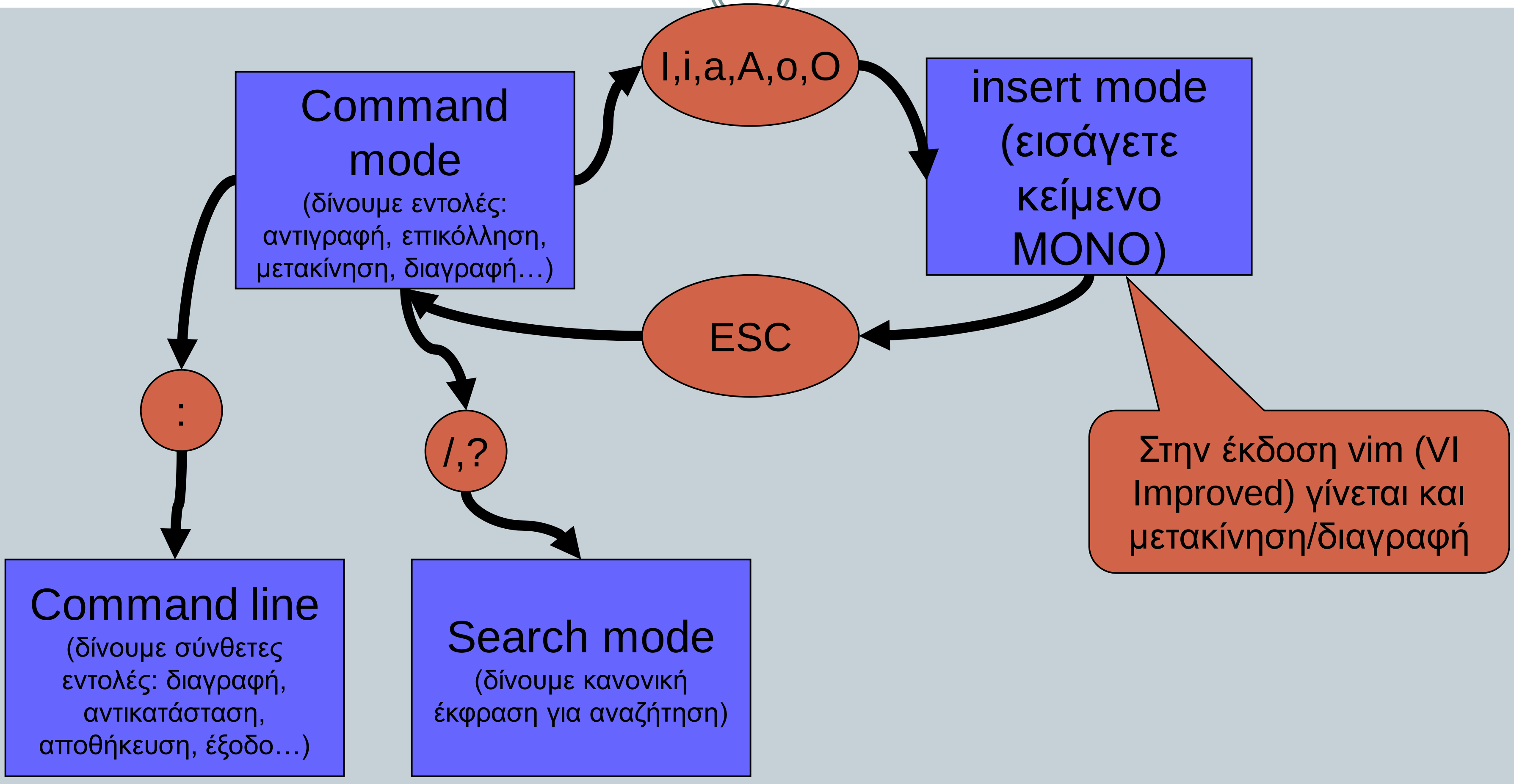
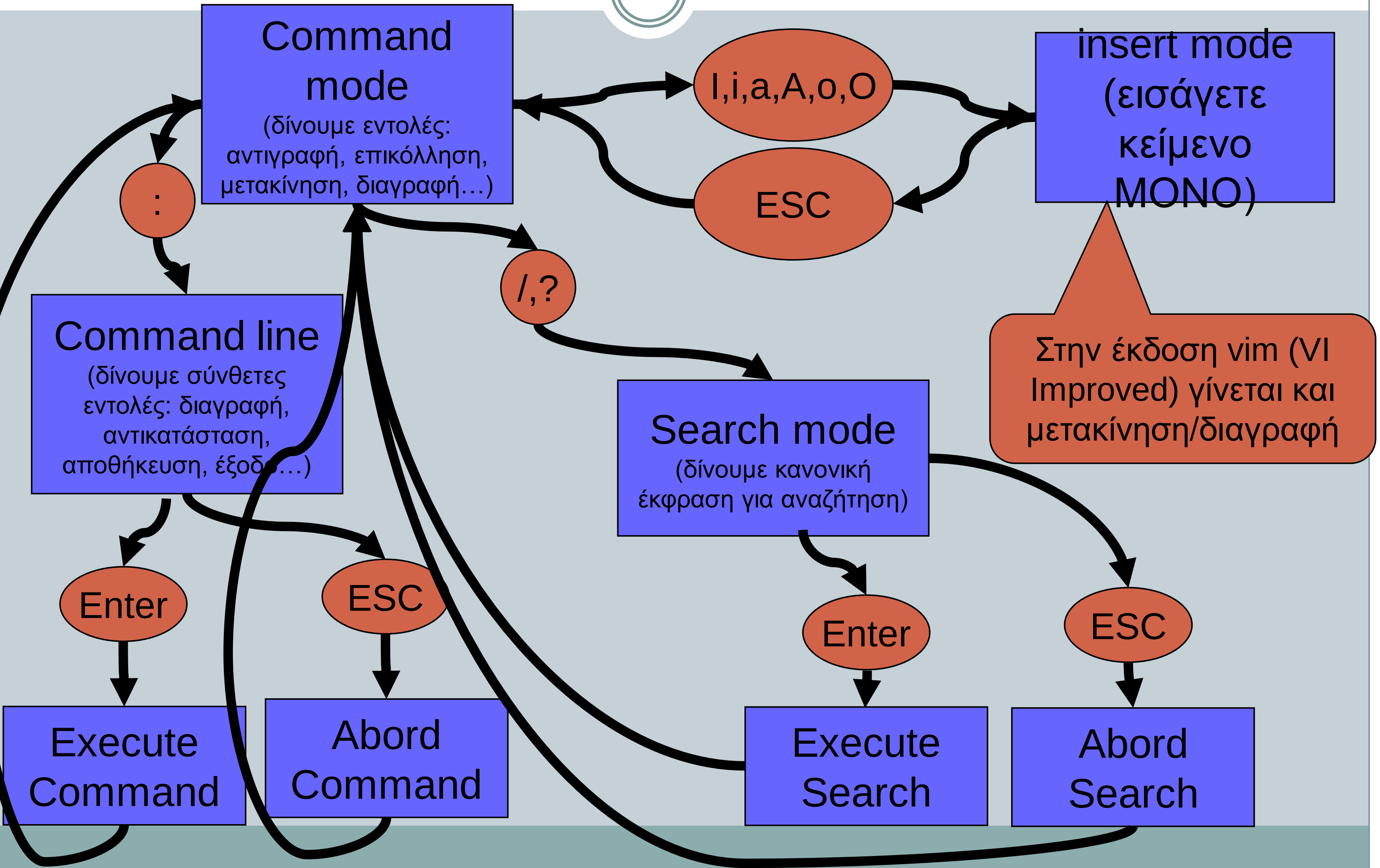
Εισαγωγή στα Λειτουργικά Συστήματα

