

Gentoo Linux

Πάρτε τον πλήρη έλεγχο!



Φανταστείτε μία διανομή Linux που μπορεί να παραμετροποιηθεί για κάθε ανάγκη, προσφέροντας ευελιξία και απόδοση... Αυτό ακριβώς είναι το Gentoo!

ΤΩΝ ΓΙΩΡΓΟΥ ΚΑΡΓΙΩΤΑΚΗ, ΝΙΚΟΥ ΡΟΥΣΣΟΥ

Ο Γιώργος είναι wannabe Μαθηματικός, εργάζεται στο R&D τμήμα μίας εταιρείας σχετικά με VoIP και ως freelance system administrator. Είναι μέλος της ομάδας ILUG και διατηρεί blog στη διεύθυνση void.gr/karglg/.

Ο Νίκος είναι Μηχανικός Η/Υ. Εργάζεται στο δημιουργικό γραφείο cyberarch.gr και ως εκπαιδευτικός Πληροφορικής. Είναι πρόεδρος του HELUG και το blog του βρίσκεται στη διεύθυνση autoverse.net.

Το Gentoo έχει ως κύριο χαρακτηριστικό τη μέγιστη δυνατή ευελιξία/προσαρμοστικότητα στις ανάγκες του χρήστη. Αυτός είναι ο λόγος που έχει χαρακτηριστεί ως μετα-διανομή, ακριβώς επειδή χτίζεις τη δική σου διανομή από την αρχή. Προσφέρει σχεδόν απόλυτο έλεγχο στις παραμέτρους των πακέτων και στον τρόπο με τον οποίο θα γίνει η εγκατάστασή τους, γι' αυτό, σε αντίθεση με τις υπόλοιπες διανομές, τα πακέτα προσφέρονται με τη μορφή του πηγαίου κώδικα και μεταγλωττίζονται στη συνέχεια ανάλογα με τις επιλογές του χρήστη.

Η κεντρική φιλοσοφία του είναι να παρέχει την ελευθερία στο χρήστη να διαμορφώσει το λειτουργικό σύστημά του, ώστε να δουλεύει όπως θέλει ο ίδιος και όχι να δουλεύει ο χρήστης όπως θέλει το λειτουργικό του, ενώ υποστηρίζει σχεδόν όλες τις αρχιτεκτονικές: x86, amd64, ppc, ppc64, alpha, arm, hppa, ia64, m68k, mips, s/390, sparc, x86-freebsd.

Το Gentoo δημιουργήθηκε από τον Daniel Robbins, το 1999, ως μία προσπάθεια να φτιαχτεί μία διανομή που θα χτίζεται από τον πηγαίο κώδικα και θα είναι πλήρως προσαρμοσμένη στο hardware στο οποίο θα τρέξει. Είναι χαρακτηριστικό πως στα πρώτα βήματα της διανομής προέκυψαν αρκετά προβλήματα με το μεταγλωττιστή GCC, που οδήγησαν τους Gentoo developers να του κάνουν αρκετές τροποποιήσεις, οι οποίες πλέον έχουν ενσωματωθεί στον compiler.

Η διανομή χαρακτηρίζεται από τα rolling updates, το οποίο πρακτικά σημαίνει πως μετά την εγκατάσταση μπο-

ρούμε συνεχώς να αναβαθμίζουμε το σύστημά μας με τα τελευταία διαθέσιμα πακέτα. Στο Gentoo δεν θα σας απασχολήσει το τι πρέπει να κάνετε όταν βγει ένα νέο release, το σύστημά σας απλώς θα περάσει σε αυτό. Σκοπός των νέων releases είναι να υπάρχει καλύτερη υποστήριξη hardware και αναβαθμισμένα πακέτα στο install CD, καθώς και αλλαγές σε κάποια κεντρικά configuration files που αποκαλούνται profiles. Το κεντρικό αποθετήριο (Portage Tree στη γλώσσα του Gentoo) παραμένει το ίδιο, ανεξάρτητα από το release που κάναμε την αρχική εγκατάσταση.

Γενικώς, όλη η διαδικασία της εγκατάστασης και αναβάθμισης των πακέτων μάς δίνει τη δυνατότητα να γνωρίζουμε ακριβώς ποια πακέτα έχουμε βάλει στο σύστημά μας και τι κάνει το καθένα, ενώ ακόμη κι αν μία εφαρμογή δεν υπάρχει στο Portage Tree, είναι αρκετά εύκολο να φτιάξουμε το δικό μας πακέτο και να την εγκαταστήσουμε.

Δομή Λειτουργίας

Οι καινοτομίες του Gentoo συνεχίζονται ακόμη και στον τρόπο με τον οποίο λειτουργεί ως project. Κατ' αρχάς, υπάρχει το Gentoo Foundation, το οποίο είναι ένας μη κερδοσκοπικός οργανισμός, υπεύθυνος για οτιδήποτε εκτός από τεχνικά θέματα: copyright λογοτύπου και ονόματος, οικονομική διαχείριση κ.λπ. Υπεύθυνο για τα τεχνικά θέματα είναι το Gentoo Council (ναι θυμίζει λίγο Star Wars :)), του οποίου τα μέλη εκλέγονται από τους developers και είναι ισότιμα μεταξύ τους. Αυτή είναι η μεγαλύτερη διοικητική διαφορά του Gentoo σε σχέση με άλλες διανομές, το ότι δεν υπάρχει ένας ηγέτης που θα καθορίσει το "δρόμο" που θα ακολουθήσει η διανομή, αλλά ένα αιρετό συμβούλιο.

ΤΙ ΘΑ ΧΡΕΙΑΣΤΟΥΜΕ

- Το CD εγκατάστασης
- Μία σύνδεση DSL
- Το Gentoo Handbook
- Μία κούπα καφέ

Τεκμηρίωση

Ενα από τα στοιχεία που έχει κάνει το Gentoo πιο γνωστό, είναι η πλούσια τεκμηρίωσή του. Πολλές φορές θα έχετε δει αναφορές σε how-to που προέρχονται από το Gentoo-Wiki, ακόμη και σε forums ή λίστες άλλων διανομών. Το wiki περιλαμβάνει οδηγούς για πάρα πολλά θέματα (πάνω από 8.500 άρθρα), από το πώς να στήσετε έναν home server, μέχρι πώς να φτιάξετε δική σας διανομή ή και για πιο απλά πράγματα, όπως γρήγορες συμβουλές για διάφορους window managers.

Το wiki, όμως, παρά το πλούσιο περιεχόμενο του, εξακολουθεί να είναι ανεπίσημο. Το Gentoo επιπλέον έχει αρκετές σελίδες τεκμηρίωσης, στις οποίες μπορείτε να περιηγηθείτε εύκολα από την κεντρική σελίδα του.

