Το HowTo του πυρήνα (kernel) του Linux

Brian Ward, bri@cs.uchicago.edu v1.0, 5 Ιουνίου 1999

Αυτός είναι ένας λεπτομερής οδηγός γιά τις ρυθμίσεις του πυρήνα, γιά compilation, αναβαθμίσεις, και λύσεις προβλημάτων σε Η/Υ τύπου ix86.

Contents

1 Εισαγωγή **3**

1.1 Πρώτα διαβάστε εδώ! (Και το εννοώ) 4

1.2 Δυο λόγια περί του στύλ 4

2 Σημαντικές ερωτήσεις, και οι απαντήσεις τους **4**

2.1 Τί, τέλος πάντων, κάνει ο πυρήνας; 4

2.2 Γιατί να αναβαθμίσω τον πυρήνα μου; 5

2.3 Τί είδους hardware υποστηρίζουν οι νέοι πυρήνες; 5

2.4 Ποιές εκδόσεις των gcc και libc χρειάζομαι; 5

2.5 Τί είναι ένα module που μπορεί να φορτωθεί (loadable); 5

2.6 Πόσο χώρο στον δίσκο χρειάζομαι; 5

2.7 Πόση ώρα κάνει; 5

3 Πώς να ρυθμίσετε πραγματικά τον πυρήνα **6**

3.1 Παίρνουμε τον πηγαίο (source) κώδικα 6

3.2 Ανοίγουμε τον source κώδικα 6

3.3 Ρυθμίζουμε τον πυρήνα 7

3.3.1 Απομίμηση του μαθηματικού συνεπεξεργαστή (Τύπος και χαρακτηριστικά του μ Pr) . . 7

3.3.2 Προχωρημένη υποστήριξη δίσκων MFM/RLL, IDE, cdrom (Block Devices) 7

3.3.3 Υποστήριξη δικτύωσης (Γενικές ρυθμίσεις) 8

3.3.4 System V IPC (Γενικές ρυθμίσεις) 8

3.3.5 Οικογένεια του processor (Τύπος και χαρακτηριστικά του processor) 8

3.3.6 Υποστήριξη SCSI 8

3.3.7 Υποστήριξη καρτών δικτύου 8

3.3.8 Συστήματα αρχείων (Filesystems) 8

3.3.9 Συσκευές χαρακτήρων (Character devices) 9

3.3.10 Ηχος 10

3.3.11 Άλλες παράμετροι ρυθμίσεων . . . 10
3.3.12 Χάκεμα του πυρήνα . . . 10
3.4 Και τώρα, τί; (Το αρχείο Makefile) . . . 10

4 Κάνοντας compile στον πυρήνα 11

4.1 Καθάρισμα και εξαρτώμενα αρχεία . . . 11
4.2 Διάρκεια του compile . . . 11
4.3 Άλλα αρχεία τύπου "make" . . . 11
4.4 Εγκατάσταση του πυρήνα . . . 12

5 Μπαλώνοντας (patching) τον πυρήνα 12

5.1 Εφαρμόζοντας ένα μπάλωμα (patch) . . . 12
5.2 Αν κάτι δεν πάει καλά . . . 13
5.3 Σβήνοντας τα .orig αρχεία . . . 14
5.4 Άλλα patches . . . 14

6 Επιπλέον πακέτα 14

6.1 kbd . . . 15
6.2 util-linux . . . 15
6.3 hdparm . . . 15
6.4 gpm . . . 15

7 Μερικές παγίδες 15

7.1 make clean . . . 15
7.2 Τεράστιοι ή αργοί πυρήνες . . . 15
7.3 Δεν δουλεύει η παράλληλη θύρα, ή ο εκτυπωτής μου . . . 16
7.4 Δεν γίνεται compile στον πυρήνα . . . 16
7.5 Η νέα έκδοση του πυρήνα δεν bootάρει . . . 16
7.6 Ξεχάσατε να τρέξετε το LILO, ή ο Η/Υ δεν bootάρει καθόλου . . . 17
7.7 Λέει "warning: bdflush not running" . . . 18
7.8 Δεν μπορώ να βάλω το IDE/ATAPI CD-ROM να δουλέψει . . . 18
7.9 Λέει παράξενα πράγματα γιά καταργημένες αιτήσεις routing . . . 18
7.10 Το firewalling δεν δουλεύει στην έκδοση 1.2.0 . . . 18
7.11 "Not a compressed kernel Image file" . . . 19
7.12 Προβλήματα με το τερματικό κονσόλας (console terminal) μετά την αναβάθμιση σε 1.3.x . . . 19
7.13 Δεν μπορώ να κάνω compile, αφού αναβάθμισα τον πυρήνα . . . 19

7.14 Αυξάνοντας τα όρια . . . 19

8 Σημείωση γιά αναβαθμίσεις προς έκδοση 2.0.x, 2.2.x 20

9 Modules 20

9.1 Εγκαθιστώντας τις utilities γιά τα modules . . . 20

9.2 Modules που διανέμονται μαζί με τον πυρήνα . . . 20

10 Συμβουλές και κόλπα 21

10.1 Επανακατεύθυνση της εξόδου των εντολών make και patch . . . 21

10.2 Εγκατάσταση του πυρήνα υπό προϋποθέσεις . . . 21

10.3 Ανανεώσεις του πυρήνα . . . 22

11 Άλλα σχετικά HOWTOs που μπορεί να φανούν χρήσιμα 22

12 Διάφορα 22

12.1 Συγγραφέας . . . 22

12.2 Προς ενέργεια . . . 23

12.3 Συνεισφορές . . . 23

12.4 Copyright, άδεια χρήσης, και τα παρόμοια . . . 24

1 Εισαγωγή

Είσαστε υποχρεωμένοι να διαβάσετε αυτό το κείμενο; Λοιπόν, ας δούμε αν παρουσιάζετε κάποιο από τ' ακόλουθα συμπτώματα :

- "Ωχ! Αυτό το τρελλοκατιτίς-46.5.6 πακέτο λέει πως χρειάζεται πυρήνα έκδοσης 2.8.193, κι εγώ ακόμη έχω την έκδοση 1.0.9!"
- Υπάρχει ένας driver ς' έναν από τους νέους πυρήνες, που πρέπει να έχετε οπωσδήποτε.
- Δεν έχετε καθόλου ιδέα γιά το πώς να κάνετε compile ς' έναν πυρήνα.
- "Είναι *πραγματικά* το README η μοναδική διαθέσιμη πηγή πληροφόρησης;"
- Το δοκιμάσατε, αλλά δεν δούλεψε.
- Πρέπει να έχετε κάποιες έτοιμες οδηγίες, να τις δίνετε ς' όσους επιμένουν να σας ζητάνε να τους εγκαταστήσετε τον πυρήνα.

1.1 Πρώτα διαβάστε εδώ! (Και το εννοώ)

Μερικά από τα παραδείγματα αυτού εδώ του κειμένου προϋποθέτουν ότι διαθέτετε τα κατά GNU `tar`, `find`, και `xargs`. Αυτά τα αρχεία είναι στάνταρ, άρα λογικά δεν θα σας προξενήσουν προβλήματα. Επίσης, προϋποτίθεται ότι γνωρίζετε τη δομή του συστήματος αρχείων (filesystem) του Η/Υ σας. Αν όχι, πρέπει να κρατήσετε ένα γραπτό αντίγραφο της (κατά τη διάρκεια της κανονικής λειτουργίας του Η/Υ) εξόδου της εντολής `mount`, ή των γραμμών του αρχείου `/etc/fstab` (αν μπορείτε να το διαβάσετε). Αυτές οι πληροφορίες είναι σημαντικές, και δεν αλλάζουν, αν δεν ξανακάνετε partition στον σκληρό σας δίσκο, αν δεν προσθέσετε έναν καινούργιο δίσκο, αν δεν κάνετε επανεγκατάσταση του Linux, ή κάποια παρόμοια ενέργεια.

Η πιο πρόσφατη έκδοση του πυρήνα "παραγωγής" κατά τον χρόνο γραφής αυτού του κειμένου είναι η 2.2.9 , που σημαίνει ότι οι αναφορές και τα παραδείγματα αντιστοιχούν ς' αυτήν την έκδοση. Όσο κι αν προσπάθησα να γράψω ένα κείμενο ανεξάρτητο από συγκεκριμένες εκδόσεις του πυρήνα, αυτός βρίσκεται υπό διαρκή εξέλιξη. Επομένως, αν πάρετε νεώτερη έκδοση, αναπόφευκτα θα έχει κάποιες διαφορές. Ξαναλέω ότι αυτό λογικά δεν θα δημιουργήσει μεγάλα προβλήματα, αλλά ίσως προξενήσει σύγχυση.

Υπάρχουν δύο εκδόσεις του source κώδικα του πυρήνα του Linux, η έκδοση "παραγωγής" και η έκδοση "ανάπτυξης". Οι εκδόσεις παραγωγής είναι όσες έχουνε ζυγούς και μικρούς αριθμούς. Πχ παραγωγής ήτανε η 1.2.x , η 2.0.x , καθώς και η 2.2.x . Αυτοί οι πυρήνες θεωρούνται (κατά τον χρόνο της κυκλοφορίας τους στην αγορά) οι πλέον σταθεροί και χωρίς bugs. Οι πυρήνες ανάπτυξης (2.1.x , 2.3.x , κλπ) προορίζονται γιά πυρήνες δοκιμών, γιά όσους χρήστες ενδιαφέρονται να τεστάρουν τους νέους (και πιθανότατα γεμάτους από bugs) πυρήνες. Σας προειδοποίησα!

1.2 Δυο λόγια περί του στύλ

`Κείμενο σαν αυτό` είναι κάτι που, είτε εμφανίζεται στην οθόνη σας, είτε είναι όνομα αρχείου, είτε κάτι που μπορείτε να πληκτρολογήσετε (πχ μιά εντολή), είτε παράμετροι μιας εντολής (αν κοιτάξετε ένα αρχείου καθαρού text, δεν διαφέρει). Οι εντολές (και γενικά η είσοδος) συχνά μπαίνουν ανάμεσα σε μονά εισαγωγικά ("), πράγμα που μας πάει στο εξής κλασικό πρόβλημα στίξης : Εάν μιά εντολή βρίσκεται στο τέλος Μιας πρότασης σε μονά εισαγωγικά, συνήθως όλοι γράφουνε και μιά τελεία μετά την εντολή, επειδή οι αμερικανικοί κανόνες στίξης λένε να βάζουμε την τελεία μέσα στα εισαγωγικά. Αν και η κοινή λογική (και δυστυχώς, αυτό προϋποθέτει ότι οι έχοντες "κοινή λογική" είναι συνηθισμένοι στο λεγόμενο αμερικάνικο στύλ εισαγωγικών) λέει ν' αφαιρούμε πρώτα την στίξη, τα περισσότερα άτομα απλά δεν το θυμούνται, επομένως σε τέτοιες περιπτώσεις θα τοποθετώ τη στίξη έξω απ' τα εισαγωγικά. Μ' άλλα λόγια, όταν δείχνω ότι πρέπει να πληκτρολογείτε πχ `"make config"`, θα γράφω `"make config"`, όχι `"make config."` . (Σ.τ.μ. : Στα Ελληνικά, η στίξη δεν ακολουθεί αυτούς τους κανόνες, άρα δεν θα χρειαστεί να θυμόμαστε τις συγκεκριμένες διευκρινήσεις.)

2 Σημαντικές ερωτήσεις, και οι απαντήσεις τους

2.1 Τί, τέλος πάντων, κάνει ο πυρήνας;

Ο πυρήνας των *nix's δρα ως μεσολαβητής μεταξύ των προγραμμάτων σας και του hardware. Πρώτα, χειρίζεται τη μνήμη (ή δίνει οδηγίες γιά τον χειρισμό της) γιά όλα τα προγράμματα που τρέχουν (processes), και σιγουρεύει το ότι όλα θα πάρουν δίκαιο (ή άδικο, αν προτιμάτε) μερίδιο από τον χρόνο του μικροπροσέσσορα. Επιπλέον, παρέχει μιά όμορφη και σε μεγάλο βαθμό μεταφέρσιμη διασύνδεση, ώστε τα προγράμματα να μιλήσουν στο hardware.