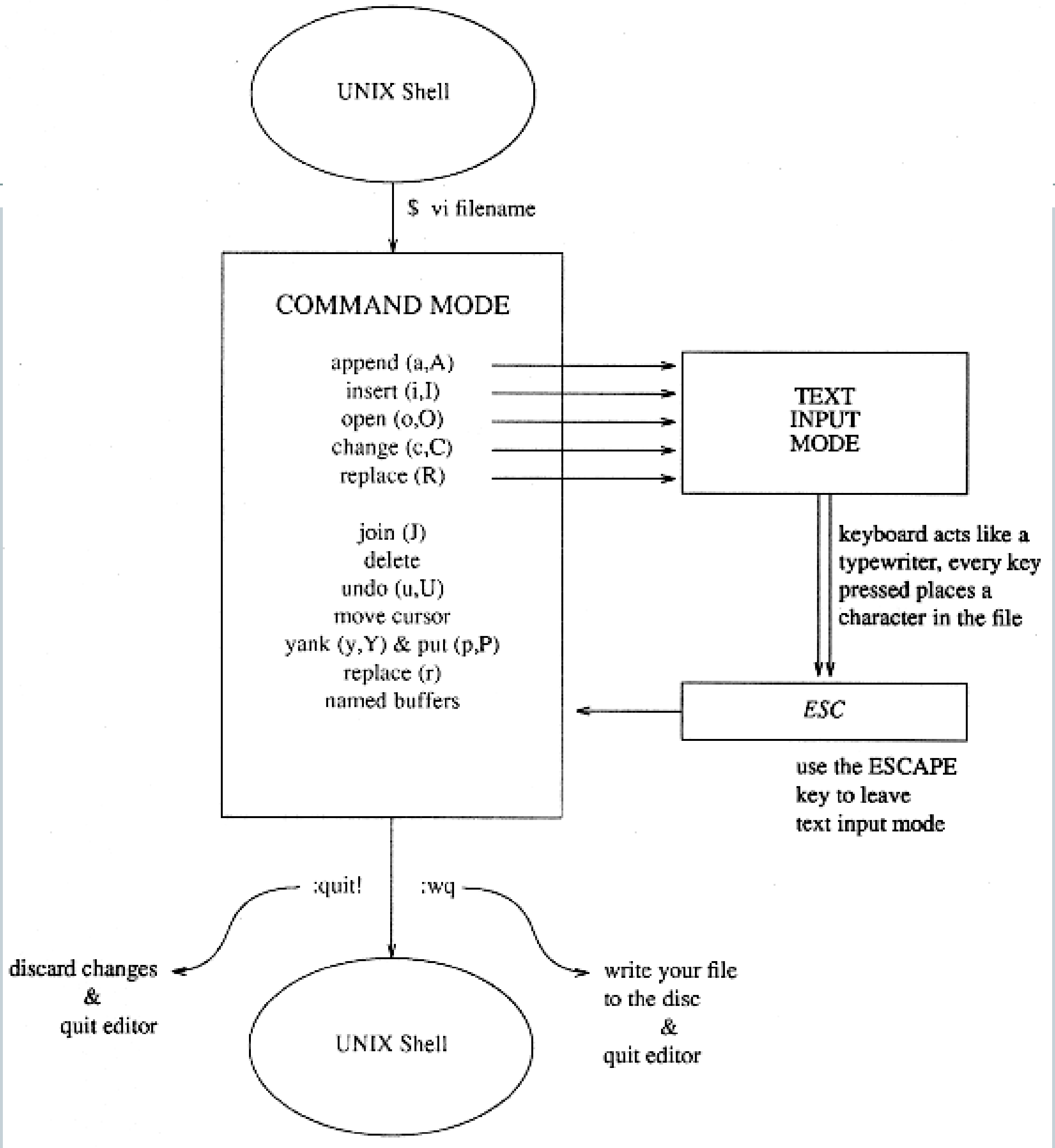
SET ΔΙΑΦΑΝΕΙΩΝ 6Α

VI

ΑΝΤΩΝΗΣ ΣΙΔΗΡΟΠΟΥΛΟΣ


Command mode
(δίνουμε εντολές: αντιγραφή, επικόλληση, μετακίνηση, διαγραφή…)
I,i,a,A,o,O
insert mode (εισάγετε κείμενο ΜΟΝΟ)
ESC
Στην έκδοση vim (VI Improved) γίνεται και μετακίνηση/διαγραφή
:
/,?
Command line
(δίνουμε σύνθετες εντολές: διαγραφή, αντικατάσταση, αποθήκευση, έξοδο…)
Search mode
(δίνουμε κανονική έκφραση για αναζήτηση)


