

Vi IMproved textový editor

Krátky úvod

Norbert Szetei, Marián Svetík, Peter Lenčేశ, Július Kinský



[Domovská stránka](#)

[Print](#)

[Titulná strana](#)

[Obsah](#)



Strana **1** z **28**

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Táto publikácia vznikla ako študentský projekt v rámci predmetu *GNU/Linux a jeho prostriedky* na Fakulte matematiky fyziky imformatiky Univerzity Komenského v Bratislave.

V práci sú použité názvy programových produktov, firiem a pod., ktoré môžu byť ochrannými známkami alebo registrovanými ochrannými známkami príslušných vlastníkov.

Sadzba programom pdfT_EX

Copyright © 2008 Norbert Szetei, Marián Svetík, Peter Lenčes, Július Kunský

Ktokoľvek má povolenie vyhotoviť alebo distribuovať doslovný alebo modifikovaný opis tohoto dokumentu alebo jeho časti akýmkoľvek médiom za predpokladu, že bude zachované oznámenie o copyrighte a autoroch. Pritom musí byť zachované aj toto oznámenie, a to v nezmenenej podobe.

Obsah

Úvod	4
1 Program Vi IMproved	5
1.1 História Vi IMproved	5
1.2 Výhody a nevýhody Vi IMproved	5
1.3 Funkcionalita	6
1.4 Inštalácia Vi IMproved	7
1.4.1 Debian GNU/Linux	7
1.4.2 MS Windows	7
2 Prvé kroky	7
2.1 Ako začať	7
2.2 Spustenie	8
2.3 Módy editora Vi IMproved	9
2.3.1 Normal mode (normálny režim)	9
2.3.2 Insert Mode (vkladací režim)	11
2.3.3 CLI Mode	11
2.3.4 ex mode	11
2.3.5 Visual mode	11
3 Práca s VIM	12
3.1 Základné pohyby	12
3.2 Tags	12
3.3 Prvý dokument	13

3.4	Príkazy	13
3.4.1	Príkazy na editovanie súboru	13
3.4.2	Vkladanie textu	14
3.4.3	Vkladanie súboru	15
3.5	Mazanie textu	15
3.6	Zmena, nahradenie textu	16
3.7	Kopírovanie a pohyb v texte	16
3.8	Späť/Zopakuj	17
3.9	Ukončenie	17
3.10	Pohyby medzi slovami	18
3.11	Pohyby na riadku	18
3.12	Presun na určitý znak	19
3.13	Presun na určitý riadok	19
4	Číslovanie (count)	19
5	Jednoduché vyhľadávanie	20
6	Značky (marks)	21
7	Konfigurácia, vimrc	21
8	Makrá	24
9	Jednoduché regulárne výrazy	24
10	Záver	26

[Domovská stránka](#)

[Print](#)

[Titulná strana](#)

[Obsah](#)



[Strana 4 z 28](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

11 Užitočné odkazy

27

[Domovská stránka](#)

[Print](#)

[Titulná strana](#)

[Obsah](#)



Strana **5** z **28**

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Úvod

Textový editor vim je najpopulárnejší editor v operačných systémoch Linux a v rôznych deriváciách Unixu. Jeho predchodcu 'vi' je možno nájsť na takmer každom unix like systéme. Opak je naozaj raritou.

Medzi jeho podstatné výhody patrí, že je dostupný na rôznych OS pre rôzne platformy. V čase písania tejto príručky, vim dokáže bežať pod MS DOS, MS Windows, Amiga, OS/2, Macintosh, Sun Solaris, FreeBSD, NetBSD, OpenBSD, HP-UX. Rovnako existuje verzia vim, ktorá neprevýši 600kB, čo je ďalším z dôvodov jeho rozšíriteľnosti.

Bratislava 2008

Autori

[Domovská stránka](#)

[Print](#)

[Titulná strana](#)

[Obsah](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Strana **6** z **28**

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

1. Program Vi IMproved

1.1. História Vi IMproved

V roku 1976 vydal Bill Joy editor vi. Názov pochádza zo slova 'visual', t.j. meno príkazu, ktorým disponoval editor ex¹. Bill je známy okrem iného aj tým, že v roku 1982 spolu s ďalšími ľuďmi zakladal firmu Sun Microsystems, vytvoril jazyk Java a obohatil svet o mnohé výtvary a zaujímavé myšlienky. Neskôr, po mnohých vylepšeniach, v roku 1991 Bram Moolenaar vypustil editor, ktorým vi natoľko vylepšil, že si zaslúžil nový názov, Vi IMproved, skrátene vim. Je spätne kompatibilný s vi, avšak sa neodporúča ho v tomto režime používať.

