



Ο Γιώργος δουλεύει στο Εθνικό Δίκτυο Έρευνας και Τεχνολογίας ως διαχειριστής συστημάτων.

## «Καθαρίστε» με τον Vim

Αν θέλετε να βελτιώσετε την παραγωγικότητά σας, τότε σας χρειάζεται ένας κειμενογράφος σαν τον Vim.

Όσοι αντικρίζουν για πρώτη φορά τον Vim, το λιγότερο που τους προκαλείται είναι δυσφορία. Μα είναι δυνατόν να εργαστεί κανείς σε έναν κειμενογράφο όπου δεν δουλεύει καν το κλασικό CTRL-C/CTRL-V για το copy paste; Είναι δυνατόν αυτός ο κειμενογράφος, ο οποίος στην έκδοση του τερματικού δεν έχει καν ένα μικρό μενού για τις βασικές επιλογές, να μπορεί να κάνει τη ζωή κάποιου πιο εύκολη; Κι όμως, αν ξεπεράσει κανείς το αρχικό σοκ και ξεκινήσει με καθαρό μυαλό, θα δει πως ο Vim μπορεί να τον βοηθήσει να γίνει πιο παραγωγικός από ποτέ.

Λίγη ιστορία πρώτα όμως. Οι πρώτοι επεξεργαστές κειμένου σε Unix μηχανήματα λειτουργούσαν σε επίπεδο «γραμμής» (line editors). Ο χρήστης είχε στα χέρια του ένα πληκτρολόγιο και έναν εκτυπωτή και ό,τι χρειαζόταν να εκτυπωθεί μπορούσε να το επεξεργαστεί γραμμή-γραμμή. Ο διασημότερος ίσως τέτοιος επεξεργαστής ήταν ο 'ed', γραμμένος από τον Ken Thompson το 1971, εξέλιξη του οποίου ήταν ο 'em', ο 'eh' και ο 'ex', με τη διαφορά πως οι τελευταίοι είχαν φτιαχτεί για να μπορούν να επεξεργάζονται γραμμές κειμένων και σε οθόνες. Εξέλιξη του visual mode του 'ex' ήταν ο 'vi' (visual in 'ex'), του οποίου η πρώτη έκδοση βγήκε το 1977 από τον Bill Joy. Από τα σημαντικότερα χαρακτηριστικά του vi είναι η τροπικότητα (modality), υπάρχουν δηλαδή διαφορετικές καταστάσεις (modes) για διαφορετικές λειτουργίες: μία για να επεξεργάζεται κανείς το κείμενο, μία για να επιλέγει και να αντιγράφει κείμενο, καθώς και μία ακόμα για να τρέχει εντολές.

Ο Vim (Vi Improved) δημιουργήθηκε το 1991 από τον Bram Moolenaar ως ένας κλώνος του vi για την Amiga. Από εκεί και πέρα όμως είχε ραγδαία ανάπτυξη και πολύ σύντομα μπόρεσε να τρέξει σε όλα τα λειτουργικά. Άλλα «παιδιά» του vi είναι ο 'stevie', ο 'elvis', ο 'nvi' και ο 'vile' (Vi Like Emacs). Κατά τους παραδοσιακούς πολέμους των χρηστών σχετικά με το ποιος είναι ο καλύτερος editor μεταξύ Vim και Emacs (ο οποίος επίσης προέρχεται από έναν διαφορετικό line-editor με το όνομα TECO), ο Vim είναι αυτός που συνεχίζει τη μάχη από τη μεριά των κλώνων του vi.

### Ξεκινώντας!

Οι ρυθμίσεις του Vim βρίσκονται σε τρία βασικά αρχεία παραμετροποίησης, το vimrc, το gvimrc και το exrc. Το πρώτο έχει να κάνει με τις βασικές ρυθμίσεις, το δεύτερο με το γρα-

#### Εκμάθηση

Ο καλύτερος τρόπος να ξεκινήσει κανείς να μαθαίνει τον Vim είναι να τρέξει από ένα terminal το πρόγραμμα vimtutor. Σε μισή περίπου ώρα θα έχει μάθει τις βασικές εντολές τρέχοντάς τες μέσα σε ένα διαδραστικό εκπαιδευτικό κείμενο.

