

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

Ç ìÜäá Ôåêçñßúóçò ôïõ FreeBSD

Íé ēÝíāēð Þ ñŰóāēð IBM, AIX, EtherJet, Netfinity, OS/2, PowerPC, PS/2, S/390, ēāē ThinkPad ā́bíāē ā̀ðĩñēēŰ óýĩāīēā òçð International Business Machines Corporation óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð, Űēēāð ÷þñāð, Þ ēāē óðā ā̀ŷř óāóðũ÷ñřřā.

Íé ēÝíāēð IEEE, POSIX, ēāē 802 ā́bíāē ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā ðřō Institute of Electrical and Electronics Engineers, Inc. óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð.

Íé ēÝíāēð Intel, Celeron, EtherExpress, i386, i486, Itanium, Pentium, ēāē Xeon ā́bíāē ā̀ðĩñēēŰ óýĩāīēā Þ ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā òçð Intel Corporation ēāē ðũř ððāāóñēēþřř òçð óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð ēāē óā Űēēāð ÷þñāð.

Íé ēÝíāēð Iomega, Zip, ēāē Jaz ā́bíāē ā́āðā ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā Þ ā̀ðĩñēēŰ óýĩāīēā òçð Iomega Corporation óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð Þ/ēāē óā Űēēāð ÷þñāð.

Ōř Linux ā́bíāē Ýřā ēāðř÷ðñũĩÝřā ā̀ðĩñēēũ óýĩāīēř ðřō Linus Torvalds óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð.

Íé ēÝíāēð Microsoft, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media, ēāē Windows NT ā́bíāē ā́āðā ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā Þ ā̀ðĩñēēŰ óýĩāīēā òçð Microsoft Corporation óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð ēāē/Þ óā Űēēāð ÷þñāð.

Íé ēÝíāēð MIPS ēāē R4000 ā́bíāē ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā òçð MIPS Technologies, Inc. óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð ēāē óā Űēēāð ÷þñāð.

Íé ēÝíāēð Netscape ēāē Netscape Navigator ā́bíāē ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā òçð Netscape Communications Corporation óðēð Ç.Ð.Á ēāē Űēēāð ÷þñāð.

Íé ēÝíāēð Motif, OSF/1, ēāē UNIX ā́bíāē ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā ēāē řē ēÝíāēð Þ ñŰóāēð IT DialTone ēāē The Open Group ā́bíāē ā̀ðĩñēēŰ óýĩāīēā ðřō The Open Group óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð ēāē óā Űēēāð ÷þñāð.

Ç ēÝřç Oracle ā́bíāē ēāðř÷ðñũĩÝřř ā̀ðĩñēēũ óýĩāīēř òçð Oracle Corporation.

Íé ēÝíāēð řē ñŰóāēð Silicon Graphics, SGI, ēāē OpenGL ā́bíāē ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā òçð Silicon Graphics, Inc. óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð ēāē/Þ óā Űēēāð ÷þñāð óðřř ēũóřř.

Íé ēÝíāēð Þ ñŰóāēð SPARC, SPARC64, SPARCengine, ēāē UltraSPARC ā́bíāē ā̀ðĩñēēŰ óýĩāīēā òçð SPARC International, Inc. óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð ēāē óā Űēēāð ÷þñāð. Ç SPARC International Inc. ēāðÝ÷āē ũēā óā ā̀ðĩñēēŰ āēēāēþřřāðā ðũř óðřāũēũř SPARC ēāē āðēðñÝðāē òçř řñēÞ ÷ñÞóç ðřðð áðũ ũēā óā řÝēç òçð, ēāðũðēř ó÷āðēēÞð Űāāēāð.

Íé ēÝíāēð Þ ñŰóāēð Sun, Sun Microsystems, Java, Java Virtual Machine, JavaServer Pages, JDK, JRE, JSP, JVM, Netra, OpenJDK, Solaris, StarOffice, Sun Blade, Sun Enterprise, Sun Fire, SunOS, Ultra ēāē VirtualBox ā́bíāē ā̀ðĩñēēŰ óýĩāīēā Þ ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā òçð Sun Microsystems, Inc. óðēð ÇřũĩÝřāð Ðřēēðā́ā́āð ēāē óā Űēēāð ÷þñāð.

Íé ēÝíāēð Þ ñŰóāēð U.S. Robotics ēāē Sportster ā́bíāē ēāðř÷ðñũĩÝřā ā̀ðĩñēēŰ óýĩāīēā òçð U.S. Robotics Corporation.

Ç ēÝřç XFree86 ā́bíāē Ýřā ā̀ðĩñēēũ óýĩāīēř ðřō The XFree86 Project, Inc.

ÐřēēÝð áðũ ðēð ēÝíāēð Þ ñŰóāēð řē řðřāð ÷ñçóēřřðřēřřřāē áðũ ðřðð ēāðāóēāðāóðÝð Þ ðřðð ðũēçðÝð ðřðð āēā řā āēāēñßñřřř óā ðñřũũřřā ðřðð ēāũñřřřřāē ā̀ðĩñēēŰ óýĩāīēā. ¼ðřð áððÝð ā̀ðāřřæřřřāē óā áððũ ðř ēāßřāñ ēāē āēā ũóāð áðũ áððÝð āřũñßæāē ç řřŰāā Ářřðððřçð ðřð FreeBSD ũðē ā́bíāē ðēēāñřř řā ā́bíāē ā̀ðĩñēēŰ óýĩāīēā, ēā āāßðā Ýřā áðũ óā óýĩāīēā: “TM” Þ “©”.

ΘΒίαεαò Ðañéα ÷ ïñÝíuí

1	ΆέόάãùāÐ	1
2	Ôâêîçñßùόç êáé ÔðíóôÐñέíç	7
3	ΆãêáoŨóóáoç	12
4	Óðîâáóüüόçôá Œêéēý	22
4.1	ÃâíéēŨ	22
4.2	Ãñ ÷ éóâêôîíéēÝò êáé ÅðáíññãáoóÝò	22
4.3	Óêêñîñß äßóêîé, ïñŨãâò ôáéíßáò, ïäçãñß CD êáé DVD	23
4.4	Ðêçêôññüüáéá êáé ðñíðßêéá	24
4.5	ÓóóêáoÝò óáéñéáêêðò äðéêîéíñíßáò êáé Äéêôýùόç	27
4.6	ÓóóêáoÝò Ð ÷ ïð	28
4.7	Œêêî ðéêéü	29
5	Áíôêîâððéόç ÐññâçñŨóñí	31
6	ÃðññééÝò ÅóáñññãÝò	46
7	ÅóáñññãÝò Ôâêéēý × ñÐóόç	49
8	Œýêîéόç ÐðñÐñá	53
9	Ãßóêîé, ÓóóðÐñáðá Ãñ ÷ äßñí êáé ŒñòòòÝò Äêêßççóç	56
10	Äéá ÷ äßñéόç ÓóóðÐñáðñò	66
11	Œñ Óýóóçñá X Windows êáé ïé ÄéêñééÝò Êñóñêâò	74
12	Äéêôýùόç	83
13	ÁóóŨêáéá	89
14	PPP	92
15	ÓáéñéáêêÝò Äðéêîéíñíßáò	104
16	ÄéŨóññâò ÅñòòÐóáéò	111
17	×êýññ êáé FreeBSD	115
18	Ðññ ÷ ùñçñÝñá ÈÝñáðá	118
19	Ãð ÷ áñéóóðßáò	125
	Bibliography	126

ÊáôÛëĩãïò ÐéíÛêùí

3-1. ÌÝãéóôá ïããÝèç áñ÷ãßùí.....	19
12-1. Network cards based on the DEC PCI chipset	85

Ê ã ö Ü ë á é ĩ 1 Å é ó á ã ù ã Þ

Êáëbò Pëèàôå óôèò Óõ÷íÝò ÅñùôPóåèò ôiõ FreeBSD 6.x-7.x!

[illegible]

÷ áé éáóááéçèàß éÜëà ðñíóðÜèáéá þóóá áóðú òí FAQ íá ðáñéÝ ÷ áé úóí òí äóíáóúí ðáñéóóúðáñáð ðëçñíóíñßáð. Áí éÝëáðá íá éÜíáðá éÜðíëáð ððíááßíáéð ò ÷ áóééÜ íá ðñúðíðð ááéðòßúçðð ðíð, óáð ðáñáéáéýíýá óóáßèðá íáð Ýíá email ððç çëáéðñííééß éßðóá ñÜááð ðáèìçñßúçðð ðíð FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc>).

1. Ôé åßíáé ôï FreeBSD:

Αί δοϊόνβα, οι FreeBSD άβίάέ Υία έάέοϊόνάέέυ όύόόçía όύðϊό UNIX® άέά όέό άñ÷έόάέόϊίέέΥò AMD64 έάέ Intel® EM64T, i386™ PC-98, IA-64, ARM®, PowerPC® έάέ UltraSPARC®. Άάόβαάόάέ όόçí Υέάϊόç “4.BSD-Lite” όϊό Δάíάέόόçíβϊό όϊό Berkeley, í έÜðϊέάό άάέόéþόάέό άðu όϊ “4.BSD-Lite2”. Άάόβαάόάέ άðβόçò Υííάόά όόçí άñάάόβα íάόάοíñÜò (port) όϊό “Net/2” (άðβόçò άðu όϊ Berkeley) ðϊό Υάέíá άðu όíí William Jolitz (ç ίðíβα Υάέíá άñυόð íά όí ύíñá “386BSD”), άí έάέ άάí Υ÷άέ íάβίάέ ó÷άάññ όðβϊόά άðu όíí άñ÷έέύ éþάέéά όϊό 386BSD. ΔέçñΥόόάñç ðάñέάñάóþ άέά όι όέ άβίάέ όϊ FreeBSD έάέ ðùò ίðíñάβ íά έάέόϊόνάþόάέ άέά άóÜò, ίðíñάβόά íά άñάβόά όόçí άέέόðάéþ όϊðïέάόβα όϊó FreeBSD (<http://www.FreeBSD.org/index.html>).

Οι FreeBSD ÷ñçοέιιιήέáßòáé áðu áòáέñßàò, ðáññ÷áßò òðçñáóéþí Internet (ISPs), áñáñçòçÝð, áðáááèíáòßàò òçò
 Ðçññòìééßò, òìéòçòÝð éáé ìéééáéñýð ÷ñßòáò áðu ùέí òñ èùòñ, óòç áñòéáéÛ òñòò, óòçí áéðáßááòóò éáé áéá
 áíáòð÷ß.

Άέá ðānéóóúðāñāð ēāðōñŸñāēāð ò÷āōēēŸ īā òī FreeBSD, ðāñāēāēŸŸŸā āēāāŸōōā òī Åā÷āēñBāēī òīō FreeBSD
 (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/index.html).

2. Ðiëiò åbíáé ĩ óëiðüò ôiõ FreeBSD Project:

[illegible][illegible]

3. ÕđŮñ÷ĩōĩ êŮđĩéíé đăñéíñéóĩĩß óôçĩ Ůăăéá ôĩō FreeBSD;

Íáε. Íε δαḡεḡεḡεḡεḡε ḡḡεḡε ḡḡε ḡεÝα÷ḡεḡε δḡḡ ÷ḡḡḡεḡεḡεḡεḡε ḡεḡ εḡεεεα, ḡεεḲḡḡḡ δḡḡ ḡḡḡα÷ḡεḡεḡεḡεḡε ḡεḡ ḡεḡεḡε ḡεḡ FreeBSD Project. Ḳḡ ḡḡḡ ḡḡεεεḡḡḡε ḡεḡḡḡḡ Ḳḡ Ḳḡḡεεα, ḡεεḡḡḡḡ ḡḡḡ ḡḡḡ (http://www.FreeBSD.org/copyright/freebsd-license.html). Ḳεε ḡεḡḡ ḡḡḡḡ δḡḡḡḡḡḡḡḡḡ, ḡ Ḳḡḡεεα δḡḡεεḡḡḡḡḡḡ ḡḡḡḡḡε:

- Ìçí éó÷÷õñéòéàßòâ ùòé ãñÛøáòâ áðòù òì ëìáéòíéèù.
- Ìç ìáò ìçíýóáòâ áí ÷áéÛóáé.

4. Iðinnáð öi FreeBSD íá áíóééâáóáôÞóáé öi öñÝ÷íí eäeöiöñäeü iüö óýóôçíá;

Ãéá ôĩõð ðãñéóóüôãñĩõð áíẽñþðĩõð, íáé. ÁëÛ áõôÞ ç áðÛíôçóç äáí ìðĩñåß íá äĩèåß ôüóí îãñÛ.

Ἰε δᾱνεόσουδᾱνιε Ὑρενυδιε ἄλ ÷ η̅ζοέιηδιεῖρῖ ο̅δζῖ δᾱᾱἰᾱδᾱέεϋδζοᾱ εϋδιεῖ εᾱέοῖῶηᾱέεϋ ο̅γοδζῖᾱ. × η̅ζοέιηδιεῖρῖ ἄοᾱηῖᾱ Ὑδ. Ἰε ἄοᾱηῖᾱ Ὑδ ἄβῖᾱε ἄοδ Ὑδ διῶ ÷ η̅ζοέιηδιεῖρῖ δι εᾱέοῖῶηᾱέεϋ ο̅γοδζῖᾱ. Ὅῖ FreeBSD ἄβῖᾱε ο ÷ ἄᾱεᾱοῖ Ὑῖη ἄεᾱ ῖᾱ δᾱᾱ Ὑ ÷ ἄε Ὑῖᾱ ο̅οεᾱᾱηῖ εᾱε ῖεῖεεζᾱηῖ Ὑῖη δᾱᾱᾱᾱ Ὑᾱεῖη ἄεᾱ ἄοᾱηῖᾱ Ὑδ. Ὄδιόοδζᾱεᾱε ῖᾱᾱ Ὑᾱεζ διεέεεβᾱ ἄδῖ ο̅οεῖηἰᾱοη̅ζοδ Ὑδ, οῖῶβῶᾱ ἄᾱᾱοᾱβῖῶ, δᾱᾱᾱ Ὑῖηᾱᾱᾱ ᾱεᾱᾱᾱηῖεῖρῖ οᾱ ÷ οᾱᾱᾱᾱβῖῶ, δᾱᾱᾱ Ὑῖηᾱᾱᾱ ἄᾱᾱοεῖρῖ, δᾱᾱᾱᾱ Ὑᾱεῖηῖῶᾱ δᾱᾱᾱᾱᾱᾱᾱᾱᾱᾱᾱᾱ, ἄῖῶδζᾱᾱᾱοδ Ὑδ ἄεεῶϋῖῶ, εᾱε ῖῶεᾱᾱᾱᾱεϋ ῖῶεᾱβῖῶᾱ Ὑᾱεῖη ῖῶᾱᾱ ῖᾱ εᾱεβῶᾱᾱ. ῖῶᾱᾱᾱᾱ ῖᾱ ἄεᾱ ÷ ἄεᾱᾱᾱᾱᾱᾱᾱ ὀεδ δᾱᾱᾱᾱᾱᾱᾱᾱ ἄδῖ ἄοδ Ὑδ ὀεδ ἄοᾱηῖᾱ Ὑδ ῖ Ὑᾱᾱ ο̅δζ Ὄοεῖῖᾱβῶ οῖῖ Ports (<http://www.FreeBSD.org/ports/>).

Αί ÷ ñæÛæáoðáé íá ÷ ñçóεíïðíεPóáoðá íéá áoáñíïP ðíø áβíáé áεáéÝóεíç óá Ýíá íuñí εáεóíøñáεéù óýóóçíá, óuóá áðεÛ ááí íðíñáβóá íá áíóεéááoóðPóáoðá áóóu ðí εáεóíøñáεéù óýóóçíá. Áβíáé, uñò, ðíεý ðεεáíu uóε εá áñáβóá íéá áíóβóóíε÷ áoáñíïP óóí FreeBSD. Αί εÝÉæáo Ýíáí óóáεáñu áíøðçñáòçòP áεá ðí áñáoáβí óáo P áεá ðí Internet, Ýíá áíεuðεóóí óóáεíu áñáoáoβáo, P áðεðò óçí éεáíuòçóá íá éÛíáoð óçí áñáoáoβá óáo ÷ uñβò áεáéíÝðÝ, áβíáé ó÷ ááuí óβáíøñí uóε ðí FreeBSD íðíñáβ íá éÛíáé uóε ÷ ñæÛæáoðá. Ðíεεíβ ÷ ñPóáoð ððíεíáεóðí óá uεí ðíí éuóí — óuóí áñ÷ Ûñέíε uóí éáé Ýíðáεñíε áεá÷ áéñέóóÝð óóóóçíÛóuñ UNIX — ÷ ñçóεíðíεíýí ðí FreeBSD uò ðí ááoééù εáεóíøñáεéù óýóóçíá óóíðò óóáεííýð áñáoáoβáo ðíðò.

Αί Υν÷ά60ά 60ί FreeBSD άδύ εÜðíεí Üεεí ðáñεάÜεεíí 6ýðí0 UNIX, ίΥñά6ά Þäç 6ά ðáñε6666άñά άδύ ά60Ü ðí0 ðñäεÜεά60ά. Αί υ66666 ç άìðáεñß 6ά6 ðñíΥñ÷ά6άε άδύ εάε6666άεÜ 66666Þíά6ά íά áñά6ε6ü ðáñεάÜεεíí, υð66 6ά Windows® εάε ίε ðάε6666άñά áεάυ6ά66 6ί0 Mac OS®, εά ðñΥðáε ίά άðáíáý6ά6ά, υð66 áßíáε áíáíáíυíáíí, άðεðεΥíí ðñύí áεά ίά áñíεάε6εάß6ά íά 6íí 6ñυðí áñάά66ά6 6ά Υíά 6ý6666 UNIX. Ά666 6í FAQ, εάεð6 εάε 6í Αά÷άεñßáεí 6ί0 FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/index.html) άðí0άεíýí áíáεñά6éεÜ áíááíÞ6íά6ά áεά ίά ίάεéíÞ6ά6ά.

5. Æáôß ïïñÛæåôáé FreeBSD:

- [illegible]

Ἀϊβάεῖς ἰὰ αἰάοΥνῖοῖα ὑόε ς ἔΥίϣ “ἄεαϑεἰν” ÷ ἡόεῖνῖοῖαῖοῖα ἄἰρ ἰὰ αἰῖ ὄνὑῖοῖο, ἰ Ἰάῖο ὄϣἰἰἰἰἰἰἰἰ “ἄὑνἰἰἰἰἰἰ”, ἔἰ ἰ Ὑεῖο ὄϣἰἰἰἰἰἰ “ἰῖῖῖῖῖῖῖ ἰἰ ὄἰ ἔἸῖῖῖῖῖ ὑόε ἔἸῖῖῖῖῖ”. Ἀέὑὑ ἰῖῖ Ἰῖῖ-ἰῖῖ ὄῖῖῖῖῖῖ ὄῖῖ *ἰῖῖ* ἰῖῖῖῖῖῖ ἰἰ ἔἸῖῖῖῖῖ ὄῖῖ FreeBSD, ἄῖ ὄῖῖῖῖῖῖ ἰἰ ἔῖ÷ῖῖῖῖῖῖ ὑόε ὄῖῖ ἰῖῖῖῖῖῖ, ἰῖῖῖῖῖ ὄῖῖ ἄῖῖῖῖῖ ἰἰ ὄῖῖ ἔἸῖῖῖῖῖ ὑόε ἔἸῖῖῖῖῖ.

6. Διέαδ ᾱβιάέ ιέ ἀέάοῖᾗ ὕδ ἰαδᾱί ὀῖδ FreeBSD ἑάέ ὀῖδ NetBSD, OpenBSD ἑάέ ὀῖδ Ὑέεῖῖ BSD ὀδὀδᾱῖ ὀῖδ ᾱῖέέὀῖ ἑᾱᾱᾱ;

Ἰ James Howard Ὑ ÷ ᾱᾱ ᾱᾱᾱ ἰέά ἑάᾱᾱ ᾱᾱᾱ ὀδ ὀδὀᾱᾱᾱ ἑάέ ὀῖδ ᾱᾱὀῖᾱᾱ ἰαδᾱί ὀῖδ ᾱᾱὀῖᾱᾱ Ὑᾱᾱᾱ ᾱᾱ ὀῖδ DaemonNews (<http://www.daemonnews.org/>), ὀῖδ Ὑᾱᾱᾱ ᾱ Ἰέᾱᾱᾱᾱᾱ ἑᾱᾱὀῖᾱᾱᾱᾱ BSD (http://ezine.daemonnews.org/200104/bsd_family.html).

7. Διέά ᾱβιάέ ᾱ ὀᾱᾱὀδὀᾱᾱ Ὑᾱᾱᾱ ὀῖδ FreeBSD;

Ἀδὀᾱ ὀδ ὀδᾱᾱᾱ ὀδὙᾱ ÷ ὀῖδ ᾱῖ ὀᾱᾱᾱᾱᾱ ἑᾱᾱᾱ ὀδᾱ ᾱᾱὀδὀᾱ ὀῖδ FreeBSD. Ἰᾱ ᾱὀὀᾱᾱᾱ ᾱᾱὀᾱᾱᾱ ὀῖδ FreeBSD ὀᾱᾱᾱᾱᾱ ὀᾱὀᾱ ÷ ᾱῖᾱ ἑάέ ᾱὀᾱ ὀῖδ ᾱῖ ὀδὀᾱᾱ ἑᾱᾱᾱᾱ. ᾱ ὀᾱᾱᾱ ὀῖδ ᾱᾱὀᾱᾱᾱ 6.x ᾱᾱὀῖᾱᾱᾱᾱᾱ ᾱὀᾱ ὀῖδ ἑᾱᾱᾱ 6-STABLE ἑάέ ᾱ ὀᾱᾱᾱ ᾱᾱὀᾱᾱᾱ 7.x ᾱᾱὀῖᾱᾱᾱᾱᾱ ᾱὀᾱ ὀῖδ ἑᾱᾱᾱ 7-STABLE.

ἸὙ ÷ ᾱᾱ ἑάέ ὀδᾱ Ὑᾱᾱᾱ 7.0, ᾱ ὀᾱᾱᾱ ᾱᾱὀᾱᾱᾱ 6.x ᾱᾱᾱ ᾱᾱὀᾱ ὀῖδ -STABLE. Ὑὀὀᾱᾱᾱ ᾱὀᾱ ὀδᾱ 7.0 ἑάέ ἰᾱὀᾱ, ᾱ ὀᾱᾱᾱ 6.x ᾱᾱᾱᾱᾱᾱ Ὑὀᾱ ᾱᾱὀᾱᾱᾱ ὀδᾱ ὀὙὀᾱ ὀδᾱ “ᾱᾱὀᾱᾱᾱᾱ ὀὀᾱὀὀᾱᾱᾱᾱ” ἑάέ ἑᾱ ἑᾱᾱᾱᾱ ἰᾱᾱ ᾱᾱᾱᾱᾱᾱ ᾱᾱ ὀᾱᾱᾱᾱ ὀᾱᾱᾱᾱᾱᾱ, Ὑὀᾱ ὀὀᾱ ᾱᾱᾱ Ὑᾱᾱᾱᾱ ὀᾱ ἑᾱᾱᾱ ᾱὀᾱᾱᾱᾱ. ἑᾱ ὀὀᾱᾱᾱ ὀᾱᾱὀᾱὀᾱᾱᾱ ᾱᾱὀᾱᾱᾱ ᾱὀᾱ ὀδ ὀᾱᾱᾱ 6-STABLE, ᾱᾱᾱ ὀὀᾱ ᾱᾱᾱᾱᾱᾱ ὀᾱ Ὑᾱ “ᾱᾱᾱᾱᾱᾱᾱᾱ” (legacy) ἑάέ ἰᾱ ὀᾱ ὀᾱὀᾱὀᾱᾱ ᾱᾱὀᾱᾱᾱ ἑᾱ ᾱᾱὀᾱὀᾱᾱᾱᾱ ἰᾱᾱ ὀᾱᾱ ἑᾱᾱᾱ 7-STABLE.

Version 9.1 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/9.1-RELEASE/>) is the latest release from the 7-STABLE branch; it was released in ἑᾱᾱὀᾱᾱᾱ 2007. Version 8.4 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.4-RELEASE/>) is the latest release from the 6-STABLE branch; it was released in ἸὙᾱᾱ 2006.

ᾱ Ὑᾱᾱᾱ 9.1 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/9.1-RELEASE/>) ᾱβιάέ ᾱ ὀᾱᾱᾱ ὀᾱᾱὀᾱᾱᾱ ὀδᾱ ὀᾱᾱᾱᾱ 7-STABLE ἑάέ ᾱᾱὀῖᾱᾱᾱᾱᾱ ὀὀᾱ ἑᾱᾱὀᾱᾱᾱ 2007. ᾱ Ὑᾱᾱᾱ 8.4 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.4-RELEASE/>) ᾱβιάέ ᾱ ὀᾱᾱᾱ ὀᾱᾱὀᾱᾱᾱ ὀδᾱ ὀᾱᾱᾱᾱ 6-STABLE ἑάέ ᾱᾱὀῖᾱᾱᾱᾱᾱ ὀὀᾱ ἸὙᾱᾱ 2006.

Ἀᾱ ὀὀᾱᾱᾱ, ᾱ ὀᾱᾱᾱ -STABLE ὀᾱᾱᾱᾱᾱᾱ ᾱᾱ ὀῖδᾱ ISPs, ὀῖδᾱ ᾱὀᾱᾱᾱᾱᾱ ÷ ᾱὀᾱᾱ ἑᾱᾱᾱ ἑάέ ὀὀᾱᾱᾱᾱᾱ ÷ ᾱὀᾱᾱ ἑᾱᾱᾱ ὀᾱᾱᾱᾱᾱᾱ ἑάέ ἰᾱᾱᾱ ᾱᾱᾱᾱ ᾱᾱᾱᾱ ὀᾱ ὀ ÷ Ὑὀᾱ ἰᾱ ὀᾱ ἰὙᾱ (ἑάέ ὀᾱᾱᾱᾱ ᾱὀᾱᾱᾱ) ÷ ᾱᾱᾱὀᾱᾱᾱὀᾱᾱ ὀὀᾱ ᾱὀᾱᾱᾱᾱᾱᾱ ὀδᾱ ὀᾱᾱᾱ -CURRENT. Ἀᾱὀᾱᾱᾱ ὀὀᾱᾱ ἰᾱ ᾱᾱᾱᾱᾱ ᾱὀᾱ ὀὀᾱᾱᾱᾱᾱ ἑᾱᾱᾱ, ᾱᾱᾱ ᾱ Ὑᾱᾱᾱ -CURRENT ἑᾱ ὀᾱᾱᾱ ἰᾱ ÷ ᾱᾱὀῖᾱᾱᾱᾱᾱ ἰᾱᾱ ᾱᾱ ᾱὀᾱᾱ ὀᾱᾱὀᾱᾱᾱᾱᾱ Ὑᾱᾱ ἰᾱ ÷ ᾱᾱὀᾱᾱᾱᾱ ὀᾱ ᾱᾱᾱ ÷ ᾱὙᾱ ὀᾱ ὀᾱᾱᾱ ÷ ᾱᾱᾱὀᾱᾱᾱᾱᾱ ὀδᾱ (ὀᾱ ὀ ÷ Ὑὀᾱ ὀὙᾱᾱ ἰᾱ ὀᾱ ᾱᾱὀᾱᾱᾱ ÷ ᾱ -STABLE).

ἸὙᾱᾱ ᾱᾱὀᾱᾱᾱ ᾱὀᾱᾱᾱᾱᾱᾱ ἑὙᾱ ἰᾱᾱᾱᾱᾱ ἰᾱᾱᾱ. Ἀᾱ ἑάέ ὀὀᾱᾱ ᾱὀᾱᾱ Ὑᾱᾱᾱ ἰᾱ ᾱᾱὀᾱᾱᾱᾱᾱ ᾱᾱᾱᾱᾱ Ὑᾱᾱ ἑάέ ὀᾱ ὀὀ ÷ ἰὙ ἰὙὀᾱ ὀῖδ ὀᾱᾱᾱᾱ ἑᾱᾱᾱ ὀῖδ FreeBSD (ᾱᾱᾱᾱ ὀὀᾱ ᾱᾱὀᾱᾱᾱ ὀὀᾱ FreeBSD-CURRENT ἑάέ FreeBSD-STABLE), ἰᾱ ᾱᾱὀᾱᾱᾱ ᾱβιάέ ἑὙὀᾱ ὀᾱᾱὀᾱᾱᾱ ᾱὀᾱ ὀὀ ÷ ᾱὙᾱᾱ, ἑᾱᾱᾱ ἰ ὀᾱᾱᾱᾱ ἑᾱᾱᾱᾱ ᾱβιάέ ὀᾱᾱὀᾱᾱᾱᾱ Ὑᾱᾱ ἑᾱᾱᾱᾱᾱ ὀὀ ÷ ὀᾱᾱᾱ.

ὀᾱᾱὀᾱᾱᾱᾱ ὀᾱᾱᾱᾱᾱᾱ ᾱᾱ ὀὀᾱ ᾱᾱὀᾱᾱᾱ ὀῖδ FreeBSD ὀὀᾱᾱᾱ ἰᾱ ᾱᾱᾱᾱ ὀδᾱ ὀᾱᾱᾱᾱ ὀᾱᾱᾱᾱᾱᾱ Ἀᾱὀᾱᾱᾱ (<http://www.FreeBSD.org/releeng/index.html>) ὀᾱᾱ ᾱᾱὀᾱᾱ ὀὀᾱᾱᾱ ὀῖδ FreeBSD.

