

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

Ç ìÜäá Ôåêçñßúóçò ôïõ FreeBSD

Ός ÷ ιΎδ ΆννδΠόάεδ άέα οί FreeBSD 6.x έάέ 7.x

άδϋ Ç ĤŨάά Ōάέιçñßùόçδ όιϋ FreeBSD

ΆçìíόέάδìŸñ \$FreeBSD: doc/el_GR.ISO8859-7/books/faq/book.sgml,v 1.16 2009/02/09 18:04:38 keramida Exp \$
ĐíάδìάόέέŨ Άέέάέπìάά © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Ç
ĤŨάά Ōάέιçñßùόçδ όιϋ FreeBSD

Ōì έάßìάñ άόδϋ đāñέŸ ÷ άέ όεδ Ός ÷ ιΎδ ΆννδΠόάεδ άέα όεδ άέäüόάεδ 6.x έάέ 7.x όιϋ FreeBSD. ¼έäò íέ
έάόά ÷ ùñΠόάεδ έäññάßόάέ üόέ έό ÷ Ÿìοί άέα όçí Ÿέäüόç 6.x έάέ όεδ ìάόäāñŸόόāñäð äέäüόάεδ, άέδϋδ άí άíάόŸñάόάέ
άέάοññάόέέŨ. Άí άíάέάόŸñάόόā íά ìάδ äìççΠόάόā όā άόδϋ όì Ÿñāì, όόāßέόā Ÿíά email όόçí çέάέδññíέέΠ έßόόā ñŨάάδ
όάέιçñßùόçδ όιϋ FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc>). Ç όäέäðόāßā Ÿέäüόç άόδìŸ όιϋ
έάέìŸñò άßíάέ đŨíόìόā άέάέŸόέìç όόçí äέέδδóäêΠ όìđìέäóßā όιϋ FreeBSD
(http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/faq/index.html). Ìđññάßόā äđßόçδ íά όçí έάόäāŨόāόā ùδ Ÿíά
ìāäŨέì άñ ÷ äßì HTML (book.html) ìŸόù HTTP Π äέüìā έάέ ùδ áđέü έάßìάñ, PostScript®, PDF, έέđ. áδϋ όñí
äìδççñāόçδΠ FTP όιϋ FreeBSD (<ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/>). Ìđññάßόā áέüìā íά øŨíäðā όόεδ Ός ÷ ιΎδ
ΆννδΠόάεδ (<http://www.FreeBSD.org/search/index.html>).

Ç áέάñΠ έάέ ÷ ñΠόç όā ññòΠ đçāāßìð έπāέέā (SGML DocBook) Π όā 'ìäðäēüόόέόìŸíç' ññòΠ (SGML, HTML, PDF, PostScript, RTF έìέ) ìā Π
÷ ùñßð äέέäāŸð, äðέδñŸđäðáέ äöüοíí íέ đāñāέŨδϋ đññüðìέŸόάεδ όçññŸíόáέ:

1. Ç áέάñΠ όā ññòΠ đçāāßìð έπāέέā (SGML DocBook) đñŸđäέ íά äέάόççñāß όçí đāñāđŨññ äΠέϋόç đíäöìäóέέπí
äέέäέüñŨδϋí, áδδΠ όç έßόόā ìā đññüðìέŸόάεδ έάέ όçí äđüìāíç đāñŨāñāöì όόεδ đñΠòäð āñāñŸð όιϋ άñ ÷ äßìð,
äìäðŨäέçðäð.
2. Ç áέάñΠ όā ìäðäēüόόέόìŸíäð ññòŸð (ìäðŨðñāόç όā Ũέέā DTD, ìäðäðññđΠ όā PDF, PostScript, RTF Π Ũέέäð
ññòŸð) đñŸđäέ íά áíáđāñŨāέ όçí đāñāđŨññ äΠέϋόç đíäöìäóέέπí äέέäέüñŨδϋí, áδδΠ όç έßόόā ìā đññüðìέŸόάεδ,
έάέ όçí đāñāέŨδϋ đāñŨāñāöì öüοí όόçí όäέìçñßùόç üοí έάέ όā Ũέέì ðέέέü đññ đāñŸ ÷ äðáέ ìāæß ìā όçí áέάñΠ.

Όçíáíόέέü: ÁŌŌÇ Ç ŌÄÈÌÇÑÈŨŌÇ ÄÉÍÄŌÄÉ ÄĐÌ ŌÇÌ ÌÄÄÄ ŌÄÈÌÇÑÈŨŌÇŌ ŌÌŌ FREEBSD "ŨŌ Å×ÄÉ"
ÉÁÉ ÄÄÍ ĐÄÑÄ×ÄŌÄÉ ÉÄÍÉÄ ÄÌÄŌÇ Ç ÄÌÄŌÇ ÄÄÄŌÇŌÇ, ŌŌÌĐÄÑÉÉÄÌÄÄÌÄÍÄÍŨÌ, ÄÉÉÄ ×ŨÑÉŌ ÍÄ
ĐÄÑÉÌÑÉÆÄŌÄÉ ÌÌÌÍ ŌÄ ÄŌŌÄŌ, ÉÁÉ ŌŨÌ ÄÌÄŌŨÌ ÄÄÄŌÇŌÄŨÌ ÄÉÄ ÄÌĐÌÑÄŌŌÉÌŌÇŌÄ Ç
ÉÄŌÄÉÉÇÉÌŌÇŌÄ ÄÉÄ ÌĐÌÉÌÄÇĐÌŌÄ ŌŌÄÉÄÑÉÍÄÌÍ ŌÉÌĐÌ. ŌÄ ÉÄÍÉÄ ĐÄÑÉĐŌŨŌÇ ÄÄÍ ÄŌÉŌÍÄŌÄÉ Ç
ÌÄÄÄ ŌÄÈÌÇÑÈŨŌÇŌ ŌÌŌ FREEBSD ÄÉÄ ÌĐÌÉÄŌÄÇĐÌŌÄ ÄÌÄŌÄŌ, ÄÌÄŌÄŌ, ŌŌ×ÄÉÄŌ, ÄÉÄÉÄŌ,
ŌÇÍÄÍŌÉÉÄŌ, Ç ÉÄŌÄ ĐÄÑÉĐŌŨŌÇ ÄÉÄÄÄŌ (ŌŌÌĐÄÑÉÉÄÌÄÄÌÄÍÄÍŨÌ, ÄÉÉÄ ×ŨÑÉŌ ÍÄ ĐÄÑÉÌÑÉÆÄŌÄÉ
ÌÌÌÍ ŌÄ ÄŌŌÄŌ, ÉÁÉ ŌÇÍ ÄÄŌÍÄÍÉÄ ĐÑÌŌÄÄŌÇŌ ŌÄ ÄÍÄÉÉÄÉŌÉÉÄŌ đÇÄÄŌ Ç đÇçñāŌÉÄŌ, ŌÇÍ
ÄÄŌÍÄÍÉÄ ×ŨÇŌÇŌ, ŌÇÍ ÄĐŨÉÄÉÄ ÄÄÄÌÄÍŨÌ Ç ÉÄÑÄÌŌŌ, ÉÁÉ ŌÇÍ ÄÉÄÉÌĐÇ ÄĐÉ×ÄÑÇÍÄŌÉÉÜÌ
ÉÄÉŌÍŌÑÄÉÜÌ), đÌŌ đññíÉÄÍŌÍŌÄÉ ìā ÌĐÌÉÌÄÇĐÌŌÄ ŌÑÌĐÌ ÄĐÌ ŌÇ ×ŨÇŌÇ ÄŌŌÇŌ ŌÇŌ ŌÄÈÌÇÑÈŨŌÇŌ.

Ōì FreeBSD äßíáέ Ÿíά έάόì ÷ đññŸññ äìđñíέέü óŸìāññ όιϋ FreeBSD Foundation.

Ìέ έŸíäð 3Com έάέ HomeConnect äßíáέ έάόì ÷ đññŸññ äìđñíέέü óŸìāññ όçδ 3Com Corporation.

Ìέ έŸíäð Π đñŨŌáéd Adobe, Acrobat, Acrobat Reader, έάέ PostScript äßíáέ äßóā έάόì ÷ đññŸññ äìđñíέέü óŸìāññ Π äìđñíέέü óŸìāññ όçδ Adobe
Systems Incorporated όόéd ÇññŸññ đññέóäßäð Π/έάέ óā Ũέέäð ÷ ðñäð.

Ç đñŨŌçç Sound Blaster äßíáέ äìđñíέέü óŸìāññ όçδ Creative Technology Ltd. όόéd ÇññŸññ đññέóäßäð Π/έάέ óā Ũέέäð ÷ ðñäð.

DieeYō ađu oēo ēYīaēo P ōnŪoāēo iē iŕiBaō ÷ nqōēniŕiēYīoāē ađu ōiŕō ēaōaōēāōaōōYō P ōiŕō ðuēqōYō ōiŕō āēā iā āēāēnBīŕi ōā ðniūiŕiā ōiŕō ēāūniYīoāē āiŕiŕēēŪ oYīaŕēā. ¼ŕŕō aŕōYō āiōāīBāēiYōāē ōā aŕōū ōi ēāBīāŕi ēāē āēā ūoāō ađu aŕōYō āūnBāēē q ħŪāā ĀŕŪðōŕiçq ōiŕō FreeBSD ūōē āBīāē ðēēāūiŕ iā āBīāē āiŕiŕēēŪ oYīaŕēā, ēā āāBōā Yīā ađu ōā oYīaŕēā: “TM” P “©”.

ΘΒίαεάο Δαñέα ÷ ñÝíuí

1	ΆέόάũāP	1
2	Ôâèìçñßùόç êάέ ÔðíóôPñέίç	7
3	ΆãêáoÛóóάόç	12
4	Óðìááôüôçôá Ôêéêÿ	22
4.1	ÃâíéêÛ	22
4.2	Ãñ ÷ éóâêôííéêÝð êάέ ÃðáíâñãáóóÝð	22
4.3	Óêëçññß äßóêÿé, ññÛããò ôáéíßáò, ïäçãñß CD êάέ DVD	23
4.4	Ðëçêôññëüüáéá êάέ ðñíðßêéá	25
4.5	ÓóóéãðÝð óáéñéáêêðò äðééíéíñíßáò êάέ Äéêôýùόçð	27
4.6	ÓóóéãðÝð P ÷ ïð	28
4.7	¶ëêÿ ðééêü	29
5	Áíôêíãððéόç ÐññâçìÛðuí	31
6	ÃìðñéêÝð ÃóãññãÝð	46
7	ÃóãññãÝð Ôäééêÿ ×ñPóόç	49
8	Ñýèíéόç ÐðñPíá	53
9	Äßóêÿé, ÓóóðPíáðá Ãñ ÷ äßuí êάέ ÔñðòòÝð Äêêßçççð	56
10	Äéá ÷ äßñéόç ÓóóðPíáðò	67
11	Ôñ Óýóôçíá X Windows êάέ ïé ÄéêñéêÝð Êñóüêãò	75
12	Äéêôýùόç	84
13	ÁóóÛéãéá	91
14	PPP	94
15	ÓáéñéáêÝð Äðééíéíñíßáò	106
16	ÄéÛðññãò ÅñòðPóáéò	113
17	×éÿññ êάέ FreeBSD	117
18	Ðññ ÷ ùñçÝíá ÈÝíáðá	120
19	Ãð ÷ áñéóóðßáò	127
	Bibliography	128

ÊáôÛëĩãïò ÐéíÛêùí

3-1. ÌÝãéóôá ïããÝèç áñ÷ãßùí.....	20
12-1. Network cards based on the DEC PCI chipset	86

ÊâöÛëáéí 1
ÁéóáãùãƆ

Êáèò Èèàâá óêè Òõ÷íÝ ÆñùôÞáèò ôïò FreeBSD 6.x-7.x!

[illegible]

÷ áé éáóááéçèáß ðÛëå ðñíóðÛëåáé þóóá áóðü ðí FAQ íá ðåñéÝ÷ áé üóí ðí äóíáðüí ðåñéóóüðåñåð ðëçñííññßåð. Áí éÝëåðå íá ðÛíåðå ðÛðíëåð ððíåáßíåð ð÷÷åðéÛ íå ðñüðíðð ååèðßüçðð ðíð, óåð ðåñåáéçíýíå óóåßèðå íåð Ýíå email ððç çëåðñíííééß ðßðóå íÛååð ðåèìçñßüçðð ðíð FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc>).

1. Ôé åßíáé ôï FreeBSD;

Αί δοιοῖνβά, οἱ FreeBSD ἀβίαέ Υἱά εἰεοῖοῖνᾶεέυ ούοόοῖα ούοῖο UNIX® ᾶεά οεό ἀν÷εόᾶεοῖνῆέΥο AMD64 εἰεά Intel® EM64T, i386™ PC-98, IA-64, ARM®, PowerPC® εἰεά UltraSPARC®. Ἀάοβᾶάοᾶε οόοῖ Υἱεᾶῖοο “4.BSD-Lite” οῖο Δάῖᾶεόοοῖοῖο οῖο Berkeley, ἰᾶ ἔῤῥῆεᾶο ᾶᾶεοῖοῖοᾶεό ᾶδῦ οῖ “4.BSD-Lite2”. Ἀάοβᾶάοᾶε ᾶδβόοο Υἱᾶόᾶ οόοῖ ᾶᾶᾶοβᾶ ἰᾶόᾶοῖνῦο (port) οῖο “Net/2” (ᾶδβόοο ᾶδῦ οῖ Berkeley) οῖο Υἱεῖᾶ ᾶδῦ οῖ William Jolitz (οῖ ἰοῖβά Υἱεῖᾶ ᾶῦοόο ἰᾶ οῖ ὑᾶᾶ “386BSD”), ᾶῖ εἰεά ᾶῖ Υἱ÷ᾶε ἰᾶβᾶε ο÷ᾶᾶῖ οβδῖοᾶ ᾶδῦ οῖ ἀν÷εέυ ἐῖᾶεᾶ οῖο 386BSD.

ΔεοῖνΥόοᾶᾶο δᾶᾶεᾶᾶοῖ ᾶεᾶ οῖ οε ᾶβᾶε οῖ FreeBSD εἰεά οῖο ἰοῖᾶᾶ ἰᾶ εἰεοῖοῖνᾶῖοᾶε ᾶεᾶ ᾶοῤο, ἰοῖᾶᾶῖοᾶ ἰᾶ ᾶᾶᾶῖοᾶ οόοῖ ᾶεεόοῖᾶῖ οῖοῖᾶῖοᾶῖ οῖο FreeBSD (<http://www.FreeBSD.org/index.html>).

Οι FreeBSD ÷ñçøëïïðíëáßòáé áðu ãòáëñßàð, ðáñï÷ãð òðçñãòéþí Internet (ISPs), ãñãòïçðÝð, ãðáããëíàòßàð òçð
 Ðççñïïíëéßð, òíëòçðÝð éáé íëééáéíýð ÷ñßòáð áðu ùëí òïí èùòïí, óòç ãíðëáéÛ òíðð, óòçí áëðáßããòóç éáé áéá
 áíáð÷ß.

Ἄεά ὀανέοόουὸαῖὰ εᾶδὸθῖν Ὑῖναέσ ο ÷ ᾠέέεÜ ιᾶ οῖ FreeBSD, ὀᾶῖᾰᾰᾰᾰᾰᾰ ᾱᾰᾰÜöᾰᾰ οῖ Ἀᾰ÷ᾰᾰῖᾰᾰᾰ οῖᾰ FreeBSD ([http://www.FreeBSD.org/doc/el GR.ISO8859-7/books/handbook/index.html](http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/index.html)).

2. Ðiëiò åbíáé ĩ óëiðüò ôiõ FreeBSD Project;

Í deĩðũð ðĩð FreeBSD Project áßĩáé íá ðãnÝ ÷ áé ĩĩáéðĩéũ ðĩ ĩðĩßĩ íá ĩðĩñáß íá ÷ ñçðéĩðĩéçéáß áéá ĩðĩéĩðóáPðĩðĩá ðéĩðĩýð éáé ÷ ùñßð ðãñéĩñéðĩýð. Ðĩēēĩß áðũ ãĩÙð Ý ÷ ĩðĩã ðãñĩáýðáé çĩĩáĩðéēŨ ðóĩĩ ēĩĩáéá (éáé ðóĩĩ ßáēĩ ðĩ Ýñĩĩ) éáé ðáßĩĩðñã ããĩ ēá ĩáð ðãßñĩáēá éáé ēŨðĩēá ÷ ñçĩĩáðéēP áðĩĩçĩßĩðũç ðĩð éáé ðĩð, áēēŨ ðáßĩĩðñã ããĩ áðēĩÝĩĩðĩã ðã áðũð. Ðéóóãýĩðĩã ĩũé ç ðñĩðç éáé ááóéēũðãñç “áðĩðóēĩP” ĩáð áßĩáé íá ðãnÝ ÷ ĩðĩã ēĩĩáéá ðã ĩēĩðð, áéá ĩðĩéĩáPðĩðĩá ðēéáĩũ ðéĩðũ, Ýðóé ðóðã ĩ ēĩĩáéáð ĩáð íá Ý ÷ áé ðçĩ áðñýðãñç áĩĩĩáðP ÷ ñPðç éáé áðéððã ÷ Ũĩĩáé ðĩ ĩããáēýðãñĩ áĩĩĩáðũ ĩũáēĩð. Ðéóóãýĩðĩã ĩũé áðũðð áßĩáé Ýĩĩáð áðũ ðĩðð ðçĩĩáĩðéēũðãñĩðð ðéĩðĩýð ðĩð Áēãýēãñĩð Ēĩĩáéðĩéĩý éáé ðĩĩ ððĩðóçñßĩĩðĩã ēãñĩŨ.

Ἰ ἐπαέεαδ, ὁδὶ αἴΥῖονι οἷθ δ̣αβαιοῖ ἰαδ ἐπαέέα, ἱ ἰδιΒιοῖ ἀιδβοδαέ ὀδς Ὑαάέα ÷ ñÞόçð GNU General Public License (GPL) (<http://www.FreeBSD.org/copyright/COPYING>) Þ GNU Library General Public License (LGPL) (<http://www.FreeBSD.org/copyright/COPYING.LIB>) ὕ÷äē δāñέοούδāñῖοδ δāñέῖñέοῖγϝ, äēÛ äδοῖß ēēßῖοῖ, äοδò÷÷ð, ðñῖοδ ὄçῖ ὀðῖ÷ñÝúοç äēäýēāñçð ðñῖοääαόçð áῖοß äéá οἱ áῖοßēäοἱ ðῖο äßῖáē éäē οἱ ðēÝῖῖ ὀóῖçēéοῖÝῖ. Èüäü ὀçð äοῖçῖÝῖçð ðῖέοðēῖēüοçόäð ðῖο ἰδιñāß ἱá ðñῖέýοäē äðu ὄçῖ äῖδιñέēÞ ÷ ñÞόç εῖαέοῖέēῖγ GPL, ðñῖοðäēῖγῖä äāēēÛ ἱá

Άίβιαέά ίά άάάάΥπιδιά υόε ς εΥίγ “άέάγέανη” ÷ ñçόείηδιδέαβόάέ άάβ ίά άγί όñυδιδόδ, ι Υίάδ όçιάβίάέ “άυñάŸη”, έάέ ι Ÿέειδ όçιάβίάέ “ιδιδñάβόά ίά όι εŸίάόά υόε έΥέάόά”. Άέόυδ άδυ Υίά-άγί δñŸάιάόά διό *άάι* ιδιδñάβόά ίά εŸίάόά ίά όιη έβραέέα διό FreeBSD, άέα δñŸάάέάίά ίά έό÷δñέόέάβόά υόε όιη ãñŸάόά, ιδιδñάβόά όόά άέβραέα ίά όιη εŸίάόά υόε έΥέάόά.

6. Διέαδo áβρίáé íé áéáöíñÝò íáðáíý òíð FreeBSD éáé òíð NetBSD, OpenBSD éáé òúñ Üëëüñ BSD óóóóçíÜòúñ áñíééðýý èþáééá;

Í James Howard Ý÷æē āñÜøæē íéá éæēP āîPāçóç ôçð éóóññBāð éæē òuì æéáöïñpí íāðáíý òuì æēÜöññuì Ýñāuì æéá òí
 DaemonNews (<http://www.daemonnews.org/>), óðí Üñēñ Ç ÍéêîāÝíáéá Èāéóíññāēēpí BSD
 (http://ezine.daemonnews.org/200104/bsd_family.html).

7. Διέα αβιάε ç ôäëãõôäβá Ýêäïóç ôiõ FreeBSD;

Ἄσὸς ὁς ὀρέσῃς ὁδῶν· ἰὼν αἶψά σου ἐξέρχεται ἐκ σου ὁδὸς ἁγία σου ὁδὸς σου FreeBSD. Ἰὲ ἀδελφοί μου ἀγαπῶντες τοὺς FreeBSD
 σου ἀγαπῶντες ἀλλήλους· ἵνα εἶς ἄνθρωπος ὁδὸς σου ἀδελφὸς σου ἐκ σου ὁδὸς σου. Ὁ ἀνὴρ σου ἀγαπῶντες 6.x ἀντιέστην ἀπὸ σου
 6-STABLE εἰς ὁ ἀνὴρ σου ἀγαπῶντες 7.x ἀντιέστην ἀπὸ σου ὁ ἀνὴρ σου 7-STABLE.

ΙΥ÷né εάε όçí Ýéäíróç 7.0, ç óäénÛ äéäüóäüí 6.x Þóäí äíüóðÞ ùò -*STABLE*. Ûóóüóí äüü óçí 7.0 εάε íäòÛ, ç óäénÛ 6.x éäüüäÞóáε üóε äéóÛÄäóáε óóç öÛóç óçò “äéðäóäíÝíçò ððíóðÞnéíçò” εάε éä εäíäÛíäε üüí äéíñεÞóáεð äéä öíäíñÛ ðñíäεÞíäóä, üðüð ð.÷. äóðÛ ðíò äíáóÝññíðáε óä éäíÛ äóóäεäíäð. Éä öðÛññíòí ðñäéóóóðññäð äéäüóáεð äüü óç óäénÛ 6-*STABLE*, äééÛ äóðÞ éäüüäÞóáε ðéÝíí “íäðññäóíÝíç” (legacy) εάε íé ðéí ðñüóóäðäð ääéðéÞóáεð éä äíóüíäóüñéýí üüí óóíí ééÛäí 7-*STABLE*.

Version 8.2 ([ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.2-RELEASE/](http://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.2-RELEASE/)) is the latest release from the 7-*STABLE* branch; it was released in ÉíÿÉí 2010. Version 7.4

(<http://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/7.4-RELEASE/>) is the latest release from the 6-*STABLE* branch; it was released in Æœôô 2010.

Ç Ýéäïð 8.2 (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.2-RELEASE/>) âßíáé ç ðēÝŷŷ ðâēâððâßâ ðçð ðâēñŰð 7-*STABLE* éâé âçŷēññâßēçâ ððéð Éŷŷēŷ 2010. Ç Ýéäïð 7.4

([ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/7.4-RELEASE/](http://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/7.4-RELEASE/)) ἁβία εῖ δεΥίι δαεᾶδὸάβα ὀço ὀαένÜð 6-STABLE
εᾶε ἀçìεrōñāèçêâ ὀόέó ÌŨñóér 2010.

[illegible]

ἸΎάδ ἀεαυοάεδ ἀιοάββαηοάεδ εὔεα ἰανέειρῳ ἰβράδ. Ἀρ ἐαέ θρεεἰβ ἀδεέΎαῖορ ἰά αεαόκνῆρῳάεδ ἀρῑαῖνῷΎῆε ἐαέ θεἰ
 δδ÷ἰῦ ἰΎῳῳ οἰῳ δῑαάβῑῳ ἐπαέέα οἰῳ FreeBSD (ἀάβῳά οέδ ἀνῑδῑῳάεδ οῳἰ FreeBSD-CURRENT ἐαέ
 FreeBSD-STABLE), ἰε ἀεαυοάεδ ἀβῑάεδ εὔῳεδ δαῖαδῦῑῑ αδῦ οδἰ÷ῆΎῡος, ἐαέῑῑ ἰ δῑαάβῑῳ ἐπαέέαδ ἀβῑάεδ δῑῑέοῳῳῳῑ
 ἸΎάδ ἐέῑῑῑῑῑδ οδῑ÷ἰδ.

Đã nêu ở trên, để tải về và cài đặt FreeBSD, bạn cần phải có một máy chủ để tải về và cài đặt. Bạn có thể tải về và cài đặt FreeBSD từ trang web của FreeBSD (<http://www.FreeBSD.org/releng/index.html>) hoặc từ trang web của FreeBSD.

8. Ôé åßíáé ôï FreeBSD-CURRENT;

Ôi FreeBSD-CURRENT

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#CURRENT) ἁβιάε ς ὄδῳ ἁίΥέειϋ Ἰέαιος ὁτῶ ἐἀέοιῶνᾱέειϋ ὁόόῶΠιαόιῶ, ὁι ἰῶιβῖ ἰᾱ ὁιῖ ἐἀέῆῆ ἐᾱ ἁῖᾱέῖ÷ἐᾱβ ὁοῖ ἰΥῖ ἐῆÜᾱῖ FreeBSD-STABLE. Ἀέᾱ ὁῖ ἐῡᾱῖ ᾱόῶῡ, ᾱᾱῖῖῶῆÜᾱᾱῖ ὁῶῖῖῖῶ ἁῖᾱέᾱῶÝῖῖῖ ἰῡῖῖ ὁᾱ ἰῡῖῖῶ ᾱó÷ῖῖῖῖῖῖῖῖῖ ἰᾱ ὁϋῖ ἁῖÜῖῖῖῖῖ ῆῖᾱέᾱ ὁῖῶ ὁόόῶΠιαόῖῶ ῆᾱῖ ὁᾱ ὁῆῆϋῖῖῖῖῖῖῖῖῖ ÷ῖῖῖῖῖῖῖῖῖ. Ἀᾱῖῖῖ ὁῖ ὁ÷ᾱῖῖῖ ὁῖΠια (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#CURRENT) ὁῶῖ ἁᾱ÷ᾱῖῖῖῖῖῖῖῖῖ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/index.html) ᾱέᾱ ῆᾱῖῖῖÝῖῖῖῖῖ ὁ÷ᾱῖῖῖÜ ἰᾱ ὁῖ -CURRENT.

Αί άάρ άββόά άήρεάεάυΐΐό ία όί εάέόιόνάεάυ όύόόά, Ρ άάρ άββόά έέάυό ίά άίάάυήνβόάά ός άέάόινΐ ίάόάί άύό όήάάάόέέί έάέ άύό όήόύήέίύ όήάέΡάόίό, ίΐέέί άάρ έά όήΎόάέ ίά ÷ήςόέίόίέΡάάά όί FreeBSD-CURRENT. ΐ έέΐάό άόόόό ήέόίΎάό όινΎό άίάέββόάάέ όίέύ άήΡάήά, έάέ ίόήήά άέύίά έάέ ς ήάάάέΡόόέός όίό ίά ίςί άήίάέ άόίάό άέά άήέάΎό όόίά÷ύίάήά ίΎήά. ΐόίέ ÷ήςόέίόίέίύ όί FreeBSD-CURRENT άίάΎίάάέ ίά άήίάέ έέάήβ ίά άίάέΎίό όά ύόίέά όήάέΡάάά έάέ ίά όά άίάόΎήόί ύύί άί έάύήίύ ύόέ όήύέάέόάέ άέά όςίάίόέέΐ έΐές έάέ ύ÷έ άέά “ίέέήόήάέΡάάά”. ΆήύόΡάέό όίό όύόίό “όί make world όάήΎάάέ έΐΐίέά όόΐέίάά ό÷άόέέΐ ήά όά groups” όός έββόά όά÷όάήήάβίό -CURRENT, ίόήήά ίά άίόέίάόύόέόόίύ όάήέόήίςόέέΐ.