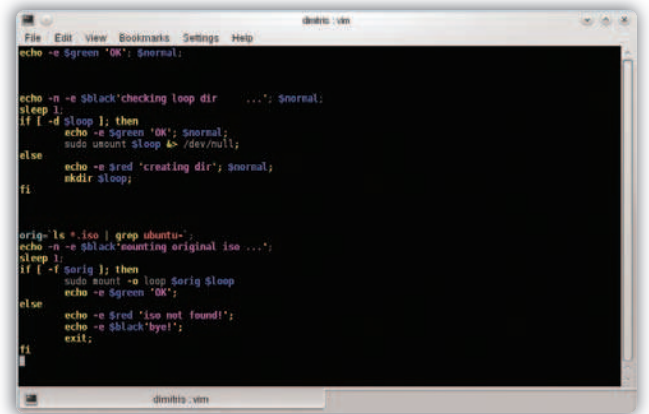


Το blog του Γιώργου

Εργαλεία: Vim

Δυσκολία: ★★☆☆☆

URL: [www.vim.org](http://www.vim.org)



### 1 Το Vim εν δράσει.

φικό κομμάτι του Vim που λέγεται gvim (περισσότερα σε μελλοντικό άρθρο) και το τρίτο έχει να κάνει με τον line editor 'ex', ο οποίος έρχεται μαζί με το πακέτο vi/vim της διανομής σας. Για κάθε αρχείο παραμετροποίησης υπάρχει ένα καθολικό (global), που βρίσκεται συνήθως στο /etc/vim/, καθώς και ένα προσωπικό που βρίσκεται στο τοπικό κατάλογο κάθε χρήστη, π.χ. ~/.vimrc.

Η εκκίνηση του επεξεργαστή γίνεται γράφοντας 'vim' σε ένα τερματικό. Για να γράψει κανείς κάτι πρέπει να πατήσει πρώτα το αγγλικό 'i' (insert) και έπειτα να γράψει αυτό που θέλει. Για να σταματήσει να γράφει πρέπει να πατήσει το πλήκτρο Esc. Για να βγει κανείς από τον vim χωρίς να «σώσει» πρέπει να πατήσει ':q!' (quit) – κανονικά η εντολή είναι ':q' για έξοδο χωρίς αλλαγές και ':wq' για σώσιμο και έξοδο. Για να σωθεί ένα κείμενο με νέο όνομα αρκεί το ':w όνομα' (write). Αν κάποιος τα κάνει αυτά, έχει ήδη γνωρίσει τις τρεις διαφορετικές καταστάσεις στις οποίες μπορεί να βρεθεί μέσα στον vim. Πατώντας το 'i' γίνεται είσοδος στο Insert mode, όπου μπορεί κάποιος να πληκτρολογήσει το κείμενό του. Πατώντας το Esc επανέρχεται στο Normal mode και πατώντας ':' βγαίνει η γραμμή εντολών του Command mode. Κάποιος που θέλει να μάθει vim πρέπει να έχει πάντα στο μυαλό του αυτές τις τρεις διαφορετικές καταστάσεις.

Η μετακίνηση μέσα στο vi γίνεται παραδοσιακά με τα τέσσερα κουμπιά: h (αριστερά), j (κάτω), k (πάνω), l (δεξιά), αλλά ο vim υποστηρίζει και τα βελιάκια για όσους δυσκολεύονται με τα προαναφερθέντα. Πατώντας w ή b μπορεί κανείς να μετακινηθεί μία λέξη μπροστά ή πίσω αντίστοιχα. Για τη μετακίνηση στο τέλος μίας γραμμής μπορεί κανείς να χρησιμοποιήσει το \$, ενώ για να μετακινηθεί στην αρχή της είτε το 0 (μηδέν) είτε το ^. Η διαγραφή ενός χαρακτήρα προς τα πίσω κατά το Insert mode γίνεται με το backspace (ή πατώντας CTRL+h), ενώ για να σβήσει κάποιος έναν χαρακτήρα από μπροστά μπορεί να χρησιμοποιήσει το πλήκτρο Del. Σε Normal mode,

**Μάθετε νέα κόλπα!**

Όσους ανθρώπους που χρησιμοποιούν Vim και να γνωρίσετε, πάντα κάποιος θα ξέρει και κάποιες άλλες εντολές ή μερικούς πιο αποδοτικούς τρόπους επεξεργασίας ενός κειμένου που εσείς δεν είχατε σκεφτεί μέχρι τώρα. Όσο χρησιμοποιείτε τον Vim θα ανακαλύψετε συνεχώς τρόπους να κάνετε τη δουλειά σας όλο και πιο αποδοτικά. Οπότε μην ντρέπεστε! Ρωτήστε φίλους και γνωστούς για τα αγαπημένα τους «κόλπα» στο vim.