8. ὀᾱ ᾱβιάέ ὀῖδ FreeBSD-CURRENT;

ὀῖδ FreeBSD-CURRENT

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#CURRENT) ᾱβιάέ ᾱ ὀὀᾱ ᾱᾱᾱᾱ Ὑᾱᾱᾱ ὀῖδ ᾱᾱὀῖᾱᾱᾱᾱ ὀὀᾱὀᾱᾱᾱ, ὀῖ ὀὀᾱᾱ ἰᾱ ὀῖ ἑᾱᾱᾱ ἑᾱ ᾱᾱᾱ ÷ ἑᾱᾱ ὀὀᾱ ἰὙᾱ ἑᾱᾱᾱ FreeBSD-STABLE. Ἀᾱ ὀῖ ἑᾱᾱ ᾱὀᾱ, ὀᾱᾱὀᾱᾱᾱᾱ ὀᾱᾱᾱ ᾱᾱᾱᾱ Ὑᾱᾱ ἰᾱᾱ ὀᾱ Ὑὀᾱᾱ ᾱ ὀᾱᾱᾱᾱ ἰᾱ ὀᾱ ᾱᾱὀᾱᾱᾱ ἑᾱᾱᾱ ὀῖδ ὀὀᾱὀᾱᾱᾱ ἑάέ ὀᾱ ὀᾱᾱᾱᾱᾱᾱᾱᾱ ÷ ᾱὀᾱὀᾱᾱ. Ἀᾱὀᾱ ὀῖ ὀ ÷ ᾱὀᾱᾱ ὀὀᾱᾱ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#CURRENT) ὀὀᾱ ᾱᾱ ÷ ᾱᾱᾱᾱᾱ

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/index.html) ἀέά έăđôñŸñăéăđ ó÷ăđéêŬ îă ôï
-CURRENT.

ËÙäá ìÞrí, ðánÛäññóáé óóéäñéùðððá (<http://www.FreeBSD.org/snapshots/>) äêäüðáñ ðñó äáóßæññóáé óóçñ ðñÝ÷ñóá äáðÛóðáóç ðññ êêÛäñ -*CURRENT* éáé -*STABLE*. Ìé óðñ÷-ñé ðßóñ áðñ ËÙäá Ýéäñóç óóéäñéùðððñó äßñáé:

- Ἰ Ἰεᾶ÷τὸ ὄςδ οἰεᾶδὸαβᾶδ Ἰεᾶϊόςδ οἶο εἰᾶεοἰεῖῖ ἱᾶεᾶδὸῦδᾶόςδ.
- Ἰᾶ ἁρᾶε ὄς ἁοῖᾶοῦδᾶ ἁῖεῖῖδ ἱᾶεᾶδὸῦδᾶόςδ οᾶ ὑοῖοδ ἁδῆεῖοῖῖ ἱᾶ ἱᾶεᾶδᾶοδᾶοῖοῖ οἶ -*CURRENT* ᾶ οἶ -*STABLE* ἁεῖῖ ἁᾶῖ Ἰ÷τᾶοῖ οἶ ÷ᾦᾦᾦ ᾶ οἶ ἁῖᾦᾦ ᾶρῖόςδ ἱᾶ οἶ δᾶᾦᾶεῖῖοῖεῖῖ ἱᾶῖ ἱᾶ ὄς ἱᾶᾦᾦ.
- Ὲ ἁεᾶδᾶᾦόςδ ἁῖᾦδ ὀᾶεᾶᾦῖ ὄςἱᾶᾦᾦ ἁῖᾶοῖᾦῖ ὄ÷ᾶδῆῖῖ ἱᾶ οἶ δςᾶᾦ ἑᾶεῖᾶ, οᾶ δᾶᾦᾦδᾶόςδ ᾦο ÷ᾶῖᾦοῖᾶ ἑῖᾦδ ᾦῖῖ Ἰ÷ςῖ ἁᾦᾦᾦᾦᾦ. (Ἄῖ ἑᾶῖ ἑῖᾦᾦ ὄςδ ÷ᾦᾶόςδ οἶο CVS ἁᾦᾶῖ ἁῖᾦῖ ἱᾶ ὀᾶᾦᾦ ἑῖᾦδ ᾦᾦᾦᾦᾦῖῖ ὀῖοῖ ᾦᾦῖᾦ.)
- Ἰᾶ ἁῖᾶὀᾶῖᾦᾦᾦ ὑᾦδ ἑῖᾦᾦ ῖῖ ÷ᾦᾦᾦᾦᾦᾦᾦᾦ ἑᾶῖ ἁῖᾦᾦᾦᾦ ᾦο ÷ᾦᾦᾦᾦᾦᾦ Ἰεᾶ÷ῖ, ἑᾶ Ἰ÷ᾦ ὀῖ ἱᾦᾦᾦᾦᾦᾦ ἁῖᾦᾦ ἑῖῖᾦ δῆᾦᾦᾦ ἁῖᾦᾦᾦᾦᾦ.

[illegible]

Ἀδβοϋία οόεαιεύοοδα ἀεαυόαι δανῦαιόας ιέα οἶνῦ οἱ ιρία ἀέα υεῖροδ οἶροδ ἀιαναιρύ εεῦαιροδ ἀίῦδοοιϋό. ἘιανΠόέα
 οόεαιεύοοδα ἀέα οέο δει ἀϋῖροέεαβδ ἀν÷έοἀεοῖιέέῚο (i386 έάε amd64) ιδιῖαβόα ιά ἀῖαβόα οόϋ οάεβἀα
<http://snapshots.us.freebsd.org/>.

9. Ôé åßíáé ôï FreeBSD-STABLE;

1/40áí êôêêîüüñçôá ôí FreeBSD 2.0.5, ç áíÜððôîç ôîô FreeBSD ÷üñßóôçêâ óá äýí êêÜäîð. Í Ýíáo êêÜäîð ïñÜóôçêâ -STABLE (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#STABLE), êáé ï Üêêîð -CURRENT

[illegible]

Ἰ ἐεῦἄρὸ 2.2-STABLE ἀὰὰἀἀὰἀἀὸἐçèâ ἰὰ ὁçí ἐðèèρῑοἰñβἰ ὁçð ὘ῇἄρὸçð 2.2.8. Ἰ ἐεῦἄρὸ 3-STABLE ὀἂῇἄβῠὀἰἰἰ ἰὰ ὁçí ἐðèèρῑοἰñβἰ ὁçð ὘ῇἄρὸçð 3.5.1, ðῑὸ Þὀἰἰ ἑἰῇ ç ὀἂῇἄὀὀἰἰἰἰ ὁçð ὀἂῇἄἄἄἄἄἄ 3.x. Ἰ ἐεῦἄρὸ 4-STABLE ὀἂῇἄβῠὀἰἰ ἰὰ ὁçí ἐðèèρῑοἰñβἰ ὁçð ὘ῇἄρὸçð 4.11, ὁçð ὀἂῇἄὀὀἰἰἰἰ ὁçð ὀἂῇἄἄἄἄἄἄ 4.x. Ἰῇ ἰῠῖἰἰ ἄῇῇἰἰἰ ὀἰὸ ἄβῑῑῑῑἰἰ ἑἰὀἄ ἄἄἄἄ ὀἰ

[illegible]

Í ðeðUäò 9.1-STABLE áßíáé í -STABLE ðeðUäò ðíò äñßóêæðáé öðu áñåñÑß áÚÜðöõç. Ç óâêæððáß êðêèïïñß ðíò ááðßæðáé öôñ ðeðUäí 9.1-STABLE áßíáé ç 9.1-RELEASE, êáé Ýæérä öóêð ÉáññÜñêíð 2007.

Ἰ ἐεῦἈρὸ 8-CURRENT ἀβίαέ ἰ ἐεῦἈρὸ -CURRENT ὄϊο ἀίαῖδῶόόάόάέ αῶδῶP ὀç ὀέέἈP ἀίαῖἈῦ ἡόόἈ ἰἈ ἈçίέἰῶῖἈçέἈβ ç
 ῥῥἈ ἈῖἈἰῦ ὀἰῶ FreeBSD. ἈἈβῶἈ ὀἰ ὀἰἈἰἈ Ὀέ Ἀβίαέ ὀἰ FreeBSD-CURRENT; ἈέἈ ὀἈἰέὀὀὀἈῖἈῖ ὀἈçἡἡἡἡἡἈῖ ὀ-Ἀὀέἰῦ
 ἰἈ Ἀὀὀὀ ὀἡἡ ἐεῦἈἡ.

10. ÊÜèà ðuôâ ãßííôáé åðßçìàð ãõêëïöíñßàð ôïõ FreeBSD;

H ñ Ûáá ÌñáÛmóçò òùí Áéäüòáúí <re@FreeBSD.org> áñíáé óðçí èðèèìòìñá íéá éáéííýnéá Ýèáíróç òìò FreeBSD èÛèá ðáñððìò òÝóóáñèð ìÞíáð, éáðÛ ìÝòì ùñí. Ìé çìáññìçíáð èðèèìòìñáð áíáéíðíñíðáé áñèáòù éáéñù ðñéí, þóðá ùòíé áíðèáýíòí ðÛìù òðí òýòóçíá íá ìÝñìòí ðùðá íé áñááóðáð òìòð ðñÝðáé íá Ý÷ìòí ðèðèçñùèáß éáé áíðèíáíðáðß. Ðñéí áðù èÛèá èðèèìòìñá, ðñìçááðóáé íéá ðáñðíáíð ðíèèíð, þóðá íá áíáðóáééóðáß ùðé ç ðñìòèðèç íÝìí ÷áñáèðçñéóðéèðí ááí Ý÷÷é áñìçðééÝð áðèððóðáèð òðç ððáéáñùðçðá òçð Ýèáíróçð. Ðíèèí ÷ñÞðóáð èàññíýí áðòù òì áíáò òçð ðñìòí÷÷ð Õð Ýíá áðù òá éáéýðáñá ðñÛáíðá ò÷÷áðèéÛ ìá òì FreeBSD, áí éáé ç áíáìíÞ ùð ùðìò òðÛòìòí ùèá òá ùñááá íÝá ÷áñáèðçñéóðééÛ òì -STABLE ìðñíáß íá áñíáé èßáí áèíáðñéóðéèÞ.

Ḑāñēōōōāñāō ḑēçñīōīñḑāō ò÷āōēēŪ īā òç āēāāēēāōḑā ēōēēīōīñḑāō (ḑāñēēāīāāñŸīō ēāē āīūō ÷ñīñāēāñŪīāōīō āḑēēāḑīāñūr ēōēēīōīñēḑī) īḑīñīŸī īā āñāēŸī ōōēō ōāēḑāāō ḑñīāōīēīāōāō āēāūōāūī
(<http://www.FreeBSD.org/releng/index.html>) ōōç āēēōōāēḑ ōīḑīēāōāō ōīō FreeBSD.

Ãéá õiõð ðéi áîèiõóépääéð, õðÛñ÷iõï éâèçiañéíÛ óôéâiëüôõðá (binary snapshots) üðùð áíäöÝñèçêâ ðéi ðÛíù.

11. Ðiéiò åbíáé õđåýèõñiò ãéá ôi FreeBSD

[illegible][illegible]

12. Ðĩõ ìđĩĩþ íá đĩũĩçèãõôþ ôĩ FreeBSD;

Ἐὐεἶα ὁγαρίαόεῖβ Ἰέαῖος ὁῖο FreeBSD ἀβίαέ ἀεάεῖόεἰς ἰῖόου ἀίπῖοῖῖο FTP ἀδῦ ὀίῖ ἀέάἡἡέόόῖβ FTP ὀῖο FreeBSD (ftp://ftp.FreeBSD.org/pub/FreeBSD/):

- Ç öäæäöäöä ööööööööä öö öö 6-STABLE, 9.1-RELEASE öööä ö äöäöä öö öäöööä
9.1-RELEASE (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/9.1-RELEASE/>).

- Άεäuöäöäð ðñ ääöbæííöäé öäÓöéäëüöðöðä (<http://www.FreeBSD.org/snapshots/>) äbíííöäé ääèçìäñéíÜ äðu öñöð èèÜäñöð -CURRENT ääé -STABLE, ääé äñöðçñäðíýí ääöÜ äÜöç üöñöð äö÷íëýíöäé ìä öçí äíÜðöðíç ääé öñí Ýèää÷í öñí öäèäöðäðäö ääíéÜö ðññäñäñÜöñí.
- Ç öäèäöðäðä èðèèüöñíðä ðñ ääöbæííöäé öóñí èèÜäñ 5-STABLE, ç 8.4-RELEASE, ìðññäð íä äñäèäð öóñí ääöÜèñäñ 8.4-RELEASE (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.4-RELEASE/>).

Δεχόμενοι το ό-αδείο είναι οι άΕύεραοί οιό FreeBSD οά CD, DVD έάέ ΰέεα ιΥόά ιοίηρύι ίά άηάειρύι οόι Άά÷άένΒάει (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/mirrors.html).

13. Ðùò ìðñþ íá Ý÷ù ðñüôâáoç óôç ÂÜóç ÄããñÝíùì ìà ôèð ÁíáoñÝò ÐññæçÜÜùì;

ôî Web (<http://www.FreeBSD.org/cgi/query-pr-summary.cgi?query>).

[illegible]

Ðnéí òðíàÜëàòà íéá áíáóíñÜ ðñíâëPíáòíð, ðáñáéáëíýíà áéááÜòóá òí ÑñÜòííóáð ÁíáóíñÝð ÐñíâëçíÜòóíí áéá òí FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/problem-reports/article.html), Ýíá Üñèñí áéá òí ðòò íá ãñÜòáòá êáéÝð áíáóíñÝð ðñíâëçíÜòóíí.

14. Ôé Üëãò đçãÝò đęcñĩĩĩĩéí òđŨñ÷ĩĩĩ;

Δανάεάεῖγῖα ἁῖΥαῖα ὁç ἔβοῶα ὁç ὈἰεῖçḂḱḱḱḱ (http://www.FreeBSD.org/docs.html) ὁçῖ ἔγḡḡḡ ἁḡḡḡḡḡ ὁῖḡḡḡḡḡ ὁῖḡ FreeBSD (http://www.FreeBSD.org).

ΈαöÜεάεί 2 Ôαêìçñßùόç έάέ ÕðïóôÞñéìç

1. Διέα έάέÜ äέäëßá ððÜñ÷ïñí ó÷äóέéÜ ìä ôñ FreeBSD;

Ôñ Project ðáñÜäáé ìéá ìäÜëç äéÜñä ôäêìçñßùόçð ðñò äéäôßäöäé online áðü ôñ ðáñäéÜðü óýñäáóñ: <http://www.FreeBSD.org/docs.html>. Ôä ßäéä äóðÜ Ýäññäóä äßíäé äéäéÝóéñä έάέ ùð ðäéÝóä óä ðñßä ìðñäßðä ìä ääéäóäóðóäðä äýññä óðñ FreeBSD óýóðçñä óäð. Ðäñéóóüðäñäð ääððñÝñäéäð ó÷äóέéÜ ìä óä ðäéÝóä ôäêìçñßùόçð, έä äñäßðä óðéð äéüñòéäð ðáñäñÜñòñðð.

Äðéðñüéäðä, έä äñäßðä έäé Üééä óðñéóðññäñä äéäéßä óðçñ Äéäéññäñäðä óðñ ôÝñò áððñý ðñò FAQ, έäé ðñò Ää÷äéñéäßñò.

2. Äßñäé äéäéÝóéç ç ôäêìçñßùόç έäé óä Üééäð ìññðÝð, üðüð äðéü έäßñññ (ASCII) Þ PostScript;

Ìäé. Ç ôäêìçñßùόç äßñäé äéäéÝóéç óä ðéÞñò äéäññäðééÞñ ìññðñ έäé ðñüðññ óðñðßäóçð, óðñ äéäéññéóð FTP ðñò FreeBSD, óðñ έäóÜññäñ /pub/FreeBSD/doc/ (<ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/>).

Ç ôäêìçñßùόç äßñäé έäóçäññéñðñéçñÝñ ìä äéÜñññòð ðñüðñòð. Äððñß ðäñééäñäÜñññ:

- Ôñ üññä ðñò ääñÜñòñò, üðüð ð.÷. **faq**, **Þ handbook**.
- H äéÞóóä έäé ç èüäééñðñßççð ðñò έäéñÝñò. ÄððÜ ääóßäññäéä óä ìññäóä ðñò Ý÷ïñ äñäß óðéð ðñðééÝð ððéñßóäéð έäé ðñò ðññäßðä ìä äñäßðä óðñ έäóÜññäñ /usr/share/locale óðñ FreeBSD óýóçñÜÜ óäð. Ìé ðñÝ÷ïóäð äéÞóóäð έäé èüäééñðñéÞóäéð ðñò Ý÷ïñ äéäéÝóéñäð áððÞ óç óðéäñÞ óðçñ ôäêìçñßùόç äßñäé ìé ðáñäéÜðü:

¼ññä	Äññññäßä
en_US.ISO8859-1	ÄääéééÜ ÇÐÄ
bn_BD.ISO10646-1	Bengali (Þ Bangla)
da_DK.ISO8859-1	ÄäñÝäééä
de_DE.ISO8859-1	ÄäññäñééÜ
es_ES.ISO8859-1	ÉóðäñééÜ
fr_FR.ISO8859-1	ÄäééééÜ
hu_HU.ISO8859-2	ÌðääñÝäééä
it_IT.ISO8859-15	ÉóäéééÜ
ja_JP.eucJP	ÄéäðñÝäééä (èüäééñðñßççð EUC)
mn_MN.UTF-8	ÌääñéééÜ (èüäééñðñßççð UTF-8)
n1_NL.ISO8859-1	ÌééäñäééÜ
pl_PL.ISO8859-2	ÐñññééÜ
pt_BR.ISO8859-1	ÐñññääéééÜ (Äñääééßä)
ru_RU.KOI8-R	ÑÞóééä (èüäééñðñßççð KOI8-R)
sr_YU.ISO8859-2	ÓÝñäééä
tr_TR.ISO8859-9	Ôñññééä
zh_CN.GB2312	ÄðéñðñéçñÝñä ÊéñÝäééä (èüäééñðñßççð GB2312)
zh_TW.Big5	ÐäñäññóéäéÜ ÊéñÝäééä (èüäééñðñßççð Big5)

[illegible]

Οἱ θεῶν Ἰεσοῦς οἱ ἰά ἐαῶν Ὑοῶν ἐὰν ἰά ἀαῶαῶοῶ Ποῶν οἱ δᾶς Ὑοῖ ἀβίᾶς ὑοῦ Ὑδᾶεῶα ἰθῖνᾶβῶα ἰά ἀεᾶ ÷ ἀεῖνῶοῶβῶα ὄς ὁ ἀεῖνβῶος ÷ ἡσῶεῖν ἰθῖνῖπῖοῶ ὁ ἀ ὀίς ἐεῶ Ἰῖᾶ ἡνᾶεᾶβᾶ ἀεᾶ ÷ ἀβῖνῶοῶ δᾶς Ὑοῖν οἷο FreeBSD ὑδῶν ὄς pkg_add(1) ἐὰν ὄς pkg_delete(1).

[illegible]

Άέά άάνÜääέäíá, öi FAQ, óóá ÁáääééêÜ, óá ïñöP PDF, åbíáé óöi άάéÝöi ïå üíñá faq.en_US.ISO8859-1.pdf.tgz.

Άí ôî ÌŸñáôá áôôü ððñáßôá íá ÷ñçóéüððēßôáôá ôçí áéüēîðēç áñòñēß áéá íá ääēáôáôôßôáôá ôî ðáēŸôî ôîò ÁääēēēîŸ
 PDF FAQ:

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

Áíõŷ òì ÈÛíáòâ áóòü, ìðñââòâ íá ÷ñçóèìðìéÞóâòâ ôçí áíóìēÞ pkg_info(1) áéá íá âñâòâ ðìõ Ý÷áé âãéâóáóôâèâß òì áñ÷âßì.

```
# 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)

¼ðuò ìðĩñâßôâ íá äâßôâ, ôĩ book.pdf èà Ý÷:æ æâêâôáóôâèß óóĩí èâôÛeĩĩ

```
/usr/share/doc/en_US.ISO8859-1/books/faq.
```

Áí áái èÝéâôâ íá ÷ñçóèîüðíëÞóâôâ ôâ ôáéÝôá, èá ðñÝðáé íá éáôââÛôâôâ iüüíé óáo ôâ ôóîðéâóíÝíá áñ÷âßá, íá ôâ áðíóóîðéÝôâôâ éáé Ýðáéôâ íá áíóéáñÛôâôâ ôâ áíóðóðíé÷á Ýáññáóá ôôç èÝóç ôîôð.

Ἄεά ὁάνÜäæëàíá, ç Ýëäîç òïð FAQ óå öóíääâàìÝíá äñ÷:âßá HTML, öòìðëäöìÝíc ìä ÷ñßöç òïð bzip2(1), ìðñâß íá äñäèâß öòì äñ÷:âßì doc/en_US.ISO8859-1/books/faq/book.html-split.tar.bz2 . Ἀέά íá éäöââÜôäðä éác íá äðìöòìðéÝöäðä äöòü òì äñ÷:âßì èä ðñÝðäé íá eÜíäðä òì ðäñäeÜòù:

```
# 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
```

Éá éáóáéΠáóá íá íéá óóēēīāΠ áðū āñ÷āÁá .html. Ōī ááóéū īīīŪæáóé index.html, éáé éá ðáñéÝ÷áé ōīī ðΒíáéá ðáñéá÷ñŸíuí, áéóááūāééū óēééū, éáé ááōīŸō ðñīō óá Ūēéá òīΠáóá òīō áāāñŪōīō. ÌðīñáΒóá Ýðáéóá íá áíóéāñŪøáóá Π íá íáóáééīΠóáóá óá āñ÷āÁá áðōŪ óóéð óáēééÝð òīōð èÝóáéð. üðūð áðáéóáΒóáé.

3. Ðiõ iðĩņp íá àñù ðęçņĩrõĩņbãð ãéá ěbõóãð çęãôñĩíéēĩý óá÷õãñĩĩãbĩõ õĩõ FreeBSD;

Ἰδιῆναβὸς ἰά ἀναβὸς θεῖπῆαεὸ θεῖχῆρῖοῖνβῖο ὀδῆί ἀίὸβὸοῖε÷ç êáðá÷ḥṇçóç ὠῖο Ἀά÷âéñéâḃῖο áéá ὀéο èβὸðâð
 çèâêðṇῖíêéíý ὀá÷õäṇṇâḃῖο

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/eresources.html#ERESOURCES-MAIL).

4. Ôé ñÜääò óœÐôçòçò åßíáé äéáéÝóéíåò ãéá ôï FreeBSD;

İđĩñãßôâ íá ãñãßôâ đẽPñãèđ đẽçñĩöĩñßàđ óôçí éáôá÷pñçóç ôĩđ Åã÷ãẽñéãßĩđ ãéá ôéđ ñÜãàđ óôæçôPóãuĩ

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/eresources-news.html).

5. ÕđÜñ÷ĩĩ êáíÜëéá óõĩ IRC (Internet Relay Chat) ãéá õĩ FreeBSD;

Íáé, ôá đăñéóóúôăăă IRC äâêôăă äéäèÝôĩôĩ êáíÜëé óăăĐôçôçò äéá ôĩ FreeBSD:

- [illegible]

ÈÜëÀ Ýíá áðu áðòÜ ðá éárÜëëá, áßíáé ÷ ùñéòòü, éáé áár áéáðòíáÝáðáé íá ðá ððüëíéðá. Ôá óððé òçð ððæÞðçðçð áéáóÝíñíí, éáé Þòòð èá èÝëáðá íá áíëíÜòáðá éáéÝíá áéá íá áñáßòá èÜðíëí ðíò íá ðáéñéÜæáé óðí áéëü óáð. ¼ðòð íá *èÜëð* òýðí ððæÞðçðçð óðí IRC, áí áßòáá áðu áðòíýð ðíò ðñíóáÜëëíðáé áýíëëá, Þ áár íðíñáßòá íá áíðéíáòððßòáðá ðíëëíýð ãÝíòð áíëñÞðíòð (éáé áñéáðíýð ðéí íááÜëíòð) íá áðéáßáííðáé óðí èáéðéëü áíðòóðíé ÷ í òçð ððáíá ÷ Þáð, íçí áó ÷ íëçèáßòá éáëüëíð.

ðāñēāñŬōīīôāē óōīī īāçāũ āāēāôŬóôāóçð ôīō FreeBSD
(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html).

3. Diō āñβóēñíóáé íē iāçãßàò áéá ôçí āāēáôÛóóáóç ôiō FreeBSD;

Ἰδιῆνᾱβῶᾱ ἰᾱ ᾱῆᾱβῶᾱ ὀῶῶ ῖᾱῇᾱβῶᾱ ὀῶῇ ῇᾱῶᾱ÷ῖᾱῇῶῇ ὀῖῶ Ἀᾱ÷ᾱῆῆᾱβῖῶ ὀ÷ᾱῶῇῆῆ ὀῖ ὀῇ ᾱῆᾱῶᾱῆῆῆῆῆ ὀῖῶ FreeBSD
 (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html).

4. Ôé ÷ ñǎéÜæñǎé ãéá íá âêôǎëÝòù ôi FreeBSD;

Άέα ääüüöääö öō FreeBSd ädu öi 5.X éäé läöÜ ÷ ñäÉÜæäöäöá Ýí PC lä äðäñäñäöäöð 486 þ ääýöäñí, lä 24 MB þ ðäñéöóüðäñç RAM, éäé öiöëÜ÷éööí 150 MB ÷ þñíö ööí öéëçñü öäð äööëí.

Idināboā íá áeoāēYōoāō uēāō oēō āēāuōāēō oīō FreeBSD íā íeā ÷āicēþī dñīrāeāāñāōþī, īīř÷÷-ñūic (MDA) ēŨñōā āñāōēēþī, āēēŨ áeā íá ÷ñcōíūīdīēPōāōā oī Xorg, eā ÷ñāeāōōāōboā ēŨñōā VGA þēāēyōāñc.

Äåßôå åðßóçò êáé ôï ÊåöÛéáéí 4.

5. Ðuò ìðĩñþ íá äçíeĩõñãÞòù òç äéêÞ ìĩõ, ðñĩóãñĩĩóĩÝíç äéóéÝóá äãéáoÜóóáóç;

[illegible]

Ἄέα ἰά ἀγέιερὸνᾱποάοᾱ ἰέα ὄνιόᾱνιὸι Ὑίς Ὑέαιός, ἀεῖριεὶεῖποάοᾱ ὀέο ἰαῖαῖαὸ ὀίι Ὑῆνι ὁ-ᾱόεῖὙ ἰᾱ ὀῖ ἸᾱῖὙῖου ᾱᾱῖὙᾱῖ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/releng/article.html).

6. İđĩñíýí íá óõĩõđŨñĩĩóí ôá Windows ìå õĩ FreeBSD;

Ἀὰὰὰὰὰὸὸὸὸὰ ὀῆῆῆ ὀὰ Windows, ἑὰ ἱὰὸὸ ὀὶ FreeBSD. Ἰ ἑὰ÷ἑἑἑὸὸὸ ἑἑἑἑἑὸὸ ὀὶ FreeBSD ἑὰ ὀὰ ἑἑἑἑἑἑἑ ἑἑἑἑἑ ἱὰ ἑἑἑἑἑἑἑ ὀὶ ἑἑἑἑἑἑἑ ἑἑἑἑἑἑ ἑἑἑἑἑἑἑ ὀὶ Windows ἑἑἑἑ ὀὶ FreeBSD. Ἀὶ ἑἑἑἑἑἑἑἑἑ ὀὰ Windows ἱὰὸὸ ὀὶ FreeBSD, ἑὰ ὀὰ ὀὰὸὶ ὀὶ ἑὰ÷ἑἑἑὸὸ ἑἑἑἑἑὸὸ, ÷ ἑἑἑ ἑἑ ἱὰ ὀὰ ἑἑἑἑἑἑἑ. Ἀὶ ὀὰ ὀὀἱἑἑ, ἑἑὀὸ ἑἑἑἑ ὀὶ ἑἑἑἑἑ ὀὶἑἑ.

7. Ôá Windows éáoYóôñãpáí ôi äéá÷äéñéóôP äêêßícòçò ìiõ! Dùò èá ôií áðáíáoYñù;

Ἰδιναβλα ἰα ἀδάρῆαεάοάοόPοαοᾶ οἱ ἀέα÷ἀεήεόδP ἀέεβίζοςό οἷο FreeBSD ÷ñζοείηιθέρπιδάο Υἱάί αἴδου οἷοδ οἡνάεο
θαῖαεÜου οἡνυθιόδ:

- Ἀεὶ ἀπερίσῳτ οἱ DOS, ἰσοαξέρεϊς αἰσθὰς ὁδῷ εἰς τοῦτο tools/ ὅς τὸ FreeBSD ἀεὶ μὴ δὲ σάω, ἐπεί τοῦτο αἰεὶ οἱ ἀν-ἀβί bootinst.exe. Ἀεὶ ἀπ' ὧν οἱ ὑποδοχῆς αἰσθὰς δαψινεύου:

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

êáé ï äéá÷âéñéóôÞò âêêßícóçò èá åđáíåãéåôáoôæåß.

- ÎâêéíPóóâ íáíŨ ÷ ñçóéñîðéíPíóáð òç áéóéÝóá âêêéßíçòçð òíð FreeBSD éáé ðçääâíáíòâ óòçí áðêéíâP Custom Installation òíð íáíŕý. ÁðêéÝíòâ Partition. ÁðêéÝíòâ òíŕ íäçäŭ ðíð éáñíééŨ éá ðñáéâß÷â òíŕ áéá÷âéñéóðP âêêéßíçòçð óáð (òðóéŕéíâééŨ òíŕ ðñðîŕ) éáé ũðáí òðŨóáòâ óòíŕ áðñáññááóðP éáðáðíPóáŭí (partition editor) áéá òíŕ íäçäŭ áðòŭ, òí ðñðîŕ ðñŨáíä ðíð éá éŨíáòâ (÷ ùñßð Ũééâð áééááÝð), âíŕáé íá áðêéÝíáòâ (W)rite. ÁðêéÝíòâ yes óòçí áðêéääââßòçð ðíð éá áìòáŕéóðâß éáé ũðáí òðŨóáòâ óòçí ðññíðñîðP áðêéíâPð Áéá÷âéñéóðP Áêêéßíçòçð, áðêéÝíòâ “Boot Manager”. Îâ òíŕ òñŭðí áðòŭ í áéá÷âéñéóðP âêêéßíçòçð éá áñáòâß íáíŨ óòí âßòéí. Îðñáßòâ òðñá íá áñáßòâ áðŭ òí íáíŕý òçð áñéáðŨóðáóçð éáé íá áðñáññééíPóáòâ áðŭ òí óéèçñŭ âßòéí, ũðŭð óòíPèùð.

- Íäëéíßóðä íä ôç äíßëäéä ôçð äéóéÝóäð äêêßíçóçð (ß öíð CD) öíð FreeBSD éäé äðëéÝíðä “Fixit” áðü öí íäíý. ÄðëéÝíðä äßðä ôç äéóéÝóä Fixit äßðä öí CD #2 (öí “live” óýóóçíä äñ÷äßíí) éäé èä äéóÝëëäðä ööí èÝëðöíð fixit. ÄëðäéÝóðä Ýðäéðä ôçí äëüëíðëç áíðíëß:

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

áíðëéäëéðíðáð öí *bootdevice* íä öíí ðñäáíäðéëü óäð äßðëí äêêßíçóçð, üðüð ð.÷. ad0 (ðñðöíð äßðëíð IDE), ad4 (ðñðöíð äßðëíð IDE ööíí äíçëçðéëü äëääëðß), da0 (ðñðöíð äßðëíð SCSI), è.è.ð.

8. Í öíñçöüð ííð ððíëíäéóððð óäëñÛð IBM Thinkpad, óäëñÛð Á, Ö ß ×, óðáíäðÛäé íä äðíëñßíðäé üðáí ðñíóðäëß íä äêêéíßðü öí FreeBSD íäðÛ ôçí ääëäðÛóóáóç. Ðüð ððíñß íä äðëéýóü äðü öí ðñüäëçíä;

Ðñüëäéðäé äéä Ýíä ðñüäëçíä óðéð äñ÷ééÝð äëäüóäéð öíð BIOS ôçð ÉÄÌ óðä óðäëäëñéíÝíä íç÷áíðíäðä, öí ððíßí áíäíññßäëé ôçí éäðÛðíçóç öíð FreeBSD ùð ðééáíü äéäíÝñéóíä FAT äéä ôç èäéðíðñäß ääñáíðíßçóçð ööí äßðëí (suspend-to-disk). Öí BIOS óðáíäðÛäé íä äðíëñßíðäé èäëðð ðñíóðäëäß íä áíäéýóäé öí äéäíÝñéóíä öíð FreeBSD. Óýíðüíä íä ôçí IBM¹, óä äëüëíðäé ííðÝéä éäé äëäüóäéð BIOS ðññééäíäÛíðí óéð äðáíäßðçðäð äéíñëðóäéð.

ÍíðÝëí	êäíóç BIOS
T20	IYET49WW ß íäðääáíÝóðäñí
T21	KZET22WW ß íäðääáíÝóðäñí
A20p	IVET62WW ß íäðääáíÝóðäñí
A20m	IWET54WW ß íäðääáíÝóðäñí
A21p	KYET27WW ß íäðääáíÝóðäñí
A21m	KXET24WW ß íäðääáíÝóðäñí
A21e	KUET30WW

÷äé áíäðññäß üðé íäðääáíÝóðäñäð äëäüóäéð öíð BIOS ôçð ÉÄÌ, ßüðð ðáñíðóéÛëíðí íáíÛ öí ðñüäëçíä. To íðíðíä äðüð (<http://docs.FreeBSD.org/cgi/mid.cgi?20010427133759.A71732>) áðü öíí Jacques Vidrine óðçí çëäëðñííéëß ëßððä öíð FreeBSD äéä öíñçöíýð ððíëíäéóðÝð (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile>) ðññéäñÛðäé íéä äéääééäóðä ç íðíßä ßüðð äíðëÝðäé óä íäüðñä ííðÝéä öíñçöíðí ôçð IBM óä íðíßä äáí äêêéíýí éáñíéëÛ öí FreeBSD, éäé óðä íðíßä íðíñäßðä íä áíäääëíßððäð öí BIOS ß éäé íä äðéóðñÝðäðä óä ðñíçáíýíäíç Ýëäíóç öíð.

Áí Ý÷äðä ðäëéüðñí BIOS, äëëÛ äáí Ý÷äðä ôçí äðëéíäß íä öí áíäääëíßððäð, Ýíäð ðñüðíð íä íäðñäÛóðäð öí ðñüäëçíä äßíäé íä ääëäðäóðßððäð öí FreeBSD, íä äëëÛíäðä öíí áíääíññéóðéëü äñéëíü (partition id) ôçð éäðÛðíçóçð, éäé íä ääëäðäóðßðäð íÝä boot blocks óä íðíßä íä íðíñýí íä ÷äëñéóðíýí öí äéäöíñäðéëü áíääíññéóðéëü ôçð éäðÛðíçóçð.

Äñ÷éëÛ, èä ðñÝðäé íä äðáíäöÝñäðä öí íç÷Ûíçíä óä íéä éäðÛóðäóç ðóðä íä ðññíÛäé áðü ôçí äñ÷éëß äéäíññéóðéëß íëüíç. Áðüð äðäéðäß íä áíäñäíðíëßððäð öí íç÷Ûíçíä ÷ññðð íä öí äóððäðä íä äñäé ôçí éäðÛðíçóçð öíð FreeBSD ööíí èýñéí äßðëí öíð. Íäð ðñüðíð äßíäé íä äðíäëñýíäðä öíí óëëçñü äßðëí éäé íä öíí íäðäëéíßððäð ðñíóðñéíÛ óä Ýíä ðäëéüðñí ThinkPad (üðüð öí ThinkPad 600) ß ëÛðíëí äðéðñäðÝäéí íç÷Ûíçíä, ÷ñçóéíðíëéíðáð öí éäðÛëëçëí éäëðäéí íäðäðñíððð. Ööí íç÷Ûíçíä äðüð íðíñäßðä ðëÝí íä äéääñÛðäðä ôçí éäðÛðíçóçð öíð FreeBSD éäé Ýðäéðä íä íäðäëéíßððäð öíí äßðëí ööíí äñ÷éëü íç÷Ûíçíä. Öí ThinkPad èä ðñÝðäé ðññä íä äëëéíäß éáñíéëÛ.

Íä öí íç÷Ûíçíä óä èäéðíðñäëëß éäðÛóðäóç, íðíñäßðä ðññä íä ÷ñçóéíðíëéíðäðä ôçí äéääééäóðä ðíð ðññéäñÛðäðäé ääð ðóðä íä Ý÷äðä ðäëéëÛ íéä ääëäðÛóðäóç öíð FreeBSD ðíð íä èäéðíðñäß.

1. ÉäðääÛóðäð óéð äéóéÝóðä boot1 éäé boot2 áðü <http://people.FreeBSD.org/~bmah/ThinkPad/>. Äðíëçéäýóðä óä äñ÷äßä äððÛ ëÛðíð öíð èä íðíñäßðä íä óä íäíäññäßðä äñäüðñä.

2. Άλλεαόάόόόόόόά έάνήρέεÜ òι FreeBSD όòι ThinkPad. *Ìçí* ÷ ñçόέήñðíέðόάόά όçí έάόÜόόάόç Dangerously Dedicated. *Ìçí* έÜíάόά άðάíάέέßíçόç üòáí όάέάέðόάέ ç äååáðÜóóάόç.
3. ×ñçόέήñðíέðόάά äßðä òí “Emergency Holographic Shell” (**ALT+F4**) áðä òí έÝέöòüò “fixit”.
4. ×ñçόέήñðíέðόάά όçí fdisk(8) áéá íá áééÜíάόά òíí áíááíññέóóέéü áñέéìü όçð έáðÜòìçόçð òü FreeBSD áðü 165 óá 166 (άóüüò äßíáé í óýðüò ðüò ÷ ñçόέήñðíέáßðáé áðü òí OpenBSD).
5. ÄñÜðάά óá äñ ÷ äßá boot1 έάé boot2 óòí òíðééü óýóόçíá äñ ÷ äßüí.
6. ×ñçόέήñðíέðόάά òí disklabel(8) áéá íá äñÜðάάά óá boot1 έάé boot2 óòí slice òü FreeBSD.

disklabel -B -b boot1 -s boot2 ad0sn

Ôí *n* äßíáé í áñέéìüò òü slice ðüò Ý ÷ áðá äååáóáóóóóάé òí FreeBSD.
7. ÊÜíóá áðάíáέέßíçόç. Óόçí ðññüññð äέéßíçόçð éá ääßðä όçí áðééñð íá íáέéíðóáάά òí OpenBSD. Óόçí ðñáñíáóééüòçóá, íá áðóü òíí ðñüðí éá íáέéíðóáάά òí FreeBSD.

Οἱ ἰά ἐΰῖαοἶ ὁἱ δᾶνάδ' ὕῖν ἰά ἐάε' ὁἱ τᾶν βόαε ὁόγῃ δᾶν βδδὸν ὁόε δῖο ἐΰεᾶοἶ ἰά ὕ÷ᾶοἶ ἀέδ' ἐρ ἀέβῖζός OpenBSD ἐάε FreeBSD ὁοῖῖ βᾶεῖ ὁῖν ὁοῖ δῖεῖ τᾶε ὁοῖρ, ὁἱ ἀόβῖν ὁἶα ὑὸ ὕόεζός ὁοῖῖ ἀῖαῖ βόοε.

9. Ìðĩñp íá êÛíu ââêáoÛóóáoç óá Ýíá äßóêî ìà ÷ áéáóíÝíñòò òññâò;

İđiñăßôă, áëëÛ ăßíáé êáêÐ éäÝá.

Ἄν ἁὰβῶα ÷ ἁεᾰοῖῶῖῖῖῖῖ ὀηᾰβῶ ὀᾰ ῶῖᾰ ὀῶᾰ÷ῖῖῖ ῖᾰᾰῖῖ IDE, ὀḀῖῖ÷ᾰε ῖᾰᾰῖᾰᾰ Ḁεᾰῖῖῖῖῖῖ ῖ ῖᾰᾰῖῖ ᾰὀὀὀ, Ḁῖῖ ὀῶῖῖᾰ, ῖᾰ ὀὀᾰᾰὀᾰὀᾰε ῖᾰ ἁεᾰοῖῖῖῖῖ ᾰὀᾰᾰὀὀ (ῖ ῖᾰᾰῖῖ ᾰᾰ ῶ ÷ ᾰε ῖᾰῖῖὀ ᾰῖᾰᾰᾰᾰᾰᾰῖῖῖ ᾰᾰᾰᾰῖῖῖῖ Ḁῖὀ ῖᾰ ῖḀῖῖῖῖ ῖᾰ ᾰὀᾰᾰᾰᾰᾰὀὀὀὀ ὀὀὀ ÷ ἁεᾰοῖῶῖῖῖῖ ῖῶῖ ὀḀ ᾰὀὀᾰᾰᾰᾰᾰ ἁεᾰοῖῖῖῖῖ ᾰᾰᾰᾰᾰᾰᾰᾰ ᾰᾰᾰᾰᾰᾰᾰᾰᾰᾰᾰᾰᾰᾰᾰ ὀῖᾰ ᾰᾰᾰῖῖῖῖ, ἁῖὀᾰ ὀῖ ῖḀῖῖ ὀᾰᾰᾰᾰᾰ ῖὀᾰ ῖ ᾰḀὀὀ ῶ ÷ ᾰε ὀᾰᾰᾰᾰᾰᾰ ὀᾰῖῖῖ). ὀᾰ ὀὀᾰὀὀῖῖᾰ ῖᾰ ᾰᾰῖῖῖῖᾰᾰ ἁεᾰῖῖῖῖᾰ ᾰḀὀᾰ.

Áí Ý÷âôâ ïäçäü SCSI ìä ÷äéáóíÝñôð òñìäðð ääððä áõðð ôçí äðÜíôçóç.

10. ÄëÜöĩñá đăñßăñăă đñÜàiaôă ôõiaăßĩĩõĩ üôái âêêéř ĩă ôç äéóêÝôă âêêßĩçôç! Ôé ôõiaăßiaé;

Áí æēÝðāōā ēāōāōōŬōāēō ũðhō òī īç÷Ŭīçīā íā īçī āðīēñßīāōāē Þ íā ēŬīāē āðāíāēēßīçōç īūñī òīō ũōáí ðñīōðāēāßōā íā āēēēíÞōāōā íā òçī āēōēÝōā āēēēßīçōçò, ēā ðñÝðāē íā ēŬīāōā òōñī āāōōū óāō ðñāēō āñūōÞōāēō:

1. \times ηςοείηδιεΠοάοά έαείηγνέαο, οηάοεί-αείηηοοιγίάο, έαεί ÷ ηηβδ εΰες αέοέΥοάο (έάοΰ θηηοβιςςός έαείΠδ άοάειηβάο οέο ιδιβάο ιυέέο άαΰεάοά έαείηγνέαδ άδου οι έιροδδ ιοιρο, οά άίοβεάος ιά ος αέοέΥοά θιο θΠηάοά ιάαβ ιά εΰθιεί θάηεάεει έαεί ς ιδιβά άηβοείιόάι εΰοδ άδου οι εηάαΰοέ οά οηβά οάεάοοάβ ÷ ηιιέά);
2. Έάοάαΰοάοά οι image οςο αέοέΥοάο ιά άοάάέει (binary P image) οηυδι ιάοάοιηΰδ; (ις ιοηΰθάοοά, άέυιá έαεί ιέ έαέγόαηιέ άδου ιάο Υ ÷ ιοι έάοΰ εΰειο έάοάαΰοάέ εΰθιεί άοάάέει άη÷άβι οά έάοΰοόάς ASCII (έάείΥιη), οιοεΰ÷έοοι ιέά οιηΰ!)
3. Άί ÷ ηςοείηδιεάβοά Windows 95 P 98, άεοάέΥοάοά οι fdimage P οι rawrite οά έάεάηP έάοΰοόάς DOS; Όά έάεοιηάέεΰ άοδΰ ιδιηάβ ιά θάηάιέςειγί οά θηηάηΰιáοά οά ιδιβά άηΰοιοι άάοεάβάο οοι οέέει, εΰοέ οι ιδιβι οοιáβιáέ έαεί ιά οά θηηάηΰιáοά άςιέιηάβάο οοι αέοέάοθι. Ιδιηάβ ιά άςιέιηάςεάβ θηυάέςιá, άέυιá έαεί άί οά άεοάέάβοά οά θάηΰεοηι DOS ιΥόά άδου οι άηάοέει θάηέαΰεει.

÷ òò ãðððð ãíáíáíáí ðáíððððááð ùðò òò Netscape® áçíçíòíááð ðííáíáíá òò éáðÝááíá òð áéóéÝóáð áéðíçòð, Ýðóé áíáíé éáéýðáíá íá ðíçóéíðíçéðááð éÙðíçé Ûççé ðííáíáíá FTP. áí áðð ãíáíé áðíáðí.

11. Ίαεβίςζοά αδι οι ATAPI CDROM ιιρ, αεεΰ οι θνιανανιါ αεεαοΰοοάοζο εΨάε ιοε αάρ ανβόεαε CDROM. Διρ δπα; ζ οοιπεζο αέοβα αδοιρ οιρ θνιιεπιαοιρ αβιαέ Ψιαδ εαεινθελεοιΨιιρ ιαζανο CDROM. Διεεΰ PC Ψν÷ιιόαε δεΨιι ια οι CDROM ιο slave οοοεαοπ οοι ααδοαανιρ ιοα αεααεοπ, ÷ιιιιρ ια οδΰν÷αε οοι βαει εάρΰεε οοοεαοπ master. Αοδι, ογιοιια ια οεδ θνιιαεααναοΨο οιρ ATAPI, αάρ αβιαέ Ψαεοιι, αεεΰ οα Windows οζνιρ ιοεδ θνιιαεααναοΨο ÷αεανΰ, αιπ οι BIOS οι αανιιβ εαοΰ οζι αεεβίςζοζ. Αοδι αβιαέ εαε ι ευαοι διρ οι BIOS εαοΰοαανι ια ααε οι CDROM εαε ια οι ÷ιιζοεοιιθιεποαε, αεεΰ εαε ι ευαοι διρ οι FreeBSD αάρ ιδιιναοα ια οι ααε αεα ια οοια÷βοαε οζι ααεαοΰοοάοζ.

N̄ōiβōōā iārŭ òi óyóóciā oáo, βóōā òi CDROM āβōā íá āβiáé ç master oóóēāōP oóii āēāāēōP ðiō āβiáé oófiāiYíc, P āāāāēùēāβōā üōē āβiáé slave óā Yíā āēāāēōP IDE i iðiβið uùuò Y÷āē Ūēēc oóóēāōP oófiāiYíc uò master.

12. Ìðìñþ íá ääêáóáóðÞóù òì FreeBSD óóì öññçòù ììð ÷ñçóéìððéþíðáð PLIP (Parallel Line IP, IP ìÝóù ÐáñÛëëçççð Ëýñáð);

Íáé. ×ñçóéüñðiePóóâ òððïðieççíÝñî éâepäeí òýðïð Laplink. Áí ÷ñäéÜæâðáé, äéääÜóóâ òï òìPiá PLIP òïð Å÷:âñéäBið (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-plip.html) äéä eäððñÝññäçð ó÷:âðééÝð ïä äééðýúóç ïÝóù ðññÜëççèð èýñäð.

13. Ôé ãâùìãðñßá íá ÷ñçóéìðíéÞòù ãéá ôï ôéëçñü ìïð ãßóêì;

Όγιὰβὺος: ἰὰ οἱ ὑνὶ “ἀαὐιὰδὸνβά”, αἰνίγγῃα οἱ ἀνέειυ οὐί ἐοεβίαναι, εᾶσάεπρ ἐάε οἷΥῖρ ἀΐῤ οἷ÷εῤ αἴυὸ ἀβοέτϑ.
 Ἀέα δὲτερεβ, ἐὰ ἀτάσνηνιάοδᾶ οἱῖρ ὑνὶ ἀδὸυ ὕὸ C/H/S (Cylinders / Heads / Sectors). Ἀδὸυδ ἀβιάε ἐάε ἰ οἷνϑιδ ἰὰ οἱ
 ἰθῖβῖ οἱ BIOS ἀνβόεᾶ ὁᾶ θῖεᾶ δᾶνεί÷β οἷδ ἀβοέτϑ ἰᾶ ᾗνῤῥᾶε.

[illegible]

Òi iurí ðrō Ÿ ÷ áε ðñááíáóεεP óçíáíóá áβíáε ç *εíεεεP* ááúíáòñá. ÁóòP áβíáε εáε ç áðŸíóçóç ðrō áŸ ÷ áòáε òí BIOS
 úóáí ñùòŸáε òí áβóεí “ðíεá áβíáε ç ááúíáòñá óí;” ðáεóá, ÷ ñçóεííðíεáβ áóòP óç ááúíáòñá áεá íá áðíεòPóáε
 ðñúóááóç óóí áβóεí. Éáεð òí FreeBSD ÷ ñçóεííðíεáβ òí BIOS εáoŸ óçí áεεβίçóç, áβíáε ðíεŸ óçíáíóεεú ç ááúíáòñá
 áóòP íá áβíáε óúóòP. Áεáεúóáñá, áí Ÿ ÷ áòá ðáñεóóúóáñá áðu Ÿ íá εáεóíòñáεεŸ óóóòPíáóá óóí áβóεí, ðñŸðáε úεá íá
 óóíòùíŸ íúóí áóíñŸ óç ááúíáòñá. ÁεáóíñáóεεŸ εá Ÿ ÷ áòá óíááñŸ ðñíáεPíáóá εáoŸ óçí áεεβίçóç!

Ἀέα ἀβ6εῖρὸς SCSI, ὃ ἀαυλᾰὸνβᾰ ὁῖο δὴ ὕδαε ἰᾰ ÷ ñçöεῖῖῖεῖεᾰβ, ἀἰᾰñðŰῖῖᾰε ᾰῖῖ ὁῖ ᾰῖ ᾰβἰᾰε ᾰἰᾰñᾰῖῖεῖεῖ ὕῖç ὃ ὁῖῖῖῖῖῖῖῖ ᾰεῖῖᾰᾰᾰᾰ ὕῖç ἰᾰðŰñᾰῖῖῖῖῖῖῖῖ (ᾰῖῖῖ ὃῖ ÷ ἰŰ ᾰἰᾰῖ ὕñᾰῖῖῖῖῖῖ ὕῖ ÷ “ὃῖῖῖῖῖῖῖῖ ᾰεᾰ ᾰβ6εῖῖῖῖῖῖῖῖ DOS >1GB” ᾰ εŰῖῖ ᾰἰῖῖῖῖῖῖῖ ÷ ἰ). Ἀῖ ᾰβἰᾰε ᾰῖᾰἰᾰñᾰῖῖῖῖῖῖῖῖ ὕῖç, ÷ ñçöεῖῖῖῖῖῖῖῖῖῖῖῖῖῖ *N* εῖῖεβἰᾰñῖῖῖῖῖῖῖῖ, 64 εᾰῖῖῖῖῖῖῖῖῖῖῖῖῖῖ ᾰεᾰ 32 ὃñᾰβῖῖῖῖῖῖῖῖ ÷ εŰῖ, ὕῖῖῖ ὃῖ *N* ᾰβἰᾰε ὃ ÷ ÷ ñçöεῖῖῖῖῖῖῖῖῖῖῖῖῖῖ ὃῖῖ ᾰβ6εῖῖῖ ὃᾰ ἰᾰ. Ἀεᾰ ῖᾰñŰᾰᾰεᾰἰᾰ, ᾰεᾰ ὕῖ ᾰῖῖῖῖῖῖῖῖ 2GB, εᾰ δὴ ὕδαε ἰᾰ ὕ ÷ ᾰῖᾰ 2048 εῖῖεβἰᾰñῖῖῖῖῖῖῖῖ, 64 εᾰῖῖῖῖῖῖῖῖῖῖῖῖῖῖ, ᾰεᾰ 32 ὃñᾰβῖῖῖῖῖῖῖῖ ÷ εŰῖ.

Άι ἀβιάς ἀναπαῖδιέσι Ἰύς (εἰς ὁ-÷-τῷ ὁανῷ ÷ ὁαῖς Ἰόοις πῶα ἰὰ ἰαῶνιέριόις εὔθιεῖς ὁανέινεσι πῶι MS-DOS®)
 εἰς ς ÷ ὑνῶδεῦδῶα ὁῖο ἀβῶεῖο ἀβιάς ἰαῶέγιῶαν ἄδῃ 1GB, ÷ ῶῶεῖριέρι πῶα M ἐδεβῶνιῶ, 63 ὁῶαδ ἰῷ ὁῶι÷εῷ
 (ῡ÷ε 64) εἰς 255 εἰσῶέῤ, ὑθῖο M ἀβιάς ς ÷ ὑνῶδεῦδῶα ὁῖο ἀβῶεῖο ὁἶ MB, ἀεἰνῶῖ Ἰύς ἰἶ ὁῖ 7.844238 (!), ῶε, ὁῖ
 ὁανῷαἰἶα ἰἶ, ἰ ἀβῶεῖο ὁῖ 2GB εἰ ἀβ÷ἶ 261 ἐδεβῶνιῶ, 63 ὁῶαδ ἰῷ ὁῶι÷εῷ εἰς 255 εἰσῶέῤ.

Αί αιί αβόα όβαιρινόι αέα οι άναδύι, π αι οι FreeBSD άδιόγ÷άε όόçi αίβ÷ιάόόç όçò ουόόπò αιιιαόηβαό έáoÛ όçi
 ααέáoÛόόáoç, ι άερίγόόανι όηυόι αέα ίά οι άναέÛιόáo, αβίάέ ορίπυό ίά άçίέρινάπáoαό Ýία ιέένυ αέαιÝνεόία
 DOS όοι αβόει. Οι BIOS Ýάέáoά έά αίέ÷ιάγόάέ όç ουόόπ αιιιαόηβά, έάέ ιόινάβáo άÛίόά ίά αέαανÛιόáo οι

[illegible]

Ἀράγεαὶοὐδὲὔ, ὁδῦν÷ἀε ὕία ἀεῖγὰεἰἡ ἀεῖαῖὕοιῖ ἀῖεῖεῖοῖεῖ ὁῖῖῖῖῖῖῖ ὁῖ ἰῖῖῖῖ ἡεῖῖῖῖῖῖῖ ἰῖ ὁῖ FreeBSD ἡεῖ ἡῖῖῖῖῖῖ `pfdisk.exe`. Ἰῖῖῖῖῖῖ ἰῖ ὁῖ ἡῖῖῖῖῖῖ ὁῖῖ ὁῖῖῖῖῖῖῖῖῖ `tools` ὁῖῖ CDROM ὁῖῖ FreeBSD **P** ὁῖῖ ἡεῖῖῖῖῖῖῖ ὁῖῖῖῖῖῖῖῖῖ FTP ὁῖῖ FreeBSD. Ὀῖ ὁῖῖῖῖῖῖῖ ἡῖῖῖ ἰῖῖῖῖ ἰῖ ÷ῖῖῖῖῖῖῖῖῖῖ ἡεῖ ἰῖ ἡῖῖῖῖῖῖῖῖ ὁῖ ἡῖῖῖῖῖῖῖῖ ÷ῖῖῖῖῖῖῖῖῖῖῖῖῖῖῖ ἡῖῖ ὁῖ Ὀῖῖῖ ἡεῖῖῖῖῖῖῖῖῖ ὁῖῖῖῖῖῖῖῖῖ ὁῖῖ ἡῖῖῖῖ ὁῖῖ ÷ῖῖῖῖῖῖῖῖῖῖῖῖῖῖῖ. Ἰῖῖῖῖῖῖ ἰῖ ἡεῖῖῖῖῖῖ ἡῖῖῖῖῖῖῖ ὁῖ ἡῖῖῖῖῖῖῖῖῖῖῖῖ ἡῖῖῖῖῖῖῖῖῖῖῖῖῖῖῖ.

14. ŌđŨñ÷ĩōĩ ēŬđięię đăñēĩñéōiĩß óōi đùò đñÝđăē íá ÷ũñßóu ôi äßóēi;

Íáε. ÐñÝðáε íá ááááεúεáßòá üóε ç ñéæéêP (root) êáoÜòìçόç áñßóεáòáε êÜòü áðu òïòò 1024 êóεßíáñïòò, þóòá òï BIOS íá ìðíñáß íá áêéεPðáε òïí ðññPíá áðu áóòP. (Όçíáεþóòá üóε áóòòò áßíáε Ýíáo ðáñéíñéóíüò óòï BIOS òïò PC, êáé ü÷÷ε óòï FreeBSD).

[illegible]

