command-line, as in the following:

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

Otherwise, just specify the network address and netmask as usual:

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

9. How do I get my 3C503 to use the other network port?

If you want to use the other ports, you will have to specify an additional parameter on the ifconfig(8) command line. The default port is `link0`. To use the AUI port instead of the BNC one, use `link2`. These flags should be specified using the ifconfig_* variables in `/etc/rc.conf` (see rc.conf(5)).

10. Why am I having trouble with NFS and FreeBSD?

Certain PC network cards are better than others (to put it mildly) and can sometimes cause problems with network intensive applications like NFS.

See the Handbook entry on NFS (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-nfs.html) for more information on this topic.

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

Some versions of the Linux NFS code only accept mount requests from a privileged port; try

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

12. Why can I not NFS-mount from a Sun box?

Sun™ workstations running SunOS 4.X only accept mount requests from a privileged port; try

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

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

The most frequent problem is not understanding the correct format of `/etc/exports`. Please review exports(5) and the NFS (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-nfs.html) entry in the Handbook, especially the section on configuring NFS (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-nfs.html#CONFIGURING-NFS).

14. Why am I having problems talking PPP to NeXTStep machines?

Try disabling the TCP extensions in `/etc/rc.conf` (see rc.conf(5)) by changing the following variable to NO:

```
tcp_extensions=NO
```

Xylogic’s Annex boxes are also broken in this regard and you must use the above change to connect through them.

15. How do I enable IP multicast support?

FreeBSD supports multicast host operations by default. If you want your box to run as a multicast router, you need to recompile your kernel with the `MROUTING` option and run mrouted(8). FreeBSD will start mrouted(8) at boot time if the flag `mrouted_enable` is set to `"YES"` in `/etc/rc.conf`.

MBONE tools are available in their own ports category, mbone (http://www.FreeBSD.org/ports/mbone.html). If you are looking for the conference tools `vic` and `vat`, look there!

16. Which network cards are based on the DEC PCI chipset?

Here is a list compiled by Glen Foster <`gfoster@driver.nsta.org`>, with some more modern additions:

Ðßíáêáò 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)

Vendor	Model
TopWare	TE-3500P
Znyx (2.2.x)	ZX312, ZX314, ZX342, ZX345, ZX346, ZX348
Znyx (3.x)	ZX345Q, ZX346Q, ZX348Q, ZX412Q, ZX414, ZX442, ZX444, ZX474, ZX478, ZX212, ZX214 (10mbps/hd)

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

You will probably find that the host is actually in a different domain; for example, if you are in foo.example.org and you wish to reach a host called `mumble` in the `example.org` domain, you will have to refer to it by the fully-qualified domain name, `mumble.example.org`, instead of just `mumble`.

Traditionally, this was allowed by BSD BIND resolvers. However the current version of **bind** (see named(8)) that ships with FreeBSD no longer provides default abbreviations for non-fully qualified domain names other than the domain you are in. So an unqualified host `mumble` must either be found as `mumble.foo.example.org`, or it will be searched for in the root domain.

This is different from the previous behavior, where the search continued across `mumble.example.org`, and `mumble.edu`. Have a look at RFC 1535 for why this was considered bad practice, or even a security hole.

As a good workaround, you can place the line

```
search foo.example.org example.org
```

instead of the previous

```
domain foo.example.org
```

into your `/etc/resolv.conf` file (see resolv.conf(5)). However, make sure that the search order does not go beyond the “boundary between local and public administration”, as RFC 1535 calls it.

18. Why do I get an error, “`Permission denied`”, for all networking operations?

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

If you had unintentionally misconfigured your system for firewalling, you can restore network operability by typing the following while logged in as `root`:

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

You can also set `firewall_type="open"` in `/etc/rc.conf`.

For further information on configuring a FreeBSD firewall, see the Handbook chapter (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/firewalls.html).

19. How much overhead does IPFW incur?

Please see the Handbook’s Firewalls (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/firewalls.html) section, specifically the section on

IPFW Overhead & Optimization (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/firewalls.html#IPFW-OVERHEAD).

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

Possibly because you want to do network address translation (NAT) and not just forward packets. A "fwd" rule does exactly what it says; it forwards packets. It does not actually change the data inside the packet. Say we have a rule like:

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

When a packet with a destination address of *`foo`* arrives at the machine with this rule, the packet is forwarded to *`10.0.0.1`*, but it still has the destination address of *`foo`*! The destination address of the packet is *not* changed to *`10.0.0.1`*. Most machines would probably drop a packet that they receive with a destination address that is not their own. Therefore, using a "fwd" rule does not often work the way the user expects. This behavior is a feature and not a bug.

See the FAQ about redirecting services, the natd(8) manual, or one of the several port redirecting utilities in the ports collection (http://www.FreeBSD.org/ports/index.html) for a correct way to do this.

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

You can redirect FTP (and other service) request with the `socket` package, available in the ports tree in category "sysutils". Simply replace the service's command line to call socket instead, like so:

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

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

22. Where can I get a bandwidth management tool?

There are three bandwidth management tools available for FreeBSD. dummynet(4) is integrated into FreeBSD as part of ipfw(4). ALTQ (http://www.csl.sony.co.jp/person/kjc/programs.html) is available for free on FreeBSD 4.X and has been integrated into FreeBSD 5.X as part of pf(4). Bandwidth Manager from Emerging Technologies (http://www.etinc.com/) is a commercial product.