Εὐεῖς ἰπρία, δάνῦναιόαε οόεαιεὐοῶδα (<http://www.FreeBSD.org/snapshots/>) ἀεῦοάις ὅιῳ ἀάβæííóáε οόçí ðñÝ ÷íῳά
εάῶῦοάοç òιι ἐεῦῶι -*CURRENT* εάε -*STABLE*. Ἰε οόι÷íε ὀβῶι ἀδῦ Εὐεῖς Ἰέαιόç οόεαιεὐοῶδῖῳ ἀβίαε:

- [illegible]

[illegible]

Ðāñéóóúôāñāò ðēçñīōīñāò áēá ôá ôôēāīēúôôðā ðīñāñôā íá āñāñôā ôôç ôāēñāā ôôēāīēúôôðūī āēāüôāūī
(<http://www.FreeBSD.org/snapshots/>).

[illegible]

9. Ôé åßíáé ôï FreeBSD-STABLE;

1/4ôáí êôëëïüüñçôá ôï FreeBSD 2.0.5, ç áíÜðôôïç ôïô FreeBSD ÷üñßóôçêâ óâ áýí êêÜäüò. Ì Ýíáò êêÜäüò ïñÜóôçêâ -STABLE (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#STABLE), êâé ï Üëëïð -CURRENT

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/current-stable.html#CURRENT). Οι *FreeBSD-STABLE* ἀδᾶδῆγιάδᾶε ᾶ Δᾶνι - ᾶδ ᾶδῆᾶᾶᾶᾶ Internet (ISPs) ᾶε ᾶᾶᾶ ᾶᾶᾶᾶᾶ - ᾶᾶᾶᾶ, ᾶᾶᾶ ᾶ ᾶᾶᾶᾶ ᾶᾶᾶᾶᾶ ᾶε ᾶ ᾶᾶ - ᾶᾶ ᾶᾶᾶᾶᾶᾶᾶ - ᾶᾶᾶᾶᾶᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶᾶᾶ ᾶᾶᾶᾶᾶᾶᾶ. ᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶᾶᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶᾶᾶᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶᾶᾶᾶᾶᾶ. ᾶᾶ ᾶᾶ ᾶᾶᾶ ᾶᾶᾶᾶ, ᾶᾶ

FreeBSD-CURRENT ἀνθεώαοάε οά ιέα ιιιαέεP αάεUειδϙ ἀναιιP αrUδδoιδϙ αδι οϙr αδι÷P οδ εδөөiοiнBαd οϙd
 Ýεαιoϙd 2.0, ιαϙαpιoάd δnιd οϙr Ýεαιoϙd 8.2-RELEASE αέεU εάε ιαdU αδι αdοP. Άέα δaνέoóuóaηaδ δϙϙnιοiнBαd
 ααBóα oι Uηeηι “δnιaδiεiαóBα Άέauoáιr oιo FreeBSD: ΆϙiεiοηαBα oιo ΈέUαiο εαιoϙd
 (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/releng/release-proc.html#REL-BRANCH)”.

I ēēÜāřō 2.2-STABLE āāēāōāēāBōēçēā iā ōçī ēōēēroĩñBā ōçō Ýēārīçōđ 2.2.8. I ēēÜāřō 3-STABLE ôāēāBūōā iā ōçī ēōēēroĩñBā ōçōđ Ýēārīçōđ 3.5.1, đřō Pōái ēāé ç ôāēāōđāBā ōçōđ ôāēñÜđ āēāüōāũ 3.x. I ēēÜāřō 4-STABLE ôāēāBūōā iā ōçī ēōēēroĩñBā ōçōđ Ýēārīçōđ 4.11, ōçōđ ôāēāōđāBād ōçōđ ôāēñÜđ 4.x. Īē iuiāđ āēēāYđ đřō āBřřōāē ēāóŬ āŬōç ôā ēŬēā Yřā āđu āđōřýđ đřōđ ēēÜāřōđ, Ý÷řōř ō÷Ýōç iā āēĩņēpōāēđ ôā ēārŬ āōōāēāBād. Ç đđřōđPņēřç đũř ēēÜāũř 5-STABLE ôđāňŬōçōā iā ōçī Ýēārīçōđ 5.5, ōçī ôāēāōđāBā Ýēārīçōđ ōçōđ ôāēñÜđ 5.x. Ç đđřōđPņēřç āēā ōç ôāēñÜ āēāüōāũ 6-STABLE ēā ôōřā÷ēōđāB āēā ēBāř ēāēñũ āēũřā, āēēŬ ēā āđēēārřōñũēāB đāņēōōũōāņř ōā āēĩņēpōāēđ ēārřř āōōāēāBād ēāē Üēēũ řřāāņř đņřāēçĪđũř.

Η έρευνα 8.2-STABLE αβιάει i -STABLE έρευνα διο ανβόεααέ οδύ ατάνP αÚδooiç. Ç oáεαooáá εöεεioiñá διο αáoβæααέ ooí έεÜäi 8.2-STABLE αβιάει ç 8.2-RELEASE, εάε Ýαεία oóéo Éíýéi 2010.

Ἰ ἐεῦἄρῶ 8-CURRENT ἁβίαέ ἰ ἐεῦἄρῶ -CURRENT ὄϊῶ ἀράδῶγῶόᾶόᾶέ ᾶῶδᾶ ὀϕ ὀόέᾶἰᾶ ᾶράἡἄῦ ἡῶᾶ ἰᾶ ᾶῑῑῑῑἡᾶῑᾶῑᾶ ῑ ῑᾶ ᾶᾶῑᾶῦ ὀῑῶ FreeBSD. Ἀᾶᾶᾶ ὀῑ ὀῑᾶἰᾶ Ὀέ ᾶβίαέ ὀῑ FreeBSD-CURRENT; ᾶῑᾶ ᾶἡᾶῑῑᾶᾶᾶ ᾶῑᾶ ᾶῑᾶῑῑᾶᾶ ὀᾶᾶᾶ ᾶᾶᾶᾶ ὀῑᾶ ἑεῦἄῑ.

10. ÊÛèà ðüôâ ãßíííôáé åđßçìàò êõêëíõíñßàò ôíõ FreeBSD;

[illegible]

Ḑāñēōōūōāñāō ḑēçñiōiñḑāō ò÷āōēēŪ iā òç āēāāēēāōḑā ēōēēiōiñḑāō (ḑāñēēāiāāñŸiñ ēāē āiūō ÷ññiñāēāñŪiāōiō āḑēēāḑiāññiñ ēōēēiōiñēḑiñ) iḑññiñí iā āñāēíýí òōēḑ òāēḑāāō ḑññiāōiēiāōāōāō āēāüōāñiñ
(<http://www.FreeBSD.org/releng/index.html>) òç āēēōōāēḑ òiḑñēāōāō òiñ FreeBSD.

Ãéá ôiôð ðéí áíèiôóéþááéò, òðŮñ÷iôí êâèçìãñéíŮ óóéâlèüôôðá (binary snapshots) üðùò áíáoÝñèçêâ ðéí ðŮû.

11. Διεύθυνση αβίασε οδωγέοιηο αέα οι FreeBSD

[illegible]

Ὑποὐνοῖ τὲ δ'ἀνεόρου ὠπᾶνδ' ὀψιὰίῃς ἔϋδ' ἀεεᾶ ἄϋδ' ὁδ' ἀϋτὶρ ἰόα' ἄδ' ὃνεί ὀόεδ' ἐβόᾶδ' ἤα' ἐὼν ἱέεϋ ὁά÷ ὁᾶν ἡλᾶσθ', ἑάε ἄλ' ὁδ' Ὑñ÷ τοῖ δ'ἀνείνῃ ὀοῖβ' ὁ÷ ἁόε' ἔϋ ἰὰ οἱ δ'ιέτ' ἐαῖ ἁ Ὑπ' τοῖ ἰ' Ὑνῖο ὁά ἁόδ' ἔϋ ὁέο ὁόαϋδ' Ὁάέο.

12. Ðíð ìðíñß íá ðññçèàððß òí FreeBSD;

ÊÛèà òçíáíðéêß Ýëáíðç òíð FreeBSD áßíáé áéáèÝóéíç ìÝóù áíðíðíð FTP áðù òíí áéáèñéóðß FTP òíð FreeBSD (<ftp://ftp.FreeBSD.org/pub/FreeBSD/>):

- Ç òáèèàððáßá èðèèíðíßá ðíð ááóßæáðáé òòí 6-STABLE, 8.2-RELEASE ìðíñáß íá àñàèáß òòíí èáðÛëíáí 8.2-RELEASE (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.2-RELEASE/>).
- Áèäüóáéð ðíð ááóßæííðáé òáÓðéáèüðððá (<http://www.FreeBSD.org/snapshots/>) áßííðáé èáèçíáñéíÛ áðù òíðð èèÛáíðð -CURRENT èáé -STABLE, èáé áíððçñáðíýí èáðÛ áÛóç üóíðð áó÷íèýíðáé ìá òçí áíÛððòíç èáé òíí Ýèää÷í òùí òáèèàððáðð ááíéÛð ðñíñáñíÛðùí.
- Ç òáèèàððáßá èðèèíðíßá ðíð ááóßæáðáé òòíí èèÛáí 5-STABLE, ç 7.4-RELEASE, ìðíñáß íá àñàèáß òòíí èáðÛëíáí 7.4-RELEASE (<ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/7.4-RELEASE/>).

Ðèçñíðíñáð ò÷áðéèÛ ìá òç áèÛèáóç òíð FreeBSD òá CD, DVD èáé Ûèèá ìÝóá ìðíñýí íá àñàèýí òòí Áã÷áénßáéí (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/mirrors.html).

13. Ðùð ìðíñß íá Ý÷ù ðñüóááóç òóç ÁÛóç ÁääñÝííí ìá òéð ÁíáóíñÝð ÐñíáèçíÛðùí;

Ìðíñáßðá íá èÛíáðá áíáæçðßðáéð òóç áÛóç áääñÝííí ìá òéð áíáóíñÝð ÐñíáèçíÛðùí ìÝóù òçð áéáðáððð áíáæçðßðáùí òòí Web (<http://www.FreeBSD.org/cgi/query-pr-summary.cgi?query>).

Ç áíðíèß send-pr(1) ìðíñáß íá ÷ñçóéíðíéçèáß áéá òçí ððíáíèß áíáóíñß ÐñíáèçíÛðùí, èáèðð èáé áéðßðáùí áéá áèèááÝð, ìÝóù çèèðñííèýíý òá÷ðñññáßíð. ÁíáèèáèðéèÛ, ìðíñáßðá íá ÷ñçóéíðíéçðáðá òçí áíðßðòíé÷ç áðíáðùðçðá ððíáíèßðð áíáóíñß ÐñíáèçíÛðùí ìÝóù òçð áéáðáððð web (<http://www.freebsd.org/send-pr.html>) ìá òç áíðèáéá áíùð ðñíñáñíáðíð òðèèñáðñçðß.

Ðñéí ððíáÛèáðá ìéá áíáóíñÛ Ðñíáèßáðíðð, ðáñáèáéýíýá áéááÛóðá òí ÁñÛííðáð ÁíáóíñÝð ÐñíáèçíÛðùí áéá òí FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/problem-reports/article.html), Ýíá Ûñèñí áéá òí ðùð íá áñÛóðá èáéÝð áíáóíñÝð ÐñíáèçíÛðùí.

14. Óé Ûèèáð ðçãÝð ðèçñíðíñéðí ððÛñ÷íóí;

Ðáñáèáéýíýá áèÝáíðá òç èßóðá òçð Óáèèçñßùóçð (<http://www.FreeBSD.org/docs.html>) òóçí èýñéá áéèððáèß òíðèèáðá òíð FreeBSD (<http://www.FreeBSD.org>).

Ôảêìçñβùόç éáé ÕđĩóôPñéìç

¼íîá

tr_TR.ISO8859-9

zh_CN.GB2312

zh_TW.Big5

Añiciáßá

Ôïýñêéêá

ÁðëïðïéçìÝíá ÊéíÝæéêá (êùäéêïðïßçóç GB2312)

ĐañaäiíoéâÛ ÊéíÝæéâ (êùäéêiđiβçóc Big5)

Όχι!Βούο: ÊÛöíéá Ýãññáoä ìðññß íá ìçí ãßíáé äéáèÝóéíá óà üëàò ôéò ãëρóóàò.

- Ός ιηθόρ οίος ααανΰοιό. Δανΰαοία όαείηηβύος όά δέπειό αέαοιηάόέηρ ιηθόρ άηαο. Εΰεά ιηθόρ Υΰ-αέ όά αέεΰ όός δεαηάεόβιαόά έαέ ιαείηάεόβιαόά. Εΰθιέά ιηθόρΰό άβιáέ δαηέόοοοάηι έαόΰεέεαό αέα άΰΆηύος online, άη ΰεεάό άβηοί δέι έαέαβόέοί άθιόΰέαόία υόάι άέόόδύειρ. Ç αέΰεάός όός όαείηηβύός όά υεάό άόόΰό όέό ιηθόρΰό άίάόόαέβιáέ υόέ ιέ άίάάηόόάό ιάό έά ιθιηίρ ιά αέαάΰοίόι όά όιβιαόά θιό όιόό άίάέαόΰηιόι, άβόά όός ιεϋς όιόό, άβόά αύίρ όά άέόόθρβιόι. Ιέ αέαέΰόέιáό άόόρ ός όόέαιρ ιηθόρΰό άβιáέ:

līnōp	ĀñiçiāBā
html-split	ÓōēēiāP īēēñī, óōīāāīYīuī iāōāīY ōīōō, āñ÷āBūī HTML.
html	īā iāāŪēī HTML āñ÷āBī ðīō ðāñēY÷āē īēūēēçñī ōī Yāāñāōī.
pdb	līnōP āŪóçð āāāñYīuī āēá ōī Palm Pilot, āēá ÷ñPóç iā ōī ðñūāñāīā áfŪāfūóçð iSilo (http://www.iSilo.com/).
pdf	līnōP ēāēīYīñō PDF ōçð Adobe.
ps	PostScript
rtf	līnōP ĀīðēīōōēōīYīñō ÊāēīYīñō ōçð Microsoft a
txt	Áðēū ēāBīāñī

Θβίαέάò Óçĩăépóăùĩ:

- a. **Ἰ**ε ἀνεέλιθ ὁάεβαυί ἀάι ἀράφαρφιδάε ἀδουιἀάα υἰάι οἰνὸπιδάα ἀδὸθ ὁς ἱιὸθ ἁαανῶοιὸ ὁοί Word. **Δ**ε^Υὸά **CTRL+A, CTRL+END, F9** ἱάὸῦ ὁς οὔνδουὸς οἶο ἐαέι^Υφῶ ἀάα ὁςι ἀρά^Υυὸς οὔι ἀνεέλιθ ὁάεβαυί.

- Ἰ ὄνυθῖο ὠοῖδβᾶόγδ ἑᾶε δᾶεᾶδᾶνβοῖᾶοῖο. Ὤδῦñ÷ῖοῖ ὠᾶᾶδ ὠñῦθῖε δῖο ÷ñçóεῖῖθῖεῖῖῖᾶε ᾶδῖδ ὠç ὠóεῖᾶῖ.
 1. ¼ὠᾶῖ ç ῖñῖῖδ ᾶβῖᾶε `html-split`, ὠᾶ ᾶñ÷ᾶβᾶ ὠοῖδεῖῖᾶεῖῖᾶε ῖᾶ ÷ñῖδῖç ὠçδῖ `tar(1)`. Ὤῖ ᾶñ÷ᾶβῖ .`tar` δῖο δñῖεῖδῖᾶε, ὠοῖδεῖῖᾶεῖῖᾶε ῖᾶεᾶῖᾶ ῖᾶ ὠῖῖδ ὠñῦθῖῖδ ὠοῖδβᾶόγδ δῖο δᾶñεᾶñῖῖῖᾶε δᾶñᾶεῖῖῖᾶ.
 2. ¼εᾶῖ ῖε ῖῖεᾶῖ ῖñῖῖᾶ ᾶçῖεῖῖῖᾶῖῖ ῖῖᾶ ᾶñ÷ᾶβῖ δῖο ῖñῖῖᾶᾶᾶε `book.iῖñῖῖ` (δ.÷., `book.pdb`, `book.html`, ἑ.ῖ.ἑ.).

Ôá añ÷âßá áôôÛ öôiðéÝæïïâé êáôüðéí ìà äýì ôñüðïò öôiðßâôçð.

Ōñüðìò	ḐāñéǎǎòḐ
zip	ÌlīnḐP 6ðìḐḐǎóçḐ Zip. Áí èÝéǎòǎ íá òí àḑíróḑìḑéÝóǎòǎ óḑí FreeBSD èá ḑñÝḑǎé íá ǎǎéǎóǎóḑḐḑǎòǎ ḑñḑḑǎ òí port archivers/unzip.

Ôñüďiò

bz2

ĐañéañáöÞ

Ç iĩnõP BZip2. Åßáé èëäüoãñí æéääãñÝíç áðu òí Zip, æëÛ äáíéÛ Äçíéíõñãß íéññüoãñá áñ÷ãß. Åæéáoáoßóóã òí port `archivers/bzip2` æéá íá áðíóðíðéÝóáoã áñ÷ãß áðòíý òíò óýðíò.

İâ õĩĩ õñũđĩ äõõũ, ç ĩĩĩõP PostScript õĩõ Åã÷ãẽñãßĩõ, õõĩđẽãõÝĩç ĩã ÷ñPõç õĩõ BZip2 ẽã äđĩẽçẽãõõãß õã Ýĩã äñ÷ãßĩ ĩã ũĩĩã book.ps.bz2 õõĩĩ ẽãõÜẽĩĩ handbook/.

[illegible]

Οι θεαίηΥέοçιά οοί ίά έάοάάÛόάοά έάέ ίά άάέάάόόóóóάά όι άάέΥόι άβίάέ υόέ Υάέόά ιδιναβόά ίά άέά÷άέnéóóάβόά
 όçí όάέιçñβυόç ÷ñçóëñιδιέπίόάό όά όόίçέέóιΥίά άñάέέάβά άέά÷άβñέόçóό άάέΥόùι όιõ FreeBSD υδùό όçí pkg_add(1)
 έάέ όçí pkg_delete(1).

[illegible]

Άέά δάνÜääéàíá, ôî FAQ, óôá ÁãäëéëÜ, óå ïñöÞ PDF, åßíáé óôî δάéÝôî ïå üñíá faq.en_US.ISO8859-1.pdf.tgz.

Άí ôî ÿŸñáôá áðôü ððñáßôá íá ÷ñçóéüððēßôáôá ôçí áéüēîðēç áñðñēß áéá íá áāēáôáôðßôáôá ôî ðāēŸôî ôñô ÁāāēēēŸ
 PDF FAQ:

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

Άϊοῦ οἱ εὐΐαοᾶ ἀοδου, ἰδιῆαβοᾶ ἱά ÷ η̣ςοείηδιέεβοᾶοᾶ ὀςι ἀίδιεῖ pkg_info(1) ᾗέα ἱά ᾏῆαβοᾶ θρω Ὶ÷ᾗ ᾗᾗᾗᾗᾗᾗᾗᾗ οἱ ᾏῆ÷ᾗβῖ.

```
# 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 ìðĩñâðâ íá äâðâ, õi book . pdf èá Ý÷áé äãéââóóâéâß óóĩ éâôÛeĩĩ

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

Áí ááí èYéàòá íá ÷ñçóèìðìéÞóàòá óá ðáéÝòá, èá ðñÝðáé íá éáóààÛóáòá iùíé óáo óá óòìðéáóíÝíá áñ÷áßá, íá óá áðíóóìðéÝóáòá éáé Ýðáéóá íá áíóéáñÛóáòá óá áíóðóòíé÷á Ýáññáóá óòç èÝóç òìòð.

[illegible]

