SOFTENGER
Strive to Excel

# Vi Editor

## Unix

# Text Editor

SOFTENGER
Strive to Excel

1. A text editor is a program that can be used to create and modify text files.

2. One of the most popular editors on Unix systems is vi.

3. Some of the other editors are ed , pico , emacs .

# Starting vi

1. Open a file with vi. Type: vi myfile.txt
2. If myfile.txt does not exist, a screen will appear with just a cursor at the top followed by tildes (~) in the first column.
3. If myfile.txt does exist, the first few line of the file will appear.
4. The status line at the bottom of your screen shows error messages and provides information and feedback, including the name of the file.

# vi Modes

- Command Mode

- Command mode is the mode you are in when you start (default mode)

- Command mode is the mode in which commands are given to move around in the file, to make changes, and to leave the file

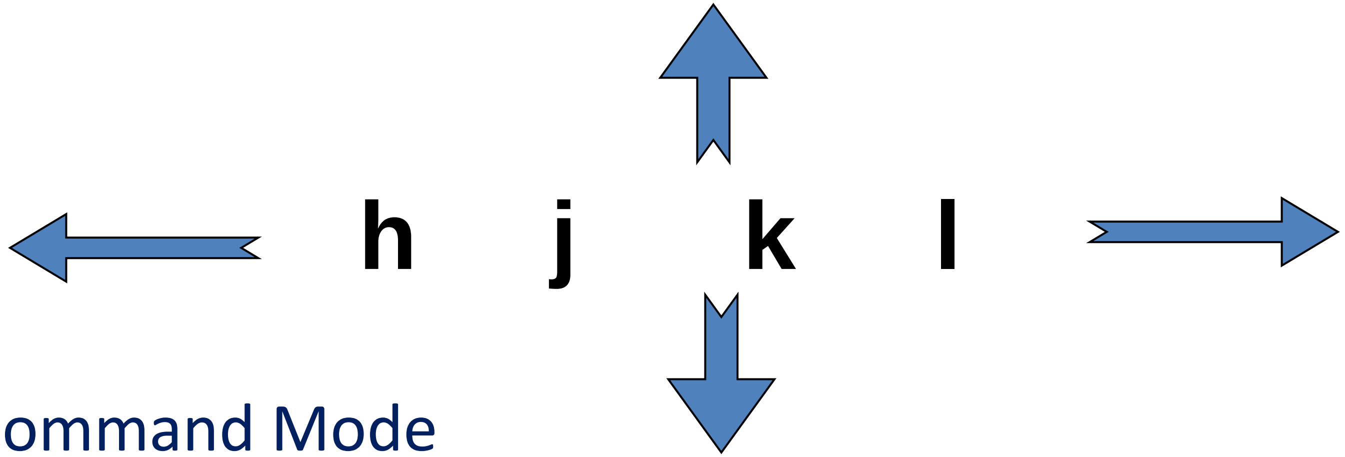
- Commands are case sensitive: j not the same as J

- Most commands do not appear on the screen as you type them. Some commands will appear on the last line: : / ?

# vi Modes

- Insert (or Text) Mode
- The mode in which text is created. (You must press <Return> at the end of each line unless you've set wrap margin.)
- There is more than one way to get into insert mode but only one way to leave: return to command mode by pressing <Esc>

# Basic Cursor Movement



**h j k l**

From Command Mode

k Up one line j Down one line

h Left one character l Right one character (or use <Spacebar>)

w Right one word b Left one word

# Entering, Deleting, and Changing Text

SOFTENGER
Strive to Excel

**Command Mode**

| | |
|---|---|
| i | *Enter text entry mode* |
| x | *Delete a character* |
| dd | *Delete a line* |
| r | *Replace a character* |
| R | *Overwrite text, press <Esc> to end* |

# Exiting vi

To exit you must be in command mode-press <Esc> if you are not in command mode

You must press <Return> after commands that begin with a : (colon)

*From Command Mode*

| | |
|---|---|
| ZZ | *Write (if there were changes), then quit* |
| :wq | *Write, then quit* |
| :q | *Quit (will only work if file has not been changed)* |
| :q! | *Quit without saving changes to file* |

# More On Cursor Movement

SOFTENGER
Strive to Excel

*From Command Mode*

**e** Move to end of current word

**$** Move to end of current line

^ Move to beginning of current line

**+** Move to beginning of next line

**-** Move to beginning of previous line

**G** Go to last line of the file

**:**n Go to line with this number (:10 goes to line 10)

# More On Cursor Movement

**<Ctrl>d** *Scroll down one-half screen*

**<Ctrl>u** *Scroll up one-half screen*

**<Ctrl>f** *Scroll forward one full screen*

**<Ctrl>b** *Scroll backward one full screen*

# More On Cursor Movement

| | |
|---|---|
| **)** | *Move to the next sentence* |
| **(** | *Move to the previous sentence* |
| **}** | *Move to the next paragraph* |
| **{** | *Move to the previous paragraph* |
| **H** | *Move to the top line of the screen* |
| **M** | *Move to the middle line of the screen* |
| **L** | *Move to the last line of the screen* |
| **%** | *Move to matching bracket: ( { [ ] } )* |

# Entering Text Mode

*From Command Mode*

**i** *Insert text before current character*

**a** *Append text after current character*

**I** *Begin text insertion at the beginning of a line*

**A** *Append text at end of a line*

**o** *Open a new line below current line*

**O** *Open a new line above current line*

# Commands and Objects

| **Format** | **Example** |
|---|---|
| *operator number object* | **c2w** |
| *number operator object* | **2cw** |