Παραμετροποίηση

Με ποιο τρόπο, όμως, παραμετροποιούμε τα πακέτα που εγκαθιστούμε; Οι παρακάτω επιλογές που περιγράφουμε, αφορούν στο αρχείο `/etc/make.conf`.

Όσον αφορά στη διαδικασία της μεταγλώττισης, έχουμε τη δυνατότητα επιλέξουμε τις επιλογές (flags) με τις οποίες θα καλείται ο GCC. Οι επιλογές αυτές περιέχονται στις μεταβλητές CFLAGS (για C) και CXXFLAGS (για C++). Για παράδειγμα, αν είμαστε σε ένα σύστημα με Quad Core επεξεργαστή σε x86 (32bit) μηχανήμα, τότε μία τυπική τιμή της CFLAGS θα ήταν:

```
-march=prescott -O2 -pipe -fomit-frame-pointer
```

όπου με το `march` ορίζουμε την αρχιτεκτονική του επεξεργαστή μας, ώστε η μεταγλώττιση να είναι βελτιστοποιημένη γι' αυτόν, και με την επιλογή `O2` επιλέγουμε μία σειρά από επιπλέον βελτιστοποιήσεις που μας παρέχει ο GCC. Για τον ίδιο επεξεργαστή για 64bit μηχανήμα, θα είχαμε:

```
-march=nocona -O2 -pipe
```

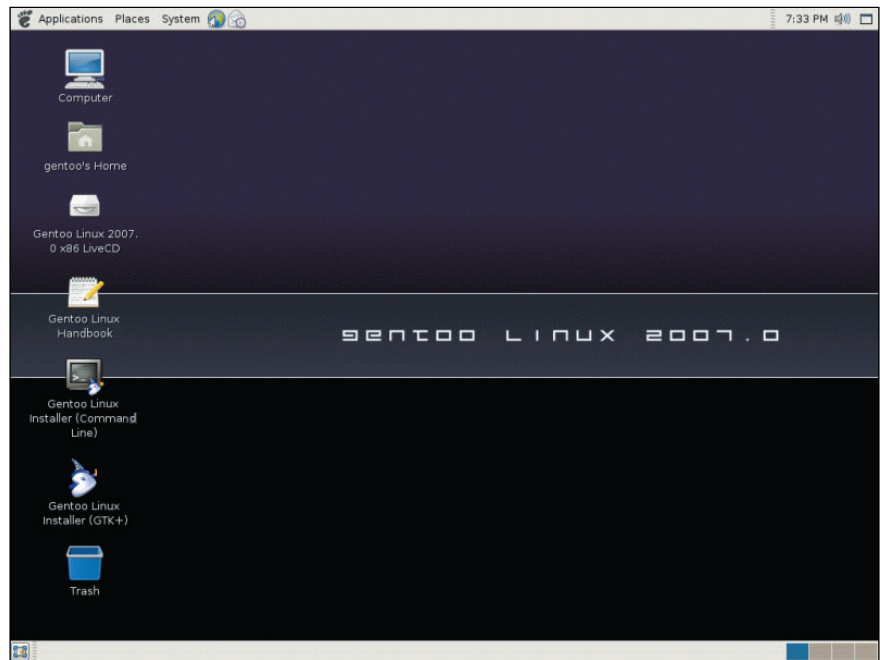
Ομοίως, πρέπει να ορίσουμε και τη μεταβλητή `CHOST` για 32 και 64bit αντίστοιχα:

```
i686-pc-linux-gnu
```

```
x86_64-pc-linux-gnu
```

Περαιτέρω για τα CFLAGS και `CHOST` (αναλυτικά για κάθε τύπο επεξεργαστή) μπορείτε να δείτε στο wiki: http://gentoo-wiki.com/Safe_Cflags.

Το πιο σημαντικό, όμως, κομμάτι παραμετροποίησης είναι τα USE flags, που μας δίνουν τη δυνατότητα εγκατάστασης πακέτων μόνο με τις επιλογές που χρειαζόμαστε, αφήνοντας έξω τα περιττά πράγματα. Τα USE flags δεν είναι κάτι "εξωτικό", αλλά ένας τρόπος να περνάμε τις επιλογές μας στο `.configure` του πακέτου που κάνει το `emerge` πριν από το `make`. Οποιος έχει κάνει `compile` (μεταγλώττιση) ένα πρόγραμμα, θα καταλάβει αμέσως τη χρησιμότητά τους. Για παράδειγμα, αν δεν έχουμε εκτυπωτή στο σπίτι μας, δεν



υπάρχει λόγος να χτίζουμε τα πακέτα μας με υποστήριξη για το CUPS. Οπότε, θέτοντας ως USE flag, `"-cups"`, έχουμε καταφέρει να απαλλαγούμε από περιττά για εμάς πράγματα. Παραδείγματα χρήσης USE flags θα δούμε παρακάτω.

Η επιφάνεια εργασίας του Live CD.

Package Manager

Το σύστημα διαχείρισης πακέτων του Gentoo είναι το Portage. Αποτελείται από δύο βασικά μέρη: το Portage Tree με τα ebuilds και το εργαλείο `emerge`.

Ενα ebuild είναι ένα αρχείο το οποίο περιγράφει όλα τα απαραίτητα βήματα που απαιτούνται για να εγκατασταθεί ένα πακέτο: από το κατέβασμα του πηγαίου κώδικα, το `compilation`, μέχρι και την τελική εγκατάσταση στο σύστημα μας. Επίσης περιγράφει τις εξαρτήσεις (dependencies) που χρειάζεται το συγκεκριμένο πακέτο και τα USE flags του. Το Portage Tree του Gentoo μαζί με όλα τα ebuilds βρίσκεται στον κατάλογο `/usr/portage`. Ουσιαστικά, τα ebuilds είναι το αντίστοιχο των `dpkg` και `rpm` και το `emerge` το αντίστοιχο των `apt` και `yum` για όσους έχουν καλύτερη επαφή με τα Debian και Redhat συστήματα αντιστοίχως.

Το `emerge` είναι το εργαλείο το οποίο χρησιμοποιεί ο χρήστης για να εγκαταστήσει, απεγκαταστήσει και να αναβαθμίσει πακέτα στο σύστημά του. Αυτό λειτουργεί διαβάζοντας τα ebuilds και εκτελώντας τις λειτουργίες που αυτά περιγράφουν. Ενώ, μέχρι πριν από λίγα χρόνια, το `emerge` ήταν το

ΔΙΑΒΑΣΤΕ ΠΡΙΝ ΑΠΟ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ...

