

Μεταγλώττιση των συσκευασιών πηγαίου κώδικα του SuSE Linux

---Πως να διαβάσετε το άρθρο
Οπού RPM εννοείτε ο μηχανισμός
Οπού rpm εννοείται το πρόγραμμα
Η έκδοση του SuSe που χρησιμοποιήσαμε είναι η 8.0
Για να μπορέσετε να διαβάσετε το άρθρο θα πρέπει να είστε ήδη εξοικειωμένος /η με τις βασικές αρχές του προγραμματισμού και να γνωρίζετε την χρήση των εντολών make diff patch και άλλων.
Θεωρούμε για ευκολία ότι όλες τις εργασίες, τις κάνετε ως υπερχρήστης (root)

--Γιατί να θελήσω να το κάνω αυτό?

Κάποιες φορές ίσως χρειαστεί να τροποποιήσετε ελαφρά κάποιο από τα προγράμματα που δίνονται με το SuSE Linux για να το φέρετε στα μέτρα σας.
Ένας τρόπος είναι να πάρετε των κώδικα πηγής (Source) από τον κατασκευαστή και να προχωρήσετε σε compile.
Ομως έτσι, ίσως φέρετε σε αστάθεια το σύστημα σας, ενώ δεν εκμεταλλεύεστε τις δυνατότητες χειρισμού με το σύστημα RPM
Είναι καλύτερα λοιπόν να προχωρήσετε χρησιμοποιώντας κώδικα τον οποίο πιθανόν ήδη έχει βελτιστοποιήσει (patch) η SuSE για να δουλεύει με το SuSE Linux.
Οι περισσότερες από τις εφαρμογές που έχει το SuSE Linux προσφέρονται με γενική δημόσια άδεια GNU (GPL)
Η άδεια αυτή απαιτεί από την SuSE να καθιστά τον κώδικα πηγής για τα προγράμματα που προσφέρει διαθέσιμο.
Στην SuSE έκαναν το προφανές και έβαλαν CD με τον κώδικα των προγραμμάτων στην συσκευασία.
Οι πηγές συσκευάζονται σαν συνήθη RPM για παράδειγμα `example.1.0.0.src.rpm`. Αυτό σημαίνει ότι η SuSE εκμεταλλεύεται τις δυνατότητες του rpm για να διευκολυνθεί ο χειρισμός του πηγαίου κώδικα των προγραμμάτων.
Το πρόγραμμα rpm δεν είναι απλά ένα βοήθημα που εγκαθιστά και αποεγκαθιστά εφαρμογές. Είναι και ένα πολύ χρήσιμο εργαλείο ανάπτυξης. Ένας από τους σημαντικούς στόχους κατά την ανάπτυξη του RPM ήταν να συμπεριλάβει το χειρισμό πηγαίου κώδικα.

Σε αυτό το άρθρο θα μάθετε πως να χρησιμοποιείτε το RPM για να εγκαταστήσετε από τον πηγαίο κώδικα και πώς να δημιουργείτε rpm's με τις αλλαγές που κάνατε ώστε να της δημοσιοποιήσετε στους κατόχους συστημάτων SuSE Linux.

--Εγκατάσταση Source RPM

Για της ανάγκης του άρθρου θα μετατρέψουμε το statserial ένα απλό πρόγραμμα που δείχνει την κατάσταση της σειριακής πόρτας.
Όλοι οι κώδικες βρίσκονται μέσα στον φάκελο `zq1 (zq2 ...)` συνήθως στο τελευταίο CD.
Πρέπει λοιπόν να βρείτε το `statserial-1.1-217.src.rpm` και να το εγκαταστήσετε πατώντας διπλό κλικ, ή χειροκίνητα με `rpm - i statserial-1.1-217.src.rpm`

Το rpm κάνει ένα αντίγραφο των αρχείων που περιλαμβάνονται στη συσκευασία στο σκληρό δίσκο.

Αντίθετα από τις συνηθισμένες συσκευασίες, τα αρχεία αυτά δεν μπορούν να αποεγκατασταθούν χρησιμοποιώντας το RPM γιατί δεν είναι κάτω από τον έλεγχο της βάσης δεδομένων.

