附錄B

vi/vim與emacs指令參考表

雖然每個編輯器有自己獨特的功能，不過它們彼此之間功能上還是有許多重疊之處。在這個附錄裡我們整理了vi/vim和emacs編輯器功能的整理與比較。這並不包含emacs非編輯的功能（例如dired功能或郵件處理功能）。在這裡的許多命令都支援重複因子 (vi) 或數字參數 (emacs)。至於可針對區域來操作的指令僅能在emacs和vim中作用，不包含vi。若要進一步說明，請見第四章以及第五章。為了方便起見，指令和它們的參數被整理在同一群組。這裡不考慮可以操作在句子和段落的指令。

如果您看到的emacs指令是完整的指令字，那麼您必須先按 *[Alt-x]* 然後再輸入指令。需要設定的變數則以set-variable來提醒您。所有以set-variable設定的變數也可以放在 **.emacs**中，利用setq 指令。

在這裡，*[Alt]* 鍵只是用來代表emacs中所用的 *[Meta]* 鍵。請使用符合您系統的相對按鍵（例如 *[Esc]*）。

瀏　覽

vi指令	**功能**	emacs**指令**
h (or *[Backspace]*)	游標向左	*[Ctrl-b]*
l (or *[Spacebar]*)	游標向右	*[Ctrl-f]*
k (or *[Ctrl-p]*)	游標向上	*[Ctrl-p]*
j (or *[Ctrl-n]*)	游標向下	*[Ctrl-n]*
[Ctrl-f]	下捲一頁	*[Ctrl-v]*
[Ctrl-b]	上捲一頁	*[Alt-v]*
[Ctrl-d]	下捲半頁	—
[Ctrl-u]	上捲半頁	—
1G	移至檔頭	*[Alt-<]*
40G	移至第40行	**goto-line 40**
G	移至檔尾	*[Alt->]*
—	切換行數顯示模式	**line-number-mode**
[Ctrl-g]	顯示目前行數以及檔案百分比	**what-line**
:set number (nu)	所有行加行號	—

在一行中瀏覽

vi 中的B, E和W指令和小寫的相對指令 (**b**, **e**, **w**) 功能類似，除了標點符號忽略不考慮之外。這在emacs中並沒有類似的情形。

vi指令	**功能**	emacs**指令**
b	向左移至一字的開頭	*[Alt-b]*
e	向右移至一字的尾端	*[Alt-f]*
w	向右移至一字的開頭	—
0 or **|**	移至行首	*[Ctrl-a]*
30|	移至第30欄	*[Ctrl-a][Alt-30][Ctrl-f]*
^	移至一行的第一個字的頭一個字元	*[Alt-m]*
$	移至行尾	*[Ctrl-e]*

插入文字

不像vi，emacs總是處在「輸入模式」。這裡的emacs指令僅是讓您從游標處開始插入文字。被插入的文字會以斜體字表示成*text*。輸入控制字元則是這樣顯示*[Ctrl-b]*。

vi**指令**	**功能**	emacs**指令**
i	從游標處左邊輸入**text**	**text**
20i-*[Esc]*	輸入20個連字號	*[Alt-20]-*
I	從行首輸入文字	*[Ctrl-a]* ***text***
a	從游標右邊附加文字	*[Ctrl-f]* *text*
A	在行尾附加文字	*[Ctrl-e]* *text*
o	從游標處下方開新一行	*[Ctrl-e][Enter]*
O	從游標處上方開新一行	*[Ctrl-a][Enter]*
[Ctrl-v][Ctrl-b]	輸入 *[Ctrl-b]*	*[Ctrl-q][Ctrl-b]*
[Ctrl-v][Esc]	輸入 *[Esc]*	*[Ctrl-q][Esc]*
:set showmode	當vi處在輸入模式時顯示訊息	—
:set sm	顯示成對的另一個) 或 }	**blink-matching paren t**（利用**set-variable**）
:set ts=n	將定位鍵設為*n*（預設值：8）	**edit-tab-stops**
:set ai	下一行自動縮排	—