Το Gentoo είναι μία διανομή με αρκετά διαφορετική φιλοσοφία από αυτή των περισσότερων διανομών. Γι' αυτό είναι απαραίτητη η ανάγνωση του Gentoo Handbook πριν ξεκινήσουμε τη διαδικασία της εγκατάστασης. Γενικώς, πρέπει να συνηθίσουμε την ιδέα πως το Gentoo είναι μία διανομή που απαιτεί αρκετό διάβασμα, ειδικά στην αρχή. Παρόλο που είναι πλέον διαθέσιμο ένα γραφικό περιβάλλον εγκατάστασης (το γνωστό ως Gentoo Installer), θα συνιστούσαμε να ακολουθήσετε τη διαδικασία της εγκατάστασης που περιγράφεται στο handbook, γιατί έτσι θα μάθετε αρκετά πράγματα και κυρίως θα μπειτε πιο εύκολα στη φιλοσοφία του Gentoo. Δεν θα περιγράψουμε αναλυτικά την εγκατάσταση (άλλωστε, θα θέλαμε το μισό τεύχος για να χωρέσει!), αλλά θα αναφέρουμε επιγραμματικά τα βασικά βήματα που απαιτούνται.

- Ξεκινάμε τον υπολογιστή μας με το CD εγκατάστασης. Αφού τελεειώσει η εκκίνηση κάνουμε τις απαραίτητες ρυθμίσεις για να έχουμε Internet και, σε περίπτωση που δεν έχουμε άλλο μηχανήμα κοντά μας με σύνδεση, ανοίγουμε το links και πηγαίνουμε στη σελίδα του handbook. Θα μας βοηθήσει αρκετά να το έχουμε ανοικτό κατά τη διάρκεια της εγκατάστασης.
- Με τη βοήθεια του `cdisk` φτιάχνουμε τα partitions μας, τα μορφοποιούμε στο απαιτημένο μας filesystem και τα προσαρτούμε.
- Ακολουθώντας τις οδηγίες του handbook κατεβάζουμε το `stage3` και το portage μέσα στο root partition. Το `stage3` περιέχει όλα τα βασικά πακέτα που χρειαζόμαστε, για να μπορέσουμε να χτίσουμε το σύστημά μας (π.χ., `glibc`, `gcc`).
- Πριν προχωρήσουμε παρακάτω, ρυθμίζουμε το

`/etc/make.conf`, όπως δείχνουμε στο άρθρο, και, στη συνέχεια, με την εντολή `chroot`, "μπαίνουμε" στο root partition. Ουσιαστικά, ό,τι κάνουμε από δω και πέρα, γίνεται στο partition που χτίζουμε τη διανομή μας.

■ Συγχρονίζουμε το Portage Tree (`emerge --sync`) και εγκαθιστούμε το πρώτο μας πακέτο που δεν είναι άλλο από τον πυρήνα. Όπως θα περιμένατε, τον πυρήνα πρέπει να τον ρυθμίσουμε μόνοι μας πριν τον μεταγλωττίσουμε, ενώ υπάρχει και το εργαλείο `genkernel`, που μπορεί να αυτοματοποιήσει λίγο τη διαδικασία και να παράγει όλα τα πιθανά modules, όπως κάνουν και οι υπόλοιπες διανομές.

■ Στη συνέχεια, αφού φτιάξουμε τα `/etc/fstab` και `/etc/mtab`, εγκαθιστούμε το grub (boot loader).

■ Τέλος, ορίζουμε ένα password για τον root (με την εντολή `passwd`) και κάνουμε `reboot` το σύστημά μας.

ΞΕΖΟΥΜΙΣΜΑ

Πολλοί κατηγορούν τους Gentoo Users για "ricing" [01]. Προσπαθούν, δηλαδή, να ξεζουμίσουν το σύστημά τους για να το κάνουν όσο πιο γρήγορο γίνεται. Δεν είναι τυχαίο πως το όνομα Gentoo προέρχεται από το ομώνυμο είδος πικουίνου, που είναι γνωστό ως το ταχύτερο είδος στην κοθύμβηση. :) Είναι αρκετά συνηθισμένο ο κατάλληλος στον οποίο βρίσκεται το portage (/usr/portage), να είναι σε ξεχωριστό partition [1]. Κατά το compilation των πακέτων αποθηκεύονται προσωρινά δεδομένα στο /var/tmp/portage. Επειδή το I/O στο δίσκο είναι καμιά φορά αργό, μπορεί κάποιος που έχει μπόλικη RAM να δώσει ένα κομμάτι της για χρήση ως δίσκου μέσω ramdisk [2]. Για τους... πολύ ricers, υπάρχει τεχνική, ώστε να χρησιμοποιεί κάποιος ακόμη και τη RAM της VGA του [3]. Αν κάποιος διαθέτει

παραπάνω από έναν υπολογιστές σε γρήγορο δίκτυο, μπορεί να εκμεταλλευτεί το distcc, ώστε να γίνεται το compile ταυτόχρονα σε όλα τα μηχανήματά του [4], ενώ χρησιμοποιώντας την cache, μπορεί να κάνει ένα caching των objects που παράγονται κατά το compile, ώστε τα recompiles να γίνονται αρκετά πιο γρήγορα [5]. Συνήθης τακτική που χρησιμοποιούν και άλλα distros για να κερδίζουν σε ταχύτητα ανοίγματος των εκτελέσιμων τους, είναι το prelinking [6].

[01] <http://ub0.cc/c/2G>
[1] <http://ub0.cc/0/3d>
[2] <http://ub0.cc/1/0b>
[3] <http://ub0.cc/1/09>
[4] <http://ub0.cc/b/t>
[5] <http://ub0.cc/6/1b>
[6] <http://ub0.cc/1/08>

μοναδικό εργαλείο γι' αυτή τη δουλειά που υπήρχε διαθέσιμο στο Gentoo, πλέον υπάρχει και ακόμη ένα, το paludis, γραμμένο από έναν πρώην developer του Gentoo. Ενώ το emerge είναι γραμμένο σε Python, το paludis είναι γραμμένο σε C++ και είναι αρκετά πιο γρήγορο. Δεν χρησιμοποιείται, όμως, παρά από ελάχιστη μερίδα χρηστών, μια και δεν παρέχεται επίσημη υποστήριξη γι' αυτό. Η ύπαρξη ανταγωνιστικού προς το emerge εργαλείου μόνο θετικά συμβάλλει στην πορεία του Gentoo. Μάλιστα καταρτίστηκε ένα πρότυπο που πρέπει πλέον να σέβονται και να τηρούν όλοι οι package managers, υπάρχοντες ή και μελλοντικοί, του Gentoo, με στόχο την αυξημένη συμβατότητα των συστημάτων. Επίσης, λόγω του ανταγωνισμού, κάθε package manager προσθέτει διαρκώς νέα features. Εδώ πάντως θα περιοριστούμε στο emerge...

