Root over nfs clients & server Howto.

Hans de Goede *hans@highrise.nl* v1.0 30 Μαρτίου 1999

Πώς να στήσουμε έναν server και να ρυθμίσουμε τους clients για να δουλεύουν χωρίς δίσκο/ους μέσα ς' ένα δίκτυο.

Contents

1 Εισαγωγή

Αυτό το howto είναι επίσης διαθέσιμο στο - `<http://xmame.retrogames.com/hans>`. Αυτό το κείμενο περιγράφει το στήσιμο του "nfs over root". Διαφέρει από τα άλλα παρόμοια howto's του τύπου "root over nfs" σε δύο σημεία :

1. Περιγράφει και την πλευρά του server και την πλευρά του client, προσφέροντας μια πλήρη λύση. Δεν περιγράφει τις γενικές αρχές του "root over nfs" σχήματος, αν και αυτές θα φανούν καθαρά. Αντίθετα, προσφέρει ένα στήσιμο (που δουλεύει στην πράξη) για το "root over nfs". Ένα από τα πολλά πιθανά στησίματα, θα μπορούσα να πω.
2. Αυτή η λύση είναι μοναδική, υπό την έννοια ότι μοιράζεται το root του server μαζί με τους τερματικούς υπολογιστές (workstations - ws), αντί να έχει ένα mini-root ανά ws. Έτσι έχουμε κάμποσα πλεονεκτήματα :
   - Χρήση μικρού ποσοστού της χωρητικότητας του δίσκου.
   - Κάθε αλλαγή στην πλευρά του server αυτόματα γίνεται και στην πλευρά του client, δηλ. όλες οι ρυθμίσεις γίνονται μόνο μία φορά!
   - Προσθέτουμε πολύ εύκολα νέους clients.
   - Έχουμε μόνον έναν Η/Υ να συντηρήσουμε.

Αυτό το κείμενο βασίζεται κυρίως στο τί γίνεται ς' έναν Η/Υ με RedHat 5.2 . Για να το διαβάσετε, απαιτείται κάμποση εμπειρία ως SysAdmin σε Linux. Εάν την έχετε, δεν θα είναι πρόβλημα να προσαρμόσετε τη λύση αυτήν εδώ και σε άλλες διανομές.

1.1 Πνευματικά Δικαιώματα

Ορίστε και η στάνταρ νομική πλευρά των howto's :

1.2 Κατάλογος αλλαγών

- v0.1, 20 Ιανουαρίου 1999 : Το πρώτο πρόχειρο κείμενο γράφτηκε στο HHS, όπου και στήθηκαν οι ρυθμίσεις που περιγράφονται.
- v1.0, 30 Μαρτίου 1999 : Γράφτηκε ένα μέρος της πρώτης μορφής προς διανομή, κατά τον χρόνο που ήμουνα στην ISM.

2 Βασικές αρχές

Καθώς ήδη είπαμε, οι clients μοιράζονται ολόκληρο το root-fs με τον server. Αλλά, φυσικά, οι clients μπορούν μόνο να το διαβάσουν : Αυτός είναι ο τρόπος λειτουργίας του συγκεκριμένου συστήματος.

2.1 Τα πράγματα δεν είναι τόσο απλά

Δυστυχώς, τα πράγματα δεν είναι τόσο απλά. Υπάρχουν καναδυό προβλήματα, που πρέπει να ξεπεράσουμε ς' αυτό το απλό σχήμα.

2.1.1 Κάθε ws χρειάζεται το (με δυνατότητα εγγραφής) δικό του αντίγραφο ενός αριθμού από καταλόγους.

Στο συνηθισμένο στήσιμο, το Linux πρέπει να μπορεί να γράφει στους εξής καταλόγους :

1. /dev

2. /var
3. /tmp

Υπάρχουν 3 λύσεις γι' αυτό, από τις οποίες η μία θα δουλέψει μόνο στο /dev :

1. φορτώνουμε ένα ramdisk και το γεμίζουμε με untarring ενός tarball, ή αντιγράφοντας ένα πρότυπο (template) κατάλογο.
   - Πλεονεκτήματα :
     (a) Καθαρίζει με κάθε επανεκκίνηση, η οποία σβήνει αρχεία .tmp και αρχεία logs. Δεν χρειάζεται συντήρηση, αντίθετα απ' ό,τι οι κατάλογοι στην πλευρά του server.
     (b) Δεν καταλαμβάνει χώρο στον server, και δε δημιουργεί κυκλοφορία στο δίκτυο. Ένα ramdisk χρησιμοποιεί λιγότερους πόρους του server και του δικτύου, και είναι ταχύτερο.
   - Μειονεκτήματα :
     (a) Καταλαμβάνει μνήμη.
     (b) Τα logs δεν διατηρούνται μετά από επανεκκίνηση. Εάν πραγματικά χρειάζεστε logging για όλα τα clients, πείτε στο syslog να επανακατευθύνει το logging προς τον server.
2. δημιουργούμε ένα dir για κάθε ws επάνω στον server, και το φορτώνουμε rw επάνω στο nfs.
   - Πλεονεκτήματα & μειονεκτήματα :
     (a) Για τα dirs του server, τα παραπάνω πλεονεκτήματα-μειονεκτήματα αντιστρέφονται.
3. Στον kernel 2.2, το devfs μπορεί να χρησιμοποιηθεί αντί του /dev . Αυτό είναι ένα virtual filesystem, σαν το /proc του /dev.
   - (a) Πλεονεκτήματα :
     (b) Το devfs χρειάζεται ελάχιστη μνήμη, μοιάζει με ramdisk / καθόλου χώρο σκληρού δίσκου επάνω στον server, και είναι ταχύτατο. Ένα συνηθισμένο /dev χρειάζεται τουλάχιστον 1.5 MB, αφού το ελάχιστο μέγεθος αρχείου (και άρα ενός device) είναι 1k, και υπάρχουν περίπου 1200 devices. Φυσικά, μπορείτε να χρησιμοποιήσετε ένα πρότυπο "απογυμνωμένου" /dev, με μόνο τα περιεχόμενα που θέλετε, για να εξοικονομήσετε κάμποσο χώρο. Το 1.5 MB είναι πολύ για ramdisk, και επίσης δεν είναι ωραίο επάνω στον server.
     (c) Το devfs αυτόματα δημιουργεί καταχωρήσεις για τις νέες και τις ανιχνευμένες συσκευές, συνεπώς δεν χρειάζεται συντήρηση.
   - Μειονεκτήματα :
     (a) Χάνεται κάθε αλλαγή στο /dev , όπως η δημιουργία symlinks για το ποντίκι και το cd-rom. Το devfs έχει ένα script, το rc.devfs, που σώζει αυτές τις αλλαγές. Τα scripts, που σας δίνω ς' αυτό εδώ το howto, αποκαθιστούν αυτόματα τις ρυθμίσεις των symlinks, καλώντας το rc.devfs . Αν κάνετε οποιεσδήποτε αλλαγές στο /dev , χρειάζεται να καλέσετε εσείς το rc.devfs, για να τις σώσετε, δίνοντας :