刪除與移動文字

在這一節的所有編輯動作都可以被還原。然而，*[Ctrl-d]*，刪除一個字元（在emacs中） 並不構成移除的動作。所以被 *[Ctrl-d]* 刪除的字元無法從移除環中還原。然而，當指令前面加上數字參數時，這種刪除動作就可以被還原。

vi**指令**	**功能**	emacs**指令**
x	刪除游標下的字元	*[Ctrl-d]* 或 *[Delete]*
6x	刪除游標下的字元以及右邊的五個字元	*[Alt-6][Ctrl-d]*

vi指令	功能	emacs指令
X	刪除前一個字元	—
dd	刪除目前行	*[Ctrl-a][Ctrl-k][Ctrl-k]*
4dd	刪除四行	*[Alt-4][Ctrl-k]*
64dd	刪除64行	*[Ctrl-u][Ctrl-u][Ctrl-u][Ctrl-k]*
dw	刪除一個字	*[Alt-d]*
—	刪除前一個字	*[Alt][Delete]*或 *[Alt][Backspace]*
d0（d 和零）	刪除至行首	*[Alt-0][Ctrl-k]* （Alt和零）
d$	刪除至行尾	*[Ctrl-k]*
—	刪除空白行	*[Ctrl-x][Ctrl-o]*
d	刪除一區域（vim適用）	*[Ctrl-w]*
p	將刪除的文字貼回游標右方（在vi中則是下方）	*[Ctrl-y]*
P	將刪除的文字貼回游標上方或左方	—
“ add	將目前行刪除並移至緩衝區a	*[Ctrl-u][Ctrl-x]* xa (on region)
“ ap	將緩衝區a的內容貼回	*[Ctrl-u][Ctrl-x]* ga
—	保留殺除環中*n*個被刪除/複製的文字	**kill-ring-max** n（利用set-variable）
ddp	對調目前行和下一行	*[Ctrl-n][Ctrl-x][Ctrl-t]*
kddp	對調目前行和上一行	*[Ctrl-x][Ctrl-t]*
J	將下一行與目前行合併	*[Ctrl-e][Ctrl-d]*
kJ	將上一行與目前行合併	*[Alt-^]*
xp	對調兩個字元	*[Ctrl-t]*
—	對調兩個字組	*[Alt-t]*
—	游標移動中間行	**center-line**

複製文字

vi指令	功能	emacs指令
yy	複製目前行	需要先定義區域（見下面）
6yy	複製6行	同上
yw	複製一個字組	同上
y	複製區域（只能使用在vim）	*[Alt-w]*
p	將複製的文字貼在游標右邊（在vi中則是下方）	*[Ctrl-y]*
P	將複製的文字貼在游標左邊或上方	—
“ ayy	複製目前行至緩衝區a	*[Ctrl-x]* xa （針對區域）
“ ap	將緩衝區a內容貼回	*[Ctrl-x]* ga

更動、讀進、過濾文字

vi指令	功能	emacs指令
r*ch*	將游標處字元代換為*ch*	*[Ctrl-d] ch*
R	從游標處起向右覆蓋文字	使用**overwrite-mode**然後*text*

vi指令	功能	emacs指令
s	將游標處單一字元代換為任意字數的文字	*[Ctrl-d] text*
S	代換整行	*[Ctrl-a][Ctrl-k] text*
cw	更動一字組	*[Alt-d] text*
c	更動區域中的文字（vim適用）	—
~	切換區域中文字的大小寫	—
—	將字元轉換成大寫	*[Alt-u]*
—	將字元轉換成小寫	*[Alt-l]*
—	將一字的開頭字元變大寫	*[Alt-c]*
!*navigation_cmd cmd*	針對區域執行指令*cmd*，以*navigation_cmd*結尾	*[Ctrl-u][Alt-|] cmd*
—	對區域執行指令cmd，輸出在另一個分別的視窗	*[Alt-|] cmd*
!navigation_cmd sort	針對區域排序，以navigation_cmd結尾	sort-lines
!tr ‘ [a-z] ’‘ [A-Z] ’	將區域文字轉成大寫（vim適用）	*[Ctrl-x][Ctrl-u]*
!tr ‘ [A-Z] ’‘ [a-z] ’	將區域文字轉成小寫 （vim適用）	*[Ctrl-x][Ctrl-l]*
:r foo	從游標處下一行讀進檔案foo	*[Ctrl-x]* i foo
:r !head -3 foo	從游標處下一行讀進檔案foo頭三行	*[Ctrl-u][Alt-!]* head -3 foo