Φυσικά, η λειτουργία του πυρήνα ασχολείται με περισσότερα πράγματα από τα παραπάνω, αλλά αυτές οι βασικές λειτουργίες είναι οι πιο σημαντικές που πρέπει να γνωρίζετε.

2.2 Γιατί να αναβαθμίσω τον πυρήνα μου;

Οι νεώτεροι πυρήνες γενικά μας παρέχουν την ικανότητα να μιλάμε σε περισσότερους τύπους hardware (δηλαδή, έχουν περισσότερους drivers συσκευών), διευθύνουν καλύτερα τις διάφορες εργασίες, τρέχουν γρηγορότερα απ' τους παλιότερους, συνήθως είναι σταθερότεροι απ' τους παλιότερους, κι έχουν διορθωμένα τα χαζά λάθη των παλαιοτέρων εκδόσεων. Οι περισσότεροι χρήστες ανανεώνουν τους πυρήνες, επειδή θέλουν να έχουν τους νέους drivers συσκευών και την εξάλειψη των bugs.

2.3 Τί είδους hardware υποστηρίζουν οι νέοι πυρήνες;

Διαβάστε το Hardware-HOWTO. Εναλλακτικά, μπορείτε να ψάξετε το αρχείο `config.in` στον source κώδικα του Linux, ή απλά να ανακαλύψετε τί γίνεται, όταν δίνετε `make config`. Αυτό σας δείχνει όλα τα είδη hardware που υποστηρίζει ο στάνταρ πυρήνας της διανομής σας, αλλά όχι κι όλα όσα υποστηρίζει το Linux. Αρκετοί συνηθισμένοι drivers συσκευών (όπως πχ γιά τις κάρτες PCMCIA και μερικές μονάδες μαγνητικής ταινίας) είναι modules που φορτώνονται (loadable), και οι προγραμματιστές τους τους γράφουν και τους διανέμουν χωριστά.

2.4 Ποιές εκδόσεις των gcc και libc χρειάζομαι;

Στο αρχείο `README`, που περιέχεται μέσα στον source κώδικα του Linux, ο Linus συνιστά την κατάλληλη έκδοση του gcc. Εάν, πάλι, δεν έχετε την κατάλληλη έκδοση της libc, η τεκμηρίωση της συνιστώμενης έκδοσης της gcc θα σας πει αν πρέπει ν' αναβαθμίσετε τη libc σας. Αυτή δεν είναι δύσκολη διαδικασία, αλλά είναι σημαντικό ν' ακολουθήσετε ακριβώς τις οδηγίες.

2.5 Τί είναι ένα module που μπορεί να φορτωθεί (loadable);

Τα modules αυτά είναι κομμάτια κώδικα του πυρήνα, που δεν ενσωματώνονται (linked) κατευθείαν στον πυρήνα. Τα κάνουμε compile χωριστά, και μπορούμε να τα βάλουμε στον και να τα βγάλουμε από τον πυρήνα (ενόσω τρέχει), σχεδόν οποιαδήποτε στιγμή. Επειδή έχει ευελιξία, αυτός είναι σήμερα ο προτιμώμενος τρόπος γιά να γράφουμε σε κώδικα κάποιες δυνατότητες του πυρήνα. Πολλοί δημοφιλείς drivers συσκευών, όπως πχ οι drivers των καρτών PCMCIA και της μονάδας μαγνητικής ταινίας QIC-80/40, είναι modules που φορτώνονται.

2.6 Πόσο χώρο στον δίσκο χρειάζομαι;

Εξαρτάται από τη συγκεκριμένη σύνθεση του Η/Υ σας. Πρώτα, ο συμπιεσμένος source κώδικας του Linux είναι σχεδόν 14 MB ytes στην έκδοση 2.2.9 . Αποσυμπιεσμένος και φτιαγμένος γιά μιά συνηθισμένη σύνθεση, καταλαμβάνει άλλα 67 MB.

2.7 Πόση ώρα κάνει;

Με νεώτερους Η/Υ, η compilation χρειάζεται εντυπωσιακά λιγότερο χρόνο, απ' ό,τι με παλιότερους. Ενας AMD K6-2/300 με γρήγορο δίσκο μπορεί να φτιάξει πυρήνα 2.2.x σε περίπου τέσσερα λεπτά. Εάν σχεδιάζετε να κάνετε compilation σε παλιότερους Pentiums, 486s, ή 386s, να προετοιμαστείτε γιά μεγάλη αναμονή, πιθανόν ωρών ή ημερών...

Αν αυτό σας προβληματίζει, και συμβαίνει να διαθέτετε γιά compile έναν γρηγορότερο Η/Υ κάπου εκεί κοντά, μπορείτε να φτιάξετε τον πυρήνα στο γρήγορο μηχάνημα (υποθέτοντας ότι του δίνετε τις σωστές παραμέτρους, ότι οι utilities που χρησιμοποιείτε είναι πρόσφατες, κλπ), και μετά να μεταφέρετε το image του πυρήνα στο αργότερο μηχάνημα.

3 Πώς να ρυθμίσετε πραγματικά τον πυρήνα

3.1 Παίρνουμε τον πηγαίο (source) κώδικα

Μπορούμε να βρούμε τον source κώδικα με anonymous ftp στο `ftp.kernel.org`, στο dir `/pub/linux/kernel/vx.y`, όπου `x.y` είναι ο αριθμός έκδοσης (πχ 2.2), και - όπως προείπαμε - οι μονοί αριθμοί στο τέλος δείχνουν εκδόσεις ανάπτυξης, που μπορεί να είναι ασταθείς. Το αρχείο τυπικά ονομάζεται `linux-x.y.z.tar.gz`, όπου `x.y.z` είναι ο αριθμός έκδοσης. Τα διάφορα παρόμοια sites συχνά έχουνε κι ένα αρχείο με κατάληξη `.bz2`, δηλ. που συμπιέστηκε με το πρόγραμμα bzip2. (Αυτά τα αρχεία είναι μικρότερα, και χρειάζονται λιγότερο χρόνο να κατεβούν.)

Καλύτερα είναι να χρησιμοποιούμε το `ftp.xx.kernel.org`, όπου `xx` είναι ο κώδικας της χώρας σας. Γιά παράδειγμα, το `ftp.at.kernel.org` είναι γιά την Αυστρία, και το `ftp.us.kernel.org` γιά τις ΗΠΑ.

3.2 Ανοίγουμε τον source κώδικα

Κάνουμε login ως (ή `su` προς) `root`, και κάνουμε `cd` στο directory `/usr/src`. Εάν εγκαταστήσατε τον source κώδικα του πυρήνα (όπως κάνουν οι περισσότεροι χρήστες), θα υπάρχει ήδη ένα dir που λέγεται `linux`, που περιέχει όλο το δέντρο του παλιού source. Αν διαθέτετε αρκετό χώρο στον δίσκο και θέλετε να ενεργήσετε εκ του ασφαλούς, κρατήστε αυτό το directory. Μιά καλή ιδέα είναι να βρείτε ποιά έκδοση χρησιμοποιεί ο Η/Υ σας τώρα, και ανάλογα ν' αλλάξετε το όνομα του dir. Η εντολή `uname -r` δείχνει την παρούσα έκδοση του πυρήνα. Επομένως, αν η `uname -r` λέει `1.0.9`, θα αλλάξετε το όνομα (με την εντολή `mv`) `linux` σε `linux-1.0.9`. Αν αισθάνεστε μετρίως ανυπόμονοι, απλά σβήστε όλο το συγκεκριμένο directory. Σε κάθε περίπτωση, πάντως, και πριν αποσυμπιέσετε τον πλήρη source κώδικα, βεβαιωθείτε πως δεν υπάρχει directory `linux` στο `/usr/src`.

Τώρα, μέσα στο `/usr/src`, αποσυμπιέστε τον κώδικα με την εντολή `tar zxpvf linux-x.y.z.tar.gz`. (Αν έχετε αρχείο απλά `.tar`, χωρίς `.gz` στο τέλος, δώστε `tar xpvf linux-x.y.z.tar`.) Θα δείτε να περνάνε μπροστά σας γρήγορα τα περιεχόμενα του source κώδικα. Όταν τελειώσετε, θα υπάρχει ένα νέο `linux` directory μέσα στο `/usr/src`. Γυρίστε με `cd` στο `linux`, και ψάξτε προσεκτικά το αρχείο `README`. Θα βρείτε μιά ενότητα με την ταμπέλα `INSTALLING the kernel` (= εγκαθιστώντας τον πυρήνα). Εφαρμόστε τις σχετικές οδηγίες όταν πρέπει - συμβολικοί δεσμοί (symbolic links) που πρέπει να παραμείνουν στη θέση τους, αφαίρεση παλιών αρχείων `.o`, κλπ.

Αν έχετε αρχείο `.bz2` και το πρόγραμμα bzip2 (διαβάστε σχετικά μ' αυτό στο `http://www.muraroa.demon.co.uk/`), δώστε :

```
bz2cat linux-x.y.z.tar.bz2 | tar xvf -
```

3.3 Ρυθμίζουμε τον πυρήνα

Σημείωση : Μερικά από τα παρακάτω αποτελούν επανάληψη / διευκρίνηση μιας παρόμοιας ενότητας στο αρχείο `README` του Linus.

Η εντολή `make config`, όσο βρισκόμαστε στο `/usr/src/linux`, ξεκινάει ένα script ρυθμίσεων, που σας κάνει πολλές ερωτήσεις. Το script αυτό απαιτεί το bash, επομένως επαληθεύστε ότι το bash βρίσκεται στο `/bin/bash`, στο `/bin/sh`, ή στο `$BASH`.

Ωστόσο, υπάρχουν μερικές πιο ευχάριστες εναλλακτικές λύσεις στο `make config`, και πιθανόν να τις βρείτε ευκολότερες και πιο άνετες στη χρήση τους. Το `make menuconfig` είναι πιθανότατα το ευρύτερα χρησιμοποιούμενο. Ο,τι και να διαλέξετε, πάντως, είναι καλύτερο να εξοικειωθείτε με το συγκεκριμένο, επειδή θα επιστρέψετε ς' αυτό συντομότερα απ' όσο φανταζόσαστε. Όσοι τρέχετε τα X-Windows, μπορείτε να δοκιμάσετε το `make xconfig`, ε, να έχετε εγκαταστήσει το Tk ("click-o-rama" - Nat). Το `make menuconfig` είναι γιά όσους διαθέτουν το (n)curses και θα προτιμούσανε ένα text μενού. Αυτά τα περιβάλλοντα διαθέτουν ένα καθαρό πλεονέκτημα : Αν κάνεις λάθος επιλογή κατά τη διάρκεια των ρυθμίσεων, είναι απλή δουλειά το να επιστρέψεις και να τη διορθώσεις.

Με τα `make menuconfig` και `make xconfig`, οι επιλογές ρυθμίσεων εμφανίζονται ιεραρχικά.

Τώρα, είσαστε έτοιμοι ν' απαντήσετε στις ερωτήσεις, συνήθως με `y` (yes, ναι), ή `n` (no, όχι). Οι drivers των συσκευών συνήθως έχουνε μιά παράμετρο `m`. Αυτή σημαίνει "module", δηλαδή ο Η/Υ θα την κάνει compile, όχι όμως κατευθείαν μέσα στον πυρήνα, αλλά σαν module που φορτώνεται. Ενας κωμικότερος τρόπος να πεις τί σημαίνει το m, είναι "maybe". (= Ίσως.) Οι παράμετροι με προφανέστερη χρήση και που δεν είναι τόσο βασικές, δεν θα συζητηθούν εδώ. Γιά σύντομη περιγραφή μερικών ακόμη, διαβάστε την ενότητα "Άλλες επιλογές ρυθμίσεων". Στο `make menuconfig`, το πλήκτρο του κενού διαστήματος εναλλάσσει την επιλογή.