23. Why do I get "`/dev/bpf0: device not configured`"?

You are running a program that requires the Berkeley Packet Filter (bpf(4)), but it is not in your kernel. Add this to your kernel config file and build a new kernel:

```
pseudo-device bpf        # Berkeley Packet Filter
```

On FreeBSD 4.X and earlier, you must also create the device node. After rebooting, go to the `/dev` directory and run:

```
# sh MAKEDEV bpf0
```

Please see the Handbook entry on device nodes (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/kernelconfig-nodes.html) for more information on managing devices.

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

Use the **SMBFS** toolset. It includes a set of kernel modifications and a set of userland programs. The programs and information are available as `net/smbfs` in the ports collection, or in the base system as of 4.5-RELEASE and later.

25. What are these messages about “icmp-response bandwidth limit 300/200 pps” in my log files?

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

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

The first number in the message tells you how many packets the kernel would have sent if the limit was not in place, and the second number tells you the limit. You can control the limit using the `net.inet.icmp.icmplim` sysctl variable like this, where `300` is the limit in packets per second:

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

If you do not want to see messages about this in your log files, but you still want the kernel to do response limiting, you can use the `net.inet.icmp.icmplim_output` sysctl variable to disable the output like this:

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

Finally, if you want to disable response limiting, you can set the `net.inet.icmp.icmplim` sysctl variable (see above for an example) to `0`. Disabling response limiting is discouraged for the reasons listed above.

26. What are these “`arp: unknown hardware address format`” error messages?

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

27. I have just installed CVSup but trying to execute it produces errors. What is wrong?

First, see if the error message you are receiving is like the one shown below.

```
/usr/libexec/ld-elf.so.1: Shared object "libXaw.so.6" not found
```

Errors like these are caused by installing the `net/cvsup` port on a machine which does not have the **XFree86** suite. If you want to use the GUI included with **CVSup** you will need to install **XFree86** now. Alternatively if you just wish to use **CVSup** from a command line you should delete the package previously installed. Then install the `net/cvsup-without-gui` port. This is covered in more detail in the CVSup section (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/cvsup.html) of the Handbook.

ÊåöÜëáéï 13 ÁóöÜëåéá

1. What is a sandbox?

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

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

  The process is said to be able to “play” inside the walls. That is, nothing the process does in regards to executing code is supposed to be able to breech the walls so you do not have to do a detailed audit of its code to be able to say certain things about its security.

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

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

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

  The most common way to accomplish this is to build a simulated environment in a subdirectory and then run the processes in that directory chroot’d (i.e. `/` for that process is this directory, not the real `/` of the system).

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

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

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

Every UNIX process is completely firewalled off from every other UNIX process. One process cannot modify the address space of another. This is unlike Windows where a process can easily overwrite the address space of any other, leading to a crash.

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

2. What is securelevel?

The securelevel is a security mechanism implemented in the kernel. Basically, when the securelevel is positive, the kernel restricts certain tasks; not even the superuser (i.e., `root`) is allowed to do them. At the time of this writing, the securelevel mechanism is capable of, among other things, limiting the ability to,

- unset certain file flags, such as `schg` (the system immutable flag),
- write to kernel memory via `/dev/mem` and `/dev/kmem`,
- load kernel modules, and
- alter firewall rules.

To check the status of the securelevel on a running system, simply execute the following command:

```
# sysctl kern.securelevel
```

The output will contain the name of the sysctl(8) variable (in this case, `kern.securelevel`) and a number. The latter is the current value of the securelevel. If it is positive (i.e., greater than 0), at least some of the securelevel's protections are enabled.

You cannot lower the securelevel of a running system; being able to do that would defeat its purpose. If you need to do a task that requires that the securelevel be non-positive (e.g., an `installworld` or changing the date), you will have to change the securelevel setting in `/etc/rc.conf` (you want to look for the `kern_securelevel` and `kern_securelevel_enable` variables) and reboot.

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

> **Ðñïåéäïðïßçóç:** Securelevel is not a silver bullet; it has many known deficiencies. More often than not, it provides a false sense of security.
>
> One of its biggest problems is that in order for it to be at all effective, all files used in the boot process up until the securelevel is set must be protected. If an attacker can get the system to execute their code prior to the securelevel being set (which happens quite late in the boot process since some things the system must do at start-up cannot be done at an elevated securelevel), its protections are invalidated. While this task of protecting all files used in the boot process is not technically impossible, if it is achieved, system maintenance will become a nightmare since one would have to take the system down, at least to single-user mode, to modify a configuration file.
>
> This point and others are often discussed on the mailing lists, particularly the çëåêôñïíéêÞ ëßóôá ôïõ FreeBSD ãéá èÝìáôá áóöÜëåéáò (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security). Please search the archives here (http://www.FreeBSD.org/search/index.html) for an extensive discussion. Some people are hopeful that securelevel will soon go away in favor of a more fine-grained mechanism, but things are still hazy in this respect.
>
> Consider yourself warned.