1.2. Výhody a nevýhody Vi IMproved

Keďže sa plánujeme naučiť pracovať s vim, mali by sme zvážiť, prečo. Existuje mnoho open-source alternatív, textových aj grafických, napríklad mcedit, pico, nano, emacs, kate. Pokiaľ zvládneme prácu vo vim-e, ovládame aj 'vi'. A ten je možno nájsť na takmer každom "unix like" os. Medzi jeho plusy tiež patrí dokumentácia, vysoká konfigurovateľnosť a efektívne spracovávanie a editovanie súborov. Možno nám bude v prvých chvíľach pripadať "iný", najmä pre užívateľov systému Windows, ktorí sú zvyknutí používať smerové šípky pre pohyb a ešte nepostrehli, že je to pomalšie. VIM nie je žiadne IDE, väčšinu aplikácií, ktoré používame, musíme volať externe.

¹poznámka: rozšírená verzia prúdového editoru ed

Rovnako môžeme samozrejme spúšťať z aplikácii editor vim, napríklad pri písaní e-mailov, editovaní crontab-u a pri nespočetnom množstve ďalších programov.

1.3. Funkcionalita

Z funkcionality programu možno vyzdvihnúť najmä:

- niekoľko režimov práce
- makrá, značky
- automatické dopĺňovanie slov
- zvýrazňovanie syntaxe a formátovanie textu pre desiatky jazykov
- okná
- kontrola pravopisu
- konfiguračné súbory
- zotavenie po páde
- porovnávanie súborov (vimdiff)
- regulárne výrazy
- ex mode

1.4. Inštalácia Vi IMproved

1.4.1. Debian GNU/Linux

Spustíme ako superužívateľ root
apt-get -y install vim

1.4.2. MS Windows

Zo stránky <http://www.vim.org/download.php> si stiahneme samorozbalovací balík, ktorý nainštalujeme.

2. Prvé kroky

2.1. Ako začať

Prvým krokom by malo byť spustenie 'vimtutor'. Ide o akýsi textový súbor, ktorý sa otvorí v našom budúcom obľúbenom editore, s počiatočnými radami, ako postupovať. Neskôr uvidíme, editor má všetku dokumentáciu v tejto forme. Buď išlo o lenivosť autorov niečo nové vymýšľať, alebo je to ich ďalší geniálny nápad. Zvládnutie tohoto tutorialu nám zaberie maximálne 30 minút času, po ktorom už môžeme smelo editovať. Po osvojení si základných príkazov (tým, že budeme VIM používať), môžeme pristúpiť k ďalšej fáze, a tou je samotná dokumentácia. Otvoríme si ju stla-

čením klávesy F1. Je logicky rozdelená na niekoľké sekcie, prvá opisuje počiatkové kroky, základné módy a minimálne zmeny v konfigurácii. Neskôr sa dostaneme k písaniu funkcií, rýchlejšiemu vyhľadávaniu a dávkovému spracovaniu textov a pochopíme, že všetko, čo sme doteraz robili, ide spraviť oveľa jednoduchšie. Celkovo môžeme nájsť okrem toho aj referenčnú príručku, rady pre programátorov a rôzne ďalšie veľmi užitočné poznámky.

2.2. Spustenie

VIM sa v závislosti od operačného systému spúšťa kliknutím na ikonu vim, prípadne príkazom *vim* z príkazového riadku. Príkazy, ktoré v nom zadávame, sa delia do dvoch skupín.

Prvou z nich sú tie, ktoré zadávame do príkazového riadku vim, ktorý otvoríme znakom ':' (dvojbodka) a potvrdíme klávesou enter. Radi by sme ešte poznamenali, že niektoré príkazy (napríklad ukončenie editoru pred uložením zmien) vyžadujú explicitne uviesť ! za príkazom. Znamená to, že sme si svojou voľbou istí a žiadame príkaz naozaj vykonať. Taktiež stačí uvádzať len niektoré prvé písmená príkazov, aby boli jednoznačné, prípadne sa vyberie ten, ktorý editor považuje za najčastejšie používaný. Napríklad, ak namiesto *:edit* uvedieme len *:e*, bude to znamenať to isté (napriek tomu, že by to mohlo byť aj echo, earlier ...).