```
# 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
```

Éa éaóáéPíáóá íá íeá óóééíáP áðü áñ÷áBá .html. Ôí ááóééü ímÜæáóáé index.html, éáé èá ðáñéÝ÷æ öíí ðBíáéá ðáñéá÷ñÝíúí, áéóááüáééü öééüü, éáé ááóííýð ðñíð óá Üééá ðìPíáóá ðíð áááñÜöíð. ÌðñíáBóá Ýðáéóá íá áíóéáñÜøáðá P íá íáóáééíPóáðá óá áñ÷áBá áóðÜ öóéó óáéééÝð öíðð èÝóáéó, üðüð áðáéóáBóáé.

3. Ðĩō ìðĩņ ĩá àñũ đęçņĩõĩđĩđò áéá ẽđóôđ çăêâňĩíéıý ôá÷öãñĩđĩđĩ õĩř FreeBSD;

Ἰδιναβόα ῥά ἀνὰβόα δεπῆαο θεϋρνιϊνῖβαο ὁδὺρ ἀφόβοδιέ÷ç ἐάδα÷-ἡνϋός διτῷ Ἀα÷ἀεñέαβιῶ ᾗέα ὀεδ ἐβόδαο
çᾗἐοññίέέϋ ὁά÷-ὄανῇαβιῶ

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

4. Ôé ñÜäàð óõæÞôçóçò åßíáé äéáèÝóéìàð äéá ôï FreeBSD;

Ἰδιναῖοι ἰά ἀναβόα θεῖναι ἐὸ δέχνηϊοῖν βαῖ οὐσί ἐάοι· πῆγοι οἷο Ἄα· ἀένειάβιο αἰά οἰό ἡῦαο οὐαῖοι Πόαιῖ
(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/eresources-news.html).

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

Íáé, ôá đăñéóóúôăňă IRC äâêôă äéäèÝôĩôĩ êáíÜëé óăæÐôçôçò äéá ôĩ FreeBSD:

- Õi éáfÜëé #FreeBSD óðì EFNet (<http://www.efnet.org/index.php>) áó÷τḗáßóáé íá òì FreeBSD, áéèÛ íçí ðÛòá àéäß áéá òá÷íéèP òðìóðÞñéíç Þ áéá íá ðñìóðäèPóáðòá íá ðáßóáðà òìðò èáìþíáð íá óáð áìçèPóìοίí áéá íá áðáèèääáßðà áðu òìí èùðì íá áéááÛóáðà óéð óáèßááð àìPèáéáð Þ íá àñáòìPóáðà ìüñð óáð. Áßíáé áðèþð Ýíá éáfÜëé óðæPðçóçò, ðñþóá áðu üèá, éáé óá èÝíáðá àéäß ìðìñáß íá èòìáßìüóáé áíÛíáóá óðì óáì, òìí áèèçòéóìü, óá ðòñçíéèÛ üðéá ùò éáé òì FreeBSD. Óáð Ý÷-ìòìá ðñìáéäìðìèPóáé! Áßíáé áéáéÝóéìì óðìí áìððçñáðçòð irc.chat.org.
- Õì éáfÜëé #FreeBSDhelp óðì EFNét (<http://www.efnet.org/index.php>) áßíáé áóéànüìÝíí óðç àìPèáéá òùí ÷ñçóðþí ðìð FreeBSD, ÷áðá ðìýý ìääáyóðañç ðééáíüðçóá íá àñáßðà áðáíðPóáéð óðéð àñüðPóáéð óáð óá áðòü, ðánÛ óðì #FreeBSD.
- Õì éáfÜëé ##FreeBSD óðì Freenode (<http://freenode.net/>) áßíáé Ýíá ááíéèü éáfÜëé àìPèáéáð ìá ðánßðìð 300 ÷ñPóðáð èÛèá öìñÛ. Íé óðæçòPóáéð ìanééÝð öìñÝð ìáðáyáìì áðu òì èÝíá, áéèÛ áßíáóáé ðñìòáñáéüðçóá óðìðò ÷ñPóðáð ìá àñüðPóáéð áéá òì FreeBSD. Áßíáóðá èáèìß óðì íá óáð áìçèPóìοίí íá éáðáèÛáðà òá ááóéèÛ, íá óáð ááßììòìá ðìð íá áíáòñÝíáðà óðì Áá÷áénßáèì üðìðá ÷ñáèÛæáðáé, éáé íá óáð éáðáèçýíìòìá àéäß üðìð ìðìñáßðà íá ìÛèáðà ðáñéóóüðáñá áéá èÝíá ðìð óáð áðáó÷τḗäß. Áßíáóðá éáðÛ áÛóç Ááäèüöüñ éáfÜëé, áí éáé Ý÷-ìòìá ÷ñPóðáð áðu üèì òìí èùðìí. Áí èÝèáðá íá ñüðPóáðà óðç ìçòñéèP óáð äèþóá, ðñìóðäèPóðá íá èÛíáðà óçí àñþóçóç óðá ÁáäèèèÛ éáé Ýðáéóá ìáðáéíçèäßðà óá èÛðìèì Üèèì éáfÜëé ##freebsd-lang áí ÷ñáèÛæáðáé.
- Õì éáfÜëé #FreeBSD óðì DALNET (<http://www.dal.net/>) áßíáé áéáéÝóéìì óðì irc.dal.net óðéð ÇÐÁ éáé óðì irc.eu.dal.net óðçí Áðñþðç.
- Õì éáfÜëé #FreeBSDHelp óðì DALNET (<http://www.dal.net/>) áßíáé áéáéÝóéìì óðì irc.dal.net óðéð ÇÐÁ éáé óðì irc.eu.dal.net óðçí Áðñþðç.
- Õì éáfÜëé #FreeBSD óðì UNDERNET (<http://www.undernet.org/>) áßíáé áéáéÝóéìì óðì us.undernet.org óðéð ÇÐÁ éáé óðì eu.undernet.org óðçí Áðñþðç. Éáèðò áßíáé éáfÜëé àìPèáéáð, èá ðñÝðáé íá áßóðá ðñìáòìèíáíóìÝíñé íá áéááÛóáðà óá Ýāāñáóá óðá ìðìßá èá óáð ðáñáðÝìøìì.
- Õì éáfÜëé #FreeBSD óðì RUSNET (<http://www.rusnet.org.ru/>) áßíáé ðñìóáíáðìèéóìÝíí óðç àìPèáéá ÷ñçóðþí òìð FreeBSD ðìð ìéèýíý óç ÑúóèèP äèþóá. Áßíáé áðßóçð Ýíá éáèü ìÝñìð áéá ìç-ðá÷íéèÝð óðæçòPóáéð.
- Õì éáfÜëé #BSDChat óðì IRCNET (<http://www.ircnet.net/>) áßíáé Ýíá éáfÜëé óá ðáñáäìòéáèP ÈèíáèèèP äèþóá, áóéànüìÝíí óðç àìPèáéá ÷ñçóðþí òìð FreeBSD. Áßíáé áðßóçð Ýíá éáèü ìÝñìð áéá ìç-ðá÷íéèÝð óðæçòPóáéð.

[illegible]

6. Ðīō īđīñþ íá āñū āđß ðęçñūīP ōđīóôPñéíc éáé âêđáßāāōóç äéá ôī FreeBSD;

Ôi DaemonNews đăñÝ÷÷ă đđβ đęçņuiP âêđăăăăôôç éăé ôđiôôPņėiç áėja ôi FreeBSD. Đăņėjoóuôăňăđ đęçņiioiňăđ iđiňăđôă íă âňăđôă ôôçi áéeôôăP ôiôđ ôiđiējaóă BSD Mall (<http://www.bsdmall.com/>).

Õĩ FreeBSD Mall ðánÝ÷æ æððòçð æðð ðëçñuĩð ððĩóðÞñéç æéá õĩ FreeBSD. Ìðĩñàßòà íá ãñàßòà ðañéóóúòãñàð ðëçñĩõĩñßàð óççĩ æéèðòáêð ðĩðð ðĩðĩèãóßá (<http://www.freebsdmail.com/>).

ἸδιείσαΠθίωα Ὑεειὸ ιναάιέοιυὸ θάνΥ÷ἀέ ἀεδάβαὰοός ἐάέ οθίόοΠñείç, ἐὰ δñÝðāē íá ἀδεέιεíμíΡοάé ìà õì Project ἀέα íá ἐάóá÷ùñçèâß óá áõðP õç èβóóá.

ΕὰοÛεὰεί 3

ÃêáôÛóôáόç

1. Διεί αν÷αβι δñÝðáέ ίά έάόääÛóù áέά ίά ðÛñù οί FreeBSD;

×ñáέÛæάόά δñβά images áέόέääðí: floppies/boot.flp, floppies/kern1.flp, έάέ floppies/kern2.flp. Óά images áóôÛ δñÝðáέ ίά ίά óά ãñÛøάά óά áέόέÝóää ÷ ñçόέιðíέðíόάό έÛðíέí ãñáάέåβí üðùò οί fdimage Þ οί dd(1).

Áί δñÝðáέ ίά έάόääÛóáôå ìüñέ óáò óά óáò áέάíñþí (distributions — áέά δãñÛääέãíá áέά ìέά äæέάôÛóóáόç ìÝóù óóóóðíáóìò ãñ÷åβí DOS), έά ãñåβóå δãñáέÛóù ìåñέÝò óóóóÛóáέò ó÷åέέÛ ìå ðíέåò áέάíñÝò ίά έάόääÛóáôå:

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

ÐèÞñáέò ìæçåβåò ó÷åέέÛ ìå όç áέάáέέάόβά έάèðò έάέ δãñέóóóóóåñåò ðèçñíóíñβåò ó÷åέέÛ ìå äáíέέÛ δñíæèðíáóά έάôÛ όçí äæέάôÛóóáόç ìðíñåβóå ίά ãñåβóå óóí ÊæòÛεάεί ðíò Ãã÷åñέåβíò ó÷åέέÛ ìå όçí äæέάôÛóóáόç ðíò FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html).

2. Óέ ìðíþ ίά έÛíù áí οί image όçò áέόέÝóáò äåí ÷ññÛάέ óå ìέά ìüñí áέόέÝóά;

Ìέά áέόέÝóά ðüí 3.5 έíòóþí (1.44MB) ìðíñåβ ίά ÷ññÝóáέ üò 1474560 bytes äåãñÝíüí. Òí ãñ÷åβí image όçò áέόέÝóáò äèèβίçόçò åβíáέ äèñέåþò áóðü ðí ìÝãæèðò.

ÓóíçέέóíÝíá έÛðç έάôÛ όçí δñíåðíέìíáóβά όçò áέόέÝóáò äèèβίçόçò:

- ÊáóÝäáóíá όçò áέόέÝóáò ÷ññβò όç ÷ñÞόç ðíò *äðäüέéý* (binary) ðñüðíò üðáí ÷ñçόέíðíέååβóάέ äíððçñåðçòðò FTP.

ÊÛðíέά δñíãñÛííáóά-ðæÛóåò áέά FTP, ÷ñçόέíðíέíýí áðü δñíåðçέíåÞ ðíí *ascii* (*έäέíÝíò*) ðñüðí ìåóáóíñÛò äåãñÝíüí έάέ δñíóðæέíýí ίά äèèÛíòí ðíò ÷ññåððñåò ðÝèò ãñåñèÞðò þóðå ίά óáέñέÛæíòí ìå óέò óóìåÛóáέò ðíò ÷ñçόέíðíέíýíóáέ áðü ðí óýóóçíá óóí ìðíβí åβíåóáέ ç èÞøç. Áóðü ó÷åäüí óβåíòñå έå έάóáóóñÝøåέ ðí image äèèβίçόçò. ÅέÝåðå ðí ìÝãæèðò ðíò image äèèβίçόçò ðíò έάóääÛóáôå: Áí äåí åβíáέ *äèñέåþò* áóðü ðíò óåβíåóáέ óóí äíððçñåðçò, ç áέάáέέάóβά ðíò έάóääÛóíáðíò έå δñÝðáέ ίå èåñççååβ ýðíðç.

Ãέά ίå δãñáέÛíøåðå ðí δññåççíá: ãñÛøðå binary óόçí ãñåñèÞ ðíò FTP, ìåôÛ ðíò έå óóíåæåβóå óóí äíððçñåðçò, έάέ ðñέí ìæέíÞóáóå ίå έάóääÛæåðå ðí image.

- ×ñÞόç όçò äíðíèðò copy ðíò DOS (ç äíòβóðíέ÷íò ãñåðέέíý ãñåæååβíò) áέά όç ìåóáóíñÛ ðíò ãñ÷åβí image óόç áέόέÝóά.

ÐñāñāŪñāāā ūðūð ðī copy, āāī ēā ēāēōīñāñāōīōī ōūōŪ ēāēð ðī image āēēñīçōçð Ŷ ÷ āē ōōēā ÷ ōāñ þōāā íā
 āēēēīāñ Ūñāāā. Ōī image Ŷ ÷ āē ōā ðēþñç ðāñēā ÷ ūñāíā ōçð āēōēŶōāð, ðñī ÷ ēŪ ðñīð ðñī ÷ ēŪ, ēāē āāī ðñŶðāē íā
 āñāōāñ ōðç āēōēŶōā ùð ēāñīēēū āñ ÷ āñī. Èā ðñŶðāē íā ðī ñāōāŶññāōā ōðç āēōēŶōā ñā “ñūū” ðñūðī,
 ÷ ñçōēīðīēþīðāð āñāāēāñā ÷ āīçēŶŶ āðēðŶāŕō (low-level) (ñðñð ð. ÷. ðī fdimage þ ðī rawrite) ðīð
 ðāñēāñŪōīñāāē ōōñī ñāçāū āāēāðŪðōāōçð ðīð FreeBSD
 (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html).

3. $\text{D}\ddot{\text{I}}\ddot{\text{o}}\ \ddot{\text{a}}\ddot{\text{n}}\ddot{\text{b}}\ddot{\text{o}}\ddot{\text{e}}\ddot{\text{i}}\ddot{\text{i}}\ddot{\text{o}}\ddot{\text{a}}\ddot{\text{e}}\ \ddot{\text{i}}\ddot{\text{e}}\ \ddot{\text{i}}\ddot{\text{a}}\ddot{\text{ç}}\ddot{\text{a}}\ddot{\text{b}}\ddot{\text{a}}\ddot{\text{o}}\ \ddot{\text{ä}}\ddot{\text{e}}\ddot{\text{a}}\ \ddot{\text{o}}\ddot{\text{ç}}\ddot{\text{i}}\ \ddot{\text{ä}}\ddot{\text{ä}}\ddot{\text{e}}\ddot{\text{a}}\ddot{\text{o}}\ddot{\text{Ü}}\ddot{\text{o}}\ddot{\text{o}}\ddot{\text{ä}}\ddot{\text{o}}\ddot{\text{ç}}\ \ddot{\text{o}}\ddot{\text{i}}\ddot{\text{o}}\ \text{FreeBSD};$

İđĩññâßôâ íá âññâßôâ ôéô ãäçâßâô ôôçí éáôá÷þñçôç ôĩô Âã÷ãéñéâßĩô ó÷âôéêÛ ìâ ôçí âãéáôÛôóâôç ôĩô FreeBSD
(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html).

4. Ôé ÷ ñǎēÛæñǎé ãéá íá âêôǎēÝòù ôï FreeBSD;

Ãéá ääüöáéó ôîõ FreeBSD áðü ôî 5.X éáé ìäÜ ÷ ñäÜæäóä Ýíá PC ìä äðáíñäáóð 486 þ éäýöáñí, ìä 24 MB þ ðäñéóóüäñç RAM, éáé ôîöÜ÷éóóí 150 MB ÷ þñîõ óôî öêçñü óáo äßóêí.

Iđĩnãßòå íá æéôæÝóåðå üëåð óéò æêüöåéð õĩõ FreeBSD ìà ìéá ÷âĩçęþĩ đĩñĩæáääñåðĩ, ĩĩũ÷-ñũĩç (MDA) êŨñóå āñåöéęþĩ, æêŮ æéå íá ÷ñçéĩñđĩēPóåðå õĩ Xorg, èá ÷ñæéåóóåßòå êŨñóå VGA P æäýóåñç.

Äåßôå åðßóçò éáé ôï ÊåöÛäéí 4.

5. Ðuò ìðĩñþ íá açíëíõñãÞòu òç æéêÞ ìiõ, ðñiíóáñiíóíÝíç æéóéÝóá åææáoÜóóáoçð;

[illegible]

Ἄέα ἰά ἀγίειϛὺνᾱβόαδᾱ ἰέα δñιόαμ̃ιὸίΎίς Ὑέαιϛός, áeiēiϛeβóδᾱ ôéð ïäçãßàð óðī Ũñēñī ó÷âôéêÛ ìà ôçí ĩñāŨîúç
Âëüöáũĩ (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/releng/article.html).

6. Ìdīnḡ íá Y÷ù ḡāñéóóúôāñá áḡū Yíá ēāēôīōñāēēÜ óóóôPíáôá óôīī òḡīēīāēóôP īīō;

Ñîîâ ìéá ìáôéÛ óôçí óáëßää ðřëáðëþí ëáôîññáëþí
(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/multi-os/index.html).

7. Ìďĩĩíý íá óõĩõđŮñĩĩõí ôá Windows ìå õĩ FreeBSD;

Ἀὰὰὰὰὰὀὀὀὀὰ ὀὀὀὀὰ ὀὰ Windows, ἑὰε ἰὰὀὔ ὀὀὀ FreeBSD. Ἰ ἑὰά÷ἑἑἑὀὀὀ ἑἑἑἑἑὀὀ ὀὀὀ FreeBSD ἑὰ ὀὰ ἑἑὀἑἑἑἑἑ ἑἑὀἑἑ ἑὰ ἑἑἑἑἑἑἑ ὀἑἑ ἑἑἑἑἑἑἑ ὀἑἑ Windows ἑἑὀ ὀὀὀ FreeBSD. Ἀὶ ἑἑἑἑἑἑἑἑἑ ὀὰ Windows ἰὰὀὔ ὀὀ FreeBSD, ἑὰ ὀὰ ὀἑἑἑἑἑ ὀὀ ἑἑἑ÷ἑἑἑὀὀ ἑἑἑἑἑὀὀ, ÷ἑἑἑ ἑἑἑ ἑὰ ὀὰ ἑἑἑἑἑἑἑ. Ἀὶ ὀὰ ὀἑἑἑἑ, ἑἑἑ ἑἑἑἑ ὀὀ ἑἑἑἑἑ ὀἑἑἑ.

8. Ôá Windows êáôÝóôñâøáí ôî áéá÷âéñéóôÞ âêêßícôçò ìîõ! Dùò èà ôîí áðáíáoÝñù;

Ἰδιναῖοις τὰ ἀδελφεία ἀγαθὰ ποιοῦσι οἱ ἀεὶ ἀδελφείᾳ ἀεὶ βίοντες οἱ τοῦ FreeBSD ἡγετῆρες. Ἰαί αὖ οἱ τοῦ ὁπαιῶ ἀδελφῆς οἱ ὁπαιῶτες:

- Άεôâēπρίαô ôî DOS, îâôâēîçèâßôâ ôôîî éâôÛëräî tools/ ôçò FreeBSD áéáîîPò óáô, éâé øÛîôâ áéá ôî äñ÷âßî bootinst.exe. ΆεôâēÝôôâ ôî üðùò ôâßîâôâé ðáñâéÛùù:
 ... \TOOLS> **bootinst.exe boot.bin**

êäé ï äéá÷äéñéóóðð äêêßíçóçð èä äðáíäêäóäóäèäß.

- Íäêéíðóðä íáíÛ ÷ñçóëïðñéðíóäð ðç äéóéÝóä äêêßíçóçð ðïð FreeBSD êäé ðçäáßíäðä óðçí äðéëíäð Custom Installation ðïð ïäñý. ÄðéëÝíðä Partition. ÄðéëÝíðä ðïí íäçäü ðïð êáñííéëÛ èä ðññéäß÷ä ðïí äéá÷äéñéóóðð äêêßíçóçð óäð (ððóëëëíäéëÛ ðïí ðñððï) êäé ïðáí ððÛóäðä óðíí äðáíäñäóðð êäóäðíðóäñí (partition editor) äéä ðïí íäçäü äððü, ðï ðñððï ðñÛäíä ðïð èä èÛíäðä (÷ññðð Ûëëäð äéëääÝð), äßíäé íä äðéëÝíðä (W)rite. ÄðéëÝíðä yes óðçí äðéääääßóç ðïð èä äïðáíéóóððä êäé ïðáí ððÛóäðä óðçí ðñíðñíðð äðéëíäðð Äéá÷äéñéóóðð Äêêßíçóçð, äðéëÝíðä “Boot Manager”. Íä ðïí ðñüðñí äððü ï äéá÷äéñéóóðð äêêßíçóçð èä äñäðäß íáíÛ óðí äßóëï. Ìðñäððä ðññä íä äääððä äðü ðï ïäñý ðçð ääéäöÛöóäóçð êäé íä äðáíäêäéíðóäðä äðü ðï óéëçñü äßóëï, üððð óðíðèðð.
- Íäêéíðóðä íä ðç äñðäéä ðçð äéóéÝóä äêêßíçóçð (ð ðïð CD) ðïð FreeBSD êäé äðéëÝíðä “Fixit” äðü ðï ïäñý. ÄðéëÝíðä äßðä ðç äéóéÝóä Fixit äßðä ðï CD #2 (ðï “live” óýóðçíä äñ÷äßñ) êäé èä äéóÝëëäðä óðí èÝëððïð fixit. ÄéðäéÝóðä Ýðäéðä ðçí äéüëïðç äñðñð:

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

äñðéäéðððíððä ðï *bootdevice* íä ðïí ðñäñíäðéëü óäð äßóëï äêêßíçóçð, üððð ð.÷. *ad0* (ðñððïð äßóëïð IDE), *ad4* (ðñððïð äßóëïð IDE óðíí äñççóéëü äéääéðð), *da0* (ðñððïð äßóëïð SCSI), è.è.ð.

9. Ì ðñçððð ïð ððñëíäéóóððð óäéñÛð IBM Thinkpad, óäéñÛð Á, Ô þ ×, óðáíäöÛäé íä äðññññíäðäé ïðáí ðñíððäèð íä äêééíðóð ðï FreeBSD íäðÛ ðçí ääéäöÛöóäóç. Ððð ìðñð íä äðéýóð äððü ðï ðñüäéçíä;

Ðñüäéðäé äéä Ýíä ðñüäéçíä óðéð äñ÷éëÝð äéäüóäéð ðïð BIOS ðçð ÉÄÌ óðä óðäéäñéñéÝíä ïç÷äíäðä, ðï ïðñíä äíäñññäéä ðçí èäðÛðíçóç ðïð FreeBSD ïð ðéäñü äéäÝñéóíä FAT äéä ðç èäéðññäßä ääñäññðßçóçð óðí äßóëï (suspend-to-disk). Õï BIOS óðáíäöÛäé íä äðññññíäðäé èäèð ðñíððäèäß íä äñäýóäé ðï äéäÝñéóíä ðïð FreeBSD.

Óýíðñíä íä ðçí IBM¹, óä äéüëïðä ïñóÝéä êäé äéäüóäéð BIOS ðññéëäíäÛññíðð óéð äðñäððçððä äéññéðäéð.

ÌñðÝëï	êäñóç BIOS
T20	IYET49WW þ íäðääñÝóðäñí
T21	KZET22WW þ íäðääñÝóðäñí
A20p	IVET62WW þ íäðääñÝóðäñí
A20m	IWET54WW þ íäðääñÝóðäñí
A21p	KYET27WW þ íäðääñÝóðäñí
A21m	KXET24WW þ íäðääñÝóðäñí
A21e	KUET30WW

÷äé äñäðññäß üðé ïäðääñÝóðäññäð äéäüóäéð ðïð BIOS ðçð ÉÄÌ, ßóðð ðäññóóéÛäññíðð íáíÛ ðï ðñüäéçíä. To ïðñíä äððü (<http://docs.FreeBSD.org/cgi/mid.cgi?20010427133759.A71732>) äðü ðïí Jacques Vidrine óðçí çêäðññíéëð èßóðä ðïð FreeBSD äéä ðñççðñýð ððñëíäéóóðð (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile) ðññéäñÛðäé íéä äéääéäððä ç ïðñä ßóðð äññéÝðäé óä íäððäñä ïñóÝéä ðñççðñí ðçð IBM óä ïðñä äñ äêééíñýí êáñííéëÛ ðïð FreeBSD, êäé óðä ïðñä ïðñäððä íä äñääéðððä ðïð BIOS þ êäé íä äðéóðñÝðäðä óä ðññäñýíäíç Ýëäñóç ðïð.

Äí Ý÷äðä ðäéëððäñí BIOS, äéëÛ äñ Ý÷äðä ðçí äðéëíäð íä ðï äñääéðððäðä, Ýíäð ðñüðñí íä íäðñÛóäðä ðï ðñüäéçíä äßíäé íä ääéäðððððäðä ðïð FreeBSD, íä äéëÛíäðä ðïí äñäñññéóóéëü äñéëü (partition id) ðçð èäðÛðíçóçð, êäé íä ääéäðððððäðä ïÝä boot blocks óä ïðñä íä ïðññýíä ÷äéñéóóðñí ðï äéäññäðéëü äñäñññéóóéëü ðçð èäðÛðíçóçð.

Äñ÷éëÛ, èä ðñÝðäé íä äðáíäöÝñäðä ðï ïç÷Ûíçíä óä íéä èäðÛöóäóç þððä íä ðäñíÛäé äðü ðçí äñ÷éëð äéäñññóóéëð ïëñíç. Äððü äðäéðäß íä äñäññññéððäðä ðï ïç÷Ûíçíä ÷ññðð íä ðï äðððäðä íä äñäé ðçí èäðÛðíçóç ðïð FreeBSD óðíí èýññé äßóëï ðïð. Íäð ðñüðñí äßíäé íä äðñäéñýíäðä ðïí óéëçñü äßóëï êäé íä ðïí ïäðäééíðóäðä ðñíðññéñÛ óä Ýíä

ðäëüðäñï ThinkPad (üðüð òï ThinkPad 600) Þ ëÛðïëï äðëðñäðÝäëï ïç÷Ûíçιά, ÷ñçóëïðïëþíðäð òï éäðÛëëçëï éäëþäëï ìäðäðñïðð. Óòï ïç÷Ûíçιά äððü ìðññäðä ðëÝï ìä äëäñÛðäðä ðçï éäðÛðìçóç òïð FreeBSD éäë Ýðäëðä ìä ìäðäëëþðäðä òï ìððëï ðòï äñ÷ëëü ïç÷Ûíçιά. Òï ThinkPad ëä ðñÝðäë ðþñä ìä äëëëíäð éäñíëëÛ.

Ìä òï ïç÷Ûíçιά óä ëäëðïðñäëëþ éäðÛðäóç, ìðññäðä ðþñä ìä ÷ñçóëïðïëþðäðä ðçï äëäëëäðä ðïð ðäñëäñÛðäðä ääþ þððä ìä Ý÷äðä ðäëëëÛ ìä ääëäðÛðäóç ðïð FreeBSD ðïð ìä ëäëðïðñäð.

1. ÊäðäðÛðäðä ðëð äëóëÝðäð boot1 éäë boot2 äðü <http://people.FreeBSD.org/~bmah/ThinkPad/>. Äðïëçëäýðäðä ðä äñ÷äðä äððÛ ëÛðïð ðïð ëä ìðññäðä ìä ðä ìäíäññäðä äñäüðäñä.
2. Ääëäðäðððäðä éäñíëëÛ òï FreeBSD ðòï ThinkPad. Ìçï ÷ñçóëïðïëþðäðä ðçï éäðÛðäóç Dangerously Dedicated. Ìçï ëÛíäðä äðäíäëëþìçóç üðäí ðäëäëþðäë ç ääëäðÛðäóç.
3. ×ñçóëïðïëþðäðä äððä òï “Emergency Holographic Shell” (ALT+F4) äððä òï ëÝëðïðð “fixit”.
4. ×ñçóëïðïëþðäðä ðçï fdisk(8) äëä ìä äëëÛíäðä òï äíäññäðñëððëü äñëëü ðçð éäðÛðìçóç òïð FreeBSD äðü 165 óä 166 (äððüð äðíäë ìä ðýðïð ðïð ÷ñçóëïðïëþðäðä äðü òï OpenBSD).
5. ÄñÛððäðä ðä äñ÷äðä boot1 éäë boot2 ðòï ðïðëëü óýððçιά äñ÷äðä.
6. ×ñçóëïðïëþðäðä òï disklabel(8) äëä ìä äñÛððäðä ðä boot1 éäë boot2 ðòï slice òïð FreeBSD.
disklabel -B -b boot1 -s boot2 ad0sn
Òï n äðíäë ìä äñëëüð ðïð slice ðïð Ý÷äðä ääëäðäðððäðä òï FreeBSD.
7. ÊÛíðä äðäíäëëþìçóç. Óðçï ðñïðñïð äëëþìçóçðä ëä ääððä ðçï äðëëäþ ìä ìäëëþðäðä òï OpenBSD. Óðçï ðñäñäððëëüðçðä, ìä äððü òïð ðñüðï ëä ìäëëþðäðä òï FreeBSD.

Òï ìä ëÛíäðä òï ðäñäðÛí ìä ëäëðïðñäðäë ððçï ðäñðððüçð ðïð ëÝëäðä ìä Ý÷äðä äëðëþ äëëþìçóç OpenBSD éäë FreeBSD ðòïð þäëï òïðçðü ððïëäëðð, òï äðþñïä üð Ûðëçóç ðòïð äíäñþðç.

10. Ìðñþ ìä ëÛíä ääëäðÛðäóç óä Ýíä äððëï ìä ÷äëäðÝíðð ðñäðð;

Ìðññäðä, äëëÛ äðíäë éäëþ ëäÝä.

Äí ääððä ÷äëäðÝíðð ðñäððä óä Ýíä óýä÷ññï ìäçäü IDE, ððÛñ÷äë ìääÛëç ðëëäñüðçðä ìä ìäçäüð äððüð, ðïëý óýíðñä, ìä ððäíäððäë ìä ëäëðïðñäðä äíðäëðð (ìä ìäçäüð ääí Ý÷äë Ûëëðð äíäëäëððëëýðð ëðëþíäññðð ðïð ìä ìðñíýí ìä äíðëëäðäðððòïð ðïðð ÷äëäðÝíðð ìÝðü ðçð äóððäñëëþðð ëäëðïðñäðä äðäíäðððëëþðçóçð ðñï äääñÝíü, ëÛðë òï ìðñþ ðçïäíäëä üðë ìä äððëðð Ý÷äë ðçïäíðëëþðð ðëññÛ). Óäð ðòïëððýíä ìä ääñÛðäðä ëäëñýñëï äððëï.

Äí Ý÷äðä ìäçäü SCSI ìä ÷äëäðÝíðð ðñäððä ääððä äððð ðçï äðÛíðçóç.

11. ÄëÛðñä ðäñþäñää ðñÛäíäðä ðòïääþñïí üðäí äëëëþ ìä ðç äëóëÝðä äëëþìçóçð! Óë ðòïääðäëä;

