

Jemný úvod do editoru Vim

Petr Hruška <hruska@popelka.ms.mff.cuni.cz>

Verze 1.11

Tento text vzniknul na základě školení pro občanské sdružení Praha12.net.

Aktuální verze tohoto dokumentu je umístěna na adrese
<http://phru8648.praha12.net/vim-tutorial/>

Obsah

1	Historie editoru vi	3
2	Spuštění editoru vi	3
3	Režimy editoru vi	3
4	Pohyb	4
4.1	Pohyb na řádce	4
4.2	Pohyb po slovech	5
4.3	Pohyb mezi řádky	5
5	Číselné prefixy	5
6	Mazání	5
7	Další operátory	6
8	Přepisování	6
9	Vyhledávání	6
9.1	Vyhledávání písmen na řádce	6
9.2	Vyhledávání slov	7
10	Speciality	7
11	Značky	7
12	Operace s bloky textu	7
12.1	Vkládání naposledy smazaného textu	7
12.2	Označování textu	8
13	Makra	8
14	Konfigurace editoru Vi	8
15	Práce s více okny	9
16	Příkazová řádka	9
17	Závěrem	9

1 Historie editoru vi

Editor *Vi* vzniknul podobně jako ropa. Napřed byli na světě dinosauři, ale potom se přežrali a vymřeli na zácpu. Po několika miliónech let přišli Arabové a začali tu ropu těžit. Snažím se jen objasnit skutečnost, že editor *Vi* vzniknul postupným a dlouhodobým vývojem. Funkce editoru byly přidávány postupně, tak jak je lidé potřebovali. Výsledkem je neobyčejně mocný nástroj pro práci s textem, který umí prakticky cokoli.

Používání editoru *Vi* je životní styl. Pokud komukoliv řeknete, že pravidelně používáte editor *Vi*, můžete se dočkat tří druhů reakcí:

1. dotyčný nebude vědět co to je,
2. dotyčný bude vědět co to je, poodstoupí, pokřížuje se a začne si sypat pepř a sůl přes rameno,
3. dotyčný sám používá *Vi* a začne si o něm s vámi povídat.

Každopádně budete vypadat jako pravý odborník na Unix (a určitě to bude z velké části pravda :-).

2 Spuštění editoru vi

Editor se spouští z příkazové řádky pomocí příkazu *vi*. Existuje vylepšená verze *Vim*, která obsahuje důležité funkce (například práce s více soubory najednou, označování bloků apod.) a je mnohem pohodlnější.

Ve zbytku tutoriálu budeme předpokládat, že čtenář používá *Vim*.

Tip: Vždy používejte variantu *Vim*, přidejte si do souboru `~/.bash_profile` řádek

```
alias vi='vim'
```

Tím způsobíte, že se příkazem *vi* bude spouštět vylepšená verze *Vim*.

Při spuštění *Vi* bez parametrů bude otevřen nový, prázdný, nepojmenovaný soubor.

Chcete-li pomocí editoru editovat již existující soubor, můžete zadat příkaz `vim jmeno_souboru`.

3 Režimy editoru vi

Editor rozlišuje tři základní režimy:

1. normální režim,
2. režim vkládání,
3. režim příkazové řádky.

Po spuštění je editor v normálním režimu. V tomto režimu funguje *Vim* tak, jak by obyčejný uživatel nikdy nečekal – stisk klávesy nevloží příslušný znak, ale provede příkaz. Nejjednodušší příkaz je jen jedno písmenko, například *x* vymaže

znak pod kurzorem. Jiný příklad je dvojice písmen **dd**, která smaže řádek, na kterém právě stojí kurzor.

Stisknutím klávesy **[Esc]** se vždy dostanete do normálního režimu.¹

Do režimu vkládání se můžete přepnout pomocí příkazů **i**, **I**, **a**, **A**, **o**, **O** a dalších. Režim vkládání poznáte podle nápisu `-- INSERT --` v levém dolním rohu obrazovky.²

i	začne vkládání za znak na kterém stojí kurzor (insert)
I	začne vkládání před první neprázdný znak na řádku
a	začne vkládání za znak na kterém stojí kurzor (append)
A	začne vkládání na konci řádku
o	vloží novou řádku pod kurzor, umístí na ní kurzor a přepne do režimu vkládání
O	totéž jako o , ale vloží novou řádku nad kurzor

Režim vkládání ukončíte klávesou **[Esc]**.