Οι δυνατότητες επέμβασης που έχουμε στις επιλογές του emerge μέσω του αρχείου /etc/make.conf, είναι πάρα πολλές. Η αρχική ρύθμιση του emerge είναι να κατεβάζει το πηγαίο κώδικα ενός πακέτου ακριβώς πριν το κάνει compile. Προσθέτοντας, όμως, τη γραμμή FEATURES="parallel-fetch" στο /etc/make.conf, τότε το emerge μπορεί να κατεβάζει τα επόμενα πακέτα κατά τη διάρκεια των μεταγλωττίσεων.

Σε περίπτωση που έχουμε καινούργιο πολυπύρηνιο επεξεργαστή, η γραμμή MAKEOPTS="-j3" θα μας φανεί σωτήρια. Ουσιαστικά, ελέγχει πόσες ταυτόχρονες μεταγλωττίσεις θα

"τρέχουν" κατά το emerge ενός πακέτου. Συνιστάται να δίνονται τιμές όσες οι αριθμοί των cores ενός μηχανήματος +1.

Με το PORTAGE_NICENESS="5" ελέγχουμε το nice level με το οποίο θα τρέχει η emerge και θα γίνονται οι μεταγλωττίσεις. Όσο μεγαλύτερη είναι αυτή η τιμή (με μέγιστο το 19) τόσο λιγότερους πόρους θα ζητάει το emerge από το σύστημά μας. Με το GENTOO_MIRRORS= ελέγχουμε από ποια mirrors θα κατεβάζει το emerge τα πακέτα μας και με τη SYNC= ελέγχουμε το rsync mirror. Οι δύο αυτές επιλογές μπορούν να ρυθμιστούν από ένα μικρό πρόγραμμα που λέγεται mirrorselect και μας βοηθά να επιλέξουμε τα καταλληλότερα mirrors, αναλόγως με την τοποθεσία όπου βρισκόμαστε. Περισσότερες επιλογές μπορούμε να διαβάσουμε στο αρχείο /etc/make.conf.example.

Παραδείγματα χρήσης

Για να καταλάβει κάποιος λίγο καλύτερα τα USE flags, ας δούμε ένα παράδειγμα για το πακέτο ffmpeg:

```
media-video/ffmpeg-0.4.9_p20070616-r2 USE="X
a52 aac amr encode ipv6 mmx ogg theora threads
truetype vorbis xvid zlib (-altivec) -debug -doc -
ieee1394 -imlib -network -oss -sdl -test -v4l -x264"
```

Εδώ έχουμε τη δυνατότητα να επιλέξουμε με ποιες επιλογές θα γίνει compile, π.χ., αν θέλουμε να έχουμε υποστήριξη για aac, theora, xvid ή αν θέλουμε να μας εγκαταστήσει τεκμηρίωση (doc) κ.λπ. Τα USE flags που έχουν ένα μείον (-) μπροστά, είναι αυτά που έχουμε επιλέξει να μη βάλουμε. Το αρχείο στο οποίο καταγράφουμε ανά πακέτο ποια USE flags θέλουμε, είναι το /etc/portage/package.use. Ας πάρουμε για παράδειγμα το abiword:

```
echo "app-office/abiword spell xml -debug -gnome"
>> /etc/portage/package.use
```

Με τον παραπάνω τρόπο, δηλώνουμε πως θέλουμε να μεταγλωττιστεί το abiword με υποστήριξη για xml και με orthoγράφο.

Μπορούμε επίσης να έχουμε USE flags για όλο το σύστημα (global use flags), τα οποία τα βάζουμε στο /etc/make.conf. Αν, για παράδειγμα, χρησιμοποιούμε KDE περιβάλλον, μπορούμε να προσθέσουμε στο USE= του /etc/make.conf το "-gnome -gtk kde qt", ώστε όσα πακέτα έχουν υποστήριξη και για Gnome και για KDE, να γίνουν compile μόνο για KDE.

Χρήση emerge

Ας δούμε μερικά απλά παραδείγματα χρήσης του emerge. Για να το χρησιμοποιήσουμε, πρέπει κατ' αρχάς να έχουμε δικαιώματα root στο σύστημα.

Το Gentoo συγχρονίζει το τοπικό Portage Tree που υπάρχει στον υπολογιστή μας με τα κεντρικά repositories του Gentoo μέσω της εντολής:

```
# emerge --sync
```

Το gentoo-netiquette (κάτι σαν κανόνες συμπεριφοράς του Gentoo) συνιστά τη χρήση της παραπάνω εντολής 2 φορές τη μέρα το πολύ. Αν χρειάζεται για κάποιον λόγο να κάνετε sync πιο συχνά, καλό θα ήταν να επιλέξετε ένα κοντινό rsync mirror μέσω της εντολής

```
# mirrorselect -i -r -o >> /etc/make.conf
```

Για να δούμε τα USE flags ενός πακέτου, καθώς και ποιες είναι οι εξαρτήσεις του, χρησιμοποιούμε τις επιλογές -p (pretend) και -v (verbose):

```
# emerge -pv mplayer
```

Στη συνέχεια, αφού είμαστε σίγουροι πως θέλουμε να εγκαταστήσουμε το συγκεκριμένο πακέτο, δίνουμε:

```
# emerge -v mplayer
```

Σε περίπτωση που το έχουμε ήδη εγκατεστημένο και θέλουμε να το αναβαθμίσουμε μαζί με τις άμεσες εξαρτήσεις



του, χρησιμοποιούμε την επιλογή -u (update):

emerge -uv mplayer

Εάν θέλουμε να αναβαθμίσουμε ακόμη και τις εξαρτήσεις των εξαρτήσεων ενός πακέτου, τότε χρειάζεται να δώσουμε και την επιλογή -D (deep):

emerge -uDv mplayer

Για να απεγκαταστήσουμε ένα πακέτο, χρησιμοποιούμε την επιλογή -C:

emerge -Cv mplayer

Για να αναζητήσουμε ένα πακέτο, αν δεν ξέρουμε ακριβώς το όνομά του, χρησιμοποιούμε τις επιλογές -s (search) και -S (searchdesc). Το πρώτο κάνει αναζήτηση με βάση το όνομα του πακέτου, ενώ το δεύτερο με βάση την περιγραφή. Επειδή, όμως, η λειτουργία αναζήτησης του emerge είναι λίγο αργή, υπάρχει το εργαλείο eix που μπορείτε να εγκαταστήσετε (emerge eix), ώστε να βελτιώσετε κατά πολύ την ταχύτητα των αναζητήσεών σας. Οι βασικές επιλογές αναζήτησής του είναι ίδιες με αυτές του emerge. Για παράδειγμα, για να ψάξετε για όλα τα πακέτα με περιγραφή η οποία περιέχει τη λέξη video, δώστε:

eix -S video

Κάθε πακέτο που γίνεται emerge, γράφεται σε μία λίστα που ονομάζεται world file και βρίσκεται στο /var/lib/portage/world. Αν, λοιπόν, κάνετε emerge το πακέτο AAA που έχει εξάρτηση το πακέτο BBB μέσω της emerge AAA, στο world file θα τοποθετηθεί μόνο το AAA. Όσο το world file μεγαλώνει τόσο περισσότερο θα αρχίσει να αργεί το emerge, ψάχνοντας για τις εξαρτήσεις κάθε πακέτου που βρίσκεται εκεί μέσα. Είναι συνηθισμένο "λάθος" αρκετών χρηστών, όταν θέλουν αργότερα να κάνουν upgrade το πακέτο BBB, να δίνουν emerge -u BBB. Αυτό θα έχει ως αποτέλεσμα να εγγραφεί και το BBB μέσα στο world file. Το σωστό είναι να δοθεί και η επιλογή --oneshot, ώστε να μη γίνει η καταγραφή στο world file.

emerge --oneshot -v BBB

Για να κάνετε upgrade όλα τα πακέτα που έχουν περαστεί στο world file, καθώς και τις εξαρτήσεις τους, μπορείτε να δώσετε:

emerge -uavt world

Η επιλογή -a (ask) θα σας εμφανίσει ότι ακριβώς και το -p (pretend) ρωτώντας σας, όμως, αν θέλετε να συνεχίσετε με το emerge των πακέτων, ενώ η επιλογή -t (tree) θα παρουσιάσει την παραπάνω λίστα με μία δεντροειδή μορφή με τις εξαρτήσεις των πακέτων.

Για να κάνετε ένα πλήρες upgrade του συστήματός σας, δηλαδή, όλων των πακέτων του world file, των εξαρτήσεών τους, καθώς και των εξαρτήσεων των εξαρτήσεων, χρησιμοποιείτε την εντολή:

emerge -uDavt world

Ενα από τα προβλήματα που δεν αντιμετωπίζει το Portage με τον καλύτερο δυνατό τρόπο είναι αυτό των reverse dependencies. Αν κάνουμε απεγκατάσταση του πακέτου AAA (emerge -Cv AAA), το emerge δεν θα αφαιρέσει αυτόματα και το πακέτο BBB. Αυτό το κάνει επειδή μπορεί να έχουμε βάλει στην πορεία και το πακέτο CCC που να έχει και αυτό ως εξάρτηση το BBB οπότε θα ήταν σφάλμα να αφαιρούσαμε το BBB. Υπάρχει, όμως, περίπτωση να ξεμένουν αχρησιμοποίητα πακέτα στο σύστημά μας μετά από αρκετά emerge και unmerge. Για να αποφύγουμε τέτοιες καταστάσεις μπορούμε να τρέχουμε την εντολή:

emerge --depclean -p

Θα μας εμφανίσει μία λίστα με πακέτα που πιθανώς δεν χρησιμοποιούνται από κανένα άλλο. Το αποτέλεσμα της λίστας αυτής δεν είναι πάντοτε 100% ακριβές, γι' αυτό πρέπει πάντα να το τρέχουμε με την επιλογή -p (pretend), ώστε να κάνουμε μόνοι μας τη διαλογή των πακέτων που θέλουμε να

```
# Copyright 1999-2008 Gentoo Foundation
# Distributed under the terms of the GNU General Public License v2
# $Header: $

inherit eutils

DESCRIPTION="Simple GTK+ Text Editor"
HOMEPAGE="http://tarot.freeshell.org/leafpad/"
SRC_URI="http://savannah.nongnu.org/download/leafpad/${P}.tar.gz"

LICENSE="GPL-2"
SLOT="0"
KEYWORDS="--amd64 ~ppc x86"
IUSE="emacs"

RDEPEND=">=x11-libs/gtk+-2.10"
DEPEND="${RDEPEND}
sys-devel/gettext
>=dev-util/intltool-0.31
>=dev-util/pkgconfig-0.9"

src_compile() {
    econf --enable-chooser --enable-print $(use_enable emacs)
    emake
}

src_install() {
    install
    dodoc AUTHORS ChangeLog NEWS README
}
```

To ebuild script του leafpad editor.

αφαιρέσουμε. Η διαλογή των πακέτων είναι μόνο θέμα εμπειρίας και αποκτάται με τον καιρό, οπότε κάντε υπομονή μέχρι να μάθετε τι κάνει κάθε πακέτο που σας εμφανίζει. Σε πολύ λίγο καιρό, θα είστε σε θέση να κρίνετε αν ένα πακέτο σας χρειάζεται πια ή όχι.

Stable και unstable πακέτα

Για κάθε αρχιτεκτονική που υποστηρίζει το Gentoo, υπάρχει το stable branch και το unstable. Τα stable branches συμβολίζονται με x86, amd64, ppc, sparc κ.λπ., ενώ τα unstable με ~x86, ~amd64, ~ppc, ~sparc κ.λπ. Είναι στο χέρι του χρήστη να επιλέξει αν θέλει να χρησιμοποιήσει το stable ή το unstable branch.

Στο unstable υπάρχουν πολύ νεότερες εκδόσεις των πακέτων και συνεχή updates, τα οποία, όμως, δοκιμάζονται από τους χρήστες και, αν δεν παρουσιάσουν προβλήματα, μεταφέρονται στο stable. Κάποια πακέτα είναι δυνατόν να μην υπάρχουν καθόλου στο stable branch, π.χ., το app-portage/porthole.