啟動編輯器

vi指令	功能	emacs指令
vi +100 foo	在第100 行處開檔案	emacs +100 foo
vi +/*pat* foo	開啟檔案，游標停在第一個 *pat*位置	—
vi + foo	開啟檔案，游標移至檔尾	—
—	載入henry的 .emacs	emacs -u henry foo
—	不要載入 .emacs	*emacs* -q foo
vi -R foo	以唯讀模式開檔	—

存檔與離開

vi指令	功能	emacs指令
:w	存檔不離開	*[Ctrl-x][Ctrl-s]*
:w bar	另存新檔bar	*[Ctrl-x][Ctrl-w]*
:w! bar	同上，會覆蓋原存之檔案bar	—
:*n1*,*n2*w foo	將行*n1*到*n2*之間的內容寫入檔案foo	write-region（針對區域）
:*n1*,*n2*w >> foo	將行*n1*到*n2*之間的內容附加於檔案foo	append-to-file（針對區域）
:.w foo	將目前行寫入檔案foo	—
:$w foo	將最末行寫入檔案foo	—
:x	存檔離開	*[Ctrl-u][Ctrl-x][Ctrl-c]*
:wq	同上	—

vi指令	功能	emacs指令
:q	不存檔離開（未更動檔案）	*[Ctrl-x][Ctrl-c]*
:q!	不存檔離開（已更動檔案）	同上，但輸入n後再輸入yes
—	打開/關掉自動儲存模式	auto-save-default （t或nil） （利用set-variable）
—	設定按鍵數超過n次之後自動儲存	auto-save-interval n （利用set-variable）
—	設定閒置時間超過n秒後自動儲存	auto-save-timeout n （利用set-variable）
—	叫回自動儲存的檔案	recover-file

編輯多重檔案

在vi中，您無法使用 :e, :n以及 :rew，除非目前檔案已經儲存（且autowrite沒有被設定）。驚嘆號！則會強制執行。emacs則可以讓您自由地在檔案間切換，不需存檔。

vi指令	功能	emacs指令
:e foo	停止目前編輯，改編輯檔案foo	*[Ctrl-x][Ctrl-f]*
:e! foo	同上，且放棄所有的修改不存檔	不須先存檔
—	將目前緩衝區放入另一檔案	*[Ctrl-x][Ctrl-v]*
:e!	叫回目前檔案上一次存檔時的版本	revert-buffer
[Ctrl-^]	回到最近編輯的檔案	*[Ctrl-x]* b *[Enter]*
:n	編輯下一個檔案（當呼叫時輸入多個檔案名時）	*[Ctrl-x]* b並從清單中挑選
:set autowrite (aw)	當檔案切換時自動將目前檔儲存（在vi中用:n）	不需要
:rew	將編輯檔案切換為原先之檔案（當呼叫時輸入多個檔案名時）	—
—	刪除目前緩衝區	*[Ctrl-x]* k

多重視窗（emacs和vim）

vi指令	功能	emacs指令
:sp	將目前視窗切為兩個	*[Ctrl-x]* 2
:new	開啟一新的空視窗	*[Ctrl-x]* b
[Ctrl-w][Ctrl-w]	在兩視窗間切換	*[Ctrl-x]* o
:on	將目前視窗設為唯一視窗	*[Ctrl-x]* 1
:q	離開目前視窗	*[Ctrl-x]* 0 (零)
:qa	離開所有視窗	—
:xa	將所有視窗存檔離開	*[Ctrl-x][Ctrl-c]* 然後 !
—	將另一個視窗文字向下捲頁	*[Ctrl][Alt]* v

vi指令	功能	emacs指令
[Ctrl-w] +	加大視窗大小	*[Ctrl-x]* ^
[Ctrl-w] -	減小視窗大小（在emacs中減小 n行）	*[Ctrl-u]* -n *[Ctrl-x]* ^
—	在另一個視窗開啟檔案	*[Ctrl-x]* 4 *[Ctrl-f]*