        /etc/rc.d/rc.devfs save /etc/sysconfig

Όπως βλέπετε, υπάρχουν κάμποσοι τρόποι για να λυθεί αυτό το πρόβλημα. Για το υπόλοιπο μέρος αυτού του howto, κρατάμε τις ακόλουθες επιλογές :

- Αντί για το /dev , θα χρησιμοποιούμε το devfs.
- Αντί για τα /var και /tmp , θα χρησιμοποιούμε ένα διαμοιραζόμενο (shared) ramdisk του ενός MB. (Το κάνουμε shared, για να χρησιμοποιήσουμε τον χώρο του όσο το δυνατόν αποτελεσματικότερα.) Το /tmp αντικαθίσταται μ' ένα symlink προς το /var/tmp , για να κάνουμε εφικτό τον διαμοιρασμό.
- Δουλεύει εξ ίσου καλά το να γεμίσουμε το ramdisk με tarballs, ή πρότυπα (template) dirs. Αλλά με τα πρότυπα directories είναι πολύ ευκολότερο να κάνουμε αλλαγές, άρα θα χρησιμοποιήσουμε αυτά.

2.1.2 Μπορεί ν' αναγκαστούμε να δώσουμε δικαίωμα εγγραφής στο /home

Δεν είναι πραγματικό πρόβλημα, αφού σε κάθε στήσιμο client/server σε *nix συστήματα το /home φορτώνεται rw από τον server. 'Αρα, θα κάνουμε ακριβώς αυτό! ;)

2.1.3 Πώς βρίσκει ένας ws το ip του, ώστε να επικοινωνήσει με τον server;

Ευτυχώς για μας, αυτό το πρόβλημα έχει ήδη λυθεί, και ο πυρήνας του Linux υποστηρίζει δύο τρόπους αυτόματου καθορισμού της διεύθυνσης ip :

1. RARP
2. Bootp

Το rarp είναι ευκολότερο στη ρύθμιση, το bootp είναι το πιο ευέλικτο. Μιά που οι περισσότερες bootroms (ROMs εκκίνησης από κάρτα δικτύου) υποστηρίζουν μόνο το bootp, αυτό και θα χρησιμοποιήσουμε.

2.1.4 Τί γίνεται με τις ρυθμίσεις για κάθε ws

Στο RedH at, τα περισσότερα αρχεία ρυθμίσεων που εξαρτώνται από τον συγκεκριμένο Η/Υ, βρίσκονται ήδη στο /etc/sysconfig . Εμείς θα μετακινήσουμε μονάχα όσα δεν βρίσκονται εκεί, και θα βάλουμε symlinks. Μετά, θα φορτώσουμε ένα ξεχωριστό /etc/sysconfig για κάθε ws. Κι αυτό είναι το μόνο μέρος των ρυθμίσεων, που εξαρτάται από τη distribution : Σε άλλες distributions, μπορείτε απλά να φτιάξετε ένα directory ρυθμίσεων, να μετακινήσετε όλα τα μη διαμοιραζόμενα αρχεία ρυθμίσεων εκεί, και να δημιουργήσετε symlinks. Επίσης, το /etc/rc.d/rc3.d (ή τα παρόμοια των υπολοίπων distributions) μπορεί να χρειαστεί να διαφοροποιηθούν στον server, απ' ό,τι είναι στους workstations. Υποθέτοντας ότι όλοι οι ws τρέχουνε τις ίδιες services στο runlevel 3, θα φτιάξουμε ξεχωριστά runlevels 3 για τους workstations και τον server :

1. κατασκευή του /etc/rc.d/rc3.ws και του /etc/rc.d/rc3.server
2. κάνουμε το /etc/rc.d/rc3.d symlink προς το /etc/sysconfig/rc3.d
3. κάνουμε το /etc/sysconfig/rc3.d symlink προς το κατάλληλο /etc/rc.d/rc3.xxx
4. αντικαθιστούμε το S99local στο rc3.ws μ' ένα link προς το /etc/sysconfig/rc.local , ώστε κάθε ws να έχει το δικό του rc.local

2.1.5 Διάφορα προβλήματα

Υπάρχουν ακόμη κάποια προβλήματα :

1. Το /etc/rc.d/rc.sysinit χρειάζεται το /var, άρα το /var πρέπει να φορτωθεί ή να δημιουργηθεί πριν τρέξει το /etc/rc.d/rc.sysinit . Επίσης, καλό θα ήταν το για κάθε ws /etc/sysconfig να φορτωθεί πριν τρέξουν οποιαδήποτε initscripts.
    - Θα δώσουμε τον κώδικα για ένα script εκκίνησης για ws, επάνω-επάνω στο /etc/rc.d/rc.sysinit . Σημειώστε ότι αυτό το script θα το τρέξει (φυσικά) και ο server κατά την εκκίνηση, άρα το script πρέπει να προβλέψει αυτό το ενδεχόμενο, και να μην κάνει τίποτε επάνω στον server.
2. Πρέπει να μπορούμε να γράψουμε στο /etc/mtab :
    - Αυτό εδώ μπορεί να μας κάνει κόλπα! Απλά φτιάξτε ένα link προς το /proc/mounts , και επίσης φτιάξτε ένα άδειο αρχείο mounts στο /proc , ώστε τα fsck και mount να μην παραπονεθούν όσο τρέχουν τα initscripts, όταν το /proc δεν έχει φορτωθεί ακόμη. Σημείωση : Το smb(u)mount δεν σέβεται το να είναι link το mtab, και γράφεται πάνω του. Αρα, αν θέλετε να χρησιμοποιήσετε το smb(u)mount, φτιάξτε wrapper scripts, που αποκαθιστούν το symlink.