Druhá skupina je špecifická pre každý z režimov, v ktorom budeme pracovať (pohyb, práca s textom, blokmi). Dvojbodka sa pred ne neuvádza a väčšinou ide o jeden až troj-písmenkovú kombináciu znakov.

Predtým ešte niečo o tom, čo sa deje hneď po spustení. Ako prvé sa prehľadávajú súbory, ktoré nám vypíše príkaz `:scriptnames`. Jedným z nich je aj súbor `~/vimrc` (v OS Windows sa tento adresár nenachádza v `$HOME`, cesta je uvedená v dokumentácii), ktorý obsahuje našu konfiguráciu. Dostaneme sa k nemu ešte neskôr. Pretože môže byť nastavenie pozmenené na rôznych systémoch a distribúciách, odporúčame po prvýkrát pridať k príkazu `vim` aj parametre `-u NONE -U NONE`, ktoré zabránia načítaniu pluginov a konfigurácie.

2.3. Módy editora Vi IMproved

Ako sa môžeme dočítať v manuálovej stránke, sú rozličné spôsoby jeho spustenia. Tým prvým je *ex režim*, ktorý je tam kvôli spätnej kompatibilite z historických dôvodov. *View* (vi s nastaveným flagom, aby neumožňoval meniť súbory), *rvim* (restricted, neumožňuje spúšťať procesy a má hlavne bezpečnostné obmedzenia) a grafická verzia editoru vim, *gvim*. Ten obsahuje navyše klikateľné menu s niektorými funkciami, ktoré sú v bežnom vime založené na cli.²

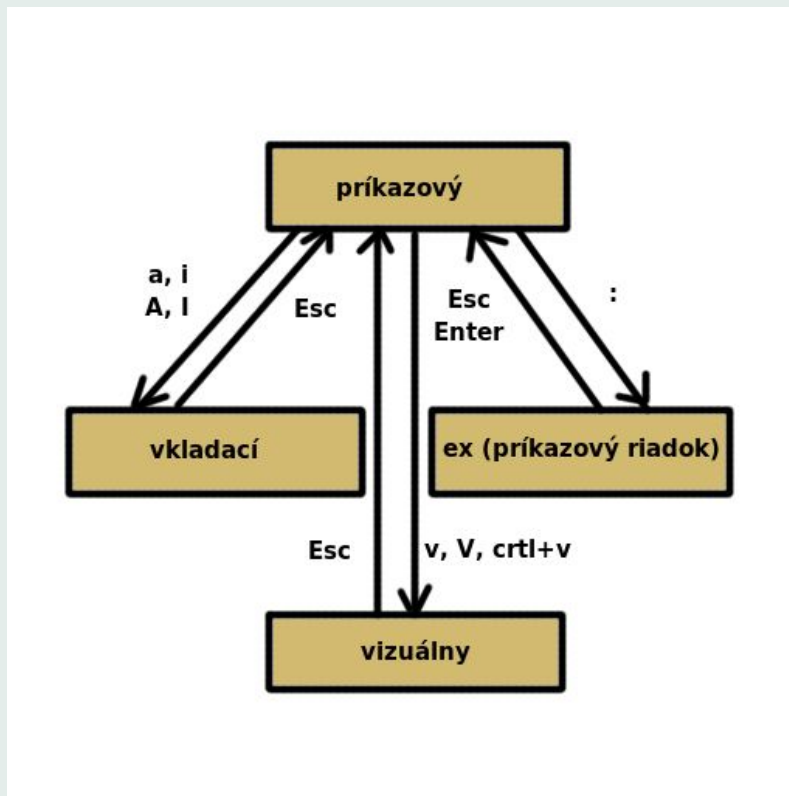
Najvýznamnejšou odlišnosťou od ostatných editorov je niekoľko režimov fungovania.

2.3.1. Normal mode (normálny režim)

Základný, v ktorom sa ocitnete po spustení editoru. Umožňuje označovať a vkladať text, mazať a podobne. Niektoré z príkazov sú tiež určené na prechod do iného

²command line interpreter, alebo príkazový riadok

režimu - vid' Obrázok 1.



