

A  
Northern Illinois University  
Academic Computing Services  
Workshop

## UNIX Basics for the Mere-Mortal User

Michael G. Prais  
Swen Parson 120  
753-1057

### Editing with the VI Text Editor

The contents of files can also be viewed and manipulated.

more dante	Displays the first screen of a file and waits for a command.
<u>ENTER</u>	Scrolls the next line in the file onto the bottom of the screen.
<u>SPACE</u>	Displays the next screen of the file.
? h	Displays a list of the commands for <i>more</i> . (BSD/SysV)
q	Terminates <i>more</i> .
more +4 dante	Displays a file on screen with line 4 at the top of the screen with one line above it and waits for further instructions.
=	Displays your location within the file.
q	Terminates <i>more</i> .

---

more +/Four dante	Displays a file on screen with a line containing <i>Four</i> at the top of the screen and waits for further instructions.
q	Terminates <i>more</i> .
vi +/Seven dante	Displays a file for full screen editing with a line containing <i>Seven</i> at the middle of the screen.
:q	Exits the editor without changing anything.
vi +7 dante	Displays a file for full screen editing with line 7 at the middle of the screen.
CTRL G	Displays the file name and the current line number.
:q	Exits the editor without changing anything.
vi + dante	Displays a file for full screen editing with the last line of the file at the middle of the screen.
:q	Exits the editor without changing anything.
vi	Opens a screen that can be used to create a new file.

The visible editor *vi* starts up in a mode in which everything typed is interpreted as an editing command.

CTRL G	Displays the file name and the current line number.
o	Opens the editor in a mode in which almost everything typed is inserted as text.

---

Line 1	Enter some text.
Line 2	
<code>ESCAPE</code>	Returns the editor to command mode.
<code>:q</code>	Exits the editor only when changes were not made.
<code>:q!</code>	Exits the editor discarding any changes that were made.
<code>vi dante</code>	Displays a file for full screen editing with the first line of the file at the top of the screen.

The visible editor *vi* reads a file into an internal buffer and changes the buffer (not the original file) on command. The original file is only replaced on command.

*Vi* has three modes:  
the command mode, the insert mode, and the line mode.

<code>:set showmode</code>	Displays a note at the bottom right of the screen whenever in an insert mode.
----------------------------	---

The *command mode* is the central mode that gives access to the other modes and allows movement through the displayed file. Commands in *vi* are case sensitive, so make sure the Caps Lock is off.

<code>ESCAPE</code>	Causes the system to beep when you are in the command mode.
---------------------	--

Since the system beeped, you are now in the command mode. If the system did not beep, you *were* not in the command mode when you pressed `ESCAPE`, but since you did press `ESCAPE`, you *are* now in command mode.

The next time you press `ESCAPE`, the system will beep.

<code>CTRL G</code>	Displays the file name and current line number.
<code>ENTER</code>	Moves the cursor to the start of the next line.
<code>RIGHT</code>	Moves the cursor right to the next character.
<code>}</code>	Moves the cursor to the next blank line (the end of the current paragraph).

Most *vi* commands take a preceding multiplier.

<code>5RIGHT</code>	Moves the cursor five characters to the right.
<code>5ENTER</code>	Move the cursor to the start of the fifth line below.
<code>&gt;&gt;</code>	Indents the current line.
<code>&lt;&lt;</code>	Unindents the current line.
<code>5&gt;&gt;</code>	Indents the current and next four lines.
<code>J</code>	Joins the current and next line.
<code>u</code>	Undoes the last change to the buffer.
<code>~</code>	Changes the capitalization of the current character.
<code>.</code>	Repeats the last command.
<code>U</code>	Undoes changes to the current line. Changing lines loses the changes made in the previous "current" line.

W~	Capitalizes the next word.
W~	Capitalizes the next word.
mm	Marks the character as position m as one of 26 (a - z) points of reference.