**Operators**

| | |
|---|---|
| **c** | *change* |
| **d** | *delete* |
| **y** | *yank* |

# Commands and Objects

## Objects and Locations

| | |
|---|---|
| **w** | *one word forward* |
| **b** | *one word backward* |
| **e** | *end of word* |
| **^, $** | *beginning of line, end of line* |
| **/pattern/** | *forward to pattern* |

# Replacing and Changing Text

*From Command Mode*

**r** Replace only the character under the cursor.
(Note: using r you remain in command mode.)

**R** Beginning with the character under the cursor, replace as many characters on this line as you want. (You are in overtype mode until you press <Esc>

**cw** Beginning with the character under the cursor, change a word to whatever you type. (You are in insert mode until you press <Esc>)

**c$**
**C** Beginning with the character under the cursor, change a line to whatever you type. (You are in insert mode until you press <Esc>)

# Deleting Text

*From Command Mode*

| | |
|---|---|
| **x** | *Delete a character* |
| **dw** | *Delete an alphabetic word and the following space*<br>*(6dw deletes six words)* |
| **dW** | *Delete a blank-delimited word and the following space* |
| **dd** | *Delete a line (6dd deletes six lines)* |
| **d$**<br>**D** | *Delete all characters to the end of the line.* |
| **d}** | *Delete all characters to the end of the paragraph.* |
| **:5,30d** | *Delete lines 5 through 30* |

# Copying and Pasting Text

SOFTENGER
Strive to Excel

*From Command Mode*

| | |
|---|---|
| **yy** | *Copy (yank) the current line* |
| **6yy** | *Copy (yank) six lines, beginning with the current line* |
| **yw** | *Copy the current word* |
| **p** | *Put the text after the cursor position* |
| **P** | *Put the text before the cursor position* |

# Copying and Pasting Text

Copied text goes into a temporary buffer that is replaced each time you copy (or delete) more text. Only the current contents of the temporary buffer can be put back into your file. As a result, when you use copy (y), use the put (p) command immediately.

A yank and put procedure using colon commands:

**1. :5,10y** *Copy lines 5-10*

*2. Move cursor*

**3. :put** *Put after cursor*

# Other Useful Commands

*From Command Mode*

**.** *Repeat last command*

*n***.** *Repeat last command n number of times*

**J** *Join next line to current line*

**u** *Undo last single change*

**~** *Change letter's case (capital to lower and vice versa)*

# Copying, Deleting, or Moving Text Using Line Numbers

These commands start with a colon (**:**) and end with a **<Return>** or **<Enter>**

**<Ctrl>g** shows the line number of the current line

The basic form of colon commands is

**:beginning_line, ending_line command destination**

where destination is the line after which you want the text placed.

*From Command Mode*

**:5,10 co 105** *Copy lines 5-10 to the line after 105*

**:5,20 m $** *Move lines 5-20 to end of file*

**:7,300 d** *Delete lines 7-300 (to buffer)*

# Searching for Text

*From Command Mode*

| | |
|---|---|
| */text* | *Search forward (down) for text (text can include spaces and characters with special meanings.)* |
| **?** | *text Search backward (up) for text* |
| **n** | *Repeat last search in the same direction* |
| **N** | *Repeat last search in the opposite direction* |
| **f***char* | *Search forward for a charcter on current line* |
| **F***char* | *Search backward for a character on current line* |
| **;** | *Repeat last character search in the same direction* |
| **%** | *Find matching ( ), { }, or [ ]* |

# Substitutions

The simplest way to do substitutions over a range of lines, or throughout the file, is to use the **s** colon command.

```
:n1,n2s/old/new/gc
```

Where ,

*n1 is the beginning line*

*n2 is the ending line number*

**s** *means to substitute text matching the pattern (old) with text specified by (new)*

**g** *(global) is optional. If you do not use* **g**, *the editor substitutes only the first occurrence on the indicated lines.*

**c** *(confirm) is optional. It indicates you want to confirm each substitution before vi completes it.*

# Substitutions

*From Command Mode*

**:%s/old/new/g** *Substitutes old with new throughout the file*

**:.,$s/old/new/g** *Substitutes old with new from the current cursor position to the end of the file*

**:^,.s/old/new/g** *Substitutes old with new from the beginning of the file to the current cursor position*

**:&** *Repeats the last substitute (:s) command*

# More About Options

*From Command Mode-within vi for the current file only*

| | |
|---|---|
| **:set all** | *Display all options* |
| **:set** | *Display current settings of options* |
| **:set nooption** | Unset option |
| **:set ai** | *Set Auto Indentation during text entry* |
| **:set ic** | *Set Ignore Case during searches* |
| **:set nonu** | *Show line Numbers* |
| **:set nu** | *Show line Numbers* |
| **:set sm** | *Show Matching ( or { when ) or } is entered* |

# Customizing vi Sessions

Options can be set in following  ways:

1. During a vi session

```
:set nu
```

2. In a **.vimrc** file in your home directory.

*Sample contents of a .vimrc file*

```
set nu
set ai
```

# Creating a .vimrc File

1. At the system prompt, type: **vi .vimrc**

2. Type the following commands, each on a separate line:

```
set ai
set ic
set nu
```

Do not leave blank lines at the beginning or end of the **.vimrc** file.

3. When you are finished, type: **<Esc> ZZ**

# Issuing Shell Commands From vi

You can issue a single shell command while in the vi editor. For example, to list the files in your directory (ls), follow these steps:

*From Command Mode*

**:w** *Write changes to your file (just in case).*

**:!ls** *List contents of your current directory on the screen.*

*Press <Return> to return to vi.*

# Issuing Shell Commands From vi

You can issue many shell commands by temporarily leaving the vi editor.

*From Command Mode*

**:w** *Write changes to your file.*

**:sh** *Return to the shell to enter a number of commands without leaving vi.*

*Press <Control>d to return to vi editing.*