3 Προετοιμάζοντας τον server

Τώρα έφτασε η στιγμή να ετοιμάσουμε τον server να εξυπηρετήσει diskless clients.

3.1 Χτίζοντας έναν πυρήνα.

Το πρώτο πράγμα που πρέπει να κάνουμε, είναι να χτίσουμε έναν πυρήνα με τα αναγκαία πράγματα για υποστήριξη "root over nfs". Κάντε τα ακόλουθα βήματα :

1. Αφού θα χρησιμοποιήσουμε το RedH at 5.2 με πυρήνα 2.2, πρέπει να βεβαιωθούμε ότι το RedHat 5.2 μας είναι έτοιμο για τον πυρήνα 2.2 . Η RedHat διαθέτει ένα εξαιρετικό HowTo επάνω ς' αυτό το θέμα.
2. Προσωπικά, χρησιμοποίησα τον ίδιο πυρήνα και για τον server και για τους ws's, για ν' αποφύγω ασυμφωνίες των modules (module conflicts), αφού server και ws's μοιράζονται τα ίδια /lib/modules . Εάν αυτό δεν είναι δυνατόν στο δικό σας δίκτυο, απομιμηθείτε διαφορετικές εκδόσεις του πυρήνα, αλλάζοντας τον αριθμό έκδοσης στο επάνω-επάνω makefile του πυρήνα σας. Οι διαφορετικοί αριθμοί εκδόσεων θα μηδενίσουν το ενδεχόμενο ύπαρξης conflicts.
3. Εκτός από τα συνηθισμένα, ο πυρήνας πρέπει να έχει τα εξής :
    - Το ext2 compiled μέσα του (εάν χρησιμοποιηθεί στον server, ή και σε client).
    - Τα nfs και root-over-nfs compiled μέσα του (εάν χρησιμοποιηθούν σε client, ή και στον server), ώστε το nfs over root στον πυρήνα 2.2 να ενεργοποιήσει το ip-autoconfig στο δίκτυο. Για τις ρυθμίσεις, θα χρησιμοποιήσουμε το bootp.
    - Υποστήριξη για κάρτα δικτύου στους ws's compiled μέσα του (αν χρησιμοποιηθεί σε client, ή και στον server).
    - Το devfs compiled μέσα του (απαιτείται για client, καλό όμως να υπάρχει και στον server).

- Ο,τιδήποτε άλλο χρησιμοποιείτε συνήθως, δηλ. modules για οποιοδήποτε άλλο device υπάρχει επάνω στον server, ή όλους / μερικούς ws's, κλπ κλπ.

4. Το source πρόγραμμα του πυρήνα χρειάζεται διόρθωση, για ν' αλλάξουμε το εξ ορισμού "root-over-nfs" mount σε /tftpboot/<ip>/root , αντί του /tftpboot/<ip>. Αυτό γίνεται για να πάρουμε ένα "καθαρό" δέντρο από directories στο /tftpboot , με ένα ανά dir ανά ws, που θα περιέχει τόσο το root για τον ws (ένα link προς το πραγματικό root του server), όσο και κάθε directory, που χρειάζεται ειδικά για τον ws..

   - Για πυρήνα 2.0 : Ψάχνουμε ένα "define" στο : "include/linux/nfs_fs.h" , με το όνομα "NFS_ROOT" .
   - Για πυρήνα 2.2, το αντίστοιχο "define" βρίσκεται στο : "fs/nfs/nfsroot.c" .

5. Τώρα κάνουμε ένα συνηθισμένο compile στον πυρήνα (βλέπε το "Kernel H owT o").

6. Εάν ακόμη δεν έχετε ένα /dev/nfsroot , φτιάξτε το δίνοντας :

   mknod /dev/nfsroot b 0 255.

7. Αφού κάνετε compile στον πυρήνα, βάλτε το root στο nfsroot, δίνοντας :

   rdev <path-to-zImage>/zImage /dev/nfsroot

8. Πριν bootάρετε με το devfs, πρέπει να κάνετε λίγες αλλαγές στο /etc/conf.modules : Βάλτε τα περιεχόμενα του conf.modules στο μέρος του devfs που αναφέρεται ς' αυτό.

9. Αφού αυτός ο νέος πυρήνας είναι φτιαγμένος για αυτόματη ρύθμιση των ip's, κατά την εκκίνηση θα προσπαθήσει να ρυθμίσει και το ip του server. Πράγμα το οποίο, φυσικά, θα αποτύχει, μιά που ο server δίνει τα ip's. Για ν' αποφύγετε τα μεγάλα timeouts, προσθέστε : append="ip=off" στο μέρος του /etc/lilo.conf για το Linux.

10. Τρέξτε το lilo και ξεκινήστε τον νέο πυρήνα.

11. Εξ αιτίας του devfs, θα έχετε χάσει όλα τα symlinks επάνω στον server. Με το RedHat, αυτά είναι συνήθως τα /dev/mouse και /dev/cdrom. Ξαναφτιάξτε αυτά τα δύο. Επίσης, αν χρησιμοποιείτε ειδικές ownerships, κάντε chown προς τα κατάλληλα αρχεία στο /dev. Τώρα, σώστε τις ρυθμίσεις του /dev (στο /etc/sysconfig, αφού μπορεί να εξαρτώνται από τους ws's) :

    - Αντιγράψτε το rc.devfs από το μέρος του devfs στο source του πυρήνα, στο /etc/rc.d/rc.devfs και κάντε το εκτελέσιμο.
    - Σώστε τις ρυθμίσεις, δίνοντας :

      /etc/rc.d/rc.devfs save /etc/sysconfig

3.2 Φτιάχνοντας και γεμίζοντάς το /tftpboot, δημιουργώντας symlinks για το /tmp, κλπ.

Το επόμενο βήμα είναι να φτιάξουμε και να γεμίσουμε το /tftpboot

3.2.1 Το αυτόματο μέρος

Αυτό το χειρίζεται ολόκληρο ένα μεγάλο script, μιά που το να συμπεριλάβω μιά μεγάλη λίστα εντολών ς' αυτό το howto μου φάνηκε εντελώς άχρηστο. Αν θέλετε να εφαρμόσετε αυτό εδώ το εγχειρίδιο, απλά διαβάστε το script και πληκτρολογήστε ό,τι βλέπετε! ;)

Αυτό το script ρυθμίζει "βρώμικα" κάποια πράγματα, πχ nukάρει το /tmp, κάνει προσωρινό kill στο syslog, κάνει umount στο /proc. Συνεπώς, βεβαιωθείτε ότι (α) κανένας άλλος δεν χρησιμοποιεί τον Η/Υ κατά τη διάρκεια αυτής της διαδικασίας, και ότι (β) δεν τρέχουνε τα X-Windows. Αρκεί το να βεβαιωθείτε ότι είσαστε το μόνο άτομο που έκανε login σε περιβάλλον κονσόλας, δεν χρειάζεται ν' αλλάξετε τα runlevels.