3. BIND (`named`) is listening on port 53 and some other high-numbered port. What is going on?

BIND uses a random high-numbered port for outgoing queries. If you want to use port 53 for outgoing queries, either to get past a firewall or to make yourself feel better, you can try the following in `/etc/namedb/named.conf`:

```
options {
        query-source address * port 53;
};
```

You can replace the `*` with a single IP address if you want to tighten things further.

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

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

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

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

Do not worry. `toor` is an “alternative” superuser account (toor is root spelt backwards). Previously it was created when the bash(1) shell was installed but now it is created by default. It is intended to be used with a non-standard shell so you do not have to change `root`’s default shell. This is important as shells which are not part of the base distribution (for example a shell installed from ports or packages) are likely to be installed in `/usr/local/bin` which, by default, resides on a different filesystem. If `root`’s shell is located in `/usr/local/bin` and `/usr` (or whatever filesystem contains `/usr/local/bin`) is not mounted for some reason, `root` will not be able to log in to fix a problem (although if you reboot into single user mode you will be prompted for the path to a shell).

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

6. Why is `suidperl` not working properly?

For security reasons, `suidperl` is installed without the suid bit by default. The system administrator can enable suid behavior with the following command.

```
# chmod u+s /usr/bin/suidperl
```

If you want `suidperl` to be built suid during upgrades from source, edit `/etc/make.conf` and add `ENABLE_SUIDPERL=true` before you run `make buildworld`.

ÊåöÜëáéï 14 PPP

1. I cannot make ppp(8) work. What am I doing wrong?

You should first read the ppp(8) manual page and the PPP section of the handbook (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/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.

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.example.com foo localhost
```

Otherwise, simply add another entry for your host. Consult the relevant manual 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 manual 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.

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 and you have omitted the line saying

```
delete ALL
```

from `ppp.conf`. If this is the case, go back to the Final system configuration (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/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 (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/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 **ppp**'s 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:

- If the problem is remote, read on entry Å: 9..
- If the problem is local, read on entry Å: 10..

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 Somers <`brian@FreeBSD.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
```

> **Óçìåßùóç:** 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 and ACK (the first) back to `B`. In the meantime, `B` sends back two more ACKs in response to the two additional REQs sent by `A` before `B` started up. `B` then receives the first ACK from `A` and enters the OPENED state. `A` receives the second ACK from `B` and goes back to the REQ-SENT state, sending another (forth) REQ as per the RFC. It then receives the third ACK and enters the OPENED state. In the meantime, `B` receives the forth REQ from `A`, resulting in it reverting to the ACK-SENT state and sending another (second) REQ and (forth) ACK as per the RFC. `A` gets the REQ, goes into REQ-SENT and sends another REQ. It immediately receives the following ACK and enters OPENED.

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

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

```
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 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.

15. 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.

16. 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 using email with a dialup connection (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/smtp-dialup.html) in the FreeBSD Handbook 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')dnl
```

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).

17. 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
```

18. 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.

19. 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 manual 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
```

20. 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.

21. 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 manual 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.

22. 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.

23. 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
  ```
- **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
  ```

24. 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.

25. Why do Mac OS 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. Mac OS 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, Mac OS does not provide an interface for changing TCP/IP settings. However, there is commercial software available, such as OTAdvancedTuner (OT for OpenTransport, the Mac OS TCP/IP stack) by Sustainable Softworks (http://www.softworks.com/), that will allow users to customize TCP/IP settings. Mac OS 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.

26. 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 çëåêôñïíéêÞ ëßóôá ãåíéêþí åñùôÞóåùí ôïõ FreeBSD (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) or the comp.unix.bsd.freebsd.misc (news:comp.unix.bsd.freebsd.misc) news group, and someone should point you in the right direction.

ÊåöÜëáéï 15 ÓåéñéáêÝò Åðéêïéíùíßåò

This section answers common questions about serial communications with FreeBSD. PPP and SLIP are covered in the Networking 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. 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 `ttydX` for dial-ins. When opening `/dev/ttydX` 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 `ttydX` device. If the port is available, it “steals” it from the `ttydX` 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.

4. 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`).

5. Can FreeBSD handle multiport serial cards sharing irqs?

Not yet. You will have to use a different irq for each card.

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

The `ttyd`*`X`* (or `cuaa`*`X`*) 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.

7. 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
```

8. 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`).

9. 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
```

10. My stock Hayes modem is not supported---what can I do?

Actually, the manual 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.

11. 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 -sspeed
```

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.

12. 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=\@
```

13. 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`.

14. 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.

15. 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.

16. 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.

