vi - invoke a screen-oriented display editor

Description

   The vi command offers a powerful set of text editing operations based on
   a set of mnemonic commands. Most commands are single keystrokes that
   perform simple editing functions. vi displays a full screen ``window''
   into the file you are editing. The contents of this window can be
   changed quickly and easily within vi. While editing, visual fedback is
   provided (the name vi itself is short for ``visual'').

   Special keys

   There are several special keys in vi. The following keys are used to
   edit, delimit, or abort commands and command lines.

   <Esc>   Used to return to vi command mode or to cancel partialy
                formed commands.

   <Return>   Used to start a newline when in insert mode.

   /   Used to specify a string to be searched for. The slash
                apears on the status line as a prompt for a search string.
                The question mark (?) works exactly like the slash key,
                except that it is used to search backward in a file instead
                of forward.

   :   The colon is a prompt for a command. You can then type
                in any command, followed by an <Esc> or <Return>, and the
                given command is executed.

   Starting and exiting vi

   To enter vi

   v   - Special Entrance into vi from viewing an MDS Printfile.
          Enter a lower case v any time after pressing the View Selection
          from the MDS View/Print Screen.

   vi file   Type this in at the Unix command prompt (# or $)
                 to Edit named file

   There are several ways to exit the editor:

   :ZZ   The editing buffer is writen to the file only if any changes were
         made.

   :q!   Cancels an editing sesion. The exclamation mark (!) tells vi to
         quit unconditionaly. In this case, the editing buffer is not writ-

ten out.

vi commands

    Cursor Movement

    The cursor movement keys alow you to move your cursor around in a file.

    forward space - right arrow or space bar

    backkspace - left arrow or back-space key

    next line - <<return>> and down arrow

    previous line - up arrow

    end of line - $

    goto line - :[LineNumber] and return key
            To go to the last line in file use <Ctrl>G

            To find the line number of the current line, use <Ctrl>g.

    SCREEN COMMANDS

    The screen commands move the cursor and are useful in paging or scrolling
    through a file. These commands are described below:

    scroll- <<ctrl>>u and <<ctrl>>d

    Syntax: [size]<Ctrl>u
               [size]<Ctrl>d

    Function: Scrolls the screen up a half window (<Ctrl>u) or down a half
              window (<Ctrl>d). If size is given, the scroll is size number
              of lines. This value is remembered for all later scrolling
              commands.

    page - <<ctrl>>f and <<ctrl>>b

    Syntax: <Ctrl>f
               <Ctrl>b

    Function: Pages screen forward and backward. Two lines of continuity are
              kept betwen pages if posible. A preceding count gives the
              number of pages to move forward or backward.

    status - <<ctrl>>g

```
Syntax:    bell
           <Ctrl>g

Function: Displays vi status on status line.  This gives you the name of
          the file you are editing, whether it has been modified, the
          current line number, the number of lines in the file, and the
          percentage of the file (in lines) that precedes the cursor.

Text Insertion
--------------

The text insertion commands always place you in insert mode.  Exit from
insert mode is always done by pressing <Esc>.  The following insertion
commands are pure insertion commands; no text is deleted when you use
them.  This difers from the text modification commands, change, replace,
and substitute, which delete and then insert text in one operation.

insert - i

Syntax:    i[text]<Esc>

Function: Insert text in editing buffer.  The lowercase i command places
          you in insert mode.  text is inserted before the character
          beneath the cursor.
          Exit insert mode by typing the <Esc> key.

append - a

Syntax:    a[text]<Esc>

Function: Appends text to the editing buffer.
          This is the only way to add text to the end of a line.
          command begins appending text after the cursor.

open new line - o and O

Syntax:    o[text]<Esc>
           O[text]<Esc>

Function: Opens a new line and inserts text.  The lowercase o command
          opens a new line below the current line; uppercase O opens a
          new line above the current line.  After the new line has ben
          opened, both these commands work like the i command.

Text Deletion
-------------

Many of the text deletion commands use the ``d'' key as an operator.
This operator deletes text objects delimited by the cursor and a cursor
movement command.  Deleted text is always saved in a buffer.  The delete
commands are described below:

delete character - x

Syntax:    x

Function: Deletes a character.
```

This is a quick and easy way to delete a few characters.

delete - dd and D

Function:
        The dd command deletes whole lines. The uppercase D command
        deletes from and including the cursor to the end of the curent
        line.

Text Modification

The text modification commands all involve the replacement of text with
other text. This means that some text will necesarily be deleted. Al
text modification commands can be ``undone'' with the u command:

undo - u and U

Syntax: u
          U

Function: Undoes the last insert or delete command. The lowercase u com-
          mand undoes the last insert or delete command. This means that
          after an insert, u deletes text; and after a delete, u inserts
          text. For the purposes of undo, all text modification commands
          are considered insertions.

          The uppercase U command restores the current line to its state
          before it was edited, no mater how many times the current line
          has been edited since you moved to it.

repeat .   This a period.

Syntax: .

Function: Repeats the last insert or delete command.

replace - r and R

Syntax: rchar
          Rtext <Esc>

Function: Overstrikes character or line with char or text, respectively.
          Use r to overstrike a single character and R to overstrike a
          whole line.

Join lines - J

Syntax: J

Function: Joins the current line with the following line.

Searching

The search commands search either forward or backward in the editing
buffer for text that matches a given regular expression.

search - / and ?

Syntax: /[pattern]<Return>
          ?[pattern]<Return>

Function: Searches forward (/) or backward (?) for pattern.
          If no pattern is given, then the last pattern
          searched for is used.

next string - n and N

Syntax: n
          N

Function: Repeats the last search command. The n command repeats the
          search in the same direction as the last search command. The N
          command repeats the search in the oposite direction of the
          last search command.