ΑΠΟΚΗΡΥΞΗ : Αυτό το script είναι δοκιμασμένο, αλλά αν ωστόσο κάνει μπάχαλο τον server σας, δεν θα σας βοηθήσω. Δεν μπορώ να πάρω οποιαδήποτε ευθύνη. Πρέπει να επαναλάβω ότι αυτό το HowTo προορίζεται μόνο για έμπειρους Linux SysAdmins. Επίσης, αυτό το script σχεδιάστηκε για να χρησιμοποιηθεί μόνο μία φορά, και εννοώ ΜΙΑ φορά. Αν το τρέξετε δύο φορές, θα nukάρει τα : /etc/fstab, /etc/X11/XF86Config, /etc/X11/X και /etc/conf.modules.

Τώρα, αφού είπαμε τα παραπάνω, απλά κάντε cut/paste το script, κάντε το εκτελέσιμο, εκτελέστε το, και προσευχηθείτε στον Άγιο Πιγκουίνο να δουλέψει! ;)

```
#!/bin/sh

SERVER_NAME=`hostname -s`

###
echo ftiaxnoume to /etc/rc.d/rc.ws
#auto edw apla kanei echo s' olo to script ;)
echo "# gia to root on nfs

SERVER=$SERVER_NAME

#xreiazomaste to proc gia to mtab, route klp
mount -t proc /proc /proc

IP=\`ifconfig eth0|grep inet|cut --field 2 -d ':'|cut --field 1 -d ' '\`

#an apotuxei to prwto mount, pi8anotata o server, h toulaxiston kati allo
#den pane kala, ara kanoume monaxa ta upoloipa, an to prwto mount petuxei
mount \$SERVER:/tftpboot/\$IP/sysconfig /etc/sysconfig -o nolock &&
{
   #other mounts
   mount \$SERVER:/home /home -o nolock
   mount \$SERVER:/ /\$SERVER -o ro,nolock

   #/var
   echo Ftiaxnoume to /var ...
   mke2fs -q -i 1024 /dev/ram1 1024
   mount /dev/ram1 /var -o defaults,rw
   cp -a /tftpboot/var /
```

```
   #network stuff
   . /etc/sysconfig/network
   HOSTNAME=\`cat /etc/hosts|grep \$IP|cut --field 2\`
   route add default gw \$GATEWAY
   ifup lo
}

#restore devfs settings
/etc/rc.d/rc.devfs restore /etc/sysconfig

umount /proc" > /etc/rc.d/rc.ws

###
echo xwrizoume to runlevel 3 gia ton client kai gia ton server
mv /etc/rc.d/rc3.d /etc/rc.d/rc3.server
cp -a /etc/rc.d/rc3.server /etc/rc.d/rc3.ws
rm /etc/rc.d/rc3.ws/*network
rm /etc/rc.d/rc3.ws/*nfs
rm /etc/rc.d/rc3.ws/*nfsfs
rm /etc/rc.d/rc3.ws/S99local
ln -s /etc/sysconfig/rc.local /etc/rc.d/rc3.ws/S99local
ln -s /etc/rc.d/rc3.server /etc/sysconfig/rc3.d
ln -s /etc/sysconfig/rc3.d /etc/rc.d/rc3.d

###
echo kanoume to tmp link gia to /var/tmp
rm -fR /tmp
ln -s var/tmp /tmp

###
echo metakinoume diafora arxeia kai ftiaxnoume symlinks gi auta
echo mtab
/etc/rc.d/init.d/syslog stop
umount /proc
touch /proc/mounts
mount /proc
/etc/rc.d/init.d/syslog start
rm /etc/mtab
ln -s /proc/mounts /etc/mtab
echo fstab
mv /etc/fstab /etc/sysconfig
ln -s sysconfig/fstab /etc/fstab
echo arxeia ry8misewn twn X
mkdir /etc/sysconfig/X11
mv /etc/X11/X /etc/sysconfig/X11
ln -s ../sysconfig/X11/X /etc/X11/X
```

```
mv /etc/X11/XF86Config /etc/sysconfig/X11
ln -s ../sysconfig/X11/XF86Config /etc/X11/XF86Config
echo conf.modules
mv /etc/conf.modules /etc/sysconfig
ln -s sysconfig/conf.modules /etc/conf.modules
echo isapnp.conf
mv /etc/isapnp.conf /etc/sysconfig
ln -s sysconfig/isapnp.conf /etc/isapnp.conf

###
echo ftiaxnoume ena protupo dir gia ta directories twn wss
echo /tftpboot/template
mkdir /home/tftpboot
ln -s home/tftpboot /tftpboot
mkdir /tftpboot/template
mkdir /$SERVER_NAME
echo root
ln -s / /tftpboot/template/root
echo sysconfig
cp -a /etc/sysconfig /tftpboot/template/sysconfig
rm -fR /tftpboot/template/sysconfig/network-scripts
ln -s /$SERVER_NAME/etc/sysconfig/network-scripts \
 /tftpboot/template/sysconfig/network-scripts
echo NETWORKING=yes > /tftpboot/template/sysconfig/network
echo `grep "GATEWAY=" /etc/sysconfig/network` >> /tftpboot/template/sysconfig/network
echo "/dev/nfsroot / nfs defaults 1 1" > /tftpboot/template/sysconfig/fstab
echo "none /proc proc defaults 0 0" >> /tftpboot/template/sysconfig/fstab
echo "#!/bin/sh" > /tftpboot/template/sysconfig/rc.local
chmod 755 /tftpboot/template/sysconfig/rc.local
rm /tftpboot/template/sysconfig/rc3.d
ln -s /etc/rc.d/rc3.ws /tftpboot/template/sysconfig/rc3.d
rm /tftpboot/template/sysconfig/isapnp.conf
echo var
cp -a /var /tftpboot/var
rm -fR /tftpboot/var/lib
ln -s /$SERVER_NAME/var/lib /tftpboot/var/lib
rm -fR /tftpboot/var/catman
ln -s /$SERVER_NAME/var/catman /tftpboot/var/catman
rm -fR /tftpboot/var/log/httpd
rm -f /tftpboot/var/log/samba/*
for i in `find /tftpboot/var/log -type f`; do cat /dev/null > $i; done
rm `find /tftpboot/var/lock -type f`
rm `find /tftpboot/var/run -type f`
echo /sbin/fsck.nfs
echo "#!/bin/sh
exit 0" > /sbin/fsck.nfs
chmod 755 /sbin/fsck.nfs
```

```
echo ola teleiwmena
```