Command mode (δίνουμε εντολές: αντιγραφή, επικόλληση, μετακίνηση, διαγραφή…)
I,i,a,A,o,O
insert mode (εισάγετε κείμενο ΜΟΝΟ)
ESC
:
/,?
Command line (δίνουμε σύνθετες εντολές: διαγραφή, αντικατάσταση, αποθήκευση, έξοδος…)
Στην έκδοση vim (VI Improved) γίνεται και μετακίνηση/διαγραφή
Search mode (δίνουμε κανονική έκφραση για αναζήτηση)
Enter
ESC
Enter
ESC
Execute Command
Abord Command
Execute Search
Abord Search


UNIX Shell
$ vi filename
COMMAND MODE
append (a,A)
insert (i,I)
open (o,O)
change (c,C)
replace (R)
join (J)
delete
undo (u,U)
move cursor
yank (y,Y) & put (p,P)
replace (r)
named buffers
TEXT INPUT MODE
keyboard acts like a typewriter, every key pressed places a character in the file
ESC
use the ESCAPE key to leave text input mode
:quit!
:wq
discard changes & quit editor
write your file to the disc & quit editor
UNIX Shell

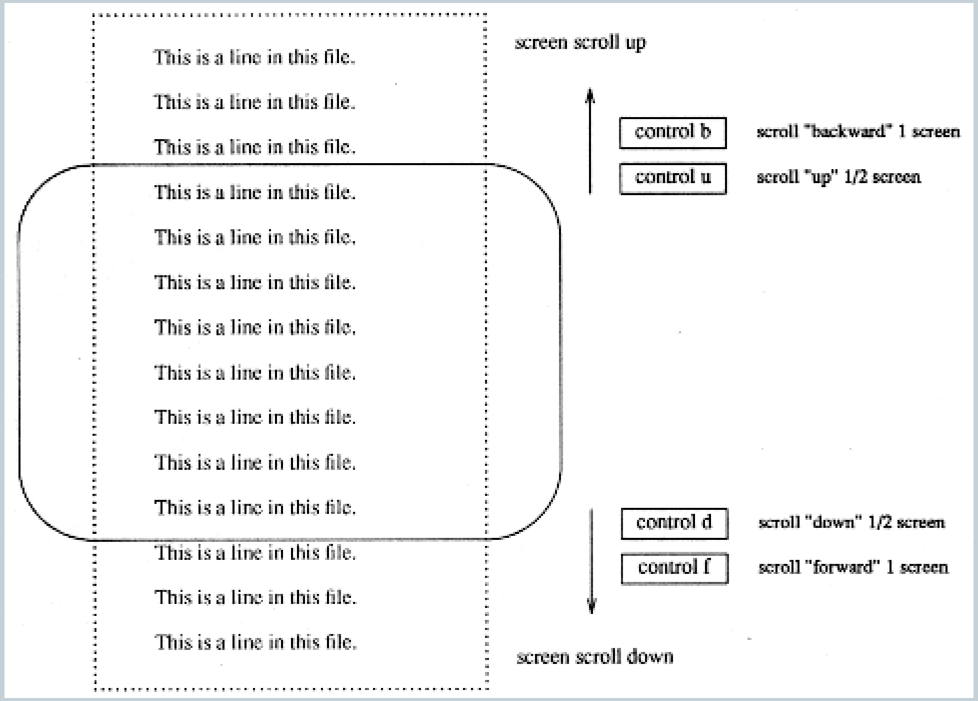
Text Input Mode

- To enter *text input mode* a user would type, while in *command mode*, the editing command to append (a, A), insert (i, I), open (o, O), change (c,C), or replace (R).
- The user leaves *text input mode* by depressing the "**ESC**" key.