Ας δούμε ένα παράδειγμα μίας εφαρμογής που έχει εκδόσεις και στο stable και στο unstable branch.

eix himerge

* app-portage/himerge

Available versions: 0.15 ~0.16.1-r1 ~0.17 ~0.17.3 ~0.17.9 ~0.18 ~0.20 ~0.21 ~0.21.9

Βλέπουμε πως όλες οι versions, εκτός της 0.15, έχουν μπροστά τους ένα "~". Αυτό σημαίνει πως ανήκουν στο unstable branch, ενώ η 0.15 ανήκει στο stable. Αν στο stable σύστημά μας θέλουμε να περάσουμε την τελευταία unstable του himerge, πρέπει να κάνουμε το εξής:

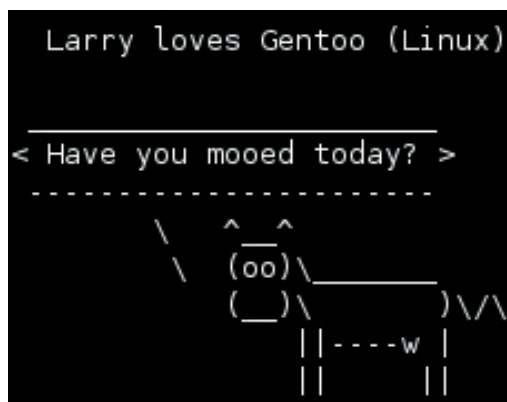
```
# echo "app-portage/himerge ~x86" >>
```

BASELAYOUT 2 & OPENRC

Το baselayout 1 είναι το πακέτο που προσφέρει το σύστημα των βασικών αρχείων και καταλόγων του Gentoo, καθώς και το σύστημα rc. Η εξέλιξή του, το baselayout 2 προσφέρει πια μόνο το σύστημα των αρχείων, ενώ το σύστημα rc έγινε ξεχωριστό πακέτο με το όνομα OpenRC. Το OpenRC είναι ένα dependency-based rc σύστημα, που μπορεί να δουλέψει με όποιο init και αν υπάρχει στο μηχανήμα. Ενώ το παλιό rc ήταν γραμμένο σε Bash, το OpenRC είναι γραμμένο σε C και η διαφορά του σε ταχύτητα σε σχέση με το παλιό αγγίζει το 50%! Αυτό σημαίνει ότι πλέον το Gentoo μπορεί να κάνει boot σε χρόνους της τάξης των 20 με 25sec!

Οδηγίες μετάβασης σε baselayout 2 - OpenRC θα βρείτε εδώ: <http://ub0.cc/6/1c>.

emerge -pv moo...



```
/etc/portage/package.keywords
```

Αν τρέξουμε και πάλι το `eix`, θα μας ειδοποιήσει πως έχει αναιρεθεί ο περιορισμός.

```
# eix himerge
```

```
* app-portage/himerge
```

```
Available versions: 0.15 (~)0.16.1-r1 (~)0.17
(~)0.17.3 (~)0.17.9 (~)0.18 (~)0.20 (~)0.21
(~)0.21.9
```

Οπότε μπορούμε πλέον να εγκαταστήσουμε κανονικά το πακέτο:

```
# emerge -avt himerge
```

Αν θέλουμε να βάλουμε μία συγκεκριμένη έκδοση από το unstable, αλλά να μην αναβαθμιστεί μόλις βγει η νεότερη unstable, τότε μπορούμε να κάνουμε:

```
# echo "=app-portage/himerge-0.17.3 ~x86" >>
```

```
/etc/portage/package.keywords
```

```
# emerge -avt =app-portage/himerge-0.17.3
```

Αρκετοί χρήστες λανθασμένα χρησιμοποιούν την εντολή:

```
# ACCEPT_KEYWORDS="~x86" emerge -avt himerge
```

Αυτή η εντολή έχει μεν ως αποτέλεσμα το να εγκατασταθεί η τελευταία unstable έκδοση, αλλά επειδή δεν καταγράφεται το keywording στο `/etc/portage/package.keywords`, την επόμενη φορά που θα κάνουμε `emerge` το `himerge` ή το `world`, θα μας κάνει downgrade στην τελευταία stable, δηλαδή, στην 0.15, σύμφωνα με το παράδειγμα.

Μασκαρεμένα πακέτα!

Πέρα από τα stable και τα unstable πακέτα, υπάρχουν και τα masked! Τα τελευταία είναι ότι πιο φρέσκο υπάρχει στο Portage Tree, αλλά είναι masked, γιατί κατά πάσα πιθανότητα υπάρχει κάποιο σοβαρό bug ή δεν έχει γίνει καθόλου testing. Οπότε, μέχρι να λυθούν αυτά τα θέματα, οι developers του Gentoo δεν μας αφήνουν να τα εγκαταστήσουμε. Εάν έχουμε κάποιο πολύ σοβαρό λόγο και θέλουμε να εγκαταστήσουμε οπωσδήποτε ένα masked πακέτο και γνωρίζουμε ότι αυτό μπορεί να αποβεί μοιραίο για το Gentoo μας, τότε μπορούμε να κάνουμε `unmask` το πακέτο με την εντολή:

```
# echo "sys-apps/test_package" >>
```

```
/etc/portage/package.unmask
```

Για τον ακριβώς αντίθετο λόγο, αν δεν θέλουμε, δηλαδή, καινούργια updates για ένα πακέτο στο σύστημά μας, μπορούμε να το κάνουμε εμείς mask! Η παρακάτω εντολή εμποδίζει την εγκατάσταση ενημερώσεων για το `test_package` μεγαλύτερων της έκδοσης 1.2.3.

```
# echo ">foo-bar/test_package-1.2.3" >>
```

```
/etc/portage/package.mask
```

Μετά από εγκαταστάσεις ενημερώσεων, δημιουργούνται πολλές φορές κάποια αρχεία στον κατάλογο `/etc` με ονόματα όπως `._cfg_packageName`. Αυτά είναι τα καινούργια configuration files που φέρνει μαζί του το πακέτο. Για να περάσουν οι αλλαγές στα υπάρχοντα configuration files, είναι απαραίτητη η χρήση ενός εκ των `etc-update` ή `dispatch-conf` εργαλείων. Και τα δύο αναλαμβάνουν τη συγχώνευση των αλλαγών που φέρνουν τα καινούργια configuration files με τα παλιά.

Για τη διευκόλυνση των χρηστών, υπάρχουν διάφορα ανεπίσημα frontends για το `emerge`. Τέτοια είναι το `Kuroo` (QT-based), `porthole` (GTK-based), `himerge` (GTK-based). Δεν παρέχεται επίσημη υποστήριξη γι' αυτά και σίγουρα δεν μπορούν να αντικαταστήσουν όλες τις λειτουργίες του `emerge`.

Βοηθητικά εργαλεία

Εκτός του `eix` για το οποίο μιλήσαμε παραπάνω, υπάρχουν και κάποια άλλα εργαλεία, τα οποία είναι πολύ χρήσιμα για τον καλό Gentoo User. Μέσα στο πακέτο `gentoolkit` υπάρχουν, μεταξύ άλλων, τα εργαλεία `equery`, `glsa-check` και `revdep-rebuild`. Το `equery` είναι ένα πολυεργαλείο γραμμένο σε Python για την εξεύρεση πληροφοριών σχετικών με τα πακέτα του Gentoo.