17. 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
```

18. 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**.

19. 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.

20. 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.

ÊåöÜëáéï 16 ÄéÜöïñåò ÅñùôÞóåéò

1. FreeBSD uses far more swap space than Linux. Why?

FreeBSD only appears to use more swap than Linux. In actual fact, it does not. The main difference between FreeBSD and Linux in this regard is that FreeBSD will proactively move entirely idle, unused pages of main memory into swap in order to make more main memory available for active use. Linux tends to only move pages to swap as a last resort. The perceived heavier use of swap is balanced by the more efficient use of main memory.

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

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

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

3. Why will `chmod` not change the permissions on symlinks?

Symlinks do not have permissions, and by default, chmod(1) will not follow symlinks to change the permissions on the target file. So if you have a file, `foo`, and a symlink to that file, `bar`, then this command will always succeed.

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

However, the permissions on `foo` will not have changed.

You have to use either `-H` or `-L` together with the `-R` option to make this work. See the chmod(1) and symlink(7) manual pages for more info.

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

4. Can I run DOS binaries under FreeBSD?

Yes, you can use `emulators/doscmd`, a DOS emulation program, available in the FreeBSD Ports Collection.

> **Óçìåßùóç:** The **doscmd** program used to be an integrated part of FreeBSD, but was removed before the release of FreeBSD 5.3.

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

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

See the Translation FAQ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/fdp-primer/translations.html) in the FreeBSD Documentation Project Primer.

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

The FreeBSD.org mail system implements some of the stricter Postfix checks on incoming mail and rejects mail that is either misconfigured or is potential spam. Your mail might bounce for one of the following reasons:

- The email is being sent from a known spam domain or IP block.

  The FreeBSD mail servers reject email from known spam sources. If you have service through a company or domain who generates or relays spam, please switch to a service provider who does not.

- The body of the email only contains HTML.

  Mail should be sent in plain text only. Please configure your mail user agent to send plain text.

- The mailer at FreeBSD.org cannot resolve the IP address of the connecting host back to a symbolic name.

  Working reverse DNS is a standard requirement for accepting mail from a host. Set up reverse DNS for your mail server's IP address. Many home services (DSL, cable, dialup, etc.) will not give you this option. In this case, relay your email through your service provider's mail server.

- The hostname given in the EHLO/HELO part of the SMTP exchange cannot be resolved to an IP address.

  A fully qualified, resolvable host name is necessary in this part of the SMTP dialogue before mail will be accepted. If you do not have a host name that is registered in the DNS, then you should use your service provider's mail server to relay your mail.

- Your message had a message ID ending with the string "localhost".

  Some mail user agents generate bad message IDs which will not be accepted. You will need to persuade your mail user agent to generate a valid message ID or else configure your mail transfer agent to rewrite them.

7. Where can I find a free FreeBSD account?

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

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

Grex (http://www.grex.org/) provides a site very similar to M-Net including the same bulletin board and interactive chat software. However, the machine is a Sun 4M and is running SunOS.

8. What is `sup`, and how do I use it?

SUP (http://www.FreeBSD.org/cgi/ports.cgi?^sup) stands for Software Update Protocol, and was developed by CMU for keeping their development trees in sync. We used it to keep remote sites in sync with our central development sources.

SUP is not bandwidth friendly, and has been retired. The current recommended method to keep your sources up to date is CVSup (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/synching.html#CVSUP)

9. What is the cute little red guy's name?

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

You can learn more about the BSD daemon on his home page (http://www.mckusick.com/beastie/index.html).

10. Can I use the BSD daemon image?

Perhaps. The BSD daemon is copyrighted by Marshall Kirk McKusick. You will want to check his Statement on the Use of the BSD Daemon Figure (http://www.mckusick.com/beastie/mainpage/copyright.html) for detailed usage terms.

In summary, you are free to use the image in a tasteful manner, for personal use, so long as appropriate credit is given. If you want to use him commercially, you must contact Kirk McKusick. More details are available on the BSD Daemon's home page (http://www.mckusick.com/beastie/index.html).

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

You will find eps and Xfig drawings under `/usr/share/examples/BSD_daemon/`.

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

Please see the FreeBSD Glossary (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/freebsd-glossary.html).

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

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

The longer and more complete answer is that after a very long argument about whether sleep(1) should take fractional second arguments, Poul-Henning Kamp <`phk@FreeBSD.org`> posted a long message entitled “A bike shed (any color will do) on greener grass...

(http://www.FreeBSD.org/cgi/getmsg.cgi?fetch=506636+517178+/usr/local/www/db/text/1999/freebsd-hackers/19991003.freebsd-hackers)". The appropriate portions of that message are quoted below.

> "What is it about this bike shed?" Some of you have asked me.
>
> It is a long story, or rather it is an old story, but it is quite short actually. C. Northcote Parkinson wrote a book in the early 1960s, called "Parkinson's Law", which contains a lot of insight into the dynamics of management.
>
> [snip a bit of commentary on the book]
>
> In the specific example involving the bike shed, the other vital component is an atomic power-plant, I guess that illustrates the age of the book.
>
> Parkinson shows how you can go into the board of directors and get approval for building a multi-million or even billion dollar atomic power plant, but if you want to build a bike shed you will be tangled up in endless discussions.
>
> Parkinson explains that this is because an atomic plant is so vast, so expensive and so complicated that people cannot grasp it, and rather than try, they fall back on the assumption that somebody else checked all the details before it got this far. Richard P. Feynmann gives a couple of interesting, and very much to the point, examples relating to Los Alamos in his books.
>
> A bike shed on the other hand. Anyone can build one of those over a weekend, and still have time to watch the game on TV. So no matter how well prepared, no matter how reasonable you are with your proposal, somebody will seize the chance to show that he is doing his job, that he is paying attention, that he is *here*.
>