Positioning Text on the Screen

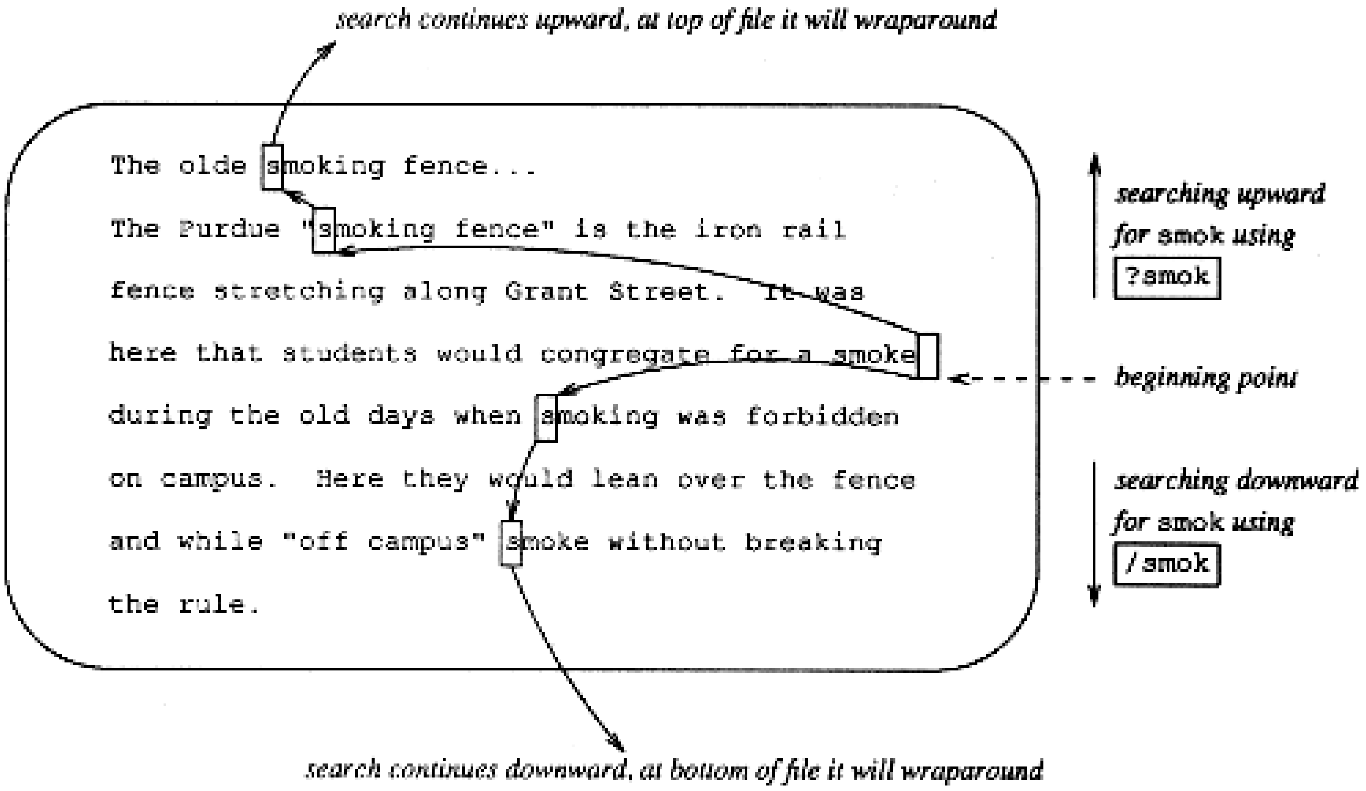
- the opening screen will show the cursor located in the home position.
- If **VI** is opening an existing file, you will see the first 20 to 24 lines of the file on the screen.
- All additional lines copied from the disk will also be in the buffer, but not viewable on the screen.

- ## Scrolling the Screen
  - exists a relationship between the command and its meaning

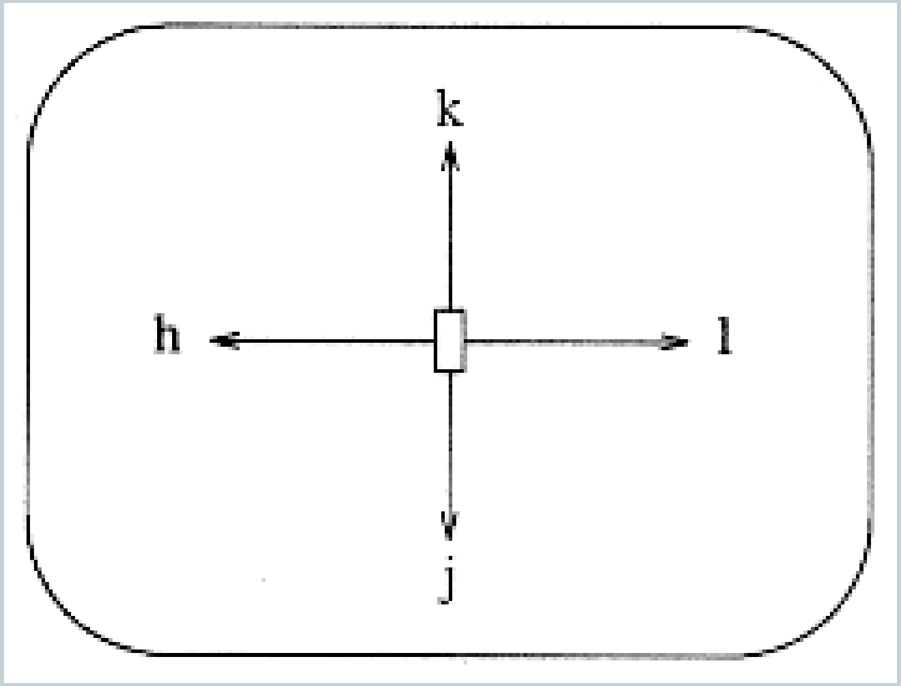


- Using the GOTO Command
  - If you know the line number or the general area you want to access, repositioning can be accomplished by using the "**G**" (goto) command. The goto command, preceded by a line number, such as "**250G**", will position the cursor on the first character space of line number 250. If the requested line is not currently on the screen, the screen will be redrawn with the requested line situated in the window.

- Moving by Searching


search continues upward, at top of file it will wraparound
The olde smoking fence...
The Purdue "smoking fence" is the iron rail
fence stretching along Grant Street. It was
here that students would congregate for a smoke
during the old days when smoking was forbidden
on campus. Here they would lean over the fence
and while "off campus" smoke without breaking
the rule.
searching upward
for smok using
?smok
beginning point
searching downward
for smok using
/smok
search continues downward, at bottom of file it will wraparound

- Positioning the Cursor
  - Major Screen Movements



Moving within the Line

Scope	Text Unit Encompassed
0	beginning of line (zero)
$	end of line
W w	word right
B b	word left
E e	end of word right

Text Insertion

- Append Command (a, A)
  - Pressing the "**a**" key will cause the cursor to move one space to the right of the current cursor position and await the new text
  - The "**A,**" form of append works in much the same way, except when the command is issued the cursor jumps to the end of the line and it is at this point that you enter *text input mode* and proceed to add text.
  - Pressing "**ESC**" returns you to *command mode* and the screen is redrawn.