Στους πυρήνες 2.0.x (και σε μεταγενέστερους) υπάρχει και επιλογή "?", η οποία δίνει σύντομη περιγραφή των παραμέτρων ρύθμισης. Αυτές οι πληροφορίες πιθανότατα είναι και οι πλέον πρόσφατα ενημερωμένες. Ορίστε και μιά λίστα μερικών σημαντικών χαρακτηριστικών, σε ποιό ιεραρχικό μενού βρίσκονται, καθώς και μιά σύντομη περιγραφή τους.

3.3.1 Απομίμηση του μαθηματικού συνεπεξεργαστή (Τύπος και χαρακτηριστικά του μ Pr)

Αν δεν έχετε μαθηματικό συνεπεξεργαστή (δηλ. έχετε σκέτο 386 ή 486SX Η/Υ), πρέπει ν' απαντήσετε `y` στη σχετική ερώτηση. Αν έχετε συνεπεξεργαστή, και παρ' όλ' αυτά απαντήσετε `y`, μην ανησυχείτε ιδιαίτερα - ο συνεπεξεργαστής θα χρησιμοποιηθεί, και το πρόγραμμα απομίμησής του θ' αγνοηθεί. Γιά οποιοδήποτε σχεδόν μοντέρνο μηχάνημα η απάντηση πρέπει να είναι no, αλλά μην ανησυχείτε αν απαντήσετε yes κατά λάθος. Αν δεν χρειαστεί, δεν θα χρησιμοποιηθεί.

3.3.2 Προχωρημένη υποστήριξη δίσκων MFM/RLL, IDE, cdrom (Block Devices)

Πιθανότατα θα την χρειαστείτε. Σημαίνει ότι ο πυρήνας θα υποστηρίξει στάνταρ σκληρούς δίσκους γιά PC, τους οποίους διαθέτουν οι περισσότεροι χρήστες. Ο συγκεκριμένος driver δεν υποστηρίζει μονάδες SCSI. Μ' αυτές θ' ασχοληθούμε πιο κάτω.

Μετά θα ερωτηθείτε σχετικά με drivers "old (παλιούς) disk-only" και "new (νέους) IDE". Σίγουρα πρέπει να διαλέξετε τον έναν απ' τους δυο. Η κύρια διαφορά τους είναι πως ο παλιός driver υποστηρίζει μόνο δύο δίσκους σε μονή διασύνδεση (δηλ. ένα μοναδικό διπλό καλώδιο δίσκων), και ο νέος υποστηρίζει δευτερεύουσα διασύνδεση (δηλ. δύο διπλά καλώδια) και μονάδες cd-rom IDE/ATAPI. Ο νέος driver is 4k μεγαλύτερος από τον παλιό, και θεωρητικά είναι και "βελτιωμένος", που σημαίνει ότι (εκτός από το να περιέχει διαφορετικό αριθμό bugs) μάλλον θα βελτιώσει την απόδοση του δίσκου σας, ειδικά αν διαθέτετε νεώτερο hardware (τύπου EIDE).

3.3.3 Υποστήριξη δικτύωσης (Γενικές ρυθμίσεις)

Στη θεωρία, πρέπει ν' απαντήσετε y μόνον αν ο Η/Υ σας βρίσκεται συνδεδεμένος ς' ένα δίκτυο (όπως πχ το Internet), ή θέλετε να χρησιμοποιήσετε τα SLIP, PPP, term, κλπ, γιά να συνδεθείτε μέσω τηλεφώνου στο Internet. Ωστόσο, επειδή που αρκετά πακέτα (όπως τα X Windows) απαιτούν δικτυακή υποστήριξη, ακόμη κι αν ο Η/Υ σας δεν βρίσκεται σε κάποιο δίκτυο, πρέπει ν' απαντήσετε y. Αργότερα, θα ερωτηθείτε αν θέλετε υποστήριξη δικτύωσης τύπου TCP/IP. Ξαναδώστε y και εδώ, αν δεν είσαστε απόλυτα βέβαιοι.

3.3.4 System V IPC (Γενικές ρυθμίσεις)

Ενας από τους καλύτερους ορισμούς της IPC (Interprocess Communication, επικοινωνία διαδικασιών) βρίσκεται στο γλωσσάριο, στο βιβλίο της Perl. Κάτι που δεν εκπλήσσει, επειδή αρκετοί προγραμματιστές σε Perl χρησιμοποιούν την IPC γιά να βάλουν τις processes να μιλήσουν η μιά στην άλλη, και παρόμοια γιά μερικά άλλα προγράμματα (με γνωστότερο παράδειγμα το DOOM), άρα δεν είναι καλή ιδέα ν' απαντήσετε n, εκτός αν γνωρίζετε ακριβώς τί κάνετε.

3.3.5 Οικογένεια του processor (Τύπος και χαρακτηριστικά του processor)

(Σε παλιότερους πυρήνες : Δώστε την παράμετρο -m486 γιά βελτιστοποίηση ειδικά γιά τον 486.)

Παραδοσιακά, εδώ κάνουμε compile με βελτιστοποίηση γιά συγκεκριμένο προσέσσορα. Παλιότερα, οι πυρήνες τρέχανε μιά χαρά σε διαφορετικούς προσέσσορες, αλλά ήσαν μεγαλύτεροι. Στους νεώτερους πυρήνες αυτό δεν ισχύει, επομένως πρέπει να πείτε γιά ποιόν προσέσσορα κάνετε compile τον πυρήνα. Ενας πυρήνας γιά "386" θα δουλέψει μ' αυτόν και κάθε νεώτερο τύπο ix86 Η/Υ.

3.3.6 Υποστήριξη SCSI

Αν έχετε συσκευές SCSI devices, απαντήστε y. Θα ερωτηθείτε γιά τις υπόλοιπες πληροφορίες, πχ υποστήριξη cd-rom, δίσκων, και τί είδους SCSI κάρτα διαθέτετε. Γιά περισσότερες λεπτομέρειες, διαβάστε το SCSI-HOWTO.

3.3.7 Υποστήριξη καρτών δικτύου

Αν έχετε κάρτα δικτύου, ή θέλετε να χρησιμοποιήσετε τα SLIP και PPP, ή μιά παράλληλη θύρα γιά σύνδεση με το Internet, δώστε y. Το script ρυθμίσεων θα σας ρωτήσει τί είδος κάρτας έχετε, και ποιό πρωτόκολλο να χρησιμοποιήσει.

3.3.8 Συστήματα αρχείων (Filesystems)

Μετά, το script ρυθμίσεων ρωτάει αν θέλετε υποστήριξη γιά τα εξής filesystems :

Το στάνταρ (minix) - Οι νεώτερες διανομές δεν δημιουργούν minix filesystems, και πολλοί χρήστες δεν το χρησιμοποιούν, αλλά ίσως να εξακολουθεί να χρειάζεται τούτο εδώ. Το χρησιμοποιούν μερικά προγράμματα σε "δισκέτα διάσωσης", αλλά κι άλλου είδους δισκέτες μπορεί να έχουν minix filesystem, μιά που είναι εύκολο να το χρησιμοποιούμε σε δισκέτες.

Το second extended - Αυτό είναι το στάνταρ filesystem του Linux. Εσείς μάλλον σίγουρα έχετε ένα τέτοιο, και πρέπει να απαντήσετε `y`.

Το msdos - Αν θέλετε να χρησιμοποιήσετε και τις MS-DOS partitions του σκληρού σας δίσκου, ή να κάνετε mount σε MS-DOS δισκέτες, δώστε `y`.

Υπάρχουν διαθέσιμοι και τύποι filesystems από διάφορα άλλα ΛΣ.

Το /proc - (Αυτό ήταν ιδέα των Bell Labs, υποθέτω.) Δεν φτιάχνουμε proc filesystem επάνω ς' έναν δίσκο : Αυτό είναι η διασύνδεση του filesystem γιά τον πυρήνα και τις διαδικασίες (processes). Το χρησιμοποιούν πολλά προγράμματα λίστας διαδικασιών (process listers), όπως πχ το `ps`. Δοκιμάστε πχ να δώσετε `cat /proc/meminfo`, ή `cat /proc/devices`. Μερικά κελύφη (shells), ειδικά το rc, χρησιμοποιούν το `/proc/self/fd` (γνωστό ως `/dev/fd` σε άλλα ΛΣ) γιά είσοδο - έξοδο (I/O). Σχεδόν με βεβαιότητα πρέπει να πείτε `y` εδώ. Πολλά σημαντικά εργαλεία του Linux βασίζονται ς' αυτό.

Το NFS - Αν ο Η/Υ σας είναι συνδεδεμένος ς' ένα δίκτυο, και θέλετε να χρησιμοποιείτε τα filesystems των άλλων Η/Υ που επίσης έχουν NFS, δώστε `y`.

Το ISO9660 - Βρίσκεται στα περισσότερα CD-ROMs. Αν έχετε μονάδα CD-ROM, και θέλετε να τη βλέπει το Linux, απαντήστε `y`.

Μα, δεν ξέρω ποιά φιλεσψστεμς χρειάζομαι! Εντάξει, δώστε `mount`. Η έξοδος της εντολής θα είναι κάπως έτσι :

```
blah# mount
/dev/hda1 on / type ext2 (defaults)
/dev/hda3 on /usr type ext2 (defaults)
none on /proc type proc (defaults)
/dev/fd0 on /mnt type msdos (defaults)
```

Κοιτάξτε σε κάθε γραμμή : Η λέξη δίπλα στην `type` δείχνει τον τύπο του filesystem. Στο παράδειγμά μας τα filesystems `/` και `/usr` είναι second extended, χρησιμοποιώ το `/proc`, και υπάρχει και μιά δισκέτα που έκανα mount, η οποία είναι του msdos (μπλιάχ!) filesystem.

Δοκιμάστε να δώσετε `cat /proc/filesystems`, αν έχετε ενεργοποιήσει το `/proc` : Θα δώσει μιά λίστα των filesystems που έχει ο πυρήνας σας.

Η ρύθμιση σπανίων, μη σημαντικών filesystems παραφουσκώνει τον πυρήνα. Διαβάστε στην ενότητα γιά τα modules πώς το αποφεύγουμε αυτό, και την ενότητα "Παγίδες" γιά το γιατί ένας παραφουσκωμένος πυρήνας είναι ανεπιθύμητος.

3.3.9 Συσκευές χαρακτήρων (Character devices)

Εδώ ενεργοποιείτε τους drivers γιά τον εκτυπωτή σας (δηλ. τον παράλληλο εκτυπωτή), γιά το ποντίκι τύπου busmouse ή PS/2 (πολλοί φορητοί Η/Υ χρησιμοποιούν το πρωτόκολλο PS/2 γιά τις trackballs που έχουνε), γιά μερικές μονάδες μαγνητικής ταινίας, και γιά άλλες παρόμοιες συσκευές "χαρακτήρων". Δώστε `y` γιά όποια συσκευή έχετε.

Σημείωση : το gpm είναι ένα πρόγραμμα, που ενεργοποιεί το ποντίκι (έξω από το περιβάλλον των X-Windows) γιά cut και paste ανάμεσα σε (virtual) κονσόλες. Πάει μιά χαρά με τα σειριακά ποντίκια, επειδή μπορεί να συνυπάρχει καλά με τα X, αλλά χρειάζονται ειδικά κόλπα γιά τους άλλους τύπους ποντικιών.

3.3.10 Ηχος

Αν έχετε ιδιαίτερη επιθυμία ν' ακούσετε το `biff` να γαβγίζει, δώστε `y`, και μετά θα πείτε στο πρόγραμμα ρυθμίσεων ό,τι έχει σχέση με την κάρτα ήχου σας. (Μιά σημείωση γιά τις ρυθμίσεις της κάρτας ήχου : Όταν σας ρωτήσει αν θέλετε να εγκαταστήσετε την πλήρη έκδοση του driver, μπορείτε ν' απαντήσετε `n` και να εξοικονομήσετε κάμποση μνήμη στον πυρήνα, διαλέγοντας μόνο τα χαρακτηριστικά που θεωρείτε απαραίτητα.)