Äí äëÝðäðä éäðäððÛðäëð üðüð òï ïç÷Ûíçιά ìä ìçï äðñëñþíäðäë Þ ìä ëÛíäë äðäíäëëþìçóç ìññ ðïð üðäí ðñïððäëäððä ìä äëëëþðäðä ìä ðçï äëóëÝðä äëëþìçóçð, ëä ðñÝðäë ìä ëÛíäðä ðòïð ääððü óäð ðñäëð äñüðððäëð:

1. ×ñçóëïðïëþðäðä ëäëñýñëäð, ðñäðëïäëäñññüðÝíäð, éäë ÷ññðð ëÛëç äëóëÝðäð (éäðÛ ðñïðþìçóç éäëþð äðäëñþäð ðëð ìðñäð ìñëëð ääÛëäðä ëäëñýñëäð äðü òï ëïððð ðïðð, óä äíððëäðð ìä ðç äëóëÝðä ðïð ðþñäðä ìäëþ ìä ëÛðïëï ðäñëäëëü éäë ç ìðñä äñððëñðäí ëÛðü äðü òï ëñääÛðë ðä ðñþä ðäëäððäðä ÷ññäëä);
2. ÊäðäðÛðäðä òï image ðçð äëóëÝðäð ìä äðääëëü (binary Þ image) ðñüðï ìäðäññÛðð; (ìç ìðñÝðäððä, äëñüä éäë ìë ëäëýðäñíë äðü ìäð Ý÷ðï éäðÛ ëÛëðð éäðäðÛðäë ëÛðïëï äðääëëü äñ÷äððä óä éäðÛðäóç ASCII (ëäëÝíñ), ðïðëÛ÷ëðð ìäð òïðñ!)

3. $\acute{\alpha}\iota \div \eta\tau\omicron\sigma\epsilon\iota\eta\iota\delta\epsilon\iota\alpha\beta\omicron\alpha$ Windows 95 **η** 98, $\acute{\alpha}\epsilon\omicron\alpha\epsilon\Upsilon\omicron\omicron\alpha$ $\delta\iota$ fdimage **η** $\delta\iota$ rawrite $\omicron\alpha$ $\epsilon\acute{\alpha}\epsilon\alpha\eta$ **η** $\epsilon\acute{\alpha}\omicron\Upsilon\omicron\omicron\alpha\acute{\omicron}\varsigma$ DOS; $\omicron\alpha$ $\epsilon\acute{\alpha}\epsilon\omicron\iota\eta\eta\alpha\epsilon\Upsilon\Upsilon$ $\Upsilon\omicron\delta\omicron\Upsilon$ $\iota\delta\iota\eta\eta\alpha$ $\iota\alpha$ $\delta\alpha\eta\alpha\iota\alpha\epsilon\chi\epsilon\iota\Upsilon\iota$ $\omicron\alpha$ $\delta\eta\eta\alpha\eta\Upsilon\iota\iota\alpha\acute{\omicron}\alpha$ $\omicron\alpha$ $\iota\delta\iota\beta\alpha$ $\alpha\eta\Upsilon\omicron\iota\omicron\iota$ $\alpha\delta\alpha\omicron\epsilon\alpha\beta\alpha\delta$ $\omicron\delta\iota$ $\sigma\epsilon\epsilon\epsilon\Upsilon$, $\epsilon\Upsilon\omicron\epsilon$ $\delta\iota$ $\iota\delta\iota\beta\iota$ $\omicron\iota\iota\alpha\beta\iota\alpha\epsilon$ $\epsilon\acute{\alpha}\epsilon$ $\iota\alpha$ $\omicron\alpha$ $\delta\eta\eta\alpha\eta\Upsilon\iota\iota\alpha\acute{\omicron}\alpha$ $\acute{\alpha}\chi\iota\epsilon\iota\eta\eta\alpha\beta\alpha\delta$ $\omicron\upsilon\iota$ $\acute{\alpha}\epsilon\omicron\epsilon\alpha\delta\iota$. $\iota\delta\iota\eta\eta\alpha$ $\iota\alpha$ $\acute{\alpha}\chi\iota\epsilon\iota\eta\eta\alpha\epsilon\beta$ $\delta\eta\upsilon\alpha\epsilon\chi\iota\alpha$, $\acute{\alpha}\epsilon\upsilon\iota\alpha$ $\epsilon\acute{\alpha}\epsilon$ $\acute{\alpha}\iota$ $\omicron\alpha$ $\acute{\alpha}\epsilon\omicron\alpha\epsilon\alpha\beta\omicron\alpha$ $\omicron\alpha$ $\delta\alpha\eta\Upsilon\omicron\delta\eta\eta$ DOS $\iota\Upsilon\omicron\alpha$ $\alpha\delta\upsilon$ $\delta\iota$ $\alpha\eta\alpha\acute{\omicron}\epsilon\Upsilon$ $\delta\alpha\eta\acute{\epsilon}\alpha\Upsilon\epsilon\iota\iota$.

÷ òì ãððçð áíáðãññ ðñèðððáéð ùðì òì Netscape® àçìéìññáá ðñíáéðíáðá òì éáðÝááðíá òçð áéóéÝðáð ãééððçðð, Ýðóé áííáé éáýððñá íá ÷ ñçóéìðíðéððáðá èÜðìéì Üééì ðññáññá FTP, áí áððì áííáé áðíáðì.

[illegible]

Ñõeĩḃoá íáiŮ òĩ óýòçĩa óáo, ḃóá òĩ CDROM áḃá íá áḃíáé ç master óóœâðḃ óôĩ äēāēðḃ ðĩō áḃíáé óõĩāĩÝíc, ḃ
ääāēùēāḃá uôē áḃíáé slave óá Ýía äēāēðḃ IDE ĩ řĩḃĩð ũũò Ý÷äē Üēēç óóœâðḃ óõĩāĩÝíc uò master.

13. Ιδιῶς ἰὰ ἀὰἐὰοάοδῖου οἱ FreeBSD οἱ ὀϊῆς οἱ ἰῖδ ÷ ἡςοέἰῖδῖἐῖῖοάδ PLIP (Parallel Line IP, IP ἰῖοῦ Δάνῦἔἔἔἔἔ
ἔῖῖῖῖῖ);

Íáε. ×ñçóéññíεΠóóá ôôðñíεççíÝñî éáεpäéí ôýðñō Laplink. Áí ÷ñāéÜæāóáé, áéááÜóóā ôñ òñPiá PLIP ôñō Å÷-áñéāBiō (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/network-plip.html) áéá éāðññÝñāéāó ó÷-āóééÝð ñā áééôýúóç ñÝóñ ðāñÜëçççð éýñāð.

14. Ôé ãåùìåõñßá íá ÷ñçóéìïðíéÞòù ãéá ôì óêçñü ìï ãßóêì;

Όçιαßυόç: Ιὰ οἱ ὑνὶ “ἁἰὺὰδὸνßá”, αἰνὶγία οἱ ἁνέειῦ οὐί ἐὼεßίανῦι, ἐὰοἀερί ἐαé οἱ Ὑνὶ ἀνὶ ðñ÷εῦ αἰὺὸ ἁßὸεῖθ. Ἀέα ἀὸερεßá, ἐὰ ἀίῶαἵνῦαῖῶῶα ὀοίρ ὑνὶ ἁδὸῦ ὑδ C/H/S (Cylinders / Heads / Sectors). Ἀδὸῦδ ἁßιάé ἐαé ἱ ὀνῦθῖθ ἰὰ οἱρ ἱθῖßῖ οἱ BIOS ἁνßὸéἁé ὁᾶ θῖεᾶ δᾶñé÷÷ ὀῖθ ἁßὸεῖθ ἱᾶ ἁñῚῶᾶé.

Οι δαναδὺν δῆτεαῖς ὅγα÷δογ ὁδοῖ ἰγῖτον ἀεά÷ἀεῖεοδὺδ ὁδοδὺδὺν. Ἐαόαν÷Πί, ζ *ὁδοεῖP* ἀαὺλαὸνβά ἀρὺδ ἱαγῖγ SCSI ἀβίαε ὁδῖτεῖῦ Ὑο÷ἀογ, ἀοῖγ οἱ FreeBSD ἐαεῖοῖναῖς ἰα ἀῦογ ὁα ἰδεῖε ἀβόεῖο. Ὀογῖ δῆαἰαόεῖουδῶ, ἀαῖ δδῦν÷ἀε ἐαῖ “ἀεγῖεP” ὁδοεῖP ἀαὺλαὸνβά, ἐαεῖδ ζ δδῖεῖουδῶ οῖν ὦνῦῖ ἰαδῶῦῖεαδῶα ἀδὺ δῆῖε÷P ὁα δῆῖε÷P ὁῖ ἀβόεῖο. Ἀοδὺ δῖο ἱε ἐαδῶεῖαδῶοδὺδ ἀἰαδῦῖοῖ ὁδ÷ῖῦ ὦδ “ὁδοεῖP ἀαὺλαὸνβά” ἀβίαε ὁῖPεῖδ ζ ἀαὺλαὸνβά ἰα ὁγῖ ὦῖεῖ ῖ÷ῖοῖ ἀἰεῖεῖῖαε ὦδ ἀβἰαδῶε ζ ἱεῖνὺδῶνγ ὁδῶῖεγ ἀεῖγῖαῖο ÷ῖοῖ. Ἀεῖ ἀβόεῖοδ IDE, οἱ FreeBSD ÷ῖγῖοῖῖεῖς δῖῦαἰαε οἱ C/H/S, ἀεῖῦ ὦεῖε ἱε ἱῖοῖῖῖε ἱαγῖῖ ἰαδῶδῖῖοῖ ἀοὺδῶῖεῖῦ ἀοδὺδ ὁεδ ἀἰαῖνῖῖ ὁα ἀἰδῖοῖε÷ῖ ἰδεῖε.

[illegible]

Ἀέα ἀβῶεῖρὸς SCSI, ἡ ἀαυῖαὸνβᾶ ὄτῳ δὴ ὕδαε ἰά ÷ ἡῖσῶεῖρὸς ἡεῖαβ, ἀῖαῖνὸ ὕδαε αὐῷ ὅτ ἀῖ ἀβῖαε ἀῖαῖαῖρὸς ἡῖς ἡ ὀδῖρὸδβῖεῖς ἡεὸαὸαῖ ἡῖς ἡ ἰᾶ ὕῖαῖρὸς (αὐὸα ὀ ÷ ἰ ὕ ἰᾶ ὕ ἡᾶδαε ὕ ὀ “ὀδῖρὸδβῖεῖς ἡέα ἀβῶεῖρὸς DOS > 1GB” β ε ὕ ὀε αῖ ὀβῶεῖ ÷ ἰ). Ἀῖ ἀβῖαε ἀδᾶῖαῖαῖρὸς ἡῖς ἡ ÷ ἡῖσῶεῖρὸς ἡεὸα *N* ἐὸεβῖαῖρὸς, 64 ἐᾶὀε ὕ ὀε 32 ὠᾶβδ/ὀῖ ÷ ε ὕ, ὕδῖ ὀ *N* ἀβῖαε ἡ ÷ ὕῖσῶεῖ ὕ ὀα ὅτ ἀβῶεῖ ὀα ἰᾶ. Ἀέα δᾶν ὕ ἡεᾶεᾶῖα, ἡέα ὕ ἰά ἀβῶεῖ 2GB, ἐα δὴ ὕ δᾶε ἰά ὕ ÷ ᾶὸα 2048 ἐὸεβῖαῖρὸς, 64 ἐᾶὀε ὕ ὀε 32 ὠᾶβδ/ὀῖ ÷ ε ὕ.

Άí *áβταίε* áñáñáññíεçíÝíç (έάε 00÷÷íŮ ðáñÝ÷÷άάε Ýόόé þόόά íá íáðñáñíεýíóáé εŮðñέíε ðáñέíñéοíñþ όόí MS-DOS®) έάé ç ÷ññçόέέúδúçόá όíñ áβόέíñ áβταίε íááάέýðáñç áðu 1GB, ÷ñçόέíññíεþόόά M έέεβñáññíòð, 63 òñáβð áíŮ ðñí÷έŮ (*ü÷έ* 64) έάé 255 έάόάέÝð, üðñ M áβταίε ç ÷ññçόέέúδúçόá όíñ áβόέíñ όά MB, áέάέñáñÝíç íá όí 7.844238 (!). þόέ, όόí ðáñŮáάέáíá íáð, í áβόέíñ ðúí 2GB έá áβ÷á 261 έέεβñáññíòð, 63 òñáβð áíŮ ðñí÷έŮ έάé 255 έάόάέÝð.

Άν ήάρ ήβρώα όββατόντ ήέα όι όήναδΰΰ, ϐ ήί όι FreeBSD ήήόγ÷ήε όόγρ ήήβ÷ήήόό όόό όούόόό άάήήήήήήή ήήόΰ όόρ ήήέήόΰόόόό, ή ήήήήόόήήή όήήήή ήέα ήά όι όήήήήήήήήή, ήήήή όόήήήή ήά ήήήήήήήήήήή ήή ήήήή ήήήήήήήήήήή DOS όόή ήβόή. Όι BIOS ήήήήή ήά ήή÷ήήήήή όό όούόόό άάήήήήή, ήέ ήήήήήήή ήΰήή ήήήή ήήήήήήήήήήή όι ήήήήήήήήή DOS ήΰή ήή όι ήήήήήήήήή ήήήήήήήήή, ήή ή ήήήή ήά όι ήήήήήήήήή. ήήήή ήόόόή ή ήήήήήή ήά όι ήήήήήή÷ή, ήήήήήή.

[illegible]

15. ÕðÛñ÷ïóí ēÛðíéíē ðãñéíñéóííß óóí ðuò ðñÝðáē íá ÷ùñßóú ôí äßóéî;

Ίάε. ÐñÝðáε íá ââââεùεâßðâ ùεε ç ñέæεêP (root) êâðÛðìççç âñßóεâðáε êÛðù áðu òïðò 1024 êðεßíâñïð, þóðâ òï BIOS íá ìðñâß íá âêêérPðáε òïï ðññPíá áðu áððP. (Ôçíâεþóðâ ùεε áððùð âßíáε Ýíáð ðñâñéíñéóíùð óòï BIOS òïð PC, êáε ù÷÷ε óòï FreeBSD).

Ἡ ἑξῆς ἔκδοσις SCSI, ὅτι πρὸς αὐτὴν ὁρίζεται ὡς ἡ ἐκδοσις εἰς τὴν ὁποία 1024MB (ἢ ὅσα ἑξῆς 4096ΓΑ αἰ - ἡ ὁποία ὁρίζεται ὡς ἡ ἐκδοσις ἡ ὁποία ὁρίζεται - ἡ ἐκδοσις ὁρίζεται ὡς ἡ ἐκδοσις ὁρίζεται). Ἡ ἑξῆς ἔκδοσις IDE, ὅτι αἰ ὁρίζεται ὡς ἡ ἐκδοσις ὁρίζεται ὡς 504MB.

16. Ἀβιάε οὐίᾱάου ὀί FreeBSD ἰᾱ ḏñĩᾱñŨἰᾱάᾱ ᾱέᾱ÷ᾱβñέόçᾱ ᾱḱᾱᾱἰ (disk managers);

To FreeBSD áíááíũñßæíæ éáé äðéôñÝðáé ôç ÷ ñßóç ðíð Ontrack Disk Manager. Äáí ððíóóçñßæííðóáé Üëëíë äéá ÷ äéñéóóÝð äßóéũí.

Άι εΥέαοα αδερο ία ÷ ñçóέιññέPóαόα οι αβóει ία οι FreeBSD, αάρ ÷ ñάέÜæάόα áέá ÷ áέñέóóP αβóειò. ΆδεPò ñóειPóόα οι αβóει áέá υóι ðñάέóóóóññ ÷ ðññ ίðññá ίá áάέ οι BIOS (óóPεύò 504ÎÁ), έάέ οι FreeBSD έá áíάέáýPάέ ðυóι áεáýεaññ ÷ ðññ Ý ÷ άόα óóçι ðñάñáóέέυóçóá. Άί ÷ ñçóέιññέάβóα έÜðíέí ðάέέυ αβóει óá áεάάέð MFM, βóóð ίá ðñÝðáέ ίá ðáβóá óóι FreeBSD ðυóιòð έðέβñññòð ίá ÷ ñçóέιññέPóáέ.

Αί εΥέαθα ίά ÷ ñçóεíñðíεΠόαθα οί αΒόεí ουοί ία οί FreeBSD υοί έάε ία εÜðíεí Üεεí εάεοíτναεεü ογόοçía, έά ðñÝðáε ίά ίðñáΒόα ίά οί εÜíτáα ÷ ññΒò áεά ÷ áεñεóóΠ αΒόεíτ: áðεðò ááááεüεαΒόα υόε οί áεáíÝñεοía áεεΒίççóçò οίτ FreeBSD έáεðò έάε ç έαóÜοιççόç οίτ Üεεíτ εάεοíτναεεíý óóóðΠáοíτò áñΒόεíτáε ίÝóά óοίτò ðñðίτò 1024 έóεΒíññίτò. Αί αΒόόα áñεáóÜ ðñíτáεóεεüð. Ýíá áεáíÝñεοía áεεΒίççóçò (boot) ίááÝείτò 20MB έá áΒίáε áñεáóü.

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

[illegible]

18. Äëäöð ääí ïðñþ íá öóíá÷ßöü ðÝñá áðu öçí ðññññðP F? öïð äëá÷äëñéöðP äëëßíçöçð;

Áðöü áßíáë Ûëëí Ýíá öýíðöüíá öïð ðññäëßíáöïð ðïð ðññëññÛöäóäë öðçí ðññçäýííáíç ãñþðçöç. Ääí öðíðððäë ç ääüíäöñßá öïð BIOS íá áððP öïð FreeBSD! Áí í äëäëðððP P öï BIOS óáð ððíóðçñßæäë íäðÛñäóç ëðëßñññí (öð÷íÛ áíáöÝññäóäë ùð ">1GB drive support"), äñëëíÛöðá íá äëëÛíäðá áððP öç öëíP ëáë íá äðáíäëäóäóðððäð öï FreeBSD.

19. ÐñÝðäë íá ääëäóäóðððöü ùëë öñí ðçäáßí ëþäëä;

ÄäíëëÛ, ù÷ë. Ûöðüöí óáð öðíëöðíýíá íá ääëäóäóðððäð, ùð äëÛ÷ëöðí, öñí ðçäáßí ëþäëä öçð äëáññðð base, í ïðíßìð ðññëëññÛíáë äñëäðÛ áðu óá äñ÷äßá ðïð áíáöÝññíóäë äþ, ëäëðð ëáë öñí ðçäáßí ëþäëä öçð äëáññðð sys, ç ïðíßá ðññëëññÛíáë öñí ðçäáßí ëþäëä öïð ððñPíá. Ääí öðÛñ÷äë ùöðüöí ëÛðë öðí öýóðçíá öï ïðíßí íá äðäëððá öçí ýðáññç öïð ðçäáßí ëþäëä äëá íá ëäëðíðñáðäë, äëöüð áðu öï ðññäññíá ñýëíëöçð ððñPíá config(8). Íá äñáßñäóç öñí ðçäáßí ëþäëä öïð ððñPíá, ç äñP öïð öðóðßíáöïð íäðäëëðððëöçð íäð áßíáë öÝöíëá, þóðá ïðññáððá ðÛíóá íá ðññóáññððäð öï äÝíðñí ðçäáßí ëþäëä íÝöü NFS (ëáë íá äëëäëþíäðá ïññí áíÛáíöçð) ëáë ðÛëë íá ïðññáððá íá äçíëíðñáððá íÝá äëðäëÝóëíá (ëüäü öïð ðññëññëöñíý öïð öðÛñ÷äë öñí ðçäáßí ëþäëä öïð ððñPíá, óáð öðíëöðíýíá íá íçí ëÛíäðá öçí ðññíöñðçöç äðäðäððáð öññí ëáðÛëññí /usr/src, äëëÛ íá ðçöçíððíëððäðá ëÛðíëá Ûëëç öñðíëäðá ëáë íá äçíëíðñáððá öïðð ëáðÛëëçññöð öññíëëçññöð ääöñíýð öïð íá áíðëññÛöñí öç äñP öçð ëáíðñëëðð ëáññ÷äðð öïð äÝíðññí ðçäáßí ëþäëä).

Áí Ý÷äðá Ûíäðá äëäëÝöçí öñí ðçäáßí ëþäëä, ëáë äñññßæäðð ðüð íá íäðäëëöððððäðá Ýíá ðññëëçññí öýóðçíá áðu áðöñí, ëá äëäðëññëäððá ðÛñá ðñëý ùðáí áíáäëëßßæäðð öï öýóðçíá óáð óá íäëëíðëëÝð äëäüöäëð öïð FreeBSD.

Äëá íá äðëëÝíäðá Ýíá öðíóýññë öïð ðçäáßí ëþäëä, áðu öçí äðëëñP Distributions öïð äñääëäððí ääëäöÛöóäóçð öðóðßíáöïð, äðëëÝíðá öï ïññý Custom.

20. ×ñäëÛæäðäë íá öðëÛñ ðññóáñññíöíÝñ ððñPíá;

Ç äçíëíðñáßá áñüð íÝíð ððñPíá þóáí äñ÷ëëÛ ó÷ääññ öðí÷ñäüðëü äþíá óá íëá ääëäöÛöóäóç FreeBSD, äëëÛ íë ðëí ðññöðäðð äëäüöäëð Ý÷íñ ùðäëçëäß áðu öçí äëóäññäP äëöçðÛ öëëëëüðñññí ðñññññÛññ ñýëíëöçð öïð ððñPíá. Áðu öï FreeBSD 5.X ëáë íäðÛ, áßíáë äñëäðÛ äýëñë íá ððëíððäðá öñí ððñPíá ðçöçíððíëððíäðð öï ðñëý ðëí äðÝëëöðí öýóðçíá öññ "hints" óá ïðíßá ïðññáððá íá ððëíððäðá öðçí ðñññññð öïð loader.

Áíäð÷ñÝñüð íá áíßæäë äëüíá íá äçíëíðñáððá Ýíá íÝí ððñPíá í ïðíßìð íá ðññëÝ÷äë ïññí óá ðññññÛíäðá íäþäçöçð öïð ðñäëÛæäðð, äëá íá äëöðððäðá ëÛðíëá íëñP ðññöðçðá ííþçð RAM, äëëÛ áðöñ ääí áßíáë ðëÝñí äðññáððöðí äëá óá ðññëöóüðññá öðóðßíáöïð.

21. Äëá öïðð ëüäëçññöð ðññöääóçð öñí ðçöçðþí, íá ðçöçíððíëðððð DES, Blowfish, P MD5 ëüäëëññíßçöç, ëáë ðüð ëá ëäëñßöüð öë ëá ðçöçíððíëçññí íë ðñððäð ïñ;

Ç ðñññðëëäñÝíç ïññP ìñððññññÛöçöçð äëá ëüäëçññöð öñí FreeBSD áßíáë öï MD5. Ç ääíëëP áíðßçççç áßíáë ùðë ðáñÝ÷íñ ëäëýðññç áóöÛëäëá óá ó÷Ýöç íá öçí ðáñäññíëäëP ïññP öïð UNIX öïð äáðßæäðäë öññí äëäññëëñ DES. Íë ëüäëëñ DES áßíáë äëüíá äëäëÝöçíë, áí ðñäëÛæäðäë íá äëäñññÛöðá öï äñ÷äßí öñí ëüäëëþí óáð íá ðäëëüðññá ëäëðíðñäëÛ öðóðßíáöïð, óá ïðíßá ðçöçíððíëçññí äëüíá öï ðäëëüðññí ëáë ëëäüðññí áóöäëÝð öýóðçíá (áßíáë äëäëÝöçíá áí ääëäóäóðððäð öç äëáññP "crypto" íÝöü öïð sysinstall P ääëäëöððíäðð öñí áíðßöðíë÷í ðçäáßí ëþäëä áí ëÛíäðá ääëäöÛöóäóç íÝöü ðçäáßí ëþäëä). Áí ääëäóäóðððäðððð öëð äëäëëëëPäð crypto ëá ïðñÝöðäð äððöçð íá ðçöçíððíëððäð äñðððñññÛöç Blowfish ç ïðíßá áßíáë äëüíá ðëí áóöäëðð. Öï ðíëá ïññP ëüäëëþí ðçöçíððíëäðäë äëá öïðð íÝíðð ëüäëçññöð, äëÝá÷äðäë áðu öçí äñíäüðçðá äëöñññ "passwd_format" öñí /etc/login.conf, öï ïðíßí ðáßñíäë öëð öëÝöð "des", "blf" (áí áßíáë äëäëÝöçíç) P "md5". Ääððá öç óäëßää manual öïð login.conf(5) äëá ðññëöóüðññäð ðççñññññäð ó÷äðëëÛ íá öëð äñíäüðçððð äëöñññ.

22. Äëäöð áñ ç äëëéÝóä äëëßíçóçð íäëéíÛäë éáñíëÛ, êñàíÛäë óççí ðëííç Probing Devices...;

Áí Ý÷äöä ääëäöäóçíÝí ðäçäü IDE Zip® P Jaz®, äóäéíÝóää öíí éäë íáíäðñíóðäëðóää. Ç äëëéÝóä äëëßíçóçð ððñäð íä ððñäðäðäð äðü äðóíðð öíðð ðäçäíðð. ÌäðÛ óççí ääëäöÛóóäóç öíð óðóððíäðð, ððñäðäð íä íáíäðñíóðää öíí ðäçäü. Äðäëëëóóíðä üðë öí ðñüäëçíä äðü ää äëñëüëäð óä äðüäíç Ýëäíóç.

23. Äëäöð ðäðñíü öí ððíäíä ëÛëíðð panic: can't mount root üðäí äëëëñ öí óýóóçíä äëä ðñðç öíñÛ ÌäðÛ óççí ääëäöÛóóäóç;

Öí óðÛëíä äðüð ðñíÝñ÷äðäé äðü óççí óýä÷ðóç ðíð ðñíëäëäðäé áíäëððäð öíð äëäöíñäðëëéý ðñüðíð Ìä öíí ððñí öí BIOS éäë öí boot block áíðëëäíäÛííðäé öíðð óëëçñíýð äðóëíðð. Öí ðñüäëçíä óñíðëüð äíðáíðäðäé óä óðóððíäðä Ìä äýí äðóëíðð IDE, äëäëëÛ üðäí íë äðóëíë äðíäé master (P ðüñíë öíðð) í éäëÝíäð óðí äëëü öíð äëääëðP IDE éäë Ìä öí FreeBSD íä äðíäé ääëäöäóçíÝíð óðí äðóëí ðíð äñðóëäðäé óðí ääððäñäýííðä äëääëðP. Öí boot block ðñðäë üðë öí óýóóçíä äðíäé ääëäöäóçíÝíð óðí ad0 (öí ääýðäñí äðóëí öíð BIOS) áñ í ððñðíäð áíäëÝóää öíí ðñððí äðóëí óðí ääððäñäýííðä äëääëðP, ad2. ÌäðÛ óççí áíð÷íäðóç ðüí óðóëäððí, í ððñðíäð ðñíóðäëäð Ìä ðñíóäñððäé äðüð ðíð öí boot block ðëóðäýäë üðë äðíäé í äðóëí äëëßíçóçð, ad0 áñ óççí ðñäñäíäðëëüðçðä äðíäé í ad2 éäë óðóëëÛ äðíððä÷Ûíäë.

Äëä íä äëñëððäðä öí ðñüäëçíä, ëÛíðä Ýíä äðü óä ðäñäëÛðü:

1. Äðäíäëëëíðóää öí óýóóçíä éäë ðëÝóää **Enter** óççí ðñíðñíðð Booting kernel in 10 seconds; hit [Enter] to interrupt. Ìä öíí ðñüðí äðüð ää ääðäðä óðí ðñüäñäíä öíð öíñðüðð äëëßíçóçð.

Êäöððëí äñÛððä set root_disk_unit="disk_number" . Öí disk_number éä äðíäé 0 áí öí FreeBSD äðíäé ääëäöäóçíÝíð óðí master äðóëí öíð ðñððíð äëääëðP IDE, 1 áí äðíäé ääëäöäóçíÝíð óðí slave äðóëí öíð ðñððíð äëääëðP, 2 áí äðíäé ääëäöäóçíÝíð óðí master äðóëí öíð ääððäñäýííðð IDE éáíäëéý éäë öÝëíð, 3 áí äðíäé ääëäöäóçíÝíð óðí slave äðóëí öíð ääððäñäýííðð IDE éáíäëéý.

ðäëðä äñÛððä boot, éäë öí óýóóçíä óäð ää ðñÝðäë íä äëëëíðäë éáñíëÛ.

Äëä íä ëÛíäðä ðüíëíç äððP óççí äëëäð (ðóää íä íçí ÷ñäëÛäðäé íä ëÛíäðä öí ðäñäðÛñ ëÛëä öíñÛ ðíð äðäíäëëëíäðä P áíäñäíðíäðä öí FreeBSD íç÷Ûíçíä óäð), äÛëðä óç äñäñP root_disk_unit="disk_number" óðí äñ÷äðñí /boot/loader.conf.local.

2. Ìäðäëëíðóää öí äðóëí öíð FreeBSD óðíí ðñüðäýííðä äëääëðP IDE, ðóää íë óëëçñíð äðóëíë íä äðíäé óðíä÷üäíë.

24. Ðíëä äðíäé óä ðñëä óçð ððíçð;

Öí ðñëí äðíäé óä 4 gigabytes óä íëä óðíçëëóíÝíç ääëäöÛóóäóç óä äñ÷ëðäëðíëëP 1386. Ìäëëíðäð äðü óëð äëäüóäëð FreeBSD 4.9 éäë 5.1, ððíóççñðäðäé éäë ðäñëóóüðäñç ððíç ðÝóü öíð pae(4). ×ñäëÛäðäé üððüóí íä Ìäðäëëüððëóðäð íáíÛ í ððñðíäð ðäñëëäíäÛííðäð éäë íëä Ýíðñä äðëëíäP äëä óççí áíäñäíðíðçóç öíð PAE:

options PAE

Öí FreeBSD/pc98 Ý÷äë ðñëí óä 4 GB ððíçð, éäë ääí ððñäð íä ÷ñçóëíðíëçëäð PAE óä äððP óççí äñ÷ëðäëðíëëP. Öí FreeBSD/alpha, öí ðñëí óçð ððíçð äíäñðÛðäé äðü öíí óýðí öíð óëëéý ðíð ÷ñçóëíðíëäðäé - äëä ääððñÝñäëä äððä óëð Óçíäëððäëð, ëäíóçð Öëëéý äëä Alpha. ¶ëëäð äñ÷ëðäëðíëëÝð ðíð ððíóççñðäñíðäé äðü öí FreeBSD, Ý÷íðí äñëäðÛ Ìäääëýðäñä ääñçðëëÛ ðñëä ó÷äëëÛ Ìä óç ðÝäëóóç ðíóüðçðä ððíçð (ðíëëÛ terabytes).

25. Ðíéá áßíáé óá ùñéá ôíð óóóóÐíáðíð áñ÷åßùí ffs;

Ãéá óóóóÐíáðá áñ÷åßùí ffs, ôí ìÝäéóóí èäùñçðéèù ùñéí áßíáé óá 8 terabytes (2G blocks), Ð 16ÔÂ äéá ðñíáðééääíÝí ìÝääèð block ðùí 8Ê. Óðçí ðñáñíáðééèùðçðá, ððÛñ÷åß Ýíá áñ÷éèù ùñéí 1 terabyte, äèèÛ ìá èÛðíéäð ìáðáðñðÝð, áßíáé äñíáðùí íá àçíéíðñäçèíýí (éáé ððÛñ÷íð) óóóóÐíáðá áñ÷åßùí ìääÝèðð 4 terabytes.

Ôí ìÝäéóóí ìÝääèð áíüð áñ÷åßùí óá Ýíá óýóðçíá ffs áßíáé ðññððíð 1G blocks, Ð 4TB ìá ìÝääèð block ðùí 4K.

Ðßíáéäð 3-1. ÌÝäéóóá ìääÝèç áñ÷åßùí