3.2.2 Ρυθμίσεις με το χέρι για μερικά αρχεία

Τώρα πρέπει να κάνουμε μερικές ρυθμίσεις με το χέρι στον server :

1. Το source του script ρυθμίσεων των ws's πρέπει να το βάλουμε επάνω-επάνω στο rc.sysinit, συνεπώς βάλτε τις εξής γραμμές αμέσως μετά τον ορισμό του PATH :

```
#Gia root over nfs workstations.
/etc/rc.d/rc.ws
```

2. Απλοποιείστε το /etc/rc.d/rc3.ws στο ελάχιστο δυνατόν. Μπορεί να σας φανεί χρήσιμο να φτιάξετε κάτι σαν : rc.local.ws , αλλά αυτό το αφήνω σε σας. Το δίκτυο και το nfs-fs είναι ήδη ρυθμισμένα. Το script έχει ήδη σβήσει ή ανανεώσει τα παρακάτω :
   - δίκτυο
   - nfsfs
   - nfs
   - rc.local

3.3 Κάνουμε export τα κατάλληλα file systems και ρυθμίζουμε το bootp

Φυσικά, ο server πρέπει να κάνει export στα κατάλληλα filesystems, και να δώσει τις διευθύνσεις ip στους clients.

3.3.1 Κάνουμε export τα κατάλληλα file systems

Πρέπει να κάνουμε export μερικά dir's για τους ws's, επομένως για το εργαστήριο εδώ στο πανεπιστήμιο έκανα τις ακόλουθες προσθήκες στο /etc/exports :

```
/ *.st.hhs.nl(ro,no_root_squash)
/home *.st.hhs.nl(rw,no_root_squash)
```

Φυσικά, χρησιμοποιείστε το κατάλληλο domain ;) και επανεκκινήστε το nfs, δίνοντας :

/etc/rc.d/init.d/nfs restart

Σημείωση για χρήστες knfsd : Το knfsd δεν επιτρέπει να έχουμε πολλαπλά exports επάνω σε μία partition με διαφορετικές άδειες. Επίσης, το knfsd δεν επιτρέπει τα clients να υπερβούν τα όρια των partitions. Πχ, αν ένα client κάνει mount στο / , και το /usr είναι σε διαφορετική partition, τότε δεν θα δει το /usr. Έτσι, αν χρησιμοποιείτε το knfsd, επάνω σε διαφορετική partition θα έπρεπε να βρίσκεται τουλάχιστον το /home . Το script προετοιμασίας του server ήδη βάζει το /tftpboot στο /home , άρα αυτό δεν χρειάζεται να μπει σε ξεχωριστή partition. Εάν υπάρχουν κι άλλες partitions, στις οποίες οι clients πρέπει να έχουν πρόσβαση, κάντε ξεχωριστό export γι' αυτές, και προσθέστε εντολές mount στο /etc/rc.d/rc.ws .

3.3.2 Ρυθμίζοντας το bootp

1. Αν το bootp δεν είναι ήδη εγκατεστημένο, εγκαταστήστε το. Περιέχεται στο RedHat.
2. Διορθώστε το /etc/inetd.conf και βγάλτε το σχόλιο από τη γραμμή που αρχίζει με το bootps. Αν θέλετε να χρησιμοποιήσετε EPROM εκκίνησης, βγάλτε το σχόλιο μπροστά από από τη γραμμή που μιλάει για το tftp.
3. Επανεκκινήστε τον inetd, δίνοντας :

   /etc/rc.d/init.d/inetd restart

4 Προσθέτοντας workstations

Τώρα, που ο server είναι έτοιμος, μπορούμε ν' αρχίσουμε να προσθέτουμε ws's.

4.1 Φτιάχνοντας δισκέτα ή EPROM εκκίνησης

Δεν χρειάζεται να τις φτιάξετε και τις δύο μαζί, για να ξεκινήσει ο ws.

4.1.1 Φτιάχνοντας δισκέτα εκκίνησης

Ακόμη κι αν θέλετε να χρησιμοποιήσετε EPROM εκκίνησης, είναι χρήσιμο να δοκιμάσετε πρώτα την εκκίνηση με μιά δισκέτα. Για να φτιάξετε τη δισκέτα, δώστε :

dd if=/<path-to-zImage>/zImage of=/dev/fd0

4.1.2 Φτιάχνοντας EPROM εκκίνησης

Υπάρχουν κάμποσα free προγράμματα για κατασκευή EPROM εκκίνησης :

1. Το netboot, που κατά την ταπεινή μου γνώμη είναι το πληρέστερο free πρόγραμμα που υπάρχει. Αυτό χρησιμοποιεί στάνταρ drivers για DOS, συνεπώς υποστηρίζονται σχεδόν όλες οι κάρτες. Μιά πολύ χρήσιμη υπόδειξη που πήρα από την mailing list τους, ήταν να σπάσω μερικούς drivers με το pklite, αφού αυτοί ήτανε πολύ μεγάλοι, ώστε να χωρέσουν σε μιά EPROM εκκίνησης. Η τεκμηρίωση του netboot είναι αρκετά πλήρης, έτσι δεν θα χάσω χρόνο να την αντιγράψω εδώ. Είναι επαρκής για να φτιάξετε μία EPROM εκκίνησης, και να εκκινήσετε έναν ws μ' αυτήν. Η σελίδα του netboot είναι η : `<http://www.han.de/gero/netboot/>` .
2. Το etherboot, ένα άλλο free πρόγραμμα που έχει μερικά όμορφα χαρακτηριστικά, όπως η υποστήριξη dhcp, αλλά έχει περιορισμένη υποστήριξη για drivers, μιά που χρησιμοποιεί το δικό του driver format. Αυτό δεν το έχω χρησιμοποιήσει, άρα δεν μπορώ να δώσω άλλες χρήσιμες πληροφορίες. Η σελίδα του Etherboot είναι η : `<http://www.slug.org.au/etherboot/>` .