Αν θέλετε να κάνετε σοβαρή υποστήριξη στην κάρτα ήχου, κοιτάξτε και τους δωρεάν drivers στο `http://www.linux.org.uk/OSS/`, και τους εμπορικούς του Open Sound System (OSS), στο `http://www.opensound.com/`.

3.3.11 Άλλες παράμετροι ρυθμίσεων

Εδώ δεν παρατίθενται όλες οι επιλογές ρυθμίσεων, επειδή αλλάζουν αρκετά συχνά, ή επειδή είναι σχετικά αυτονόητες (πχ η υποστήριξη γιά 3Com 3C509 σημαίνει να κάνουμε compile τον driver της συγκεκριμένης κάρτας ethernet.) Υπάρχει μιά αρκετά περιεκτική λίστα όλων των επιλογών (συν ένας τρόπος να τις εισάγουμε στο script `Configure`), από μιά προσπάθεια που άρχισε και συντηρείται από τον Axel Boldt (`boldt@math.ucsb.edu`), και πρόκειται γιά την online βοήθεια. Αυτή η λίστα είναι επίσης διαθέσιμη σε μορφή ενός τεράστιου αρχείου (που βρίσκεται στην έκδοση 2.0), του `Documentation/Configure.help` στο δέντρο των dirs του source κώδικα του πυρήνα του Linux σας.

3.3.12 Χάκεμα του πυρήνα

>Από το README του Linus :

Οι λεπτομέρειες της ρύθμισης γιά "χάκεμα του πυρήνα (kernel hacking)" συνήθως δημιουργούν έναν μεγαλύτερο ή αργότερο πυρήνα (ή και τα δύο), και μπορούν ακόμη να κάνουν τον πυρήνα λιγότερο σταθερό, όταν ρυθμίζουμε ρουτίνες, που πάντα ψάχνουν στον κακογραμμένο κώδικα γιά να βρουν τα προβλήματα του πυρήνα (η kmalloc()). Επομένως, πιθανότατα να πρέπει ν' απαντήσετε "n" σε σχετικές ερωτήσεις γιά πυρήνα "παραγωγής".

3.4 Και τώρα, τί; (Το αρχείο Makefile)

Αφού τελειώσετε τις ρυθμίσεις, ένα μήνυμα σας λέει ότι ο πυρήνας σας είναι ρυθμισμένος, και να "ελέγξετε το επάνω-επάνω επίπεδο του `Makefile` γιά επιπλέον ρυθμίσεις (check the top-level `Makefile` for additional configuration)", κλπ.

Αρα, κοιτάζουμε το `Makefile`. Πιθανότατα να μη χρειαστεί να το αλλάξουμε, αλλά ποτέ δεν βλάπτει μιά ματιά. Μπορούμε επίσης ν' αλλάξουμε τις επιλογές του όταν ετοιμαστεί ο νέος πυρήνας, με την εντολή `rdev`. Εάν αισθάνεστε στα χαμένα κοιτάζοντας το αρχείο αυτό, τότε μην ασχοληθείτε.

4 Κάνοντας compile στον πυρήνα

4.1 Καθάρισμα και εξαρτώμενα αρχεία

Όταν τελειώσει η δουλειά του script ρυθμίσεων, σας λέει επίσης να `make dep` (= δημιουργείστε τις εξαρτήσεις) και (πιθανά) `clean` (= καθαρίστε). Επομένως, δώστε `make dep`. Αυτό θα επιβεβαιώσει πως όλα τα εξαρτώμενα αρχεία, όπως πχ τα include, βρίσκονται στη θέση τους. Αυτή η δουλειά δεν παίρνει πολύ χρόνο, εκτός αν ο Η/Υ σας είναι αρκετά αργός. Γιά παλιότερες εκδόσεις του πυρήνα, όταν τελειώσετε πρέπει να δώσετε και `make clean`. Αυτό σβήνει όλα τα object αρχεία και μερικά άλλα πράγματα, που οι παλιότερες εκδόσεις αφήνουν πίσω τους. Σε οποιαδήποτε περίπτωση, *μην* ξεχνάτε αυτό το βήμα πριν προσπαθήσετε να κάνετε compile έναν πυρήνα.

4.2 Διάρκεια του compile

Μετά το `depend` και το `clean`, τώρα μπορείτε να δώσετε `make bzImage`, ή `make bzdisk`. (Αυτό είναι το μέρος της εργασίας που παίρνει τον περισσότερο χρόνο.) Το `make bzImage` κάνει compile τον πυρήνα, και αφήνει ένα αρχείο στο directory `arch/i386/boot`, με το όνομα (μεταξύ άλλων) `bzImage`. Αυτός είναι ο νέος πυρήνας, σε συμπιεσμένη μορφή. Το `make bzdisk` κάνει τα ίδια πράγματα, αλλά επίσης γράφει το καινούργιο `bzImage` σε μιά δισκέτα, την οποία υποθέτω πως έχετε τοποθετήσει στη μονάδα "A:". Το `bzdisk` είναι αρκετά ευκολοχείριστο, γιά δοκιμές νέων πυρήνων. Αν τα κάνει θάλασσα (ή απλά δεν δουλέψει σωστά), απλά βγάλτε τη δισκέτα και επανεκκινήστε με τον παλιό πυρήνα σας. Μπορεί επίσης ν' αποτελέσει έναν εύκολο τρόπο να bootάρετε, αν από λάθος σβήσετε τον παλιό σας πυρήνα (ή κάνετε κάτι εξ ίσου φοβερό). Μπορείτε επίσης να το χρησιμοποιήσετε γιά νέες εγκαταστάσεις, όταν απλά μεταφέρετε τα περιεχόμενα ενός σκληρού ς' έναν άλλον. ("Ολ' αυτά και περισσότερα ακόμη! Πόσα θα πληρώνατε ΤΩΡΑ γι' αυτά;")

Όλοι οι (ακόμη και με ορισμένη παραδοχή) πρόσφατοι πυρήνες είναι συμπιεσμένοι, εκ ού και το `bz` μπροστά στα ονόματα. Ενας συμπιεσμένος πυρήνας αυτόματα αποσυμπιέζεται, όταν εκτελείται.

Σε παλιότερους πυρήνες, δεν υπάρχει επιλογή να φτιάξετε `bzImage`. Το όνομα ήταν απλά `zImage`. Προς το παρόν, η επιλογή γιά ασυμπίεστο πυρήνα εξακολουθεί να υπάρχει, ωστόσο, με το μέγεθος που έχουν οι νέοι πυρήνες, είναι σχεδόν υποχρεωτικό να τον φτιάξετε συμπιεσμένο.

4.3 Άλλα αρχεία τύπου "make"

Το `make mrproper` κάνει πιο εκτεταμένο `clean`, το οποίο μερικές φορές είναι απαραίτητο. Μπορεί να θέλετε να το κάνετε μετά από κάθε μπάλωμα (patch). Το `make mrproper` θα σβήσει και το αρχείο των ρυθμίσεών σας, άρα ίσως πρέπει να κάνετε ένα backup αντίγραφό του (`.config`), αν δείτε πως κάτι τέτοιο σας είναι χρήσιμο.

Το `make oldconfig` θα προσπαθήσει να ρυθμίσει τον πυρήνα από ένα παλιό αρχείο ρυθμίσεων. Θα διατρέξει τη διαδικασία `make config`. Αν δεν έχετε ξανακάνει compile κάποιον πυρήνα, ή δεν έχετε κάποιο παλιό αρχείο ρυθμίσεων, τότε πιθανότατα δεν πρέπει να κάνετε κάτι τέτοιο, καθώς πιθανότατα θα θελήσετε ν' αλλάξετε την default ρύθμιση.

Διαβάστε την ενότητα γιά τα modules, γιά περιγραφή του `make modules`.

4.4 Εγκατάσταση του πυρήνα

Αφού αποκτήσετε έναν νέο πυρήνα, που φαίνεται να δουλεύει όπως σας αρέσει, είναι ώρα να τον εγκαταστήσετε. Οι περισσότεροι χρήστες χρησιμοποιούν το LILO (Linux Loader) γιά τη δουλειά αυτή. Το `make bzlilo` θα εγκαταστήσει τον πυρήνα, θα τρέξει το LILO, θα σας ετοιμάσει να bootάρετε, ΑΛΛΑ ΜΟΝΟΝ εάν το lilo έχει ρυθμιστεί στον Η/Υ σας με τον ακόλουθο τρόπο : ο πυρήνας είναι ο `/vmlinuz`, το lilo βρίσκεται στο `/sbin`, και το αρχείο ρυθμίσεων του lilo (`/etc/lilo.conf`) συμφωνεί με όλα αυτά.

Διαφορετικά, πρέπει να χρησιμοποιήσετε το LILO κατευθείαν. Είναι σχετικά εύκολο πρόγραμμα να το εγκαταστήσετε και να δουλέψετε μ' αυτό, αλλά έχει μιά τάση να μπερδεύει τους ανθρώπους με το αρχείο ρυθμίσεών του. Κοιτάξτε αυτό το αρχείο ρυθμίσεων (`/etc/lilo/config` γιά παλιότερες εκδόσεις του Linux, `/etc/lilo.conf` στις νέες εκδόσεις), και δείτε πώς είναι γραμμένο. Πρέπει να μοιάζει κάπως έτσι :

```
image = /vmlinuz
label = Linux
root = /dev/hda1
...
```

Η παράμετρος `image =` δείχνει τον εγκατεστημένο πυρήνα, και οι περισσότεροι χρήστες γράφουν `/vmlinuz`. Η `label` (= ετικέτα) χρησιμοποιείται από το lilo γιά να καταλάβει με ποιόν πυρήνα ή με ποιό ΛΣ θα bootάρει, και το `root` είναι το αντίστοιχο του directory `/` γιά το συγκεκριμένο ΛΣ. Πάρτε ένα backup του παλιού σας πυρήνα και αντιγράψτε στο σωστό μέρος το `bzImage` που μόλις φτιάξατε. (Πρέπει να δώσετε `cp bzImage /vmlinuz`, αν χρησιμοποιείτε τον `/vmlinuz`). Μετά, ξανατρέξτε το lilo. Σε νεώτερες εκδόσεις απλά δίνετε `lilo`, αλλά σε παλιότερες πρέπει να κάνετε `/etc/lilo/install` (εγκατάσταση), ή ακόμη και `/etc/lilo/lilo -C /etc/lilo/config` (ρύθμιση).

Αν θέλετε να μάθετε περισσότερα γιά τις ρυθμίσεις του LILO, ή δεν έχετε το LILO, βρείτε την πιο πρόσφατη έκδοσή του από τον αγαπημένο σας ftp server, και ακολουθείστε τις οδηγίες της.

Γιά να bootάρετε από τον σκληρό δίσκο μ' έναν από τους παλιούς πυρήνες σας (πράγμα που είναι ένας τρόπος να διασωθείτε, αν τα κάνετε θάλασσα με τον νέο πυρήνα), αντιγράψτε στο τέλος του αρχείου τις γραμμές κάτω από (και μαζί με) το `image = xxx` που βρίσκονται στο αρχείο ρυθμίσεων του LILO, και αλλάξτε το `image = xxx` σε `image = yyy`, όπου `yyy` είναι το πλήρες όνομα path του αρχείου, στο οποίο κάνατε backup τον παλιό σας πυρήνα. Μετά, αλλάξτε την `label = zzz` σε `label = linux-backup` και ξανατρέξτε το `lilo`. Μπορεί να χρειαστεί να βάλετε μιά γραμμή στο αρχείο ρυθμίσεων, που να λέει `delay=x`, όπου x είναι μιά τιμή σε δέκατα του δευτερολέπτου, που λέει στο LILO πόσο να περιμένει πριν το bootάρισμα, ώστε να μπορείτε να το διακόψετε (πχ πατώντας το πλήκτρο shift), και να δώσετε την label (ετικέτα) του backup αρχείου (σε περίπτωση που συμβούν δυσάρεστα πράγματα).