搜尋與重複

不像emacs，vi對於字串或者正規表示式均使用同樣的搜尋和重複指令。emacs可以使用遞增式搜尋。vi使用另外的按鍵來在一行中搜尋一個字元。

vi指令	功能	emacs指令
/*pat*	非遞增向下搜尋字串*pat*	*[Ctrl-s][Enter] pat*
/*pat*	同上，但*pat*為正規表示式	*[Ctrl][Alt]* s *[Enter] pat*
?*pat*	非遞增向上搜尋字串*pat*	*[Ctrl-r][Enter] pat*
?*pat*	同上，但*pat*為正規表示式	*[Ctrl][Alt]*r*[Enter] pat*
—	遞增向下搜尋字串*pat*	*[Ctrl-s] pat*
—	同上，但*pat*為正規表示式	*[Ctrl][Alt]* s *pat*
—	遞增向上搜尋字串*pat*	*[Ctrl-r]* pat
—	同上，但*pat*為正規表示式	*[Ctrl][Alt]* r *pat*
n	重複搜尋，方向相同	*[Ctrl-s]*
N	重複搜尋，方向相反	*[Ctrl-r]*
n	重複正規表示式搜尋，方向相同	*[Ctrl][Alt]* s *[Enter][Enter]*
N	重複正規表示式搜尋，方向相反	*[Ctrl][Alt]*r*[Enter][Enter]*
—	從游標處開始向下搜尋字	*[Ctrl-s][Ctrl-w]*
—	取消搜尋	*[Esc]*
:set wrapscan (ws)	搜尋若遇到檔案邊界，繞捲至另一端繼續搜尋	—
:set ignorecase (ic)	比對時忽略大小寫	case-fold-search t（利用set-variable）
:set magic	維持正規表示式字元的意義	—
f*c*	向下搜尋字元*c*	*[Ctrl-s] c*
F*c*	向上搜尋字元*c*	*[Ctrl-r] c*
;	重複上一個字元搜尋	*[Ctrl-s]*
,		*[Ctrl-r]*

代　換

vi指令	功能	emacs指令
:1,$s/*s1*/*s2*/g	全域性地將字串*s1*代換為*s2*	replace-string
:1,$s/*s1*/*s2*/g	同上，但*s1*為正規表示式	replace-regexp
:1,$s/*s1*/*s2*/gc	互動式代換字串	*[Alt- %]*

vi指令	功能	emacs指令
:1,$s/*s1*/*s2*/gc	同上，但*s1*為正規表示式	query-replace-regexp
—	取代時維持原字串之大小寫	case-replace t（利用set-variable）
:s	重複目前行上一次代換動作（vim適用）	

標記和書籤

vi指令	功能	emacs指令
ma	設定標記a	*[Ctrl-x]* /a
‘ a	移至標記a	*[Ctrl-x]* ja
‘ ’	在兩位置間切換	*[Ctrl-x][Ctrl-x]*
—	設定書籤	*[Ctrl-x]* rm
—	跳至書籤	*[Ctrl-x]* rb

還原與取消還原

vi指令	功能	emacs指令
.	重複上一個指令（emacs需先按 *[Alt-x]* 再輸入指令）	*[Ctrl-x][Esc][Esc]*
u	還原上個指令	*[Ctrl-x]* u 或 *[Ctrl--]*
[Ctrl-r]	取消上個還原指令（vim適用）	同上，但前提是所有還原動作均已回復
U	還原目前行的所有更動	—
“ 4p	還原第四個先前刪除於後衝區內容（vi中是完整行）	*[Ctrl-u]* 4 *[Ctrl-y]*
u.，在啟始 “ 1p 之後	取消上一個還原，並從下一個緩衝區繼續還原（vi中是完整行）	*[Alt-y]* after initial *[Ctrl-y]*

