



## Linux Labs – Vim Tricks

Του Γιώργου Καργιωτάκη <kargig@void.gr>



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

# Καθαρίστε με το Vim (Μέρος 2ο)

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



Το blog του  
Γιώργου

Εργαλεία: Vim

Δυσκολία: ★★☆☆☆

URL: <http://www.vim.org>

Έχοντας μάθει κανείς τους βασικούς τρόπους για να κινείται μέσα στο περιβάλλον του Vim, πώς να «μιλάει» στο πρόγραμμα και να δίνει εντολές, καθώς και λίγα πράγματα για το αρχείο παραμετροποίησής του, είναι ώρα «να μάθει 5 πράγματα παραπάνω»...

### Γρηγορότερη μετακίνηση σε αρχείο

Η μετακίνηση μέσα σε ένα αρχείο με τα βέλη ή τα πλήκτρα h, j, k, l, όπως είχε αναφερθεί στο προηγούμενο άρθρο, δεν είναι, φυσικά, ό,τι γρηγορότερο, αλλά υπάρχουν κάποιες συντομεύσεις του Vim που κάνουν τη ζωή του χρήστη πολύ πιο εύκολη. Υπάρχει η δυνατότητα να διαφοροποιείται ο τρόπος μετακίνησης σε ένα αρχείο ανάλογα με τα περιεχόμενά του. Διαφορετικές ανάγκες μετακίνησης έχει κανείς μέσα σε ένα αρχείο κειμένου και διαφορετικές μέσα σε ένα αρχείο κώδικα. Για τα αρχεία κειμένου, σε normal mode, η μετακίνηση στην αρχή ή στο τέλος μιας παραγράφου γίνεται με την { και την } αντίστοιχα. Στην ουσία, η μετακίνηση θα γίνει προς την πρώτη κενή γραμμή πάνω ή κάτω από την παράγραφο που βρίσκεται αυτή τη στιγμή προς επεξεργασία. Για τη μετακίνηση στην προηγούμενη ή στην επόμενη πρόταση μπορεί κανείς να χρησιμοποιήσει τις ( και ).

Για τη μετακίνηση μέσα σε αρχείο με κώδικα, το Vim έχει τη

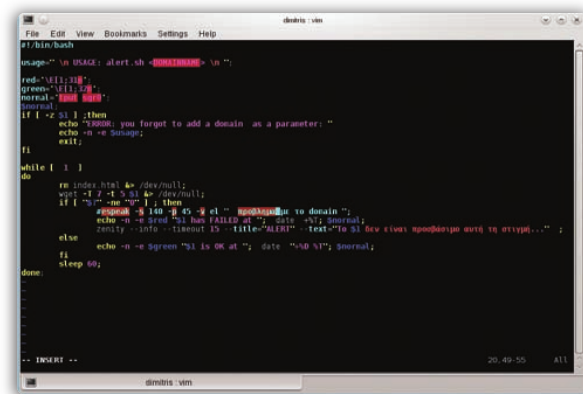
**Τις εντολές set XXXXX που χρειάζεται κανείς σε μόνιμη βάση, πρέπει να τις περνάει στο αρχείο ρυθμίσεων του χρήστη του (~/.vimrc).**

δυνατότητα να καταλαβαίνει τα λογικά blocks κώδικα για διάφορες γλώσσες και επιτρέπει τη μετακίνηση στην αρχή ή το τέλος τους. Για παράδειγμα, σε ένα κομμάτι κώδικα C που περιέχει ένα for loop, πατώντας το % μετακινείται κανείς ανάμεσα στις { }. Πατώντας το % στη γραμμή ορισμού ενός if,

### Μετακίνηση & Λέξεις

Όσοι δοκίμασαν να μετακινηθούν μέσα σε μία πρόταση με το w, b, θα είδαν ότι το vim θεωρεί ως λέξεις ακόμα και το κόμμα, την τελεία κ.λπ.

Στην πραγματικότητα, το vim έχει δύο κατηγορίες: τις «λέξεις», που αποτελούνται από αριθμητικά σύμβολα, παύλα και κάτω παύλα, και τις «ΛΕΞΕΙΣ» που αποτελούνται από οτιδήποτε εκτός από κενούς χαρακτήρες (space ή tab). Για να μετακινηθεί κανείς μία ΛΕΞΗ μπροστά ή πίσω, αρκεί να πατήσει W ή B, αντίστοιχα. Για παράδειγμα, για να μετακινηθεί κανείς στην παρακάτω έκφραση «ένα, δύο, τρία-τέσσερα. Πέντε, έξι. Επτά», από το «ένα» στο «Πέντε», μπορεί είτε να πατήσει «8w» είτε «3W». Φυσικά, θα μπορούσε να το κάνει πατώντας απλά και την ).



Έχοντας ενεργοποιήσει τον ορθογραφικό έλεγχο, το Vim μας ειδοποιεί ότι γράψαμε λάθος τη λέξη «πρόβλημα»...