5 Μπαλώνοντας (patching) τον πυρήνα

5.1 Εφαρμόζοντας ένα μπάλωμα (patch)

Οι κατά τμήματα βελτιώσεις του πυρήνα διανέμονται ως patches (= μπαλώματα). Πχ, αν έχετε την έκδοση 1.1.45, και πάρει το μάτι σας ότι κάπου υπάρχει `patch46.gz` γι' αυτήν, σημαίνει ότι μπορείτε να την αναβαθμίσετε σε έκδοση 1.1.46, ενώνοντας το patch. Φυσικά, μπορεί πρώτα να χρειαστεί να κάνετε backup του δέντρου των dirs του source κώδικα (το `make clean`, και μετά το `cd /usr/src; tar zcvf old-tree.tar.gz linux`, θα σας φτάξουν ένα συμπιεσμένο tar αρχείο).

Έτσι, συνεχίζοντας το παραπάνω παράδειγμα, ας υποθέσουμε ότι έχετε το `patch46.gz` στο `/usr/src`. Μετά γυρίστε με `cd` στο dir `/usr/src` και δώστε `zcat patch46.gz | patch -p0` (ή `patch -p0 < patch46`, αν το patch είναι ασυμπίεστο). Θα δείτε διάφορα πράγματα να φεύγουν με ταχύτητα (ή να αργοπετάνε, αν ο Η/Υ σας είναι τόσο αργός), πράγμα που σας λέει ότι προσπαθεί να τσοντάρει τα κομμάτια, και αν πέτυχε ή όχι. Συνήθως αυτή η δραστηριότητα εξελίσσεται τόσο γρήγορα, ώστε να μη μπορείτε να διαβάσετε τα μηνύματά της, και δεν είμαστε πάντα σίγουροι αν πετυχαίνει ή όχι. Επομένως, μπορεί να θέλετε να βάλετε την παράμετρο `-s` στη διαδικασία προσκόλλησης του `patch`, η οποία βγάζει στην οθόνη μόνο τα μηνύματα λάθους. (Δεν θ' αποκτήσετε πολύ συναίσθημα του τύπου "γιά δες, ο Η/Υ μου κάνει κάτι γιά ν' αλλάξει!", αλλά μπορεί και να σας αρέσει αυτός ο τρόπος.) Γιά να βρείτε ποιά μέρη της διαδικασίας δεν δούλεψαν σωστά, πάτε στο dir `/usr/src/linux` και ψάξτε γιά αρχεία με κατάληξη `.rej` . Μερικές εκδόσεις της `patch` (παλιότερες εκδόσεις, που γίνανε compile με τη βοήθεια κατώτερου filesystem), αφήνουν τα αρχεία αυτά με κατάληξη `#` . Μπορείτε να βάλετε την εντολή `find` να ψάξει γιά χάρη σας. Δίνοντας

```
find .  -name '*.rej' -print
```

βγάζει στην οθόνη όσα αρχεία με κατάληξη `.rej` που υπάρχουν στο τρέχον dir και στα subdirs του.

Αν όλα πήγαν καλά, κάντε `make clean`, `config`, και `dep`, όπως τα περιγράψαμε στις ενότητες 3 και 4.

Υπάρχουν κάμποσες παράμετροι της εντολής `patch`. Όπως προαναφέραμε, η `patch -s` καταπνίγει όλα τα μηνύματα, εκτός από τα μηνύματα λαθών. Αν βάλετε τον source κώδικα του πυρήνα σε κάποιο dir διαφορετικό από το `/usr/src/linux`, η `patch -p1` (ς' αυτό το άλλο dir) θα κάνει το patch μιά χαρά. Οι υπόλοιπες παράμετροι της `patch` είναι τεκμηριωμένες καλά στην αντίστοιχη σελίδα man.

5.2 Αν κάτι δεν πάει καλά

(Σημείωση : Αυτή η ενότητα αναφέρεται κυρίως σε παλιότερους πυρήνες.)

Το πιο συχνό πρόβλημα που υπήρχε, ήταν όποτε ένα patch άλλαζε ένα αρχείο με όνομα `config.in`, πράγμα που δεν ήταν απόλυτα σωστό, επειδή ο κάθε χρήστης αλλάζει τις ρυθμίσεις στον δικό του Η/Υ. Αυτό έχει πλέον εξαλειφθεί, αλλά μπορεί να το ξαναβρείτε μπροστά σας σε παλιότερες εκδόσεις. Γιά να το διορθώσετε, ψάξτε στο αρχείο `config.in.rej`, να δείτε τί άφησε το patch. Οι αλλαγές τυπικά σημειώνονται με `+` και `-` στην αρχή κάθε γραμμής. Διαβάστε τις γραμμές που περιστοιχίζουν τις γραμμές των αλλαγών, και θυμηθείτε αν δώσατε ανάλογα `y` ή `n`. Τώρα, διορθώστε το `config.in`, και αλλάξτε τα `y` σε `n` και τα `n` σε `y`, όπως χρειάζεται. Δώστε

```
patch -p0 < config.in.rej
```

και αν λέει ότι πέτυχε (δηλ. δεν έχει μηνύματα αποτυχίας), τότε μπορείτε να συνεχίσετε με ρυθμίσεις και compilation. Το αρχείο `config.in.rej` θα παραμείνει ως έχει, αλλά μπορείτε να το σβήσετε.

Αν συναντήσετε κι άλλα προβλήματα, μπορεί να έχετε εγκαταστήσει ένα patch που δεν δουλεύει. Εάν η εντολή patch βγάλει το μήνυμα `previously applied patch detected:  Assume -R?` (ανιχνεύτηκε προηγούμενο patch, να το αντικαταστήσω;) πιθανότατα πάτε να βάλετε patch παλιότερης έκδοσης από τον πυρήνα σας. Αν απαντήσετε `y`, θα προσπαθήσει να αγνοήσει τον source κώδικα που έχετε, και πιθανότατα θ' αποτύχει. Συνεπώς, θα χρειαστείτε ένα ολόκληρο νέο δέντρο source κώδικα (πράγμα που, τελικά, μπορεί και να μην είναι κακή ιδέα).

Γιά να αφαιρέσετε ένα patch, χρησιμοποιείστε την εντολή `patch -R` επάνω στο αρχικό patch.

Το καλύτερο που έχετε να κάνετε, όταν τα patches αποδεικνύονται λανθασμένα, είναι να ξαναρχίσετε μ' ένα καθαρό του κουτιού δέντρο source κώδικα (πχ από ένα από τα αρχεία `linux-x.y.z.tar.gz`), και να ξανακάνετε όλη τη διαδικασία.

5.3 Σβήνοντας τα .orig αρχεία

Μετά από μόλις λίγα patches, θ' αρχίσουν να συσσωρεύονται αρχεία `.orig` . Πχ, ένα δέντρο έκδοσης 1.1.51 που είχα, τελευταία φορά καθαρίστηκε στην έκδοση 1.1.48 . Με το που έσβησα τα .orig αρχεία, εξοικονόμησα περισσότερο από μισό MB. Η εντολή

```
find .  -name '*.orig' -exec rm -f {} ';'
```

θα φροντίσει τα πράγματα γιά σας. Οι εκδόσεις της `patch` που βάζουν # αντί γιά .rej, χρησιμοποιούν tilde (περισπωμένη) αντί γιά `.orig` .

Υπάρχουν καλύτεροι τρόποι γιά να σβήσουμε τα αρχεία `.orig`, οι οποίοι εξαρτώνται από τα GNU `xargs`. Πχ :

```
find .  -name '*.orig' | xargs rm
```

ή η "αρκετά ασφαλής, αλλά λίγο πιο κυριολεκτική" μέθοδος :

```
find . -name '*.orig' -print0 | xargs --null rm --
```

5.4 Άλλα patches

Υπάρχουν κι άλλα patches (αυτά θα τα αποκαλώ "μή στάνταρ"), εκτός απ' όσα διανέμει ο Linus. Αν δοκιμάσετε να τα προσθέσετε, τα patches του Linus μπορεί και να μη δουλέψουν σωστά, και είτε θα πρέπει να τα ξαναβγάλετε - να διορθώσετε τον source κώδικα του patch - να εγκαταστήσετε ένα νέο δέντρο source κώδικα, είτε να κάνετε έναν συνδυασμό όλων των παραπάνω. Αυτό μπορεί να είναι πολύ ενοχλητικό, επομένως, αν δεν θέλετε ν' αλλάξετε τον source κώδικα (με πιθανότητα γιά πολύ αποτυχημένο αποτέλεσμα), αφαιρέστε τα μή στάνταρ patches πριν βάλετε τα του Linus, ή απλά εγκαταστήστε ένα νέο δέντρο. Τότε, μπορείτε να δείτε εάν τα μή στάνταρ patches δουλεύουν ακόμη. Αν όχι, είτε έχετε κολλήσει ς' έναν παλιό πυρήνα, και παίζετε με το patch ή τον source κώδικα προσπαθώντας να τα κάνετε να δουλέψουν, ή περνάτε την ώρα σας περιμένοντας (πιθανότατα παρακαλώντας) να βγει μιά νέα έκδοση του patch.

Πόσο συχνά υπάρχουν patches εκτός της στάνταρ διανομής; Πιθανότατα κάποια στιγμή να συναντήσετε κάποιο. Προσωπικά χρησιμοποιούσα το noblink patch στις virtual κονσόλες μου, επειδή μισώ τους κέρσορες που αναβοσβήνουν. (Το συγκεκριμένο patch ανανεώνεται συχνά - τουλάχιστον ανανεωνόταν κατά το παρελθόν - στις νέες εκδόσεις του πυρήνα.) Ωστόσο, με τους περισσότερους νέους drivers συσκευών να έχουν πλέον τη μορφή modules που φορτώνονται, η συχνότητα εμφάνισης των "μή στάνταρ" patches μειώνεται σημαντικά.

6 Επιπλέον πακέτα

Ο πυρήνας του Linux έχει πολλές ικανότητες, οι οποίες δεν επεξηγούνται ούτε ακόμη και μέσα στον ίδιο τον source κώδικα. Αυτές οι ικανότητες συνήθως χρησιμοποιούνται από εξωτερικά προγράμματα, των οποίων μερικά από τα πιο συνηθισμένα παρουσιάζονται εδώ.

6.1 kbd

Πιθανότατα η κονσόλα του Linux να έχει περισσότερες ικανότητες απ' όσες δικαιούται. Μεταξύ αυτών βρίσκεται η ικανότητα να αλλάζουμε fonts, να αναδιατάσσουμε το πληκτρολόγιό μας, να αλλάζουμε ανάλυση οθόνης (αυτό στους νεώτερους πυρήνες), κλπ. Το πακέτο kbd έχει προγράμματα που επιτρέπουν στον χρήστη να κάνει όλα τα παραπάνω, συν πολλά fonts και διατάξεις πλήκτρων γιά σχεδόν κάθε πληκτρολόγιο, και είναι διαθέσιμο από τα ίδια sites, όπου βρίσκουμε τον source κώδικα του πυρήνα.

6.2 util-linux

Ο Rik Faith (`faith@cs.unc.edu`) μάζεψε ς' ένα πακέτο μιά μεγάλη συλλογή utilities γιά Linux, το οποίο - εντελώς τυχαία - ονομάζεται util-linux. Αυτές οι utilities τώρα συντηρούνται από τον Andries Brouwer (`util-linux@math.uio.no`). Είναι διαθέσιμες με ανώνυμο ftp από το sunsite.unc.edu στο dir `/pub/Linux/system/misc`, και περιέχουν προγράμματα όπως το `setterm`, το `rdev`, και το `ctrlaltdel`, τα οποία είναι σχετικά με τον πυρήνα. Όπως λέει ο Rik, *μην κάνετε εγκαταστάσεις χωρίς να σκέφτεστε.* Δεν χρειάζεται να εγκαταστήσετε το κάθε τί μέσα στο πακέτο. (Αν κάνετε κάτι τέτοιο, πολύ πιθανό να σας προξενήσει σοβαρά προβλήματα.)

6.3 hdparm

Όπως συμβαίνει με πολλά πακέτα, τούτο εδώ κάποτε ήταν ένα πακέτο με προγράμματα και υποστήριξης του πυρήνα. Τα patches ενσωματώθηκαν στον επίσημο πυρήνα, και τα προγράμματα γιά να τον βελτιστοποιήσετε (και να παίξετε με τον σκληρό σας δίσκο) διανέμονται χωριστά.