Obrázok 1: Diagram znázorňujúci vzájomné väzby a prechody režimov

[Domovská stránka](#)

[Print](#)

[Titulná strana](#)

[Obsah](#)

◀

▶

◀

▶

Strana 12 z 28

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Z ľubovoľného režimu sa do normal dostanete stlačením klávesy esc (niekedy 2x, kvôli zrušeniu predošlého príkazu).

2.3.2. Insert Mode (vkladací režim)

Slúži na písanie samotného textu, mód sa aktivuje stlačením niektorej z kláves a, i, A, I, o, O, R s rôznymi účinkami. Ich popis bude vysvetlený nižšie.

2.3.3. CLI Mode

Sem sa zadávajú samotné príkazy, konfigurácia, poskytuje vyhľadávanie, funguje dopĺňovanie názvov tabulátorom.

2.3.4. ex mode

Jeho funkcionalita je mimo rozsahu tejto príručky, v súčasnosti sa takmer nepoužíva.

2.3.5. Visual mode

Hromadné označovanie textu a práca s blokmi. Často nám umožňuje efektívnejšie riešiť niektoré úlohy, keďže si môžeme vybrať ľubovoľný úsek textu a aplikovať naň takmer ktorýkoľvek z príkazov, ktoré fungujú pre normal mode.

[Domovská stránka](#)

[Print](#)

[Titulná strana](#)

[Obsah](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

[Strana 13 z 28](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



Na jeho aktiváciu slúžia klávesy v, V alebo ctrl+v.

3. Práca s VIM

3.1. Základné pohyby

Pre tých, ktorí si prečítali vimtutor, nebude v najbližších riadkoch nič nové, odporúčame pokračovať touto sekciou. Editor vim disponuje nasledujúcimi klávesami pre pohyb:

h	dolava
j	dole
k	hore
l	doprava

3.2. Tags

Pokiaľ ste sa už dostali k help-u (F1), určite si všimnete, že niektoré slová sú vyznačené hrubším písmom. Niekedy bývajú ohraňované symbolmi |, napríklad | takto |. Ide o akési menovky (labels) alebo, ako sa v dokumentácii nazývajú, tagy. Možno nimi prechádzať, a to nasledovne:

ctrl-]	dopredu
ctrl-o alebo ctrl-t	naspät

Ide o veľmi užitočnú vec aj pre programátorov, napríklad v jazyku C môžeme použiť externý program `ctags` (najčastejšie s parametrom `-R`), ktorý vygeneruje súbor `ctags` a umožňuje jednoduché vyhľadávanie deklarácií funkcií, premenných atď. s použitím vyššie uvedených kláves.

3.3. Prvý dokument

Napíšeme v príkazovom riadku `vim subor.txt`. Objaví sa editor s novým súborom, ktorého meno je prvý argument. Znak `~` označujú riadky, ktoré ešte nie sú vytvorené (tým, že je dokument prázdny, okrem prvého riadku, na ktorom sa nachádza kurzor, obsahuje dokument len `~`). Sme v režime `normal`.

3.4. Príkazy

Nasledovné príkazy, začínajúce dvojbodkou sú pre `cli`, ostatné sa vzťahujú pre `normal` režim, alebo iný, ktorý uvedieme.

Parametre v hranatých `[]` zátvorkách su nepovinné, v kučeravých `{ }` povinné.

3.4.1. Príkazy na editovanie súboru

:e[dit]	Edituje aktuálne otvorný súbor. Použiteľné najmä v prípade, že sa dokument zmenil mimo vim-u
:e[dit]!	”Zahodí” všetky zmeny a otvorí dokument ešte raz v pôvodnej verzii
:e[dit] {súbor}	Otvorí 'súbor', ktorý zadáme ako parameter, na editáciu
:e[dit]! {súbor}	To isté ako vyššie, avšak zahodí zmeny v otvorenom neuloženom dokumente
gf	Edituje súbor, ktorého názov sa nachádza na mieste kurzoru. (Mnemotechnická pomôcka – goto file)

3.4.2. Vkladanie textu

a	(append), insert mode, kurzor bude za znakom, na ktorom sa nachádzame
A	(append), to isté, ale kurzor umiestni na konci riadka
i	(insert), insert mode, kurzor pred znakom, na ktorom sa nachádzame
I	(insert), to isté, ale kurzor umiestni na začiatku riadka
gI	insert mode, vloží text na stĺpec číslo 1
o	(open) insert mode, vloží nový riadok pod kurzorom
O	(open) insert mode, to isté, nad kurzorom