> In Denmark we call it "setting your fingerprint". It is about personal pride and prestige, it is about being able to point somewhere and say "There! *I* did that." It is a strong trait in politicians, but present in most people given the chance. Just think about footsteps in wet cement.
>
> —Poul-Henning Kamp <`phk@FreeBSD.org`> on freebsd-hackers, October 2, 1999

ÊåöÜëáéï 17 ×éïýìïñ êáé FreeBSD

1. How cool is FreeBSD?

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

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

Seriously, both FreeBSD and Linux use the HLT (halt) instruction when the system is idle thus lowering its energy consumption and therefore the heat it generates. Also if you have APM (advanced power management) configured, then FreeBSD can also put the CPU into a low power mode.

2. Who is scratching in my memory banks??

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

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

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

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

One thousand, one hundred and sixty-nine:

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

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

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

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

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

Five to complain about buildworld being broken;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

One to complain that the new lightbulb lacks fairings;

Nine (including the PR originators) to ask “what is MFC?”;

Fifty-seven to complain about the lights being out two weeks after the bulb has been changed.

Nik Clayton <`nik@FreeBSD.org`> adds:

I was laughing quite hard at this.

And then I thought, “Hang on, shouldn’t there be ’1 to document it.’ in that list somewhere?”

And then I was enlightened :-)

4. Where does data written to `/dev/null` go?

It goes into a special data sink in the CPU where it is converted to heat which is vented through the heatsink / fan assembly. This is why CPU cooling is increasingly important; as people get used to faster processors, they become careless with their data and more and more of it ends up in `/dev/null`, overheating their CPUs. If you delete `/dev/null` (which effectively disables the CPU data sink) your CPU may run cooler but your system will quickly become constipated with all that excess data and start to behave erratically. If you have a fast network connection you can cool down your CPU by reading data out of `/dev/random` and sending it off somewhere; however you run the risk of overheating your network connection and / or angering your ISP, as most of the data will end up getting converted to heat by their equipment, but they generally have good cooling, so if you do not overdo it you should be OK.

Paul Robinson adds:

There are other methods. As every good sysadmin knows, it is part of standard practice to send data to the screen of interesting variety to keep all the pixies that make up your picture happy. Screen pixies (commonly mis-typed or re-named as “pixels” are categorized by the type of hat they wear (red, green or blue) and will hide or appear (thereby showing the color of their hat) whenever they receive a little piece of food. Video cards turn data into pixie-food, and then send them to the pixies - the more expensive the card, the better the food, so the better behaved the pixies are. They also need constant stimulation - this is why screen savers exist.

To take your suggestions further, you could just throw the random data to console, thereby letting the pixies consume it. This causes no heat to be produced at all, keeps the pixies happy and gets rid of your data quite quickly, even if it does make things look a bit messy on your screen.

Incidentally, as an ex-admin of a large ISP who experienced many problems attempting to maintain a stable temperature in a server room, I would strongly discourage people sending the data they do not want out to the network. The fairies who do the packet switching and routing get annoyed by it as well.

ÊåöÜëáéï 18 Ðñï÷ùñçìÝíá ÈÝìáôá

1. How can I learn more about FreeBSD's internals?

At this time, there is only one book on FreeBSD-specific OS internals, namely "The Design and Implementation of the FreeBSD Operating System" by Marshall Kirk McKusick and George V. Neville-Neil, ISBN 0-201-70245-2, which focuses on version 5.X of FreeBSD.

Additionally, much general UNIX knowledge is directly applicable to FreeBSD.

For a list of relevant books, please check the Handbook's Operating System Internals Bibliography (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/bibliography-osinternals.html).

2. How can I contribute to FreeBSD?

Please see the article on Contributing to FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/contributing/article.html) for specific advice on how to do this. Assistance is more than welcome!

3. What are SNAPs and RELEASEs?

There are currently three active/semi-active branches in the FreeBSD CVS Repository (http://www.FreeBSD.org/cgi/cvsweb.cgi). (Earlier branches are only changed very rarely, which is why there are only three active branches of development):

- `RELENG_5` AKA *5-STABLE*
- `RELENG_6` AKA *6-STABLE*
- `HEAD` AKA *-CURRENT* AKA *7.X-CURRENT*

`HEAD` is not an actual branch tag, like the other two; it is simply a symbolic constant for "*the current, non-branched development stream*" which we simply refer to as "-CURRENT".

Right now, "-CURRENT" is the 7.X development stream; the `5-STABLE` branch, RELENG_5, forked off from "-CURRENT" in October 2004, and the `6-STABLE` branch, RELENG_6, forked off from "-CURRENT" in November 2005.

4. How do I make my own custom release?

Please see the Release Engineering (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/releng/article.html) article.

5. Why does `make world` clobber my existing installed binaries?

Yes, this is the general idea; as its name might suggest, `make world` rebuilds every system binary from scratch, so you can be certain of having a clean and consistent environment at the end (which is why it takes so long).

If the environment variable `DESTDIR` is defined while running `make world` or `make install`, the newly-created binaries will be deposited in a directory tree identical to the installed one, rooted at `${DESTDIR}`. Some random combination of shared libraries modifications and program rebuilds can cause this to fail in `make world` however.

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` 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. Why does my system say "(bus speed defaulted)" when it boots?

The Adaptec 1542 SCSI host adapters allow the user to configure their bus access speed in software. Previous versions of the 1542 driver tried to determine the fastest usable speed and set the adapter to that. We found that this breaks some users' systems, so you now have to define the TUNE_1542 kernel configuration option in order to have this take place. Using it on those systems where it works may make your disks run faster, but on those systems where it does not, your data could be corrupted.

8. Can I follow -CURRENT with limited Internet access?

Yes, you can do this *without* downloading the whole source tree by using the CTM facility (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/synching.html#CTM).

9. How did you split the distribution into 240k files?

Newer BSD based systems have a `-b` option to split(1) that allows them to split files on arbitrary byte boundaries.

Here is an example from `/usr/src/Makefile`.