μετακινείται κανείς μεταξύ των ( ), καθώς και από το #if στο #else και στο #endif ενός definition. Φυσικά, μεταξύ ενός block κώδικα που περιβάλλεται από { } είναι σύνηθες να υπάρχουν και άλλα τέτοια blocks στο εσωτερικό τους, οπότε, αν βρίσκεται κανείς στο εσωτερικό block, πατώντας το % δεν γίνεται να μετακινηθεί στην αρχή του function. Αυτό μπορεί να το καταφέρει, όμως, χρησιμοποιώντας τα [[ και ]], με τα οποία μπορεί κανείς να μετακινηθεί στην αρχή του προηγούμενου function και στην αρχή του επόμενου function, αντίστοιχα. Με τα [] και ]] γίνεται η μετακίνηση στο τέλος του προηγούμενου function και στο τέλος του επόμενου, αντίστοιχα.

Ακόμα μία χρήσιμη συντόμευση για τη μετακίνηση μέσα σε ένα αρχείο, ανεξαρτήτως περιεχομένου, είναι η μετακίνηση μεταξύ των γραμμών που έχουν γίνει οι τελευταίες αλλαγές. Έχοντας ένα αρχείο ανοιχτό, το Vim κρατάει μέχρι και τις τελευταίες 999 αλλαγές και μπορεί κανείς να μετακινηθεί μεταξύ αυτών των σημείων. Για δει κάποιος όλες τις αποθηκευμένες αλλαγές, μπορεί να γράψει σε command mode (πατώντας :) «changes». Σε normal mode πατώντας «g» θα μετακινηθεί στην προηγούμενη αλλαγή, ενώ έχοντας πάει κάποιες αλλαγές πίσω, μπορεί να μετακινηθεί στις επόμενες με το συνδυασμό «g,». Τέλος, για τη μετακίνηση απευθείας στην κορυφή του αρχείου αρκεί να δώσει κανείς σε normal mode «gg», ενώ για να μετακινηθεί στο τέλος «G».

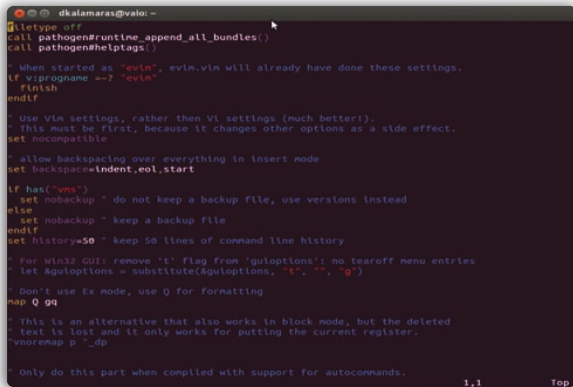
### Έλεγχος ορθογραφίας

Στην έκδοση 7.0 του Vim έχει προστεθεί εγγενής έλεγχος ορθογραφίας. Έτσι, είναι δυνατόν το Vim να ελέγχει ένα κείμενο για ορθογραφικά λάθη ταυτόχρονα σε πολλαπλές γλώσσες και να προτείνει ακόμα και διορθώσεις. Η ενεργοποίηση του ελέγχου ορθογραφίας γίνεται με την εντολή

```
set spell
```

σε command mode. Μετά από αυτό το Vim θα δείχνει τις λέ-





**Το αρχείο ρυθμίσεών μου. Μπορείτε να το κατεβάσετε από τους Συνδέσμους στο τέλος του άρθρου.**

Ξεις που είναι γραμμένες λάθος με ένα πιο έντονο background, αναλόγα και με το colorscheme που χρησιμοποιείται. Η επιλογή γλώσσας ορθογραφίας γίνεται με την εντολή «set spelllang». Αν κάποιος θέλει να ενεργοποιήσει αγγλικό και ελληνικό έλεγχο ορθογραφίας ταυτόχρονα, δεν έχει παρά να θέσει:

```
:set spelllang=en,el
```

Αν το λεξικό δεν υπάρχει για κάποια γλώσσα, το Vim θα ρωτήσει αν θέλει ο χρήστης να το κατεβάσει εκείνη τη στιγμή. Για τις πιθανές διορθώσεις σε μία λέξη αρκεί να πατήσει κανείς «z=» σε normal mode ενώ βρίσκεται πάνω από τη λέξη και έπειτα να διαλέξει ποια λέξη προτιμά από τη λίστα που θα του εμφανιστεί. Για προτάσεις διορθώσεων, ενώ βρίσκεται κανείς σε insert mode, αρκεί να πατήσει Ctrl+X πάνω από τη λέξη, να πατήσει s και έπειτα επιλογή πάλι από την προβαλλόμενη λίστα. Για μετακίνηση στην επόμενη λέξη με ορθογραφικό λάθος αρκεί να δώσει κανείς «]s» και αντίστοιχα «[s» για την προηγούμενη. Με «zg» προσθέτει κανείς την τρέχουσα λέξη ως «σωστή» σε ένα από τα spellfiles του Vim.