Αυτό συμβαίνει γιατί ο κώδικας πηγής δεν είναι στην πραγματικότητα μέρος του συστήματος. Τα περιεχόμενα του rpm είναι ένα tar που περιέχει τους κώδικες όπως τους δίνει ο κατασκευαστής του προγράμματος, ένα η περισσότερα αρχεία με κατάληξη `dif` (δες `man dif`) που περιέχουν τις αλλαγές που έχει κάνει η SuSe και ένα αρχείο με κατάληξη `spec`, το αρχείο αυτό περιγράφει το πρόγραμμα, τις αλλαγές που έχει υποστεί από την SuSE και οδηγίες για το RPM.

Τα αρχεία αυτά γράφονται σε προκαθορισμένο σημείο στο σύστημα αρχείων.

Ο φάκελος `/usr/src/packages` περιέχει τους υποφακέλους

SOURCES όπου γράφονται τα tar και dif αρχεία
SPECS για τα αρχεία .spec
(περιέχονται και άλλοι φάκελοι, αλλά θα τους εξηγήσουμε παρακάτω.)

Αν δεν θέλουμε να αλλάξουμε τίποτα στον πηγαίο κώδικα του προγράμματος αλλά απλά να μεταγλωττίσουμε γιατί π.χ. αλλάξαμε την έκδοση του gcc δίνουμε την εντολή `rpm -ba statserial.spec` .
(σαν root και αφού βρισκόμαστε `/usr/src/packages/SPECS`) και τότε το RPM θα κάνει αυτόματα όλη την δουλειά δηλαδή θα περάσει τα patches στον κώδικα ,θα κάνει compile και θα εγκαταστήσει τα εκτελέσιμα του statserial.
Παρατηρείστε ότι στον υποφάκελο BUILD έχει δημιουργηθεί ο φάκελος statserial-1.1 που περιέχει τα αρχεία κώδικα του προγράμματος και το εκτελέσιμο statserial .
Στον υποφάκελο (Ανάλογα με την αρχιτεκτονική του υπολογιστή και τις οδηγίες του makefile του προγράμματος)
`/RPMS/i386`
θα βρείτε το rpm που μπορείτε πλέον να εγκαταστήσετε κανονικά
Στον υποφάκελο
`/SRPMS`
θα βρείτε το αρχείο `statserial-1.1-217.src.rpm`
εδώ μπορεί ίσως να ξαφνιαστείτε και να μην καταλάβετε τι ακριβώς έγινε.
Μήπως το rpm έκανε αντιγραφή του αρχείου ?
Και όμως στην πραγματικότητα αυτό που έγινε ήταν να αναδημιουργηθεί το αρχείο από τα συστατικά του όπως τα εξάγαμε.
Αν αυτό σας φαίνεται ανούσιο, θα καταλάβετε την χρησιμότητα του, αν σκεφτείτε ότι μπορεί στα αρχεία να είχαμε κάνει μετατροπές ;-)

Τα στάδια που εκτέλεσε η εντολή `rpm -ba statserial.spec` ήταν αναλυτικά τα παρακάτω

1 Προετοιμασία

Το RPM άνοιξε το αρχείο tar έγραψε τα περιεχόμενα στο BUILD/statserial-1.1 και έκανε την αναβάθμιση(patch) σύμφωνα με της οδηγίες του statserial-1.1.dif
Οι οδηγίες για να γίνουν τα παραπάνω βρίσκονται στο αρχείο statserial.spec στο label %prep

2 Δημιουργία εκτελέσιμου

Σύμφωνα πάντα με της οδηγίες του statserial.spec στο label %build εκτελείτε το make και δημιουργείτε έτσι το εκτελέσιμο

3 Εγκατάσταση

Σύμφωνα με το label %install τοποθετούνται τα αρχεία που δημιουργήθηκαν στις θέσεις τους.
Προσοχή αυτό δεν σημαίνει ότι η εγκατάσταση αυτή προστέθηκε στην βάση του RPM άρα δεν μπορείτε να απεγκαταστήσετε το πακέτο.