3.4.3. Vkladanie súboru

:r[ead] [meno]	vloží subor s menom 'meno', alebo aktuálne editovaný
:r[ead] {cmd}	Spustí príkaz 'cmd' (čo môže byť externý skript, program) a vloží z neho výstup pod kurzor

3.5. Mazanie textu

x, delete	zmaže znak pod kurzorom
X	zmaže znak pred kurzorom
d{smer}	zmaže riadok alebo znak text v udanom smere. Možno tiež použiť w alebo e na zmazanie slova alebo d na celý riadok
D	zmaže všetky znaky za kurzorom až po koniec riadku
{visual}x alebo d	zmaže označený text
{visual}x alebo D	to isté, ale až do konca riadkov
:[rozsah]d[elete]	zmaže rozsah riadkov, bez udania rozsahu sa použije aktuálny riadok. Príklad :5,10d

3.6. Zmena, nahradenie textu

r{znak}	Nahradí znak pod kurzorom znakom 'znak'
R	Špeciálny insert mode, v ktorom sa prepisujú znaky
~	Zmení veľké písmená na malé a naopak
{visual}~	To isté, v bloku
J	Spojí dva riadky dokopy

3.7. Kopírovanie a pohyb v texte

”{a-zA-Z0-9.%#:-}”	Použije register z danej množiny pre nové označenie, vloženie
:reg[isters]	Zobrazí zoznam všetkých registrov
:reg[isters] {meno}	Zobrazí register, s menom 'meno'
[”x]y{smer}	Uloží do registra x znak v smere 'smer'. Akceptuje tiež podobné voľby ako d{smer}.
[”x]yy	Uloží do registra x celý riadok
[”x]Y	To isté
{visual}[”x]y	Uloží do registra x označený blok
:{rozsah]y[ank][x]	Uloží riadky [v uvedenom rozsahu] [do registra x]
[”x]p	Vlož text [z registra x] za kurzor
[”x]P	Vlož text [z registra x] pred kurzor
[”x]gp	To isté, ale umiestni kurzor za nový text.
[”x]gP	To isté, pred nový text

Ako sme ste mohli všimnúť, vim umožňuje vkladať text do registrov, ktoré môžu byť pomenované z množiny {a-zA-Z0-9}.

Napríklad, na označenie riadku a následné vloženie môžeme použiť implicitný register stlačením 'yy p' alebo, s použitím registra 'a' "ayy" ap.

3.8. Späť/Zopakuj

u	Vráť sa o krok naspäť
:u[ndo]	To isté
CTRL-R	Zopakuj vrátený krok
:red[o]	To isté
U	Vráť všetky zmeny na riadku, kde sa nachádza kurzor
.	Zopakuje posledný krok

3.9. Ukončenie



:q[uit]	Ukončí vim. Tento príkaz zlyhá, ak sú v dokumente neuložené zmeny
:q[uit]!	Ukončenie bez zapísania zmien v dokumente
:cq[uit]	To isté
:wq {súbor}	Zapíše zmeny do súboru a ukončí sa, ak je otvorený len jeden súbor na editovanie. V prípade uvedenia súboru sa zapíšu zmeny do neho.
:wq!	To isté, aj pri viacerých súboroch
ZZ	Zapíše zmeny, ak nastali, a ukončí sa.
ZQ	To isté ako :q!

3.10. Pohyby medzi slovami

w	o slovo ďalej, skončí na prvom písmene
b	o slovo späť, skončí na prvom písmene
e	o slovo ďalej, skončí na poslednom písmene
ge	o slovo späť, skončí na poslednom písmene

3.11. Pohyby na riadku

\$	kurzor na koniec riadka
^	kurzor na prvý nebiely znak
0	kurzor na prvý znak na riadku

3.12. Presun na určitý znak

f	skok na prvé písmeno uvedené za f
F	opačné vyhľadávanie
tT	obdobne ako fF, avšak zastane 1 znak predtým
;	vyhľadávanie podľa predošlých kritérií, dopredu/dozadu
%	nájde zodpovedajúcu zátvorku

