Textový editor Vim

http://www.vim.org

Mgr. Ľubomír Host

Platon Software Development Group

<rajo AT Platon.SK>

Osnova prednášky

- trochu teórie na úvod
- ako začať písať v editore
  . . . a ako editor ukončiť ☺
- efektívnejšie používanie editora Vim
- prehľad najlepších vlastností editora
- programové rozšírenia – pluginy
- to najlepšie na záver: pripojenie do databázy textovým editorom (predstavenie vlastného pluginu **database-client.vim**)

Použité značky

- príkaz zadávaný v interprete príkazov (shelli):

```
vim slides.tex
```
shell

- príkaz zadávaný v editore Vim:

```
:set hlsearch
```
Vim

- klávesy, ktoré treba „postláčať“: `5dd`

Projekt `vimconfig`

- Dostupný na adrese:
  `http://Platon.SK/projects/vimconfig`
- Jednoduchá inštalácia:

```
tar xvzf vimconfig-1.9.tar.gz
cd vimconfig-1.9
make install
```

shell

- Získanie aktuálnej verzie priamo z CVS:
  `http://Platon.SK/article.php?cvs::instructions`

```
less vimconfig-1.9/README
```

shell

Módy editora Vim

- 6 základných módov

1. **Normal mode** – po normálnom spustení editora
2. **Visual mode** – po stlačení kláves `v` `V` `CTRL-V`
3. **Select mode** – po stlačení kláves `v` `V` `CTRL-V`, záleží od nastavenia voľby `'selectmode'`
4. **Insert mode** – po stlačení kláves `i` `I` `a` `A` `c` `C` `s` `S`
5. **Command-line mode** – po stlačení kláves `:` `?` `/` `!`
6. **Ex mode** – po stlačení klávesy `Q`

- 5 rozšírených módov

1. **Operator-pending mode** – po stlačení príkazovej klávesy (napr. `d` `c` `y`) v režime **Normal**
2. **Replace mode** – špeciálny typ **Insert** módu
3. **Insert Normal mode** – po stlačení klávesy `CTRL-O` v **Insert** móde
4. **Insert Visual mode** – po stlačení klávesy `CTRL-O` + `v` `V` `CTRL-V` v **Insert** móde
5. **Insert Select mode** – po ťahaní myšou v **Insert** móde

Ako začať...

- spustenie editora

```
vim prednaska.tex
```
shell

- ukončenie editora bez uloženia zmien

```
<Esc>:qa!
```
Vim

- uloženie zmien v súbore a ukončenie editora

```
:wq
```
Vim

- začať písať možno po stlačení klávesy `i`, po napísaní textu stlačiť `<Esc>:wq`
- získať pomoc k danému kľúčovému slovu možno príkazom

```
:help
:help motion.txt
:help getwinposx()
:help CTRL-W_P
:help BufWritePost
:help 'nocompatible'
```

Vim

- pri prázdnej konfigurácií textového editora použiť príkazy:

```
:set nocompatible
:syntax on
```

Vim

Konfigurácia

- nastavenia možno ukladať do súboru `~/.vimrc`, prípadne do `~/.gvimrc`
- zásuvné moduly (pluginy) – `~/.vim/plugin/`
- nastavenia podľa typu súboru – `~/.vim/ftplugin/`
- projekt `vimconfig` má konfiguráciu uloženú v CVS, preto bol vyvinutý systém, ako mať rôzne nastavenia na rôznych serveroch pre rôznych užívateľov rozdelenú do viacerých súborov.
  Viď neskôr plugin **local/vimrc.vim**

Pohyb kurzora v texte

- klasické klávesy – h j k l
- smerové šípky – <Left> <Right> <Up> <Down>
- pohyb po stránkach – <PageUp> <PageDown>
- na začiatok súboru – gg 1G
- na koniec súboru – G
- na dané číslo riadku – 135G
- na začiatok riadku – 0
- na koniec riadku – $
- na prvý neprázdny znak riadku – ^
- pohyb po slovách – w W e E

- skok **dopredu** na dané písmenko v aktuálnom riadku – `fb` `tb`
- skok **dozadu** na dané písmenko v aktuálnom riadku – `Fb` `Tb`
- pohyb po vetách v odstavci – `(` `)`
- ďalšie možnosti pohybu v texte

```
:help motion.txt
```

Využívanie registrov

- bežné editory majú iba jeden register schránky (clipboard)
- Vim ich má **48**, rozdelených na **9** typov

1. bezmenný register `""`
2. 10 číslovaných registrov `"0` to `"9`
3. malý mazací register `"-`
4. 26 pomenovaných registrov `"a` až `"z` alebo `"A` až `"Z`
5. 4 registre iba na čítanie `":`, `".`, `"%` a `"#`
6. výrazový register `"=`
7. označovacie a mazacie registre `"*`, `"+` a `"~`
8. register „čierna diera“ `"_`
9. register s vyhľadávacím vzorom `"/`

Operácie s registrami

- prezeranie registrov – `:reg`
- vloženie textu do registra – `y` `Y` `"aY`
- opätovné vloženie textu z registra – `p` `P` `"xp`
- nahratie kombinácie kláves do registra – `qa` ... klávesy ... `q`
- vykonanie registra – `@a`

```
:help yank
:help put
:help registers
:help recording
```

Vim

Úloha: v súbore sú uložené dátové záznamy, z ktorých každý má 5 riadkov. Preveďte tieto záznamy do jednoriadkového formátu.

- zmazať prvé slovo, spojiť s druhým riadkom, na treťom riadku vymazať posledné slovo, . . . , ďalší záznam

Riešenie: zapamätám si klávesy do registra a ten nechám vykonať 1000 krát!

