

「詰めViVi」 ～ 最少キーストローク数編集の奥義 ～
14-Nov-2009
Nobuhide Tsuda

vi/ViVi コマンドを使用すると非常に少いキーストローク数でテキストを編集することができます。しかも自然なキー操作なので小指がつったりすることはありません。本セッションでは、最少キーストローク数編集方法をデモを交えて具体的に紹介し、vi/ViVi コマンドの奥義の一端を紹介します。

■ 目次

#目次	内容	分
はじめに	自己紹介, ViVi とは, 近況, ViVi 3.x 紹介	7
詰めViViとは	最少キーストローク数編集の薦め	7
viコマンド	基本 vi コマンド紹介	10
詰めViVi	詰めViVi 問題・回答・解説を時間の許す限り…	35
さいごに	まとめ	1

■ はじめに (7)

テニス、オセロ、その他ゲーム類が趣味の年齢不詳のおじさん
ずーっと ViVi を開発してます。

津田の開発なつぶやき：<http://twitter.com/vivisuke>

津田の開発な日記：<http://vivi.dyndns.org/u/1/last>

ViVi : vi コマンドをサポートする Windows 用テキストエディタ

<http://vivi.dyndns.org/>

ver 2.10 正式版リリース @2009年8月

ver 3.01/02 開発中

「頑固爺開発モデル」

エディタ : ViVi、コンパイラ : VC9

CSVモード、バイナリモード、他

単体テスト

2009年末正式版リリース予定

■ 詰めViViとは (7)

参考文献：[1] 「達人プログラマー」 by Andrew Hunt and David Thomas, 2000
「第3章 16 パワーエディット」より引用：

- ・ひとつのエディターを知りつくしておき、コーディング、ドキュメント作成、メモ、システム管理といった編集作業は、すべてそれを使用して行うことが望ましいでしょう。
- ・← や BackSpace を10回叩いてカーソルを行の左端に移動させるのは、^A または Home あるいは 0 を1回叩くのとは比べ物にならないのです。

#あなたの考え	我々のお勧め
色々なエディターの基本機能だけを使っている。	強力なエディターをひとつ選んで、それに習熟してください。
お気に入りのエディターを使っているが、すべての機能を使っていない。	その機能を学習して、キーを叩く回数を減らすようにしてください。
お気に入りのエディターを使っており、可能な限り使いこなしている。	より優れた使い方を研究し、すでにやっている事以上の効率化を目指してください。
ばかなことを！メモ帳で十分じゃないか。	必要を感じず、生産性も高いのであればそれで結構です。 しかし、今後他のエディターが良いと感じることがあれば考え方を改めてください。

「詰めViVi」

与えられた編集結果を得るための最少ストローク数のコマンド群を求める問題

<http://vivi.dyndns.org/vivi/tsume/tsumevivi.html>

常に最少ストロークにこだわって時間を費やすようなことは本末転倒だが、
様々な編集テクニックを知っておくことは重要

■ vi/ViVi コマンド (10)

vi editor とは

vi (ヴィーアイ) は、Emacsと共にUNIX環境で人気があるテキストエディタ。
 1980年頃、ビル・ジョイによって開発された。名の由来はVisual editorの略。
 Linuxなどに搭載されているのは上位互換のVimやnviであることが多い。

vi の特徴

コマンドモード・挿入モード・上書きモード・コマンド入力 (:/?) モード

対称性 : cdy × 移動コマンド

「vi の達人は画面内のどの位置へでも3ストローク以内で移動できる」

無知な流言 : 「vi は挿入の度にモード切り替えが必要なのでストローク数が多い」

基本viコマンド

カーソル移動 h j k l Space BackSpace w b e W B E O ^ \$
 + Enter - H L M } {] [[% G

検索コマンド / ? n N f F + T : ,

スクロールコマンド ^ F ^ B ^ D ^ U ^ E ^ Y z <Enter> z. z-

挿入コマンド i l a A s S cc c <move> C

削除コマンド x X dd d <move> D

ヤंकペースト y <move> Y p P

基本exコマンド : {lineNum} : {range} d : {range} s/pat/rep/ {opt}

その他 Esc u U . 繰り返し回数指定 (* U は ViVi 独自のキー割り当て)