### Cursor Movement Commands

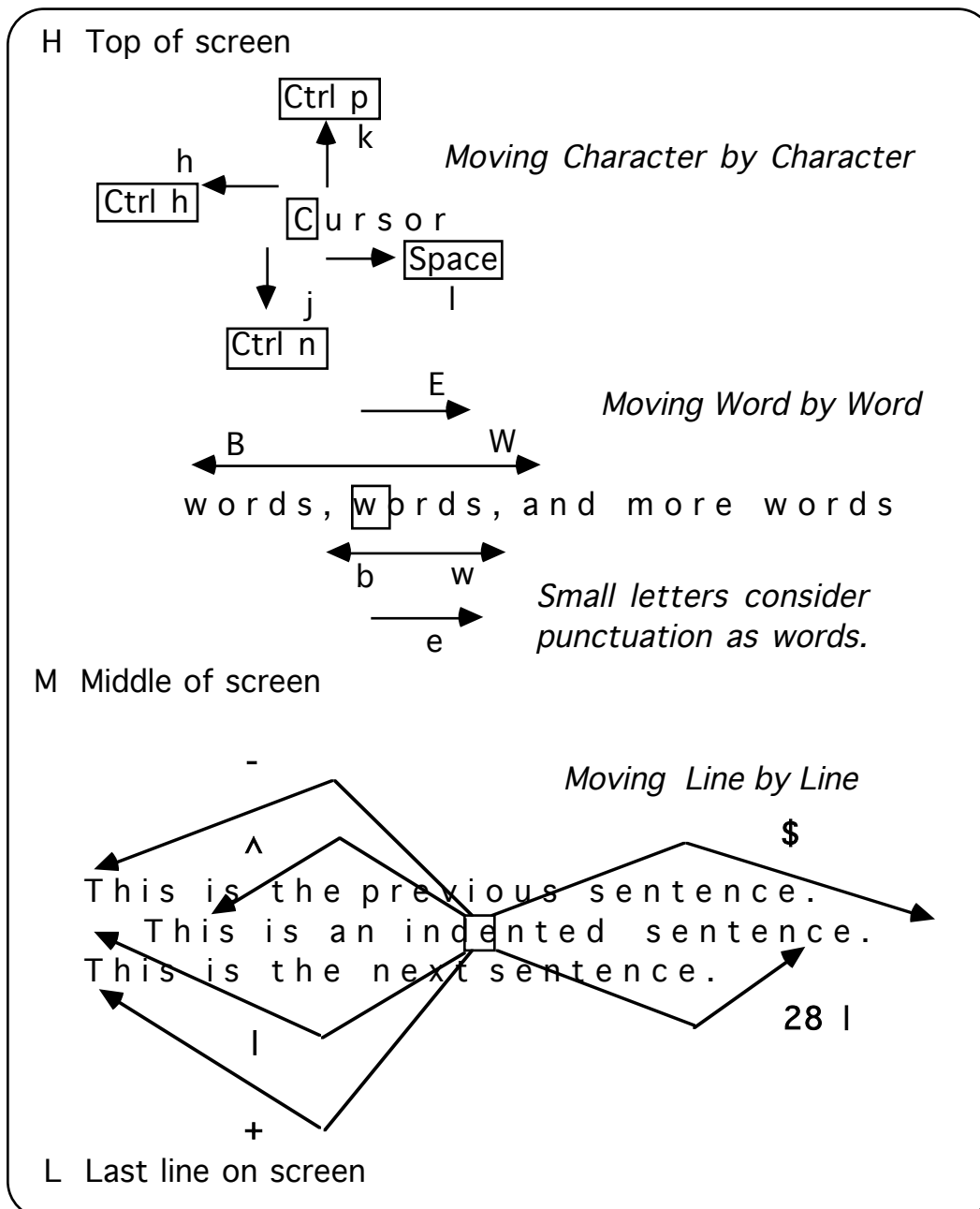
1G First line in file

" Returns the cursor to its previous line

'm Line marked with position m

` Returns the cursor to its previous character

`m Character marked as position m



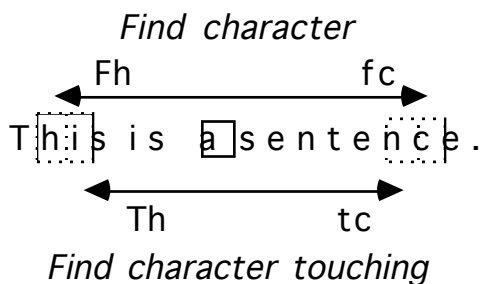
nG Line n in file

G Last line in file

### Window Movement Commands

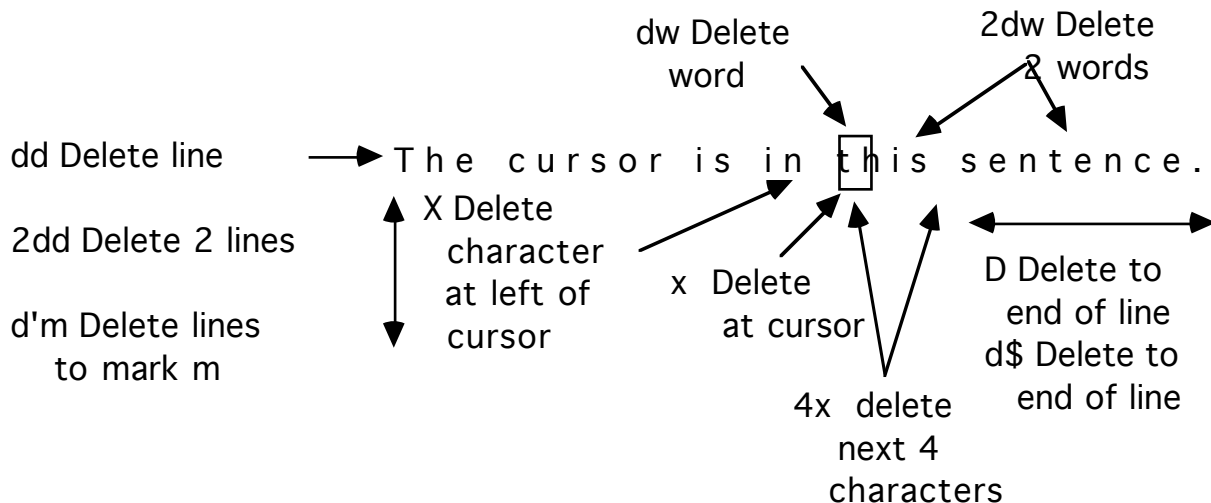
	Scroll Screen:		Place Current Line:
<code>CTRL B</code>	Up/Back one screen	<code>z+</code>	At top of screen
<code>CTRL U</code>	Up/Back half screen		
<code>CTRL E</code>	Up/Back one line		
	---- Current Line ----	<code>z.</code>	At middle of screen
<code>CTRL Y</code>	Forward/Down one line		
<code>CTRL D</code>	Forward/Down half screen		
<code>CTRL F</code>	Forward/Down one screen	<code>z-</code>	At bottom of screen