Insert Command (i, I)

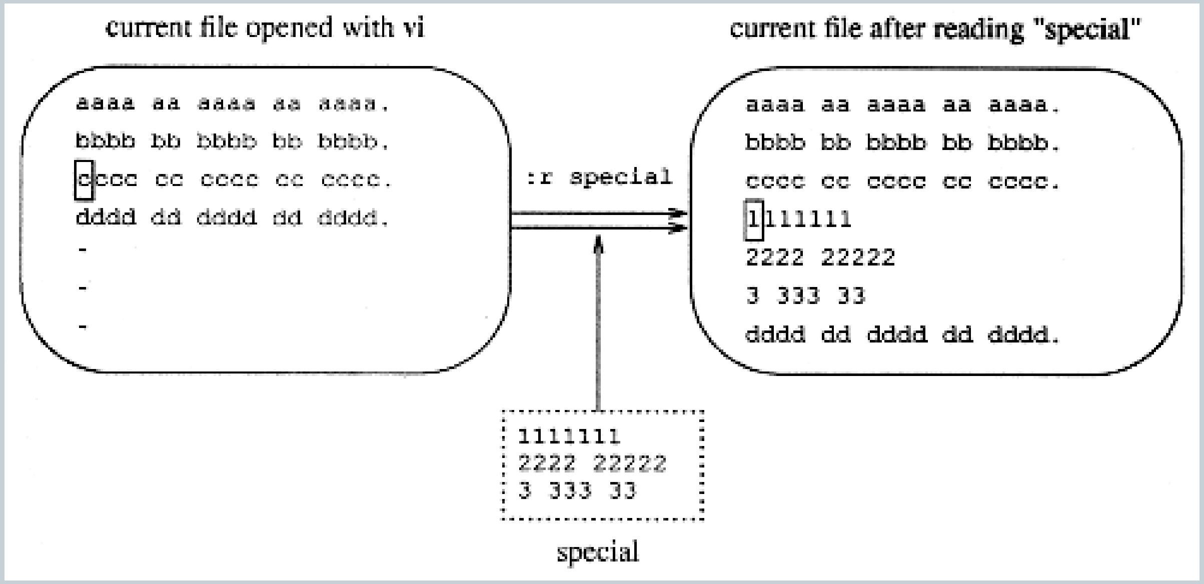
- The lower-case "**i**" command inserts the new text you type to the left of the cursor.
- The upper-case "**I**" command inserts the text at the beginning of the current line.
- By giving the insert command, *text input mode* is activated, allowing you to enter as much text as you want.
- To stop entering text you must press the "**ESC**" key.

Open Command (o, O)

- The open command is used to create a blank line in the file where additional text is to be typed in.
  - The lower-case "o" command opens a new line below the line the cursor is on and the upper-case "O" command opens a new line above the line the cursor is on.

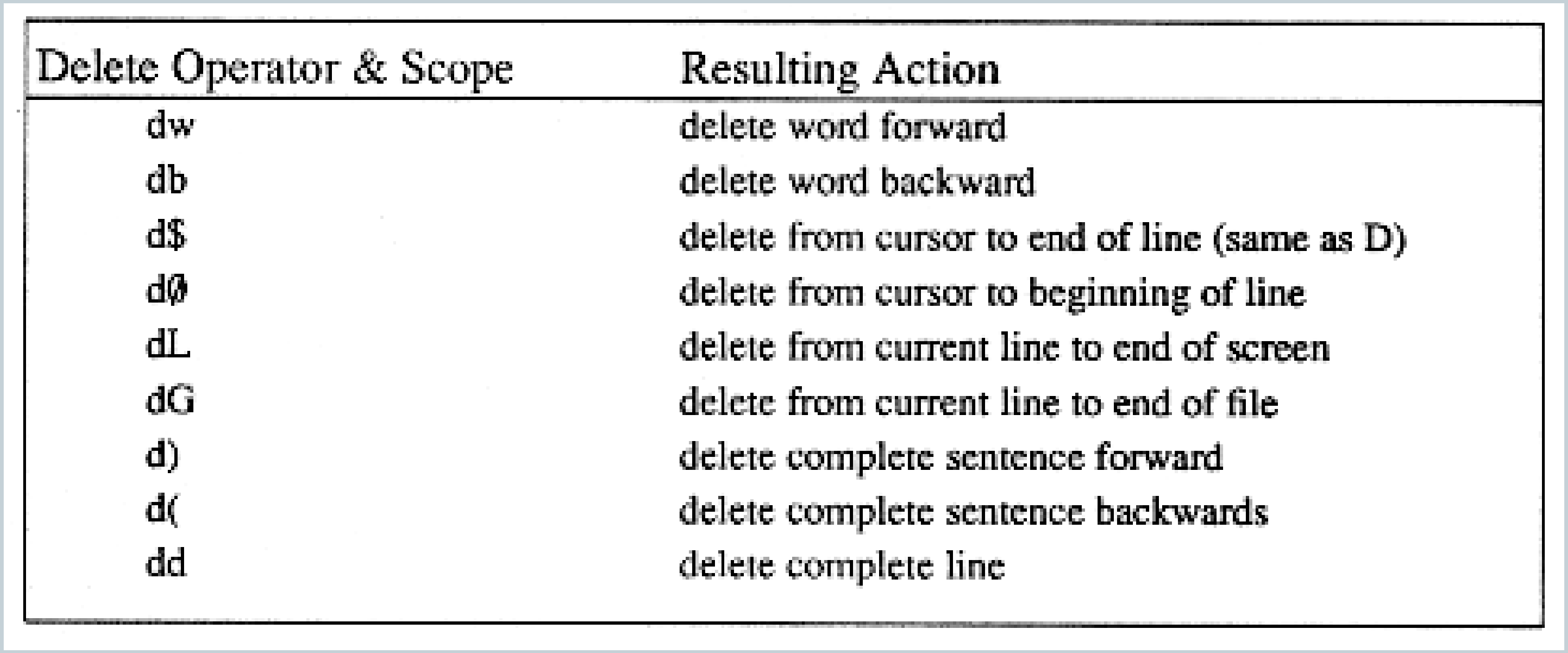
Read Command (:r)