6.4 gpm

Το gpm σημαίνει general purpose mouse (ποντίκι γιά κάθε δουλειά). Αυτό το πρόγραμμα σας επιτρέπει να κάνετε cut και paste καθαρό text στις virtual κονσόλες, καθώς και να κάνετε κι άλλα πράγματα, με μιά μεγάλη ποικιλία τύπων ποντικιού.

7 Μερικές παγίδες

7.1 make clean

Αν ο πυρήνας σας κάνει τρελά πράγματα μετά από μιά συνηθισμένη αναβάθμιση, πιθανότατα έχετε ξεχάσει το `make clean` πριν κάνετε compile τον νέο πυρήνα. Τα συμπτώματα μπορεί να είναι οποιαδήποτε, από κρέμασμα του Η/Υ και παράξενη συμπεριφορά του I/O, μέχρι πεσμένη απόδοση. Επίσης, θυμηθείτε να κάνετε και `make dep`.

7.2 Τεράστιοι ή αργοί πυρήνες

Αν ο πυρήνας σας απορροφά πολλή μνήμη, είναι πολύ μεγάλος, και/ή κάνει αιώνες γιά compile ακόμη και στο νέο σας Quadbazillium-III/4400, πιθανότατα ρυθμίζετε τόνους από άχρηστα πράγματα (drivers συσκευών, filesystems, κλπ). Αν δεν χρησιμοποιείτε κάτι, μην το ρυθμίζετε, επειδή καταλαμβάνει πόρους του Η/Υ. Το προφανέστερο σύμπτωμα παραφουσκώματος του πυρήνα είναι η υπερβολική ανταλλαγή δεδομένων (swapping) από - προς μνήμη και δίσκο. Αν ο δίσκος σας κάνει πολύ θόρυβο, και δεν είναι από κείνους τους παλιούς Fujitsu Eagles, που κατά το κλείσιμο κάνουν σαν αεριωθούμενο που προσγειώνεται, ψάξτε το στήσιμο του πυρήνα σας.

Μπορείτε να βρείτε πόση μνήμη χρησιμοποιεί ο πυρήνας, παίρνοντας το ολικό ποσό μνήμης του Η/Υ σας, και αφαιρώντας το ποσό της "total mem" (ολικής μνήμης) στο αρχείο `/proc/meminfo`, ή το ποσό που δίνει η εντολή `free`.

7.3 Δεν δουλεύει η παράλληλη θύρα, ή ο εκτυπωτής μου

Οι επιλογές ρύθμισης γιά τα PCs είναι : Πρώτα, στην κατηγορία "General Setup" (γενικές ρυθμίσεις), διαλέξτε "Parallel port support" (υποστήριξη παράλληλης θύρας) και "PC-style hardware" (hardware τύπου PC). Μετά, στις "Character devices" (συσκευές χαρακτήρων) διαλέξτε "Parallel printer support" (υποστήριξη παράλληλου εκτυπωτή).

Μετά, υπάρχουν και οι ονομασίες. Το Linux 2.2 ονομάζει τον εκτυπωτή διαφορετικά απ' ό,τι οι προηγούμενες εκδόσεις. Το συμπέρασμα είναι ότι αν είχατε `lp1` με τον παλιό πυρήνα σας, τώρα πιθανότατα έχετε `lp0` με τους καινούργιους. Γιά να βρείτε το όνομα του εκτυπωτή, δώστε την εντολή `dmesg`, ή ψάξτε τα αρχεία καταγραφής (logs) στο dir `/var/log`.

7.4 Δεν γίνεται compile στον πυρήνα

Αν δεν γίνεται compile, τότε πιθανότατα απέτυχε η προσθήκη ενός patch, ή κατά κάποιο τρόπο δεν είναι σωστός ο source κώδικας. Επίσης, μπορεί να είναι παλιά ή όχι σωστή η έκδοση της gcc που έχετε (πχ τα αρχεία include). Βεβαιωθείτε ότι οι συμβολικοί δεσμοί (symbolic links) που περιγράφει ο Linus στο `README`, είναι στημένοι σωστά. Γενικά, αν δεν μπορούμε να κάνουμε compile έναν στάνταρ πυρήνα, τότε κάτι δεν πάει καθόλου καλά με το Linux, και είναι αναγκαία η επανεγκατάσταση μερικών εργαλείων.

Σε μερικές περιπτώσεις, η gcc κρασάρει εξ αιτίας προβλημάτων στο hardware. Το μήνυμα λάθους είναι κάτι σαν "xxx exited with signal 15" (= το xxx έπαψε να λειτουργεί με μήνυμα 15), και γενικά αυτά τα μηνύματα έχουνε μυστηριώδη εμφάνιση. Πιθανότατα δεν θα μίλαγε γιά κάτι τέτοιο, αλλά είναι κάτι που μου συνέβη κάποτε - είχα χαλασμένη μνήμη cache, και ο compiler κρεμούσε σε τυχαίες στιγμές. Αν έχετε προβλήματα, πρώτα προσπαθήστε να επανεγκαταστήσετε την gcc. Πρέπει να γίνετε καχύποπτοι μονάχα αν κάνετε απροβλημάτιστα compile στον πυρήνα σας με off την εξωτερική cache, με περιορισμένη ποσότητα RAM, κλπ.

Ενοχλεί τους χρήστες, όταν τους λες ότι το hardware τους είναι προβληματικό, αλλά δεν βγάζω τα πράγματα αυτά απ' το μυαλό μου. Υπάρχει ένας σχετικός κατάλογος με συχνές ερωτήσεις (FAQs), που βρίσκεται στο `http://www.bitwizard.nl/sig11/`.

7.5 Η νέα έκδοση του πυρήνα δεν bootάρει

Δεν τρέξατε το LILO, ή δεν το ρυθμίσατε σωστά. Ένα πράγμα που με "διέλυσε" κάποτε, ήταν ένα πρόβλημα στο αρχείο ρυθμίσεων, όπου έγραφε `boot = /dev/hda1`, αντί `boot = /dev/hda`. (Αυτό μπορεί να είναι πολύ ενοχλητικό στην αρχή, αλλά αν έχετε ένα αρχείο ρυθμίσεων που δουλεύει, δεν χρειάζεται να το αλλάξετε.)

7.6 Ξεχάσατε να τρέξετε το LILO, ή ο Η/Υ δεν bootάρει καθόλου

Ούπς! Το καλύτερο πράγμα που μπορείτε να κάνετε εδώ είναι boot από δισκέτα ή CDROM, και να ετοιμάσετε μιά ακόμη δισκέτα εκκίνησης (όπως πχ θα την έφτιαχνε η εντολή `make zdisk`). Πρέπει να γνωρίζετε που να βρείτε το root filesystem σας (/), και τί τύπος είναι (δηλ. second extended, minix). Στο παρακάτω παράδειγμα, πρέπει επίσης να γνωρίζετε επάνω σε ποιό filesystem βρίσκεται το δέντρο του source κώδικά σας (`/usr/src/linux`), τον τύπο του, και πού το κάνουμε mount συνήθως.

Στο ακόλουθο παράδειγμα, το / είναι το `/dev/hda1`, και το filesystem που κρατάει το `/usr/src/linux` βρίσκεται στο `/dev/hda3`, που συνήθως το κάνουμε mount στο `/usr`. Και τα δύο είναι filesystems τύπου second extended. Το image του ενεργού πυρήνα στο `/usr/src/linux/arch/i386/boot` ονομάζεται `bzImage`.

Η ιδέα είναι ότι, αν υπάρχει ένα ενεργό `bzImage`, πιθανόν να μπορούμε να το βάλουμε στη νέα δισκέτα. Μιά άλλη εναλλακτική λύση, που ίσως να δουλεύει καλύτερα, ίσως και όχι (εξαρτάται με ποιά μέθοδο κάνατε μπάχαλο τον Η/Υ σας), θα τη συζητήσουμε μετά το παράδειγμα.

Πρώτα, κάντε boot από συνδυασμό boot/root δισκετών (ή δισκέτα διάσωσης - rescue disk), και κάντε mount το filesystem που περιέχει το image του ενεργού πυρήνα :

```
mkdir /mnt
mount -t ext2 /dev/hda3 /mnt
```

Αν η `mkdir` σας πει ότι το dir αυτό υπάρχει ήδη, απλά αγνοήστε την. Τώρα, πάτε με `cd` στο dir όπου βρισκότανε το image του ενεργού πυρήνα. Σημειώστε πως

```
/mnt + /usr/src/linux/arch/i386/boot - /usr = /mnt/src/linux/arch/i386/boot
```

Βάλτε μιά φορμαρισμένη δισκέτα στο drive "A:" (όχι τη δισκέτα boot ή τη root!), ρίξτε το image επάνω της, και ρυθμίστε το γιά το root filesystem σας :

```
cd /mnt/src/linux/arch/i386/boot
dd if=bzImage of=/dev/fd0
rdev /dev/fd0 /dev/hda1
```

Μετά, γυρίστε με `cd` στο / και κάντε unmount το κανονικό filesystem `/usr` :

```
cd /
umount /mnt
```

Τώρα, λογικά πρέπει να μπορείτε να επανεκκινήσετε τον Η/Υ σας κανονικά απ' αυτή τη δισκέτα. Μην ξεχάσετε να τρέξετε το lilo (ή να κάνετε σωστά ό,τι κάνατε πριν λάθος) μετά την επανεκκίνηση!

Όπως προαναφέραμε, υπάρχει και μιά άλλη συνηθισμένη εναλλακτική λύση. Αν συμβεί να έχετε ένα αντίγραφο του ενεργού πυρήνα στο / (πχ το `/vmlinuz`), μπορείτε να το χρησιμοποιήσετε γιά μιά δισκέτα εκκίνησης. Υποθέτοντας όλα τα παραπάνω ως σωστά, και ότι το image του πυρήνα ονομάζεται `/vmlinuz`, αλλάξτε τα εξής στο παραπάνω παράδειγμα : Αλλάξτε το `/dev/hda3` σε `/dev/hda1` (στο filesystem /), το `/mnt/src/linux` σε `/mnt`, και το `if=bzImage` σε `if=vmlinuz`. (Αγνοήστε τη σημείωση που λέει πώς να φτιάξετε το `/mnt/src/linux`.)

Η χρήση του LILO με μεγάλους δίσκους (πάνω από 1024 κυλίνδρων) μπορεί να προξενήσει προβλήματα. Διαβάστε το LILO mini-HOWTO, ή οποιοδήποτε σχετικό κείμενο.

7.7 Λέει "warning: bdflush not running"

(Δηλ. "προειδοποίηση : δεν τρέχει το bdflush".) Αυτό μπορεί να είναι σοβαρό πρόβλημα. Με αρχή κάποια έκδοση του πυρήνα μεταγενέστερη της 1.0 (γύρω στις 20 Απριλίου 1994), αναβαθμίσθηκε και αντικαταστάθηκε το πρόγραμμα `update` (το οποίο αδειάζει σε περιοδική βάση τα βοηθητικά αρχεία του filesystem). Πάρτε τον source κώδικα του `bdflush` (μπορείτε να τον βρείτε εκεί, όπου υπάρχει και ο source κώδικας του πυρήνα), και εγκαταστήστε τον. (Όσο κάνετε την εγκατάσταση, πιθανότατα να θέλετε να τρέχει ο παλιός πυρήνας σας.) Το πρόγραμμα αυτό εγκαθίσταται ως `update`, και μετά την επανεκκίνηση ο νέος πυρήνας δεν θα παραπονεθεί ξανά.

7.8 Δεν μπορώ να βάλω το IDE/ATAPI CD-ROM να δουλέψει

Αρκετά παραδόξως, αρκετοί χρήστες δεν μπορούν να βάλουνε τις μονάδες ATAPI να δουλέψουν, επειδή είναι πολλά τα πράγματα που πιθανόν να μην πάνε καλά.