Do režimu příkazové řádky se dostanete pomocí příkazu **:**. Po stisknutí znaku **:** skočí kurzor do levého dolního rohu obrazovky a editor čeká na zapsání řádkového příkazu. Například příkaz `:read jmeno_souboru` vloží obsah zadaného souboru do textu. Příkaz `:w` uloží právě editovaný soubor. Příkaz ukončíte klávesou **[Enter]**, případně **[Esc]** (v tom případě se nic neprovede).

Po stisku dvojtečky můžete šipkou nahoru procházet předchozí příkazy.

4 Pohyb

Ve *Vim* fungují klasické klávesy pro pohyb, tj. šipky, **[PgUp]**, **[PgDown]**, **[Home]** a **[End]**. Tyto klávesy v dobách vzniku *Vi* neexistovaly, takže k nim existují písmenkové ekvivalenty. Ve *Vi* se dá bez problémů editovat, i když terminál chybně interpretuje kurzorové klávesy apod. Kromě toho se lze v souboru pohybovat mnoha dalšími způsoby.

4.1 Pohyb na řádce

0 (nula)	první znak na řádce
^ , [Home]	první neprázdný znak na řádce (prázdný znak je například tabulátor či mezera)
\$, [End]	poslední znak na řádce
gm	prostředek okna (middle)
g_	poslední neprázdný znak na řádce
7 	sedmý sloupec okna

¹ Výjimkou je **Ex mode**. Viz příkaz **Q**.

² Zobrazení nápisu je ve *Vim* nastavitelné a je nastaveno jako výchozí hodnota. Ve *Vi* se nápis nezobrazuje vůbec.

4.2 Pohyb po slovech

w	začátek dalšího slova
W	jako w , ale některé znaky (například „-“) nepovažuje za oddělovače slov
b	nejbližší začátek slova nalevo od kurzoru
B	jako b , ale některé znaky (například „-“) nepovažuje za oddělovače slov
e	nejbližší konec slova
E	jako e , ale některé znaky (například „-“) nepovažuje za oddělovače slov
ge	nejbližší konec slova nalevo od kurzoru
gE	jako ge , ale některé znaky (například „-“) nepovažuje za oddělovače slov

4.3 Pohyb mezi řádky

h , [Backspace]	vlevo
k	nahoru
j	dolu (připomíná šipku dolů)
l , [Mezerník]	vpravo
-	o řádek výš na první neprázdný znak, ekvivalent k^
+	o řádek níž na první neprázdný znak, ekvivalent j^
G	poslední řádek
gg	první řádek
37%	řádka ve 37% z celkové délky souboru
57G	řádka číslo 57
M	prostřední řádek konzole (Middle)
H	první řádek konzole (Home)
L	poslední řádek konzole (Last)

5 Číselné prefixy

Většinu příkazů je možné doplnit číselnými prefixy, které většinou způsobí opakování příkazu. Například **j** posune kurzor o jednu řádku níž, **3j** posune kurzor o tři řádky níž. **10w** skočí o 10 slov vpřed. Příkaz **x** smaže písmeno pod kurzorem, příkaz **30x** smaže 30 písmen.

V některých případech, pokud opakování nemá význam, způsobí prefix jinou rozumnou změnu chování. Například příkaz **H** umístí kurzor na první řádek konzole, **3H** umístí kurzor na třetí řádek konzole.

Pokud jste nechtěně stiskli nějaké klávesy s číslicemi, můžete je odvolat pomocí klávesy [Esc].

6 Mazání

Mazání se ve *Vi* provádí především pomocí příkazu **d**, který je obvykle následovaný příkazem pro pohyb. Protože příkaz **d** sám o sobě (bez pohybu) nemá

smysl, říká se mu také operátor. Operátor **d** smaže text mezi pozicí kurzoru a místem kam by se kurzor dostal příkazem pohybu.

Příklady:

dG	smaže vše až do konce souboru
dgg	smaže vše od začátku souboru do pozice kurzoru
d3l	smaže tři písmena za kurzorem
dw	smaže slovo

Speciální příkaz **dd** smaže aktuální řádku. Příkaz **5dd** smaže pět řádek.