```
bin-tarball:
(cd ${DISTDIR}; \
tar cf - . \
gzip --no-name -9 -c | \
split -b 240640 - \
${RELEASEDIR}/tarballs/bindist/bin_tgz.)
```

10. I have written a kernel extension, who do I send it to?

Please take a look at the article on Contributing to FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/contributing/article.html) to learn how to submit code.

And thanks for the thought!

11. How are Plug N Play ISA cards detected and initialized?

By: Frank Durda IV <`uhclem@nemesis.lonestar.org`>

In a nutshell, there a few I/O ports that all of the PnP boards respond to when the host asks if anyone is out there. So when the PnP probe routine starts, it asks if there are any PnP boards present, and all the PnP boards respond with their model # to a I/O read of the same port, so the probe routine gets a wired-OR "yes" to that question. At least one bit will be on in that reply. Then the probe code is able to cause boards with board model IDs (assigned by Microsoft/Intel) lower than X to go "off-line". It then looks to see if any boards are still responding to the query. If the answer was `0`, then there are no boards with IDs above X. Now probe asks if there are any boards below `X`. If so, probe knows there are boards with a model numbers below X. Probe then asks for boards greater than X-(limit/4) to go off-line. If repeats the query. By repeating this semi-binary search of IDs-in-range enough times, the probing code will eventually identify all PnP boards present in a given machine with a number of iterations that is much lower than what 2^64 would take.

The IDs are two 32-bit fields (hence 2ˆ64) + 8 bit checksum. The first 32 bits are a vendor identifier. They never come out and say it, but it appears to be assumed that different types of boards from the same vendor could have different 32-bit vendor ids. The idea of needing 32 bits just for unique manufacturers is a bit excessive.

The lower 32 bits are a serial #, Ethernet address, something that makes this one board unique. The vendor must never produce a second board that has the same lower 32 bits unless the upper 32 bits are also different. So you can have multiple boards of the same type in the machine and the full 64 bits will still be unique.

The 32 bit groups can never be all zero. This allows the wired-OR to show non-zero bits during the initial binary search.

Once the system has identified all the board IDs present, it will reactivate each board, one at a time (via the same I/O ports), and find out what resources the given board needs, what interrupt choices are available, etc. A scan is made over all the boards to collect this information.

This info is then combined with info from any ECU files on the hard disk or wired into the MLB BIOS. The ECU and BIOS PnP support for hardware on the MLB is usually synthetic, and the peripherals do not really do genuine PnP. However by examining the BIOS info plus the ECU info, the probe routines can cause the devices that are PnP to avoid those devices the probe code cannot relocate.

Then the PnP devices are visited once more and given their I/O, DMA, IRQ and Memory-map address assignments. The devices will then appear at those locations and remain there until the next reboot, although there is nothing that says you cannot move them around whenever you want.

There is a lot of oversimplification above, but you should get the general idea.

Microsoft took over some of the primary printer status ports to do PnP, on the logic that no boards decoded those addresses for the opposing I/O cycles. I found a genuine IBM printer board that did decode writes of the status port during the early PnP proposal review period, but MS said “tough”. So they do a write to the printer status port for setting addresses, plus that use that address + `0x800`, and a third I/O port for reading that can be located anywhere between `0x200` and `0x3ff`.

12. Can you assign a major number for a device driver I have written?

FreeBSD-CURRENT after February 2003 has a facility for dynamically and automatically allocating major numbers for device drivers at runtime. This mechanism is highly preferred to the older procedure of statically allocating device numbers. Some comments on this subject can be found in `src/sys/conf/majors`.

If you are forced for some reason to use a static major number, the procedure for obtaining one depends on whether or not you plan on making the driver publicly available. If you do, then please send us a copy of the driver source code, plus the appropriate modifications to `files.i386`, a sample configuration file entry, and the appropriate MAKEDEV(8) code to create any special files your device uses. If you do not, or are unable to because of licensing restrictions, then character major number 32 and block major number 8 have been reserved specifically for this purpose; please use them. In any case, we would appreciate hearing about your driver on the çëåêôñïíéêÞ ëßóôá ôå÷íéêþí óõæçôÞóåùí ôïõ FreeBSD (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers).

13. What about alternative layout policies for directories?

In answer to the question of alternative layout policies for directories, the scheme that is currently in use is unchanged from what I wrote in 1983. I wrote that policy for the original fast filesystem, and never revisited it. It works well at keeping cylinder groups from filling up. As several of you have noted, it works poorly for find. Most filesystems are created from archives that were created by a depth first search (aka ftw). These directories end up being striped across the cylinder groups thus creating a worst possible scenario for future depth first searches. If one knew the total number of directories to be created, the solution would be to create (total / fs_ncg) per cylinder group before moving on. Obviously, one would have to create some heuristic to guess at this number. Even using a small fixed number like say 10 would make an order of magnitude improvement. To differentiate restores from normal operation (when the current algorithm is probably more sensible), you could use the clustering of up to 10 if they were all done within a ten second window. Anyway, my conclusion is that this is an area ripe for experimentation.

Kirk McKusick, September 1998

14. How can I make the most of the data I see when my kernel panics?

[This section was extracted from a mail written by Bill Paul <`wpaul@FreeBSD.org`> on the freebsd-current mailing list by Dag-Erling C. Smørgrav <`des@FreeBSD.org`>, who fixed a few typos and added the bracketed comments]