3.13. Presun na určitý riadok

gg	prvý riadok v texte
G	posledný riadok v texte
<num>%	skočí na miesto % num v dokumente
<num>G	skočí na určený riadok v dokumente
H	od slova 'home', kurzor hore napravo
M	od slova 'middle', kurzor v strede
L	od slova 'last', kurzor v poslednom riadku

4. Číslovanie (count)

Predstavme si situáciu, keď potrebujeme nejaký úkon zopakovať niekoľkokrát. Pohyb o osem riadkov nadol, vloženie troch otáznikov za vetou, zmazanie piatich znakov. VIM umožňuje pred príkazmi tiež špecifikovať, koľkokrát sa má daná postupnosť

príkazov vykonať. Číslovanie (COUNT) možno uviesť takmer všade, kde sa nám to bude zdať logické.

Príklady:

Označenie 7 riadkov v texte, od aktuálneho kurzoru: 7yy

Pohyb o 20 riadkov nahor, o 5 napravo: 20k 5l

Zmazanie 10 riadkov: 10dd

5. Jednoduché vyhľadávanie

/reťazec	hľadá reťazec v texte, zhora nadol
?reťazec	hľadá reťazec v texte, spätne

Niektoré neštandardné znaky, ako napríklad zátvorky, sa používajú aj v špeciálnych prípadoch, takže je potrebné pred ne uviesť znak \.

Taktiež môžeme použiť niektoré z nastavení, t.j. set hlsearch, set ignorecase, set incsearch. Príklady použitia sú v help-e.

6. Značky (marks)

Značky sa používajú pri presune z rôznych miest a dokonca v rôznych dokumentoch behom stlačenia 2 kláves. Nastavujú sa príkazom `m{a-zA-Z}`, ďalší znak slúži na jednostrannú identifikáciu. Po nastavení značiek sa na ne presunieme stlačením klávesy `'` (apostrof). Existuje taktiež implicitná značka, `'`, ktorá nastaví pri skoku medzi dokumentami, prípadne sa uloží posledný stav pred naším skokom. Na prechod medzi dvoma poslednými miestami z toho dôvodu funguje príkaz `''` (dvakrát apostrof).

7. Konfigurácia, vimrc

Aby sme mohli používať niektoré nastavenia bez opätovného písania pri spustení každej novej inštancii vim, používa sa súbor s nastaveniami. Tie bývajú v rovnakom formáte, ako pri písaní v editore, až na znak `:`, ktorý sa v súbore už nenachádza. Globálny konfiguračný súbor pre operačný systém Linux je v adresári `/etc/vimrc`. Odporúčame zmeniť si užívateľský, ku ktorému sa dostaneme príkazom `vim ~/.vimrc`. Ako príklad uvedieme niektoré často používané nastavenia, pričom pri väčšine volieb stačí uviesť prefix `'no'`, aby sme dosiahli opačný efekt.

set nocompatible - Vypne vi-kompatibilný mód, takmer nutnosť. Bez tejto možnosti je história vykonaných zmien v dokumente zúžená na posledný jeden krok.

language sk_SK.UTF-8 - Jazykové nastavenie.

syntax on - Farebné syntaktické zvýrazňovanie.

nobackup - Nevytvára backup ~ súbory.

history - Uchováva posledných 500 riadkov pri histórii príkazového riadku.

set background=dark - Tmavé pozadie, pri svetlých farbách sa používa light.

set backspace=indent,eol,start - Umožňuje použiť backspace pri novom riadku

set confirm - Pri "nerozhodnosti" navrhne riešenie.

set ruler - Pravítko, zobrazuje pozíciu kurzora po celý čas.

set showcmd - Zobrazuje ešte nedokončené príkazy.

set showmode - Vľavo dole zobrazí aktuálny režim.

set showmatch - Ak zapíšeme pravú zátvorku, ukáže k nej príslušnú ľavú.

set ignorecase - Nerozlišuje veľkosť písmen, napríklad pri hľadaní.

set smartcase - Len pri malých písmenkách ignoruje veľkosť.

set incsearch - Inkrementálne vyhľadávanie.

set nohlsearch - Nezafarbujú vyhľadávané časti.

set wildmenu - Menu pri TAB-e pri dokončovaní príkazov.

set nonumber - Vypne číslovanie riadkov.