Εάν το CD-ROM σας είναι η μόνη συσκευή επάνω ς' ένα καλώδιο IDE, πρέπει να το κάνετε (με τους jumpers - βραχυκυκλωτήρες - στο σασί του) "master" ή "single". Αυτό είναι το πλέον συνηθισμένο λάθος - αλλά εντελώς υποθετικά.

Γιά παράδειγμα, η εταιρία Creative Labs έχει βάλει διασυνδέσεις IDE επάνω στις κάρτες ήχου που παράγει. Ωστόσο, αυτό οδηγεί στο ενδιαφέρον πρόβλημα ότι, ενώ μερικοί χρήστες έχουνε μόνο μία διασύνδεση IDE, αρκετοί έχουνε δύο απ αυτές επάνω στη motherboard τους (συνήθως στην IRQ15), άρα η διασύνδεση στις κάρτες SoundBlaster γίνεται η τρίτη IDE (πολλές φορές στην IRQ11).

Αυτό προξενεί προβλήματα στο Linux, επειδή οι εκδόσεις 1.2.x δεν υποστηρίζουν τρίτη διασύνδεση IDE. (Η σχετική υποστήριξη ξεκινάει κάπου στις εκδόσεις 1.3.x , αλλά θυμηθείτε ότι αυτές είναι εκδόσεις ανάπτυξης, και δεν κάνουν αυτόματη ανίχνευση περιφερειακών.) (Σ.τ.μ. : Το Linux έχει βελτιωθεί πολύ από τότε που γράφτηκε το παρόν HowTo, αλλά στην πράξη εξακολουθεί να παραμένει πολύ σπάνια η ανάγκη γιά 3η IDE.) Γιά να ξεπεράσετε αυτόν τον σκόπελο, έχετε κάποιες επιλογές :

Αν ήδη έχετε δεύτερη IDE, πιθανότατα ή δεν την χρησιμοποιείτε, ή δεν έχει δύο μονάδες δίσκου - cdrom επάνω της. Πάρτε τη μονάδα ATAPI από την κάρτα ήχου, και βάλτε την επάνω στη δεύτερη IDE. Μετά, απενεργοποιείστε την IDE της κάρτας ήχου (και εξοικονομήστε έτσι μία IRQ).

Αν δεν έχετε δεύτερη IDE, καλωδιώστε με τους jumpers την IDE της κάρτας ήχου (όχι το μέρος που έχει να κάνει με τις λειτουργίες ήχου!) στην IRQ15, ώστε να γίνει δεύτερη IDE. Αυτό το σχήμα πρέπει να δουλεύει κανονικά.

7.9 Λέει παράξενα πράγματα γιά καταργημένες αιτήσεις routing

Βρείτε τις νέες εκδόσεις του προγράμματος `route`, καθώς και όποιου άλλου προγράμματος χειρίζεται το routing. Έχει αλλάξει το `/usr/include/linux/route.h` (το οποίο είναι ένα αρχείο στο `/usr/src/linux`).

7.10 Το firewalling δεν δουλεύει στην έκδοση 1.2.0

Αναβαθμιστείτε τουλάχιστον στην έκδοση 1.2.1 .

7.11 "Not a compressed kernel Image file"

(Δηλ. "Δεν πρόκειται γιά συμπιεσμένο image του πυρήνα".) Μην χρησιμοποιήσετε γιά bootάρισμα το αρχείο `vmlinux` που δημιουργήθηκε στο `/usr/src/linux`. Το σωστό είναι το `[..]/arch/i386/boot/bzImage`.

7.12 Προβλήματα με το τερματικό κονσόλας (console terminal) μετά την αναβάθμιση σε 1.3.x

Στη γραμμή γιά την console termcap του αρχείου `/etc/termcap`, αλλάξτε τη λέξη `dumb` σε `linux`. Ίσως χρειαστεί να δώσετε και μία σχετική γραμμή terminfo.

7.13 Δεν μπορώ να κάνω compile, αφού αναβάθμισα τον πυρήνα

Ο source κώδικας του πυρήνα του Linux περιλαμβάνει κάμποσα αρχεία include (αυτά που έχουν κατάληξη `.h`) στο dir `/usr/include`, στα οποία και αναφέρονται τα υπόλοιπα. Η τυπική κλήση των αρχείων αυτών γίνεται ως εξής (όπου `xyzzy.h` είναι οποιοδήποτε αρχείο include μέσα στο `/usr/include/linux`) :

```
#include <linux/xyzzy.h>
```

Κανονικά, υπάρχει στο `/usr/include` το link `linux` προς το directory `include/linux` του source κώδικα του πυρήνα σας (συνήθως `/usr/src/linux/include/linux`). Αν αυτό το link δεν βρίσκεται εκεί, ή δείχνει προς λάθος μέρος, τα περισσότερα αρχεία δεν μπορούν να γίνουν compile. Αν αποφασίσατε ότι ο source κώδικας του πυρήνα τρώει χώρο στον δίσκο, και τον σβήσατε, αυτό πιθανότατα θ' αποτελέσει πρόβλημα. Μιά ακόμη αιτία προβλημάτων, είναι οι άδειες χρήσης των αρχείων. Εάν το dir `root` έχει εξ ορισμού umask, που δεν επιτρέπει σε άλλους χρήστες να δούνε τα περιεχόμενά του, και ανοίξατε τον source κώδικα του πυρήνα χωρίς την παράμετρο `p` (preserve filemodes - διατήρηση του τρόπου λειτουργίας των αρχείων), αυτοί οι συγκεκριμένοι χρήστες δεν θα μπορέσουν να χρησιμοποιήσουν μετά τον compiler της γλώσσας C. Αν και μπορείτε να δώσετε την εντολή `chmod` γιά να διορθώσετε αυτή την κατάσταση, είναι μάλλον ευκολότερο να ξανανοίξετε τα αρχεία include. Αυτό γίνεται με τον ίδιο τρόπο που ανοίξατε όλον τον source κώδικα του πυρήνα στην αρχή, μόνο που έχει μιά παράμετρο παραπάνω :

```
blah# tar zxvpf linux.x.y.z.tar.gz linux/include
```

Σημείωση : Η `"make config"` θα ξαναφτιάξει το link προς το `/usr/src/linux`, εάν αυτό δεν υπάρχει.

7.14 Αυξάνοντας τα όρια

Οι ακόλουθες, *ως παράδειγμα* εντολές, μπορεί να φανούν χρήσιμες ς' όσους ρωτάνε πώς αυξάνονται κάποια όρια, που βάζει ο πυρήνας από μόνος του :

```
echo 4096 > /proc/sys/kernel/file-max
echo 12288 > /proc/sys/kernel/inode-max
echo 300 400 500 > /proc/sys/vm/freepages
```

8 Σημείωση γιά αναβαθμίσεις προς έκδοση 2.0.x, 2.2.x

Οι εκδόσεις 2.0.x και 2.2.x του πυρήνα έφεραν κάμποσες αλλαγές στον τρόπο εγκατάστασής του. Το αρχείο `Documentation/Changes` στο δέντρο dirs του source κώδικα των εκδόσεων 2.0.x περιέχει πληροφορίες που πρέπει να ξέρετε, αν κάνετε αναβάθμιση προς οποιαδήποτε από τις δύο αυτές εκδόσεις. Πιθανότατα θα χρειαστεί να αναβαθμίσετε αρκετά βασικά πακέτα, όπως τα gcc, libc, SysVInit, και ίσως ν' αλλάξετε μερικά αρχεία, επομένως πρέπει να αναμένετε τέτοιες αλλαγές. Ωστόσο, μην πανικοβάλλεστε.

9 Modules

Τα modules του πυρήνα που φορτώνονται (loadable), συνήθως κάνουν οικονομία στη μνήμη και διευκολύνουν τις ρυθμίσεις. Το πεδίο των modules έχει μεγαλώσει, ώστε σήμερα περιλαμβάνει filesystems, drivers καρτών δικτύου, drivers μονάδων μαγνητικής ταινίας, drivers εκτυπωτών, και άλλα.

9.1 Εγκαθιστώντας τις utilities γιά τα modules

Οι utilities γιά τα modules είναι διαθέσιμες (ως αρχείο `modutils-x.y.z.tar.gz`) από εκεί, όπου πήρατε τον source κώδικα του πυρήνα. Διαλέξτε την πλέον πρόσφατη έκδοση `x.y.z` του σχετικού patch, η οποία είναι ίση με (ή αμέσως παρακάτω από) την έκδοση του πυρήνα σας. Ανοίξτε το με `tar zxvf modutils-x.y.z.tar.gz`, πάτε με `cd` στο directory που δημιουργήθηκε (`modutils-x.y.z`), διαβάστε το `README`, και ακολουθήστε τις οδηγίες εγκατάστασης (που συνήθως είναι απλές, πχ κάτι σαν `make install`). Τώρα, κανονικά πρέπει να έχετε τα προγράμματα `insmod`, `rmmod`, `ksyms`, `lsmod`, `genksyms`, `modprobe`, και `depmod` στο dir `/sbin`. Αν θέλετε, κάντε μιά δοκιμή στις utilities αυτές με τον driver "hw", που περιλαμβάνεται στην `insmod` ως παράδειγμα. Γιά λεπτομέρειες, ψάξτε το αρχείο `INSTALL` στο σχετικό subdir.

Η `insmod` βάζει ένα module στον ενεργό πυρήνα. Τα modules συνήθως έχουνε επέκταση `.o` . Ο driver δοκιμών που προαναφέραμε, ονομάζεται `drv_hello.o`. Επομένως, γιά να τον βάλουμε στον πυρήνα, θα δώσουμε `insmod drv_hello.o`. Γιά να δούμε ποιά modules χρησιμοποιεί αυτή τη στιγμή ο πυρήνας μας, δίνουμε `lsmod`. Η έξοδος είναι κάπως έτσι :

```
blah# lsmod
Module:        #pages:  Used by:
drv_hello         1
```

Το `drv_hello` είναι το όνομα του module, καταλαμβάνει μιά σελίδα των 4k μνήμης, και προς το παρόν δεν υπάρχουν άλλα modules που εξαρτώνται απ' αυτό. Γιά να το αφαιρέσουμε, δίνουμε `rmmod drv_hello`. Σημειώστε ότι η `rmmod` χρειάζεται *όνομα module* ως όρισμα, όχι όνομα αρχείου. Το όνομα του module το παίρνετε από τη λίστα του `lsmod`. Ο σκοπός των υπολοίπων utilities γιά τα modules αναφέρεται στις αντίστοιχες σελίδες man.

9.2 Modules που διανέμονται μαζί με τον πυρήνα

Από την έκδοση 2.0.30, τα περισσότερα πακέτα είναι διαθέσιμα ως loadable modules. Γιά να τα χρησιμοποιήσετε, πρώτα βεβαιωθείτε ότι δεν τα έχετε στήσει μαζί με τον πυρήνα. Δηλαδή, μην απαντάτε `y` γι' αυτά, κατά τη διάρκεια της `make config`. Κάντε compile έναν νέο πυρήνα, και επανεκκινήστε μ' αυτόν. Μετά, ξαναπάτε με `cd` στο `/usr/src/linux`, και δώστε `make modules`. Αυτό κάνει compile όλα τα modules που δεν βάλατε στο στήσιμο του πυρήνα, και τοποθετεί links προς αυτά στο `/usr/src/linux/modules`. Μπορείτε να τα χρησιμοποιήσετε κατευθείαν από το dir αυτό, ή να δώσετε `make modules_install`, που θα τα εγκαταστήσει στο dir `/lib/modules/x.y.z` (όπου `x.y.z` είναι η έκδοση του πυρήνα).

Αυτό είναι πολύ εύκολο ειδικά με τα filesystems. Πχ, πιθανότατα δεν θα χρησιμοποιήσετε συχνά τα filesystems minix και msdos, και, αν προσωπικά εύρισκα μιά δισκέτα msdos (μπρρρρ!!!), θα έδινα `insmod /usr/src/linux/modules/msdos.o`, και μετά `rmmod msdos`. Αυτή η διαδικασία εξοικονομεί περίπου 50k RAM στον πυρήνα κατά την υπόλοιπη λειτουργία του. Εδώ επιβάλλεται μιά μικρή σημείωση γιά το filesystem minix : *Πάντα* να το στήνετε κατευθείαν μέσα στον πυρήνα, όταν πρόκειται γιά χρήση σε δισκέτες "διάσωσης".