15. Ἀβιάε οὐίᾱάου̣ οἱ FreeBSD ιὰ ðñĩāñŨìáóá äéá÷äǃñéòçø äßóëùí (disk managers);

To FreeBSD áíááíũñßæéé éáé äðéôñÝðáé ôç ÷ ñßóç ðíð Ontrack Disk Manager. Äáí ððíóóçñßæííðóáé Üëëíé äéá÷ äéñéóóÝð äßóëuí.

Αί εΥέαθα αδερο ία ÷ ñçόείιδιέΠόαθα οι άβόει ία οι FreeBSD, άάρ ÷ ñάεΥέααθα άέα ÷ άεήεόοΠ άβόειθ. Αδερο ñόειβόθα οι άβόει άέα υόι θαήεόούθα ÷ ññι ιδιñάβ ία άάε οι BIOS (όόίΠευδ 504ÎÄ), έάε οι FreeBSD έα άίάέάέϑάε θυόι άεάέάñι ÷ ññι Υ ÷ άθα όόçι ðñάñιάόέευόçόά. Αί ÷ ñçόείιδιέάβθα εΥθιέι θαέευ άβόει όά άεάέόοΠ MFM, βούδ ία ðñΥθαε ία θαβθα όόι FreeBSD θυόιθδ έθέβññιθδ ία ÷ ñçόείιδιέΠόάε.

Άν έχετε άδρά τά ÷ ñçóεíñðíεPóóαά òí αBóεíñ ðυοί íα òí FreeBSD υοί έάε íα εÜðíεíñ Üεεí εάεóíñαεέυ ογóðçíá, εά ðñÝðάε
 íα íðíñáBóá íα òí εÜííáðá ÷ ùñBð áεά ÷ áεñεóðB αBóεíñ: áðεpð ááááευεάBóá υóε òí áεáñÝñεóíá áεεBíççóç ðíñ FreeBSD
 εάεpð έάε ç εαóÜòíçç òíñ Üεεíñ εάεóíñαεέý οóóðPíáðíñ αñBóεíñάε íÝóά óóíñ ðñpðíñ ðñ 1024 εóεBíññíñð. Áí
 áBóóá αñεáóÜ ðññóáεóεευð, Ýíá áεáñÝñεóíá áεεBíççóç (boot) íááÝεíñð 20MB εά áBíáε αñεáóυ.

16. ¼ôáí âêêéíþ öi FreeBSD áéá ðñþôç öiñÛ, ðáßñíù öi ìÞíöiá Missing Operating System. Ôé ööiâáßiáé;

Ἄοδρ ἀβιάς ιέα ἐεάοέερ δᾶνβδδουός ἀέΎίαιϋο ιαοαίϣ οῖο FreeBSD ἐάε οῖο DOS ϐ ἐϋδῖεῖο Ὑεῖο ἐαέοιονᾶεέϣ
 ϑ÷ἀοέεϋ ιὰ οϋϋ ἐάΎΑ δῖο Ὑ÷ἀε οῖ ἐαέΎίΑ ἀέα οϋϋ ἀαυιαδνβᾶ οῖο ἀβόεῖο. Ἐὰ δνΥδαε ιά ἀδαῖααεάοαοδρῶαδᾶ οῖ
 FreeBSD ἀέϋ οϋνβῖαδᾶ δνῖοἀεδέεϋ οέδ ιαεᾶβᾶδ δῖο ἀρῶια δέῖ δϋῖο, ἀβιάς ϑ÷ἀαυῖ οβᾶῖονῖ υόε ἐὰ δᾶ ἐαοάοϣνᾶδᾶ.

17. Ἀέας ἀϊ ἰδῖν ἰά οῖα ÷ βού ὀΥῖα ἀδῦ ὀγι ὀνῖοῖδῖ F? ὀῖο ἀέα ÷ ἀέneoῖδῖ ἀέβίγοç;

Άρδου ἀβιάε Ὑεῖ Ἰά ὀγιδδουιά ὄτῳ δῆπᾶεπιδάοτῳ δῖῳ δᾶνῆεᾶνῶοᾶοᾶε ὀόϋ δῆπῆαῖγῖαῖε ἀπῆῶϋόϋ. Ἀῖ ὀοῖδβδῶᾶε ϋ ᾶῶιᾶδῆβᾶ ὄτῳ BIOS ἰᾶ ᾶῶῶ ὄτῳ FreeBSD! Ἀῖ ῖ ᾶῆᾶῆῶῶῶ ῶ ὀῖ BIOS ὀᾶῶ ὀδῖῶῶῆβᾶῆᾶ ἰᾶῶῶῶῶῶ ῆῶῆβῖᾶνῖ (ῶῶ÷ῖῶ ᾶῖᾶῶῶῶᾶ ῶῶ “>1GB drive support”), ᾶῖῆῖῶῶῶ ἰᾶ ᾶῆῖῶῶᾶ ᾶῶῶ ὀϋ ῆῖῶ ῆᾶῖ ἰᾶ ᾶῶᾶῖᾶῆᾶῶᾶῶᾶῶᾶ ὀῖ FreeBSD.

18. ÐñÝđǎé íá āāēáôáóôÞòù üēī ôīī đcǎǎβī êþǎēēá;

ĀāīēēÜ, ü÷-é. Üöüüōī ōáo ōōīéōōīŷīā íá āāēāōáoōōōāōā, ùò āēÜ÷-éóōī, ōīī ðçāāī ēēāēēā ôçð áēáīīīð base, ī īōīīō
 ðāēēēāīāÜīāē āēēāōÜ āđū ōā āñ÷āāā đīō áíáoÝñīīōāē āā. ēāēð ēāē ōīī ðçāāī ēēāēēā ôçð áēáīīīð sys, ç īōīīā

[illegible]

Ár Ý ÷ áoá Ûíáoá áéáéÝóeñ ðíř ðçááßí eþaééa, éáé áíunþæááo ðuð íá láoááæuðóðóáoá Ýíá íeueēçñí óýóðçíá áðu áoouí, éa áéáoereoreáðáo ðÛná ðrey uóar áíáááæíþæááo ðí óýóðçíá óað óá láæeířóééÝð áæauóáéð oíř FreeBSD.

Ἀέα ἰά ἀδέεΎῖαὸὰ ὕῖα ὀδῖογῖρεῖ ὄρὸ δχαάβῖτὸ ἐπαέέα, αἴυ ὁγῖ ἀδέεῖῖαB Distributions ὀῖτὸ ἀῖῖαέῖαβῖτὸ ἀῖῖαὸUόδαόçð òòòòΒῖαὸῖδ, ἀδέεΎῖαὸὰ οῖ ἱᾶνῖ Custom.

19. ×ñǎēÛæǎôǎé íǎ öôéÛù ðñĩóǎñĩĩóĩ Ýĩ ðõñÞíǎ;

Ç açiqeronaβa aβuo rYio ðonPia Poaia an÷eēÜ o÷aauu oði÷nauoēēu aPia oā iēa āāēaoÜoōaoç FreeBSD, aēēÜ iē ðēi ðnuoōaoāo āēauoāēo Y÷toī uoāēçēaβ aðu oçī āēoāauaP āēoçðÜ oēēēēuōaauī ðniāauiÜoūi nYēiēoçð oīo ðonPia. Aðu oī FreeBSD 5.X ēaē iāoÜ, āβiaē ānēaoÜ āYēreī iā nōēiBoāoā oīi ðonPia ÷nçōēiūðiēpīoāo oī ðiēY ðēi āoYēēēoī oYōoçia oūi "hints" oā iðβia iðniāBoāo iā nōēiBoāoā oççī ðniōniðP oīo loader.

Αἰτᾶ÷ἡ Ὑψὺς ἰὰ αἰβᾶε αἰυῖα ἰὰ ἀçíεíõñᾱΠόαòᾱ Ὑῖα ἰΨὶ ðõñΠῖᾱ ἰ ἰðῖΒῖò ἰᾱ ðᾱñέΨ÷ᾱε ἰυῖ ὁᾱ ðñῖᾱñṼἡᾱόᾱ ῖᾱΠᾱççòð ðῖò ÷ñᾱέṼᾱάοᾱ, ᾱῆᾱ ἰᾱ ᾱῆòðπόᾱᾱ εṼṼἰῆᾱ ἰῇñΠ ðῖóυòçᾱ ἰᾱΠῖçð RAM, ᾱῆṼ ᾱóòυ ᾱᾱ ᾱΒῖᾱέ ðῆΨῖ ᾱðᾱᾱΒòçðῖ ᾱῆᾱ ὁᾱ ðᾱñῆóóυðᾱñᾱ óðóðΠῖᾱóᾱ.

20. Ἄεά οἶοδ ἐυαέειγύ δνῡοάαόζο διῡ ÷ η̣ζοόρῐ, ῐά ÷ η̣ζοείῡδιέβου DES, Blowfish, P MD5 ἐυαέείδιβζός, έάέ διὸ εά έάειηβού δε εά ÷ η̣ζοείῡδιέιγί ῐέ ÷ η̣βόδαδ ῐῡ;

Ç õñññäëäñÝíç ñññß ãñðððññÜöçðð äéä ëüäëëýð ððí FreeBSD äßñé ðí *MD5*. Ç äñëëß áíðßëçðç äßñé üðé ðññÝ÷íðí äéäýðäñç áóðÜëäéä ðä ó÷Ýóç ìä ðçí ðññäññóéäëß ñññß ðñò UNIX ðñò äáðßäðäé ðññí äéäññéëí *DES*. Ìé ëüäëëß DES äßñé äéññä äéäéÝóëíé, áí ÷ñäëÜäðäé ìä äéäñññÜðäðä ðí äñ÷äß ðññí ëüäëëß ðäð ìä ðäëéðäññä ääéðññäëëÜ ððððßñäðä, ðä ìðßä ÷ñçóëíðñëýí äéññä ðí ðäëéðäññí äéä ëäüðäññí áóðäéÝð óýððçñä (äßñé äéäéÝóëíä áí ääéäðäððßðäðä ðç äéäññß “crypto” ìÝù ðñò sysinstall ß ääéäéçððßðäðä ðññ áíðßððíé÷ ðçäñß ëðäëä áí ëÜñäðä ääéäÜððäðç ìÝù ðçäññß ëðäëä). Áí ääéäðäððßðäðä ðéð äéäëëëëßðäðä crypto ëä ìðñÝðäðä äðßçð ìä ÷ñçóëíðñëëßðäðä ãñðððññÜöçðç Blowfish ç ìðßä äßñé äéññä ðñí áóðäëðð. Õí ðñéä ñññß ëüäëëß ÷ñçóëíðñëäðäé äéä ðñð ìÝñð ëüäëëýð, äéÝä÷äðäé äðñ ðçí äññäðñðçðä äéðñññ “passwd_format” ðññ /etc/login.conf, ðí ìðßí ðäñññäé ðéð ðéÝð “des”, “blf” (áí äßñéä äéäéÝóëíç) ß “md5”. Ääðä ðç óäëßää manual ðñò login.conf(5) äéä ðññéðððäññäð ðççñññññäð ó÷äðéëÜ ìä ðéð äññäðñðçðäðä äéðñññ.

21. $\tilde{A}\acute{e}\acute{\alpha}\tilde{o}\tilde{\beta}$ $\acute{\alpha}\tilde{\iota}\tilde{\rho}$ \tilde{c} $\tilde{a}\tilde{e}\tilde{o}\tilde{e}\tilde{Y}\tilde{o}\tilde{a}$ $\tilde{a}\tilde{e}\tilde{e}\tilde{\beta}\tilde{i}\tilde{c}\tilde{o}\tilde{c}\tilde{o}$ $\tilde{i}\tilde{a}\tilde{e}\tilde{e}\tilde{i}\tilde{U}\tilde{a}\tilde{e}$ $\tilde{e}\tilde{a}\tilde{i}\tilde{i}\tilde{e}\tilde{e}\tilde{U}$, $\tilde{e}\tilde{n}\tilde{\alpha}\tilde{i}\tilde{U}\tilde{a}\tilde{e}$ $\tilde{o}\tilde{o}\tilde{c}\tilde{i}$ $\tilde{i}\tilde{e}\tilde{u}\tilde{i}\tilde{c}$ Probing Devices...;

Αί Ύ÷ ãðã ãããðãðõçĩΎĩ ããçãũ IDE Zip® P Jaz®, áõáẽñÝõðã õĩĩ éáé íáíáðñĩíõðãẽPõðã. Ç áéõẽÝõá ãẽẽßĩçõçõ ìðĩñãß
 íá ìðãñããðõðã ãðũ ãðõĩýð õĩðõ ããçãĩýð. ÌãðÛ õçĩ ãããðÛðõðãçõ õĩð õðõðPĩãĩð, ìðĩñãðã ìá íáíáõõĩãÝõðã ðĩĩ ããçãũ.
 Æðãðéðõĩγĩã ùðé õĩ ðñũãẽçĩã áðõũ éã áẽĩñèùãß ðã áðũĩáĩç Ýẽãĩõç.

22. Ἀέαὸβ δαβνήũ õĩ ìÞĩõĩḃ ḗṼẽĩõð panic: can't mount root uðáĩ ḃẽẽẽĩÞ õĩ óýóõçĩḃ ḃẽḃ ḃñÞõç õĩñṼ ìḃðṼ õçĩ ḃḃẽḃḃṼõóḃḃõç;

Òi òòÙeíà àdòù ðñíÝñ÷÷:àðáé àðù òçí òýá÷÷÷òç ðìò ðñíéàéàðòáé àíáéòðàð òìò áéàòíñàòééý òñùðìò ìà òìí òììì òì BIOS éáé òì boot block áíòéééàìàÙííòáé òìòò òéèçñíýð àðòéìòð. Òì ðñùàéçíà òòíPèùð àìòáíðæàðáé òá òòòòPíáðà ìà àýí àðòéìòð IDE, áéáéèÙ ùòáí íé àðòéíé àðíáé master (P ùíííé òìòð) ì éàéÝíáð òòí áéèù òìò áéàáèòð IDE éáé ìà òì FreeBSD ìá àðíáé àáéáðááòòçÝííò òòí àðòéí ðìò àñðòéàðáé òòí ààòòàñýííòá áéàáèòð. Òì boot block ñìðæáé ùòé òì òýòòçíà àðíáé àáéáðááòòçÝííò òòí ad0 (òì àáyòàñí àðòéí òìò BIOS) àñ ì ðòñPíáð áíáéÝòáé òìí ðñìòì àðòéí òòí ààòòàñýííòá áéàáèòð, ad2. ÌàòÙ òçí áíð÷÷÷ìàòç òùí òòòéàòòí, ì ðòñPíáð ðñìòðáéàð ìá ðñìòáñòPòáé àðòù ðìò òì boot block ðéòòàýñáé ùòé àðíáé ì àðòéìò àéèèðçòçð, ad0 àñ òòçí ðñáàíìàòéèùòçðá àðíáé ì ad2 éáé òòòéèÙ àðìòòá÷÷Ùíáé.

Ãéá íá äéĩñèpóáôâ ôĩ ðñũäëçíá, êÛíôâ Ýíá áðũ ôá ðáñáêÛôù:

1. Ἀδάρειέειρβόδα ὁι ὀγδόγια ἐὰε δεΰδοα **Enter** ὁόκι ὀνιὀνιὀν **Booting kernel in 10 seconds; hit [Enter] to interrupt.** Ἰὰ ὁιρ ὀνιὀνι ἄδοι ἐὰ ἀὰὰβόα ὁοί ὀνιῶνἰἰὰ ὁιρ ὀνιὀνι **ἀέβιςόο.**

Εὰν οὖν εἶναι ἡ ρίζα τοῦ `root_disk_unit` ἡ `"disk_number"` . Ὁ `disk_number` εἶναι ἀριθμὸς 0 ἀνὰ τὴν FreeBSD ἀριθμὸς ἀειάριθμοι τοῦ τοῦ master ἀριθμοῦ τοῦ τοῦ τοῦ IDE, 1 ἀνὰ τὴν ἀειάριθμοι τοῦ τοῦ slave ἀριθμοῦ τοῦ τοῦ τοῦ IDE, 2 ἀνὰ τὴν ἀειάριθμοι τοῦ τοῦ master ἀριθμοῦ τοῦ τοῦ ἀειάριθμοι τοῦ IDE ἀειάριθμοι, 3 ἀνὰ τὴν ἀειάριθμοι τοῦ τοῦ slave ἀριθμοῦ τοῦ τοῦ ἀειάριθμοι τοῦ IDE ἀειάριθμοι.

đăêôă ãñÛøăă boot, êáé ôĩ óýôôçía óáo èá đñÝđăê íá âêêêíPóăê êáííêêÛ.

[illegible]

2. ĩãôáéíĤóôă ôĩ äăóëĩ ôĩô FreeBSD óóĩĩ đñũôăýĩĩôă äăăăêĤ IDE, Ĥóôă ĩé ôêëçñĩß äăóëĩé ĩá äăĩáé óĩĩă÷ũĩĩĩé.

23. Ðīéá āḅíáé ôá ũñéá ôçò ìíḐìçò;

Οι υνεί αβίαιε όα 4 gigabytes όα ιέα οοίγεοεί Υίγ άαεάο Ûοόáoç όά άñ÷έοάêñíêß ι386. Ιάέίπρíoáo άδñ όέο άέäñúáoéο FreeBSD 4.9 έάé 5.1, ðñíoçñßæáoάé έάé ðñéóóúoðñç ιΠιç ιΥíoú oño pae(4). ×ñåëÛæáoάé ùoðúoí íá ιáoääëuòéóoðß ίårÛ ι ðoñπíoáo ðñééäíäÛííóáo έάé ιέα Υíoñä άðέéíäΠ áéá oçí άíññüíßçoç oño PAE:

options	PAE
---------	-----

Οι FreeBSD/pc98 ὄντ' ἀέ ὑνέϊ ὁὰ 4 GB ἰφῖπὶδ, ἐὰς ἀὰς ἰδῖνᾶς ἰὰ ÷ ñçóεῖῖθῖεçᾶς PAE ὁὰ ἀδῶP ὁçῖ ἀñ÷-εῶᾶεῖῖθῖεP. Οῖοι FreeBSD/alpha, οἱ ὑνέϊ ὁçð ἰφῖπὶδ ἀἡἡῶῖῖῖῖῖῖ ἰῖῖ ὁῖῖ ὁῖῖ ὁῖῖ ὁῖῖ ὁῖῖ ÷ ñçóεῖῖθῖεᾶῖῖῖῖῖ - ἰῖῖ ἰῖῖῖῖῖῖῖῖῖῖῖῖ ἰῖῖῖῖ ὁῖῖ Ὀçῖᾶῖῖῖῖῖῖ ἰῖῖῖῖ Ὀῖῖῖῖῖ ἰῖῖῖῖ ἰῖῖῖῖ Alpha. ἰῖῖῖῖ ἰῖῖ÷-εῶᾶεῖῖῖῖῖῖῖῖ ὁῖῖ ὁῖῖῖῖῖῖῖῖῖῖῖῖ ἰῖῖ ὁἱ FreeBSD, ὄ÷-ῖῖῖ ἰῖῖῖῖῖῖ ἰῖῖῖῖῖῖῖῖῖῖῖῖ ἰῖῖῖῖῖῖῖ ὁ÷ ὁ÷ ἰῖῖῖῖῖῖ ὁἱ ὁç ἰῖῖῖῖῖῖ ὁἱῖῖῖῖῖ ἰφῖπὶδ (ðῖῖῖῖ terabytes).

24. Ðięá ąńíáé ôá üńéá ôĩõ óõóôPíáôĩò áñ÷ąűí ffs;

Ἄέά ὀόόᾀᾀᾀᾀᾀᾀ ᾀᾀ÷ᾀᾀᾀᾀ ᾀᾀᾀ, ᾀᾀ ἰ Ἰᾀᾀᾀᾀᾀᾀ ᾀᾀᾀᾀᾀᾀᾀᾀᾀ ᾀᾀᾀᾀ ᾀᾀᾀᾀᾀ ᾀᾀ 8 terabytes (2G blocks), ᾀ 16ᾀᾀ ᾀᾀᾀ ᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀ Ἰᾀᾀ ἰ Ἰᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀ ᾀᾀᾀ 8ᾀ. ᾀᾀᾀᾀ ᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀ, ᾀᾀᾀᾀᾀ÷ᾀᾀ Ἰᾀᾀ ᾀᾀ÷ᾀᾀᾀ ᾀᾀᾀᾀ 1 terabyte, ᾀᾀᾀᾀ ἰᾀ ᾀᾀᾀᾀᾀᾀᾀ ἰᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀ, ᾀᾀᾀᾀᾀ ᾀᾀᾀᾀᾀᾀ ἰᾀ ᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀᾀ (ᾀᾀᾀ ᾀᾀᾀᾀᾀ÷ᾀᾀᾀ) ὀόόᾀᾀᾀᾀᾀ ᾀᾀ÷ᾀᾀᾀᾀ ἰᾀᾀᾀᾀᾀᾀᾀ 4 terabytes.

Ôi ÿYăéóôî ÿYăăèò áíüò añ÷ăăîō óă Ýíá óýóôçíă ffs âbíăé đăñăđîō 1G blocks, P 4TB iă ÿYăăèò block ôúi 4K.

Ðáíáëáð 3-1. ÌÝãéóðá íããÝëç áñ÷ãßíí

ÌÝããëò block fs	ëãëòíõñããß	ðñÝðáë íá ëãëòíõñããß
4K	4T-1	>4T
8K	>32G	32T-1
16K	>128G	32T-1
32K	>512G	64T-1
64K	>2048G	128T-1

¼ðáí ðí ÌÝããëò block ðíð fs áßíáë 4K, ëãëòíõñããíý íá ðñëðëÛ Ýííáóá blocks (triple indirect blocks) éáë óá ðÛíðá éá Ýðñáðá íá ðññéíñßæííðáë íííí áðü ðí ÌÝãéóðí áñëëíü block ðíð íðííãß íá áíáðñáóóáëãß íá ðç ÷ñðóç ðñëðëíí Ýííáóóíí blocks (ðññßðíð $1K^3 + 1K^2 + 1K$), áëëÛ óãëëëÛ í ðññéíñëóíüð íðãßëãðáë óá Ýíá (ëÛëíð) ùñéí 1G-1 óðíðð áñëëííýð ðííí blocks. Õí ùñéí óðíðð áñëëííýð ðííí block éá Ýðñáðá íá áßíáë 2G-1. ÕðÛñ÷íðí ëÛðíéá ðñíáëðíáðá ùðáí íë áñëëííß ðííí block ðíð fs ðëçóéÛæííðí ðí 2G-1, áëëÛ ðÝðíéíë áñëëííß block äáí íðíííý íá ðñíóããëóóóíý íðáí ðí ÌÝããëò block fs áßíáë 4Ê.

Ãéá íããÝëç block 8Ê éáë íãããéýðãñá, óá ðÛíðá éá Ýðñáðá íá ðññéíñßæííðáë áðü ðí ùñéí 2G-1 óðíðð áñëëííýð ðííí block ðíð fs, áëëÛ óðçí ðñáñíáðéëüðçðá í ðññéíñëóíüð íðãßëãðáë óðíí ëÛëíð ùñéí 1G-1 óðíðð áñëëííýð ðííí block ðíð fs. H ÷ñðóç ðíð óóóóíý íñßíð ðííí 2G-1 blocks, äçíéíðñããß ðñÛáíáðé ðñíáëðíáðá.

25. Ãéáðß ðáßñíü ðí íðíðíá ëÛëíð, archsw.readin.failed íáðÛ ðçí íáðáãëðððéóç éáë äëëßíçóç íÝíð ððñðíá;

Ãéáðß í ððñðíáð óáð éáë ðí ððüëíéðí ðíð ááóéëíý óðóððíáðíð (world) áßíáë áëðüð óðã÷ñííéóííý. Ç ëãëòíõñããá óá áððð ðçí éáðÛóóáóç äáí ððíóðçñßæãðáë. Åãããéëëãßðá ùðé ÷ñçóéííðíéãßðá ðéð áíðíëÝð make buildworld éáë make buildkernel ãéá íá áíáããëíßðáðá ðííí ððñðíá óáð.

Ìðííãßðá íá áëëéíðóáðá íñßæííðáð ðííí ððñðíá áðãðëãßáð áðü ðí äãýðãñí óðÛáëí, ðéÝæííðáð íðíéíãððíðá ðëðëðñí íüëéð äãßðá ðí íéáë ðñéí íáëéíðóáë í loader.

26. Ç äãëáðÛóóáóç éáðãññÝáë éáðÛ ðçí äëëßíçóç. Õé íðíñ íá ëÛíü;

ÄíëëíÛóðá íá áðáíãñãíðíéðóáðá ðçí ððíóððñéíç ACPI. Íüëéð íáëéíðóáë í ðííðóððð äëëßíçóçð, ðéÝóðá ðí ðëðëðñí space. To óýóóçíá óáð éá äíðáíßóáë

OK

. ÅñÛððá

unset acpi_load

éáë éáðüðéí

boot

.

Óçìáépóáéò

1. Óå Ýía e-mail áðü ôíí Keith Frechette <kfrechet@us.ibm.com>.

ΕὰοÛεάεί 4 Óõìâáôüôçôá Ôëéëÿ

4.1 ΆάίεëÛ

1. ÈÝèù íá áãñÛòù òëéëù áέα òí FreeBSD óýóçíá ïò. Ðíεί ïíðÝεί / ïÛñέα / óýðìò áβίάέ òí έάέýðãñí;

ÕðÛñ÷ ïòí òòíÝ÷÷ áέα óðæçððóáέð áέα òí èÝìá áðòù óðέð èβóðáð çέάέòñííέέý ðá÷ òãññáβìò òìò FreeBSD. Άðòù ùóòùóí áβίάέ áíáíáíùíáíí, έάèð òí òëéëù òùí òðñέíáέóððí áέèÛæáέ ðñέý ãñðáíñá. Άìáβð *átáèññòèýíá* íá óòíέóðíýíá íá Ýìóáóç, íá áέαáÛóáððá ðέð Óçìáèðóáέð Ôëééý òìò FreeBSD 9.1

(<http://www.FreeBSD.org/releases/9.1R/hardware.html>) Þ 8.4

(<http://www.FreeBSD.org/releases/8.4R/hardware.html>) έάέ íá øÛíáðá óðá áñ÷áβá

(<http://www.FreeBSD.org/search/#mailinglists>) òùí έέóððí çέάέòñííέέý ðá÷ òãññáβìò, ðñέí áñ÷βóáðá íá ñùðÛðá ó÷áðéèÛ íá òí ðáèáððáβì έάέ έάέýðãñí òëéëù. Άβίάέ áñèáðÛ ðέέáíù íá áέαðέóððóáðá, ùðé áέα òí òëéëù ðìò áíáæçðÛðá, òððñíá ó÷áðéèð óðæðçóç ìüééð ðñέí íέα áãñÛÜáá.

Áí øÛ÷íáðá áέα òñçòù òðñέíáέóðð, áèÝáíðá ðá áñ÷áβá ððð èβóðáð çέάέòñííέέý ðá÷ òãññáβìò freebsd-mobile (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile>). ΆέáðññáðéèÛ, ìÛëëíí έá èÝèáðá íá ááβðá ðá áñ÷áβá ðçð freebsd-questions (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>) Þ ðέέáíùí íέα èβóðá ðìò íá áíáέάέýáðáέ óòí óýðì òìò òëééý ðìò øÛ÷íáðá.

4.2 Άñ÷έðáèòñííέéÝò έάέ ΆðáíññáóôÝò

1. Ôðìóççñβæάέ òí FreeBSD áñ÷έðáèòñííέéÝð áέαðññáðééÝð áðù ðçí x86;

Íάέ. Άððð ðç óðéáìð òí FreeBSD ìðññáβ íá áέðáèáóðáβ ðá áñ÷έðáèòñííέéÝð x86 έάέ DEC (ððñá ðèÝíí Compaq) Alpha. Άðù òí FreeBSD 5.0 έάέ ìáðÛ, òðìóççñβæííðáέ áðβóçð íé áñ÷έðáèòñííέéÝð AMD64 έάέ Intel EM64T, ç IA-64 έάèð έάέ ç SPARC64®. Άñ÷έðáèòñííέéÝð ðìò έá òðìóççñβæííðáέ ìáèñíðéèÛ, ðãñέéáíáÛíòí ðçí MIPS® έάέ PowerPC. Ìðññáβðá íá áãññáóðáβðá óðέð èβóðáð ðá÷ òãññáβìò freebsd-ppc (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ppc>) έάέ freebsd-mips (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mips>) áíðβððíé÷á áέα ðèçñíñññáð ó÷áðéèÛ ìá ðçí ðñüñäí òùí ãññáóéðí ðá áðòÝð ðέð áñ÷έðáèòñííέéÝð. Άέα ááíέéÝð ðèçñíñññáð ó÷áðéèÛ ìá íÝáð áñ÷έðáèòñííέéÝð, áãññáóðáβðá ðççí çέάέòñííέéð èβóðá òìò FreeBSD áέα ðέð ìç-Intel ðέáðüññáð (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms>).

Áí òí ìç÷Ûçíá óáð áβίάέ áέαðññáðéèðð áñ÷έðáèòñííέéðð έάέ ÷ñáèÛæáððá έάέðìòñáέéù Ûíáóá, óáð óòíέóðíýíá íá ñβíáðá íέα ìáðéÛ óðá NetBSD (<http://www.netbsd.org/>) Þ OpenBSD (<http://www.openbsd.org/>).

2. Ôðìóççñβæάέ òí FreeBSD Óðñáðñέèð Ðñèðáðáíññáóðá (SMP);

Íάέ. Õí SMP Þðáí áíáñáðñέçíÝñí áðù ðñíáðééñáÞ óòí ðòñÞíá *GENERIC* Þæç áðù òí FreeBSD 5.2.

Ç áñ÷έèð ðñüéáóç Þðáí íá áβίάέ áíáñáðñέçíÝñí áðβóçð áðù ðñíáðééñáÞ έάέ óòí ðòñÞíá ððð Ýέäíóçð 5.3 òìò FreeBSD, áέèÛ èüäù èÛðñέùí ðñíáèçíÛðùí ðóçí áέðÝέáóç òìò ðòñÞíá SMP óá ìç÷áíÞáðá ÷ññβð ðñέéáðñέýð áðáíññáóðÝð, áðñóáóβóçèá íá ìáβίάέ áíáíññáù ìÝ÷ñé ðçí áíðέíáðððέóç òìòð. Άðòù ùóòùóí áðñóáèáβ ðñíðãñáέüðçðá áέα ðçí Ýέäíóç 5.4 òìò FreeBSD.

4.3 Óêëçñĩß äßóêíë, ïïïÜäâò ôáéíßáò, ïäçãĩß CD êáé DVD

1. $\mathbb{D}^{\text{éa}} \mathfrak{a}^{\text{ßac}} \mathfrak{o}^{\text{êeçñ}} \mathfrak{p}^{\text{í}} \mathfrak{ä}^{\text{ßóêùí}} \mathfrak{õ}^{\text{đíóóçñ}} \mathfrak{ß}^{\text{æííóáé}} \mathfrak{á}^{\text{đũ}} \mathfrak{ô}^{\text{í}}$ FreeBSD;

[illegible]

2. Ðiëié äëãäêôÝò SCSI þ SAS õðííoôçñßæííôáé;

Άαββââ ôçí ðēΠñç έββóâά óôêó Óçíâépóâέó Õěéÿÿ ôiõ FreeBSD 9.1
(<http://www.FreeBSD.org/releases/9.1R/hardware.html>) þ 8.4
(<http://www.FreeBSD.org/releases/8.4R/hardware.html>).

3. Ôé ôýďiē iāçāpí ôáéíßàò òďiíoôçñßæiíôáé;

Ôi FreeBSD òðíóçñßæâé iäçãĩyò SCSI êáé QIC-36 (iä interface QIC-02). ÐãñëäiäÛííóáé iäçãĩß 8-mm (ãíuóóĩß ùò Exabyte) êáèò èáé iäçãĩß DAT.

Ἰνέοι Ὑῖνε ἀδῦ οἰῶδ ᾖᾖᾖᾖᾖ ἰαῖῖῖῖῖ 8-mm ᾗᾗ ᾗᾗᾗᾗ ἑᾗᾗᾗᾗᾗᾗ ᾗᾗᾗᾗᾗᾗ ἰᾗ SCSI-2, ἑᾗᾗ ἰᾗᾗᾗᾗ ἰᾗ ἰῖ ἑᾗᾗᾗᾗᾗᾗᾗᾗᾗ ἑᾗᾗᾗᾗᾗᾗᾗᾗᾗ ἰᾗ ᾗᾗ FreeBSD.

4. Õđiíoçñßæáé õi FreeBSD ïïÜäòð áíäëääPò ôáéíéþí (tape changers);

Όι FreeBSD οδίοοçñßæåë ιιŮÛáo áíäëääÞò öýðĩö SCSI ιά ôçí ÷ ñÞóc ôçò óðóêâðÞò ch(4) êáé ôçò áíoīēÞò chio(1). Ιðñāßòâ íá āñāßòâ ôéo ēãðõñÝñæáo ó÷ãoéèŮ ιά õīī õñũðī æēÝā÷ĩ ôçò ιιŮÛáo áíäëääÞò óôç óâēBáá manual õīö chio(1).

Άν αάρ ÷ ηςοείηθιέαβθα οι **AMANDA** π εϋθιει ΰεει θηηιίη θιθ ίά αηηηηαέε θυθ ίά ÷ αέηεοάβ οςι ιηϋάά άάεεααθθ
 θάεηέπ, εά θηΥθάε ίά εοιϋοά υοέ αάιέεϋ οά θηηηηϋηηάθ αηηηηαίθι ιιηθ θυθ ίά έεθθθιθι ίεά θάέίβά αδυ Ύίά οςιαβθ οά