### Find in Current Line:

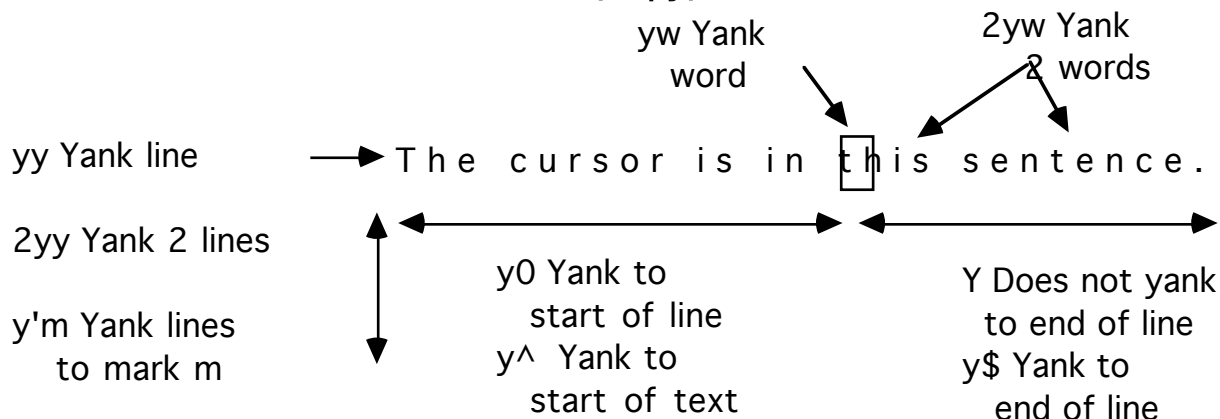


; Repeat Last Find

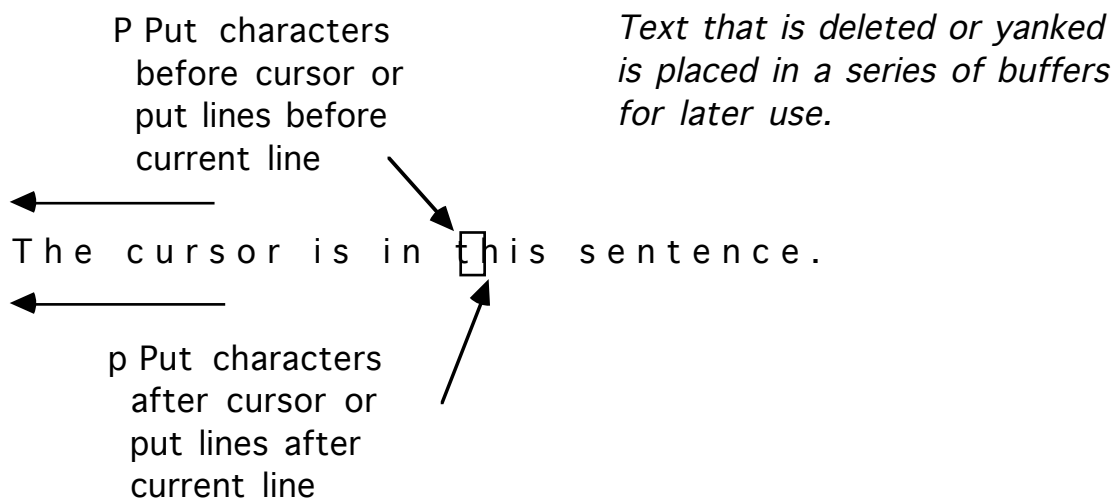
### Delete (Cut) Text



### Yank (Copy) Text



### Put (Paste) Text





*Deletes*, *yanks*, and *changes* place text in an unnamed buffer which is used by the *put* commands.

There are also 26 named buffers (a through z) in which to hold text during *deletes* and *yanks*.

"adw	Deletes word into buffer <i>a</i> .
"ap	Puts the contents of buffer <i>a</i> after cursor.
"A5yy	Yanks and appends the next five lines to buffer <i>a</i> .
"ap	Puts contents of buffer <i>a</i> after cursor.