Για να δούμε όλα τα αρχεία που έχει τοποθετήσει στο σύστημά μας ένα πακέτο, χρησιμοποιούμε την επιλογή `f` (files):

```
# equery f app-shells/bash
```

Με την επιλογή `b` (belongs) μπορούμε να βρούμε σε ποιο πακέτο αντιστοιχεί ένα αρχείο:

```
# equery b /usr/bin/base64
```

Χρησιμοποιώντας την επιλογή `d` (depends), μπορούμε να δούμε ποια πακέτα εξαρτώνται άμεσα από το πακέτο που ψάχνουμε. Αυτό είναι πολύ χρήσιμο όταν θέλουμε να αφαιρέσουμε ένα πακέτο από το σύστημά μας και θέλουμε να δούμε αν αυτό μπορεί να προκαλέσει προβλήματα.

```
# equery d xorg-server
```

Με το εργαλείο `glsa-check` μπορεί κάποιος να κάνει έλεγχο του συστήματός του και να βρει αν υπάρχει κάποιο security update για κάποιο από τα εγκατεστημένα πακέτα που δεν έχει ακόμη "περαστεί".

Η παρακάτω εντολή μάς δείχνει όλα τα security updates που έχουν σχέση με το σύστημά μας:

```
# glsa-check -l -c
```

Ενώ η επόμενη εντολή μάς δείχνει μόνο αυτά που δεν έχουν εγκατασταθεί, τις προβληματικές εκδόσεις που έχουμε στο σύστημά μας, καθώς και λίγες πληροφορίες σχετικά με το πρόβλημα:

```
# glsa-check -l -v -c affected
```

Για να μας δείξει ποια πακέτα πρέπει να περάσουμε στο σύστημά μας, δίνουμε:

```
# glsa-check -p affected
```

Το εργαλείο `revdep-rebuild` κάνει έλεγχο σε όλα τα εκτελέσιμα του συστήματός μας, για να βρει αν κάποιο έχει πρόβλημα με τις shared libraries. Είναι πολύ χρήσιμο, όταν για κάποιους λόγους αφαιρούμε μία βιβλιοθήκη από το σύστημά μας, πάνω στην οποία είχαν βασιστεί κάποια προγράμματα, και τώρα θέλουμε να τα χτίσουμε ξανά χωρίς αυτή τη βιβλιοθήκη. Η επιλογή `-p` (pretend) θα κάνει όλους τους ελέγχους, χωρίς να τρέξει `emerge` στο τέλος, αφήνοντάς μας να το κάνουμε μόνοι μας:

```
# revdep-rebuild -p
```

Άλλο ένα πακέτο που περιέχει πάρα πολλά εργαλεία, είναι το `portage-utils`. Πολλά από αυτά έχουν τις ίδιες λειτουργίες με επιλογές του `equery`, αλλά επειδή είναι γραμμένα σε C, είναι αρκετά πιο γρήγορα:

```
qfile -> equery b
```

```
qlist -> equery f
```

ΟΜΟΡΦΕΣ ΕΚΠΛΗΞΕΙΣ

Τα γνωστά Easter eggs δεν λείπουν από το Gentoo. Δοκιμάστε στο σύστημά σας να περάσετε την επιλογή `FEATURES="candy"` στο `/etc/make.conf` και ξεκινήστε το `emerge` ενός πακέτου. Επίσης, υπάρχει το κλασικό `emerge moo`. :)

```
qdepends -> equery d
```

```
qsize -> equery s
```

```
κ.λπ.
```

Τέλος, το εργαλείο udept μπορεί να βοηθήσει σε πάρα πολλούς τομείς, κυρίως, όμως, στο καθάρισμα του συστήματός μας από άχρηστα πακέτα, καθώς ενσωματώνει μία λειτουργία για την εύρεση του reverse dependency tree ενός πακέτου:

```
# dep -T gpm
```

Πολύ χρήσιμη επιλογή του udept είναι η επιλογή s (springclean), που κάνει ενδελεχή έλεγχο των εξαρτήσεων κάθε πακέτου, ώστε να βρει ποια πακέτα δεν χρειάζονται πια στο σύστημά μας. Είναι ουσιαστικά μία καλύτερη (και πιο αργή) εκδοχή του emerge-depclean.

```
# dep -p -s
```

Τεχνικά Χαρακτηριστικά

Τα runlevels και τα init scripts του Gentoo είναι αρκετά κοντά στο System V τρόπο, που ακολουθούν σχεδόν όλες οι διανομές, αλλά με κάποιες διαφοροποιήσεις. Κατ' αρχάς, όλα τα runlevels βρίσκονται στο φάκελο /etc/runlevels και αντί να είναι αριθμημένα, έχουν ονόματα φιλικά προς το χρήστη :). Ετσι, μας βοηθούν να καταλάβουμε τι κάνει το καθένα, όπως τα runlevels: boot και default. Μέσα σ' αυτούς τους καταλόγους θα βρούμε links για τις υπηρεσίες που θα θέλουμε να τρέξουν κατά την εκκίνηση, οι οποίες είναι συγκεντρωμένες στο /etc/init.d.

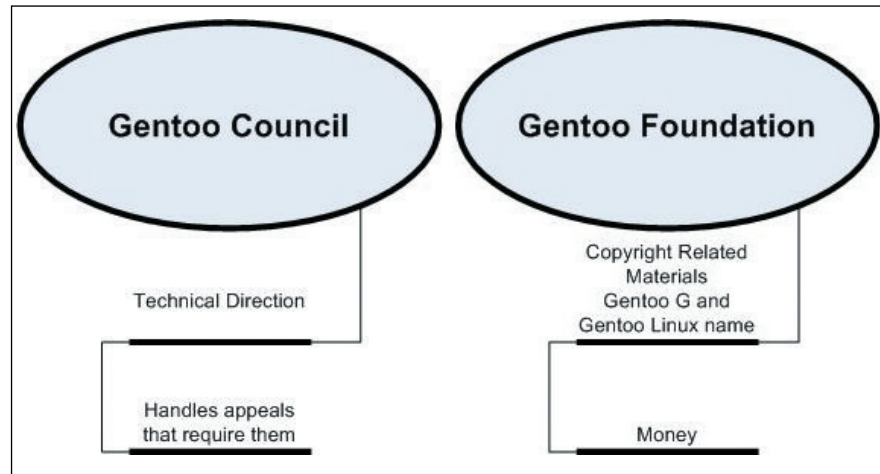
Μία ακόμη διαφορά υπάρχει στον τρόπο με τον οποίο εκτελούνται οι υπηρεσίες. Αντί να υπάρχει ένας αύξοντας αριθμός μπροστά από κάθε υπηρεσία, όπως στις περισσότερες διανομές (π.χ. S10mysql, S11apache), στο Gentoo οι υπηρεσίες έχουν εξαρτήσεις (dependencies). Μ' αυτόν τον τρόπο, αν θέλουμε να δηλώσουμε πως μία υπηρεσία θέλουμε να τρέξει πριν από κάποια άλλη, απλώς πρέπει να γράψουμε κάτι όπως after exotic_service και είμαστε σίγουροι πως θα τρέξει μετά την υπηρεσία exotic_service, χωρίς να μας απασχολεί αν θα τρέξει 10η ή 11η. Ουσιαστικά, η διεργασία εκκίνησης (init), αφού τελειώσει με τα πολύ βασικά πράγματα (πχ. να κάνει mount το root partition που είναι δηλωμένο στο /etc/fstab) καλεί το script /sbin/rc, το οποίο με τη σειρά του θα "λύσει το γρίφο" των εξαρτήσεων των υπηρεσιών και θα τις ξεκινήσει με τη βέλτιστη σειρά, ξεκινώντας πρώτα με αυτές που βρίσκονται στο /etc/runlevels/boot και μετά στο /etc/runlevels/default.