Σχετικά με τις ίδες τις EPROMs : Οι περισσότερες κάρτες δικτύου δέχονται συνηθισμένες EPROMs σε βάση 28 ακίδων dip. Αυτές οι EPROMs φτάνουν μέχρι τα 64kB. Για τις περισσότερες κάρτες θα χρειαστείτε EPROMs των 32kB με το netboot. Μερικοί drivers χωράνε σε 16kB, αλλά η διαφορά τιμής των σχετικών EPROMs είναι ελάχιστη. Αυτές οι EPROMs γράφονται με οποιοδήποτε συνηθισμένο EPROM burner.

4.2 Φτιάχνοντας ένα ws dir

Απλά αντιγράψτε το πρότυπο dir, δίνοντας :

cd /tftpbootcp -a template <ip>

Φυσικά, θα μπορούσατε απλά ν' αντιγράψετε τα dirs των ws's με ίδιο mouse, ίδια κάρτα γραφικών και ίδιο monitor, και να παραλείψετε τις ρυθμίσεις στο βήμα 5.4 .

4.3 Προσθέτοντας γραμμές στα /etc/bootptab και /etc/hosts

Διορθώστε το /etc/bootptab και προσθέστε μιά γραμμή για τον ws των δοκιμών. Πχ, μιά κατάλληλη γραμμή είναι :

```
nfsroot1:hd=/tftpboot:vm=auto:ip=10.0.0.237:\
:ht=ethernet:ha=00201889EE78:\
:bf=bootImage:rp=/tftpboot/10.0.0.237/root
```

Αντικαταστήστε το nfsroot1 με το όνομα host, που θέλετε να έχει ο ws. Αντικαταστήστε το 10.0.0.237 με το ip που θέλετε να έχει ο ws (κάντε το αυτό 2 φορές), και αντικαταστήστε το 00201889EE78 με την αντίστοιχη διεύθυνση μηχανής (της κάρτας δικτύου) του ws. Αν δεν γνωρίζετε τη διεύθυνση μηχανής του ws, απλά εκκινήστε τον με τη δισκέτα εκκίνησης που μόλις φτιάξατε, και ψάξτε τον αριθμό της διεύθυνσης μηχανής ανάμεσα στα μηνύματα εκκίνησης. Υπάρχει μιά πιθανότητα το bootp να τρέχει ήδη, συνεπώς βεβαιωθείτε ξαναξεκινώντας το με :

killall -HUP bootpd

Μην ανησυχείτε αν αποτύχει αυτό, αυτό σημαίνει ότι το bootp δεν έτρεχε. Επίσης, μπορεί να το ξεκινήσει και ο inetd, αν του το ζητήσουμε.

4.4 Εκκινώντας τον ws για πρώτη φορά

Απλά ξεκινήστε τον ws με τη δισκέτα. Αυτή η διαδικασία θα πρέπει να δώσει έναν ws που δουλεύει σε text mode, με ακριβώς τις ίδιες ρυθμίσεις του server, εκτός αυτών για τα ip-nr και τις services που τρέχουνε. Ακόμη κι αν θέλετε να βάλετε EPROM εκκίνησης, θα ήταν έξυπνο πρώτα να δοκιμάσετε με τη δισκέτα. Αν δουλέψει η δισκέτα, μπορείτε μετά να δοκιμάσετε με την EPROM. Για περισσότερα, διαβάστε την τεκμηρίωση για EPROMs εκκίνησης.

4.5 Κάνοντας τις ειδικές για τον ws ρυθμίσεις.

Τώρα, ήρθε η στιγμή να κάνουμε τις ειδικές ρυθμίσεις του ws :

1. Πρώτα-πρώτα, για να δουλέψει το mouse, απλά τρέχουμε το mouseconfig. Για να εφαρμόσουμε τις αλλαγές και να δοκιμάσουμε αν δουλεύει το ποντίκι, δίνουμε :

/etc/rc.d/init.d restart

2. Τρέχουμε το Xconfigurator. Όταν αυτό βρει την κάρτα οθόνης και σας ζητήσει να πατήσετε ok, μην το κάνετε! Αφού έχουμε μετακινήσει το symlink του Xserver από το /etc/X11/X στο /etc/sysconfig/X11/X , το Xconfigurator δεν θα φτιάξει το σωστό link. Έτσι, για να σιγουρευτούμε ότι και οι υπόλοιπες ρυθμίσεις του Xconfigurator θα γίνουν σωστά, γυρίζουμε σε μιά άλλη κονσόλα, και φτιάχνουμε το link στο /etc/sysconfig/X11 για τον συγκεκριμένο server. Μετά, απλά τελειώνουμε με το Xconfigurator και δοκιμάζουμε τα X.

3. Ρυθμίζουμε ο,τιδήποτε διαφορετικό απ' όσα έχει ο server και το πρότυπο dir :

   - Ηχος : Πιθανότατα θα χρειαστεί ν' αλλάξετε τα isapnp.conf και conf.modules, διότι και τα δύο (από το script ρυθμίσεων του server) είναι ήδη έτοιμα links προς το /etc/sysconfig .
   - cd-rom : Ένα link στο /dev, μιά γραμμή στο /etc/fstab? , κλπ.
   - rc.local : Κάντε οποιεσδήποτε απαραίτητες αλλαγές.

4. Σώστε τα links και όποιες άλλες αλλαγές στο /dev . Δώστε :

   /etc/rc.d/rc.devfs save /etc/sysconfig

5. Όλα έτοιμα.

5 Πρόσθετο δωράκι : Εκκινώντας από cd-rom

Πολλά από τα παραπάνω εφαρμόζονται επίσης σε εκκίνηση από cd-rom. Μιά πού - έτσι κι αλλιώς - ήθελα να γράψω και για εκκίνηση από cd-rom, προσθέτω εδώ τις κατάλληλες πληροφορίες, ώστε να μην ξαναγράψω δυο φορές τα ίδια πράγματα.

Γιατί να θέλουμε να bootάρουμε έναν Η/Υ από cd-rom; Η εκκίνηση από cd-rom μας ενδιαφέρει σε περιπτώσεις, όπου θέλουμε να τρέξουμε μιά πολύ ειδική εφαρμογή, πχ ένα kiosk (ς.τ.μ. : Δες το Kiosk HowTo), ένα πρόγραμμα βάσης δεδομένων βιβλιοθήκης, ή ένα intenet cafe, και δεν έχουμε δίκτυο ή server, για να κάνουμε στήσιμο root over nfs.