set pastetoggle - Prepínanie z/do 'paste' módu. Pri kopírovaní textu do vim nastávajú problémy, pokiaľ je zapnutý indent a mod voľba :set nopaste je nastavená. Keďže nám však so :set paste nefunguje mapovanie a niektoré ďalšie podobné voľby, býva najčastejšie zapnuté len počas krátkej doby, medzi kopírovaním a opätovným písaním.

set wrap - Rozdelí dlhý riadok (v editore, nie v súbore).

set showbreak=+ - Dlhé riadky rozlíšime s +.

noremap j gj - Bez tejto voľby, t.j. namapovanie klávesy gj na j, by sme preskakovali nerozdelené dlhé riadky.

noremap k gk - To isté, nahor.

set nolinebreak - Rozdeľuje slová ľubovoľne, nie podľa whitespace.

set textwidth=0 - Neobmedzená dĺžka pravého okraja.

set smartindent - Jemne sofistikovanejšie zarovnávanie.

set scrolloff=0 - Samovoľný posun textu pri pohybe, vypnuté.

set noexpandtab - Nenahrádza [tab] medzerami.

set tabstop=4 - Veľkosť tabulátora.



8. Makrá

Notácia definovania je veľmi podobná ako pri značkách. Umožňujú "nahrať" ako záznam ľubovoľnú kombináciu stlačenia kláves a v prípade potreby ju prehrať (užitočné najmä vďaka spomínanému count-u). Syntax používania je $q\{a-zA-Z\}$. Záznam opätovne spustíme stlačením $@\{a-zA-Z\}$.

Príklad:

Potrebujeme odobrať z každého riadku prvé slovo a pridať na koniec riadku.

Uloženie makra: `qadw$p0jq`

Aplikovanie makra na 10 riadkov: `10@a`

9. Jednoduché regulárne výrazy

$:[\text{rozsah}]s[\text{ubstitute}]/\text{výraz}/\text{reťazec}/[c][e][g][p][r][i][I]$

Každý riadok $[v \text{ udanom rozsahu}]$, ktorý vyhovuje výrazu, nahradí reťazcom. Rozsahu môže byť nasledovný:

x,y	Riadky medzi číslami x, y
.	Aktuálny riadok
\$	Posledný riadok v súbore
%	Celý súbor. To isté ako 1,\$
'a	Pozícia značky a

Môžeme použiť nasledovné modifikátory:

c	VIM sa spýta na každú zmenu, ktorú musíme následne potvrdiť
e	Pokiaľ sa vzor nenašiel, nehlási chybu
g	Nahradí na riadku všetky výskyty reťazcom, nielen prvý najdený
i	Ignoruje veľkosti písmien vo vzore
I	Neignoruje veľkosť písmien, štandardné správanie
p	Vypíše riadok obsahujúci poslednú substitúciu

Príklad:

*Nahradenie všetkých výskytov slova 'first' slovom 'second', zmeny chceme potvrdzo-
vať:*

```
:% s/first/second/Icgp
```

10. Záver

Kedže táto príručka bola určená pre začiatočníkov, naším cieľom nebolo ukázať maximálny výkon, akým VIM disponuje, ale skôr vytvorenie sprievodcu jeho prvými krokmi. Dúfame, že svoj účel splnil.

[Domovská stránka](#)

[Print](#)

[Titulná strana](#)

[Obsah](#)



Strana **28** z **28**

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

11. Užitočné odkazy

Slovenské a české odkazy:

1. Český tutorial od Pavla Satrapu
<http://www.kit.vslib.cz/~satrapa/docs/vim/>
2. Pavel Satrapa, konfigurácia
<http://www.root.cz/clanky/jak-si-prizpusobit-vim/>
3. Článok o pluginoch
<http://www.root.cz/clanky/nej-vim-plugins/>
4. Slovenská klávesnica pre VIM
http://www.pv2c.sk/free/sk_vimkeymap/
5. Príklad na konfiguračný súbor
<http://people.ksp.sk/~misof/programy/vimrc.html>

[Domovská stránka](#)

[Print](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Strana 29 z 28

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Zahraničné odkazy:

1. Vim, regulárne výrazy
<http://www.geocities.com/volontir/>
2. Vim, triky
<http://www.zinkwazi.com/unix/notes/tricks.vim.html>
3. Vim ctags
<http://www.eecs.harvard.edu/mdw/course/cs161/handouts/ctags.html>