Ας δούμε, όμως, ένα παράδειγμα:

```
depend() {
need net
use mysql dns logger netmount postgresql
after sshd
}
```

Το παραπάνω είναι από το αρχείο /etc/init.d/apache2, που είναι υπεύθυνο για την εκκίνηση του apache web server. Βλέπουμε πως έχει ως εξάρτηση την υπηρεσία net, η οποία πρέπει να εκτελεστεί οπωσδήποτε, αλλιώς ο Apache δεν θα ξεκινήσει καν. Επίσης, χρησιμοποιεί τις υπηρεσίες mysql, dns, logger, netmount, postgresql, τις οποίες θα καλέσει, αν είναι διαθέσιμες, ενώ, αν υπάρχει υπηρεσία sshd, θέλουμε να τρέξει μετά από αυτή.

Τα αρχεία ρυθμίσεων που έχουν σχέση με τις υπηρεσίες που εκκινούν από το /etc/init.d, βρίσκονται όλα κάτω από τον κατάλογο /etc/conf.d. Εκεί μπορούμε να παραμετροποιήσουμε τις υπηρεσίες μας όπως μας βολεύει. Εκεί θα βρείτε επίσης τα αρχεία που ρυθμίζουν την ώρα του συστήματος (clock), το όνομά του (hostname), τον ήχο (alsasound), το δίκτυο (net) και πολλά ακόμη, ενώ υπάρχει και το αρχείο local.start, στο οποίο μπορούμε να προσθέσουμε κάποιες εντολές που θα θέλαμε να τρέξουν κατά την εκκίνηση, και το local.stop για να τρέχουν εντολές κατά το shutdown.



Η δομή λειτουργίας του Gentoo.

Κοινότητα

Η κοινότητα του Gentoo είναι από τις πιο ενεργές και πολλές φορές μπορούμε να δούμε στην πράξη πώς εφαρμόζεται το μοντέλο του Open Source (γνωστό και ως παζάρι) με την αλληλεπίδραση των χρηστών με τους developers.

Αυτό επιτυγχάνεται με πολλούς τρόπους. Κατ' αρχάς, με το forum του Gentoo, το οποίο αριθμεί πάνω από 110.000 μέλη και περισσότερα από 4.200.000 άρθρα. Εκεί θα βρείτε υποστήριξη στα προβλήματα που μπορεί να συναντήσετε, ενώ αρκετά τμήματα του forum προσφέρονται για γενικότερη συζήτηση γύρω από την πορεία του Gentoo. Πρόσφατα μάλιστα, προστέθηκε ένα link κάτω από κάθε είδηση που βγαίνει στην κεντρική σελίδα του Gentoo, που παραπέμπει σε σχετική συζήτηση στο forum, διευκολύνοντας έτσι την αλληλεπίδραση χρηστών και developers.

Παράλληλα με το forum λειτουργούν οι λίστες ταχυδρομείου (περίπου 60), χωρισμένες ανά κατηγορία θεματολογίας και με κύριο ρόλο το συντονισμό των developers του Gentoo.

Ακόμη πιο ζωντανή είναι η κατάσταση στο IRC (στο δίκτυο Freenode), όπου θα βρείτε περίπου 60 επίσημα κανάλια του Gentoo, ανάλογα με τον τομέα της ανάπτυξης, ενώ υπάρχουν και πιο γενικά κανάλια, όπως το #gentoo, για γενική συζήτηση και βοήθεια, το #gentoo-dev, όπου μπορείτε να συμμετέχετε στις συζητήσεις των developers, #gentoo-dev-help, αν θέλετε να μάθετε πώς μπορείτε να βοηθήσετε στην ανάπτυξη του Gentoo.

Τα blogs παίζουν το δικό τους ρόλο στην ανάπτυξη της κοινότητας, οπότε δεν μπορούσαν να λείπουν οι σχετικοί "πλανήτες" (δείτε στις παραπομπές), ένας για να συλλέγει τις δημοσιεύσεις από τα blogs των developers και ένας από τα blogs των χρηστών.

Ελληνική κοινότητα

Δεν θα μπορούσαμε, βέβαια, να μην αναφερθούμε στην ελληνική κοινότητα του Gentoo. Τα σημεία συνάντησής της είναι κυρίως στο IRC, στα κανάλια #gentoo-el (στο Freenode) και #gentoo (στο GrNet), αλλά και σε forums, όπως το ελληνικό τμήμα στο επίσημο forum του Gentoo, αλλά και στο σχετικό τμήμα στο forum του Hellug.

Πρόσφατα ο Αγγελος Ορφανάκος, ένας εκ των τριών Ελλήνων developers του Gentoo, ξεκίνησε μία προσπάθεια καταγραφής των μελών της ελληνικής κοινότητας, με προοπτική να έρθουμε πιο κοντά. Αν είστε χρήστης Gentoo, στείψτε στο <http://ub0.cc/k/2p> :

Ελπίζουμε να σας πείσαμε για τις τρομερές δυνατότητες του Gentoo. Και μην ξεχνάτε... If it moves, compile it! :)

ΔΕΙΤΕ

■ Το επίσημο site του Gentoo
<http://gentoo.org/>

■ Το forum για βοήθεια και συζήτηση
<http://forums.gentoo.org/>

■ Ο πλανήτης των developers
<http://planet.gentoo.org/>

■ Ο πλανήτης των χρηστών
<http://larrythecow.org/>

■ Ένας τεράστιος θησαυρός τεκμηρίωσης
<http://gentoo-wiki.org/>

■ Το επίσημο documentation project
<http://www.gentoo.org/doc/>

■ RSS feed για τα νέα πακέτα
<http://packages.gentoo.org/>

■ USE Flags:
<http://www.gentoo.org/dyn/use-index.xml>