```
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 Rosengart posted the following panic message]

```
> 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
```

[When] you see a message like this, it is not enough to just reproduce it and send it in. The instruction pointer value that I highlighted up there is important; unfortunately, it is also configuration dependent. In other words, the value varies depending on the exact kernel image that you are using. If you are using a GENERIC kernel image from one of the snapshots, then it is possible for somebody else to track down the offending function, but if you are running a custom kernel then only *you* can tell us where the fault occurred.

What you should do is this:

1. Write down the instruction pointer value. Note that the `0x8:` part at the beginning is not significant in this case: it is the `0xf0xxxxxx` part that we want.
2. When the system reboots, do the following:

   ```
   % nm -n /kernel.that.caused.the.panic | grep f0xxxxxx
   ```

   where `f0xxxxxx` is the instruction pointer value. The odds are you will not get an exact match since the symbols in the kernel symbol table are for the entry points of functions and the instruction pointer address will be somewhere inside a function, not at the start. If you do not get an exact match, omit the last digit from the instruction pointer value and try again, i.e.:

   ```
   % nm -n /kernel.that.caused.the.panic | grep f0xxxxx
   ```

   If that does not yield any results, chop off another digit. Repeat until you get some sort of output. The result will be a possible list of functions which caused the panic. This is a less than exact mechanism for tracking down the point of failure, but it is better than nothing.

I see people constantly show panic messages like this but rarely do I see someone take the time to match up the instruction pointer with a function in the kernel symbol table.

The best way to track down the cause of a panic is by capturing a crash dump, then using gdb(1) to generate a stack trace on the crash dump.

In any case, the method I normally use is this:

1. Set up a kernel config file, optionally adding `options DDB` if you think you need the kernel debugger for something. (I use this mainly for setting breakpoints if I suspect an infinite loop condition of some kind.)
2. Use `config -g` *`KERNELCONFIG`* to set up the build directory.
3. `cd /sys/compile/`*`KERNELCONFIG`*`; make`
4. Wait for kernel to finish compiling.
5. `make install`
6. reboot

The make(1) process will have built two kernels. `kernel` and `kernel.debug`. `kernel` was installed as `/kernel`, while `kernel.debug` can be used as the source of debugging symbols for gdb(1).

To make sure you capture a crash dump, you need edit `/etc/rc.conf` and set `dumpdev` to point to your swap partition. This will cause the rc(8) scripts to use the dumpon(8) command to enable crash dumps. You can also run dumpon(8) manually. After a panic, the crash dump can be recovered using savecore(8); if `dumpdev` is set in `/etc/rc.conf`, the rc(8) scripts will run savecore(8) automatically and put the crash dump in `/var/crash`.

> **Óçìåßùóç:** FreeBSD crash dumps are usually the same size as the physical RAM size of your machine. That is, if you have 64MB of RAM, you will get a 64MB crash dump. Therefore you must make sure there is enough space in `/var/crash` to hold the dump. Alternatively, you run savecore(8) manually and have it recover the crash dump to another directory where you have more room. It is possible to limit the size of the crash dump by using `options MAXMEM=(foo)` to set the amount of memory the kernel will use to something a little more sensible. For example, if you have 128MB of RAM, you can limit the kernel's memory usage to 16MB so that your crash dump size will be 16MB instead of 128MB.

Once you have recovered the crash dump, you can get a stack trace with gdb(1) as follows:

```
% gdb -k /sys/compile/KERNELCONFIG/kernel.debug /var/crash/vmcore.0
(gdb) where
```

Note that there may be several screens worth of information; ideally you should use script(1) to capture all of them. Using the unstripped kernel image with all the debug symbols should show the exact line of kernel source code where the panic occurred. Usually you have to read the stack trace from the bottom up in order to trace the exact sequence of events that lead to the crash. You can also use gdb(1) to print out the contents of various variables or structures in order to examine the system state at the time of the crash.