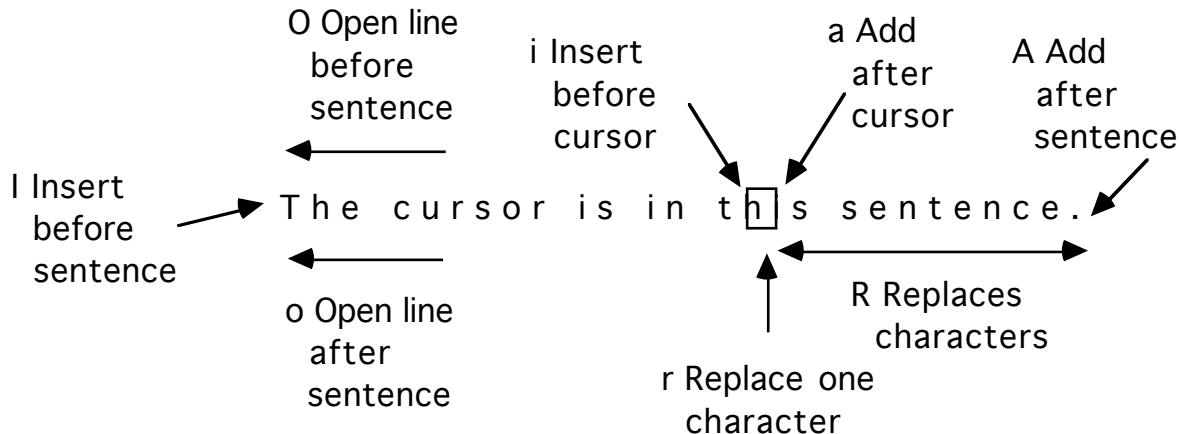
The buffers hold either words or lines; one cannot be appended to the other.

The *insert mode* allows text entry and replacement.

Any of the insert or change commands can be used to start a new file.

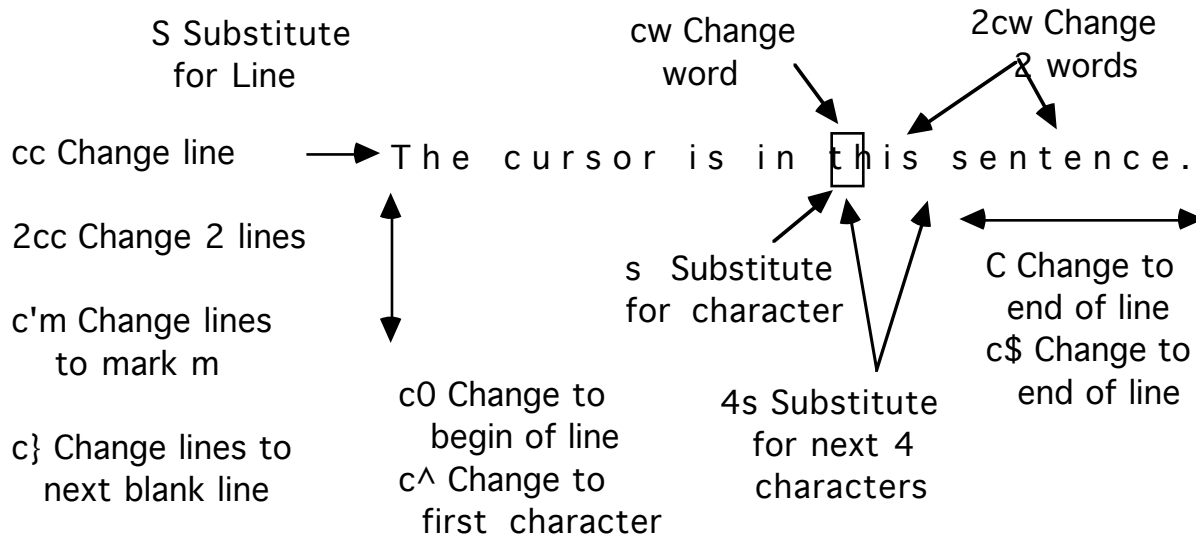
<u>ESCAPE</u>	Changes the system into command mode when you are in the insert mode.
<u>BACKSPACE</u>	Deletes the last character entered while in insert mode.
<u>CTRL W</u>	Deletes the last word entered (up to the last space) while in insert mode.
<u>CTRL U</u>	Deletes the last line entered.