ÌÝääèð block fs	èäéðíðñäß	ðñÝðáé íá èäéðíðñäß
4K	4T-1	>4T
8K	>32G	32T-1
16K	>128G	32T-1
32K	>512G	64T-1
64K	>2048G	128T-1

¼ðáí ôí ìÝääèð block ôíð fs áßíáé 4K, èäéðíðñäýí óá ðñéðèÛ Ýíáðá blocks (triple indirect blocks) éáé óá ðÛíðá éá Ýðñáðá íá ðñéíñßæíðáé ìüí áðü ôí ìÝäéóóí áñéèí block ðíð ìðñäß íá áíáðñáóóäèäß ìá ðç ðñðç ðñéðèß Ýíáðùí blocks (ðññððíð 1K^3 + 1K^2 + 1K), äèèÛ ðäèèèÛ ðñéíñéóíüð ðäßèäáé óá Ýíá (èÛèð) ùñéí 1G-1 óðíðð áñéèíýð ðùí blocks. Ôí ùñéí óðíðð áñéèíýð ðùí block éá Ýðñáðá íá áßíáé 2G-1. ððÛñ÷íð èÛðíéá ðñíäèÐíáðá üðáí íé áñéèíß ðùí block ôíð fs ðèçóéÛæíðí ôí 2G-1, äèèÛ ðÝðíéé áñéèíß block äáí ìðñíýí íá ðñíóääéóðíýí üðáí ôí ìÝääèð block fs áßíáé 4Ê.

Ãéá ìääÝèç block 8Ê éáé ìääéýðñá, óá ðÛíðá éá Ýðñáðá íá ðñéíñßæíðáé áðü ôí ùñéí 2G-1 óðíðð áñéèíýð ðùí block ôíð fs, äèèÛ óðçí ðñáñíáðééèùðçðá ðñéíñéóíüð ðäßèäáé óðí èÛèð ùñéí 1G-1 óðíðð áñéèíýð ðùí block ôíð fs. H ðñðç ôíð óüóðíý ìñßð ðùí 2G-1 blocks, àçíéíðñäß ðññáíáðé ðñíäèÐíáðá.

26. Ãéáðß ðáßññü ôí ìÐíðíá èÛèð, archsw.readin.failed ìáðÛ ðçí ìáðáæðððéóç éáé äèèßçóç íÝíð ððñÐíá; Ãéáðß ððñÐíáð óáð éáé ôí ððüèíéðíð ôíð ááóééý óóóóÐíáðíð (world) áßíáé äèðüð óðá÷ñíéóíý. Ç èäéðíðñäß óá áððð ççí éáðÛóóäóç äáí ððíðçñßæíðáé. Åääéèèäßðá üðé ðñçóéíðíéäðá ðéð áíðíèÝð make buildworld éáé make buildkernel äéá íá áíáäèíßðáðá ôíð ððñÐíá óáð. ððñäðá íá äèèíðóáðá ìñßæíðáð ôíð ððñÐíá áðäèèäßðð áðü ôí ääýðññí óðÛæí, ðéÝæíðáð ððíðððíðð ðèðèðñí ìüèð ääðá ôí ì éáé ðñéí íäèíðóáé í loader.

27. Ç ääéäöÛóóäóç éáðáññÝäé éáðÛ ðçí äèèßçóç. Ôé ìðñð íá èÛí; ÄíèèÛóðá íá áðñáññáíðíèèðóáðá ðçí ððíððñéíç ACPI. ìüèð íäèíðóáé ðññðððð äèèßçóçð, ðéÝóðá ôí ðèðèðñí space. To óýóðçíá óáð éá äíðáíßóáé
OK
. ÅñÛððá
unset acpi_load
éáé éáðüðéí

boot

.

Óçìâépóâéò

1. Óâ Ýíá e-mail áðü ôïï Keith Frechette <kfrechet@us.ibm.com>.

ΕὰοÛεάει 4

ΌϊάάôÛôçôά Όëéêÿ

4.1 ΆαίέÛ

1. Έÿÿù íá ááñÛóù ðëëëù áéá ôï FreeBSD óýóðçíá ïñ. Ðñéï ïïôÿÿéï / ïÛñëá / óýðñò áßíáé ôï éáëýðáññï;

ΌðÛñ÷ïñí óñíÿ÷áéá óðæçðßóáéð áéá ôï èÿíá áððù óðéð èßóðáð çäëðññíëéëÿ óá÷ðáññáßñò ðñò FreeBSD. Áððù ùóððùí áßíáé áíáíáññáñ, éáèðð ðï ðëëëù ðñí ððñéáéóððí áëëÛæáé ðñëý áñßáñá. Áíáðð áñáëññðñéÿíá íá óðíéóðñíÿ íá ÿñóáóç, íá áéááÛóáððá ðéð Óçíáéßóáéð Όëéëÿ ðñò FreeBSD 8.2

(<http://www.FreeBSD.org/releases/8.2R/hardware.html>) Þ 7.4

(<http://www.FreeBSD.org/releases/7.4R/hardware.html>) éáé íá øÛñáððá óðá áñ÷áßá

(<http://www.FreeBSD.org/search/#mailinglists>) ðñí ééóððí çäëðññíëéëÿ óá÷ðáññáßñò, ðñéí áñ÷ßóáððá íá ñùðÛðá ó÷áðéëÛ íá ðï ðáëáððáßí éáé éáëýðáññ ðëëëù. Áßíáé áñëáðÛ ðëéáñí íá áéáðéóððóáððá, ùðé áéá ôï ðëëëù ðñò áíáæçðÛðá, ððßññá ó÷áðéëß óðæßðçóç ïñëéð ðñéí íéá áááñÛáá.

Áí øÛ÷íáððá áéá ðñççðù ððñéáéóðð, áëÿáñðá óá áñ÷áßá ððð èßóðáð çäëðññíëéëÿ óá÷ðáññáßñò freebsd-mobile (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile>). ÁéáðññáðéëÛ, ïÛëëï éá èÿëáððá íá ááßðá óá áñ÷áßá óçð freebsd-questions (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>) Þ ðëéáñí íéá èßóðá ðñò íá áñáéáëëÿáðáé óðñí óýðñ ðñò ðëëëÿ ðñò øÛ÷íáððá.

4.2 Άñ÷éðáëðñíëéÿð éáé Άðáññááóðÿð

1. Όðñóçñßæáé ôï FreeBSD áñ÷éðáëðñíëéÿð áéáðññáðéëÿð áðù óçí x86;

Íáé. Áððß óç óðéáñ ðï FreeBSD ïðññáß íá áëðáëáóðáß óá áñ÷éðáëðñíëéÿð x86 éáé DEC (ðßñá ðëÿÿí Compaq) Alpha. Áðù ôï FreeBSD 5.0 éáé ïáðÛ, ððñóççñßæííóáé áðßóçð ïé áñ÷éðáëðñíëéÿð AMD64 éáé Intel EM64T, ç IA-64 éáèðð éáé ç SPARC64®. Άñ÷éðáëðñíëéÿð ðñò éá ððñóççñßæííóáé ïáëëñíóéëÛ, ðáñéëáñÛññííí óçí MIPS® éáé PowerPC. ïðññáßðá íá áááñáðáßðá óðéð èßóðáð óá÷ðáññáßñò freebsd-ppc (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ppc>) éáé freebsd-mips (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mips>) áíðßóðñé÷á áéá ðëçññññßáð ó÷áðéëÛ íá óçí ðñññáñ ðñí áññáóéðñí óá áððÿð ðéð áñ÷éðáëðñíëéÿð. Áéá áñíëéÿð ðëçññññßáð ó÷áðéëÛ íá ñÿð áñ÷éðáëðñíëéÿð, áááñáðáßðá óóçí çäëðññíëéëß èßóðá ðñò FreeBSD áéá ðéð ïç-Intel ðëáððññáð (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms>).

Áí ôï ïç÷ÿçíá óáð áßíáé áéáðññáðéëëðð áñ÷éðáëðñíëéëðð éáé ÷ñáëÛæáóðá éáëðññáëëù Ûñáóá, óáð óðñéóðñíÿ íá ñßñáðá íéá ïáðéÛ óá NetBSD (<http://www.netbsd.org/>) Þ OpenBSD (<http://www.openbsd.org/>).

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

Υαεάο ιέ εΰνωάο θρω αάρ άβιάε SCSI άβιάε αάρεεΰ αίαεναόεεΰ θεί ανήΥό υοάí οόαενβίρφοάε ία άίροβόοίε÷ τώο SCSI ίααίρύο, έάε εΰθιερίε ίααίβ όγθίθ ΑΤΑΡΙ βούò ία ία έαείóιñάΠόριφ.

Ôá âðßßçíá FreeBSD CDROM ISO, êáèð êáé óá CDROM áðu ôï Daemon News êáé ôï FreeBSD Mall, ððïóçñßæïïí áêêßçícç áðáðêáßáð áðu ôï CD.

6. Ðiëié iäcãîß CD-RW õðíóôcñßæííôáé áðü ôî FreeBSD;

Ôi FreeBSD ððíóóçñßææé ððíéíäÞðíðä ðäçäü IDE CD-R Þ CD-RW óðíäáðü ðä ATAPI. Äßðâ ôí burned(8) äéä çäððñÝñäçäð.

Ôi FreeBSD ððíóôçñßæææ ððßôçð ððíëíäððíôä íäçäü SCSI CD-R þ CD-RW. ÅæéäóáóðÞóóä éæé ÷ñçóëíððíëÞóóä ôçí
 áíôíëÞ cdrecord áðü ôçí óðëëíäÞ ðüí ports þ áðü ðäéÝôí, éæé ááááéùèäßôä üóé Ý÷âôä áíóüíäðþóáé óóíí ððñÞíá óáð
 ôç óððéâðÞ pass.

7. Öđiíoçñßæåé ôi FreeBSD iäçãiyò Zip;

[illegible][illegible]

Ἄεα! Ὁσά ἀδβογόδ οἱ FAQ ὁ÷ἀοέε! Ὡ ἁάαεινῖγ!αῖνῖοδ ῖαῖαῖνῖοδ ὁά ἀδῖαῖν ὀαῖαῖν ἀοοῖν ὁῖο εἰσάεαῖν, εἰαῖο εἰε ὀέο ὀαῖαῖν ὁ÷ἀοέε! Ὡ ὀαῖ “αἰαῖνῖοδ” ὀοῖ εἰσῖαῖν Ἄεα÷αῖνῖοδ.

8. Õđioôcñβæåē òï FreeBSD Jaz, EZ êáé Üëëiòð áóäëñíýlãíiòð iäcãíýð;

Άδερφο ἐὰςὀριοναῖρε. Ἰὲ ὁἰδᾶνέοοῦδᾶἵαδ ἀδῦ ὁῶδ ὁῶδῆαδὺδ ἀδὺδὺδ ἄβῖαέ SCSI, ἐὰς ὺῶῶς ἄἄβ÷ῖῥῶῖ ὀαῖ SCSI ἄβῶῆῖς ὀῶῖ FreeBSD. Ὀῖ IDE EZ ἄῖῥῶῖβᾶῖῶᾶῖ ὀαῖ ῖᾶῇῡῡ IDE.

Âãääéùèàßôâ üöé Ý÷:ãôâ áíãñãĩđięŁóáé ôõ÷:üí âîùôãñééÝò óõóêãõÝò đñéí âêêéíŁóáôâ ôĩ óýóôçíá óáo.

Ãéá íá áëÛiãôâ ÿÝóí áðrèPêãóçð ãí pñã êãêôirñãBàð, âëÝãíôã ôéð mount(8), umount(8), êãé ôçí camcontrol(8) (ãéá ôóôêãôÝð SCSI) P atacontrol(8) (ãéá ôóôêãôÝð IDE), êãé âðBçðð ôéð ôðæçðPóãéð ó÷ãôééÛ iã ôç ÷ñPçç äöáénñiãíãíúí iãcãbí óã äðüiãí òiPíã ôiö FAQ.

4.4 Đệchônĩệũãéá êáé òĩĩôβééá

1. Öđioôcñßæåé ôi FreeBSD ôi USB đęcêôñiëüãéi iio;

Ôi FreeBSD òðíóôçñßæáé åãååíðò USB ðëçêôññíëüæá. ÁíåñåíðíëÉóòå ôçí òðíóôðñéíç USB óíí /etc/rc.conf.

[illegible][illegible]

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

ΔανάοçΠρόα υόε άί οί ðεçέοññευαέí USB άβράε οί ιñάάέεü ðεçέοññευαέí, έá άβράε άέάεÝóέíí υò /dev/ukbd0, έάε ç άíóñεP έá άάβ÷íάε υòυò ðάñάεÜòυ:

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

já êaeü iÝñiö áéá íá ðñiöeÝóáôâ ôcí ðañaðÜüü áíöireÞ, áßíáé ôí añ÷÷âßi /etc/rc.i386.

Ἰυεὸδ ἀβῖαε ἀδοῦ, οἱ USB θεξεῶνιευαεῖ εἰ δὴ Ὡθαε ἰά εἰεῶιῶναᾶβ εἰε ὀοῖ × δᾶνεἰῶεεῖ, ÷ ὑνβὸ ἀῖῶαεε ἀεἰεεῖ
 ῶεῖβῶῶῖ.

Ç áí çàñìþ óýíääóç éáé áðíóýíääóç USB ðεçεññíεíεäíñò, ßóυò íá íç εάεóíññääß áεúíä òυóóŸ. Óáo óóíεóóíýíä íá óóíäÝóäòä òí ðεçεññíεúεéí ðñéí óçí äéεßíçóç òíò óóóóßíäíò, éáé íá òí äóßóäòä óóíäíÝí íÝ÷ñé òíí ðäñíäóεóíυ, äéá íá äðíóýääóä òð÷úí ðñíäεßíäíä.

Äåßå öç óäëää manual ukbd(4) äéä ðañéóóüôåñåð ðëçñïïñßåð.

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

Ôi FreeBSD ôðioôçñßæåê ðñíôßééá ôýðïð bus éáé ôñí ôýðï InPort bus áðu éáoáoéâáoáoÝð üðuò Microsoft, Logitech éáé ATI. Ì ðõñÞñáo GENERIC ááí ðññéÝ÷÷é ôñí áðñáñáßçôï ãäçüü óóóéâôðð. Áéá íá ðññééÙááðá ôï ðñññññíá ãäÞäçóçð óóï áééü ðáo ðññóáññíïÙÝñ ðõñÞñá, ðññíóéÝóðá ôçí áéüüñïðèç áññáñÞ ðóïí áñ÷ßñ ðñéìßðóáñ ðõñÞñá:

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

Óá õĩĩóðééá òýðĩõ bus óõĩðeùð ÿñ÷ĩĩóáé ìá áééÿð òĩõð êÛñòáð áðÿêòáóçð. Áĩáá÷ñÿÿùð ìá ÿ÷÷áðá áõĩáðüòçðá ìá ñõèìðóáðá òçĩ êÛñòá òá áéáõĩñáðéêð áéáýõèĩóç èýñáð éáé IRQ áðü áðòÛ ðĩõ òáßĩĩĩóáé ðáñáðÛü. Óõĩáõéáðéáðòá òĩ áá÷áéñáäèĩ òĩõ ðĩĩĩóéêèÿ óáð éáé òç óáèßáá manual mse(4) áéá ðáñéóóóüðáñáð ðèçñĩĩĩñáð.

3. Дуò ìdĩñb íá ÷ñcóèiĩdiéPòu òi ðiíôβéé iĩ ôýðĩõ PS/2 (“ðũñôáo ðiíôéééíý” Þ “ðēcêôñiēiãβĩõ”);

Οἱ θεοὶ ἐπεὶ ὁ γὰρ PS/2 οὐκ ἔστιν ἀποφασίζοντα τὰς ἀρχαίας ἀποδείξεις. Οἱ ἀδελφοὶ ἐπεὶ δὲ νυνὶ ἀναίτια ἰσχυρίζονται, psm, ἀνεξαρτήτως τοῦ ὅτι δὴν πάλιν.

Áí í ðñîóáññîíîíÝíò ðññÞñáo óáo ááf ôñ ðññÝ÷áé, ðññóéÝóóá öçí áéüetöðç ãñññÞ óóí áñ÷ãß ñöèìóóáí ðññÞñá, éáé ìáóáäëüðöðóóá ìáfÜ öñ ðññÞñá óáo.

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


6. Ôĩ ðĩĩôßêê ïĩô Ý÷âé æëÛöĩñá Ýĩôðĩá ðëÞêôñá êáé ñĩáÝêá êýêéóçò. ÌðĩñÞ íá ôá ÷ñçóêĩĩðĩêÞò òôĩ FreeBSD;

Ç áðŰĩóçóç, äöóóð÷þò, áβíáé “áíáñòŰóáé”. Óá ðĩĩòβééá íä Ýĩòñá äűűáóũíòçóðò óűĩPèùò áðáéóĩŷĩ áíáéáééãòĩÝíá ðñĩāñŰĩíáóá ãāPāçóçò. Áí ðĩ ðñũāñāĩā ãāPāçóçò ðĩò ðĩĩóéééŷĩ P ðĩ áíóóóűĩé÷ ðñũāñāĩā ðĩò ÷ñPóóç áāĩ ðāñÝ÷÷űűĩ óðāéäēñēĩÝĩç ðĩűíóðPñēĩç áéá ðĩ ðĩĩóβéé, éá éäéóĩðñāāβ ùò Ýíá áðēũ ðĩĩóβéé áŷĩ P ðñēβĩ ðēPèòñũĩ.

Ãéá ðéèáíÐ ÷ ñÐóc ôcò ñïäÝëáo óå ðãñéâÜëëíí X Window, äåßôå ôcí áíôßóôíë÷ç áíüôçôå.

7. Ðùò ìðĩñþ íá ÷ ñçóèĩĩðĩéÞòù òĩ ðĩĩôßéé / trackball / touchpad óôĩĩ öĩñçôũ ìĩõ õðĩëĩãéóôÞ;

Äåâà ôçí áĎŪíôçóç óôçí đñiçãĩyìáíç ãñbôçóç.

8. $\text{Dùò} \text{ iǎĩnǐ} \text{ íá} \div \text{ñ} \text{ c} \text{ óe} \text{ i} \text{ i} \text{ d} \text{ i} \text{ é} \text{ P} \text{ óu} \text{ ôi} \text{ ð} \text{ é} \text{ P} \text{ ê} \text{ ô} \text{ ñi} \text{ delete} \text{ ó} \text{ ôi} \text{ sh} \text{ ê} \text{ áé} \text{ csh};$

Άέά δι ΈΎεοοο Bourne, οηίοέΥοοά οέο άέυειροέάο αηάηΥο οοί αη÷άβι οάο .shrc. Άάβόά άδβογδ οέο οάεβαάο manual sh(1) έάέ editrc(5).

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

Ἀέα ὅι ἘΎϵοοῖο C, ὀπιόε Ὑόόα ὀέο ἀέυειῖοεᾶο ἀᾶᾶᾶᾶ Ὑό ὀοῖ ἀᾶ÷ᾶᾶ ὀάο .cshrc. Ἀᾶᾶᾶ ᾶᾶᾶᾶ ὀς ὀᾶᾶᾶᾶ manual ὀῖο csh(1).

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

Ãéá ðãñéóóüôãñàò ðëçñïïïñßàò, äåßôå áõôP ôç óåëßåå (<http://www.ibb.net/~anne/keyboard.html>).

4.5 ÓóóâãÕÝò óáéñéáêÞò åðëêëïíúíßàò êáé Äéêôýùòçò

1. Ðięąò êŨñôąò äéêôýĩõ õđĩóôçñæąé ôĩ FreeBSD;

Άέά όεί ðēPñç ëβόόά, äåβôå ôéò Óçíäépóåèò Õëëëíý ðíð ðånÝ÷ííôåé íå êÛèå Ýëäíóç ôíð FreeBSD.

2. Õðïóôçñβæáé ôï FreeBSD modems ðïõ εάέôïññäíýí ìå ôç âïPεàέά ëïäέóíéêéý üðùò ôά Winmodems;

To FreeBSD [õðĩõõçñßæåæ](#) [ãnêãõ](#) software modems [ĩã õçĩ ãĩPèãéã](#) [ãðéðñüõéãõĩõ](#) [ẽĩãéõĩéĩĩ](#). [Ôĩ port](#) [comms/ltmdm](#)
[ðñĩõéYõãé](#) [õðĩõõPñéĩç](#) [ãéã](#) modems [ðĩõ](#) [ããóßæĩĩõãé](#) [õõĩ](#) [ãçĩõééYõ](#) [éyèèüĩã](#) Lucent LT. [Ôĩ port](#) [comms/mwavem](#)
[õðĩõõçñßæåæ](#) [õĩ](#) modem [ðĩõ](#) [ãéãéYõĩõĩ](#) [ĩé](#) [õĩñçõĩß](#) [õðĩẽĩãéóõYõ](#) Thinkpad 600 [éãé](#) 700 [õçõ](#) IBM.

Ääŕ iðŕnãßôá íá ääëáoóóöÞóáòâ ðí FreeBSD íÝóù software modém. Ôí ëĩäéóíëëü áðóü ðñÝðäë íá ääëáoóóóäëßß íàõÜ ðçí ääëáoÜóóáoç ôĩð FreeBSD.

3. ÕđÜñ÷:æé åãããîŸò đñũãñãîlà räPãçòçò ãéá ôéo êÜñôão Broadcom 43xx;

¼÷é, êáé ìÛëëíí äáí èá õđÛñîé.

Ç Broadcom áñíáßðáé íá áρðáé áçíυúóéá ðεçñíοíñßáð ò÷·áðéÜ ïá òíí ðñíáñáìáðéòíυ òíí íεíεεçñíυÝíυí ðçð ðíò ÷ñçóέííðíεýíðáé óá áðáñííáÝð áóýñíáðíυí áέéðýíυí, ðéáíυí áðáéáP éáé òí òíPíá ðíò ðñðíáÝéðç ðçð éÜñðáð

[illegible]

4. Ðíéàò êÛñòàò ðíēēáðēþí óáēñéáēþí èññþí òðííóçñßæííóáé áðü ôí FreeBSD;

ÕđŨñ÷:âé iéá ěßóôá áéá áõôÝò óóçí áíüôçôá äéŨöĩñùí óõóêãõþí

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

Áeúia oáβáíaoáe úoé eáeéioñáryí eáe eÜñieáo eÜñóao ðio áβáíe áíoeáñáoÝð áðpñoiur iiróÝeür, áeáeÜ uóao ðñioóçñβæñí oúe áβáíe óoiááoÝð ia óeó áíóβóoié÷áo óeó AST.

Äåßôâ ôç óääßää manual sio(4) äéá ðañéóóúôañð ðëçñïïĩñßð ó÷âôéêÛ ìâ ôç ñýèìéóç ôÝôïéúì éañôþí.

5. Дуò ìðĩñþ íá àìöáíßòù ôçí ðñĩôñĩðÐ boot: óå ìéá óåñéåêÐ êñíóüëá;

1. Ἀς ἐιρῆναι πόσῳ δὲ πρὶς τοῦ ἰαθεῖν ὅτι ἔστιν ἀδελφὰς options COMCONSOLE.
2. Ἀς ἐιρῆναι πόσῳ οἱ /boot.config ἔατ ἀνὸς ἰσὺς ὅς ἀδελφὰς ἰσὺς ὅτι ἔστιν ἀδελφὰς -P.
3. Ἀδελφὰς ἰσὺς οἱ δὲ ἐπὶ τοῦ ἀδελφὰς ἀδελφὰς οἱ ὅς ὅτι.

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

4.6 ÓóóâãõÝò Þ÷ïõ

1. **Điẻàò êÛñôàò P÷ĩõ òđĩóóçñßæĩĩôáé áđũ ôĩ FreeBSD;**

Οι FreeBSD οδισογνβιαεε aeUoiηaδ eUηoαδ p÷iθ, ooiδaηeeaiαaαmYfui ouf SoundBlaster®, SoundBlaster Pro, SoundBlaster 16, Pro Audio Spectrum 16, AdLib, eae Gravis UltraSound (aeα δaηeeoouoηaδ oδeγoiηiηBaδ, ααBoα oeo DeγoiηiηBaδ eaiogeo oiθ FreeBSD (<http://www.FreeBSD.org/releases/>) eae oγ oαeBaα manual snd(4)). OδUη÷ae αδBoeo δaηeiηeoYiYi oδiooPneic aeα eUηoαδ MIDI θiθ αβiae ooiαaoYδ ia oi θηuodoθi MPU-401. ΑδBoeo oδiooγnβiηoae ie eUηoαδ θiθ αβiae ooiαaoYδ ia oi θηuodoθi Microsoft® Sound System.

Όχιαβύθος: Οἱ θανάδῳῖν εὐ÷γαεῖ ἰῳῖ ἁεά οἷρ P÷! Ὁ δῆνῳῖνῳῖ ἰῳPῳῖοῖ ἁῖ ὀδῖοῖῖῖῖῖῖ ὀ÷ῳῖ CDROM, SCSI P joysticks δῖο ὀῖῖῖῖῖῖῖ ὀῖῖ ὀῖ SoundBlaster. Αῖ ἁεά ς ἁεάῖῖῖῖῖ SCSI ὀῖῖ SoundBlaster ἁεῖῖ ἁεά ἔῖῖῖῖῖ ἰς-SCSI CDROM ὀδῖοῖῖῖῖῖῖῖῖ, ἁῖ ἰῖῖῖῖῖ ἰῖῖῖῖ ἰῖ ἰῖῖῖῖῖῖῖῖῖῖ ἁεά ὀς ἁεάἁεἁἁῖῖ ἁἁἁῖῖῖῖῖ.

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

Εὐθιῆαο εὐνόαο π-ῖο, ὑδὸο ς es1370, ἱϋαῖαβῆοῖο ὀϋί Ἰγίαός οῖο π-ῖο ὀα εὐῆα ἀἐβίϋός. ΔῆΥθαί ἰά ἀœœαβὸα ὀϋί ἀεὐεῖοὲς ἀῖοῖεπ εὐῆα ὀῖῆ ῖο ἰἀέῖῤ ὀῖ ἱϋ-ῤίϋῖα:

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

4.7 ¶ëëï òëéêü

1. Ðíéàò Üëëàò óóóêãõÝò òðĩóôçñßæĩíôáé áðu ôĩ FreeBSD;

Äåßôå ôï Åã÷åéñßäéï

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/install.html#INSTALL-MISC) ἁέά ὁς ἔβόδᾱ ὀὐί
 ὀδῶιἔδῶι ὀδῶἔᾱὀἱ ὀἰ ὀἰὀδῶἱἔἱὀἱᾱἔ.

2. Ὁδὸς τοῦ FreeBSD ἀπὸ τοῦ Linux πρὸς τὸ FreeBSD

Άδου οι FreeBSD 4.X έαέ ιαòÛ, òðìíóçñßæàóáέ οι APM óά óóääêñêñíÝíá ιç÷-άίPìáíóá. Ðãñέóóúòãñò ðççññííßò ìðñãßòá íá ãñãßòá óóì apm(4).

[illegible]

3. Dùò iǒiǎǎ íá áđáíáñǎǎiđiéPóu ôi ACPI;

ĐñĩóèÝóôầ ôç ãñàìĐ

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

óôï áñ÷åßï /boot/device.hints.

4. Ãéáôß ôĩ Micron óýóôçιά ïĩõ êñãĩÜãé êáôÜ ôçí âêêßícóc;

[illegible]

Άέά ίά δάñάēÛìøåòå ôì ðñüåēçíá, áðáíáññíðíēΠóòå ôçì áðēēíāΠ “Plug and Play Operating System” áðü ôì BIOS.

5. Ç äéóêÝôá âêêßícóçò êñàìÛäé óóç ìçônéêP ASUS K7V. Ðùò ìðññ íá ôï äéññèþòù áõõù;

Óêò ñõèìáóáèò ôiõ BIOS, áðáíáñáíðíéPóòâ ôçí áðééíãP “boot virus protection”.

6. $\tilde{A} \acute{e} \acute{o} \beta \text{ } \zeta \text{ PCI } \acute{e} \tilde{U} \tilde{n} \acute{o} \acute{a} \text{ } \acute{a} \acute{e} \acute{e} \acute{o} \acute{y} \acute{i} \acute{o} \text{ } \acute{i} \acute{i} \acute{o} \text{ } \acute{o} \acute{\zeta} \text{ } 3 \text{Com}^{\circledR} \text{ } \acute{a} \acute{a} \acute{i} \text{ } \acute{e} \acute{a} \acute{e} \acute{o} \acute{i} \acute{o} \tilde{n} \tilde{n} \acute{a} \acute{\beta} \text{ } \acute{i} \acute{a} \text{ } \acute{o} \acute{i} \text{ } \text{Micron } \acute{o} \acute{\delta} \acute{i} \acute{e} \acute{i} \acute{a} \acute{e} \acute{o} \acute{p} \text{ } \acute{i} \acute{i} \acute{o}$;

ἸνεοίΥῖαὸ ἰϥὸνῆῆῆΥὸ δϣὸ Micron Υ̇ ÷ ἰῶῖ ἰϥ-ὸδῶῖῖῆῆῆῆΥῖ PCI BIOS οῖ ῖῶῖῖ ἁῖῖ ῖῶῖῖῖῖῖῖ ὀῶδ ὀῶῶῶῶΥὸ PCI ὀῶῶ ῖῶῶῶῶῖῖῖῖῖ ῖῖῖ ἰῖῖῖῖῖῖῖῖ. Ἀδῶῖ ἁῖῖῖῖῖῖῖῖῖ ῖῖῖῖῖῖῖῖῖ ῖῖῖῖῖῖῖῖ ῖῖῖῖ ῖῖῖῖῖῖῖῖ ὀῖῖ FreeBSD.

Àéá íá ðáñáêÛøâôâ ôî ðñüâêçìá, áðáíãñäîðíéÞóôâ ôçí áðééíãÞ “Plug and Play Operating System” áðü ôî BIOS.

7. Ç PCMCIA êÛñôá ïï ääí ääéôïðññâß. ÂëÝðù ôï äîðò ìðíðíá: “cbb0: unsupported card type detected.” Ñé ìðñþ íá êÛñù;

Ìðññâðâ íá äíëéìÛóâðâ íá ÷ñçóéìðñéðóâðâ ôçí äñ÷éð ðëððíçççç OLDCARD. Ññðððñéðóâðâ ôï äñ÷âßí ñýëéóçð ôïð ððñðíá óáð, êáé áðáéñÝóâð ðéð áêüëïðëáð ãñáìÝð:

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

ÌáðÛ ðñíóèÝóâð:

```
device pcic
device card 1
```

Ìáðáäëùðððóâðâ íáíÛ êáé ääéáðáóððóâðâ ôï íÝí ððñðíá üðùð ðññéãñÛððáé óðçí Ñýëéóçç ððñðíá ôïð FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/kernelconfig.html).

ΕὰοÛεάεί 5

Αίôείαôþðέόç ÐñĩâëçìÛôùì

1. Αέάôß ôì FreeBSD ânßóέάέ εÛεìð ðìóùðçðά ìíþìçð;

Αôðù ðåßεάôάέ óðç áέάóìñÛ ìåðáíý ððóέεþí έάέ áέέííέεþí áέåðέýíóåùí ìíþìçð.

Ç óýíåáóç ðìð έåðÛ åÛóç áέííðέåßðάέ óðì ðέέέù ðìð PC, åßíάέ íå ÷ ñçóέííðìέåßðάέ ç ìíþìç ìåðáíý 3.5G έάέ 4G áέå áέåέέù óέíðù, óðíðέùð áέå óçí ðñùóååóç óå εÛñðåð PCI. Αôðù Ý÷å ùð åðìðÝέåðíå íå ìçí ìðñåß íå áíóέóðìέ÷çέåß ððóέέþ ìíþìç óå åððþ óçí ðåñέí÷ ð áέåðέýíóåùí.

Ôì ðέέέù ðìð ððìέíåóðð óåð εå έåíñßðάέ ðέ åßíåðάέ ìå óçí ìíþìç ðìð έåííέέÛ åíåíñæåðάέ óå åððþ óç εÝóç. Αôððð÷þð, óå εÛðìέåð ðåñέðððåðð ðì ðέέέù ååí εÛíåð ðßðìðå, έάέ ÷ Ûíåðάέ ç åðíåðùðçðå ÷ ñþðçð ðùí ðååðððåßñ 500ì ìíþìçð RAM.

Αôðð÷þð, ðððð ðåñέóóùðåñåð ðåñέðððåðð ðì ðέέέù áíåååððέýíåð óç ìíþìç óå ðççέùðåñç εÝóç, þððå íå åßíάέ åέùíå åðíåðþ ç ÷ ñþðç óçð. Αôðù ìðñåß ùððùóì íå óåð ðñíέåÝóåέ εÛðìέå óýå÷ðóç áí ðåñåέííðέåßðå óå ìçíýíåðå åέßßíçóçð.

Óðçí 32 bit Ýέåíðç ðìð FreeBSD, ç ìíþìç ðåßíåðάέ íå Ý÷å ÷ åέåß έåèðð áíåååððέýíåðå ðÛñ ððù óå 4G, óå ìðñå ååí åßíάέ ðñíåðåÛóέíå ððù 32 bit ððñþíå. Óðçí ðåñððððçð åððþ ç έýóç åßíάέ íå ððέÛíåðå Ýíå ððñþíå óýðìð PAE. Αåßðå åððþí óçí έåðå÷þç óðì FAQ áέå ðåñέóóùðåñåð ðέçññìðñåð.

Óðçí 64 bit Ýέåíðç ðìð FreeBSD, þ ùðåí ÷ ñçóέííðìέåßðάέ ððñþíåð óýðìð PAE, ðì FreeBSD εå áíέ÷íåýóåέ έåέ εå áíåååððέýíåð ðùððÛ óç ìíþìç þððå íå åßíάέ ÷ ñçóέííðìέþóέíç. ΈåðÛ óçí åέßßíçóç ùððùóì, ìðñåß íå ðåßíåðάέ ùðέ ðì FreeBSD áíέ÷íåýå ðåñέóóùðåñç ìíþìç ððù åððþ ðìð Ý÷å óðçí ðñååíåðέέùðçðå ðì óýóðçíå. Αôðù åßíάέ ððóέέííέέέù έάέ ç áέåέÝóέíç ìíþìç εå åέíñèùέåß έåèðð ðìέççñþíåðάέ ç áέååέååðå ðçð åέßßíçóçð.

2. Ì óέççñùð ìðð åßóέìð Ý÷å ÷ åέåóìÝñìð ðñåßð. Ôέ ìðñþ íå εÛñ;

Óðìðð åßóέìðð SCSI, ì ðåçåùð ìðñåß óðíðέùð íå åðåíåðìðìέåßðåðå åððùíåðå óå åååñÝíå óå áíåέåέðέέýðð ðñåßð. Ûððùóì ìέ ðåñέóóùðåññέ åßóέìέ Ýñ÷íðåέ ìå óçí åðíåðùðçðå åððþ åðåñåññìðìέçìÝíç.

Άέå íå åíåñåñìðìέþðåðå óçí åðåíåðìðìέÝðçóç ÷ åέåóìÝññ ðññÝñ, åðåñåñååððåðå óçí ðñþðç óåßåå έåðÛððåóçð óçð ððóέåððð (modem), åßñíðåð óçí ðåñåέÛðð åíðìέþ (ùð root):

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

έάέ åέέÛíåð ðέð ðέíÝð ðùí AWRE έάέ ARRE åðù 0 óå 1:

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

Ìέ óýå÷ñññέ ðåçåñþ óýðìð IDE Ý÷íñí åðßðçð åíåñåñìðìέçìÝíç åðù ðì åñåñððÛóέíç óç åðíåðùðçðå åðåíåðìðìέÝðçóçð ÷ åέåóìÝññ ðññÝñ.

Αí ååßðå ðñíååñìðìέþåðð ð÷åðέÛ ìå ÷ åέåóìÝññ ðñåßð (óå ìðìέíåþðìåðå åßåðð åßóέìð), åßíάέ þñå íå óέåððåßðå íå åέέÛíåðå ðññ ðåçåñ. ðððð ìðññÝóåðå íå ÷ ñçóέííðìέþðåðå ðì åέååñùðέέέù ðññåñåñíå ðìð åßíάέ ìέåðåóέåðåóððð ðìð

6. I æeôôðuôPò iïõ óóçí ðánÜëëççç èýná åbíáé áðæðéóóêêÜ åñåùò. Ôé iðñp íá êÜü;

Ár ôî ïüñ ðñüäëçíá áßíáé í òðãñãñëëëÛ äñãüò äêôððùðòð, ìðññãðòá íá äñëëÛóàðòá íá äëëÛíàðòá ðçí éäðÛóððáðç ëäëðìðññãðòá ðçð ðãñÛëëççèð èýñãäð (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/printing-intro-setup.html#PRINTING-PARALLEL-PORT-MODE) üððò ðãñëãñÛóàðáé óðì êâðÛëáéí òïò Åã÷-äéñéãßïð ó÷-äðëëÛ íä ðçí ÅãëäðÛóððáç Åêôððùðòð (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/printing-intro-setup.html).

7. Ā́éáôß ôá ðñĩāñŨĩááôá ĩĩō ðāñéóôáóéáêŨ ôāñĩááôßæĩōĩ ĩā óóŨëĩá Signal 11;

Óá óóÙeíáóá óýðíō Signal 11 áçíeíōñāíýíóáe úóáí íeá áeāñāáóá ðñíóðāeāß íá ðñíóðāeÜóáe ðñāñeí÷P ííðíçō áeá ðçí íðíßá āāí Ý÷÷áe ðÜñāe Üāāeá āðū ðí ēāēíōíñāēēū óýóðçíá. Áí óōíāāßíāe ēÜóe óÝóðeíe óā óāēñāíāeēÜ ðō÷āßá ÷ñííeēÜ áeáóóðíáóá, ēá ðñÝðāe íá āñ÷ßóāðā íá ðí āñāóíÜðā ðíey ðñíóāēóēēÜ.

Ôá ðñîäëÞîáôá áõôÛ óõíÞeùò ìõäëîíõáé óá êÛðíeíí áðu ôîõð ðáñáêÛòù eüäîõð:

- [illegible]

Ἀέα θανῶαέαια, ὕϊαο ἀνῶανῖο οἰνῶο ἰά αέαθεοοῶοαοα οἰε *αἰῖ* θῆαέαοαέ αέα θῆαέαῖα οῖο FreeBSD, ἀβίάε αἰ οἰ θῆαέαῖα αἰοαῖβαέαοαέ εἰαῶ ὅε ἰαοαέθοοέοε εὔθιεῖο θῆανῶαοοο, αέεῦ εὔεα οἶνῶ εἰε οἰ αέαοἶαοέεε οἰαβῖ.

[illegible]

Ôé ðñÝðæé íá êÛíaôâ:

Όδες δηλαδή δανεισθένους ιδιαιβάδα ία ÷ η̣οςείηιδίεΠοάδα εΰθιεή debugger ιιδὺδ οἱ gdb ἀέα ία ἀνὰβόα οἱ ογίαΒι οόι
δηυάηηαι ία ογι δηηάεçιαόεεΒ αέάγεορός έάε ία οἱ αείηεβοάδα.

Óôç ääýôâñç đăñßđôùôç, èà đñÝđăé íá âđăëçèăýôăđă ûôé äăí ôôăăăé ôî ôëëëû óáo.

Óôèò óóíçèéóíÝíàò áéôßàò áõõíý õîõ ðñííâëPiáõîò, ðãñééâîâÜííîáé:

- [illegible]

[illegible]

- [illegible]

Èa ðnÝðáε αðΒοçð íá αεάáÜοαðá ðí SIG11 FAQ (ðí ιðίΒí ðáΒíáðáε ðáñáεÜòü) ðí ιðίΒí ðáñεεαíáÜííáε áíáεñáðóεéÝð áðáíçáΒοáεð áεá υεά αοóÜ ðá ðñíáεΒíáðá, áí εáε ðñεεÝð áðu αοóÝð áΒíáε áñáñíÝíáð áðu óçí óείðεÜ ðíò Linux®. Íá áíáεáoÝñíí ðíΒíá ðíò SIG11 FAQ áΒíáε εáε αóου ðíò áíáoÝññáðáε óóçí ðεεáíυóçðá íá ιçí áíε÷íáÝáðáε ðñíáεçíáðóεεΒ ιñΒιç áðu αεάáíυóóεéÜ ðñíáñÜñíáðá Β οóóεáσÝð áεÝá÷÷ò.

[illegible]

İdiñâßôâ íâ ãñâßôâ âêôâôâíÝíç áíÜëöóç óôî FAQ ó÷âôêêÜ ìâ ôî ðñüâêçíâ SIG11. (<http://www.bitwizard.nl/sig11/>)

8. **Ōi** óyóóciá iir óóáiáóŬáé áßóá iá Fatal trap 12: page fault in kernel mode, **p** iá panic:, áß÷firóáo éáé iéá óáénŬ áðu ðecnrioinßáo. **Óé** ðnÝðáé íá éŬrù;

Ç nÜää afÜððoñçð ðïð FreeBSD äfaäáoÝñäðáé éäéäBoðñä äéä äððÜ ðä eÜeç, äeeÜ ÷ñäeÜæäðäé ðñäéðóðüðñäð ðççñïññäð äéðüð äðu ðï ïðñïä eÜeñðð ðñð äeÝðäðä. ÁñðéäñÜððä ðï ðeðñäð ïðñïä éäé Ýðäéðä ðñïäñeäðeäBoðä ççï äñüçðä ðñð FAQ ð÷äðééÜ ïä ðä kernel panics, äçñeñðñäðóðä Ýñä ðññðñä ïä äññäðüçðä äéððäeñÜðüçðð (debugging kernel) éäé äéðäeÝðä Ýñä backtrace. Áðüð ïðññäð ïä äeññäðäé äýóeñeñ, äeeÜ ääñ ÷ñäeÜæäððä ððçï ðñäñäðééüçðä äñðäéð ðññäññäðééññ. Áñeäð ïä äeñeñðeðóðäðä ðéð ñäçäðäð.

9. Ἀέας ἔϊεύς ἰῖο ἰάωνβææέ έάέ ÷ Ûíáέ õí óõ÷ñíέóìù õçò έάòÛ õçí æêêβícç;

[illegible]

ÏÝ÷ñé íá äéíñèùèåß ôï ðñüäëçíá áõöü, ìðíñåßòå íá ÷ñcóëíððéßóåòå ôï ðññéÛöù òÝ÷íáóíá äéá íá ôï ðññéÛöùòå:

- [illegible]

Ἄρ ἐΥεᾶὸα ἰά ÷ ñçóóíñðíεἰΠóαὸα óεὸ óáεñέáεΥὸ εὕñáð, εᾶ ðñΥðáε ἰά äçíεíñāΠóαὸα ἱΥἱ ððñΠῑά, ἰά ðçí áέυεἰíðεç ἰáðáðññðᐅ: Óðἱ áñ÷áḂἱ /usr/src/sys/i386/isa/sio.c áñāḂðā ðἱ ðñḂðἱ óçἱáḂἱ ðἱò àἱòáἱḂæáðáé ðἱ áεὸáñέἱðçðéέυ 0x2e8 έáé áóáεñΥòðā áðòυ ðἱ áεὸáñέἱðçðéέυ έáé ðἱ έυñἱá ðἱò áñḂðéáðáé ðñέἱ áðυ áðòυ (έñáðΠóðā ðἱ έυñἱá ðἱò áñḂðéáðáé ἰáðŮ). ἈέίεἱἱðéΠóðā ðññā ðç óðἱçέέçἱΥἱç áéááέéáóḂā äçíεíñāΠáð ἱΥἱ ððñΠῑά.

Ἀέυια έάέ ιάδῚ ὀςι ἀόανιρῖαᐅ ἄδὀβι οῦι ἁέινἔβὀαῦι, βὀδὀ ἁίἁέἁέἃὀἁὀ ὑὀέ ὀι ὀὀόὀῑῑῑ X Window ἁἁι ἁἁέὀῖῖῖἁᐅ
 ὀὀὀὀῚ. Ἀί ὀὀἁἁἁἁἁἁ ἄδὀὑ, ἁἁἁἁἁἁἁᐅἁὀ ὑὀέ ÷ ἥὀὀὀὀὀὀὀὀὀὀὀ ὘ἁὀῑὀ 3.3.3 ᐅ ιἁἁἁἁὀἁụ ὀὀὀ XFree86™. Ἀὀὑ ὀςι
 ὘ἁὀῑὀ ἄδὀᐅ ἁἁἁ ιάδῚ, ὀὀὀᐅ ÷ ἁἁἁ ἁἁὀἁἁὀἁὀὀὀὀ ὀὀὀὀὀᐅụἁἁἁ ἁἁἁ ἔὀụὀἁὀ Mach64 ἁἁἁ ἁὀβὀὀὀ ἁἁὀᐅᐅἁὀἁἁ
 ἁἁἁἁἁἁὀὀὀὀὀ ἁἁὀὀὀụἁὀὀᐅᐅ X ἁἁἁ ὀςι ἔὀụὀἁ ἄδὀᐅ.

10. Ἄεάοῃ οἱ FreeBSD ὀγδόηία ἡδ ÷ ἵςόέἡἡἡἡἡἡ ἡἡἡ 64MB RAM, ἁἡ ἡ ἡἡἡἡἡἡἡἡἡ ἡἡ Ḃ ÷ ἁἡ ἁἡἡἡἡἡἡἡἡ Ḃ ἡἡ 128MB;

Ἀἰαέλοβαο οἷο οἰνῶδιος ἰὰ οἷι ἰθῖβι οἷ FreeBSD ἀεᾶᾶᾰᾰᾰᾰ οἷ ἰΎᾶᾶᾶᾶ οἷο ἰθῖοἷ αᾶᾦ οἷ BIOS, ἰθῖᾶᾦ ἰὰ ἀῖᾦ÷ἰᾶῖᾶᾦ ἰῖῖ 16 bits ἰΎᾶᾶᾶᾶ οᾶ Kbytes (65536 Kbytes = 64MB) (ᾦ ἑᾶᾦ ἑᾶᾦᾦᾦᾦ... ἰᾦᾦᾦᾦᾦ BIOS ᾶᾦᾦᾦᾦ ᾦᾦᾦᾦᾦᾦᾦᾦᾦᾦᾦ ἰΎᾶᾶᾦᾦ ἰθῖοἷ 16). Ἀῖ ᾦ÷ᾦᾦ ᾦᾦᾦᾦᾦᾦᾦᾦᾦᾦ αᾶᾦ 64MB, οἷ FreeBSD ἑᾶ ᾦᾦᾦᾦᾦᾦᾦᾦᾦ ἰὰ ᾦᾦ ἀῖᾦ÷ἰᾶῖᾶᾦ. ᾦ ᾶᾦ÷ἰᾦᾦᾦᾦ ᾦᾦᾦᾦᾦ ἰθῖᾦᾦ ἰὰ αᾦᾦᾦ÷ᾦᾦ.

Ἀέά íá ðáñáέṼíøáðá ðí ðñíúάέíá, έά ðñÝðáέ íá ÷ ñçóέííðíέíøáðá ðçí áðέέíáð ðíø ððñíá ííø íáβíáðáέ ðáñáέṼíø. ὬðṼñ ÷ áέ ðñíðíð íá έçøέíý í ðέβñáέð ðέçñíííñβáð ó÷áðέέṼ íá ðç ííðíç áðu ðí BIOS, áέέṼ óðí bootblock ááí ððṼñ ÷ áέ áñέáíðíð ÷ βñíð áέά íá áβíáέ áðí. ἘṼðíέά íÝñá, úóάí áέíñέúέáβ ðí ðñíúάέíá ðçð Ṽέέáέøçð ÷ βñíð óðά bootblocks, έά ÷ ñçóέííðíέíøííá ðέð áέðáðáíÝíáð έάέííñáβáð ðíø BIOS áέά íá áíáέðíøííá ðέβñáέð ðέçñíííñβáð ó÷áðέέṼ íá ðç ííðíç. Ἀέά ðçí βñά, ðñÝðáέ íá ðáñέíñέóðíýíá óðçí ñýέíέóç ðçð áíðβóðíέ÷ð áðέέíáð ðíø ððñíá.

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

140r0 dī n ābīāē dī īYāāēi0 oç0 īpīç0 oā kilobytes. Āēā īç÷Üīçīā īā 128 MB, ēā ðñYðāē īā ÷ñç0ēīīðīēP0ā0ā dī 131072.

11. Οι ούόόεία ιιό Υ÷άέ δάνέοούόάνι άδύ 1 GB RAM, έάέ δάβññ panics ιά ιςίίάάά “kmem_map too small”. Διό άβίάέ οι δññάέεία;

[illegible][illegible]

12. Οι όγδόηιά ιῖο ἀάί Υ÷âé IGB RAM, éâé ðŬëé ùùò òι FreeBSD äçìéïñããß panic ìâ òι ιPíðíá kmem_map too small!

Ōī panic āā÷÷īāē ūōē ōī óýóōcīā Ý ÷÷āē īāīāē āđū āēēīēēP īīPīç āēā đñīōūñēīP āđīēPēāōōç āāāñÝīūī āēēōýīō
 (network buffers, ēāē āēāēēūōāñā mbuf clusters). ĮđīñāBōā īā āōīPōāōā ōī īÝāēēō đçō āēēīēēPō īīPīçō đīō
 āēāōPēāōāē āēā mbuf clusters, āēīēīōēPīōāō ōēō īācāBāđ ōōcī āīūōōcā ¼ñēā Āēēōýīō

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html#NMBCLUSTERS) òïð
 Åã÷:ãéñéäβïð.

13. Æéáôß đǎßñîù ôî ì!îïǎ üËèĩòò /kernel: proc: table is full;

[illegible]

Ἄεά ίά ἡ ἡὲἰβόαῶα ὀγί ὀεἰβ ὀῖο MAXUSERS, ἁἁβῶα ὀγί ἁῖυὀγῶα ¼ἡέά Ἀñ÷ἁβῖι/Ἀέἡἡἁῶέῖῖ
(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html#KERN-MAXFILES) ὀῖο Ἀñ÷ἁέἡέἡῖῖ. (Ἀί ἐάέ ς ἁῖυὀγῶα ἁὀὀβ ἁίἁῶῚἡἁῶάέ ὀἁ ἁῖῖ÷ὀῚ ἁñ÷ἁβῖ, ὀἁ βἁέἁ ὑἡέἁ ἐὀ÷ῖῖῖ ἁέἁ ἁέὀ ἁέἡἡἁῶḡἁὀ.)

Αί οι ις÷Üicia óað éæðioññāāā ðā ÷÷içēü viññōñi, æēÜ æððæāā iāāÜēi añēēü æéññāāóēpi, iðññāōā ððēð íá
 ñðēiβóāðā ðñ añēēü ðñð æēÜæññóāð ðçí ðēiP ðçð iāðāāēçðPð kern.maxproc. Αί ðñÝðæ íá ñðēiβóāðā ðððP ðç
 iāðāāēçðP, éä ðñÝðæ íá ðçí iññβóāðā ðññ añ÷āñi /boot/loader.conf. Ç ñýēiέóç āāi éä éó÷ýóāē Ý÷ñé íá
 āðñāēēēñPðāðā ðñ óýóðçíā. Åéä ðññéóóüðññāð ðççññiñññāð ð÷āðēēÜ iā ðēð iāðāāēçðÝð ðñð ðñññíā, āāñðā ðēð
 óāēñāāð manual loader.conf(5) éäē sysctl.conf(5). Αί üēð ðððÝð iē æéññāāóβāð æððæññýíóāē áðñ Ýíā iñññ÷ññPðç,
 éä ðñÝðæ ððβóçð íá ñðēiβóāðā ðçí ðēiP ðçð iāðāāēçðPð kern.maxproccperuid þððā íá āññāē éāðÜ Ýíā iēēññüðññç
 áðñ ðçí iÝā ðēiP ðçð kern.maxproc. (ÐñÝðæ íá āññāē éāðÜ Ýíā iēēññüðññç, æéāðβ ððÜñ÷æ ðÜññóā Ýíā ðññññññññ
 ððððñññññ, ðñ init(8), ðñð ðñÝðæ íá æððæñññāðæ óñÝ÷æä.).

[illegible]

14. $\tilde{A}\acute{e}\acute{\alpha}\acute{o}\tilde{\beta}$ $\acute{d}\acute{a}\tilde{\beta}\tilde{n}\tilde{i}\tilde{u}$ $\acute{o}\tilde{i}$ $\tilde{i}\tilde{P}\tilde{r}\tilde{o}\tilde{i}\tilde{a}$ $\tilde{e}\tilde{U}\tilde{e}\tilde{i}\tilde{o}\tilde{o}$ CMAP busy $\tilde{u}\acute{o}\acute{a}\tilde{i}$ $\acute{a}\acute{d}\acute{a}\tilde{i}\acute{a}\tilde{e}\tilde{e}\tilde{e}\tilde{i}\tilde{\beta}$ $\tilde{i}\tilde{a}$ $\tilde{i}\tilde{Y}\tilde{i}$ $\acute{d}\acute{o}\tilde{n}\tilde{P}\tilde{i}\tilde{a}$;

Ç ēīāēēP oīo oōōōPīāōīō ðīo ðīōīōðāēāP íā áíē÷íāyōāē ðō÷ūí ðāēēYō āēāūōāēō ðūí āñ÷āBūí /var/db/kvm*_ .db
ēÜīēīēāō vīñYō āðīōðā÷Üīāē, ēāē ç ÷ñōPç áfūīīēūí āēāūōāūí īðīñāP óā īñēōīYíāð ðāñēðōbōāēō íā rāçāPōāē óā panic.

Áí óáo óõïââß áõõü, ððáíáêêéíÞóôâ óá êáoÜóóáoç áíüð ÷ñÞóôç (single user) êáé ãñÜøðâ:

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

15. **Ôé óçíáßíáé ôï ìÞíõíá** ahc0: brkadrint, Illegal Host Access at segaddr 0x0;

ÕđŨñ÷:æé iéá æéÝíâîç iâ ôcí êŨñôá Ultrastor SCSI Host Adapter.

Ἐὰν οὖν ὁς ἀεὶ ὑπὸ νόμῳ ᾖ, οὐκ ἔσται υἱὸς τοῦ νόμου, ἀλλ' υἱὸς τῆς ἐλπίδος, ὅτι ἡ ἐλπίς ἐστὶν ἡ ἀρετὴ τοῦ νόμου, ἡ ἡμέρα τῆς χάριτος καὶ τῆς ἀγαθῆς ἐλπίδος, ἡ ἡμέρα τῆς ἀγαθῆς ἐλπίδος, ἡ ἡμέρα τῆς ἀγαθῆς ἐλπίδος.

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

Ç içönéeP ðeéeÝóá óáo áái Ý÷áé óá áðáéeöiyíáíá áíuðáñéèÜ òèèeþíáóá þóóá íá òðíóðçñBæáé áóóuíaóí ðáñíáðéeóíu òíò áeáyëíò SCSI. ÁíòB íá ááoBæáóóá óóíí áóóuíaóí ðáñíáðéeóíu, açèþóðá óóí SCSI BIOS òíí óuóóu ðáñíáðéeóíu áeá ðç áéÜóáiç óóóèáðí þíò Ý÷áóá. Òí ðñuáñáíá íäþäçóçð òíò AIC7XXX áái íðíñáB íá éáèíñBóáé áí áBíáé áeéeÝóéíí òí éýèèuía ðíò ÷ñçóéííðíeáðBáé áeá ðçí áíB÷íáðóç òíò éáèuáBíò (Üñá éáé òíò áóóuíaóíò ðáñíáðéeóííý). Òí ðñuáñáíá íäþäçóçð ððíeÝóáé uóé ððÜñ÷áé ððíóðBñéíç, áóóóíí íé ðñèíBóáéð ðíò ðáñéÝ÷ííðáé ðóç óáéñéáéP EEPROM áíáóÝñíòí "áóóuíaóí ðáñíáðéeóíu". Óó÷íÜ, ÷ññBð òí áíuðáñéèü éýèèuía áíB÷íáðóçð òíò éáèuáBíò, òí ðñuáñáíá íäþäçóçð éá ðñèíBæáé éáíeáóíÝíá òíí ðáñíáðéeóíu, èÜóé ðíò íðíñáB íá açìéíòñäþBóáé ðñuáèçìá óóçí áíeíðéóóBá òíò áeáyëíò SCSI.

17. $\tilde{A}\acute{e}\acute{o}\beta\ \acute{o}\tilde{i}\ \text{Sendmail}\ \acute{a}\tilde{\beta}\acute{i}\acute{a}\acute{e}\ \acute{o}\tilde{i}\ \grave{i}\Pi\acute{\iota}\acute{o}\acute{i}\acute{a}\ \acute{e}\tilde{U}\acute{e}\acute{i}\acute{o}\acute{o}$ “mail loops back to myself”;

Áõöü đăñéăñŬöăôăé ôôi sendmail FAQ üđùò öăβíăôăé đăñăêŬôù:

* Ðáßñíù ìçíýíáôá ëÛèíðò "Local configuration error" üðùò ôí:

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

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

Đuò ìđĩñþ íá ăđéěýòù ôĩ đñüâēcìá;

÷ ãðã æçððóáé íá éáðãðëýíáðã ðí mail ðñíð ðí domain (ð.÷. domain.net)
ðñíð ëÛðíëí ððãëãëñëíÝí íç ÷ Ûíçíá (ðóçí ðãñðððóóçé áððð, ðí
relay.domain.net) ÷ ñçóëííðíëíðíðãð íéá áããããðð MX, áëÛ Õ ðí íç ÷ Ûíçíá
ðíð ëÛíáé ðçí áíáéáðãýëðíóçé áãí áíáãíñññæãé ðíí ááððü ðíð ùð
domain.net. ÐñíðéÝððã ðí domain.net ððí /etc/mail/local-host-names
(áí ÷ ñçóëííðíëíáððã ðí FEATURE(use_cw_file)) ð ðñíðéÝððã
"Cw domain.net" ððí /etc/mail/sendmail.cf.

Ç õñÝ÷÷÷÷÷÷ Ñëäíõç ðí÷ sendmail FAQ (<ftp://rtfm.mit.edu/pub/usenet/news.answers/mail/sendmail-faq>) ääí
 ð÷íðçñäðäë ðëÝíñ ìä ëÙëä Ñëäíõç ðí÷ sendmail. Ùððúíí, äçííóëäýäðäë áíÙ ääëðÙ äëäóððíäðä óðëð ëððóäð
 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), êäë news.answers (news:news.answers). Ìðññäðä
 äððõç ìä ëÙääðä áíððäñäí÷ ÌÝòù email, óðÝëñííóä Ñíä Ìðñíä ðí÷ <mail-server@rtfm.mit.edu> ìä ðçí áíðíëð
 send usenet/news.answers/mail/sendmail-faq ðí÷ ëýñëí ÌÝñíð ðí÷ ìçýíäððð.

18. Ἄεάοβ̅ αᾶί οὐιδᾶνεόΎνιιόαέ ουόóŨ ié āōāñiīāÝð ðēPñiōð ièuicíð óå áðñiāēñōíÝía ic÷áíPiaôá;

Ἄβριας δεεφαῖν οἱ ἀδναῖνοι ὧν ἱς ÷ Ὑίς τῆς ἰᾶ πῶς βεᾶς οἱ ὅς τοῖς ἀνῖαδῶς ὅς ὁ εὔος ἀεοῖν ἀδου οἱ ὅς
cons25 ὅς ἀδᾶδᾶδᾶς ἀδου ὅς εἰρῶς οἱ FreeBSD.

ÕðÛñ ÷ iõí æéÛöirié ôñuðié æéá íá ðañaéÛìøåðå áõöü ôi ðñuâëçíá:

- [illegible]

δυναμικὰ. Ἐνταῦθα δὴν ὁ ὁρὸς **screen** οὐδὲν ἄλλο ἢ τὸ δῶκεν ὅτι VT100, ὅπου ἡ ἀσφάλεια τῆς TERM οὐκ ἀντιλαμβάνεται τὴν ἀσφάλεια τῆς vt100.

- Ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν cons25 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ἡ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.
- Ὅτι οὐδὲν ὅπου FreeBSD ἡ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

19. Ἀσφάλεια τῆς ἀσφάλειας τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

Ἀσφάλεια τῆς ἀσφάλειας τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

Ἐνταῦθα δὴν ὁ ὁρὸς **SIGXCPU exceeded cpu time limit**.

Αἱ ἀσφάλεια τῆς ἀσφάλειας τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

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

Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

20. Ἀσφάλεια τῆς PnP οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

Ὁ ἀσφάλεια τῆς PnP οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς PnP οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς PnP οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

Ἡ ἀσφάλεια τῆς PnP οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς PnP οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100. Ὁ ἀσφάλεια τῆς PnP οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

Ὁ ἀσφάλεια τῆς vt100 οὐκ ἔχει τὴν ἀσφάλεια τῆς vt100.

Ὅτι PNP bios οἱ δῆνι-νῦείεσθαι [οἱ modern] ἐὰν οἱ Ὑόροα ὁδοί δῆνι-νῦείεσθαι ἄεσθαι ἴσθαι οὐκ ἐνῶντι, ἐὰν Ὑόρος [ὁδοί Ὑεῖρος 3. x] εἰ
 δῆνι-νῦείεσθαι ὁδοί ἴσθαι ISA οἱ "ἀνῶν" ἄεσθαι.

Ὁδὸν Ἰεραῖος 4.0, ἡ ἐπαεῶν ἀεὶ–αβνέος τοῦ ISA, ἀβιάς διεῖς θἀνεόουόανῃ θνιόαράθεεῖοῖ Ἰνδὸ οἰ PnP ἡνδὶ Ἰε. Ὁδὸ 3. × Pοάι αῖοῖαυῖς ὁ αἰβ–ῖαόος ISA ῖα αἰοῖθβόαῖ ῖα “–αἰ Ἰγῖ” οῶοῶαP εἰς Ἰθᾶεῶς ὁ PNP οῶοῶαP ῖα οἰεῖῖαῖ εἰς ῖα αἰοῖγ–ᾶς ὁ ἡγῖεός ὁδὸ εἰαυῖ αἰῖαῖθδὸ θνῖν. ὁε, αἰαῖαῖαῖθδῖεῖγῖαῖ αἡ–εῖ ῖ θνῖαῖαῖαῖαῖεῖαῖαῖ εἰῖαῖ, Pοά ῖα ἡ οἰαῖαῖ αῖοP ὁ αἰδP αἰβ–ῖαόος. Ἀόου αἰθβός ὁγῖαῖαῖ ῖος ὁ αἰβ–ῖαόος θνῖῖαῖ ῖ αἡνῖαῖαῖ ὁα PnP ἡδ οἡ θδῖόοςῖεῖαῖαῖ οῶοῶαῖ. Ἀβιάς οῶοθ θνῖῖαῖαῖ ῖα ῖ εἰῖαῖαῖ ὁ αἰαῖεῖαῖαῖ αῖοP θἀνεόουόανῃ θνιόαῖαῖ οἰοῖ –ῖPοάθ.

Άέά ίά έάέοιῶηάΠόάέ ίάίŨ ç οόόεάöΠ, ḏñÝðáé ίά άηάεαß öi PNP id öçð éäé ίά ḏñïóðáεαß öçð ëßóóá öui áíé÷ íáyóauí
ISA ḑïō ÷ñçóëüḑiēíýíóáé áéá öçí áíaāīḑñéóç PnP öóóéåðí. Áðöü iðññāß ίά āßíáé íā öç ÷ñḑóç öçð pnpinfo(8) áéá öçí
áiß÷ íāöóç öçð öóóéåöΠò, áéá ḑánŬääéaiā äöôḐ åßíáé ç Ýññäö öçð pnpinfo(8) áéá Ýíá áóúöāñééü 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)

[ðáñáëǽßðííôáé ié õðüëiéðǽò 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
```