4 Πακετάρισμα

Δημιουργούνται τα rpm's για τα εκτελέσιμα και για τον κώδικα
Χρήσιμο θα ήταν να τρέξουμε τώρα το `rpm -i statserial-1.1-217.i386.rpm`
για να εγκαταστήσουμε το πρόγραμμα και να μπορούμε πλέον να το χειριζόμαστε με το RPM
(Αν δεν κάνουμε το παραπάνω δεν θα μπορούμε να δούμε το πρόγραμμα με το kpackage ή άλλο εργαλείο)

Για ευκολία δίνονται εδώ οι παράμετροι για την εντολή build (-b) του rpm
δώστε `man rpm` για να δείτε αναλυτικά όλες της οδηγίες
stages
`-bp` Executes the "%prep" stage from the spec file.
Normally this involves unpacking the sources and applying any patches.

-bl Do a "list check". The "%files" section from the spec file is macro expanded, and checks are made to verify that each file exists.

-bc Do the "%build" stage from the spec file (after doing the prep stage).
This generally involves the equivalent of a "make".

-bi Do the "%install" stage from the spec file (after doing the prep and build stages).
This generally involves the equivalent of a "make install".

-bb Build a binary package (after doing the prep, build, and install stages).

-bs Build just the source package (after doing the prep, build, and install stages).

-ba Build binary and source packages (after doing the prep, build, and install stages).

options

--short-circuit Skip straight to specified stage (ie, skip all stages leading up to the specified stage). Only valid with -bc and -bi.

--timecheck Set the "timecheck" age (0 to disable). This value can also be configured by defining the macro "_timecheck".

The timecheck value expresses, in seconds, the maximum age of a file being packaged.

Warnings will be printed for all files beyond the timecheck age.

--clean Remove the build tree after the packages are made.

--rmsource Remove the sources and spec file after the build (may also be used standalone, eg. "rpm --rmsource foo.spec").

--test Do not execute any build stages. Useful for testing out spec files.

--sign Embed a PGP signature in the package.

This signature can be used to verify the integrity and the origin of the package.

See the section on PGP SIGNATURES for configuration details.

--buildroot <dir> When building the package, override the BuildRoot tag with directory <dir>.

--target <platform> When building the package, interpret <platform> as arch-vendor-os and

set the macros _target, _target_arch and _target_os accordingly.

--buildarch <arch> When building the package, set the architecture to <arch>.

This option has been obsoleted by --target in RPM 3.0.

--buildos <os> When building the package, set the architecture to <os>.

This option has been obsoleted by --target in RPM 3.0.

Όμως εμείς θέλουμε να κάνουμε μια μικρή παρέμβαση στον κώδικα οπότε δίνουμε την εντολή `rpm -bp statserial.spec` με αυτόν το τρόπο γίνεται μοναχά η εξαγωγή των αρχείων από το tar και η τοποθέτηση των βελτιώσεων (patches) από τα dif αρχεία

Μετά από αυτό στο υποφάκελο BUILD έχει δημιουργηθεί ο φάκελος `statserial-1.1` πάμε και κάνουμε `rename` σε

`statserial-1.1.orig` ξανατρέχουμε την `rpm -bp statserial.spec` ώστε να ξαναδημιουργηθεί ο φάκελος

Τώρα μπορούμε να κάνουμε την αλλαγή μας που είναι η απλή πρόσθεση ενός ακόμη "διακόπτη" της γραμμής εντολών

βρίσκουμε λοιπόν στο `BUILD/statserial-1.1/statserial.c`

```

στην γραμμή 64
const char      *flags = "ndx";
και αλλάζουμε σε
const char      *flags = "ndxu";
στην γραμμή 78
case '?':
    usage();
    break;
}
και αλλάζουμε σε
case 'u':
    usage();
    break;
case '?':
    printf("statserial: unknown option try statserial -u for
usage\n");
    exit(1);
    break;
}

```

Το σύστημα RPM απαιτεί να βρίσκονται ξεχωριστά τα αρχεία diff από τα αρχικά αρχεία του προγράμματος για αυτό και χρειάστηκε να έχουμε δύο ξεχωριστά δένδρα αρχείων. Πάμε λοιπόν στον υποφάκελο BUILD και τρέχουμε

```

diff -ruN statserial-1.1.orig statserial-1.1
>../SOURCES/mystatserial.dif

```

αυτό δημιουργεί το αρχείο διαφορών σε σχέση με το ήδη "βελτιωμένο" κώδικα από την SuSE θα πρέπει λοιπόν τώρα να πειράξουμε το αρχείο spec για να δώσουμε την δυνατότητα να μπαίνει και η δικιά μας βελτίωση κατά την εγκατάσταση του προγράμματος

Ας μιλήσουμε όμως λίγο ακόμα για το spec.