便利な vi/ViVi コマンド

カーソル移動に検索を使おう

/ ? n N と f F + T : , の組み合わせ

c <move> d <move> を使おう

/pat cw n . n . を使おう

正規表現置換を使おう

動的・静的補完を使おう

シフトを使おう

:renum を使おう

■ 詰めViVi (35)

初期状態を編集して目標状態にする最もキーストローク数の少ない手順を求める。

カーソルは1行目1カラム目にあるものとする。

初期状態は vi コマンドモードで、最後もコマンドモードで終了することとする。

Shift、Ctrl、Alt で修飾したキー、カーソル・ファインクションキーも1ストロークと数える。

マウスは使用不可とする。

○ 問題 1

```
1: // 演算子の前後に空白を入れる
2: for(int ix=0: ix<100: ++ix)
3:   sum+=value[ix]:
```

上記文書を下記の様に修正しなさい。

```
1: // 演算子の前後に空白を入れる
2: for(int ix = 0: ix < 100: ++ix)
3:   sum += value[ix]:
```

・普通のエディタ

= の位置に移動 (↓ → *10)、半角空白を挿入、0 の位置に移動 (→)、半角空白を挿入
 < の位置に移動 (→*5)、半角空白を挿入、1 の位置に移動 (→)、半角空白を挿入、value
 の位置に移動 (↓ ←*10)、半角空白を挿入、+= の位置に移動 (←*3)、半角空白を挿入
 合計: 11 + 1 + 1 + 1 + 5 + 1 + 1 + 1 + 11 + 1 + 3 + 1 = 38手

=、+= への移動はマウスを使う人もいるかと思われる。右手をマウスへ移動、目的地への
 ポインタ移動、クリック、右手をキーボードへ移動 とすれば4手~6手と換算できるので、
 5手とみなすと、38手 → 26手 となる。

・vi コマンド |v| くまー

まずは /= <Enter> で = の位置に移動し (3手)、i <Space> <Esc> で半角空白を挿入する
 (3手、合計6手)。

= の直後にも半角空白を挿入しなくてはいけないので、<Space> <Space> または | | または
 f0 または 2w または ww で移動し (2手、合計8手)、. (ピリオド) で半角空白を挿入する
 (1手、合計9手)。

次は f< で < の位置に移動し . (ピリオド) で空白を挿入 (3手、合計12手)。f1 など
 で移動して . (ピリオド) で空白を挿入 (3手、合計15手)。

最後に += への移動 (+ を検索すると ++ にマッチしてしまうので += を検索) だが、+ は
 正規表現のメタ記号なので /#+= <Enter> で移動して . (ピリオド) で空白を挿入
 (6手、合計21手)。

fv または 2w で v の位置に移動して . (ピリオド) で空白を挿入 (3手、合計24手)。

というのが素直な回答。

※ カーソル移動は検索コマンドを使おう!

※ 繰り返し編集は . (redo) コマンドを使おう!

・vi コマンド lv2 くまー

正規表現を使用した置換を実行すると、少ストローク数で編集を行うことができる場合が多い。
以下の全置換を行えば19手。

`:%s/(¥+?=|<)/ & /g<Enter>`

- ※ 広範囲にわたる規則的な編集は正規表現置換を使おう!
- ※ 範囲を指定すれば、その範囲だけ置換することも可能だぞ!

・vi コマンド lv2 くまー

`n` コマンドを使用すると1ストロークで再検索が可能。
素直に `= <` などを検索せずに再利用可能な文字を検索するとストローク数を減らせる場合がある。また、`i` でなく `a` コマンドを使用することで、カーソル移動を省略できる。
以下のようにすると19手 (by ももたろ氏)

`/x<Enter>a <Esc>w.n.w./m<Enter>.ll`

最後の `ll` は `e` に変えれば 18手 (by 津田)

- ※ 検索する文字や挿入コマンドを工夫すると手数が減ることがあるぞ!