[illegible]

ΑίάέέάέôéêÛ, αί ôī pnpinfo(8) ääí ääß÷íáé ôçí æçôīýíáίç êÛñôá, ìðīñáßôâ íá ÷ñçóéīīðīéÞóáôâ ôī pciconf(8).

Đa ñáêÛòù òáßíáôáé Ýíá ìÝñìò ôçò âñüäñò ôçò pciconf -v1 ãéá Ýíá êýêëùìá Þ÷ïò áíóùìáôùìÝññò óôç ìçñéêêÞ:

```
# 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 ÷ñçóëíïðíéíýóâíå ôçí ôèìþ ôïõ chip, “0x24158086”.

Ç ðëçñïöïñßá áóôÞ (Vendor ID Þ ôêîÞ chip) èá ðñÝðäé íá ðñïóóàèàß óôï áñ÷åß /usr/src/sys/isa/sio.c.

Εὰ δὴ Ὡδαὲ δὴπὸα ἰὰ ἐναὸΠαῶα Ἰά ἰαὸΒαῖαὶ αὐῶαεἰῶδ οἶο sio.c, ἀεὶ ὀγι δᾶνβδδουογ δῖο εὔοε δῦαε ὀνααῦ. Ἀδβὸγδ, εἰ ÷ πᾶεαὐῶαὶ οἱ ἰαὸΒαῖαὶ ἀεὶ ἰὰ ἀγῖεῖοῖαΠαῶα Ἰά patch οἱ ἰδῖβ ἰα εἰῶεἸῶα ἰα ὀγι ἰαῖοῖνῦ δὴἰαεἰαῖοδ (PR) δῖο εἰ ἰαδ ὀαῖεἰῶα (εἰε εἰ ἰαδ ὀαῖεἰῶα PR, Ἰῶε);. Ἐἰουδῖε ἰδᾶἰαῖαὐῶαὶ οἱ sio.c εἰε ὀῦἰῶα ἀεὶ ὀγ ἰᾶἰῖ

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

Ýðáεáá íááááεéεááááá ðññð áá εÛðð áεá íá áñáááá áí óúóóú íÝññð íá ðññðéÝóááá óçí εááá÷þñçóç óçð óóóεááááá óáá. Íε εááá÷÷ññááááá óááññááááá ùðùð ðáñáεÛðð εáε ááíáε ááíεñçñÝíáð εááÛ áí áεóáñéεéçóóéú ASCII Vendor ID áí íðññ éá ðñÝðáε íá ðáñéεçðéáá óáí ó÷úεéí óáí ááíεú íÝññð óçð áñáññð íááá íá ùεç óçí ðáñéáñááááá *Device Description* (áí ÷÷ññÛáε, áéééεáá íÝññð óçð) áðù óçí Ýíññáí óçð 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 */
```

ÐñiøëÝóôâ òì ääëäâíäëëü Vendor ID äéá òç óóóëäòÐ óád òòì óóóóü ìÝñìò, áðñèçëäýóôâ òì äñ÷âì, áíääçìèíòñâÐóôâ òì ðòòñíá óád, éäë äðáíäëëíÐóôâ. Èá ðñÝðäë òðñá ç óóóëäòÐ óád íá äñäëâ ùò óóóëäòÐ sio uðùð òòìÝääëíä éäë ìä òì FreeBSD 3.X

21. `Ãéáôß đáßñiù ôi ëÛèìò nlist failed üôáí âêôâëþ, ãéá đánÛääéãíá, ôi top Þ ôi systat;`

Ôi ðñũæçia ábíaé uôe ç áoãñiãþ ðiõ ðñiódæãðòã íá æeôæ Ýoãòã øÛ÷íæ áéá Ýía óoãæãñeîÝíi óyãìæi óóii ðõñPía,
 æeeÛ áéá eÛðieí euiã áãí iðñiãþ ía ôi áíôîððòæ. Ôi óoÛeia áoõu iðñiãþ ía iõãðeãòæ óã áýi ðñiæPiaóá:

- [illegible]

22. Ἄεάοβ δάβñíáε ουοί ÷ñuí íá οοíáαεβ íα οíí ὀδíεραεόοP iiō íYóu ssh P telnet:

[illegible]

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/configtuning-kernel-limits.html) οἷο
 Ἀῖ÷ἀνέειβῖο, ἀέᾶ ἀνῖαῖβᾶ ἐᾶέ ἀδβῖοός οἷο δῖνῖαῖβᾶοἷο.

25. Ἀέᾶδβ οἷ νῖεῦε οἷ οἷνῖοῦ ἱῖο δῖνῖαῖβᾶοἷο ἀᾶ ἑνᾶδῖᾶέ οῖ οῦοῶβ ἡνᾶ;

Ἰ οἷνῖοῦ δῖνῖαῖβᾶοἷο οᾶδ ῖ÷ᾶέ ἄγῖ β δᾶνέοοῦοᾶᾶ νῖεῦᾶέᾶ, ἐᾶέ οἷ FreeBSD ῖ÷ᾶέ ἀδῖῖᾶέ ἰᾶ ÷ῖνῖοἷδῖεβᾶέ οἷ
 ἔῖῖο.

Ἀέᾶῖῖᾶέ οῖ dmesg(8), ἐᾶέ ᾶῖᾶῖᾶ ᾶέᾶ ᾶᾶᾶᾶᾶ δῖο δᾶνῖῖ÷ῖοἷ οῖ ἔῖῖ Timecounter. ῖ οᾶῖᾶᾶᾶ ᾶδῖ οῖδ
 ᾶᾶᾶᾶᾶ δῖο ἐᾶ ᾶέοδῖῖᾶᾶ ᾶᾶᾶ÷ῖᾶέ οἷ νῖεῦε δῖο ἀδῖῖῖ÷ῖῖᾶ ᾶδῖ οἷ FreeBSD ἐᾶέ ο÷ᾶᾶῖ ὀβᾶῖῖᾶ ἐᾶ ᾶῖᾶέ οἷ TSC.

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

Ἰδῖᾶᾶᾶ ἰᾶ οἷ ἀδῖᾶᾶᾶῖᾶᾶ ᾶδῖ, ᾶῖᾶ÷ῖᾶᾶ οῖ οῖᾶβ οἷο kern.timecounter.hardware sysctl(3).

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

Οἷ BIOS βῖοῦ ἰᾶ δῖνῖδῖῖᾶᾶ οῖ οῖᾶβ οἷο νῖῖᾶῖῖ TSC— ᾶῖᾶ÷ῖᾶᾶ ᾶέᾶ ἰᾶ ᾶῖῖᾶέ οῖ οᾶ÷ῖᾶᾶ οἷο ᾶδᾶᾶᾶᾶᾶᾶ
 ῖᾶᾶ ῖᾶῖᾶᾶᾶ ἰᾶ ἰᾶᾶᾶᾶᾶ, β ῖᾶᾶ ᾶῖᾶᾶ÷ᾶᾶ ὀᾶ ῖᾶᾶᾶᾶᾶ ÷ᾶῖᾶᾶ ῖᾶᾶᾶᾶᾶᾶ, ᾶῖῖ οἷ FreeBSD ᾶᾶ
 ᾶῖᾶᾶᾶ ᾶῖᾶ ᾶᾶᾶᾶ ὀῖδ ᾶῖᾶᾶᾶ ῖᾶ ὀβᾶῖᾶᾶ ἰᾶ ῖᾶᾶᾶᾶ β ἰᾶ ÷ῖᾶέ ÷ῖᾶᾶ.

Οἷ δᾶᾶᾶᾶᾶ ἰᾶ, ᾶῖᾶ ᾶδβῖᾶ ᾶῖᾶῖᾶ οἷ νῖεῦε i8254 ἐᾶέ ἰδῖᾶᾶ ἰᾶ οἷ ἀδῖῖᾶᾶ ᾶᾶᾶᾶᾶ οἷ ῖᾶᾶ οἷο οἷ
 sysctl(3) kern.timecounter.hardware.

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

Ἰ οἷνῖοῦ δῖνῖαῖβᾶοἷο οᾶδ ἐᾶ δῖᾶᾶᾶ ὀᾶ ἰᾶ ᾶῖᾶ ὀῖ ᾶῖᾶᾶᾶ ὀῖ ὀβῖᾶᾶ οἷο ÷ῖᾶᾶ.

Ἀέᾶ ἰᾶ δᾶᾶᾶᾶᾶ ῖ ᾶῖᾶᾶ ᾶᾶᾶ ὀᾶ ἔῖᾶ ᾶῖᾶᾶᾶ, δῖᾶῖᾶᾶ ὀᾶ ὀᾶᾶᾶᾶ ᾶᾶᾶᾶ /etc/sysctl.conf.

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

26. Ἀέᾶδβ ἰ οἷνῖοῦ ἱῖο δῖνῖαῖβᾶοἷο ἀᾶ ᾶῖᾶᾶᾶᾶ ὀῖδ ἔῖᾶᾶ ὀῖδ PC card;

Οἷ δῖᾶῖᾶ ᾶῖᾶ ἔῖῖ ὀᾶ οἷνῖοῦ δῖο ᾶῖῖᾶῖᾶ δᾶνέοοῦοᾶᾶ ᾶδῖ ῖᾶ ῖᾶῖᾶᾶᾶᾶᾶ ὀῖᾶᾶᾶ. Ἰᾶῖᾶῖ ἰῖ-BSD
 ῖᾶῖᾶᾶᾶᾶ ὀῖᾶᾶᾶ ᾶᾶᾶᾶ ὀῖ PC cards ὀᾶ ἰῖ-δῖᾶῖᾶᾶ ῖᾶᾶᾶᾶ. ῖ ᾶῖᾶᾶ pccardd ὀᾶ ᾶᾶᾶ ὀῖ
 δᾶᾶᾶᾶᾶ, ᾶῖᾶ÷ῖᾶᾶ ὀῖ ἔῖᾶᾶ ῖδ " (null) " " (null) " ᾶῖᾶ ᾶῖᾶ οἷ δᾶᾶᾶᾶᾶᾶ ὀῖ ἱῖᾶῖᾶ.

Δῖᾶᾶ ἰᾶ ᾶῖᾶᾶᾶᾶ ᾶῖᾶᾶᾶ ὀῖ δῖᾶᾶᾶᾶ ᾶδῖ ὀῖ ῖᾶᾶ PC card ἡᾶᾶ οἷ ὀῖᾶᾶ ἰᾶ ᾶᾶᾶᾶᾶ ὀῖ ᾶᾶ÷ῖᾶ οἷο
 ῖᾶᾶᾶᾶ. Ἀᾶᾶᾶᾶᾶᾶ ὀῖ οἷνῖοῦ δῖνῖαῖβᾶοἷο οᾶδ. (Ἰῖ οἷ ᾶῖᾶᾶ ὀᾶ ῖᾶᾶᾶᾶ ᾶῖᾶᾶᾶ β ῖᾶᾶ, ἐᾶ
 δῖᾶᾶ ἰᾶ ᾶᾶᾶᾶᾶᾶᾶ ᾶῖᾶᾶᾶ.) Δᾶᾶῖᾶᾶ ᾶῖᾶ ἔβᾶ ῖᾶᾶ ῖᾶ ᾶᾶᾶᾶᾶᾶ. ἔᾶ δῖᾶᾶ ὀᾶ ῖ PC card ἰᾶ
 ῖᾶῖᾶᾶᾶ ῖᾶῖᾶᾶ.

Οἷ ὀῖᾶᾶ ἔῖᾶᾶ οἷνῖοῖ δῖνῖαῖβᾶοἷο ὀῖ δᾶᾶᾶᾶᾶᾶᾶ ᾶᾶᾶῖᾶ ᾶᾶᾶᾶ, ᾶῖᾶ ῖᾶ ῖᾶᾶ ὀῖᾶᾶᾶᾶ ῖᾶ ἰ
 δῖνῖαῖβᾶοἷο ᾶῖᾶ ᾶῖᾶᾶᾶ. Ἀῖ οἷ δᾶᾶᾶᾶ ᾶᾶ ῖ÷ᾶέ οἷ ᾶῖῖᾶᾶ ᾶῖᾶᾶᾶ, ὀᾶᾶᾶᾶᾶ ὀῖ ῖᾶῖᾶᾶᾶ οἷο
 δῖνῖαῖβᾶοἷο οᾶδ, ᾶᾶῖᾶᾶ ὀῖ ἰᾶᾶᾶᾶ, δᾶᾶῖᾶᾶ ἔβᾶ, οἷᾶᾶᾶᾶ ἰᾶᾶ ὀῖ ἰᾶᾶᾶᾶ ῖᾶ ᾶᾶᾶᾶᾶᾶ.

27. Ἄεάδθ ι öññòùòÐð ãêêßíçóçð ðíð FreeBSD äãß÷íáε ðí ìßíðíá ëÛëïðð Read error êáé óóáíáðÛáε ìáðÛ ðçí ðèùíç ðíð BIOS;

Ἰ οἶνόνουορδὸ ἀέθελβζορδὸ οἷο FreeBSD ἀὰρ ἀράῃῃῃῃῃῃῃ ὀούοῦ ὀζῖ ἄῃῃῃῃῃῃῃ οἷο ὀέεζῃῃ ἄβὀέῃ. Ἰῃῃῃῃῃῃ ῃῃ ὀζῖ ῃῃῃῃῃῃῃῃ ÷ ἄῃῃῃῃῃῃῃῃ Ἰῃῃῃ ἄῃῃ ὀζῖ fdisk ῃῃῃῃ ὀζῖ ἄῃῃῃῃῃῃῃῃ ῃ ὀῃῃῃῃῃῃῃῃ ὀζῖ slice οἷο FreeBSD.

Ἰδῖνᾱβδᾱ ἰά ᾱᾱβδᾱ δεὸ οὐοὸὶὸ δειὶὸ ᾱά ὀçi ᾱᾱῖᾱδᾱβᾱ οἶθ ῖᾱçῖῖ ὀῖθ BIOS οἶθ ἰç÷ᾱἰᾱῖᾱδῖθ. ØÛῖᾱ ᾱά ὀἶ ᾱᾱῖῖῖ ὀῖ ῖᾱῖᾱᾱῖῖ, ῖᾱῖᾱῖῖ ῖᾱῖ ὀἶῖῖ ᾱά ὀἶ ῖᾱçᾱῖ ῖῖθ ῖῖᾱῖᾱ.

İYóá áđũ ôçí fdisk ôĩõ sysinstall(8), đẾYóôâ ôĩ G ãéá íá ĩĩBoâôâ ôçí ãũũĩôĩĩBá ôĩõ ĩãçĩĩY.

Èà àìòáíéóóàß Ýíáð æÛëïáìò ðìò èá æçðÛáé òìí áíéèìù òùí èòëßíáíùí, èáòáëðí èáé òíÝùí. ÐëçèðííëíáÐóóà òìòð áíéèìýð ðìò áíÐëáðá áðù òí BIOS, ÷-ùíßæííóáð òìòð ìà èáíííéÝð èáéÝòìòð. Áéá ðánÛááèàì, áéá 5000 èòëßíáíìòð, 250 èáòáéÝð èáé 60 òííáßð, èá áíÛóáìá **5000/250/60**.

ĐéÝôâ enter ãéá íá ĩñBoââ ôèò ôéiÝò, éáé Ýđæôá ôi W ãéá íá ãŨøââ ôi íÝi đBíáéá éâôâôìPoâuì ôôíi ĩäçũ.

28. ἰά Ὑεῖ ἑἀέοῖρῶῆἑῦ ὀύόοçῖᾱ ἑἀόΎόῶῆᾰ ὀῖ ἁῑᾱ÷ἁῆῆῑῑᾰ ἁῆῆῖῑῑῑῑ ἱῖῑ. Ðùῑ ἱῑῖῇ ἰᾱ ὀῖ ἁῑῖᾱῑῑῑῑᾰ;

[illegible]

29. **Ôé óçíáβíáé ôî ìΠíóíá ëÜëíõð** swap_pager: indefinite wait buffer;;

Όχιαβλαί υδὲ ιεά ἀεάαεεάοβα ὀπιοδᾶεβᾶ ἰά ἀνῚῥᾶε ἰεά οᾶεβαά ἡπῖϗὸ οῶι ἀβῶεῖ, ἐάε ϗ ἄδῡδᾶεῖνᾶ ἀοῶβ Ὶ ÷ ἄε ἐρεεῖρῶᾶε ὀπιοδᾶεπῖοᾶδ ἰά ἀδῖεῶρῶᾶε ὀπυῶᾶᾶορ οῶι ἀβῶεῖ ἀεά ὀᾶνεοῶυῶᾶῖ ἄδῡ 20 ᾶᾶῶᾶᾶᾦᾦᾶῶᾶ. Ἀῶῶῡ ἰῶῖᾶᾶ ἰά ὀῶἰᾶᾶ ἄδῡ ÷ ᾶεᾶῖ Ὶῖῖῶ ὀᾶᾶῶ οῶι ὀεεϗῖῡ ἀβῶεῖ, ὀᾶῖᾶεῖᾶῶεῖῚ ἐᾶεῖᾶεά, ῡ Ὶεεῖ ὀεεῖῡ ὀἰ ἰῶῖῖ ἰά ὀ ÷ ᾶῶῖᾶᾶᾶε ἰᾶ I/O. Ἀῖ ὀᾦᾦᾶεᾶῶᾶε ἀεά ὀᾶῖᾶεῖᾶῶεῖῚ ἀβῶεῖ, ἐᾶ ᾶᾶῶᾶ ᾶῶῶῶ ἑᾶε ᾶῖῶῶῶῖ ÷ ᾶ ἰεῖῖᾶῶᾶ ὀῶι /var/log/messages ἐᾶε ὀῶεῖ Ὶῖᾶῖ ὀεῶ ᾶῖῶῖῶῶ dmesg. ἈεᾶῖᾦᾶῶεῖῚ, ἄεῚᾶῖᾶ ὀεῶ ὀῶῖᾶῚᾶεῶ ἐᾶε ὀᾶ ἐᾶεῖᾶε ὀᾶῶ.

30. Ôé âßíáé ôá óöÛëíáôá “UDMA ICRC”, êáé ðùò ìðññ íá ôá äëññèðù;

[illegible]

Οἱ θηυαεγια ἰθιναβ ἱα θηηεεεαβ αδυ θηεεηγδ δανῦανῶαδ, αἱ εαε ἱ θεἱ οῳεεεοἱ ῤηῖθ ἄβιαε ς θηηαεγιαοεεP P εαρεαοἱ ῤε εαευαβῦοε. ΑεYαῖῶα ὑοε οα εαεπαεα ἈΟΑ ααἱ ῤ ÷ ἱοἱ οῳἱοοαβ αεεἱῤ, εαε ὑοε ἄβιαε εαοῤεεεεῖθ θηηαεαανῶπἱ αεα οεἱ εαοῤοοαοε εαεοἱοηαβαδ Ultra DMA θῖθ ÷ ηεοεἱθῖεἱαβῶα. Αἱ ÷ ηεοεἱθῖεἱαβῶα αοαεηἱἱαἱ οῳθῶῤηεα ἄβῶεῖθ, εα θηῤθαε ἄδβῶοδ ἱα ἄβιαε οῳηαοῤ. Ἀααεεεαβῶα ὑοε οῳῤ ÷ αε εαεP ἄδῶαP οα εεαδ οεο οῳαῤῶαε, ÷ ἱοἱ ἄδβῶοδ αἱαοῶηεαβ θηηαεPἱαοα ὑοαἱ ἄβιαοαε ἄαεαῤοοαοε αἱυδ θαεεῖθ ἱαεῖθ οῳἱ βαεἱ εαἱῤεε DMA ἱα ῤἱα ἄβῶεἱ Ultra DMA 66 (P θεἱ ἄηPαηἱ). Ὀῤεῖθ, οα εῤεε αοοῤ ἰθιναβ ἱα οεἱαβἱῖθ ὑοε ἱ ἄβῶεἱθ θηυεαεοαε οῖῖῶηα ἱα ÷ αεῤῶαε. ἱε θῶηεοῳοῶηἱε εαοαοεἱαοαοῤ ἄβῶεῖθ δανῤ ÷ ἱοἱ εἱαεοἱεεῖ ἄεῤα ÷ ἱῶ αεα ὅῖοδ ἱαεῖθ οἱῖο, ἄεῤῶα εἱεῖθ οἱ ἄβῶεἱ οαο, εαε αἱ ÷ ηαεῤαοαε, ὀῤῶα αἱοδῶαῖθ οἱ ααἱῤῤῖθ οαο εαε αἱοεεαοαοοPοοα οἱἱ.