- This command allows the user to place a copy of another file into the current file.
  - :r filename



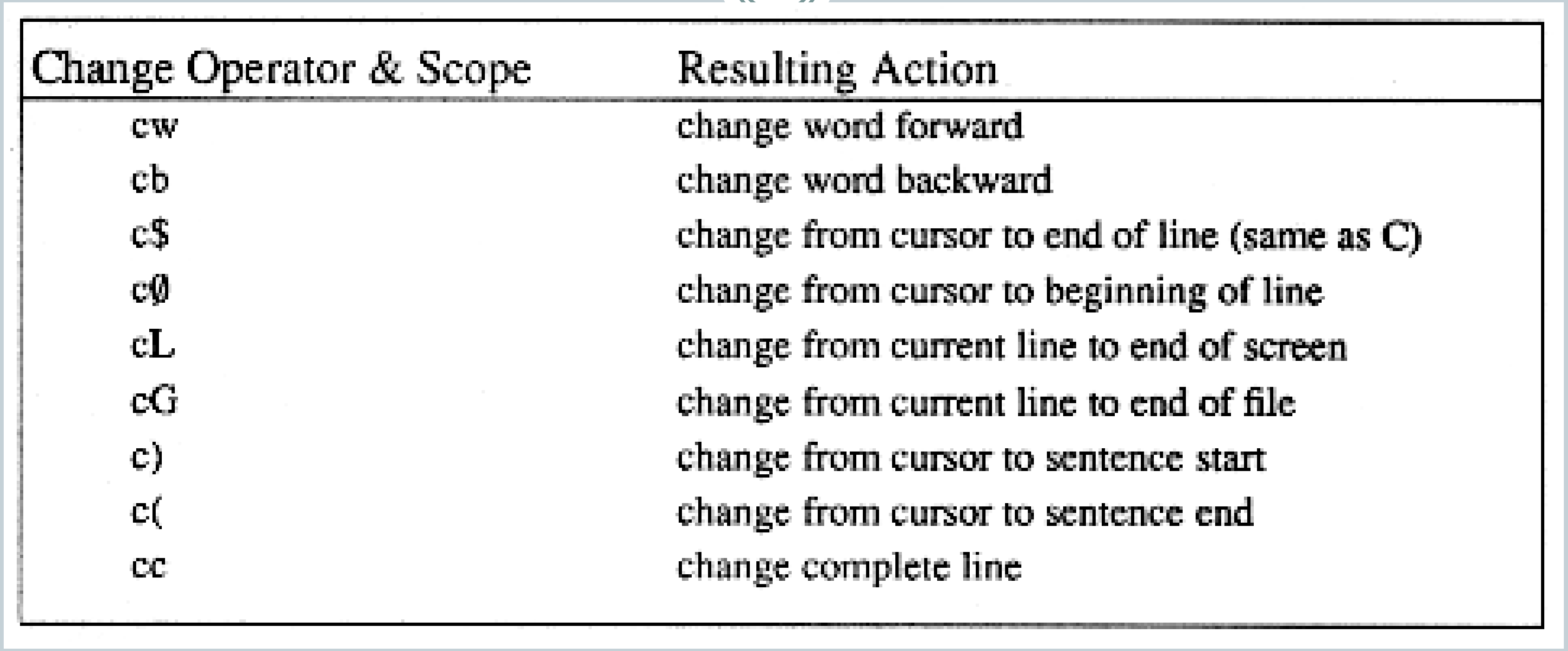
Delete Text (d, D)

- The delete command is used in *command mode* to remove portions of text from the file being edited. The delete command is available for use in either the upper-case "**D**" format as a command, or in the lower-case "**d**" format as an operator.
- **The upper-case D"** command removes the text on the current line from the cursor to the end of the line. Most people tend to forget the "**D**" and instead use the "**d**" with the addition of scopes.
- **The lower-case "d"** operator is very flexible because it can be used in conjunction with scopes to delete characters, words, and lines. The scope must be specified after the delete operator. Some of the most common scopes used with the delete operator shown in the next table.



Delete Operator & Scope	Resulting Action
dw	delete word forward
db	delete word backward
d$	delete from cursor to end of line (same as D)
d∅	delete from cursor to beginning of line
dL	delete from current line to end of screen
dG	delete from current line to end of file
d)	delete complete sentence forward
d(	delete complete sentence backwards
dd	delete complete line

Change Text (c, C)



Change Operator & Scope	Resulting Action
cw	change word forward
cb	change word backward
c$	change from cursor to end of line (same as C)
c∅	change from cursor to beginning of line
cL	change from current line to end of screen
cG	change from current line to end of file
c)	change from cursor to sentence start
c(	change from cursor to sentence end
cc	change complete line

Replace Command (r, R)

- The replace command is used to replace portions of text on the screen with new characters in an overlay fashion.
- The lower-case "**r**" command replaces the single character with the new character you key in.
- The upper-case "**R**" command replaces characters on the screen one at a time with characters you type in. Once you type "**R**", *text input mode* is activated permitting you to enter as much text as you want.
- It is important to remember that for each character you key in one is removed from the file until the end on the current line is reached

Erase Command (x, X)

- The lower-case "**x**" command will erase or "eat" the character the cursor is sitting on.