 Ύίά ΰεει, έάε εά θηΥθάε αοάβθ ίά οςιαέρθαθ οά θιέα εΎος (slot) αηβόεαθάε ς θάέίβά, έάε οά θιέα εΎος θηΥθάε ίά
 θϋάε ς θάέίβά θιθ αηβόεαθάε αθθπ ος οόεαηπ ιΎοά οοιθ ιαέαη.

5. Ðiëié iäçãĩß CDROM õđiíoçñßæiĩóáé áđũ õĩ FreeBSD;

Ōđiíoôçñßæåôáé iđiēiôäPđiôä iäçäuò SCSI đĩō àbíáé óoíäâiÝñiò óå áíôßóôie÷á õđiíoôçñæuìâñ ãëääêôP.

Õðíóôçñβæííóáé áêùíà óá áêùèíøèá ìç-ôððíðíèçíÝíá CDROM interfaces:

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

- IDE CDROM óõîâáôÛ ìâ ATAPI

¼ëàò ìé èÙñòàò ðìò ààí àíáíé SCSI àíáíé àáíéèÙ àíáéñàòéèÙ ðéí àñáÝò ùòáí òòàéñíñííóáé ìá áíòòòóíé-íòò SCSI òäçáíýò, éáé èÙòíéíé òäçáíí òýðìò ÒÁÐÌ Òòòò ìá ìç èáéòìòñáÐóíòí.

Ôá âððþçíá FreeBSD CDROM ISO, êáèð êák ôá CDROM áðu ôï Daemon News êák ôï FreeBSD Mall, ððïóðñæïï áêêßíçç ððáðêáßáð áðu ôï CD.

6. Ďiěié řáčãĩß CD-RW õđĩóôçñßæĩíôáé áđũ ôĩ FreeBSD;

Ôi FreeBSD ððřóóçñßææē ððřēřāPðřřā ïäçāü IDE CD-R þ CD-RW óðřāáôü ìā ATAPI. Äåßôā ôi burned(8) æéā ēāððñÝñææð.

Ôi FreeBSD ôðîóôçññææê ãððçò ìðîéññððîã ñãçäü SCSI CD-R ð CD-RW. Åãêããóððóã êáé ÷ñçóéññðîéðóã ôçí áíðîéð cdrecord áðü ôçí ôðêéññ ðüñ ports ð áðü ðáéÝðî, êáé áããáéüèãðã üðé Ý÷ãã áíóüñáððóáé óðñ ððñðñá óáð ôç óðóêãð pass.

7. Õđĩóôçñßæåé ôĩ FreeBSD ĩäçãĩýò Zip;

Οι FreeBSD οδηγούν βάσει αλφαριθμητικού ταυρίδιος Zip SCSI εώς ATAPI. Ής ταυρίδιος SCSI ZIP ιδινιγίς ίς εαεοίωνα ποίος ιυγίς ίς Υ-ί-ί-ίς νωείεοάβς οά SCSI IDs 5 π 6, αεεΰ ίς οι οδηγούν βάσει ς αεαεοπς SCSI θρω ÷ ηςοείιθιεάβθα, ιθιηάβθα αευία εας ίς αεεεπθαοά αθι αοοίγ. Άας άβίας ίαεΰεανί θιείε αεαεοΰ SCSI οδηγούν βάσεις αεεβίςος αθι οοοεαοΰ ίς ID αεαοιηαοεευ αθι 0 π 1, Ύοοε εα θνΎθαε ίς οοιαιρεαοεάβθα οςίς οαείςηβςος θιθ αεαεοπς αοό, ίς εΰεαό ίς ÷ ηςοείιθιεπθαοά αδοπ ος αοίςαοιςοα.

[illegible][illegible]

8. Õđiíoçñßæåē ôï FreeBSD Jaz, EZ êáé Üëëiõð áöáéñiÿiáíiõð iäçãiyò;

Άδερò εάέοιòññáíí. Íé ðáñέóóùòáññò áðu óεò óóóεάòÝò áóòÝò áβίáέ SCSI, έάέ Ýóóέ áββ-ñíοί óái SCSI áβóέíε óοί FreeBSD. Ôí IDE EZ áìöáíβεάòάέ óái íäcäüò IDE.

Âãääéùèàßôâ üøé Ý÷âôâ áíãñāīđīéΠóáé ôõ÷uí âîùôãñêêÝò óõóéãõÝò ðñéí âêêéíΠóãôâ ôī óýóôçíá óáo.

Ãéá íá äëëÛiaôâ ÿÝóí áðrèPêãóçòç áí þñä ëääöirõñãBáo, æÉÝãiôâ ôéo mount(8), umount(8), êáé ôçí camcontrol(8) (ãéá óóôëãöÝð SCSI) P atacontrol(8) (ãéá óóôëãöÝð IDE), êáé åðßòçð ôéo óóæçðÞóáéo ó÷åðééÛ iã ôç ÷ñÞóç åöáénîÿiãúí iãcãþí óå åðüiãûr òiÞiã ôïö FAQ.

4.4 Đệchônĩệũãéá êáé òĩĩôβééá

1. Öđioôcñßæåé ôi FreeBSD ôi USB đęcêôñiëuãéi iio;

Ôi FreeBSD òðíóôçñßæáé åãååíðò USB ðççêôññíëüåéá. ÁíåñåíðíéÞóåå ôçí òðíóôÞñéíç USB óîí /etc/rc.conf.

[illegible][illegible]

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

ΔανάοçΠρόα υόε αί οι ðεçεόντρευαέρ USB άβράε οι ιρίαεέυ ðεçεόντρευαέρ, έα άβράε άεάεΥόείι υò /dev/ukbd0, έάε ç άíοτρεΡ έα άάß÷ίάε υòυò δάναέΥòυ:

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

já êáëü ìÝñìò áéá íá ðñìòéÝóáòâ ôcí ðáñáðÜüü áíòìèÞ, áßíáé ôí áñ÷÷áßi /etc/rc.i386.

Ἰυεὸδ ἀβῖαε ἀδοῦ, οἱ USB θεξεῶνιευαεῖ εἰ δὴ Ὡθαε ἰά εἰεῶιῶναᾶβ εἰε ὀοῖ × δᾶνεἰἸεεῖ, ÷ ὑνβὸ ἀῖἸαεε ἄεἰεεῖ
 ῶεῖβῶῶῖ.

Ç áí çàñìþ óýíääóç éáé áðíóýíääóç USB ðεççêññēīēāñīō, ßòυò íá ιç εάέοιōñāñāñ áέυιā òυòóŨ. Óáo óοίέóοίýīā íá óοίāÝóāòā òī ðεççêññēυāēī ðñēí òçí āēēßīçóç òīò óóóóßíāòīò, éáé íá òī áóßóāòā óοίāñīÝī īÝ÷ñē òīí ðāñīāóέóιυ, āéā íá áðíöýāāòā òō÷ūī ðñīāēßíāòā.

Äåßå öç óäëää manual ukbd(4) äéá ðañéóóüôañåð ðëçñïiñßåð.

2. ÷ù Ýíá ìç-ôõđéëù đííôßéé ôýđíõ bus. Đùò éá ôî ñõèìßóù;

[illegible]

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

Όά θιιόβεέα όγθιό bus όοίPεuό Yñ÷ιιόάέ ιά αέέYό οίωό εUñόαό αθYέόαόό. Άίάα÷ñYíuό ίά Y÷άόά αοίάóuόόόά ίά ñεìβόάόά όγί εUñόά όά αέάοιñάόέεP αέάγέοιόό έγñάό έάέ IRQ αóu αόóU θιό οάβñιόάέ θάñάθUñ. Óοίαιρεάόεάβόά όι άά÷άέñβάέι οίό θιιόέέέY όάό έάέ όε όάέβάά manual mse(4) αέά θάñέόóóuόάñάό έεεññοιñβάό.

3. Дуò ìďĩñþ íá ÷ ñçóëíĩďiéÞóù òĩ đĩíôßéé ïĩ ôýďĩõ PS/2 (“đũñôáo đĩíôéééíý” Þ “đęçêõñĩeĩãßĩõ”);

Ôi ðiíôBêe ôýðîø PS/2 ôðîíôçñBæåôáé ååååîðø. Ôî áðåñåBôçôî ðñuåñåîåîä îäPäçôçð, psm, ðåñééåîäÛîåôé óôîí ðõñPíå.

Áí í ðñîóáññîíîí Ýñò ðññÞñáo óáo ááf öñ ðåñÝ÷æ, ðññöèÝóóà öçí åéüetðeç ãñññÞ öóí äñ÷ãß ñöèìöóäñ ðññÞñá, éæé ïáöääëùðöðöóá ïáfÜ öñ ðññÞñá óáo.

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

Ἰουεὸ ἰ ḍōñPíad̄ áíé÷íáýóáé ouóóŨ ôç ôóôêâōP_{psm0} éáoŨ ôçí âêêβίçόç, ââââéùêâβôâ üêê ḍōŨñ÷âé ç áíôβóōíê÷ç
 éáoá÷ḥñçόç áéá ôi_{psm0} óōñí éáoŨêrāñ /dev. Ἰōññâβôâ íá ôi âçíçíêññâPóáôâ ãñŨōñíôáð:

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

üôáí Ý÷âôâ áéóÝëèâé ùò ÷ñÞóôçò root.

ΌγιαΒυός: ðəŋāβəðá íá ðāŋēāβəðāðā áðəŋ ōī āpīá áí ÷ŋçəŋŋēīðīēāβəðā FreeBSD 5.0-RELEASE p íāpəāŋī íā áíāŋāīðīēçīYīī ōī devfs(5), ēāēəpə ōā āðāŋāBəçəðā āŋ÷āβā ōðŋēāəpī ēā āçīēīŋŋāçēīYī áðəŋīāðā ēŪðŋ áðŋ ōīī ēāðŪēīāī /dev.

4. Ἀβίαέ ἀοίαοῦι ιά ÷ñçóéïïðieçèàß ôi ðïiôßéé ιà èÛðieíi ðñüðï Ýñü áðu ôi ðañéáÛëëíi ôiô óðóôÞiáôïò X Window;

Αί ÷ η̇ςοείηδιεάβω̇α̇ ο̇ι̇ δ̇η̇ι̇α̇δε̇ε̇α̇ι̇Υ̇η̇ δ̇η̇υ̇α̇η̇α̇ι̇α̇ ι̇α̇Ṗα̇ς̇ο̇δ̇ ε̇η̇ι̇ο̇υ̇ε̇α̇δ̇, syscons(4), ι̇δ̇η̇ι̇α̇β̇ω̇α̇ ι̇α̇ ÷ η̇ςοείηδιεṖο̇ά̇ω̇α̇ ο̇ι̇
 α̇α̇β̇ε̇ο̇ς̇ ο̇ι̇ο̇ δ̇η̇ι̇ο̇έ̇ε̇ε̇ί̇Υ̇ ο̇α̇δ̇ ο̇α̇ ε̇η̇ι̇ο̇υ̇ε̇α̇δ̇ ε̇α̇ε̇ι̇Υ̇η̇ο̇ α̇ε̇α̇ ι̇α̇ ε̇U̇ι̇α̇δ̇α̇ δ̇α̇ι̇ε̇ι̇δ̇Ṗ ε̇α̇ε̇ δ̇α̇ε̇ε̇υ̇ε̇ε̇ς̇ο̇ς̇ ε̇α̇ε̇ι̇Υ̇η̇ο̇. Α̇ε̇ο̇α̇ε̇Υ̇ο̇α̇ ο̇η̇ι̇ α̇α̇β̇η̇ι̇α̇
 ο̇ι̇ο̇ δ̇η̇ι̇ο̇έ̇ε̇ε̇ί̇Υ̇, moused(8), ε̇α̇ε̇ α̇ι̇α̇η̇α̇ι̇δ̇ι̇ε̇Ṗο̇ά̇ ο̇ι̇ α̇α̇β̇ε̇ο̇ς̇ ο̇ι̇ο̇ δ̇η̇ι̇ο̇έ̇ε̇ε̇ί̇Υ̇ ο̇δ̇α̇ι̇ α̇ε̇ε̇ι̇ε̇ε̇Ṗ ε̇η̇ι̇ο̇υ̇ε̇α̇:

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

14διὸ οἱ xxxx ἀβίαε οἱ ὑμῖα οοόεαδῶδ οἷο διφδέεερύ εάε οἱ yyyy ἀβίαε ι ογθῖο οἷο δῆνδῖεῦεεῖο οἷο. Ἰ ααβῖαδ οἷο διφδέεερύ ἰδῖνῖα ἰα ἀααῖνβόαε αδοῦιαόα οἱ ἀβῖαδ οἷο δῆνδῖεῦεεῖο αεά οα δᾶνέοοοῦοᾶνα διφδβέα, ἀεοῦδ αῖῦ δαεεῦ οάενεάεῦ ἱνῖοῦεά. Ἐαεῖνβόοα οἱ δῆνδῖεῦεεῖ auto αεά ἰα ÷ ἡζοέῖνδῖεῖβόαδ ογῖ αδοῦιαόε ἀβ÷ῖαοόε. Ἀῖ αοδῖ αἱ ἱνῖοῦεάε, αἱβόα δε οἱεβᾶ manual moused(8) αεά ἰεά εῖβόα ἰα οῖδῖοδνεαῖαῖνδῖο ογθῖοδ δῆνδῖεῦεεῖ.

Ἄν Ὑ ÷ ἀὸα θῖρὸβέε ὀγῶθρ PS/2, ἀδὲρθ ὠῖρὸεῚ ὀδᾶ moused_enable="YES" ὀῶῖ ἀν÷ἀβῖ /etc/rc.conf ᾱῆα ἰᾶ
 ᾱῆῆῆῆῆῆ ἰ ᾱᾱῖῖῖῖᾶ ὀῖθ ὠῖρὸῆῆῆῆῆῆ ῆᾶῖ ὀῆῆῆῆῆῆῆῆ. Ἀδῆῖῖῖῖῖῆᾶᾶ, ἰᾶ Ἀδῆῆῆῖᾶᾶᾶ ἰᾶ ÷ ῖῖῖῆῆῖῆῆῆῆᾶ ὀῖ ᾱᾱῖῖῖᾶ ὀῖθ
 ὠῖρὸῆῆῆῆῆ ὀᾶ ῖῆᾶῖ ὀᾶῖ ᾱῆῆῆῆῆῆῆ ῆῖῖῖῆᾶῖ, ῆᾶῖ ÷ ῆῖ ῖῖῖ ὀῆῆῆῆ ὀῖὀὀᾶῖᾶῖῖ, ὠῖρὸεῚ ὀᾶ ὀῆ ᾱᾱῖῖῖ
 allscreens_flags="-m on" ὀῶῖ /etc/rc.conf.

¼ðáí æêðæððáé ī ææðñáo ðīð ðīðéééíý, ç ðñūóááóç óðī ðīīððéé ðñÝðáé íá óðīðīīæððáé īððáý ðīð ææðñíá éáé Ûéèūī ðñīāñāñÛðūī, ùðùð óá X Windows. ÊíéðÛīðá óðī FAQ óçí āñðóçóç Áéáðð ðī ðīīððéé ñīð āāī āīðéāýáé óðá ×; æáá ðññéóóúðāñāð ðèçñīñīðñáð ó÷:ððééÛ īá áððū ðī ðñūāèçíá.

5. Δὐὸ ἰδιῶπ ἰά ἐϋῖν ἀδιῆιῶπ ἐάε ἀδέεϋεῖς ἐὰν ἰά ὁῖ διῖοβῆε ὁὰ ἰέα ἐῖριόϋεα ἐὰν ἰά ὁῖ;

[illegible]

Αί οἱ δι' ὁβέεε οάο ἀαί Ὑ ÷ ἀέ ἰαοάβι δεΡεοñι, ἰδιñάβ ἰά εἶΥεάοα ἰά οἱ ἀññεπόαοα Ρ ἰά ἀεεῦἰαοα οεο εἰεοἰοñάβαιο οὐἰ
 δεΡεοñῡ ÷ ñçοείῡδιεἰπόαο οεο ἀδεεῖαΥο διῶ δάνΥ ÷ ἰοάε αἰῡ οἱ ἰαβῖñἰα οἱῶ διῶεεείΥ Ἀάβοα οç οἰεβἰά manual
 moused(8) ἰεά οεο εἰδδῶñΥñἰεάο.

6. Ôĩ ðĩĩôßêê ïĩõ Ý÷áé æÛöĩñá Ýĩõðĩá ðēÞēõñá êáé ñĩáÝēá êýēéóçð. ÌðĩñÞ íá ôá ÷ñçóēĩðĩēÞòũ óôĩ FreeBSD;

Ç áðŰĩřóç, äöóð÷þò, áßíáé “áíáñŰóáé”. Óá ðĩĩřßéá ìá Ýĩñá äĩřáðuðçðáð óĩřèùð áðáéçĩřĩ áĩáéçéäĩřĩá ðñĩāñŰĩāóá ãāřāçóçð. Áĩ ðĩ ðñũāñāĩã ãāřāçóçð ðĩð ðĩĩřééçĩř Þ ðĩ áĩðßóĩé÷ ðñũāñāĩã ðĩð ÷ñÞóç äāĩ ðāñÝ÷÷ĩĩř ððäēäēñēĩÝĩç ððĩóðñēĩç áéá ðĩ ðĩĩřßé, éä ēäéðĩðñāā ùð Ýĩá äēũ ðĩĩřßé äĩř Þ ðñēĩř ðēřèðñũĩ.

Άέα δέεαίP ÷ñPóc ôçò ñiaÝéào óå ðãnéáÜëëí X Window, äåßòå ôçí áíóßóðíë÷ç áíuôçóå.

7. Ðuò ìðĩñþ íá ÷ ñçóèiïðieÞòù ôi ðiíôðêé / trackball / touchpad óôíí öiñçòù iìõ õðieíiãéóôÞ;

Äåßôâ ôçí áðŰíôçóç óôçí ðññçãïýìáíç åñßôçóç.

8. $\text{Dùò } i\ddot{o}i\ddot{n}b \text{ íá } \div \tilde{n}\acute{c}o\acute{e}i\ddot{i}\ddot{d}i\acute{e}P\acute{o}u \text{ ôi } \ddot{d}\acute{e}P\acute{e}\hat{o}\ddot{n}i \text{ delete } \acute{o}\hat{o}i \text{ sh } \acute{e}\acute{a}\acute{e} \text{ csh;}$

Άέα δι ΕΎεοοο Bourne, οηιοέΥοοά οεδ άεüεrοεάο αηαüÝο οοί αη÷αβί οάο .shrc. Άαβόά αδβογδ οεδ οάεβααδ manual sh(1) έάέ editrc(5).

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

Άέά ôî ÊŸëöüö C, ðñîóèÝóâ ôéó áêüëîðëâð ãñâîîÝò ôôî ãñ÷âñî óáð .cshrc. Äâðâð äðßòçð ôç óäëßää manual ôïð
csh(1).

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

Ãéá đăñéóóüôăñăò đęcñĩĩöĩñßăò, äăßôăă áõôÞ ôç óăëßăă (<http://www.ibb.net/~anne/keyboard.html>).

4.5 ÓóóâãÕÝò óáéñéáêÞò åðëêëïíúíßàò êáé Äéêôýùòçò

1. Ðięąò êŰñôąò äéêôýĩõ õđĩóôçñæąé ôĩ FreeBSD;

Άέά όící đěPñc ëβóόά, äåßôå ôéò Óçíåéβóåéò Õëëëíý đĩõ đånÝ÷ííóáé ìå êÛèå Ýëäĩóç ôĩõ FreeBSD.

2. Õđiíoôcñbæåé ôi FreeBSD modems đĩõ ẽãéôĩõñĩáĩý íà ôc âiPèãéá ẽĩãéóíéẽĩý üđùò ôá Winmodems:

To FreeBSD ὀδιόοçñßæå ãñêåÕ software modems ìà ὀçì ãìPèæáá áðèðñüøèåðïõ èñæóíéêý. Ôì port `comms/l1tmdm`
ðñíøèÝðæ ὀδιόðPñéìç æéá modems ðïõ ááóßæñíðáæ óõì æñìøéèÝð èýèèùìá Lucent LT. Ôì port `comms/mwavem`
ὀδιόοçñßæå òì modem ðïõ æéáÝðìõí ìé õìñçòìß ὀðìèñæóóðÝð Thinkpad 600 éáé 700 ὀçð IBM.

Ääŕ íðĩñáßöä íá äãëäöáóðÞöäöä öí FreeBSD ìÝöü software modém. Ôî ëĩäëöíëëü áðöü ðñÝðäë íá äãëäöáóðäëß ìäÜ öçí äãëäöÜöäóç öïð FreeBSD.

3. ÕðÜñ÷:æé åååårÝò ðñuãñàííá íäPäcòçò äéá öéò êÜñôäò Broadcom 43xx;

$\frac{1}{4} \div e, \hat{e} \acute{a} \acute{e} \grave{\text{ı}} \ddot{\text{U}} \ddot{e} \ddot{e} \ddot{\text{ı}} \acute{\text{ı}} \text{ ä} \text{ä} \acute{\text{ı}} \grave{e} \acute{a} \text{ õ} \text{đ} \ddot{\text{U}} \text{ñ} \hat{\text{ı}} \acute{a} \acute{e}.$

Ç Broadcom áñíáßðáé íá áþðáé áçíπðóáé ðεçñíðíñßð ð÷ ððééÙ íá ðíí ðñíáññíáððéðíπ ðíí íεíεççñííÝíπ ðçð ðíð ð÷ ñçðéíðíεíýðáé ðá ððáñííáÝð áóýñíáðíπ áééðýπ, ðééáíπ áðáéáP éáé ðí ðíðíá ðíð ðñðíáÝéðç ðçð éÙñðáð

ἁεΨα÷ἁδοᾶε ἰᾶ ὁς ἁἱΠεᾶεᾶ εἱεῶεῶεῶ. Ἀεᾶ ἰᾶ ἁδιῆδῶἶοἱ Ὡᾶεῶεῶς ἁεᾶ ὀεῶ εὔἸῆῶᾶ ὀἱῶδ ἁδι ὀἱ FCC, ὁῆΨῶᾶε ἰᾶ ἁἱᾶῶῶῶῶῶῶῶῶ ὕῶε ἰε ὀᾶεῶεῶῶ ÷ῆἸῶῶᾶ ἁἱ ἔᾶ ἁἱἱᾶε ὀᾶ ἔΨῶς ἰᾶ εὔἸῶἱ ῆῶεἱῶῶᾶῶ ὕῶῶ ἁεῶᾶᾶ ὁς ὀ÷ἰῶῶῶῶᾶ ἔᾶεῶἱῆῆᾶᾶῶ, ὀἱ ὁᾶῆᾶ Ὡῶῆἱ ἁεᾶἱῆῶῶῶῶ ἔᾶε ὁς ὀ÷ῶ ἁεῶῆῶῶῶ. Ἀεῶ ÷ῆῆῶ ὀεῶ ὀεῶῆῶἱῆῆᾶ ὁῆἱᾶῆἱᾶἱᾶεῶῶῶ, ἁἱἱᾶε ὀ÷ᾶἱἱ ἁἱῶἱᾶἱ ἰᾶ ἁῆᾶᾶᾶ ὁῆῶᾶῆἱᾶ ἱᾶἸᾶῶῶ.

4. Ðięàò êŨňàò ðięěáðępí óąęńéąępí ęňpí ðđíoôçņðæiíôáé áđŭ ôi FreeBSD;

ÕđŨñ÷ǎé iéá ěóóâ áéá áõôÝò óôçí áíüôçôâ äëŨöĩñùì óóóêǎõþí

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html#INSTALL-MISC)

Áeúia oáβáíaoáe úoé eáeéoiñáryí eáe eÜüieáo eÜñóao ðiñ áβáíe áíoeáñáoÝð áðpíoiuñ iíróÝeui, áeáeÜ üoáo ðiíooéñβeáíoi úoé áβáíe oóiaáoÝð ia óeó áíóβóoié÷áo óeó AST.

Äåßôâ ôç ôåëßää manual sio(4) äéá đăñéóóúôăñăđ đęçñïöĩñßăđ ó÷ăôéêÛ ìă ôç ñýèìéóç ôÝôïéúì êăñôîí.

5. Дуò ìðĩñþ íá àìöáíβòù ôçí ðñĩôñĩðþ boot: óå íéá óåñéáêþ ëĩíóüëá;

1. ἈῖεῖρῆνᾱΠόδα δὲ Πρία θίω ἰά θᾶνεΥ ÷ αέ ὅςι ἀδέειᾱΒ options COMCONSOLE.
2. ἈῖεῖρῆνᾱΠόδα οἱ /boot.config εἶε ἀνῪθόα ιΥόα οἶ ἀδodu iūfi ὅςι ἀδέειᾱΒ -P.
3. ἈθιόοῖᾱΥόα οἱ δεçêðñīēuāēī αδu οἱ ούόοçíᾱ.

Äåßôâ ôï áñ÷âßï /usr/src/sys/i386/boot/biosboot/README.serial ãéá đăñéóóúôăñăđ đęcñïiöĩñßăđ.

4.6 ÓóóâãõÝò Þ÷ïõ

1. **Điẻàò êÛñôàò Þ÷ĩõ òđĩóóçñßæĩĩôáẻ áđũ ôĩ FreeBSD;**

Οἱ FreeBSD ὀδηγόχρηστέες ἀεὶ τοῖντο εὐχόμεθα πρὸς, ὁμιλητέες αἰσθητῶν ἡμῶν οὐρ SoundBlaster®, SoundBlaster Pro, SoundBlaster 16, Pro Audio Spectrum 16, AdLib, ἐὰν Gravis UltraSound (ἀεὶ ὁμιλητέες τοῖντο ὁδεχτῶν τοῖντο, ἀλλὰ ὁδεχτῶν τοῖντο ἐκτὸς τοῦ FreeBSD (<http://www.FreeBSD.org/releases/>) ἐὰν ὁδεχτῶν τοῖντο manual snd(4)). Ὁδεχτῶν τοῖντο ἀδελφὸς ὁμιλητέες ἡμῶν ὀδηγὸς χρεὶς ἀεὶ εὐχόμεθα MIDI ὅτι ἀβρίας ὁμιλητέες ἡμῶν ἰὰ ὁδεχτῶν τοῖντο MPU-401. Ἀδελφὸς ὀδηγὸς χρεὶς ἀεὶ εὐχόμεθα ὅτι ἀβρίας ὁμιλητέες ἡμῶν ἰὰ ὁδεχτῶν τοῖντο Microsoft® Sound System.

Όχιαβύθος: Οἱ δαῖαδῶϊυι εὐ÷γὰε ιυῖῖ ἑάε οἶῖ P÷ἱ! Οἱ δῆυᾱᾱἱἱἱἱ ἱᾱPᾱççòç ᾱἱ ὀδῖόδçᾱBᾱᾱε ὀ÷÷ῖ CDROM, SCSI P joysticks δῖῖ ὀδῖᾱῖῖῖῖῖῖ δῶῖυι ὀᾱ ᾱὀὀῖὀ ὀὀὀ ἑῖᾱᾱὀὀ, ᾱὀὀὀ ᾱὀὀ ὀçῖ SoundBlaster. Ἀῖ ἑάε ç ἑᾱᾱὀᾱὀP SCSI ὀçὀ SoundBlaster ἑᾱᾱὀ ἑᾱᾱ ἑῖὀῖᾱ ἱç-SCSI CDROM ὀδῖόδçᾱBᾱῖῖὀᾱᾱ, ᾱἱ ἱδῖῖῖῖῖ ὑὀὀὀὀῖ ἱᾱ ÷ᾱçὀἑῖῖὀῖçἑῖῖῖ ἑᾱᾱ ὀç ἑᾱᾱἑἑᾱὀᾱ ᾱἑἑῖῖçὀçὀ.

2. $\tilde{O}\tilde{\delta}\tilde{U}\tilde{n} \div \acute{a}\acute{e} \acute{e}\tilde{U}\tilde{\delta}\tilde{i}\acute{e}\acute{a} \acute{e}\acute{y}\acute{o}\acute{c} \acute{a}\acute{e}\acute{a} \acute{o}\tilde{i} \acute{\delta}\tilde{n}\tilde{u}\tilde{a}\acute{e}\tilde{c}\tilde{i}\acute{a} \acute{o}\tilde{i}\acute{o} \text{P} \div \tilde{i}\acute{o} \acute{o}\acute{o}\acute{c}\acute{i} \acute{e}\tilde{U}\tilde{n}\acute{o}\acute{a} \tilde{i}\tilde{i}\acute{o} \acute{\delta}\tilde{i}\acute{o} \acute{o}\tilde{\delta}\tilde{i}\acute{o}\acute{o}\tilde{c}\tilde{n}\beta\acute{x}\acute{a}\acute{o}\acute{a}\acute{e} \acute{a}\acute{\delta}\tilde{u} \acute{o}\tilde{i} \text{pcm}(4);$

Εὐθιέας ἐϋνόας π-ῖο, ὑδὸν ς es1370, ἱσαίβαϊοί οἱ Ἰόας οἱ π-ῖο αἱ ἐὺα ἀεβίος. Δὴ ὕθας ἰα ἀεαεαβὰ οἱ ἀεὺεὸς αἰοίεπ ἐὺα οἱ ὅθι ἰαεῖ ὅι ἱ-ῖ:

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

4.7 ¶ëëï òëéêü

1. Ðięąđ Üęęąđ óőóęąđÝđ őđięóőčņβæięőąé áđű őię FreeBSD;

 $\text{Äßôâ ôï Äã÷äéñßäéï}$

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html#INSTALL-MISC) ἁέά ὅς ἐβόδᾱ οὐί
 ὀδῡεῖδῡί ὀδῶἑἄοβί θῡῡ ὀθῡὀδῶñβæῖῡἰὀἁἑ.

2. Ὁδὸς τοῦ FreeBSD ἀπὸ τοῦ Linux

Άδου οι FreeBSD 4.X έαέ ιαòÛ, òðìíóçñßæàóáé οι APM óά óóääêñêñíÝíá ιç÷:άίPìáíóá. Ðãñéóóúóãñò ðççññííßò ìðñãßòá íá ãñãòá óóì apm(4).

[illegible]

3. Dùò iǒiǎǎ íá áđáíáñǎǎiđiéPóu ôi ACPI;

ĐñĩóèÝóôầ ôç ãñàìP

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

óôï áñ÷åßï /boot/device.hints.

4. $\tilde{A} \acute{e} \acute{o} \beta \acute{o} i$ Micron $\acute{o} \acute{y} \acute{o} \acute{o} \acute{c} \acute{i} \acute{a} \acute{i} \acute{i} \acute{o} \acute{e} \acute{n} \acute{a} i \acute{U} \acute{a} \acute{e} \acute{e} \acute{a} \acute{o} \acute{U} \acute{o} \acute{c} \acute{i} \acute{a} \acute{e} \acute{e} \acute{B} \acute{i} \acute{c} \acute{o} \acute{c} ;$

Ἰνεοί Ὑῖαο ἰχθονεῖ Ὑο Micron οὐδῖαιεῖ ὁι PCI BIOS ἰα ἰχ ὁδῖδῖεῖ Ὑῖι ὁνῖδῖ, ὁνῖεαῖρῖαο ὁνῖαῖρῖαο ὁοῖ ἁεῖβῖς ὁῖ ὁ Free BSD, εἶεῖ ἰε PCI ὁοῖαο Ὑο ἁῖρ ὁνῖεαῖρῖαο ὁοῖ ἁεῖβῖς ὁοῖ ὁ ἁῖαο Ὑῖρῖαο.

Āéá íá đāñāēŪiøāōā ôi đñüāēçíá, áđāíāñāĩđiēPóōā ôçí āđēēĩāP “Plug and Play Operating System” áđü ôi BIOS.

5. Ç äéóēÝôá âêêßícóçò êñàìÛäé óóç ìçônéêP ASUS K7V. Ðùò ìðĩñþ íá ôĩ äéĩñèþòù áõõù;

Óôèò ñõèìáóáèò ôiõ BIOS, áðáíáñáíðíéPóôâ ôçí áðéëíãP “boot virus protection”.

6. Άέάοß ç PCI êÛñóá äééôýĩö ñò ôçð 3Com® äái ëäéoïrñãåß ìà ôì Micron öðieiaéoóp ñò;

ἸνεοίΥἱὰο ἱϷονῆεῖΥὸ δϷὸ Micron Υ̅ ÷ ἱϷις ἱϷ-οδῶἰῖῆεῖςΥἱ PCI BIOS οἱ ἰῖῖἱ ἁὰρ ἡῶἱβᾶεᾶ οῶδ οῶοῶᾶῶΥὸ PCI οῶῶ ᾶῶᾶῶῖῖῶᾶῶ δῖῶ ᾶῖῶᾶΥἡἱῖῖᾶῖ. Ἀῶῶ ᾶῖῖῖῖῖᾶᾶᾶ δῖῖᾶῖᾶᾶᾶ ῖῖῖῖ ῖῖῖῖῖῖ ῖῖῖ ᾶῖῖῖῖῖῖ οἱῶ FreeBSD.

Άέά ίά δάñáêÛìøåôå ôì ðñüæçìá, áðáíáññíðìéßóåå ôçí áðéçíāß “Plug and Play Operating System” áðü ôì BIOS.

7. Ç PCMCIA êÛñôá ïïö äáí êácôïrñãâß. ÂëÝðù òï äïðò ìÞïíá: “cbb0: unsupported card type detected.” Óé ìðñþ íá êÛñù;

Ἰδιῆναβὺς ἰά ἀρεῖσι Ὑόαδὰ ἰά ÷ ἡγοεῖν διέτ' ἐπὶ αὐτῇ οὔρι ἀν÷έει' περὶ δὲ βύος OLDCARD. Ὅνῃ δὲ διέτ' ἐπὶ αὐτῇ οὔρι ἀν÷έει' νύει εὐρύοιο
οἴω δὲ ὅτι Πρία οἶον, ἐὰν ἀσθέν' Ὑόαδὰ οἶον ἀεὺ εὐροῦσθαι ἀνάη' Ὑόν:

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

ÌàôÛ ðñïóèÝóôâ:

```
device pcic
device card 1
```

İaôâæüôðßóâ íáŬ éâé ââéâóðßóâ ôî ŕŶĩ ðõñÞía üðùò ðãñéãnŬôâóâé óôçí Ñýèìéóç ÐõñÞía ôïð FreeBSD
(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/kernelconfig.html).

ΕὰοÛεάέϊ 5 Αίôείαôþðέος ÐñïäēçïÛôùϊ

1. Ἀέάðß ðï FreeBSD äñßóēäē ËÛēïð ðïóïðçðά ιίÞìçð;

Ἀðòù τòäßēäðäē óðç äέάοïñÛ ïäðáíý òðóέēþï éäē äέέïíέēþï äέäðέýíòäùï ιίÞìçð.

Ç óýíäáós ðïò éäòÛ äÛós äέτēïðēäßðäē óðï ðέέέù ðïò PC, äßíäē íá ÷ñçóέïïðñέäßðäē ç ιίÞìç ïäðáíý 3.5G éäē 4G äέä äέäέέù óέïðù, óðíÞèùð äέä óçï ðñùóäáós óä ËÛñòäð PCI. Ἀðòù Ý÷äē ùð äðïðÝέäóíä íá ιçï ïðññäß íä áíðέóðñέ÷çèäß òðóέēÞ ιίÞìç óä äððÞ óçï ðäñέï÷Þ äέäðέýíòäùï.

Ôï ðέέέù ðïò ððñέäέóðð óäð ää äέέïñßðäē ðē äßíäðäē ïä óçï ιίÞìç ðïò éäñíέέÛ äïðáíßæäðäē óä äððÞ óç òÝós. Ἀðóðð÷Þð, óä ËÛðñέäð ðäñέððððäέð ðï ðέέέù ääï ËÛíäē ðßðñíä, éäē ÷Ûíäðäē ç äðñíäðùðçðä ÷ñÞóçð ðùï ðäēäððäßùï 500ï ιίÞìçð RAM.

Ἀððð÷Þð, óðέð ðäñέóóùðäñäð ðäñέððððäέð ðï ðέέέù áíäέäðäðέýíäē óç ιίÞìç óä ðççέùðäñç òÝós, þðä íä äßíäē äέùíä äðñäðÞ ç ÷ñÞóç ðçð. Ἀðòù ïðññäß ùððùóï íä óäð ðññέäēÝðäē ËÛðñέä óýä÷ðós áí ðäñäέτēïðēäßðä óä ιçïýíäðä äέēßíçðçð.

Óóçï 32 bit Ýέäïós ðïò FreeBSD, ç ιίÞìç ðäßíäðäē íä Ý÷äē ÷äèäß éäèðð áíäέäðäðέýíäðäē ðÛíù äðù óä 4G, óä ïðñä ääï äßíäē ðññóäÛóέíä äðù 32 bit ððñÞíä. Óóçï ðäñßððððä äððÞ ç èýós äßíäē íä ððέÛíäðä Ýíä ððñÞíä óýðñò PAE. Ἀäßðä äððÞï óçï éäðä÷Þñçós óðï FAQ äέä ðäñέóóùðäñäð ðççññññäð.

Óóçï 64 bit Ýέäïós ðïò FreeBSD, Þ ùðäí ÷ñçóέïïðñέäßðäē ððñÞíäð óýðñò PAE, ðï FreeBSD ää áíέ÷íäýðäē éäē ää áíäέäðäðέýíäē ðùóðÛ óç ιίÞìç þðä íä äßíäē ÷ñçóέïïðñέðóέìç. ἘäòÛ óçï äέēßíççós ùððùóï, ïðññäß íä ðäßíäðäē ùðē ðï FreeBSD áíέ÷íäýäē ðäñέóóùðäñç ιίÞìç äðù äððÞ ðïò Ý÷äē óóçï ðñäñíäðέέùðçðä ðï óýóðçíä. Ἀðòù äßíäē ððóέτēïäέέù éäē ç äέäēÝóέìç ιίÞìç ää äέïñèùèäß éäèðð ãέτēçñññññäðä ç äέääέäðäðä ðçð äέēßíççð.

2. ï óέççñùð ïï äßóέïð Ý÷äē ÷äέäóïÝññð ðññäð. Ôé ïðñÞ íä ËÛíù;

Óóïðð äßóέïðð SCSI, ï ãäçäùð ïðññäß óðíÞèùð íä äðäíäððñέäðððäē äððùíäðä óä ääññÝíä óä áíäέäέðέέέýð ðññäð. Ûóðùóï ãέ ðäñέóóùðäññέ äßóέïé Ýñ÷íðäē ïä óçï äðñíäðùðçðä äððÞ äðäñäñäðñέçïÝíç.

Ἀέä íä áñäñäðñέððäðä óçï äðäíäððñέðçðçós ÷äέäóïÝíù ðññÝùï, äðäñäñäðäððäðä óçï ðñÞðç óäēßää éäòÛððäóçð ðçð ððóέäððð (modepage), äßññíðäð óçï ðäñäēÛðð äíðñέÞ (ùð root):

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

éäē äέēÛíðä ðέð ðέíÝð ðùï AWRE éäē ARRE äðù 0 óä 1:

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

Ïé óýä÷ñññέ ãäçäñß óýðñò IDE Ý÷íðï äðßóçð áñäñäðñέçïÝíç äðù ðï äñäñóðÛóέï ðç äðñíäðùðçðä äðäíäððñέðçðçð ÷äέäóïÝíù ðññÝùï.

Ἀí ääßðä ðññäέäññέððäέð ð÷äðέέÛ ïä ÷äέäóïÝññð ðññäð (óä ïðñέτððððä ääñð äßóέïð), äßíäē þñä íä óέäððäßðä íä äέēÛíðäð ðññ ãäçäù. ðòðð ïðññÝðäðä íä ÷ñçóέïïðñέððäðä ðï äέääñùððέέù ðññäñäñä ðïò äßíäē ï éäðäóέäðäðððð ðïò äßóέïð äέä íä äðñññÞðäðä ðïðð ÷äέäóïÝññð ðññäð, äέēÛ óóçï éäέýðäñç ðäñßððððçós äðèðð ää èäñäßðäðä èßäï ðäñέóóùðäññ ÷ñññ.

ó÷âœêÛ ìà ôçí ÃãéáôÛóôáóç ÂêôõðùôÐ

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/printing-intro-setup.html).

7. Άέάοβ όά θñιανÛιáόά ιιό θãñέοόάόέάέÛ όãñιáόβæιόι ιá όóÛειá Signal 11;

Óa óoÙeíaoáa òyðĩo Signal 11 açiérõnãĩyĩoáe uoáĩ iéa aeãnaáoBa ðnióðaeãB ía ðnióðaeÙoáe ðañeĩ÷P ífPĩçò aeá òçĩ
 iðĩBa aãĩ Y÷ae ðÙñae Ùaaéa aðu òĩ eãeóĩõnãeêeú óyóoçĩa. Áí óoĩaaBíae êÙoe òYðieĩ oá oáeĩmãíeêÙ òð÷ãBa
 ÷nĩieêÙ aeáóoPiaóá, eá ðñYðae ía añ÷Boáoã ía òĩ añaóĩÙoá ðiey ðnióaeóeêÙ.

Ôá ðñiäëPiáôá áôôÜ óóíPèuò iöäßeñiôáé óá êÜðieií áðu ôiöð ðáñáéÜòù euäiöð:

1. **Αί οι θηυαεχιά αιθαίβααοαε iufi oá iea oðaeañeiYíc aòañiāP oçí iðiBa áíaðòýoóāoā āoāBò, āBíae ðeeāíPò eÜeìò oóii aeèu oáo epaeéa.**
2. **Αί οι θηυαεχιά āñBoeāoae oā òiPiā oìo āaoeēý oóoòPiāòìò oìo FreeBSD, iðñāB āðBoçò íā āBíae ðñiāeçíāoēeùò epaeéao, aeèÜ oéo ðāñeóoúoāñāò oñYò, oā ðñiāePiāoā aòoÜ āñBoeñiōae éae aeñepñiōae ðñeí aeáíaiçeíý oóìòò ðāñeóoúoāñiòò aðu āoÜò ðìò aeáāÜaeāoā oì FAQ (aeá oì eüāi aóou Üeèuóoā oðÜñ÷ae éae ç āñāñP áíÜðoðìçò -current).**

Ā́éá dāñŨāāēāiā, Ýíáo āñPāĩñĩō ðñũðĩō íá áéáðéóôpóáôâ ũôē *uui* ðñũĉāēôáē áéá ðñũuāēĉiā ôĩō FreeBSD, āñíáē áí ôĩ
 ðñũuāēĉiā āìôáíĤæāôáē éáóŨ Őĉ íáoáāēpôôóĉ ēŨðĩēĩō ðñĩāñŨiāôĩō, áēēŨ ēŨēā ôĩñŨ éáē óá áéáôĩñāôēēū óĉiāñĩ.

[illegible]

Ôé ðñÝðăé íá êŮíăôă:

Όσοι δηλαδή θα ήθελαν να δοκιμάσουν το `gdb` σε ένα πρόγραμμα που έχουν γράψει, τότε μπορούν να χρησιμοποιήσουν το `gdb` ως εξής:

Óôç ääýôãñç ðãñßðòùôç, èá ðñÝðáé íá áðáεçèäýóãòã üôé äáí öôáßáé ôí öééèù óáo.

Óôèò óóíçèéóíÝíàò áéôßàò áóôíý ôîõ ðñíîæÞíáôîò, ðññééîîâÛííîáé:

1. **Ι**ε **ο**εεεηηβ **ο**α**ο** **α**β**ο**ε**ι**ε **ι**θ**ι**η**α**β **ι**α **ο**δ**α**η**ε**α**η**ι**α**β**η**ι**ο**α**ε**. **Α**ε**Υ**α**ι**ο**α** **υ**ο**ε** ε**α**ε**ο**ι**ο**η**α**ρ**ι** **ι**ε **α**ι**α**ι**ε**ο**ο**η**π**η**α**ο **ο**ο**ι** ε**ι**ο**ο**β **ο**α**ο**. **Α**ι **α**α**ι** ε**α**ε**ο**ι**ο**η**α**ρ**ι**, **α**β**ι**α**ε** **δ**ε**ε**α**ι** **η**ε **α**β**ο**ε**ι**ε **ο**α**ο** (ε**α**ε **β**ο**υ**ο **ε**α**ε** **Υ**ε**ε**α **α**ι**α**η**ο**η**ι**α**ο**α) **ι**α **ο**δ**α**η**ε**α**η**ι**α**β**η**ι**ο**α**ε**.
2. **Ι** **α**δ**α**ι**α**η**α**α**ο**ο**η** **ο**α**ο** **Υ** ÷ **α**ε **ο**δ**α**η**ε**α**η**ι**α**ε**α**β: **Α**ο**ο**υ **ι**θ**ι**η**α**β **ι**α **ο**ι**α**α**β** **ο**α **δ**α**η**β**δ**ο**υ**ο**ς** **ο**ι**ο** **ο**η **ε**α**ε**ο**ι**ο**η**α**α**β**ο**α **ο**α **ι**α**α**α**ε**γ**ο**α**η**ς **ο**ο ÷ **ι**υ**ο**ς**ο**α **α**δ**υ** **ο**ς**ι** ε**α**η**ι**ε**η**ε**η** (overclocking) **η** **α**ι **ο**ι **α**ι**α**ι**ε**ο**ο**ς**η**Υ**ε**ε **ο**ι**ο** **α**δ**α**ι**α**η**α**α**ο**ο**η** **Υ** ÷ **α**ε **ο**ο**α**ι**α**ο**η**α**ε** **ι**α **ε**α**ε**ο**ι**ο**η**α**α**β. **Ο**α **ε**Υ**ε**α **δ**α**η**β**δ**ο**υ**ο**ς**, **ε**α **δ**η**Υ**δ**α**ε **ι**α **α**ι**α**ο**α**ε**β**ο**α**ο**α** **υ**ο**ε** ÷ **η**ς**ο**ε**ι**ο**ι**θ**ι**ε**α**β**ο**α **ο**ι **ο**ε**ε**υ **ο**α**ο** **ο**γ**ι**ο**ι**α **ι**α **ο**ε**ο** **δ**η**ι**α**ε**α**η**α**ο**Υ**ο** **ο**ι**ο**, **ο**ι**ο**ε**Υ** ÷ **ε**ο**ο**η **α**ε**α** **υ**ο**ι** **α**ε**Υ**ο**ο**ς**ι** ÷ **η**α**ε**Υ**α**ο**α**ε **α**ε**α** **ι**α **α**δ**ε**γ**ο**α**ο**α **ο**ι **δ**η**υ**α**ε**ς**ι**. **Α**ε**α** **δ**α**η**Υ**α**ε**α**ι**α**, **α**ι **Υ** ÷ **α**ο**α** **ε**Υ**ι**αε **overclocking**, **α**δ**ε**ο**ο**η**Υ**ο**α** **ο**η **α**δ**α**ι**α**η**α**α**ο**ο**η** **ο**ο**ς** **ε**α**η**ι**ε**η**ε**η **ο**ι**ο** **ο**ο ÷ **ι**υ**ο**ς**ο**α.

[illegible]

3. Ðñĩāēçĩāōēēþ ĩþĩç: Áĩ Ý ÷ āōā āāēāōāōōçĩÝĩā ðāñēōōūōāñā āðū Ýĩā SIMMS / DIMMS, āōāēñÝōāā ōā ēāē ðñĩōðāēþōōā ĩā ēāēōĩñāþōāā ōĩ ĩç÷Üĩçĩā ĩā Ýĩā-Ýĩā ÷ñēōōÜ þōōā ĩā āĩōĩðōāōā ōĩ ðñũāēçĩā ōā āððāāĩ āũō SIMM / DIMM, þ þōūð ōā Ýĩā ōōĩāōāōĩū ōĩōð.

- [illegible]

[illegible]

ÔÝēřò, áí òððřòá áđũ òá đáñáđŨřũ áář ářçèðóáé, áßřáé đééářũř íá Ý÷:áoá ářòřðßóáé Ýřá đñũâēçřá (bug) óóř FreeBSD éáé èá đñÝđáé íá áēēřēōēðóáòá óéđ řāçāßáo áéá íá óóáßēáòá ářáõřŨ đñřāēðřáòēđ.

İđĩñâßôâ íá ãñâßôâ âêôâôâĩÝıç áíÜëöóç óôĩ FAQ ó÷âôêêÜ ıâ ôĩ đñüâêçıá SIG11. (<http://www.bitwizard.nl/sig11/>)

8. Ôi óyóôçíá iĩřó óóáiáoÜäç äßôâ íâ Fatal trap 12: page fault in kernel mode, P íâ panic:, ääß÷ñííóáo äáé íéá óáénÜ ádu ðecñíöĩñßò. Óé ðñÝðäé íá êÜü;

Ç n'Üää ar'Üðððððð ðð FreeBSD áñáááÝñáðáé éáéáßðáñá áéá áððÜ ðá e'Üèç, áéèÜ ÷ ñáè'Üæáðáé ðáñéðóððáñáð ðèçñððððððð áéððð áðð ðð ð'ðððá e'Üèððð ððð áe'Ýðáðá. Áðéçáñ'Üððá ðð ðè'ðñáð ð'ðððá éáé 'Ýðáéðá ðððáððéáðéáßðá ðçð áñððçðá ððð FAQ ð÷-áðéèÜ ðá ðá kernel panics, áçððððñá'ððá 'Ýðá ððñ'ððá ðá áððáðððçðá áéððáé'Üðððçð (debugging kernel) éáé áéðáé'Ýðá 'Ýðá backtrace. Áððð ðððñáß ðá áéððáðáé áýðéèè, áéèÜ ááð ÷ ñáè'Üæáððá ððçð ðñáñáðééèðçðá áñ'ðáéð ðñðñáñáðééððð. Áñéáß ðá áéèðððððáð ðéð ðäçáßáð.

9. Ἄέαδβ ς ièuíc iĩõ iãõñbæåæ éæé ÷ Üíæ õĩ õõã÷ñĩíéõĩũ õçð èáõÜ õçĩ æêêßĩçóc;

Δημιέαδασ ἀέα ἀφουόυ δημιέαγια ιά ογι εὔηοα ἀναόεερί ΑΤΙ Mach64. Οἱ δημιέαγια ἀβιάε υόε ς εὔηοα αόορ
 ÷ ηςοέριθιέαβ ογι αέαγέοιόρ 2ε8, ς ἰθιβά ÷ ηςοέριθιέαβόας ἀδβόγοδ εας αδου ογι οὔοαηόρ οάεηέαςρ εγῆα. Εὔαυ
 εὔθιερο δημιέαβιόορ (ρ ογοδ ο÷ααβόγοδ) οιο δημιανῦλιαδιο ἱαβόγοδ σιο(4), οἱ δημιανῦλια υ÷ε ἱυρι εα δημιόδαερόας ἱα
 αίε÷ἱαγόας αόορ ος αέαγέοιόρ αέυια εας αἱ αἱρ ὕ÷αοα οὔοαηόρ οάεηέαςρ εγῆα, αέεῦ αέυια εας οόγι δαηβδδουός θιρ
 ὕ÷αοα αδαἱανῖθιερόας ος οάεηέαςρ εγῆα σιο3 (αῆε. ογι οὔοαηόρ) ς ἰθιβά οόοέριεἱαέεῦ ÷ ηςοέριθιέαβ αόορ ος
 αέαγέοιόρ.

İÝ÷ñé íá æéñèùèåß ôî ðñüâëçíá áõòü, ìðñåßòå íá ÷ñçóéíìðñéßóåòå ôî ðåñåêÛòü ôÝ÷íáóíá æéá íá ôî ðåñåêÛìøåòå:

1. ἌνῸϋόᾱ – ϙ όόϙί ḡñōññḡᐅ ḡēēᐅίϙόϙ. (Ἰᾱ ḡñ ḡñṽḡ ḡḡḡ ḡᾱ ḡῸēᾱḡ ḡñ ḡñᐅᐅᾱ ḡᾱ ḡᾱḡῸḡᾱḡ ᐅ´ēἰέόϙ).
2. Ἀḡᾱñᾱñᾱñḡᾱᐅᐅᾱ ḡḡḡ sio0, sio1, sio2 ḡᾱ sio3 (ṽēᾱḡ). Ἰᾱ ḡñ ḡñṽḡ ḡḡḡ ḡñ ḡñṽᾱñᾱ ḡᾱᐅᾱϙόϙ ḡᾱ ḡᾱñᾱñḡᾱᐅᾱᐅᾱ ḡᾱ, Ὸñᾱ ḡᾱ ḡ´ἰἰḡñᾱᐅᾱᐅᾱ ḡñṽᾱḡ´ᾱ.
3. ἌνῸϋόᾱ exit ḡᾱ ḡᾱ ḡḡᾱ÷ᐅᾱḡᾱ ḡ´ ḡēēᐅίϙόϙ.

Áí ē Yēāōā íá ÷ ñçōéíīđīē Pōāōā óéō óáēñéáē Yō ē yñāō, ēá đñ Yđāē íá āçīēĩōñā Pōāōā í Yí đōñ Píá, íā ôçí áēüēĩēōç
íāōāōñīđP: Óōí āñ ÷ ābí /usr/src/sys/i386/isa/sio.c āñāōā ôĩ đñbōĩ ócíābí đĩō àìōáíæāōáē ôĩ

[illegible]

Άνευα εαε ιαδὺ ὀγι ἀσάνιαιP ἀδὸβι οὐι αέιηεβσάυι, Βσὺδ αίαεαέγσάδσ υδε δι ούόογια X Window ααί εαεὶοιῖαB ουόδὺ. Αί οσιαBαίε ἀδὸυ, ααααεὺεαBσά υδε ÷ησὶοίιθιεαBσά Ἰεαίος 3.3.3 P ιαααέγδανς οἷσ XFree86™. Ἄδὺ ὀγι Ἰεαίος ἀδὸP εαε ιαδὺ, δδὺñ÷αε αίσὺιαδὺιΥίς δδἰόδPηείς αέα εὔῖῖσδ Mach64 εαε ἀδβὸςδ αέαδBeαδσε αίαεαεεαὶΥἰδ ἀιδδςῖαδςδPδ X αέα ὀγι εὔῖῖσδ ἀδὸP.

10. Ἄεάοθ οι FreeBSD όύόόçιά ιιό ÷ñçóειῖδιέάβ ιιüí 64MB RAM, αίρ ι ððĩẽĩαέóòðò ιιό Ý÷άε άάέáoάóóçìÝía 128MB;

[illegible][illegible]

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

140ĩōō òĩ ñ āĤĩāē òĩ ĩŸāāēĩō òçō ĩĤĩçō òā kilobytes. Āēā ĩç÷Ÿĩçĩā ĩā 128 MB, ēā ðñŸðāē ĩā ÷ñçōēĩĩðĩēĩōāōā òĩ 131072.

11. Οι ούοόçιά ιῖθ ὲ÷άε δᾶνεόούδᾶνι ἄδῃ 1 GB RAM, ἐάε δᾶβῆνῦ panics ἰὰ ἱçýíᾱḁᾱ “kmem_map too small”. Δῖθ ᾱβίᾱέ οἱ δῇüᾱεçιά;

[illegible]

ΆçìεrōnāPóðā òì áέέü óáð ðññPíá, éáé ðñìøéÝóðā òçì áðέέēīāP VM_KMEM_SIZE_MAX óòì āñ÷āñì ññèìßóāuì òìø, þóðā íá áòìPóáðā òì ÌÝáέóòì ÌÝāāèð óā 400 MB (options VM_KMEM_SIZE_MAX=419430400). Óā 400 MB óāßíāðáé íá āðāñēīýì áéá ìç÷āìPíáðā ìā ÌÝāāèð ìPìçð ùð 6 GB.

12. **Ôi óýóóçíá ññ ããí Ÿ÷ãé 1GB RAM, éáé ðŸéé ùùòò ðí FreeBSD áçíéíõñããß panic ìä ðí ìþíðíá kmem_map too small!**

Οἱ panic ἀὰβ÷ἱὰε ὑδὲ οἱ οὔοδῖα ὕ÷ᾶε ἱαβῖαε ἀδῦ ἀεῖῖῖεῖβ ἱβῖς ἀεά ὀνῖοῦνῖβ ἀδῖεβᾶοδς ἀαῖῖὕῖῖ ἀεέοῖῖο (network buffers, ἐὰε ἀεᾶεῖῖοᾶῖ mbuf clusters). Ἰδῖῖᾶβῖο ἱά ἀῖῖβῖοᾶ οἱ ὕᾶαεῖ οδς ἀεῖῖῖεῖβ ἱβῖς δῖο ἀεᾶδβᾶοᾶε ἀεά mbuf clusters, ἀεῖῖῖῖῖῖῖ οδὲ ἱᾶᾶβᾶδ οδς ἱῖῖῖῖᾶ ¼ῖᾶ Ἀεέοῖῖο (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-

16. ¼ðáí íáëéíí òí òýóðçíá ïò ðáβññü òí εÜèò ahc0: illegal cable configuration. Ç éáèüäβüóç ïò áβíáé ðüòðð. Ôé ððíááβíáé;

Ç içönéeP ðeéeÝóá oáo aái Ý÷áe óá aðáeoiyiiaíá aiúoáñeéÜ eñeepiáoá þóóá íá oðíróoçñBæáe áoóuiáoí oáñiáoéoiú oíó
 áeáyēiō SCSI. Áíōþ íá ááoBæáóóá oóii áoóuiáoí oáñiáoéoiú, açēþóóá oíi SCSI BIOS oíi ouóóu oáñiáoéoiú áeá oç
 áeÜóáiç oóóeáðopi ðiō Ý÷áoá. Ôi ðñuāñaiá iāþaçóçð oíō AIC7XXX aái iðiniāþ íá eáeinþóáe ái āþiáe áeáeÝóeií oí
 éýēēuiá ðiō ÷ñçóeiíðieāþáe áeá oçí aíþ÷íáóóç oíō eáēuaþiō (Üñá eáe oíō áoóuiáoíō oáñiáoéoiý). Ôi ðñuāñaiá
 iāþaçóçð oðieÝóáe uóe oðÜñ÷áe oðíóóþñeíç, áoúoií iē ñoēiþóáeð ðiō ðañēÝ÷ííóáe oóç óáēñeáēP EEPROM
 áíáoÝñiōi "áoóuiáoí oáñiáoéoiú". Óó÷íÜ, ÷ññþò ði aiúoáñeēü éýēēuiá aíþ÷íáóóçð oíō eáēuaþiō, oí ðñuāñaiá
 iāþaçóçð eá ñoēiþæáe eáieáoíÝíá oíi oáñiáoéoiú, eÜóe ðiō iðiniāþ íá açíeiōñāþóáe ðñuāeçíá oóçí áíeiðeóóþá oíō
 áeáyēiō SCSI.

17. Ἄέαὐβ ὁἱ Sendmail ἄβἰάε ὁἱ ἰβῖοῖα εὔβῆοδ “mail loops back to myself”;

Áõöü đăñéăñŬöăôăé ôôi sendmail FAQ üđùò öăβíăôăé đăñăêŬôù:

* Ðáßñiù ìçýíáôá ëÛèiðò "Local configuration error" üðùò ôi:

553 relay.domain.net config error: mail loops back to myself

554 <user@domain.net>... Local configuration error

Đùò ìđĩñþ íá ăđéěýòù ôĩ đñüâēcìá;

÷ ãð ãæððððáé íá éáððððýíððð ðí mail ðñíð ðí domain (ð.÷. domain.net)
ðñíð ðÜðíðéí ððððððñéíÝí íç ÷ Üíçíá (ððçí ðññðððððçð áððð, ðí
relay.domain.net) ÷ ñçððííðíðéíðððð íéá áðñáððð MX, áðÜ ðí íç ÷ Üíçíá
ðíð ðÜíáé ðçí áíáéáððýððððçð ááí áíááíñññðððð ðíí ááððü ðíð ùð
domain.net. ÐñíððéÝððð ðí domain.net ððí /etc/mail/local-host-names
(áí ÷ ñçððííðíðéíðððð ðí FEATURE(use_cw_file)) ð ðñíððéÝððð
"Cw domain.net" ððí /etc/mail/sendmail.cf.

Ç õñÝ÷÷ïõõå Ýêäïõç õïõ sendmail FAQ (<ftp://rtfm.mit.edu/pub/usenet/news.answers/mail/sendmail-faq>) ääí
 õõíõçñååõåäêÝíí íå ëÛåä Ýêäïõç õïõ sendmail. Ûõõõõí, äçííõåýåõåäé áíÛ åäëõÛ äéåõõõåõåõ õõõõ ëõõõåõ
 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), êåé news.answers (news:news.answers). Ìõññåõå
 åõõõõç íå ëÛåõåõ åíõõñååõí Ýíõõ email, õõÝêñííõåõ Ýíå íõíõåõ õõí <mail-server@rtfm.mit.edu> íå õçí åíõíêõ
send_usenet/news.answers/mail/sendmail-faq õõí êýñêí Ýíñíõ õïõ íçýíåõõõ.

18. Ἄεαὐβ̄ ᾱαί̄ οὐ̄ιδ̄ᾱνεὸ̄Ῡν̄ῑο̄ᾱε̄ οὐ̄ο̄ῶ̄῰̄ ἰ̄ε̄ ἄ̄ο̄ᾱν̄ῑᾱῩὀ̄ δ̄ε̄P̄ν̄ῑο̄δ̄ ἰ̄ε̄ῡ̄ῑç̄ὀ̄ ὁ̄ᾱ ἄ̄δ̄ν̄ῑᾱε̄ñ̄ο̄ί̄Ῡί̄ᾱ ἰ̄ç̄÷̄ᾱί̄P̄īāὀ̄ᾱ;

Ἄβριας δεεσίου οἱ ἀδναῖνοι ὧν ἱς ÷ Ὑίγια ἰά η̅οειβαεῖ οἱφ ὀδυῖ οἱο ὀα̅νιαδὲεῖρ ὀαὸ ὀά ε̅ῤοε ἀεῖοἱναδὲεῖ ἀδῦ οἱφ ὀδυῖ
cons25 ὀῖο ἀδῖαδὸαβῖα ἀδῦ ὀαῖ εῖῖοῖα οἱο FreeBSD.

ŌðŮñ ÷ iōī æēŮōīniē ōñūðīē æéá íá ðañæēŮīøåōå áōōū ōī ðñūæēcīá:

- [illegible]

ðnũāāñāñā. ÊÛēā ðāñÛēðñĩ ðĩð **screen** ðĩðĩðāñēðŸñāðāē ùð ðāññāðēēũ ðĩð VT100, Ÿðóē ç ãāðāāēçðP TERM ðĩðĩ ðññāēñðĩŸñĩ ðĩðĩēāēóðP ēā ðñŸðāē íā ñðēìēððāñ ðā vt100.

- ΆαεάααόδPόόα όçí έαόά÷þñçόç `cons25` όδç αÜόç ααάνYíñí όάνíαόέεþí όíό αδñíαέñόíYíñí όδñíαέόδP. Í όññóðíό έαé íá αβíáé αόóñ, αíάνóÜόάé αδñ όí έαέόíόñþαέéñ όYόόçíá όíό αδñíαέñόíYíñí όδñíαέόδP. ÖόóέñíαέéÜ, έá άñáßόά αόόYò όέó ðεçñíóíñßáó όόά άá÷άέñßαέά έαé÷άßñέόçð όόόóPíαόíò όíό αδñíαέñόíYíñí íç÷áíPíαόíò.
- Óóí όíðέéñ όáo FreeBSD íç÷Üíçíá, ðñçόέíñðíéPόόά όíí X server έáé éÜíòá login όóí αδñíαέñόíYíñ íç÷Üíçíá ðñçόέíñðíéþíόáó éÜðíéí άññíéñóP όάνíαόέéñY üðñò όí xterm P όí rxvt. Óόçí όάνßðóñç έόóP, έá ðñYðáé óóí αδñíαέñόíYíñ íç÷Üíçíá íá ðñèðíόáόά όçí íáόάáεçóP TERM όά xterm P vt100.

19. $\tilde{A}\acute{e}\acute{o}\tilde{\beta}$ $\acute{o}\tilde{i}$ $\imath\grave{c}\div\grave{U}\grave{i}\grave{c}\acute{i}\acute{a}$ $\imath\ddot{o}$ $\tilde{a}\tilde{a}\tilde{\beta}\div\acute{i}\acute{a}\acute{e}$ $\acute{o}\tilde{i}$ $\imath P\acute{i}\acute{o}\imath\acute{a}$ calcru: negative time...;

Ἀδού ἰδῆνᾱς ἰὰ ροῖαᾱς ἀδὺ ἀεῤῥοῖνᾱ ἀέοβᾱο διῶ ÷-ἀοβᾱῖῖῥῖῥῖᾱ ἰᾱ interrupts, οὐοῖ ροῖ ῥεέεῤῥ ὑοῖ ἑᾱε ροῖ ῖᾱῖῥῖῥῖῥῖῥῖ. Ἰδῆνᾱς ἰὰ ροᾱῖῥῖῥῖᾱ ὁᾱ δῆνᾱῖῥῖᾱῥῖᾱ (bugs) ἀεῤῥ ἰδῆνᾱς ἀδῖῥῖῥῖ ἰὰ δῆνῖῥῖῥῖᾱς ἰᾱῖῥῖῥῖᾱ ὁῥῖ ὕῥῖῥῖ ἑῤῥῖῥῖ ὁῥῖῥῖῥῖῥῖ. ἰᾱ ὁῥῖῥῖῥῖῥῖ ὀνῖῥῖ δῆνῖῥῖῥῖῥῖ ὁῥῖ δῆνᾱῖῥῖᾱῥῖᾱ, ἰᾱῖῥῖ ῥ ἰῥῖῥῖᾱῥῖ TCP/IP ἰᾱ ἰᾱῤῖῥῖ MTU ἰῤῖῥῖ ὁῥῖ δᾱῖῥῖῥῖῥῖ ἑῤῥᾱ. Ἰδῆνᾱς ἀδῖῥῖῥῖ ἰὰ δῆνῖῥῖῥῖᾱς ἀδὺ ἑῤῥῖῥῖῥῖ ἰᾱῖῥῖ ÷-ῥῖῥῖᾱ ἰᾱῖῥῖῥῖ, ἑᾱε ροῥῖ δᾱῖῥῖῥῖῥῖ ἰῥῖῥῖ ὁῥῖῥῖ δῆῤᾱῖᾱ διῶ ἑᾱ δῆῤῖᾱ ἰᾱ ἰῥῖᾱῖᾱ ἰᾱῖῥῖ ῥ ῆῤῖῥῖ interrupts ὁῥῖ ἰῥῖῥῖῖῖῖῖῖῖ ÷ ῥῖ ἑῤῥᾱ.

ḐānārŸnāāéá áóōŸŸ óŸō ḏñŕāēḐŕāóōŸ āŕŕāé ĩ áḑūōñŕō ḑāñŕāóéóŕō ḑēāñāāóēŕŕ ĩā ōŕ ĩḐŕōŕā “SIGXCPU exceeded cpu time limit”.

Áí ôî ðñüâëçìá äáí ìðĩñâß íá ëõëâß ìâ äéáöĩñâôëëü òñüðĩ, ç ëýóç åßíáé íá ìñßóâôâ ôçí ðãñáëÛôù ìâôââëçôß ôîõ sysctl:

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

ΌγιὰΒιούος: Ç äðëëĩāP –w oĩō sysctl(8) èàũñāBðáé ðāñũ÷çĩYĩç éáé áāĩĩāBðáé óéũðçēŬ äðũ oĩ FreeBSD 4.4-RELEASE éáé ĩāðŬ. ĩðĩñāBðā ĩā áóóŬëéáé ĩá oĩ ðāñāéēāBðāðā éáðŬ ôç ñýëìéóç ðũĩ äðëëĩāpĩ ĩā ôçĩ sysctl ððũð óāBĩāðáé ðāñāðŬĩũ.

Ὅτι δαῖναδ' Ὀὔτιν ἐὰ Ὑ ÷ ἁε ἀδβᾶναός οόγι ἀδουᾶρός, ἀεεῦ οᾶ ο ÷ Ὑός ιὰ όγι ἀέοβα οῖο δνῖαεπῖαδῖο, ιῦεεῖτ ἀὰρ ἐὰ οῖ
 δαῖναόγνπῶαδᾶ. Αἵ οῖ δνῦαεγῖα ἀδεῖ Ὑῖαε, ἀεάογνπῶαδ όγι όεῖπ οῖο systcl όδῖ Ὑῖα, ἐαε νῶεῖβῶαδ όγι ἀδεεῖαπ
 NTIMECOUNTER όοῖτ δδῶπῖα όαδ, οᾶ ιεῖ Ὑῖα ἀοῖαῖνῖαῖαδ όεῖ Ὑδ. Αἵ οδῦῶαδᾶ όγι όεῖπ NTIMECOUNTER=20 ἐαε οῖ
 δνῦαεγῖα ἀὰρ Ὑ ÷ ἁε εῶεᾶβ, όᾶ interrupts όδῖ ις ÷ Ὑῖαῖα όαδ ἄβῖαε δῖεῖ δνῖαεγῖαόεεῦ ἐαε ἀεᾶδῦεεγῖα ἀεᾶ ἀεῖαεᾶπ
 πῖεῖεός όδῶ βῆαδ.

20. Ἄεάδβ ς PnP eÜñôá iïò äáí áíé÷íäýâðáé ðēÝíí (P áíé÷íäýâðáé ùð unknown) ìäðÜ ôçí áíááÜèìéóç óå FreeBSD 4.X;

Οι FreeBSD 4.X ἀειῖροεᾶß ðēYīī ānēāðŬ ðēī ðéóðŬ òī ðñūòððī PnP éáé áðòū àçìéìõñāāß ïnéòìYíáð òīñYð òçí ðāñāīYñāāé íá ìç ēāéòìõñāīrī ēŬðīéāð òðóēāðYð PnP (ð.÷. ēŬñðāð P÷īō éáé áóòðāñéŬ modems) íé ìðīßāð ùòðòúì ēāéòìõñāīrYóáí òòī FreeBSD 3.x.

Ἰε εὐαῖρε ἀεά ὄγι ὀσιδᾶνεῖσιν Ὑ ἀδῶP, ἀῖκαῖρῖσάε ὀσι ἀεὐεῖτοῖ e-mail, ὀι ἰθῖβι ὀδῦῖεῖεᾶ ὀδς ἐβῶα freebsd-questions (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>) ἀδῦ ὀῖ Peter Wemm, ὕδ ἀδῦῖσῶς ὀᾶ ἠῖῖῶς ὀ-ᾶῖεῖ Ὑ ἰᾶ ᾶῶὀᾶῖεῖ modem ὀι ἰθῖβι ᾶᾶ Pῶᾶ ᾶ-ῖ-ῖᾶῶῖ ᾶδῦ ὀι ὀῶὀᾶ ἰᾶὀ ᾶδῦ ᾶᾶᾶῖεῖ ὀᾶ FreeBSD 4.X (ὀᾶ ὀ-ῖεᾶ ἰῶᾶ ὀᾶ ☐ ῖ-ῖῖ ὀῖ ὀῖὀᾶᾶ ᾶᾶ ῖᾶ ᾶῖᾶ ὀῖ ᾶὀᾶῖὀῖ ὀι ᾶὀᾶᾶῖᾶ ὀδς ὀᾶῖῶς).

ΌσιὰΒυόος: Ôĩ ðãñéã÷üìãĩĩ áõõÞò òçò ðãñÛèãóçò Ý÷ãé áíáíãùèãß óã ó÷Ýóç ìã ôĩ áñ÷ééü èãßìãĩĩ.

Ὅτι PNP bios οἱ δῆνι-νῦείεσθαι [οἱ modern] ἐὰν οἱ Ὑόροα οὐκ εἰδῇσι ÷ P ἀεὶ ἀδελφίσαντες οὐκ ἐνὸντι, ἐὰν Ὑόρε [οὐκ ἔστιν Ὑεἰρὸς 3. x] εἰ
 ἀεὶ ἀδελφίς οὐδὲν αἶψα ÷ ἰσθόος ISA οἱ "ἀνδρῶν" ἀεὶ ἀδ.

[illegible]

Ἄεά ίά έάέοιῶηάΠόάέ ίάίŨ ç οόόέάῶΠ, ḏñŸḑáέ ίά ḁñáεάβ οί PNP id όçð έάέ ίά ḑñίῶόάεάβ όçð έβόόά όυί άίέ÷ίάýόαυί
 ISA ḑίῶ ÷ñçόέῆῆḑίεῆýίόάέ έάά όçί άίάἱñḑḑέόç PnP όόόέάḑḑί. Ḑόόῶ ḑῆñἱάβ ίά ḁḑίáέ ἱά όç ÷ñḑόç όçð pnpinfo(8) έάά όçί
 άίβ÷íáόόç όçð όόόέάḑḑḑ, έάά ḑḁñŨάάέάἱ άόόḑ ḁḑίáέ ç Ÿἱἁῶό όçð pnpinfo(8) έάά Ḣίά άόόῶḁḁέέῶ 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)

[ðáñáëǽðđííôáé ié õðüëíéðǽò ãñàìÝò 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
```