**Αναζήτηση**

Από τις πιο συχνές λειτουργίες που εκτελεί κάποιος έχοντας ανοικτό ένα αρχείο, είτε κείμενο είτε κώδικα, είναι να αναζητά κάποιες λέξεις μέσα σε αυτό. Η απλούστερη αναζήτηση είναι η εύρεση της επόμενης φοράς που παρουσιάζεται μία λέξη- αυτό γίνεται σε normal mode πατώντας το / και γράφοντας έπειτα τα γράμματα της λέξης που ενδιαφέρουν. Η μετακίνηση στην επόμενη παρουσία μίας λέξης γίνεται με το n, ενώ στην προηγούμενη με το N. Για να ψάξει κάποιος ένα κείμενο προς τα πίσω, μπορεί να χρησιμοποιήσει το ? και να γράψει έπειτα τη λέξη που τον ενδιαφέρει. Αρκετά χρήσιμη είναι η εντολή

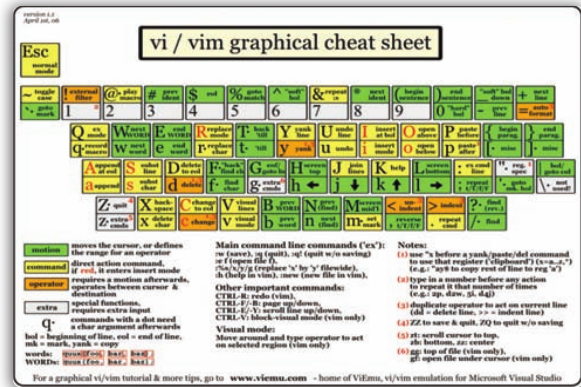
```
set incsearch
```

με την οποία, όταν γράφει κάποιος τη λέξη προς αναζήτηση, ο cursor πηγαίνει αυτόματα στην πρώτη λέξη στο κείμενο που ταυριάζει με την αναζήτηση.

```
set hlsearch
```

τονίζονται όλες οι εμφανίσεις των χαρακτήρων/λέξεων προς αναζήτηση σε ολόκληρο το κείμενο. Φυσικά, το Vim υποστηρίζει αναζήτηση μέσω regular expressions, κάτι που ενδυναμώνει ακόμα περισσότερο τη συγκεκριμένη λειτουργία.

Για να εκτελεστεί μία case-insensitive αναζήτηση, αρκεί να προηγηθεί ένα \c στο search pattern. Αν όμως κάποιος επιθυ-



**Σε μία εικόνα περιγράφονται οι λειτουργίες του Vim, η χρήση όλων των κουμπιών και τα βασικά χαρακτηριστικά. Δείτε περισσότερα εδώ: [goo.gl/7tznv](http://goo.gl/7tznv).**

μεί οι αναζητήσεις του να είναι μόνιμα case-insensitive, τότε μπορεί να προσθέσει στο αρχείο .vimrc του την εντολή

```
set ignorecase
```

Υπάρχει όμως και η δυνατότητα αναζήτησης επόμενων και προηγούμενων αναφορών μίας λέξης χωρίς να μπει κανείς σε command mode, αρκεί να πάει με τον cursor πάνω από τη λέξη που θέλει να βρει και να πατήσει είτε «\*» για τις επόμενες αναφορές είτε «#» για τις προηγούμενες αναφορές. Πατώντας ξανά τα πλήκτρα αυτά, έχει κανείς τη δυνατότητα να μεταπηδήσει στις επόμενες αναφορές της ίδιας λέξης. Πολλές φορές, όμως, η λέξη κάτω από τον cursor είναι απλά κομμάτι μίας λέξης που θέλει κανείς να ψάξει, για να βρει τις επόμενες αναφορές που περιέχουν τη λέξη κάτω από τον cursor, αρκεί να πατήσει «g\*», ενώ με το «g#» μπορεί να πάει στις προηγούμενες.

**Αν σε κάποια εντολή/ρύθμιση μπει το prefix «no», τότε αντιστρέφεται η λειτουργία της, π.χ. set nospell, set noincsearch κ.λπ.**

**Επικόλληση κειμένου**

Ένα από τα συχνότερα παράπονα χρηστών είναι ότι όταν επικollούν κείμενο μέσα στο Vim –ενώ βρίσκονται σε insert mode– αυτό μπορεί να χάσει τη διάταξη του. Λόγω του αυτόματου indentation, που πολλές φορές είναι ενεργοποιημένο στο default vimrc μέσω του «set autoindent», ο κώδικας/κείμενο κάνει «σκαλάκια» προς το εσωτερικό. Κάθε γραμμή μπαίνει πιο μέσα από την προηγούμενη. Αυτό λύνεται εύκολα με την εντολή:

```
set paste
```

ενώ, για να γυρίσει κανείς στο προηγούμενο καθεστώς, αρκεί να δώσει

```
set nopaste
```

ενώσ βρίσκεται σε command mode. Καλή συνέχεια!

**Σύνδεσμοι**

- kargig's vimrc: <http://www.void.gr/kargig/dot/vimrc>
- vim wiki: [http://vim.wikia.com/wiki/Vim\\_Tips\\_Wiki](http://vim.wikia.com/wiki/Vim_Tips_Wiki)