Now, if you are really insane and have a second computer, you can also configure gdb(1) to do remote debugging such that you can use gdb(1) on one system to debug the kernel on another system, including setting breakpoints, single-stepping through the kernel code, just like you can do with a normal user-mode program. I have not played with this yet as I do not often have the chance to set up two machines side by side for debugging purposes.

[Bill adds: "I forgot to mention one thing: if you have DDB enabled and the kernel drops into the debugger, you can force a panic (and a crash dump) just by typing 'panic' at the ddb prompt. It may stop in the debugger again during the panic phase. If it does, type 'continue' and it will finish the crash dump." -ed]

15. Why has dlsym() stopped working for ELF executables?

The ELF toolchain does not, by default, make the symbols defined in an executable visible to the dynamic linker. Consequently `dlsym()` searches on handles obtained from calls to `dlopen(NULL, flags)` will fail to find such symbols.

If you want to search, using `dlsym()`, for symbols present in the main executable of a process, you need to link the executable using the `-export-dynamic` option to the ELF linker (ld(1)).

16. How can I increase or reduce the kernel address space?

By default, the kernel address space is 256 MB on FreeBSD 3.X and 1 GB on FreeBSD 4.X. If you run a network-intensive server (e.g. a large FTP or HTTP server), you might find that 256 MB is not enough.

So how do you increase the address space? There are two aspects to this. First, you need to tell the kernel to reserve a larger portion of the address space for itself. Second, since the kernel is loaded at the top of the address space, you need to lower the load address so it does not bump its head against the ceiling.

The first goal is achieved by increasing the value of `NKPDE` in `src/sys/i386/include/pmap.h`. Here is what it looks like for a 1 GB address space:

```
#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
```

To find the correct value of `NKPDE`, divide the desired address space size (in megabytes) by four, then subtract one for UP and two for SMP.

To achieve the second goal, you need to compute the correct load address: simply subtract the address space size (in bytes) from 0x100100000; the result is 0xc0100000 for a 1 GB address space. Set LOAD_ADDRESS in `src/sys/i386/conf/Makefile.i386` to that value; then set the location counter in the beginning of the section listing in `src/sys/i386/conf/kernel.script` to the same value, as follows:

```
OUTPUT_FORMAT("elf32-i386", "elf32-i386", "elf32-i386")
OUTPUT_ARCH(i386)
ENTRY(btext)
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)    }
```

Then reconfig and rebuild your kernel. You will probably have problems with ps(1) top(1) and the like; `make world` should take care of it (or a manual rebuild of `libkvm`, ps(1) and top(1) after copying the patched `pmap.h` to `/usr/include/vm/`.

NOTE: the size of the kernel address space must be a multiple of four megabytes.

[David Greenman <`dg@FreeBSD.org`> adds: *I think the kernel address space needs to be a power of two, but I am not certain about that. The old(er) boot code used to monkey with the high order address bits and I think expected at least 256MB granularity.]*

ÊåöÜëáéï 19 Åõ÷áñéóôßåò

This innocent little Frequently Asked Questions document has been written, rewritten, edited, folded, spindled, mutilated, eviscerated, contemplated, discombobulated, cogitated, regurgitated, rebuilt, castigated, and reinvigorated over the last decade, by a cast of hundreds if not thousands. Repeatedly.

We wish to thank every one of the people responsible, and we encourage you to to join them (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/contributing/article.html) in making this FAQ even better.

Bibliography

FreeBSD Unleashed, Michael Urban êáé 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, êáé John Quarterman, Addison-Wesley, Reading, 1996, ISBN 0-201-54979-4.

The Design and Implementation of the FreeBSD Operating System, M. K. McKusick êáé 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, êáé 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.

The FreeBSD Handbook, FreeBSD Documentation Project, BSDi, 1st edition, November 1999, 489 pages, ISBN 1-57176-241-8.

[McKusick et al, 1994] *Berkeley Software Architecture Manual, 4.4BSD Edition*, M. K. McKusick, M. J. Karels, S. J. Leffler, W. N. Joy, êáé 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, êáé 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 êáé Cricket Liu, 1998, 3rd edition.

Sendmail, O'Reilly & Associates, Inc, 1997, 2nd edition, Brian Costales, ISBN 1-56592-222-0.

Essential System Administration, Æleen 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 êáé Guy L. Jr. Steele, ISBN 0-13-326224-3.

The C Programming Language, Prentice Hall, 1998, Brian Kernighan êáé 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 êáé W. Richard Stevens, 1995, Addison-Wesley, ISBN 0-201-63354-X.

Firewalls and Internet Security: Repelling the Wily Hacker, William R. CHeswick êáé Steven M. Bellovin, Addison-Wesley, 1995, ISBN 0-201-63357-4.

Practical UNIX Security, Simson Garfinkel êáé 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 êáé 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, êáé Simon Garfinkel, IDG Books Worldwide, Inc, 1994, ISBN 1-56884-203-1.

Life with UNIX — special edition, Don Libes êáé Sandy Ressler, Prentice-Hall, 1989, ISBN 0-13-536657-7.

The BSD Family Tree (ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/FreeBSD-current/src/share/misc/bsd-family-tree), 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.