5.1 Βασική αρχή

Η βασική αρχή πάλι είναι απλή : να εκκινήσουμε ως root μ' ένα cd-rom. Για να μπορέσουμε : Θα χρησιμοποιήσουμε την επέκταση rockridge, για να τοποθετήσουμε ένα unix-like filesystem επάνω στο cd, και την επέκταση Eltorito, για να κάνουμε το cd εκκινήσιμο.

5.1.1 Τα πράγματα δεν μπορεί να είναι τόσο απλά

Φυσικά, αυτό το στήσιμο παρουσιάζει μερικά προβλήματα. Τα περισσότερα είναι τα ίδια με τα παραπάνω :

1. Θα χρειαστούμε πρόσβαση εγγραφής στα : /dev, /var & /tmp.

   - Θα χρησιμοποιήσουμε τις ίδιες λύσεις, όπως για το root over nfs (δες παραπάνω) :
     - Για το /dev , θα χρησιμοποιήσουμε το Devfs .

   – Για τα /var και /tmp , θα χρησιμοποιήσουμε ένα διαμοιρασμένο (shared) ramdisk του 1 MB. Το κάνουμε shared, για να χρησιμοποιήσουμε τον χώρο όσο αποτελεσματικότερα γίνεται. Το /tmp αντικαθίσταται μ' ένα symlink προς το /var/tmp , για να καταστήσουμε δυνατό τον διαμοιρασμό.
   – Δουλεύει εξ ίσου καλά το να γεμίσουμε το ramdisk με tarballs, ή πρότυπα (template) directories. Όμως, είναι πολύ ευκολότερο να κάνουμε αλλαγές με τα πρότυπα dirs, άρα θα χρησιμοποιήσουμε αυτά.

2. Μερικές εφαρμογές χρειάζονται πρόσβαση εγγραφής στο /home.
   - Βάλτε στο /var τα home directories των χρηστών που θα τρέξουν τις εφαρμογές αυτές, και γεμίστε τα με τα υπόλοιπα περιεχόμενα του /var σε κάθε επανεκκίνηση.
3. Το /etc/mtab πρέπει να μπορεί να γράφεται :
   - Φτιάξετε ένα link προς το /proc/mounts και δημιουργήστε ένα άδειο αρχείο mounts στο /proc, όπως παραπάνω.

5.2 Φτιάχνοντας ένα δοκιμαστικό στήσιμο.

Τώρα, που γνωρίζουμε τί να κάνουμε και πώς, ήρθε η στιγμή να φτιάξουμε ένα δοκιμαστικό στήσιμο :

1. Κατ' αρχήν, πάρτε έναν Η/Υ απ' αυτούς που θέλετε να χρησιμοποιήσετε, και βάλτε μέσα έναν μεγάλο σκληρό δίσκο κι ένα cd recorder.
2. Βάλτε το Linux που σας αρέσει επάνω ς' αυτό το μηχάνημα, κι αφήστε ελεύθερη μιά partition των 650 MB για το δοκιμαστικό στήσιμο. Αυτή η εγκατάσταση θα φανεί χρήσιμη στο να φτιάξουμε το iso-image του cd και να το ρίξουμε στα cd's εκκίνησης, άρα εγκαταστήστε τα απαραίτητα εργαλεία. Επίσης, θα μας χρειαστεί στο να διορθώσουμε όλα τα bugs, που δεν μας επιτρέπουν να bootάρουμε το μηχάνημα της δοκιμής.
3. Εγκαταστήστε το Linux της επιλογής σας επάνω στην partition των 650 MB, με το στήσιμο που θέλετε να περάσει και στο cd. Αυτό θα είναι και το στήσιμο που θα χρησιμοποιήσουμε στις δοκιμές.
4. Ξεκινήστε από τη δοκιμαστική partition.
5. Κάντε compile έναν πυρήνα, όπως περιγράψαμε στο τμήμα 3.1 . Ακολουθήστε όλα τα βήματα, διότι οι απαραίτητες αλλαγές για το devfs εξακολουθούν να χρειάζονται! Στο βήμα 3 του τμήματος 3.1, προσθέστε τα εξής :
   - isofs compiled μέσα στον πυρήνα
   - devfs compiled μέσα στον πυρήνα
   - υποστήριξη cdrom compiled μέσα στον πυρήνα
   - ό,τι άλλο χρειάζεστε, είτε compiled μέσα στον πυρήνα, είτε ως module.
6. Ρυθμίστε το δοκιμαστικό στήσιμο :
   - Δημιουργήστε τον χρήστη, που θα τρέξει την εφαρμογή.
   - Βάλτε το home directory του στο /var.
   - Αν χρειαστεί, εγκαταστήστε την εφαρμογή.

- Αν χρειαστεί, κάντε ρυθμίσεις για την εφαρμογή.
- Κάντε τις ρυθμίσεις του συγκεκριμένου χρήστη, ώστε η εφαρμογή να τρέξει αυτόματα μετά το login.
- Ρυθμίστε το Linux, ώστε αυτόματα να κάνει login για τον χρήστη.
- Ρυθμίστε ο,τιδήποτε άλλο χρειάζεται ρύθμιση.

7. Δοκιμάστε το ότι όλα δουλεύουν σωστά.

8. Εκκινήστε την κυρίως εγκατάσταση και κάντε mount την partition των 650 MB στο directory /test της κυρίως εγκατάστασης.