・vi コマンド lv3 くまー

一度に全部置換せずに、後で一部を修正。18手

`:%s/[=<]/ & /g<Enter>wxp`

全置換を行った直後は「`sum+ = value[ix]:`」となっているので、`+` と半角空白を入れ替える。
(by Lei氏)

- ※ `xp` は1文字を、`ddp` は1行を入れ替えるイディオムだぞ!
- ※ 一度に全部完全にやるよりも、不完全にやってから部分対処した方がいい場合もあるぞ!

補足：

本問題では半角空白を6個挿入している。
従って、手数を6で割れば、1個挿入あたりの平均ストローク数を得ることができる。
挿入自体を1ストロークとすれば、上記から-1したものは移動のための平均ストローク数。
18手の場合は、空白1個あたり3手。カーソル移動のための平均ストローク数は2となる。

○ 問題2

- 1: 23
- 2: 16
- 3: 5
- 4: 30
- 5: 32
- 6: 20
- 7: 19
- 8: 24
- 9: 8

上記文書を下記の様に修正しなさい。

- 1: 23 *****
- 2: 16 *****
- 3: 5 *****
- 4: 30 *****
- 5: 32 *****
- 6: 20 *****
- 7: 19 *****
- 8: 24 *****
- 9: 8 *****

• vi コマンド lv1 くまー

23個の * を入力するには 23i***<Esc>** とすればよい。
 その数だけ * を入力するよりはましたが、合計ストローク数がいくつになるのか数えるのが嫌になるくらい多いのは確かだ。が、一応数えてみる。
 まずは、:s/\$/ **<Enter>** で行末に空白を加える (8手)。
 カーソルは最終行にあるので、8A***<Esc>** で8個のアスタリスクを付加 (4手)。
 後は k で8行目に移動し、24. でアスタリスクを付加 (4手)。同様に1行目まで編集する。
 合計: 8 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 3 + 4 + 4 = 43手
 ちなみに、* の数は 23 + 16 + 5 + 30 + 32 + 20 + 19 + 24 + 8 = 177 なので、単純に入力するよりははるかに少ない。
 ※ <数値>i<文字列>**<Esc>** で <文字列>を数字回数入力できるぞ
 ※ その後は <数値>. (ピリオド) で指定回数文字列を入力できるぞ

• ViVi コマンド lv2 くまー

ViVi の :s/pat/rep/ には e オプションがあり、置換文字列を ViViScript の式として評価した結果で文字列を置き換えることができる。
 式の中では \$& でマッチした部分を参照可能なので、以下の置換いっばつで目的の結果を得ることができる (32手)。
 :s/¥d+/\$&+ " + * . repeat (\$&) /e**<Enter>**

• vi コマンド lv3 くまー

vi には @@ という知る人ぞ知るコマンドがある。これはヤンクバッファの内容をコマンドとして実行するというコマンドだ。
 @@ を使用すると、以下の手順で目的の結果を得ることができる (25手 by tsuka氏)。
 :s/d+/& &a***<Ctrl+T>****<Esc>****<Enter>**H**<Ctrl+FZ>**wD@@+**<Ctrl+FZ>**8**<FZ>**
 まずは、前半の :s/d+/& &a***<Ctrl+T>****<Esc>****<Enter>** で、数値を数値 数値a***<Esc>** に置換する。後は各行の数値a***<Esc>** の部分を D でヤンクバッファに入れ、@@ で実行すればよい。各行で繰り返し実行するためには wD@@+ の部分をキーボードマクロで記録し、8**<FZ>** で8回繰り返す。
 この手順は1手減らすことが出来る。tsuka氏は H で1行目に移動してから、各行の処理を行っているが、最終行から上方向に処理することも可能である。後半部分を以下の様にするとよい。24手 (by 津田)
<Ctrl+FZ>wD@@-**<Ctrl+FZ>**8**<FZ>**