設定文字縮寫

vi指令	功能	emacs指令
—	啟動（在emacs中是切換）縮寫模式	abbrev-mode
:ab *stg name*	設定*name*的縮寫為*stg*	*stg [Ctrl-x]* aig *name*（ail為區域性的縮寫）
—	將縮寫擴展為全字串	*[Alt-/]*
:ab	列出所有的縮寫	list-abbrevs
:unab *stg*	刪除縮寫*stg*	利用edit-abbrevs刪除縮寫
—	刪除所有縮寫	kill-all-abbrevs
—	儲存目前縮寫	write-abbrev-file
—	讀取縮寫儲存檔	read-abbrev-file
—	儲存所有之後的縮寫	save-abbrevs t（利用set-variable）

巨集與鍵盤對應

vi指令	功能	emacs指令
輸入一指令序列然後按 “ ayy	定義巨集（在vi中名稱為a）	*[Ctrl-x]*(commands *[Ctrl-x]*)
@a	執行上一個定義的巨集（在vi中名稱為a）	*[Ctrl-x]* e
—	將上一個定義的巨集取名為 *macroname*	name-last-kbd-macro
@m	執行巨集m(vi)或*macroname* (emacs)	*macroname* （先按 *[Alt-x]*）
:map *key commands*	設定鍵盤對應至指令*commands*(vi)或*macroname* (emacs)	global-set-key, 輸入*key*以及*macroname*
:map! *key commands*	設定按鍵*key*對應至指令*commands*在輸入模式下	同上
將指令 :map command 放入 .exrc	將巨集存檔	insert-kbd-macro
—	讀取巨集定義檔	load-file
:map	顯示所有命令模式下之指令鍵盤對應	—
:map!	顯示所有輸入模式下之指令鍵盤對應	—
:unmap *key*	刪除指令模式之鍵盤對應*key*	—
:unmap! *key*	刪除輸入模式之鍵盤對應*key*	—

與UNIX之溝通介面

編輯器可以利用 *[Ctrl-z]* 來暫停編輯工作，不過必須shell支援工作排程控制。

vi指令	功能	emacs指令
:!*cmd*	執行UNIX指令*cmd*	*[Alt-!] cmd*
:!%	執行編輯中之指令手稿或perl手稿	不支援
:sh	跳到UNIX shell下	shell
[Ctrl-z]	暫停編輯器	*[Ctrl-z]* 或 *[Ctrl-x][Ctrl-z]*
:!cc %	編譯目前編輯中的C程式	compile
:!javac %	編譯目前編輯中的Java程式	不支援

求助(emacs適用)

vi指令	功能	emacs指令
—	按鍵所對應的功能（詳細）	*[Ctrl-h]* k
—	按鍵所對應的功能（僅一行）	*[Ctrl-h]* c
—	指令所對應的功能	*[Ctrl-h]* f
—	指令對應的按鍵	*[Ctrl-h]* w
—	變數的功能以及目前的設定	*[Ctrl-h]* v
—	使用此種觀念的指令	*[Ctrl-h]* a
—	執行線上教學	*[Ctrl-h]* t
—	執行info閱讀器	*[Ctrl-h]* i

vi**指令**	**功能**	emacs**指令**
—	取消一序列	*[Ctrl-g]*
:set all	顯示所有的設定(vi)或變數 (emacs)	**list-options**
[Ctrl-l]	重繪螢幕	*[Ctrl-l]*
v	定義區域起點(vim適用)	*[Ctrl-@]* 或 *[Ctrl][Spacebar]*
—	命令stg自動展開，先按 *[Alt-x]*	**stg** *[Tab]*
:set ro	更改為唯讀模式(vi)或切換模式(emacs)	*[Ctrl-x][Ctrl-q]*