9. Βάλτε τα παρακάτω ς' ένα αρχείο, με το όνομα /test/etc/rc.d/rc.iso . Το source αυτού του αρχείου μπαίνει στην αρχή του rc.sysinit, για να φτιάξει το /var .

```
#/var
echo Ftiaxnoume to /var ...
mke2fs -q -i 1024 /dev/ram1 1024
mount /dev/ram1 /var -o defaults,rw
cp -a /lib/var /

#Epanaferoume tis ry8miseis tou devfs, xreiazomaste to proc
mount -t proc /proc /proc
/etc/rc.d/rc.devfs restore /etc/sysconfig
umount /proc
```

10. Διορθώνουμε το /test/etc/rc.sysinit , βάζοντας σχόλιο στις γραμμές όπου ο root φορτώνεται rw, και προσθέτουμε τις ακόλουθες 2 γραμμές αμέσως μετά τον ορισμό του PATH :

```
#gia na bootaroume apo cdrom
. /etc/rc.d/rc.iso
```

11. Αντιγράφουμε τα παρακάτω ς' ένα script και το εκτελούμε. Αυτό θα φτιάξει ένα πρότυπο για το /var και θα δημιουργήσει τα links των /tmp και /etc/mtab.

```
#!/bin/sh
echo tmp
rm -fR /test/tmp
ln -s var/tmp /test/tmp

###
echo mtab
touch /test/proc/mounts
rm /test/etc/mtab
ln -s /proc/mounts /test/etc/mtab
```

```
###
echo var
mv /test/var/lib /test/lib/var-lib
mv /test/var /test/lib
mkdir /test/var
ln -s /lib/var-lib /test/lib/var/lib
rm -fR /test/lib/var/catman
rm -fR /test/lib/var/log/httpd
rm -f /test/lib/var/log/samba/*
for i in `find /test/lib/var/log -type f`; do cat /dev/null > $i; done
rm `find /test/lib/var/lock -type f`
rm `find /test/lib/var/run -type f`
```

12. Αφαιρώντας τη δημιουργία του /etc/issue* από το /test/etc/rc.local, απλά θα το αναγκάσουμε να αποτύχει.

13. Τώρα επανεκκινούμε τη δοκιμαστική partition, η οποία θα διαβαστεί απλά σαν ένα cd-rom. Αν κάτι δεν δουλέψει, επανεκκινήστε με την κυρίως partition, διορθώστε το, ξαναδοκιμάστε το, κ.ο.κ. . Ή, μπορείτε να κάνετε remount / rw τη δοκιμαστική partition, να το διορθώσετε, και να επανεκκινήσετε κατευθείαν με τη δοκιμαστική partition ξανά. Για να κάνετε remount / rw , δώστε :

    mount -o remount,rw /

5.3 Φτιάχνοντας το cd

5.3.1 Φτιάχνοντας ένα boot image

Πρώτα-πρώτα, εκκινήστε με την κυρίως partition. Για να φτιάξετε ένα εκκινήσιμο cd, θα χρειαστείτε ένα image μιας bootable δισκέτας. Δεν θα δουλέψει το απλό dd ς' ένα zimage, επειδή ο loader στην αρχή του zimage δεν φαίνεται ν' αγαπάει την ψεύτικη δισκέτα, που δημιουργεί το εκκινήσιμο cd. Έτσι, θα χρησιμοποιήσουμε το syslinux :

1. Παίρνουμε το boot.img από ένα cd του RedHat

2. Κάνουμε mount το boot.img κάπου ανάμεσα στο loopback, δίνοντας :

    mount boot.img kapou -o loop -t vfat

3. Αφαιρούμε ο,τιδήποτε από το boot.img , εκτός από τα :

    - ldlinux.sys
    - syslinux.cfg

4. Αντιγράφουμε το kernel image της δοκιμαστικής partition στο boot.img .

5. Διορθώνουμε το syslinux.cfg , ώστε να περιέχει τα εξής (αντικαθιστώντας, φυσικά, το zImage με το κατάλληλο όνομα image) :

```
default linux

label linux
kernel zImage
append root=/dev/<balte edw to onoma tou cdrom sas>
```

6. Umount boot.img:

   umount kapou

7. Αν το /etc/mtab σας είναι link για το /proc/mounts , το umount δεν θ' αφήσει αυτόματα ελεύθερο το /dev/loop0 , συνεπώς ελευθερώστε το δίνοντας :

   losetup -d /dev/loop0

5.3.2 Δημιουργώντας το iso image

Αφού έχουμε το boot image και μιά εγκατάσταση που μπορεί να ξεκινήσει από ένα readonly mount, έφτασε η στιγμή να φτιάξουμε ένα iso image του cd :

1. Αντιγράψτε το boot.img στο /test
2. Κάντε cd στο directory, όπου θέλετε ν' αποθηκεύσετε το image. Βεβαιωθείτε ότι βρίσκεται επάνω σε μιά partition με αρκετό ελεύθερο χώρο.
3. Τώρα, φτιάξτε το image, πληκτρολογώντας :

   mkisofs -R -b boot.img -c boot.catalog -o boot.iso /test

5.3.3 Επαληθεύοντας το iso image

1. Κάντε mount το image ανάμεσα στο loopback device, δίνοντας :

   mount boot.iso kapou -o loop -t iso9660

2. Τώρα επιβεβαιώστε ότι τα περιεχόμενα είναι εντάξει.
3. Κάντε umount το boot.iso :

   umount kapou

4. Αν το /etc/mtab σας είναι link προς το /proc/mounts , το umount δεν θ' απελευθερώσει αυτόματα το /dev/loop0 . Αυτό ελευθερώστε το, πληκτρολογώντας :

   losetup -d /dev/loop0

5.3.4 Γράφοντας το CD

Υποθέτοντας ότι έχετε εγκαταστήσει και ρυθμίσει το cdrecord για τον εγγραφέα σας, δώστε :

cdrecord -v speed=<epi8umhth taxuthta eggrafhs> dev=<path pros to generic scsi device tou eggrafea sas> boot.iso

5.4 Bootάρετε το CD και δοκιμάστε το

Ε, ο τίτλος αυτής της παραγράφου τα λέει όλα! ;)

6 Ευχαριστώ

- Το HHS (Haagse Hoge School), ένα Ολλανδικό κολέγιο, όπου για πρώτη φορά έστησα και δοκίμασα αυτό το σχήμα σε καναδυό εργαστήρια. Και όπου γράφτηκε η αρχική έκδοση αυτού του HowTo.
- Την ISM, μιά Ολλανδική εταιρία, όπου εργάζομαι στο τελικό μου project. Ένα μέρος του project έχει να κάνει με diskless Η/Υ, άρα έπρεπε να ψάξω περισσότερο το συγκεκριμένο στήσιμο του δικτύου, και είχα τον χρόνο ν' αναθεωρήσω αυτό εδώ το HowTo.
- Όλους τους χρήστες που θα μου δώσουν χρήσιμες υποδείξεις, από τη στιγμή που αυτή εδώ η πρώτη έκδοση θα δει το φως της δημοσιότητας. ;)

7 Σχόλια

Σχόλια, υποδείξεις, κλπ είναι ευπρόσδεκτα. Μπορείτε να τα στείλετε στον Hans de Goede, στο : j.w.r.degoede@et.tudelft.nl