Όπως είδαμε παραπάνω το statserial.spec περιέχει οδηγίες για το RPM χωρίζεται σε περιοχές με διαφορετική σημασία

1 Preamble

Εδώ δίνονται σύντομες πληροφορίες για το πρόγραμμα και για τον δημιουργό του. για το πια αρχεία βελτιστοποίησης (dif) θα χρησιμοποιηθούν και άλλα..

2 Prep

Εδώ είναι η οδηγίες προς το RPM για το πώς να εξαγεί το πρόγραμμα και ποιες βελτιώσεις να του κάνει, βασικά δεν είναι τίποτα άλλο από οδηγίες script που μπορούν να τρέξουν για παράδειγμα το patch να κάνουν configure και άλλα σχετικά με την προετοιμασία του προγράμματος πριν το make.

3 Build

και εδώ θα βρούμε ένα script το οποίο συνήθως δεν κάνει τίποτα άλλο από το να εκτελεί την make

4 Install

και εδώ επίσης οδηγίες για το που θα τοποθετηθούν τα αρχεία κλήση της make install και άλλα σχετικά με την εγκατάσταση

5 Scripts

σε αυτή την περιοχή βρίσκονται τα διάφορα scripts για τη εγκατάσταση απεγκατάσταση και των έλεγχο

6 Clean

εδώ μπαίνουν οδηγίες για τον καθαρισμό των αρχείων που δεν χρειάζονται μετά την εγκατάσταση όπως ο πηγαίος κώδικας και άλλα

7 Files

εδώ είναι η λίστα με τα αρχεία που περιέχει το πακέτο και το είδος τους

8 Changelog

Εδώ μπαίνουν πληροφορίες για της αλλαγές που έχει υποστεί το πακέτο

Για περισσότερες πληροφορίες δείτε το σχετικό HOWTO <http://www.rpm.org/RPM-HOWTO/>

Στην περίπτωση μας ενδιαφέρον έχει η περιοχή `prep`. Εκεί πρέπει να δώσουμε οδηγίες για να χρησιμοποιηθεί το `mystatserial.dif` για την Βελτιστοποίηση του κώδικα. Ανοίγουμε λοιπόν με έναν editor το `statserial.spec`

Στην αρχή (Preamble) βρίσκουμε μέσα στις εγγραφές και την
`Patch: statserial-%{version}.dif`
προσθέτουμε ακριβώς από κάτω
`Patch1: mystatserial.dif`

στην περιοχή κάτω από το `label %prep`
βρίσκουμε το `%patch`
προσθέτουμε ακριβώς από κάτω
`%patch1 -p1`
διαγράφουμε τώρα τα δύο δέντρα με τους κώδικες
`BUILD/statserial-1.1`
`BUILD/statserial-1.1.orig`

Πηγαίνουμε στον υποφάκελο SPECS και δίνουμε `rpm -ba statserial.spec` αν όλα έχουν πάει καλά δημιουργούνται τα αρχεία `rpm` για τα εκτελέσιμα και για τα αρχεία με τις αλλαγές που έχουμε κάνει.

Τώρα πλέον μπορείτε να εγκαταστήσετε το `statserial` και να μοιράσετε τα `rpm's` σε όποιον επιθυμείτε.

Χρήσιμες διευθύνσεις

<http://www.rpm.org>

http://sdb.suse.de/sdb/en/html/pakete_manuell.html

http://sdb.suse.de/sdb/en/html/ke_rpm.html

http://sdb.suse.de/sdb/en/html/ke_source-rpm.html

Παναγιώτης Σκαρβέλης etsoft@daem.gr
Πάυλος Καλλίας kallias@daem.gr