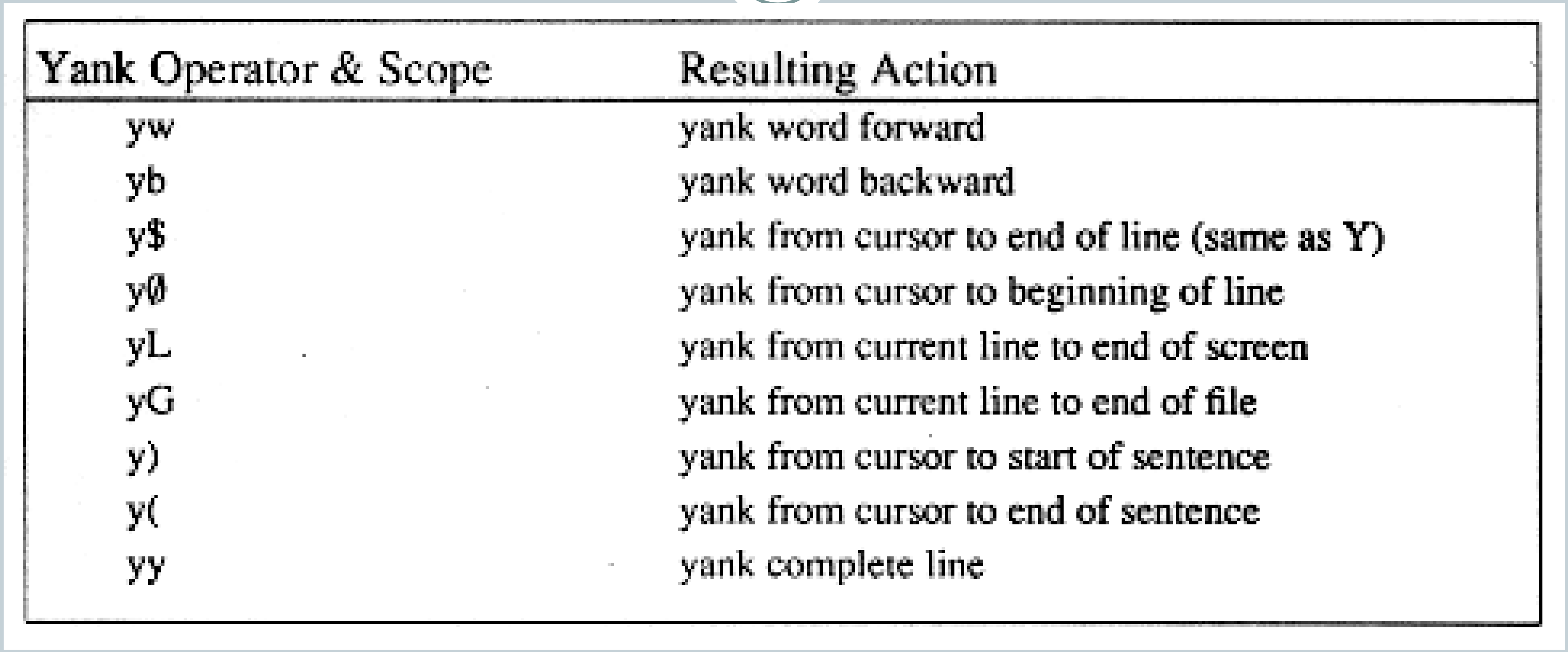
Undo Command (u, U)

- The lower-case "**u**" (undo) command undoes the effects of the last command (only one) that you issued.
- The upper-case "**U**" undoes all of the changes that you have made to the current line.

Rearranging and Duplicating Text

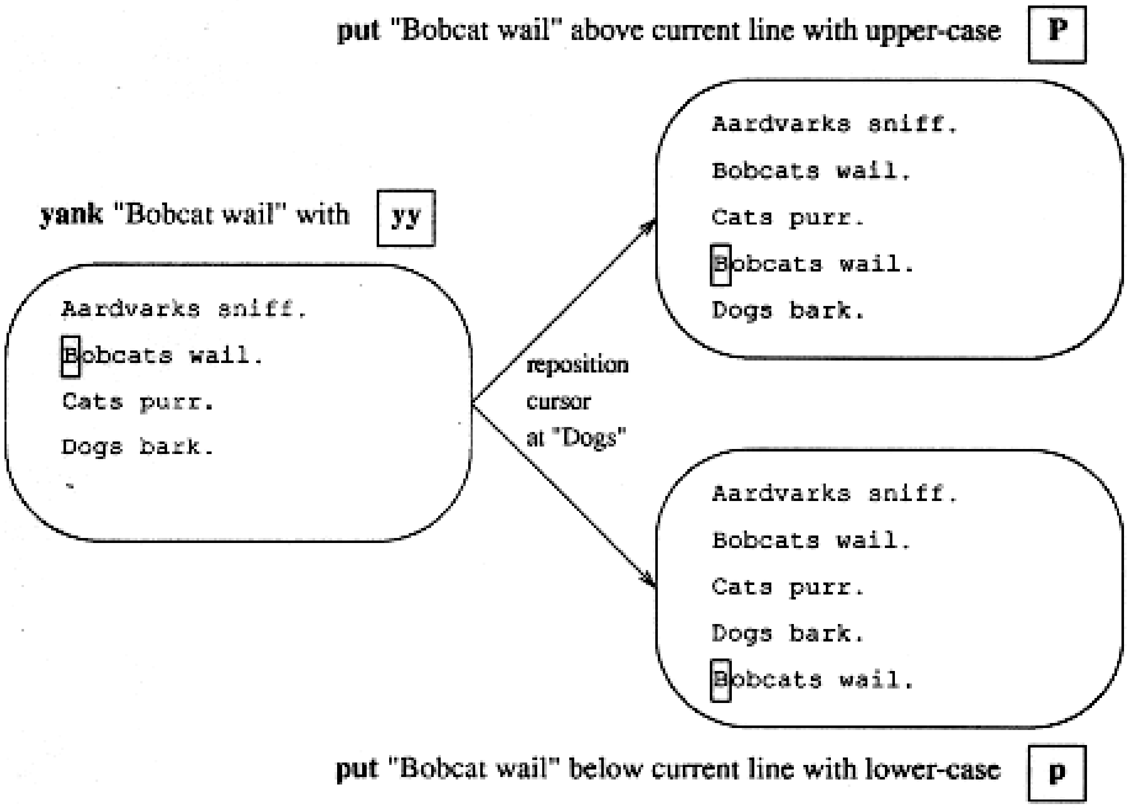
- Copying Text and Moving the Copy
  - **Step 1: Copying Text with the Yank Command (y, Y)**

    The yank command is used to make copies of words, lines, and sections of text being edited and place them into the unnamed buffer associated with the editor. The yank command is available for use in either the lower-case "**y**" format as an operator, or the upper-case "**Y**" format as a command. The lower-case "**y**" can be thought of as a yank operator which will combine with scopes to make an operator-scope command much in the same way as the delete and change operators. The scope to be yanked must be specified after the "**y**" command.