Příkaz **D** smaže vše od pozice kurzoru až do konce řádky (ekvivalent **d\$**)

Příkaz **x** smaže znak pod kurzorem, příkaz **X** smaže znak před kurzorem.

7 Další operátory

gU	změní písmena na VELKÁ
gu	změní písmena na malá
g~	prostřídá velikost písmen

Například **gU3w** změní tři slova na velká písmena.

Ve *Vi* je obvyklé, že zopakování operátoru zapůsobí na celý aktuální řádek. Například **gugu** nebo **guu** změní všechna písmena na řádce s kurzorem na malá a **gUU** změní všechna písmena na řádce s kurzorem na velká.

8 Přepisování

Příkaz **R** přepne editor do speciálního režimu přepisování, který je funguje podobně jako režim vkládání.

Příkaz **r** následovaný písmenem přepíše znak pod kurzorem.

Operátor **c** (change) následovaný příkazem pohybu nejprve vymaže text, podobně jako **d**, ale potom navíc přepne do režimu vkládání.

Příkaz **cc** smaže aktuální řádku a přepne do režimu vkládání.

Příkaz **C** smaže vše až do konce řádku a přepne do režimu vkládání (ekvivalent **c\$**).

Příkaz **s** smaže jeden znak a přepne editor do režimu vkládání.

Příkaz **S** dělá totéž co příkaz **cc**.

9 Vyhledávání

9.1 Vyhledávání písmen na řádce

f<k>	najde nejbližší znak <k> na řádce a postaví na něj kurzor (find)
F<k>	totéž jako f směrem vlevo
t<k>	najde nejbližší znak <k> na řádce a postaví před něj kurzor (till)
T<k>	najde nejbližší předchozí znak <k> na řádce a postaví za něj kurzor
;	zopakuje poslední f , t , F , T
,	zopakuje poslední f , t , F , T , ale opačným směrem

Příklad: **dt)** smaže všechny znaky až k pravé závorce.

9.2 Vyhledávání slov

Vyhledávání probíhá pomocí příkazové řádky. V tomto případě se však nezačíná dvojtečkou, ale lomítkem nebo otazníkem.

/ahoj	najde nejbližší výskyt slova „ahoj”
?ahoj	funguje jako /, ale hledá směrem vzad
n	zopakuje poslední hledání
N	zopakuje poslední hledání v opačném směru
/	zopakuje poslední hledání, hledá směrem vpřed
?	zopakuje poslední hledání, hledá směrem vzad

Při hledání lze používat kouzelné znaky:

.	libovolný znak
^	začátek řádku
\$	konec řádku

Například /a.\$ najde všechny řádky, které mají předposlední znak „a”.
d/ahoj smaže vše až k prvnímu výskytu slova „ahoj”.

10 Speciality

u U .	undo, vezme zpět poslední editaci vezme zpět všechny poslední změny na řádku zopakuje poslední akci (vkládání, mazání apod.)
K J	spustí manuálovou stránku pro slovo pod kurzorem spojí dva sousední řádky dohromady, první z nich je ten na kterém stojí kurzor
Ctrl+A Ctrl+X	zvýší číslo pod kurzorem o 1 sníží číslo pod kurzorem o 1
ZZ Q	ekvivalent :wq přepne do režimu Ex, podaří-li se vám stisknout Q, napište visual[Enter] pro návrat do normálního režimu.
z.	nastaví kurzor doprostřed obrazovky tak, aby se pozice kurzoru v textu nezměnila (scrolluje) a nastaví kurzor na první neprázdný znak na řádku.

11 Značky

Příkazem **m** se označují pozice souboru. Značí se písmeny **a** až **z** a **A** až **Z**. Na pozici je možno skočit pomocí příkazu ‘ (obrácený apostrof).

Například: **mp** označí pozici kurzoru jako **p**. ‘**p**’ kdykoliv později skočí na pozici **p**.

12 Operace s bloky textu

12.1 Vkládání naposledy smazaného textu

Příkaz **p** (paste) vloží obsah registru. Registr je kus textu. Do registru se automaticky ukládá vše co se maže.

Chceme-li například přesunout slovo, můžeme ho nejprve smazat pomocí **dw** (tím se dostane do registru), potom přemístit kurzor na cílovou pozici a příkazem **p** ho vložit.