[illegible]

Áí óõíõñßá, áðõíý õîð åßäïðð ïé ðññâéäïðéþóáéð åáí åßíáé óõíþèùð ïïéñáßåð, áëëÛ òðù ïñéóìÝíåð áðð÷åßð ðññùðñèÝóáéð, ïðññåß íá ðññéåéÝóìõí áíåðéèýìçðåä öáéíìíííáä íðá ïðñíåä êõíåßññíóáé áðù ïéå öðéåñéååßåä ððþóç óðçíí áðùèñéóç õîð óðóðþíåðò, ïÝ÷ñé ðèþñçð éåðÛññåðóç.

33. Åéåðß ç äéåäééåóåßåä buildworld/installworld óðåííåðÛåé ïå õî ïþñòíåä touch: not found;

Ôî ïþñòíåä áððù åáí óçíåßíåé ùðé óåð éåßðåé õî äñçèçðéèù ðññåññåííåä touch(1). Ôî ëÛèðð áððù ðññéåéååßðåé óõíþèùð áðù éåíéåóìÝíç, ïåëëíðééþ, óþíåíóç çñññçííåäð õùí åñ÷åßåßñ. Áí õî ïññèé CMOS õîð ððññéåéóðþ óåð åßíåé ïðèééóìÝíñ äéåððééþ þñå, ðñÝðåé íåðåéÝóåðåä óçííñðéþ adjkerntz -i åéå íåðèßóåðåä õî ïññèé õîð ððñþñåä ùðåíí åêééíåßðåä óå éåðÛðóåóç éåéõññåßåð åñùð ÷ñþóç.

Chapter 6

Commercial Software Vendors

ÖçİâBüóç: 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.

ÖçİâBüóç: 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/>)

Chapter 7

Ports and Packages

I. So, where are all the user applications?

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

Most ports should work on the 4.X, 5.X, and 6.X branches. Each time a FreeBSD release is made, a snapshot of the ports tree at the time of release 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 çäêôññíéêþ ëßóôá ôüí 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).

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

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

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

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

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

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

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

```
makeoptions DEBUG=-g
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7. What is `kern.quantum`?

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

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

See Å: 7.

ÊåöÜëáéï 9

Äßóêïé, Óõóôßìáôá Áñ÷ åßùí êáé ÖïñôùôÝò Äêêßíçóçò

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

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

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

The best way is to reinstall the OS on the new disk, then move the user data over. This is highly recommended if you have been tracking -STABLE for more than one release, or have updated a release instead of installing a new one. You can install booteasy on both disks with `boot0cfg(8)`, and dual boot them until you are happy with the new configuration. Skip the next paragraph to find out how to move the data after doing this.

Should you decide not to do a fresh install, you need to partition and label the new disk with either `/stand/sysinstall`, or `fdisk(8)` and `disklabel(8)`. You should also install booteasy on both disks with `boot0cfg(8)`, so that you can dual boot to the old or new system after the copying is done. See the `formatting-media` article (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/formatting-media/index.html) for details on this process.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Once you have got the BIOS and FreeBSD agreeing about the geometry of the disk, your problems are almost guaranteed to be over, and with no need for “DD” mode at all. If, however, you are still greeted with the dreaded `read error` message when you try to boot, it is time to cross your fingers and go for it - there is nothing left to lose.

To return a “dangerously dedicated” disk for normal PC use, there are basically two options. The first is, you write enough NULL bytes over the MBR to make any subsequent installation believe this to be a blank disk. You can do this for example with

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

Alternatively, the undocumented DOS “feature”

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

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

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

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

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

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

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

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

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

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

5. What is inappropriate about my ccd?

The symptom of this is:

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

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

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

The symptom of this is:

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

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

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

7. Can I mount other foreign filesystems under FreeBSD?

FreeBSD supports a variety of other filesystems.

Digital UNIX

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

Linux

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

Windows NT®

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

FAT

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

ReiserFS

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

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

8. How do I mount a secondary DOS partition?

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

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

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

9. Is there a cryptographic filesystem for FreeBSD?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

12. How do I boot FreeBSD and Linux using GRUB

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

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

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

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

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

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

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

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

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

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

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

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

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

if it is a floppy, or this:

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

for a ZIP disk with the factory configuration.

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

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

Unless it is a floppy, or a removable you plan on sharing with other people, it is probably a better idea to stick a BSD filesystem on it. You will get long filename support, at least a 2X improvement in performance, and a lot more stability. First, you need to redo the DOS-level partitions/filesystems. You can either use `fdisk(8)` or

/stand/sysinstall, or for a small drive that you do not want to bother with multiple operating system support on, just blow away the whole FAT partition table (slices) and just use the BSD partitioning:

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

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

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

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

and mount it:

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

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

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

16. Why do I get Incorrect super block when mounting a CDROM?

You have to tell mount(8) the type of the device that you want to mount. This is described in the Handbook section on optical media (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html), specifically the section Using Data CDs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html#MOUNTING-CD).

17. Why do I get Device not configured when mounting a CDROM?

This generally means that there is no CDROM in the CDROM drive, or the drive is not visible on the bus. Please see the Using Data CDs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html#MOUNTING-CD) section of the Handbook for a detailed discussion of this issue.

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

Your CDROM probably uses the “Joliet” extension for storing information about files and directories. This is discussed in the Handbook chapter on creating and using CDROMs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html), specifically the section on Using Data CDROMs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html#MOUNTING-CD).

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

You most likely burned a raw file to your CD, rather than creating an ISO 9660 filesystem. Take a look at the Handbook chapter on creating CDROMs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html), particularly the section on burning raw data CDs (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/creating-cds.html#RAWDATA-CD).

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

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

21. Why can I not mount an audio CD?

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

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

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

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

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

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

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

2. As `root` assign the appropriate permissions to the block device associated with the removable media.

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

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

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

```
# chgrp operator /dev/acd0c
```

```
# chmod 640 /dev/acd0c
```

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

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

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

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

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

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

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

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

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

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

Unmounting the device is simple:

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

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

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

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

You need to understand what `du` and `df` really do. `du` goes through the directory tree, measures how large each file is, and presents the totals. `df` just asks the filesystem how much space it has left. They seem to be the same thing, but a file without a directory entry will affect `df` but not `du`.

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

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

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

25. How can I add more swap space?

In the Configuration and Tuning

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

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

Disk manufacturers calculate gigabytes as a billion bytes each, whereas FreeBSD calculates them as 1,073,741,824 bytes each. This explains why, for example, FreeBSD's boot messages will report a disk that supposedly has 80GB as holding 76319MB.

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

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

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

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

Chapter 10

System startup and configuration

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. 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.

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

```
ttymc 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 `vnlru`?

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

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

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

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

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

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

30. How much free memory is available?

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

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

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

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

Chapter 11

Installing X Windows

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, 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.

Notes: 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`.

Διάρθρωση: 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.

Όχι: 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 ¶ 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`.

Οῖαῖοῖεῖ: 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:

```
ttv9  "/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.

Chapter 12

Networking

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:

Βίááo 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

Vendor	Model
Danpex	EN-9400P3
JCIS	Condor JC1260
Linksys	EtherPCI
Mylex	LNP101
SMC	EtherPower 10/100 (Model 9332)
SMC	EtherPower (Model 8432)
TopWare	TE-3500P
Znyx (2.2.x)	ZX312, ZX314, ZX342, ZX345, ZX346, ZX348
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.

ÊâöÛëáéï 13

ÁóöÛëåéá

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 Å: 9..
- If the problem is local, read on entry Å: 10..

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

There is very little you can do about this. Most ISPs will refuse to help if you are not running a Microsoft OS. You can enable `lqr` in your `ppp.conf` file, allowing ppp(8) to detect the remote failure and hang up, but this detection is relatively slow and therefore not that useful. You may want to avoid telling your ISP that you are running user-PPP...

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

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

Then reconnect to ensure that this makes no difference. If things improve or if the problem is solved completely, determine which setting makes the difference through trial and error. This will provide good ammunition when you contact your ISP (although it may make it apparent that you are not running a Microsoft product).

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

If your ISP is helpful, they should be able to enable logging on their end, then when the next link drop occurs, they may be able to tell you why their side is having a problem. Feel free to send the details to Brian Somers <brian@FreeBSD.org>, or even to ask your ISP to contact me directly.

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

Your best bet here is to rebuild ppp(8) by adding `CFLAGS+=-g` and `STRIP=` to the end of the Makefile, then doing a `make clean && make && make install`. When ppp(8) hangs, find the ppp(8) process id with `ps ajxww | fgrep ppp` and run `gdb ppp PID`. From the gdb prompt, you can then use `bt` to get a stack trace.

Send the results to Brian Somers <brian@FreeBSD.org>.

11. Why does nothing happen after the “Login OK!” message?

Prior to FreeBSD version 2.2.5, once the link was established, ppp(8) would wait for the peer to initiate the Line Control Protocol (LCP). Many ISPs will not initiate negotiations and expect the client to do so. To force ppp(8) to initiate the LCP, use the following line:

```
set openmode active
```

ÖçİâBüöç: 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 xzf 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 `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_MULTIPOINT` option.

The following example is for an AST 4-port serial card on `irq 7`:

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

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

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

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

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

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

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

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

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

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

So you want to become an Internet service provider, eh? First, you will need one or more modems that can auto-answer. Your modem will need to assert carrier-detect when it detects a carrier and not assert it all the time. It will need to hang up the phone and reset itself when the data terminal ready (DTR) line goes from on to off. It should

probably use RTS/CTS flow control or no local flow control at all. Finally, it must use a constant speed between the computer and itself, but (to be nice to your callers) it should negotiate a speed between itself and the remote modem.

For many Hayes command-set-compatible modems, this command will make these settings and store them in nonvolatile memory:

```
AT &C1 &D3 &K3 &Q6 S0=1 &W
```

See the section on sending AT commands below for information on how to make these settings without resorting to an MS-DOS terminal program.

Next, make an entry in `/etc/ttys` (see `ttys(5)`) for the modem. This file lists all the ports on which the operating system will await logins. Add a line that looks something like this:

```
ttyd1 "/usr/libexec/getty std.57600" dialup on insecure
```

This line indicates that the second serial port (`/dev/ttyd1`) has a modem connected running at 57600 bps and no parity (`std.57600`, which comes from the file `/etc/gettytab`, see `gettytab(5)`). The terminal type for this port is `dialup`. The port is `on` and is `insecure`---meaning `root` logins on the port are not allowed. For dialin ports like this one, use the `ttydx` entry.

It is common practice to use `dialup` as the terminal type. Many users set up in their `.profile` or `.login` files a prompt for the actual terminal type if the starting type is `dialup`. The example shows the port as `insecure`. To become `root` on this port, you have to login as a regular user, then `su(1)` to become `root`. If you use `secure` then `root` can login in directly.

After making modifications to `/etc/ttys`, you need to send a hangup or HUP signal to the `init(8)` process:

```
# kill -HUP 1
```

This forces the `init(8)` process to reread `/etc/ttys`. The `init` process will then start `getty` processes on all `on` ports. You can find out if logins are available for your port by typing

```
% ps -ax | grep '[t]tyd1'
```

You should see something like:

```
747 ?? I      0:00.04 /usr/libexec/getty std.57600 ttyd1
```

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

If you are using another computer as a terminal into your FreeBSD system, get a null-modem cable to go between the two serial ports. If you are using an actual terminal, see its accompanying instructions.

Then, modify `/etc/ttys` (see `ttys(5)`), like above. For example, if you are hooking up a WYSE-50 terminal to the fifth serial port, use an entry like this:

```
ttyd4 "/usr/libexec/getty std.38400" wyse50 on secure
```

This example shows that the port on `/dev/ttyd4` has a `wyse50` terminal connected at 38400 bps with no parity (`std.38400` from `/etc/gettytab`, see `gettytab(5)`) and `root` logins are allowed (`secure`).

9. Why can I not run `tip` or `cu`?

On your system, the programs `tip(1)` and `cu(1)` are probably executable only by `uucp` and group `dialer`. You can use the group `dialer` to control who has access to your modem or remote systems. Just add yourself to group `dialer`.

Alternatively, you can let everyone on your system run `tip(1)` and `cu(1)` by typing:

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

10. My stock Hayes modem is not supported---what can I do?

Actually, the manual page for `tip(1)` is out of date. There is a generic Hayes dialer already built in. Just use `at=hayes` in your `/etc/remote` (see `remote(5)`) file.

The Hayes driver is not smart enough to recognize some of the advanced features of newer modems---messages like `BUSY`, `NO DIALTONE`, or `CONNECT 115200` will just confuse it. You should turn those messages off when you use `tip(1)` (using `ATX0&W`).

Also, the dial timeout for `tip(1)` is 60 seconds. Your modem should use something less, or else `tip` will think there is a communication problem. Try `ATS7=45&W`.

Actually, as shipped `tip(1)` does not yet support it fully. The solution is to edit the file `tipconf.h` in the directory `/usr/src/usr.bin/tip/tip`. Obviously you need the source distribution to do this.

Edit the line `#define HAYES 0` to `#define HAYES 1`. Then make `and` `make install`. Everything works nicely after that.

11. How am I expected to enter these AT commands?

Make what is called a “direct” entry in your `/etc/remote` file (see `remote(5)`). For example, if your modem is hooked up to the first serial port, `/dev/cuaa0`, then put in the following line:

```
cuaa0:dv=/dev/cuaa0:br#19200:pa=none
```

Use the highest bps rate your modem supports in the `br` capability. Then, type `tip cuaa0` (see `tip(1)`) and you will be connected to your modem.

If there is no `/dev/cuaa0` on your system, do this:

```
# cd /dev
# sh MAKEDEV cuaa0
```

Or use `cu` as `root` with the following command:

```
# cu -lline -sspeed
```

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

12. Why does the <@> sign for the pn capability not work?

The <@> sign in the phone number capability tells tip to look in /etc/phones for a phone number. But the <@> sign is also a special character in capability files like /etc/remote. Escape it with a backslash:

```
pn=\@
```

13. How can I dial a phone number on the command line?

Put what is called a “generic” entry in your /etc/remote file (see remote(5)). For example:

```
tip115200|Dial any phone number at 115200 bps:\
      :dv=/dev/cuaa0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600 bps:\
      :dv=/dev/cuaa0:br#57600:at=hayes:pa=none:du:
```

Then you can do something like `tip -115200 5551234`. If you prefer `cu(1)` over `tip(1)`, use a generic `cu` entry:

```
cu115200|Use cu to dial any number at 115200bps:\
      :dv=/dev/cuaa1:br#57600:at=hayes:pa=none:du:
```

and type `cu 5551234 -s 115200`.

14. Do I have to type in the bps rate every time I do that?

Put in an entry for `tip1200` or `cu1200`, but go ahead and use whatever bps rate is appropriate with the `br` capability. `tip(1)` thinks a good default is 1200 bps which is why it looks for a `tip1200` entry. You do not have to use 1200 bps, though.

15. How can I more easily access a number of hosts through a terminal server?

Rather than waiting until you are connected and typing `CONNECT host` each time, use tip’s `cm` capability. For example, these entries in /etc/remote (see remote(5)):

```
pain|pain.deep13.com|Forrester’s machine:\
      :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank’s machine:\
      :cm=CONNECT muffin\n:tc=deep13:
deep13|Gizmonics Institute terminal server:\
      :dv=/dev/cuaa2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

will let you type `tip pain` or `tip muffin` to connect to the hosts `pain` or `muffin`; and `tip deep13` to get to the terminal server.

16. Can tip try more than one line for each site?

This is often a problem where a university has several modem lines and several thousand students trying to use them...

Make an entry for your university in /etc/remote (see remote(5)) and use <\@> for the `pn` capability:

```
big-university:\
    :pn=\@:tc=dialout
dialout:\
    :dv=/dev/cuaa3:br#9600:at=courier:du:pa=none:
```

Then, list the phone numbers for the university in `/etc/phones` (see `phones(5)`):

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

`tip(1)` will try each one in the listed order, then give up. If you want to keep retrying, run `tip(1)` in a while loop.

17. Why do I have to hit **CTRL+P** twice to send **CTRL+P** once?

CTRL+P is the default “force” character, used to tell `tip(1)` that the next character is literal data. You can set the force character to any other character with the `~s` escape, which means “set a variable”.

Type `~sforce=single-char` followed by a newline. *single-char* is any single character. If you leave out *single-char*, then the force character is the nul character, which you can get by typing **CTRL+2** or **CTRL+SPACE**. A pretty good value for *single-char* is **SHIFT+CTRL+6**, which I have seen only used on some terminal servers.

You can have the force character be whatever you want by specifying the following in your `$HOME/.tiprc` file:

```
force=single-char
```

18. Why is everything I type suddenly in UPPER CASE?

You must have pressed **CTRL+A**, `tip(1)` “raise character”, specially designed for people with broken **Caps Lock** keys. Use `~s` as above and set the variable “raisechar” to something reasonable. In fact, you can set it to the same as the force character, if you never expect to use either of these features.

Here is a sample `.tiprc` file perfect for Emacs users who need to type **CTRL+2** and **CTRL+A** a lot:

```
force=^^
raisechar=^^
```

The ^^ is **SHIFT+CTRL+6**.

19. How can I do file transfers with `tip`?

If you are talking to another UNIX system, you can send and receive files with `~p` (put) and `~t` (take). These commands run `cat(1)` and `echo(1)` on the remote system to accept and send files. The syntax is:

```
~p <local-file> [<remote-file>]
~t <remote-file> [<local-file>]
```

There is no error checking, so you probably should use another protocol, like `zmodem`.

20. How can I run zmodem with **tip**?

First, install one of the zmodem programs from the ports collection (such as one of the two from the comms category, **lrzsz** or **rzsz**).

To receive files, start the sending program on the remote end. Then, press enter and type `~C rz` (or `~C lrz` if you installed **lrzsz**) to begin receiving them locally.

To send files, start the receiving program on the remote end. Then, press enter and type `~C sz files` (or `~C lsz files`) to send them to the remote system.

Example 16

ÄéÜöïñå ÆñùôÞóåéò

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.

Ðñïäéäïðíßçóç: 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.

Όχι ἂν: 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

ÊåöÜëáéï 17

× éïýìî ìñ êáé *FreeBSD*

1. How cool is FreeBSD?

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

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

Seriously, both FreeBSD and Linux use the HLT (halt) instruction when the system is idle thus lowering its energy consumption and therefore the heat it generates. Also if you have APM (advanced power management) configured, then FreeBSD can also put the CPU into a low power mode.

2. Who is scratching in my memory banks??

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

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

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

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

One thousand, one hundred and sixty-nine:

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

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

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

One to commit an untested lightbulb which breaks buildworld, then back it out five minutes later;
Eight to flame the PR originators for not including patches in their PRs;
Five to complain about buildworld being broken;
Thirty-one to answer that it works for them, and they must have cvsupped at a bad time;
One to post a patch for a new lightbulb to -hackers;
One to complain that he had patches for this three years ago, but when he sent them to -CURRENT they were just ignored, and he has had bad experiences with the PR system; besides, the proposed new lightbulb is non-reflexive;
Thirty-seven to scream that lightbulbs do not belong in the base system, that committers have no right to do things like this without consulting the Community, and WHAT IS -CORE DOING ABOUT IT!?
Two hundred to complain about the color of the bicycle shed;
Three to point out that the patch breaks style(9);
Seventeen to complain that the proposed new lightbulb is under GPL;
Five hundred and eighty-six to engage in a flame war about the comparative advantages of the GPL, the BSD license, the MIT license, the NPL, and the personal hygiene of unnamed FSF founders;
Seven to move various portions of the thread to -chat and -advocacy;
One to commit the suggested lightbulb, even though it shines dimmer than the old one;
Two to back it out with a furious flame of a commit message, arguing that FreeBSD is better off in the dark than with a dim lightbulb;
Forty-six to argue vociferously about the backing out of the dim lightbulb and demanding a statement from -core;
Eleven to request a smaller lightbulb so it will fit their Tamagotchi if we ever decide to port FreeBSD to that platform;
Seventy-three to complain about the SNR on -hackers and -chat and unsubscribe in protest;
Thirteen to post “unsubscribe”, “How do I unsubscribe?”, or “Please remove me from the list”, followed by the usual footer;
One to commit a working lightbulb while everybody is too busy flaming everybody else to notice;
Thirty-one to point out that the new lightbulb would shine 0.364% brighter if compiled with TenDRA (although it will have to be reshaped into a cube), and that FreeBSD should therefore switch to TenDRA instead of GCC;
One to complain that the new lightbulb lacks fairings;
Nine (including the PR originators) to ask “what is MFC?”;
Fifty-seven to complain about the lights being out two weeks after the bulb has been changed.
Nik Clayton <nik@FreeBSD.org> adds:
I was laughing quite hard at this.
And then I thought, “Hang on, shouldn’t there be ’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.

ÊâöÛëáéï 18

Đñĩ÷ ùñçìÝía ÈÝìáôá

1. How can I learn more about FreeBSD's internals?

At this time, there is only one book on FreeBSD-specific OS internals, namely “The Design and Implementation of the FreeBSD Operating System” by Marshall Kirk McKusick and George V. Neville-Neil, ISBN 0-201-70245-2, which focuses on version 5.X of FreeBSD.

Additionally, much general UNIX knowledge is directly applicable to FreeBSD.

For a list of relevant books, please check the Handbook's Operating System Internals Bibliography (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/bibliography-osinternals.html).

2. How can I contribute to FreeBSD?

Please see the article on Contributing to FreeBSD

(http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/contributing/article.html) for specific advice on how to do this. Assistance is more than welcome!

3. What are SNAPS and RELEASEs?

There are currently three active/semi-active branches in the FreeBSD CVS Repository (<http://www.FreeBSD.org/cgi/cvsweb.cgi>). (Earlier branches are only changed very rarely, which is why there are only three active branches of development):

- `RELENG_5` AKA *5-STABLE*
- `RELENG_6` AKA *6-STABLE*
- `HEAD` AKA *-CURRENT* AKA *7.X-CURRENT*

`HEAD` is not an actual branch tag, like the other two; it is simply a symbolic constant for “*the current, non-branched development stream*” which we simply refer to as “*-CURRENT*”.

Right now, “*-CURRENT*” is the 7.X development stream; the *5-STABLE* branch, `RELENG_5`, forked off from “*-CURRENT*” in October 2004, and the *6-STABLE* branch, `RELENG_6`, forked off from “*-CURRENT*” in November 2005.

4. How do I make my own custom release?

Please see the Release Engineering (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/releng/article.html) article.

5. Why does `make world` clobber my existing installed binaries?

Yes, this is the general idea; as its name might suggest, `make world` rebuilds every system binary from scratch, so you can be certain of having a clean and consistent environment at the end (which is why it takes so long).

If the environment variable `DESTDIR` is defined while running `make world` or `make install`, the newly-created binaries will be deposited in a directory tree identical to the installed one, rooted at `${DESTDIR}`. Some random combination of shared libraries modifications and program rebuilds can cause this to fail in `make world` however.

6. Why isn't `cvsup.FreeBSD.org` a round robin DNS entry to share the load amongst the various CVSup servers?

While CVSup mirrors update from the master CVSup server hourly, this update might happen at any time during the hour. This means that some servers have newer code than others, even though all servers have code that is less than an hour old. If `cvsup.FreeBSD.org` was a round robin DNS entry that simply redirected users to a random CVSup server, running CVSup twice in a row could download code older than the code already on the system.

7. Why does my system say “(bus speed defaulted)” when it boots?

The Adaptec 1542 SCSI host adapters allow the user to configure their bus access speed in software. Previous versions of the 1542 driver tried to determine the fastest usable speed and set the adapter to that. We found that this breaks some users' systems, so you now have to define the `TUNE_1542` kernel configuration option in order to have this take place. Using it on those systems where it works may make your disks run faster, but on those systems where it does not, your data could be corrupted.

8. Can I follow `-CURRENT` with limited Internet access?

Yes, you can do this *without* downloading the whole source tree by using the CTM facility (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/books/handbook/synching.html#CTM).

9. How did you split the distribution into 240k files?

Newer BSD based systems have a `-b` option to `split(1)` that allows them to split files on arbitrary byte boundaries. Here is an example from `/usr/src/Makefile`.

```
bin-tarball:
(cd ${DISTDIR}; \
tar cf - . \
gzip --no-name -9 -c | \
split -b 240640 - \
${RELEASEDIR}/tarballs/bindist/bin_tgz.)
```

10. I have written a kernel extension, who do I send it to?

Please take a look at the article on Contributing to FreeBSD (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/contributing/article.html) to learn how to submit code. And thanks for the thought!

11. How are Plug N Play ISA cards detected and initialized?

By: Frank Durda IV <uhclem@nemesis.lonestar.org>

In a nutshell, there a few I/O ports that all of the PnP boards respond to when the host asks if anyone is out there. So when the PnP probe routine starts, it asks if there are any PnP boards present, and all the PnP boards respond with their model # to a I/O read of the same port, so the probe routine gets a wired-OR “yes” to that question. At least one

bit will be on in that reply. Then the probe code is able to cause boards with board model IDs (assigned by Microsoft/Intel) lower than X to go “off-line”. It then looks to see if any boards are still responding to the query. If the answer was 0, then there are no boards with IDs above X. Now probe asks if there are any boards below x. If so, probe knows there are boards with a model numbers below X. Probe then asks for boards greater than X-(limit/4) to go off-line. If repeats the query. By repeating this semi-binary search of IDs-in-range enough times, the probing code will eventually identify all PnP boards present in a given machine with a number of iterations that is much lower than what 2^{64} would take.

The IDs are two 32-bit fields (hence 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 çääëññíéê ðßóóá ôâ÷íéêþí óðæçôþóáúí ôïö 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.

Óçîâßúóç: 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.*]

ÊåöÜëáéï 19

Ãõ÷ 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.
- The FreeBSD Handbook*, FreeBSD Documentation Project, BSDi, 1st edition, November 1999, 489 pages, ISBN 1-57176-241-8.
- [McKusick et al, 1994] *Berkeley Software Architecture Manual, 4.4BSD Edition*, M. K. McKusick, M. J. Karels, S. J. Leffler, W. N. Joy, & 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. Dobb's Journal, pp68-71, pp97-99.
- UNIX System Architecture*, Prentice-Hall, Inc, 1990, Prabhat K. Andleigh, ISBN 0-13-949843-5.

- Porting UNIX to the 386*, William Jolitz, Dr. 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* (<ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/FreeBSD-current/src/share/misc/bsd-family-tree>), 1997.

Absolute BSD, Michael Lucas, No Starch Press, June 2002, ISBN 1-886411-74-3.

The C/C++ Users Journal, R&D Publications Inc., ISSN 1075-2838.

Sys Admin — The Journal for UNIX System Administrators, Miller Freeman, Inc, ISSN 1061-2688.