Íé ðeçñĩõĩñBão ðĩõ áðáeõĩĩyíóáé, ânßöeĩĩóáé óôç ãñãĩP “Vendor ID”, óôçĩ ãñ÷P óôð ãĩũãĩõ. Ĩ ääeããĩääeëũ ãñeëĩũò óóeð ðãñãĩeYóáeð (óõĩ ðãñŮääeãĩã iáo 0x3024a341) ãßĩáé õĩ PnP id ãĩP õĩ æöãñeëĩççóeëũ ðĩõ ânßöeãðáé æẽeããðò ðñeĩ áðũ áðõũĩ ãßĩáé Yĩã ĩĩĩääeëũ ASCII áĩãĩũñeóðeëũ.

ΑἰάεεάεοέεÜ, αἰ öi pnpinfo(8) ääí ääß÷íáé öcí æçöiýíáíc êÜñöá, ìðĩñáßöâ íá ÷ñçöéiüðíéÞöáöâ öi pciconf(8).

Đa ñáêÜòù öäßíáôáé Ýíá ìÝñiò ôcò âñüäiñò ôcò pciconf -v1 ãéá Ýíá êýêëùìá Þ÷iñ ãíóùìáôùìÝñiñ óóc ìcññéêÞ:

```
# 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 ÷ñçóëiïðïéíýóáîâ ôçí ôëiþ ôiõ chip, “0x24158086”.

Ç ðëçñïöĩñßá áóôÞ (Vendor ID Þ ôëìÞ chip) èá ðñÝðáé íá ðñíóôâèåß óôi áñ÷åß /usr/src/sys/isa/sio.c.

[illegible]

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

Ýðáεóá íáðáεέίεíεáßðá ðññð óá εÛðð áεá íá áñáßðá ðí óðóðð ÝÛññð íá ðññðéÝóáðá ðεί εáðá÷ ðñεός ðεð ððóεáðßð óáð. Íé εáðá÷ ùñßóáεð óáßññðáε ùððð ðáñáεÛðð εáε áßíáε ðáίεíñçÝíáð εáðÛ ðí áεðáñέείçðóέεù ASCII Vendor ID ðí ðñßß εá ðñÝðáε íá ðáñέεççéáß óðí ó÷ ùεέí óðí ááίεù ÝÛññð ðεð áñáññðð íáεß íá ùεç ðεί ðáñέáñáðß *Device Description* (άί ÷ ùñÛáε, áέεεðð ÝÛññð ðεð) áðù ðεί Ýíñäí ðεð 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 */
```

ÐniøēÝóòā òī āāēāāīāāēēū Vendor ID āēā òç óóòēāòÐ óáo óòī óúóóū īÝñiò, āðīēçēāÝóòā òī āñ÷āBi, āīāāçīēīōñāÐóòā òīī ðññPíā óáo, ēāē āðāīāēēēīÐóòā. Èā ðñÝðāē òþñā ç óóòēāòÐ óáo íā āñāēāBi ùò óóòēāòÐ sio uðùò óòīÝāāēīā ēāē íā òī FreeBSD 3.X

21. $\tilde{A}éáô\beta$ $\delta\alpha\beta\eta\iota\upsilon$ $\tilde{o}\tilde{i}$ $\tilde{e}\tilde{U}\tilde{e}\tilde{i}\tilde{o}$ nlist failed $\tilde{u}\tilde{o}\tilde{a}\tilde{i}$ $\tilde{a}\tilde{e}\tilde{o}\tilde{a}\tilde{\epsilon}\tilde{\rho}$, $\tilde{a}\tilde{e}\tilde{a}$ $\tilde{\delta}\tilde{a}\tilde{n}\tilde{U}\tilde{a}\tilde{a}\tilde{e}\tilde{a}\tilde{i}\tilde{a}$, $\tilde{o}\tilde{i}$ top P $\tilde{o}\tilde{i}$ systat;

Ôi ðñuâëçia åbíáë uðë ç åðãñiãð ðið ðñiððáëåðå ïá åëðåëÝóðåð øÜ÷-íáë áéá Ýía óðååñëêiÝíi óýiãñií óóíi ðññPía,
 áëëÜ áéá êÜðñií euaíi áãí iðññãð íá ði áíðíððåé. Ôi óóÜëiã áððü iðññãð íá iðåðåððáë óå áýi ðñiãëPiaóá:

- [illegible]

22. Ἄεαὸβ δαβῆῖαέ ουοί ÷ñüüí íá οῖῖααèþ íà οῖῖ ὀδῖεῖαέοόP ιῖο ιΎού ssh P telnet;

Οἱ γυῖδοῦναι: Ὁδῦν-ἄε ἰαᾶῦεç ἑᾰέοόδῖνçόç ἰᾰόᾰίγ ὁçð ὀόέᾰἰðð ðῖο ᾰðῖᾰέᾰβόόᾰᾰᾰᾰᾰ ç TCP ὀγῖᾰᾰόç ἑᾰᾰ ὁçð ὀόέᾰἰðð ðῖο οἱ ðνῖᾰᾰᾰἰᾰ ὀόç ἰᾰᾰᾰῦ ὀῖο ðᾰᾰῦðç æçðῦᾰᾰᾰ ὀῖῖ ἑῦᾰᾰᾰῦ ðñῖᾰᾰᾰᾰᾰᾰᾰ (ð ὀççῖ ðᾰñḂðòḂḂḂ ὀῖο telnet(1), ὁçð ὀόέᾰἰðð ðῖο ἰᾰᾰᾰἰᾰᾰᾰᾰᾰ ç ðñῖᾰḂḂḂ login).

Οι δηναεεγια: Οι δετ δεεαφυ αβλαε υεε ε εαεεοοδΥηηερε ιοαβεαοαε οοει οηιοοδΥεαεα οιο εαοααΥεεαε οι ειαεοιεευ οοε ιαηεΥ οι οιο αιηοδχηαοοεβ ια αηαε οι υηηα οιο ιε÷αηβιαοιο - οαεΥοε αδυ οει IP αεαγεοιφε οιο. Ιε οαηεοοουοαηε αιηοδχηαοοεδΥο, οοιοαηεεαηααηΥιυ οιο Telnet εαε SSH οιο Υη÷ιηοαε ια οι FreeBSD, εαεοιοηαηιρ ια αοου οηι οηυοι, ιοοα ιαοαηΥ Υεευι, ια αδιεεεαγιοιοι οι υηηα οιο ιε÷αηβιαοιο οα Υια αη÷αβι εαοααηαοδ εαε ιαεεηιφεεβ αηαοηΥ αδυ οηι αεα÷αεηεοεβ.

Ç eañaðãßa: Áí ôi ðñuãeçia ðñieýðoáe eÛeã oĩñÛ ðño óoĩaÝáoòã áðñ ôĩi ððieĩaeóòp óáo (ôĩi ðãeÛôç) óã ãðieĩaßðieòã áĩððçñãòçòp, ôi ðñuãeçia ãñßoéãoáe óoĩi ðãeÛôç. Ìã ôĩi ßãeĩ ðñuðĩ, áí ôi ðñuãeçia óoĩããßãe ìuĩ uóái eÛðieĩò óoĩaÝáoáe óoĩi ððieĩaeóòp óáo (ôĩi áĩððçñãòçòp), ôi ðñuãeçia ãñßoéãoáe óoĩi áĩððçñãòçòp.

Άν οι δνυαεγία άβρίαε όοίρ δάεΥόε, ς ιυίε εαηάδαβία άβρίαε ίά αείηεηοάοά οι DNS, ηόόά ί άιόδϋηάοεοηό ίά ίοηηάβ ίά οι άηάε. Άν οι δνυαεγία άιόάίβαεάοάε όοί όιθεέυ όάο άβεόοι, εαηηάβόόά οι δνυαεγία όοίρ άιόδϋηάοεοηό εάε όοίά÷βόόά όϋί άρΥαηόε. Άίοβεάόά, άν οι δνυαεγία άιόάίβαεάοάε όά όοίάΥόάέο ίΥόυ Internet, εάόΥ δΥόά δέεάφύόϋά εά ÷ηάεάόόάβ ίά άδéeieíυηόάόά ίά όίί ISP όάο εάε ίά αϋόηόάόά ίά όάο οι αείηεηοάε.

Αί οι δννυαεχία αβρίαε ια οι αιτοεχναοεορ, εαε αιαιβαεαοαε οοι οιθεευ οαο αβεοοι, εα δνΥθαε ια οι ποειβοαοα ιοοα ια ιδιναβ ια αεοαεαβ αιαιεοροαεο ογθω αεαγερτοε οα υμνα, αεα οει οιθεερ θαειε÷ρ αεαοεγτοαυ οαο. Ααβοα οεο οαεβααο manual ουι hosts(5) εαε named(8) αεα θαειοοουοαηαο δεεηνωιηβαο. Αί οι δννυαεχία αιαιβαεαοαε οοεο οοιαΥοαεο ιΥου Internet, ιδιναβ ια τοαβεαοαε οα εαερ εαεοιτοηαβ οι resolver οοι αιτοεχναοεορ οαο. Αεα ια οι αεΥαιαοα, αιειεΥοοα ια αηαβοα εΥθιει Υεει ιε÷Υιεια, αεα θαηΥααεαια οι www.yahoo.com. Αί ιγοα αοου αιεαγαιε, αεαβ αηβοεαοαε οι δννυαεχία οαο.

[illegible]

23. $\mathbb{D}i\acute{e}a\ \acute{a}b\acute{i}a\acute{e}\ \zeta\ \acute{Y}\acute{i}\acute{i}\acute{e}a\ \acute{o}\acute{i}\acute{o}$ stray (ðǎñéðëáípìáíiö) IRQ;

Óá stray IRQs áßíáé ôçíÛáéá ðññäëçíÛòuí öëëëý ðíð ÷ñçóëíðíéáß IRQs, áéáíëúòáñá ó÷áòßæáòáé íà öëëëú ðíð éáòÛ ôç íÝóç ðíð éýëëíð áíááíðñéóçð (acknowledge cycle) ðíð interrupt, óðáíáðÛáé íá íàðááßááé ôçí áíóßóðíé÷ç áßòçóç áéáêíðòð.

÷:âôâ ôñâéò ãðééïãÝò ãéá íá áíôéïãòùðßóãôã áóôù ôï ðñüâéçïá:

- $\Delta\iota\alpha\div\epsilon\alpha\beta\theta\alpha$ $\theta\epsilon\delta\theta$ $\delta\eta\eta\alpha\epsilon\alpha\iota\theta\iota\epsilon\theta\alpha\epsilon\delta$, $\theta\acute{o}\epsilon$ $\epsilon\alpha\epsilon$ $\alpha\epsilon\epsilon\epsilon\theta$, $\iota\alpha\theta\tilde{\cup}$ $\theta\epsilon\delta$ 5 $\delta\eta\theta\theta\alpha\delta$, $\alpha\alpha\iota$ $\epsilon\alpha$ $\alpha\alpha\beta\theta\alpha$ $\tilde{\cup}\epsilon\epsilon\alpha\delta$.
- $\acute{\omicron}\delta\alpha\iota\alpha\delta\theta\theta\alpha$ $\alpha\iota\theta\alpha\epsilon\theta$ $\theta\epsilon\delta\theta$ $\delta\eta\eta\alpha\epsilon\alpha\iota\theta\iota\epsilon\theta\alpha\epsilon\delta$, $\alpha\epsilon\epsilon\tilde{\cup}\alpha\iota\theta\alpha\delta$ $\theta\iota$ 5 $\theta\alpha$ 0 $\theta\acute{o}\zeta\iota$ `isa_strayintr()`.
- $\acute{\omicron}\delta\alpha\iota\alpha\delta\theta\theta\alpha$ $\theta\epsilon\delta\theta$ $\delta\eta\eta\alpha\epsilon\alpha\iota\theta\iota\epsilon\theta\alpha\epsilon\delta$ $\alpha\alpha\epsilon\alpha\epsilon\epsilon\theta\theta\iota\theta\alpha\delta$ $\theta\epsilon\epsilon\epsilon\tilde{\cup}$ $\alpha\epsilon\alpha$ $\theta\zeta\iota$ $\delta\alpha\eta\tilde{\cup}\epsilon\epsilon\zeta\epsilon\zeta$ $\delta\eta\theta\theta\alpha$ $\delta\iota\theta$ $\iota\alpha$ $\div\eta\zeta\theta\epsilon\iota\theta\iota\epsilon\alpha\beta$ $\theta\iota$ IRQ 7 $\epsilon\alpha\epsilon$ $\theta\iota$ $\alpha\iota\theta\theta\theta\theta\iota\epsilon\div\iota$ $\alpha\epsilon\alpha$ $\alpha\theta\theta\tilde{\cup}$ $\delta\eta\eta\alpha\eta\alpha\iota\iota$ $\iota\alpha\beta\alpha\zeta\theta\zeta$ PPP ($\alpha\theta\theta\tilde{\cup}$ $\theta\theta\iota\alpha\beta\iota\alpha\epsilon$ $\theta\theta\alpha$ $\delta\eta\alpha\epsilon\theta\theta\theta\theta\alpha\eta\alpha$ $\theta\theta\theta\theta\theta\iota\alpha\theta\alpha$) $\epsilon\alpha\epsilon$ $\alpha\alpha\epsilon\alpha\theta\alpha\theta\theta\theta\alpha$ $\tilde{\Upsilon}\iota\alpha$ $\iota\alpha\zeta\tilde{\cup}$ IDE θ $\tilde{\cup}\epsilon\epsilon\iota$ $\theta\epsilon\epsilon\epsilon\tilde{\cup}$ $\delta\iota\theta$ $\iota\alpha$ $\div\eta\zeta\theta\epsilon\iota\theta\iota\epsilon\alpha\beta$ $\theta\iota$ irq 15 $\iota\alpha\epsilon\beta$ $\iota\alpha$ $\theta\iota$ $\epsilon\alpha\theta\tilde{\cup}\epsilon\epsilon\zeta\epsilon\zeta$ $\delta\eta\eta\alpha\eta\alpha\iota\iota$ $\iota\alpha\beta\alpha\zeta\theta\zeta$ $\theta\iota\theta$.

24. Æéáôß æëÝðù óõíÝ÷æéá ôî ìÞíõíá file: table is full óõî dmesg;

Όι ιΡιοιά άόοι οçιαβίάε υόε Υ÷άοά άίάιόεΡόάε οίι άνέειι οίι άεάεΥόέιι άñέάñάόΥύι άñ÷άβιι (file descriptors) οόι όύόόçιά οάό. Δάñáέάειγιά άάβόά οί kern.maxfiles (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html#KERN-MAXFILES) οίΡιά οόι έάόΥέάέι Νýέιέόç Ιñβιι ΔόñΡιά

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html) οἷο
 Ἀῖ÷ἁἡἡἡἡ, ἁἡ ἁἡἡἡἡ ἡἡ ἁἡἡἡἡ οἷο ἁἡἡἡἡἡἡ.

25. Ἀἡἡἡ οἷο ἡἡἡἡ οἷο οἡἡἡἡ ἡἡ ὁἡἡἡἡἡἡ ἁἡ ἡἡἡἡἡ ὁἡ ὁἡἡἡ ἡἡ:

Ἠ οἡἡἡἡ ὁἡἡἡἡἡἡ ὁἡ Ὑ÷ἡ ἡἡ ἡ ἡἡἡἡἡἡἡ ἡἡἡἡἡ, ἡἡ οἷο FreeBSD Ὑ÷ἡ ἡἡἡἡἡ ἡ ἡἡἡἡἡἡἡἡ οἷο
 ἡἡἡἡ.