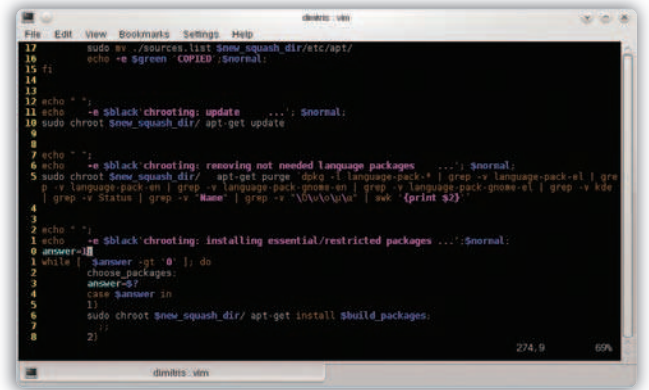
μπορεί κανείς να σβήσει το χαρακτήρα κάτω από τον cursor πατώντας το 'x'.

Όπως βλέπετε, στο Vim πρέπει κανείς να «μιλάει». Οι εντολές προέρχονται από αγγλικές λέξεις και έτσι δεν χρειάζεται κάποιος να αποστηθίσει συντομειώσεις, αρκεί να του «λέει» τι θέλει να κάνει. Αυτό θα φανεί έντονα στα παρακάτω παραδείγματα. Έχοντας πλέον κάποιος τη δυνατότητα να γράψει και να σβήσει κείμενο, καθώς και να μετακινηθεί μέσα σε αυτό, τίθεται το πρώτο βασικό πρόβλημα. «Μα πώς μπορώ να κάνω copy paste;» Όπως ειπώθηκε παραπάνω, ο Vim ξεκινάει ως line editor, οπότε ο πιο απλός τρόπος για να αντιγράψει κανείς κάτι είναι σε επίπεδο γραμμής. Όντας σε Normal mode και έχοντας φτάσει με τα πλήκτρα/βέλη στη γραμμή που θέλουμε να αντιγράψουμε, πρέπει να πατήσουμε 'yy' (yank-yank). Έπειτα, πάλι με τα πλήκτρα να κινηθούμε στο σημείο κάτω από το οποίο θέλουμε να γίνει επικόλληση της γραμμής που έχουμε αντιγράψει, και να την εκτελέσουμε πατώντας το 'p' (paste).

Για να διαγράψει κανείς μία γραμμή ο απλούστερος τρόπος είναι να πάει στη γραμμή εκείνη και να πατήσει 'dd' (delete). Προφανώς, για το cut-n-paste αρκεί να δώσουμε 'dd' και έπειτα 'p' στο σημείο όπου θέλουμε να γίνει η επικόλληση. Οι παραπάνω εντολές όμως μπορούν να δουλέψουν και με πολλαπλασιαστές, βάζοντας δηλαδή έναν αριθμό μπροστά από κάθε εντολή, αυτή επαναλαμβάνεται τόσες φορές όσες και ο αριθμός. Για παράδειγμα με την εντολή 5yy αντιγράφει κανείς έξι συνεχόμενες γραμμές, ενώ με την 2p μπορεί να επικολλήσει δύο φορές αυτό που έχει αντιγράψει παραπάνω. Αντίστοιχα, η εντολή 3dd σβήνει τρεις γραμμές. Ήδη οι πρώτοι σκεπτικιστές θα παραπονεθούν για το γεγονός πως όλα γίνονται σε επίπεδο γραμμής. Αν κάποιος θέλει να αντιγράψει δύο λέξεις ή επτά χαρακτήρες τότε τι γίνεται; Εκτός λοιπόν από τους πολλαπλασιαστές γραμμών που μπαίνουν πριν από τις εντολές, υπάρχουν και ειδικοί τελεστές που μπαίνουν μετά από αυτές και καθορίζουν το πεδίο εφαρμογής τους. Για παράδειγμα, η εντολή 3yw αντιγράφει τις τρεις επόμενες λέξεις (w->word) μετά τον cursor. Αντίστοιχα η εντολή 7yl (yank-last) αντιγράφει τους επόμενους επτά χαρακτήρες και η 4x σβήνει τους επόμενους τέσσερις χαρακτήρες.