Sekvence **ddp** prohodí řádky. Schválně si to zkuste ;-)

Příkaz **P** funguje podobně jako **p**, ale nekládá za kurzor, nýbrž před kurzor.

Chcete-li text jen kopírovat, můžete místo **d** použít příkaz **y** (yunk). **yw** nastaví do registru totéž, co by bylo smazáno příkazem **dw**. Tento text můžete později vložit pomocí **p**. Příkaz **yy** funguje jako **dd**, s tím rozdílem, že nic nemaže. **Y** je ekvivalent **yy**.

12.2 Označování textu

Po stisknutí klávesy **v** se dostanete do režimu označování textu (režim visual). V tomto režimu můžete pohybovat kurzorem jako obvykle, ale je označen text od pozice kurzoru, kde byla stisknuta klávesa **v**, až po aktuální pozici kurzoru. Z režimu označování můžete vyskočit pomocí **[Esc]**.

Stisknete-li v režimu označování **d**, smažete označený text.

Příkaz velké **V** funguje jako malé **v**, ale text je označován po řádkách místo po znacích.

Příkaz **Ctrl+V** funguje podobně jako **v**, ale je označen obdélník. Zkuste si to!

Máte-li označený text, můžete ho vložit na vstup externího programu a na jeho místo vložit jeho výstup napsáním **!:prikaz[Enter]**. Zadáte-li příkaz **!:sort[Enter]**, tak setřídíte označené řádky.

V označeném textu můžete nahradit výskyty libovolného řetězce jiným řetězcem. Napište-li **:s/CoHledat/CoMistoToho/[Enter]**.

13 Makra

Záznam makra se spouští pomocí příkazu **qm**, kde **m** může být libovolné písmeno, které slouží k označení makra. Od spuštění záznamu se zaznamenávají všechny příkazy až do ukončení záznamu příkazem **q** (k ukončení záznamu není potřeba jméno makra).

Zaznamenané makro se spouští příkazem **@m**, který v našem případě spustí všechny příkazy, které byly provedeny uživatelem mezi **qm** a **q**.

Makro lze spouštět několikrát za sebou pomocí číselného prefixu, například **100@m** provede zaznamenanou sekvenci stokrát.

14 Konfigurace editoru Vi

Uživatelská konfigurace je uložena v souboru **~/.vimrc**. Typický obsah je následující:

```
syntax on
set ruler
```


15 Práce s více okny

Editor *Vim* umožňuje pracovat s více okny najednou. V každém okně je otevřený nějaký soubor. Nové okno se souborem se otevírá příkazem `:new jmeno_souboru` a zavírá příkazem `:q`.

Okna se ovládají pomocí klávesové kombinace `[Ctrl]+w`, po které vždy následuje nějaká klávesa. Mezi okny se lze pohybovat příkazy `[Ctrl]+w [šipka]`, kde `[šipka]` označuje libovolnou kurzorovou klávesu.

Okno lze zvětšit pomocí příkazu `[Ctrl]+w +` a zmenšit pomocí `[Ctrl]+w -`. Příkaz `[Ctrl]+w =` nastaví velikost všech oken stejně. Příkaz `[Ctrl]+w _` zvětší okno na maximální možnou velikost.

16 Příkazová řádka

<code>:w</code> <code>:q</code> <code>:q!</code> <code>:wq</code> <code>:wa</code> <code>:x</code>	zapiše aktuální soubor na disk zavře uložený soubor (pokud byl otevřen jediný soubor, tak ukončí editor) zavře soubor, i když nebyl uložený ekvivalent dvou po sobě jdoucích příkazů <code>:w :q</code> uloží všechny otevřené soubory ekvivalent <code>:wq</code>
<code>:new jmeno_souboru</code> <code>:new ++enc=iso-8859-2 jmeno_souboru</code> <code>:vne jmeno_souboru</code> <code>:split</code>	otevře soubor v novém okně otevře soubor se zadaným kódováním jako <code>:new</code> , ale pokusí se nové okno umístit vedle ostatních oken rozdělí soubor do dvou oken
<code>:help příkaz</code> <code>:syntax on</code> <code>:set ruler</code>	nápověda na příkaz zapne zvýraznění syntaxe zobrazí pravítko (údaje o pozici v souboru a režimu (např. -- INSERT --) vpravo dole)
<code>:set fileformat=dos</code> <code>:set fileformat=unix</code>	nastaví, že se má soubor ukládat s dosovskými konci řádků nastaví, že se má soubor ukládat s unixovými konci řádků
<code>:read jmeno_souboru</code> <code>:read !prikaz</code>	vloží obsah zadaného souboru do textu vloží výstup příkazu do textu (například <code>:read !date</code>)
<code>:redo</code> <code>:TOhtml</code>	opět provede to, co bylo odvoláno pomocí u převěd soubor do html včetně zvýrazněné syntaxe