○ 問題3

1: ntsuda@master.email.ne.jp

上記文書を下記の様に修正しなさい。

1: jp.ne.email.master@ntsuda

• vi コマンド lv1 くまー

Rjp.ne.email.master@ntsuda<Esc> と打ち直すのが最も単純な方法。27手。

• vi コマンド lv3 くまー

p が最後の文字だけというのを利用すると、dwfpp0 でひとつの単語を p の直後に移動できる。これをキーボードマクロで記録し、7<F2> で7回繰り返すと反転が完了する。

<Ctrl+F2>dwfpp0<Ctrl+F2>7<F2>

合計10手 (by Lei氏)

※ コマンド群の繰り返しはキーボードマクロを使うと手数が短縮できる場合が多いぞ!

• vi/ViVi コマンド lv3 くまー

ViVi は Ctrl + Shift + X で選択部分を削除してクリップボードに追加コピーできる。

したがって、最後から順に単語選択してクリップボードに追加コピーし、最後にペーストすれば順序が入れ替わる。12手 (by tsuka氏)

\$vv<Ctrl+X><Ctrl+F2>vv<Ctrl+Shift+X><Ctrl+F2>7<F2><Ctrl+V>

上記は初期クリップボードの状態に依存しないが、空であるという前提をおけば、

下記の手順で入れ替えることができる。9手 (by tsuka氏)

\$(Ctrl+F2)vv<Ctrl+Shift+X><Ctrl+F2>8<F2><Ctrl+V>

○ 問題4

1: ice coffee

2: hot coffee

3: ice tea

4: hot tea

5: ice milk

6: hot milk

上記文書を下記の様に修正しなさい。

1: coffee

2: tea

3: milk

ちなみに、ノートパッドでの最短手順は、以下の12手だそうです (某掲示板某スレより引用)

1. `ice` を置換で削除 (矩形選択が無い)
 2. `hot` を置換で削除
 3. 正規表現が使えないため1行ずつ削除することにする。Homeキーで文頭へ
 4. Shift + ↓
 5. Backspace
 6. Shift + ↓
 7. Backspaceを叩くが誤って削除し過ぎてしまう
 8. Ctrl + Z
 9. Ctrl + Z
 10. Ctrl + A
 11. Backspace
 12. coffee<Enter>tea<Enter>milk
- 以上12手です

・ vi コマンド lv1 くまー

まずは自然な手順。行頭の `ice` と `hot` を消すために、`:%s/.../ <Enter>` の10手。
次に重複した行を消すには、`:2d <Enter> :3d <Enter> :4d <Enter>` の $4 \times 3 = 12$ 手。
合計: $10 + 12 = 22$ 手だ。

・ ViVi コマンド lv2 くまー

行の先頭4文字を消すにはBOX選択を使うと手数が縮まる。
文書先頭にカーソルがあれば、`<F5>` でBOX選択モードに移行し、`G3I` で先頭の4文字をすべて選択し `x` で消去 (5手)
1行おきに削除は `ddj. j.` とすれば6手。合計: $5 + 6 = 11$ 手だ。

・ ViVi コマンド lv2 くまー

先に1行おきに削除してから `hot` を削除してもよい。
`ddj. j.` で1行おきに削除し (6手)、`4xk. k.` で `hot` を削除 (6手)。
合計: $6 + 6 = 12$ 手だ。 (by Lei氏)

・ ViVi コマンド lv3 くまー

このような繰り返し編集はキーボードマクロを使うと手数が短縮できる場合が多い。
`<Ctrl+F2>dd4xj<Ctrl+F2>` で削除を記録し、`<F2><F2>` または `2<F2>` で目的の結果を得る。
合計: $1 + 2 = 3$ 手だ。 (by Lei氏)

・ vi コマンド lv? くまー

「???????? が、1手ですね。」 (by さね氏)

■ さいごに (1)

vi/ViViコマンドを使って編集操作のキーストローク数を減らすテクニックの数々を紹介した。
本セッションにより少しでもvi/ViViコマンドに興味をもっただけなら幸いである。