

Howto use VI editor in Linux ?

Problem:

I am new to Linux and do not understand how to use VI editor in Linux. Is there any detailed documentation and explanation ?

ABOUT VIScreen-oriented (visual) display editor based on ex.SYNTAXvi [-l -s] [-l] [-L] [-R] [-r [filename]] [-S] [-t tag] [-v] [-V] [-x] [-w] [-n] [-C] [+command | -c command] filename

- | -sSuppress all interactive user feedback. This is useful when processing editor scripts.-lSet up for editing LISP programs.-LList the name of all files saved as the result of an editor or system crash.-RReadonly mode; the readonly flag is set, preventing accidental overwriting of the file.-r filenameEdit filename after an editor or system crash. (Recovers the version of filename that was in the buffer when the crash occurred.)-SThis option is used in conjunction with the -t tag option to tell vi that the tags file may not be sorted and that, if the binary search (which relies on a sorted tags file) for tag fails to

find it, the much slower linear search should also be done. Since the linear search is slow, users of large tags files should ensure that the tags files are sorted rather than use this flag. Creation of tags files normally produces sorted tags files. See ctags for more information on tags files.-t tagEdit the file containing the tag, tag, and position the editor at its definition.-vStart up in display editing state using vi . You can achieve the same effect by simply typing the vi command itself.-VVerbose. When ex commands are read by means of standard input, the input will be echoed to standard error. This may be useful when processing ex commands within shell scripts.-xEncryption option; when used, vi simulates the X command of ex and prompts the user for a key. This key is used to encrypt and decrypt text using the algorithm of the crypt command. The X command makes an educated guess to determine whether text read in is encrypted or not. The temporary buffer file is encrypted also, using a transformed version of the key typed in for the -x option. If an empty encryption key is entered (that is, if the return key is pressed right after the prompt), the file will not be encrypted. This is a good way to decrypt a file erroneously encrypted with a mistyped encryption key, such as a backspace or undo key.-wnSet the default window size to n. This is useful when using the editor over a slow speed line.-CEncryption option; same as the -x option, except that vi simulates the C command of ex . The C command is like the X command of ex , except that all text read in is assumed to have been encrypted.+command | -c commandBegin editing by executing the specified editor command (usually a search or positioning command).filenameThe file to be edited.USER COMMANDS

Arrow keysMove cursorhjklSame as arrow keysitextESCInsert textcwnewESCChange word to neweasESC pluralize word (end of word; append s; escape from input state)xdelete a characterddelete a wordddddelete a line3ddddeletes 3 linesuundo previous changeZZexit vi , saving changes:q!CRquit, discarding changes/textCRsearch for text^U ^Dscroll up or down:cmdCRany ex or ed commandESCend insert or incomplete commandDEL(delete or rubout) interrupts:wCRwrite back changes:w!CRforced write, if permission originally not valid:q!CRquit:q!CRquit, discard changes:e nameCRedit file name:e!CRedit, discard changes:e + nameCRedit, starting at end:e +nCRedit, starting at line n:e #CRedit alternate file:e! #CRedit alternate file, discard changes:w nameCRwrite file name:w! nameCROverwrite file name:shCRrun shell, then return:!cmdCRrun cmd, then return:nCRedit next file in arglist:n argsCRspecify new arglist^Gshow current file and line:ta tagCRposition cursor to tagFforward screen^Bbackward screen^Dscroll down half screen^Uscroll up half screennGgo to the beginning of the specified line (end default), where n is a line number/patnext line matching pat?patprevious line matching patnrepeat last / or ? commandNreverse last / or ? command/pat/+n nth line after pat?pat? -n nth line before pat]]next section/function[[previous section/function(beginning of sentence)end of sentence{beginning of paragraph}end of paragraph%find matching () or { }^Lclear and redraw window^Rclear and redraw window if ^L is -> keyzCRredraw screen with current line at top of windowz-CRredraw screen with current line at bottom of windowz.CR redraw screen with current line at center of window/pat/z-CRmove pat line to bottom of windowzn.CR use n-line window^E scroll window down one line^Yscroll window up one line``move cursor to previous context"move cursor to first non-white space in linemxmark current position with the ASCII lower-case letter x`x move cursor to mark x`xmove cursor to first non-white space in line marked by xHtop line on screenLlast line on screenMmiddle line on screen+next line, at first non-white space character-previous line, at first non-white space characterCRreturn, same as +down-arrow or jnext line, same columnup-arrow or kprevious line, same column^first non-white space character0beginning of line\$end of line| or ->forwardh or <-backward^Hsame as <- (backspace)spacesame as -> (space bar)fxfind next xFxfind next xtmove to character following the next xTxmove to character following the previous x;repeat last f, F, t, or T,repeat inverse of last f, F, t, or Tn|move to column n%find matching () or { }wforward a wordbback a wordeend of word)to next sentence}to next paragraph(back a sentence{back a paragraphWforward a blank-delimited wordBback a blank-delimited wordEend of a blank-delimited word^Herase last character (backspace)^Werase last worderasesyour erase character, same as ^H (backspace)killyour kill character, erase this line of input\quotes your erase and kill charactersESCends insertion, back to command modeCTRL-Cinterrupt, suspends insert mode^Dbacktab one character; reset left margin of autoindent^^Dcaret (^) followed by control-d (^D); backtab to beginning of line; do not reset left margin of autoindent0^Dbacktab to beginning of line; reset left margin of autoindent^Vquote non-printable characteraappend after cursorAappend at end of lineiinsert before cursorlinsert before first non-blankooopen line belowOopen line abovevreplace single character with xRtextESCreplace charactersddeletecchangeeyyank lines to buffer>left shift<right shift!filter through commandCchange rest of line (c\$)Ddelete rest of line (d\$)ssubstitute characters (cl)Ssubstitute lines (cc)Jjoin linesxdelete characters (dl)Xdelete characters before cursor dh)Yyank lines (yy)3yyyank 3 lines3lyyank 3 characterspput

back text after cursorP
put back text before cursor " .nr)l xp"n
put from buffer x " .nr)l xy"n
yank to buffer x " .nr)l xd"n
delete into buffer xu
undo last changeU
restore current line.repeat last change " .nr)l dp"n
retrieve d'th last deleteEXAMPLES
vi myfile.txtEdits the file myfile.txt.