**Προσαρμογή του vim.**

Για να χρησιμοποιήσει κάποιος αποδοτικά έναν επεξεργαστή κειμένου, πρέπει και να μπορεί να τον τροποποιήσει θεματικά όπως του αρέσει. Στην περίπτωση που το Vim χρησιμοποιείται από κάποιο τερματικό, τότε, για να αλλάξει η γραμματοσειρά, πρέπει να αλλάξει το font του τερματικού. Ο Vim διαθέτει δυνατότητες αλλαγής των χρησιμοποιούμενων χρωμάτων του (theming) μέσω της εντολής ':colorscheme'. Αφήνοντας ένα κενό μετά το :colorscheme και πατώντας TAB, μπορεί κανείς να δει όλα τα θέματα που είναι διαθέσιμα. Η

**2 Το Vim με όλα τα κολπάκια που ενεργοποιήσαμε...**

επιλογή ενός καλού θέματος είναι πάρα πολύ σημαντική, γιατί επηρεάζει τον τρόπο με τον οποίο γίνονται highlight τα κομμάτια του κειμένου/κώδικα που διαβάει ή επεξεργάζεται κανείς. Εκτός από τα θέματα που θα βρει κανείς προεγκατεστημένα, υπάρχουν και πάρα πολλά διαθέσιμα στο Internet. Καλό είναι να δοκιμάσει κάποιος αρκετά, ανοίγοντας και διαφορετικού τύπου αρχεία πριν αποφασίσει να κρατήσει κάποιο, ώστε να δει τις διαφορές πιο αναλυτικά. Για να γίνει η αλλαγή του θέματος μόνιμη, πρέπει να ανοίξει κανείς, με το vim φυσικά, το αρχείο ~/.vimrc και να προσθέσει μία γραμμή όπως η παρακάτω:

```
colorscheme XXXXXX
```

όπου XXXXXX το όνομα του θέματος. Φυσικά, έχοντας τις κατάλληλες γνώσεις μπορεί κανείς να τροποποιήσει τα θέματα ώστε να τον βολεύουν ακόμα περισσότερο.

Άλλο ένα πολύ χρήσιμο στοιχείο του vim είναι η μπάρα κατάστασης που υπάρχει στο κάτω μέρος. Εκεί μπορεί κανείς να εμφανίσει χρήσιμα στοιχεία, όπως το file format του τρέχοντος ανοιχτού αρχείου, τον τύπο του αρχείου, τη θέση του cursor μέσα στο αρχείο και άλλα διάφορα. Η παρακάτω γραμμή στο ~/.vimrc θα σας δώσει μερικά από αυτά:

```
set statusline=%F%m%r%h%w\ [FORMAT=%{&ff}]\
[TYPE=%Y]\ [ASCII=\%04.8b]\ [HEX=\%04.4B]\
[POS=%04i,%04v][%p%%%\] [LEN=%L]
```

Πολύ βολική κατά τη διάρκεια προγραμματισμού αλλά και κατά τη συγγραφή κειμένων είναι η προσθήκη της ένδειξης της τρέχουσας γραμμής στα αριστερά του κειμένου. Αυτό στο Vim γίνεται με την προσθήκη του set number στο ~/.vimrc, ενώ αν διαθέτει κανείς έκδοση Vim 7.3 και άνω, τότε η εντολή

```
set relativenumber
```

δείχνει την απόσταση των υπόλοιπων γραμμών από την τρέχουσα, γεγονός ιδιαίτερα χρήσιμο όταν θέλει κανείς να αντιγράψει/διαγράψει ένα κομμάτι κειμένου που αποτελείται από αρκετές γραμμές. Το relativenumber μάς ενημερώνει για τον αριθμό τους, χωρίς να χρειάζεται ειδικά να τις μετρήσουμε.

Το άρθρο αυτό ήταν μία πολύ βασική εισαγωγή σε δομικά στοιχεία του vim. Σε επόμενα άρθρα θα δίνονται όλο και πιο εκτενείς λεπτομέρειες χρήσης στοιχείων του vim.

**Σύνδεσμοι**

- [1] Vim: <http://www.vim.org/>
- [2] Vim wiki: [http://vim.wikia.com/wiki/Vim\\_Tips\\_Wiki](http://vim.wikia.com/wiki/Vim_Tips_Wiki)
- [3] Το δικό μου vimrc: <http://www.void.gr/kargiv/dot/vimrc>