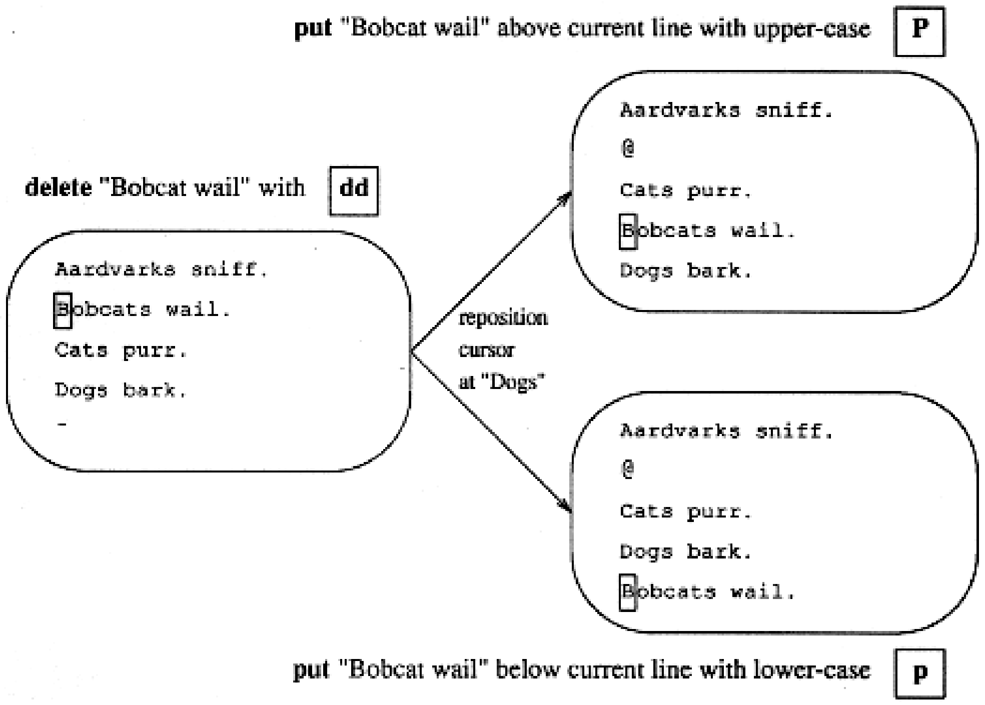
Yank Operator & Scope	Resulting Action
yw	yank word forward
yb	yank word backward
y$	yank from cursor to end of line (same as Y)
y0	yank from cursor to beginning of line
yL	yank from current line to end of screen
yG	yank from current line to end of file
y)	yank from cursor to start of sentence
y(	yank from cursor to end of sentence
yy	yank complete line

Step 2: Relocating the Copy with the Put Command (p, P)



Deleting Text and Moving It

- Step 1: Removing Text with the Delete Command (d, D)
- Step 2: Relocating the Deleted Text with the Put Command (p, P)


put "Bobcat wail" above current line with upper-case P
Aardvarks sniff.
@
Cats purr.
Bobcats wail.
Dogs bark.
delete "Bobcat wail" with dd
Aardvarks sniff.
Bobcats wail.
Cats purr.
Dogs bark.
-
reposition cursor at "Dogs"
Aardvarks sniff.
@
Cats purr.
Dogs bark.
Bobcats wail.
put "Bobcat wail" below current line with lower-case p

Miscellaneous Information

- Creating Line Numbers
  - :%nu
  - :set number
  - :set nonu
- Temporarily Interrupting VI
  - The current editing session may be interrupted by then typing "**:!**" followed by the desired command. When the requested command action is completed, the message:

    [Hit return to continue]

    will appear at the bottom of the screen. After pressing "**RETURN**", you will be returned to the **VI** editor to the same location you were at when you temporarily interrupted the editing session.

- Editing Multiple Files Using VI
  - The **VI** editor provides an advanced feature which allows a user to invoke the editor and then edit multiple files by use of the "**:e**" (edit) command. This ability to access multiple files without leaving the editor permits a user to look up information in another file without exiting the editor.

- When **VI** is invoked, a work area called a buffer is created for editing purposes. It is into this work space that a copy of a specified disk file is placed. The editor permits only one file copy in this buffer space at a time. Thus after making changes to a file (delete, add, or change), you must inform the editor what you wish done to the current buffer contents before you will be permitted to bring another file into this space. You do this by use of the "**:w**" (write current buffer contents to opened file), "**:e!\ newfile**" (toss current buffer contents, no update to opened file, and place a copy of newly called file in buffer), or "**:quit!**" (exit editor and toss buffer and buffer contents). The editor is smart enough to know that if all you have done is copy or read from a file, it can dispose of the unneeded buffer copy without further instructions from you when a new file is called.

- When you have two files open, **VI** permits toggling between files by use of "**:e\ #**". This works because whenever **VI** sees the character "#" used in a command where a filename is expected, it substitutes the "#" with the name of the previous file. For example if you had been in *apples* then opened *oranges*, the command "**:e #**" would return you to where you were in the *apples* file. Repeat "**:e #**" and you would be back in *oranges*.