10 Συμβουλές και κόλπα

10.1 Επανακατεύθυνση της εξόδου των εντολών make και patch

Αν θέλετε καταγραφή του τί κάνανε οι εντολές `make` και `patch`, μπορείτε να επανακατευθύνετε την έξοδο προς ένα αρχείο. Πρώτα, δείτε τί κέλυφος (shell) τρέχει : Δώστε `grep root /etc/passwd`, και ψάξτε γιά κάτι σαν `/bin/csh`.

Αν χρησιμοποιείτε το sh ή το bash, δίνοντας :

```
(command) 2>&1 | tee (output file)
```

θα πάρετε ένα αντίγραφο της εξόδου της `(command)` (= εντολής) στο αρχείο `(output file)` (= αρχείο εξόδου).

Γιά τα csh και tcsh, δώστε :

```
(command) |& tee (output file)
```

Γιά το rc (σημείωση : το οποίο πιθανότατα δεν χρησιμοποιείτε) δώστε :

```
(command) >[2=1] | tee (output file)
```

10.2 Εγκατάσταση του πυρήνα υπό προϋποθέσεις

Εκτός από τη χρήση δισκετών, υπάρχουν αρκετές μέθοδοι να τεστάρουμε έναν νέο πυρήνα χωρίς να πειράζουμε τον παλιό. Αντίθετα με ό,τι συμβαίνει σε άλλες παραλλαγές του Unix, το LILO έχει την ικανότητα να bootάρει τον πυρήνα απ' οπουδήποτε στον σκληρό. (Αν έχετε δίσκο μεγαλύτερο των 500 MBytes, παρακαλώ διαβάστε στην τεκμηρίωση του LILO τις πιθανές αιτίες προβλημάτων.) Επομένως, αν προσθέσετε κάτι που μοιάζει με

```
image = /usr/src/linux/arch/i386/boot/bzImage
    label = new_kernel
```

στο τέλος του αρχείου ρυθμίσεων του LILO, θα μπορέσετε να δοκιμάσετε έναν νέο πυρήνα, χωρίς ν' ακουμπήσετε τον παλιό σας `/vmlinuz` (φυσικά, αφού τρέξετε το `lilo`). Ο ευκολότερος τρόπος να πούμε στο LILO να bootάρει έναν νέο πυρήνα, είναι να πατήσουμε το πλήκτρο shift κατά την εκκίνηση (όταν γράφει

LILO στην οθόνη, και τίποτε άλλο), πράγμα που σας βγάζει σε μιά αναμονή (prompt). Εδώ μπορείτε να πληκτρολογήσετε new_kernel (δηλ. το όνομα του νέου πυρήνα), ώστε να bootάρει ο νέος πυρήνας.

Αν θέλετε να έχετε στον Η/Υ σας ταυτόχρονα περισσότερα από ένα δέντρα source κώδικα πυρήνα (όμως, προσέξτε : αυτό μπορεί να φάει *πολύ* χώρο δίσκου), ο πλέον συνηθισμένος τρόπος είναι να τα ονομάσετε /usr/src/linux-x.y.z, όπου x.y.z είναι η έκδοση του πυρήνα. Μετά, μπείτε να "διαλέξετε" ένα δέντρο source κώδικα μ' έναν συμβολικό δεσμό (symbolic link). Πχ, δίνοντας ln -sf linux-1.2.2 /usr/src/linux, ενεργοποιούμε το δέντρο 1.2.2 . Πριν φτιάξουμε ένα symbolic link, βεβαιωνόμαστε ότι η τελευταία παράμετρος της ln δεν είναι ένα πραγματικό directory (οι παλιοί symbolic links είναι μιά χαρά), διότι τα αποτελέσματα δεν θα είναι αυτά που περιμένουμε.

10.3 Ανανεώσεις του πυρήνα

Ο Russell Nelson (nelson@crynwr.com) κάνει περιλήψεις των αλλαγών στις νέες εκδόσεις του πυρήνα. Αυτές είναι σύντομες, και μάλλον πρέπει να τις διαβάσετε πριν από μιά αναβάθμιση. Είναι διαθέσιμες με ανώνυμο ftp από το ftp.emlist.com (στο dir pub/kchanges), ή από το site

```
http://www.crynwr.com/kchanges
```

11 Άλλα σχετικά HOWTOs που μπορεί να φανούν χρήσιμα

- Sound-HOWTO : κάρτες ήχου και σχετικές utilities
- SCSI-HOWTO : όλα τα σχετικά με SCSI controllers και συσκευές
- NET-2-HOWTO : δικτύωση
- PPP-HOWTO : ειδικά η δικτύωση με PPP
- PCMCIA-HOWTO : οι drivers γιά τον φορητό σας
- ELF-HOWTO : Τί είναι το ELF, σχετικές μετατροπές
- Hardware-HOWTO : γενική επισκόπηση του υποστηριζόμενου hardware
- Module mini-HOWTO : περισσότερα γιά τα modules του πυρήνα
- Kerneld mini-HOWTO : σχετικά με τον daemon kerneld
- BogoMips mini-HOWTO : αν αναρωτιέστε... (γιά την απόδοση του Η/Υ)

12 Διάφορα

12.1 Συγγραφέας

Ο συγγραφέας και συντηρητής του Linux Kernel-HOWTO είναι ο Brian Ward (bri@cs.uchicago.edu). Παρακαλώ, στείλτε μου σχόλια, προσαρτήματα, διορθώσεις (ειδικά αυτές μου είναι οι πλέον χρήσιμες).

Μπορείτε να ρίξετε μιά ματιά στη σελίδα μου, σε κάποιο από τα εξής URLs :

http://www.math.psu.edu/bri/
http://blah.math.tu-graz.ac.at/~bri/

Αν και προσπαθώ να δίνω προσοχή στα mails μου, παρακαλώ να θυμάστε ότι παίρνω *πολλά* απ' αυτά κάθε μέρα, άρα μπορεί ν' αργήσω λίγο να σας απαντήσω. Ειδικά όταν μου στέλνετε ερωτήσεις, παρακαλώ να εκφράζεστε με όση σαφήνεια και λεπτομέρεια μπορείτε. Εάν γράφετε γιά harware που δεν δουλεύει (ή γιά παρόμοια θέματα), πρέπει να ξέρω τί hardware έχετε. Αν έχετε να μου αναφέρετε κάποιο σφάλμα λειτουργίας, μη λέτε απλά "προσπάθησα αυτό κι αυτό, αλλά έδωσε λάθος". Πρέπει να ξέρω τί λάθος ήτανε αυτό. Επίσης, πρέπει να γνωρίζω ποιες εκδόσεις πυρήνα, gcc, και libc έχετε. Αν απλά μου γράψετε ότι χρησιμοποιείτε την τάδε διανομή, αυτό δεν μου λέει πολλά. Τέλος, οι απλές ερωτήσεις δεν με πειράζουν, διότι αν δεν ρωτήσετε, ποτέ δεν θα πάρετε απάντηση! Θα ήθελα να ευχαριστήσω τον καθένα που μου έστειλε e-mail.

Αν η ερώτησή σας δεν σχετίζεται με τον πυρήνα, ή είναι γραμμένη σε γλώσσα που δεν καταλαβαίνω, δεν θα απαντήσω.

Αν μου γράψατε και δεν πήρατε απάντηση σε λογικό χρονικό διάστημα (δηλ. περίπου τρείς εβδομάδες), τότε πιθανότατα έσβησα κατά λάθος το e-mail σας. Συγνώμη. Παρακαλώ, ξαναπροσπαθήστε.

Παίρνω πολλά e-mails σχετικά με πράγματα, που ουσιαστικά είναι προβλήματα ή θέματα του hardware. Αυτό είναι ok, αλλά παρακαλώ να θυμάστε ότι δεν γνωρίζω όλο το hardware του κόσμου. Προσωπικά χρησιμοποιώ AMD προσέσσορες, Adaptec και Sybios SCSI controllers, και δίσκους IBM SCSI.

Η έκδοση -0.1 αυτού εδώ του HowTo γράφτηκε στις 3 Οκτωβρίου 1994. Αυτό το κείμενο είναι διαθέσιμο σε μορφή SGML, PostScript, TeX, roff, και σκέτο text.

12.2 Προς ενέργεια

Η ενότητα "Συμβουλές και κόλπα" είναι λίγο μικρή. Ελπίζω να την επεκτείνω μέσω συμβουλών άλλων.

Το ίδιο και τα "Επιπλέον πακέτα".

Χρειάζονται κι άλλα επάνω στα θέματα debugging και crash recovery (= επαναλειτουργία του Η/Υ μετά από κρέμασμα).

12.3 Συνεισφορές

Περιέλαβα ένα μικρό μέρος του README του Linus (το χάκεμα του πυρήνα). Ευχαριστώ, Linus!

uc@brian.lunetix.de (Ulrich Callmeier) : Γιά την patch -s και την xargs.

quinlan@yggdrasil.com (Daniel Quinlan) : Διορθώσεις και πρόσθετα κομμάτια σε αρκετές ενότητες.

nat@nat@nataa.fr.eu.org (Nat Makarevitch) : Γιά τις mrproper, tar -p, και πολλά άλλα.

boldt@math.ucsb.edu (Axel Boldt) : Συγκέντρωσε περιγραφές από το Internet γιά τις ρυθμίσεις του πυρήνα, και μου έδωσε τη λίστα.

lembark@wrkhors.psyber.com (Steve Lembark) : Υποδείξεις γιά το πολλαπλό bootάρισμα.

kbriggs@earwax.pd.uwa.edu.au (Keith Briggs) : Διορθώσεις και υποδείξεις.

rmcguire@freenet.columbus.oh.us (Ryan McGuire) : πρόσθετα κομμάτια γιά τα makeables.

dumas@excalibur.ibp.fr (Eric Dumas) : Μετάφραση στα Γαλλικά.

simazaki@ab11.yamanashi.ac.jp (Yasutada Shimazaki) : Μετάφραση στα Ιαπωνικά.

jjamor@lml.ls.fi.upm.es (Juan Jose Amor Iglesias) : Μετάφραση στα Ισπανικά.

mva@sbbs.se (Martin Wahlen) : Μετάφραση στα Σουηδικά.

jzp1218@stud.u-szeged.hu (Zoltan Vamosi) : Μετάφραση στα Ουγγρικά.

bart@mat.uni.torun.pl (Bartosz Maruszewski): Μετάφραση στα Πολωνικά.

donahue@tiber.nist.gov (Michael J Donahue) : Τυπογραφικά λάθη, νικητής του "διαγωνισμού φέτας ψωμιού".

rms@gnu.ai.mit.edu (Richard Stallman) : Η σημείωση της "ελεύθερης" τεκμηρίωσης και διανομής.

dak@Pool.Informatik.RWTH-Aachen.DE (David Kastrup) : Τα του NFS.

esr@snark.thyrsus.com (Eric Raymond) : Διάφορα κομμάτια.

Επίσης, με βοήθησαν αρκετά και όσοι μου έστειλαν mails με ερωτήσεις και προβλήματα.

12.4 Copyright, άδεια χρήσης, και τα παρόμοια

Συστάσεις : Η εμπορική αναδιανομή επιτρέπεται και ενθαρρύνεται. Ωστόσο, θερμά συνίσταται ο αναδιανομέας να επικοινωνήσει με τον συγγραφέα πριν την αναδιανομή, ώστε να έχουμε σύγχρονες εκδόσεις. (Μπορείτε να μου στείλετε ένα αντίγραφο αυτού που ετοιμάζετε, ενόσω το ετοιμάζετε.) Επίσης, οι μεταφραστές παρακαλούνται να επικοινωνούν με τον συγγραφέα πριν τη μετάφραση. Η έντυπη μορφή είναι πιο όμορφη. Ανακυκλώστε.