### Insert Text



*Type text immediately after the command, and press Escape to return to Command mode.*

### Change Text



*Notice the \$, type text immediately after the command, and press Escape to return to Command Mode.*

The text that was changed is saved in the delete buffer and can be placed elsewhere.

*Line mode* offers extensive search and replace capability and file manipulation. The symbols /, ?, :, !, and Q place the cursor at the bottom of the screen and place *vi* in line mode.

: Places the cursor at the bottom of the screen and starts the line mode.

ESCAPE Returns the system to command mode when you are in line mode and aborts any unexecuted commands.

*Vi* returns to command mode automatically after the line mode commands are completed unless Q was used to place *vi* in line mode.

Q Places *vi* in line mode permanently.

vi Returns from permanent line mode.

/Level Searches for the first occurrence of *Level* toward the end of the file.

/ Searches for the next occurrence toward the end of the file.

? Searches for the next occurrence toward the start of the file.

:s/is/IS/ Replaces the first occurrence of *is* in the current line with *IS*.

://s Replaces the next occurrence.

:g/ is /s// IS /cg Searches for all occurrences of the word *is* (surrounded by blanks), checks for confirmation (y), and replaces them with the word *IS* (surrounded by blanks).

---

<code>:w</code>	Writes out the buffer on to the original file.
<code>:1,w first_part</code>	Writes out the first part of the buffer (up to and including the current line) and creates a new file.
<code>:\$w &gt;&gt; first_part</code>	Appends the last part of the buffer (starting with the current line) to a file.
<code>:e first_part</code>	Begins editing another file. Note the duplicate line.
<code>:e#</code>	Returns to the original file.
<code>:e!</code>	Starts editing an original version of the file.
<code>:f new_dante</code>	Changes the name of the output file.
<code>:cd mydir</code>	Changes the current directory in which files are accessed.
<code>:r myfile5</code>	Reads in the file myfile5 after the current line.
<code>:cd ..</code>	Changes the current directory in which files are accessed.
<code>:r!date</code>	Reads the output of the command date into the buffer at the cursor.
<code>!ls</code>	Displays the files in the current directory.
<code>o</code>	Opens a new line to insert text.
<code>532+694</code>	Enters a new line of text.
<code>ESCAPE</code>	Returns to command mode.

---

!!bc	Uses the current line as input to <i>bc</i> , and replaces the current line with the output of the command <i>bc</i> .
!}sort	Sorts the text to next blank line and replaces it with the sorted text.
:sh	Starts a new shell without quitting <i>vi</i> .
exit	Returns to the edited file.

A series of commands can be yanked or deleted into a buffer and then simply executed over and over again to avoid retyping.

o	Opens a new line to place text.
W~ <u>ESCAPE</u>	Enters as text the command to capitalize the next word and returns to command mode.
lb	Moves back to the beginning of the command.
"ad\$	Deletes the command text into buffer <i>a</i> . The end of line is not included.
@a	Executes contents of buffer <i>a</i> (as a macro).
@@	Executes last macro.

Keys can also be redefined to execute other commands. The use of function keys is great for saving long substitutions across files. These same commands can be placed in your *.exrc* file and automatically executed when *vi* or *ex* start up.

:map Y y\$	Identify the key <i>Y</i> as the command <i>y\$</i> , that is, redefine <i>Y</i> as <i>yank</i> to the end of the line so that it is similar to the commands <i>D</i> and <i>C</i> .
------------	--



- `:map #1 iandCTRL V ESCAPE` Identify the key `F1` to insert *and* before the cursor. `CTRL V` is used to protect the `ESCAPE` key from immediate action.
- `:map` Displays the key mappings.
- `:x` Exits *vi* after saving all changes.
- `:q` Exits *vi* when no changes have been made.
- `:q!` Exits *vi* without saving changes.