Ἀἡἡἡἡἡ ὁἡ dmesg(8), ἡἡ ἡ Ὑἡἡἡ ἡἡ ἡἡἡἡἡ ὁἡ ἡἡἡἡἡἡἡ Ὑ÷ἡ ὁἡ ἡἡἡἡἡἡἡἡ. Ἠ ὁἡἡἡἡἡ ἡἡ ὁἡ
 ἡἡἡἡἡ ὁἡ ἡἡ ὁἡἡἡἡἡ ἡἡἡἡἡ ὁἡ ἡἡἡἡ ὁἡ ἡἡἡἡ ὁἡ ἡἡἡἡἡἡἡ ἡἡ ἡἡἡἡ ὁἡ TSC.

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

Ἠἡἡἡἡ ἡ ὁἡἡἡἡἡἡἡ ἡἡἡ, ἡἡἡἡἡἡἡ ὁἡ ὁἡἡ οἷο kern.timecounter.hardware sysctl(3).

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

Οἷο BIOS ὁἡ ὁἡἡἡἡἡἡ ὁἡ ὁἡἡ οἷο ἡἡἡἡἡἡἡ TSC— ἡἡἡἡἡἡ ἡἡ ἡἡ ἡἡἡἡἡ ὁἡ ὁἡἡἡἡἡ οἷο ἡἡἡἡἡἡἡἡ
 ἡἡἡ ἡἡἡἡἡἡἡ ἡ ἡἡἡἡἡἡ, ἡ ἡἡἡ ἡἡἡἡἡ ὁἡ ἡἡἡἡἡἡἡἡ ἡἡἡἡἡ ἡἡἡἡἡἡἡἡ, ἡἡἡ οἷο FreeBSD ἡἡ
 ἡἡἡἡἡἡ ἡἡ ἡἡἡἡ ὁἡ ἡἡἡἡἡἡ ἡἡ ὁἡἡἡἡἡ ἡ ἡἡἡἡἡἡ ἡ ἡἡ Ὑἡἡἡἡἡἡἡ.

Οἷο ἡἡἡἡἡἡ ἡἡ, ἡἡἡ ἡἡἡἡἡ ἡἡἡἡἡἡ οἷο ἡἡἡ i8254 ἡἡ ἡἡἡἡἡ ἡ ὁἡ ἡἡἡἡἡἡ ἡἡἡἡἡἡ οἷο ἡἡἡ οἷο
 sysctl(3) kern.timecounter.hardware.

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

Ἠ οἡἡἡἡ ὁἡἡἡἡἡἡ ὁἡ ἡἡ ἡἡἡἡ ὁἡ ἡἡἡ ὁἡ ἡἡἡἡἡ ὁἡἡ ὁἡἡἡἡ ὁἡ ὁἡἡἡἡ.

Ἀἡ ἡ ἡἡἡἡἡἡ ἡ ἡἡἡἡ ἡἡἡ ὁἡ ἡἡἡ ἡἡἡἡἡἡ, ἡἡἡἡἡἡ ὁἡ ἡἡἡἡἡἡ ἡἡἡἡἡἡ /etc/sysctl.conf.

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

26. Ἀἡἡἡ ἡ οἡἡἡἡ ἡἡ ὁἡἡἡἡἡἡ ἡἡ ἡἡἡἡἡἡἡ ὁἡἡἡ ὁἡἡἡἡ ὁἡἡἡ PC card;

Οἷο ἡἡἡἡἡ ἡἡἡ ἡἡἡ ὁἡ οἡἡἡἡ ὁἡ ἡἡἡἡἡἡ ἡἡἡἡἡἡἡ ἡἡ Ὑἡἡ ἡἡἡἡἡἡἡἡ ὁἡἡἡἡἡἡ. Ἠἡἡἡἡ ἡἡ ἡἡ-BSD
 ἡἡἡἡἡἡἡ ὁἡἡἡἡἡἡ ἡἡἡἡ ὁἡ ἡἡἡἡἡἡἡἡ ἡἡἡἡἡἡἡ. Ἠ ἡἡἡἡἡ pccardd ὁἡ ἡἡἡ ὁἡ
 ἡἡἡἡἡἡἡ, ἡἡἡἡἡἡ ὁἡ ἡἡἡἡ ἡἡ "(null)" "(null)" ἡἡἡ ἡἡ ὁἡἡἡἡἡἡἡ ὁἡ ἡἡἡἡἡ.

Ἠἡἡἡ ἡ ἡἡἡἡἡἡἡ ἡἡἡἡἡ ὁἡ ὁἡἡἡἡἡἡ ἡἡ ὁἡ ἡἡἡ PC card ἡἡἡ ὁἡ ὁἡἡἡ ἡ ἡἡἡἡἡἡ ὁἡἡ ἡἡἡἡ ὁἡ
 ἡἡἡἡἡἡἡ. Ἀἡἡἡἡἡἡἡἡἡ ὁἡἡἡἡ ὁἡἡἡἡἡἡ ὁἡἡ. (Ἠἡ ὁἡ ἡἡἡἡ ὁἡ ἡἡἡἡἡἡἡ ἡἡἡἡἡ ἡ Ὑἡἡἡ, ἡἡ
 ὁἡἡἡ ἡ ἡἡἡἡἡἡἡἡἡ ἡἡἡἡἡ.) Ἠἡἡἡἡἡἡ ἡἡ ἡἡἡ ἡἡἡἡ ἡἡ ἡἡἡἡἡἡἡἡἡ. Ἠἡ ὁἡἡἡἡ ὁἡἡ ἡἡ PC card ἡ
 ἡἡἡἡἡἡἡ ἡἡἡἡἡ.

Οἷο ὁἡἡἡ ἡἡἡἡἡ οἡἡἡἡ ὁἡἡἡἡἡἡ ὁἡἡ ἡἡἡἡἡἡἡἡἡ ἡἡἡἡἡἡ ἡἡἡἡἡἡἡ, ἡἡἡἡ ἡἡ ἡἡἡ ὁἡἡἡἡἡἡἡ ἡἡ ἡ
 ὁἡἡἡἡἡἡ ἡἡἡ ἡἡἡἡἡἡ. Ἀἡ ὁἡ ἡἡἡἡἡἡ ἡἡ Ὑἡἡ ὁἡ ἡἡἡἡἡἡ ἡἡἡἡἡἡἡ, ὁἡἡἡἡἡἡ ὁἡ ἡἡἡἡἡἡἡ ὁἡ
 ὁἡἡἡἡἡἡ ὁἡἡ, ἡἡἡἡἡἡ ὁἡ ἡἡἡἡἡἡ, ἡἡἡἡἡἡἡ ἡἡἡ, ὁἡἡἡἡἡἡ ἡἡἡ ὁἡ ἡἡἡἡἡἡ ἡἡ ἡἡἡἡἡἡἡἡἡ.

27. Ἄεάδθ ι öññòùòÐð ãêêßícýóçð ðíð FreeBSD äãß÷íáé ðí ìßíðíá ëÛëïðð Read error êáé óóáíáðÛáé ìáðÛ ðçí ðèùíç ðíð BIOS;

[illegible]

Iðirnaðoá íá añaðoá ðeo úouóÝò ðeiÝò áeá ôçí àauiaðñBá oïr tãçãý oði BIOS oïr iç÷áiPiáoìð. ØÜîa áeá oïí añèèù
ðuí èðëßíanuí, êäöáepí êáé õñÝúí áeá oïí tãçäu ðio èÝêäð.

İYóá áđũ ôçí fdisk ôĩõ sysinstall(8), đéYóôâ ôĩ G ãéá íá ĩñBoâôâ ôçí ãâũĩãõñBá ôĩõ ĩãçĩĩý.

Éá àìòáíéóóàß Ýíáð æÛëïãðð ðïð éá æçðÛáé òíí áñéèìù òùí èðëßíáñùí, èáðáéþí éáé òñÝùí. Ðëçèðñíëïãßóðá òïðð áñéèìýðð ðïð áñÞéáðá áðù òí BIOS, ÷ññßæííóáð òïðð ìá éáñííééÝð éáéÝòïðð. Áéá ðánÛááéàìá, áéá 5000 èðëßíáñðð, 250 èáðáéÝð éáé 60 òññáßð, éá ãñÛðáìá **5000/250/60**.

ΔéÝòâ enter ãéá íá ïñßóâòâ ôéò ôéíÝò, éáé Ýðãéóá ôí W ãéá íá ãñÛøâòâ ôí íÝí ðßíáéá éáôâòíÞóâuí óóíí ïäçü.

28. ἰά Ὑεῖ ἑαέοἰοῖῃῃῃῃ ὀύοοçἰά ἑαὐὶόοῃῃῃ ὀῖῖ ἁέα÷ἁέῃῃῃῃῃ ἁέῃῃῃῃῃ ἰῖῃ. Δὐὸ ἰῃῃῃ ἰά ὀῖῖ ἁῃῃἑαὐὶόοῃῃῃ;

[illegible]

29. $\hat{O} \in \mathcal{O}(n^2)$ swap_pager: indefinite wait buffer::

Όχιαβλαί υόε ιέα αέααεαόβα θπιοθάαβ ία ανΰθαι ιέα οαεβαά ιπΐχδ οοι αβοεί, έαε ς αδυθαείνα αοοβ Ύ÷ έε έρεεΡοάε θπιοθαεπΐοαδ ία αδιεθΡοάε θπυοααοϋ οοι αβοεί αέα θανεοουοαπΐ αδυ 20 ααοοανυεαθθα. Αοδυ ιθΐπαβ ία οοιααβ αδυ ÷ αεαοΐΎπιοδ θπαβδ οοι οέεϋνυ αβοεί, θπιαεϋαοέεΰ έαεπαέα, Ρ ΰεετ οέεεϋ οι ιθΐβι ία ο÷αοβ:αοάε ια Ι/Ο. Αί θπυεαέοάε αέα θπιαεϋαοέεϋ αβοεί, έα ααβθα άθβοϋδ έαε αίθβοοίε÷α ιϋΐΎαοά οοι /var/log/messages έαε οόϋΐ Ύπιαΐ οϋδ αίορεθδ dmesg. Αεαοιπιαοέεΰ, αεΎαίθα οεδ οοΐαΎοαεδ έαε οα έαεπαέα οαδ.

30. Ôé âßíáé ôá óóÜëíáôá “UDMA ICRC”, êáé ðùò íðñþ íá ôá äéíñêþù;

[illegible]

Οι δηναεχία ιδινάβ ιά δνιεεεαβ άδν δνιεεγδ δάνΰανόαδ, άί έαέ ι δει όοιςεεοιΰνιό άβιάέ ς δνιαιεχιαόεεP P έαίεαοιΰις έαεναβνός. ΆεYαιόα υόε όα έαεπαέα ΑΟΑ άάι Y÷ιόι όδιόόάβ æçieÛ, έαέ υόε άβιάέ έαόÛεεçειρ δνιαιέαανόπρ έέα όçí έαόÛόόάός έαέοιόναβάδ Ultra DMA δτι ÷ñçóeiιδιείαβόα. Άί ÷ñçóeiιδιείαβόα άόαενίγίαιά όόñόÛνέα άβόει, έα δñYδαι άδβόçδ ιά άβιάέ όοιαάόÛ. Άααέυεαβόα υόε όδÛñ÷αι έαεP άδάδP όα υεάδ όεδ όοíáYάέδ, ÷ιόι άδβόçδ άίάόανεαβ δνιαιεPιαόα υόάí άβιαόαι άαέαόÛόόάός άíυδ δαέειγ ίαçáγύ όοι βαέí έάρÛεé DMA ιά Yía άβόει Ultra DMA 66 (P δει ανPαννι). ΌYειό, όα εÛεç άόόÛ ιδινάβ ιά όçíαβñιό υόε ι άβόειό δνυεαέόαι όγíοñά ιά ÷άεÛάέ. Ιέ δανέοóυόανιέ έαόάόέαόάόYδ άβόειρ δανY÷ιόι εíαιέοιέευ άεYá÷ιό έέα όιόδ ίαçáγύδ όιόδ, άεYαιόα εíεδñι όι άβόει όάδ, έαέ άί ÷ñáεÛάάόαι, δÛñόα άíόδβανάοι όνι άανñYíυι όάδ έαέ άíόέέαόάόόPόόά όνι.

[illegible]

Áí óõíòñßá, áðõíý òïð åßäïðð ïé ðññáéäïðéþóáéð åáí åßíáé óõíþèùð ïïéñáßåð, áëëÛ òðù ïñéóìÝíåð áðð÷åßð ðññùðñèÝóáéð, ïðññåß íá ðññéáéÝóìïí áíåðéèýìçðåä öáéíìííá íåä ïðñíåä èòìáßñííóáé áðù ïéá óðéäìéåáßåä ððþóç óðçíí áðùèñéóç òïð óðóðþíåð, ïÝ÷ñé ðèþñçð éåðÛññåðóç.

33. Åéåðß ç äéåäééåóßåä buildworld/installworld óðåíåðÛåé ïå òï ïþñíåä touch: not found;

Ôï ïþñíåä áððù åáí óçìáßíåä ùðé óåð èåßðåé òï äïçèçðéèù ðññåññåííåä touch(1). Ôï èÛèðð áððù ðññéåäåßðóáé óõíþèèùð áðù éåíéåóìÝíç, ïäèèíðéèèþ, óþíåíçç çìññçííåð òùí åñ÷åßåßí. Áí òï ïñèèé CMOS òïð ððññéåóðð óåð åßíåä ïðèèéóìÝíí åéåä òïðèèþ þñåä, ðñÝðåé íåä äèðåéÝóåðåä óçí åíðèèþ adjkerntz -i åéå íåä ïð ïðèèéåðåä òï ïñèèé òïð ððñíåä ùðåí åèèéåßðåä óå éåðÛðóåç èåèèèðñåðåä åíùð ÷ñþóç.

ΕὰöÛëáéĩ 6 ÅìðĩñëéÛò ÅöáñĩĩãÛò

Όçĩåßùόç: 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.

Όçĩåßùόç: 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/>)

ÊäöÜëáéï 7 ÄöáñïïãÝò Ôäëëëïÿ × 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 24,000 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 is 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 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 çäêðññíëêÞ ëßóôá ôñí 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.

Όχι!Βούτ: 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.

ΕὰöÜëáéï 8 Ñýèìéόç ÐõñΠία

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).

ΌçïâΒùόç: 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 ¶ 7.

ΕὰöÛëáéĩ 9 ÄΒóëïé, ÓõóôΠιάôá Äñ÷åΒùí êáé ÖïñôùôÝò ÄêêΒίçóçò

1. 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.

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

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

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

For example, if you are going to move root to `/dev/ad1s1a`, with `/mnt` as the temporary mount point, it is:

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

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

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

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

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

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

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

The installation procedure allows you to 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 scrollbar 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 install a new master boot record as well, thus clobbering the BSD bootstrap.

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

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

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

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

If a system should crash after the kernel accepts a chunk of data for writing to disk, but before that data is actually written out, data could be lost or corrupted. This risk is extremely small, but generally manageable. Use of IDE write caching greatly increases this risk; it is strongly recommended that you disable IDE write caching when using Soft Updates.

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

Vital information on the root partition changes very rarely. Files such as `/kernel` and the contents of `/etc` only change during system maintenance, or when users change their passwords. If the system crashed during the thirty-second window after such a change is made, it is possible that data could be lost. This risk is negligible for most applications, but you should be aware that it exists. If your system cannot tolerate this much risk, do not use Soft Updates on the root filesystem!

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

4. What is inappropriate about my ccd?

The symptom of this is:

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

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

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

The symptom of this is:

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

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

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

6. Can I mount other foreign filesystems under FreeBSD?

FreeBSD supports a variety of other filesystems.

Digital UNIX

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

Linux

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

Windows NT®

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

FAT

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

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)`).

7. How do I mount a secondary DOS partition?

The secondary DOS partitions are found after ALL the primary partitions. For example, if you have an “E” partition as the second DOS partition on the second SCSI drive, you need to create the special files for “slice 5” in `/dev`, then `mount /dev/dals5:`

```
# cd /dev
# sh MAKEDEV dals5
# mount -t msdosfs /dev/dals5 /dos/e
```

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

8. 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.

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

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

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

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

`/boot/boot0` needs to be installed using `sysinstall` 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

11. How do I boot FreeBSD and Linux using 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.

12. 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).

13. 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.

14. 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
```

15. 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).

16. 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.

17. 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).

18. 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).

19. 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.

20. 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.

21. 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.

22. 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/mttools` package in the ports collection.

ΕὰöÛëáéï 10 Äéá÷ǎßñéόç 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

```
ttymq none network
```

The order of the letter designations is `ttym[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
```

Όχι! Βùόç: 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 than 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 `vnlr`?

`vnlr` 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).

Εἰσαγωγή 11 Ὁ Ὄγος τῶν X Windows εἰς τὸ Ἀέθῆρ τῶν Εἰσόδων

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 7.5.2, 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) (<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](http://www.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.

Ὁ γὰρ βούλος: 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`.

Δημιουργία: 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.

ὉΐαΒύος: 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 Α: 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 § 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`.

Εἰσαγωγή 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
```

Εἰσαγωγή 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`.

Ὁρισμός: 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:

```
ttvv9  "/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`, `xm` should be started without any arguments (i.e., as a daemon). `xm` must start AFTER `getty` runs, or else `getty` and `xm` will conflict, locking out the console. The best way around this is to have the script sleep 10 seconds or so then launch `xm`.

If you are to start `xm` from `/etc/ttys`, there still is a chance of conflict between `xm` and `getty(8)`. One way to avoid this is to add the `vt` number in the `/usr/local/lib/X11/xm/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 `vt`y from one, whereas the FreeBSD kernel numbers the `vt`y 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 ($\leq 3.3.6$) 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 `xm` or another display manager (e.g., `kdm`); the second is to use the `Xwrapper`.

`xm` 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 `xm` 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.

ΕὰöÛëáéï 12 Äéêôýùόç

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 0xfffff00
```

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 <gfooster@driver.nsta.org>, with some more modern additions:

Ενότητα 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.

Chapter 13 Sandboxing

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 breach 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.

Δηλαδή: `Securelevel` is not a silver bullet; it has many known deficiencies. More often than not, it provides a false sense of security.

One of its biggest problems is that in order for it to be at all effective, all files used in the boot process up until the `securelevel` is set must be protected. If an attacker can get the system to execute their code prior to the `securelevel` being set (which happens quite late in the boot process since some things the system must do at start-up cannot be done at an elevated `securelevel`), its protections are invalidated. While this task of protecting all files used in the boot process is not technically impossible, if it is achieved, system maintenance will become a nightmare since one would have to take the system down, at least to single-user mode, to modify a configuration file.

This point and others are often discussed on the mailing lists, particularly the `FreeBSD` `security` list (<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`.

Chapter 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 9.
- If the problem is local, read on entry 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
```

Warning: It usually does no harm if both sides initiate negotiation, so openmode is now active by default. However, the next section explains when it *does* do some harm.

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

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

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

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

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

```
set openmode passive
```

This tells ppp(8) to wait for the server to initiate LCP negotiations. Some servers however may never initiate negotiations. If this is the case, you can do something like:

```
set openmode active 3
```

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

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

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

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

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

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

```
set openmode passive
```

command. Care should be taken with this option. You should also use the

```
set stopped N
```

command to limit the amount of time that ppp(8) waits for the peer to begin negotiations. Alternatively, the

```
set openmode active N
```

command (where *N* is the number of seconds to wait before starting negotiations) can be used. Check the manual page for details.

14. Why does ppp(8) lock up 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 `Frame Check Sequence`. Each PPP packet has a checksum attached to ensure that the data being received is the data being sent. If the FCS of an incoming packet is incorrect, the packet is dropped and the HDLC FCS count is increased. The HDLC error values can be displayed using the `show hdlc` command.

If 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 `freebsd-questions` mailing list at 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.

Chapter 15 Serial Communications

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 `ttidx` (or `cuaax`) device is the regular device you will want to open for your applications. When a process opens the device, it will have a default set of terminal I/O settings. You can see these settings with the command

```
# stty -a -f /dev/ttyd1
```

When you change the settings to this device, the settings are in effect until the device is closed. When it is reopened, it goes back to the default set. To make changes to the default set, you can open and adjust the settings of the “initial state” device. For example, to turn on CLOCAL mode, 8 bits, and XON/XOFF flow control by default for `ttyd5`, do:

```
# stty -f /dev/ttyid5 clocal cs8 ixon ixoff
```

A good place to do this is in `/etc/rc.serial`. Now, an application will have these settings by default when it opens `ttyd5`. It can still change these settings to its liking, though.

You can also prevent certain settings from being changed by an application by making adjustments to the “lock state” device. For example, to lock the speed of `ttyd5` to 57600 bps, do

```
# stty -f /dev/ttyld5 57600
```

Now, an application that opens `ttyd5` and tries to change the speed of the port will be stuck with 57600 bps.

Naturally, you should make the initial state and lock state devices writable only by `root`. The `MAKEDEV(8)` script does *NOT* do this when it creates the device entries.

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 `ttidx` 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 -speed
```

with `line` being the serial port (e.g. `/dev/cuaa0`) and `speed` being the speed (e.g. `57600`). When you are done entering the AT commands hit `~.` to exit.

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.

Ερώτημα 16 Αιτιολογία Άντη

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.

Προσοχή: 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.

ÔçĩÄßüóç: 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/pccemu` 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

ÊäöÛëáéï 17 × éïýìñ 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 cvsupperd 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 ‘I 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.

ÊäöÛëáéï 18 Đñĩ÷-ùñçìÝíá ÈÝíáôá

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 are 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. It 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 2^{64}) + 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 çäâëññíéê Ëßóóá ôâ÷ íéêþí óðæçôþóâúí ôïð 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.

ÓçîâBuóç: 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(bttext)
SEARCH_DIR(/usr/lib); SEARCH_DIR(/usr/obj/elf/home/src/tmp/usr/i386-unknown-freebsdelf/lib);
SECTIONS
{
    /* Read-only sections, merged into text segment: */
    . = 0xc0100000 + SIZEOF_HEADERS;
    .interp      : { *(.interp)      }
```

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.*]

ÊåöÛëáéï 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 & Brian Tiemann, Sams, 1st edition, 992 pages, October 2001, ISBN 0-67232-206-4.
- 4.4BSD System Manager's Manual*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 804 pages, ISBN 1-56592-080-5.
- 4.4BSD User's Reference Manual*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 905 pages, ISBN 1-56592-075-9.
- 4.4BSD User's Supplementary Documents*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 712 pages, ISBN 1-56592-076-7.
- 4.4BSD Programmer's Reference Manual*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 866 pages, ISBN 1-56592-078-3.
- 4.4BSD Programmer's Supplementary Documents*, Computer Systems Research Group, University of California, Berkeley, O'Reilly and Associates, 1st edition, June 1994, 596 pages, ISBN 1-56592-079-1.
- The Design and Implementation of the 4.4BSD Operating System*, M. K. McKusick, Kirk Marshall, Keith Bostic, Michael J Karels, & John Quarterman, Addison-Wesley, Reading, 1996, ISBN 0-201-54979-4.
- The Design and Implementation of the FreeBSD Operating System*, M. K. McKusick & George V. Neville-Neil, Addison-Wesley, Boston, 2004, ISBN 0-201-70245-2.
- Unix System Administration Handbook*, Evi Nemeth, Garth Snyder, Scott Seebass, Trent R. Hein, & John Quarterman, Prentice-Hall, 3rd edition, 2000, ISBN 0-13-020601-6.
- The Complete FreeBSD*, Greg Lehey, Walnut Creek, 3rd edition, June 1999, 773 pages, ISBN 1-57176-246-9.
- [McKusick et al, 1994] *Berkeley Software Architecture Manual, 4.4BSD Edition*, M. K. McKusick, M. J. Karels, S. J. Leffler, W. N. Joy, & R. S. Faber, 5:1-42.
- FreeBSD for PC 98'ers (in Japanese)*, SHUWA System Co, LTD., ISBN 4-87966-468-5 C3055 P2900E.
- FreeBSD (in Japanese)*, CUTT, ISBN 4-906391-22-2.
- Complete Introduction to FreeBSD (in Japanese)*, Shoeisha Co., Ltd, ISBN 4-88135-473-6 P3600E.
- Personal UNIX Starter Kit FreeBSD (in Japanese)*, ASCII, ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (Japanese translation)*, ASCII, ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (in German)*, Computer und Literature Verlag/Vertrieb Hanser, 1998, ISBN 3-932311-31-0.
- FreeBSD install and Utilization Manual (in Japanese)*, Mainichi Communications Inc..
- Building Internet Server with FreeBSD (in Indonesia Language)*, Elex Media Komputindo, Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, & Widjil Widodo.

- The FreeBSD Corporate Networker's Guide*, Addison-Wesley.
- UNIX in a Nutshell*, O'Reilly & Associates, Inc., 1990, ISBN 093717520X.
- What You Need To Know When You Can't Find Your Unix System Administrator*, O'Reilly & Associates, Inc., 1995, Linda Mui, ISBN 1-56592-104-6.
- FreeBSD User's Reference Manual (Japanese translation)*, Mainichi Communications Inc., Jpman Project, Japan FreeBSD Users Group, 1998, ISBN 4-8399-0088-4 P3800E.
- Online Guide for newcomers to the UNIX environment* (<http://unixhelp.ed.ac.uk/>), Edinburgh University (<http://www.ed.ac.uk/>).
- DNS and BIND*, O'Reilly & Associates, Inc, ISBN 1-56592-512-2, Paul Albitz Albitz & Cricket Liu, 1998, 3rd edition.
- Sendmail*, O'Reilly & Associates, Inc, 1997, 2nd edition, Brian Costales, ISBN 1-56592-222-0.
- Essential System Administration*, Aileen Frisch, 2nd edition, O'Reilly & Associates, 1995, ISBN 1-56592-127-5.
- TCP/IP Network Administration*, Craig Hunt, 2nd edition, O'Reilly & Associates, Inc, 1997, ISBN 1-56592-322-7.
- Managing NFS and NIS*, Hal Stern, O'Reilly & Associates, Inc, 1991, ISBN 0-937175-75-7.
- FreeBSD System Administration's Manual* (<http://www.pc.mycom.co.jp/FreeBSD/sam.html>), Jpman Project, Japan FreeBSD Users Group (<http://www.jp.FreeBSD.org>), Mainichi Communications Inc. (<http://www.pc.mycom.co.jp/>), 1998, ISBN 4-8399-0109-0 P3300E.
- X Window System Toolkit*, Digital Press, Paul Asente, ISBN 1-55558-051-3.
- C: A Reference Manual*, Prentice Hall, 1995, 4th edition, Samuel P. Harbison & Guy L. Jr. Steele, ISBN 0-13-326224-3.
- The C Programming Language*, Prentice Hall, 1998, Brian Kernighan & 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. Dobbs'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. Dobbs's Journal, January 1991-July 1992.
- TCP/IP Illustrated, Volume 1: The Protocols*, W. Richard Stevens, Addison-Wesley, 1996, ISBN 0-201-63346-9.
- Unix Systems for Modern Architectures*, Addison-Wesley, Curt Schimmel, 1994, ISBN 0-201-63338-8.

- TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*, Addison-Wesley, 1996, W. Richard Stevens, ISBN 0-201-63495-3.
- UNIX Internals -- The New Frontiers*, Uresh Vahalia, Prentice Hall, 1996, ISBN 0-13-101908-2.
- TCP/IP Illustrated, Volume 2: The Implementation*, Gary R. Wright & W. Richard Stevens, 1995, Addison-Wesley, ISBN 0-201-63354-X.
- Firewalls and Internet Security: Repelling the Wily Hacker*, William R. Cheswick & Steven M. Bellovin, Addison-Wesley, 1995, ISBN 0-201-63357-4.
- Practical UNIX Security*, Simson Garfinkel & Gene Spafford, 1996, 2nd edition, O'Reilly & Associates, Inc, ISBN 1-56592-148-8.
- PGP Pretty Good Privacy*, Simson Garfinkel, O'Reilly & Associates, Inc, 1995, ISBN 1-56592-098-8.
- Pentium Processor System Architecture*, Don Anderson & Tom Shanley, Addison-Wesley, 1995, 2nd edition, ISBN 0-201-40992-5.
- Programmer's Guide to the EGA, VGA, and Super VGA Cards*, Richard F. Ferraro, 3rd edition, Addison-Wesley, 1995, ISBN 0-201-62490-7.
- 80486 System Architecture*, Tom Shanley, Addison-Wesley, 1995, 3rd edition, ISBN 0-201-40994-1.
- ISA System Architecture*, Tom Shanley, Addison-Wesley, 3rd edition, 1995, ISBN 0-201-40996-8.
- PCI System Architecture*, Tom Shanley, Addison-Wesley, 1995, 3rd edition, ISBN 0-201-40993-3.
- The Undocumented PC*, Frank Van Gilluwe, Addison-Wesley, 1994, ISBN 0-201-62277-7.
- Bell System Technical Journal, Unix Time-Sharing System*, American Telephone & Telegraph Company, July-August 1978, Vol 57, No 6, Part 2, ISSN0005-8580.
- Lion's Commentary on UNIX*, John Lion, ITP Media Group, 1996, 6th edition, ISBN 1573980137.
- The New Hacker's Dictionary*, Eric S. Raymond, MIT Press, 1996, 3rd edition, ISBN 0-262-68092-0.
- A quarter century of UNIX*, Peter H. Salus, Addison-Wesley, 1994, ISBN 0-201-54777-5.
- The UNIX-HATERS Handbook*, Steven Strassman, Daniel Weise, & Simon Garfinkel, IDG Books Worldwide, Inc, 1994, ISBN 1-56884-203-1.
- Life with UNIX — special edition*, Don Libes & Sandy Ressler, Prentice-Hall, 1989, ISBN 0-13-536657-7.
- The BSD Family Tree* (<http://svnweb.FreeBSD.org/base/head/share/misc/bsd-family-tree?view=co>), 1997.
- Absolute BSD*, Michael Lucas, No Starch Press, June 2002, ISBN 1-886411-74-3.
- The C/C++ Users Journal*, R&D Publications Inc., ISSN 1075-2838.
- Sys Admin — The Journal for UNIX System Administrators*, Miller Freeman, Inc, ISSN 1061-2688.