Postup:

- zapnem nahrávací mód klávesou `q` a vyberiem si register `a`
- vykonám úpravy na prvom zázname a klávesou `q` ukončím nahrávací mód
- na druhom zázname už len vykonám tie isté úpravy vyvolaním registra – `@a`, prípadne `998@a`

Editovanie viacerých súborov naraz

- Vim štandardne otvorí iba jedno okno

```
vim x.sql test.txt index.html
```

shell

```
:next
:previous
```

Vim

- otvorenie ďalšieho okna

```
:split test.txt
:vsplit index.html
```

Vim

- prepínananie medzi oknami – `<CTRL-W><CTRL-W>` `<CTRL-W>p` `3<CTRL-W>`

Zlučovanie zmien vo viacerých verziách súboru

- máme dve (alebo tri) rozličné verzie zdrojového kódu a samozrejme nepoužívame CVS ☺

```
diff -u verzia1.c verzia2.c
```

shell

- spustíme **vimdiff** a súbory zeditujeme:

```
vimdiff verzia1.c verzia2.c
```

shell

Sprehľadňovanie programov skrývaním kódu – folding

- výhoda nekonečného papiera – časť možno zložiť do harmoniky (vytvoriť záhyb) a na chvíľu skryť
- telá funkcií vo zvláštnych záhyboch, zložité zanorenia a cykly vo zvláštnych záhyboch
- rôzne metódy vytvárania záhybov: **manuálna**, **odsadzovacia**, **výrazová**, **značkovacia**, **syntaxová**, **rozdielová**
- podrobnejšie informácie:

```
:help folding
```

Vim

- nastavenie metódy skrývania

```
:set foldmethod=marker
:set foldmethod=manual
:set foldcolumn=3
```

Vim

- štandardné značky `{{{` a `}}}`
- otvorenie záhybu – `zo`
- zatvorenie záhybu – `zc`
- vytvorenie záhybu – `zf`
- zrušenie záhybu – `zd`

Automaticky spúšťané príkazy

- **autocommands**

```
:help autocmd
```
Vim

- po uložení konfiguračného súboru `httpd.conf` spraviť reštart webservera:

```
vim /etc/apache/httpd.conf
```
shell

```
:au BufWritePost *.conf :wall
:au BufWritePost *.conf
    :!/etc/init.d/apache reload
```
Vim

- vysádzať LaTeXovský dokument po uložení súboru:

```
:au BufWritePost *.tex :wall
:au BufWritePost *.tex :!make
```

Vim

- automaticky načítať uloženú konfiguráciu Vimu:

```
:au BufWritePost ~/.vimrc
    :source ~/.vimrc
```

Vim

Debugger regulárnych výrazov

- chcem sa to konečne naučiť!
- vôbec mi to nefunguje . . .
- môj problém je naozaj **zložitý**

```
vim /var/log/httpd-access.log
```

shell

```
:set nowrap
:set hlsearch
:set incsearch
/^\([^ ]*\).*\[09\/Feb\/.* "GET \([^ ]*\)
```

Vim

Rozšírenia textového editora (pluginy)

- možnosť doprogramovať si takmer ľubovoľnú vlastnosť v skriptovacom jazyku **Vimscript** - konfiguračné súbory, syntaxové súbory
- možnosť doprogramovať rozšírenia v inom skriptovacom jazyku: **Perl, Python, Ruby, Tcl**

Pluginy použité vo `vimconfigu`

- **taglist.vim** – `:Tlist`
  – zoznam funkcií a premenných v aktuálne editovanom súbore
- **calendar.vim** – `:Calendar`
  – kalendár s možnosťou písať si poznámky pre daný deň
- **increment.vim** – `:Inc`
  – vytváranie stĺpčekov s rastúcou postupnosťou čísel
- **templatefile.vim**
  – načítanie šablóny do novo vytváraného súboru, vytvorenie hlavičky s menom autora, aktuálnym dátumom, . . .
- **local/vimrc.vim**
  – sada konfiguračných súborov Vimu pre rôznych užívateľov
  – predefinovanie globálnych nastavení podľa požiadaviek užívateľa

- **diacritics.vim**
  písanie textu s diakritikou v požadovanom kódovaní bez ohľadu na použitú klávesnicu, typ terminálu, grafické prostredie, operačný systém, ...

Modul database-client.vim

Popis: priame pripojenie do databázy textovým editorom Vim

Implementácia: Vimscript, Perl, SQL

Cieľ: vytvoriť rozhranie, ktoré spojí editačné schopnosti textového editora s organizáčnými schopnosťami databázy

Situácia: **neuspokojivá**, málo prenositeľných použiteľných nástrojov na správu databázy: MySQL-Front, TOra, mysql-navigator, textový mysql klient

Stav vývoja

- pripojenie do databázy a odpojenie
- viacero simultálnych spojení
- vytvorenie viacerých okien pre prácu
- základy užívateľského menu
- zistenie zoznamu databáz
- vykonanie SQL príkazu
- získanie dát z databázy
- zaznamenanie vykonaného príkazu
- základné zvýrazňovanie syntaxe

TODO

- Dokumentácia!!!
- **Dokumentácia!!!**
- **Dokumentácia!!!**
- dokončiť používateľské rozhranie
- zlepšiť zobrazovanie údajov z databázy
- implementovať inteligentné rozhranie na pridávanie a aktualizovanie záznamov v databáze
- Spolupracovníci sú **vítaní . . .**

Záver

Ďakujem za pozornosť

Mgr. Ľubomír Host

`<rajo AT Platon.SK>`

Platon Software Development Group

`http://Platon.SK`

Powered by: Vim, LaTeX, GNU/Linux Debian, MySQL, Perl, `vimconfig`

Zoznam fólií