17 Závěrem

Tento krátký manuál vyčerpá celou abecedu příkazů, přesto se zdaleka neblíží k úplnosti. Úplná je (možná) jen dokumentace (viz `:help`).

Síla editoru *Vi* nespočívá v množství příkazů, ale v možnosti je kombinovat navzájem a tedy v důvtipu uživatele. I s malým množstvím příkazů se dají dělat velké věci!

<http://www.vim.org> – Homepage editoru *Vim*.
Za připomínky k textu bude autor velmi vděčný.

Index

+, 5
„, 6
-, 5
— INSERT —, 4, 9
., 7
/, 7
/a.\$, 7
/ahoj, 7
:, 4
 prikaz, 8
 sort, 8
:TOhtml, 9
:help, 9
:help příkaz, 9
:new, 9
:new ++enc=iso-8859-2 jmeno_souboru,
 9
:new jmeno_souboru, 9
:q, 9
:read
 date, 9
 prikaz, 9
:read jmeno_souboru, 4, 9
:redo, 9
:s/CoHledat/CoMistoToho/, 8
:set fileformat=dos, 9
:set fileformat=unix, 9
:set ruler, 9
:split, 9
:syntax on, 9
:vne jmeno_souboru, 9
:w, 4, 9
:wa, 9
:wq, 7, 9
:x, 9
;, 6
?, 7
?ahoj, 7
[Backspace], 5
[Ctrl]+w, 9
[Ctrl]+w +, 9
[Ctrl]+w -, 9
[Ctrl]+w =, 9
[Ctrl]+w [šipka], 9
[Ctrl]+w _, 9
[End], 4
[Enter], 4, 7, 8
[Esc], 4, 5, 8
[Home], 4
[Mezerník], 5
[PgDown], 4
[PgUp], 4
[šipka], 9
\$, 4, 7
~/ .bash_profile, 3
~/ .vimrc, 8
, 7
'p, 7
10w, 5
30x, 5
37%, 5
3H, 5
3j, 5
57G, 5
5dd, 6

0, 4
m, 8

A, 4, 7
a, 4, 7

B, 5
b, 5

C, 6
c, 6
c\$, 6
cc, 6
Ctrl+A, 7
Ctrl+V, 8
Ctrl+X, 7

D, 6
d, 5, 6, 8
d/ahoj, 7
d3l, 6
d\$, 6
dd, 4, 6, 8
ddp, 8
dG, 6
dgg, 6
dt), 6
dw, 6, 8

E, 5
 e, 5
 Ex mode, 4

 F, 6
 f, 6

 G, 5
 g, 6
 g_, 4
 gE, 5
 ge, 5
 gg, 5
 gm, 4
 gU, 6
 gu, 6
 gU3w, 6
 gugu, 6
 gUU, 6
 guu, 6

 H, 5
 h, 5
<http://www.vim.org>, 10

 I, 4
 i, 4

 J, 7
 j, 5

 K, 7
 k, 5

 L, 5
 l, 5

 M, 5
 m, 7, 8
 mp, 7

 N, 7
 n, 7

 O, 4
 o, 4

 P, 8
 p, 7, 8

 Q, 4, 7
 q, 8

 qm, 8

 R, 6
 r, 6

 S, 6
 s, 6

 T, 6
 t, 6

 U, 7
 u, 7, 9

 V, 8
 v, 8
 vi, 3
 vim jmeno_souboru, 3
 visual, 7

 W, 5
 w, 5

 x, 3, 5, 6

 Y, 8
 y, 8
 yw, 8
 yy, 8

 Z, 7
 z, 7
 z., 7
 ZZ, 7