
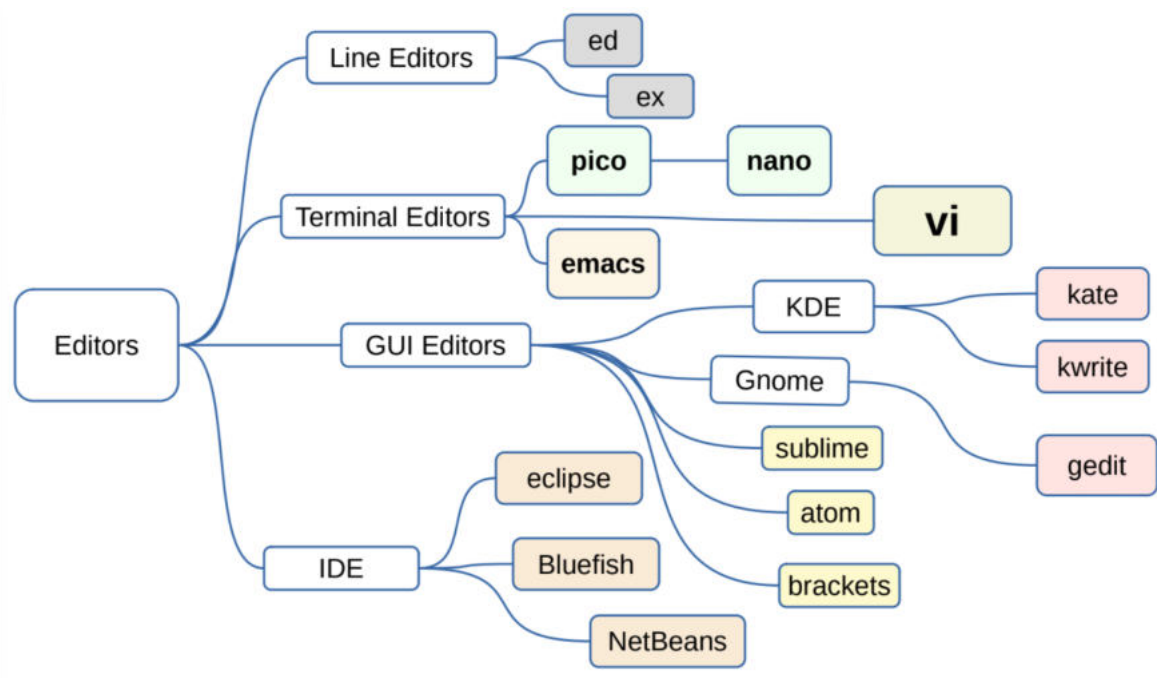


CLI Editors

| | |
|---------------|---|
| ▼ Type |  Lecture |
| 📅 Date | @February 22, 2022 |
| ☰ Lecture # | ? |
| 🔗 Lecture URL | |
| 🔗 Notion URL | https://21f1003586.notion.site/CLI-Editors-77dfbc4e33a9426498d94207eaea1962 |
| # Week # | 5 |

Command line editors

Working with text files in the terminal



Features

- Scrolling, view modes, current position in the file
- Navigation (char, word, line, pattern)
- Insert, Replace, Delete
- Cut-Copy-Paste
- Search-Replace
- Language-aware syntax highlighting
- Key-maps, init scripts, macros
- Plugins

ed

| | |
|--|--------------------------|
| Show the Prompt | P |
| Command Format | [addr[,addr]]cmd[params] |
| commands for location | 2 . \$ % + - , ; /RE/ |
| commands for editing | f p a c d i j s m u |
| execute a shell <i>command</i> | ! <i>command</i> |
| edit a file | e <i>filename</i> |
| read file contents into buffer | r <i>filename</i> |
| read <i>command</i> output into buffer | r ! <i>command</i> |
| write buffer to filename | w <i>filename</i> |
| quit | q |

Command Format → [starting-address[,ending-address]command[command-parameters]]

To locate the cursor on any particular line of the file ...

We can use the line number itself

- press **2** → in the 2nd line of the text file
- press **.** (**dot**) → referring to the current line the cursor is
- press **\$** → refers to the last line
- press **%** → refers to all the lines, i.e. any action we are doing applies to all the lines
- press **+** → line after the cursor
- press **-** (**minus sign**) → line before the cursor
- press **,** (**comma**) → represent the entire buffer, i.e. the whole file
- press **;** (**semicolon**) → refers the end of the text file from the current position
- **/RE/** → To match a specific Regular Expression in the file

A way to invoke the **ed** editor

```
ed file.txt
```

and it shows the number of bytes in the file, instead of the contents of the file ... ayo wut



```

117
P This gives a prompt
*1 Move the cursor to line #1
line-1 hello world
*$ Move the cursor to end of the buffer
line-4 end of file
*,p Show all the lines in the file
line-1 hello world
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of file
*2,3p Show lines from 2 to 3
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
*/hello/ Find the line with the pattern "hello"
line-1 hello world
*/oldest/ Find the line with the pattern "oldest"
line-3 ed is perhaps the oldest editor out there
*1 Move to line #1
line-1 hello world
*+ Move to the next line
line-2 welcome to line editor
*- Move to the previous line
line-1 hello world
*3 Move to line #3
line-3 ed is perhaps the oldest editor out there
*;p Get all the lines from current line to the end of file
line-3 ed is perhaps the oldest editor out there
line-4 end of file
*%p Display all the lines of the file
line-1 hello world
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of file
*. Get the current line
line-4 end of file

```

Run a bash command, read the bash command's output and write it to the file

```
~/Documents/week5 ed test.txt
117
P
*!date
Tuesday 01 February 2022 09:03:06 PM IST
!
*r !date
41
*w
158
*,p
line-1 hello world
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of file
Tuesday 01 February 2022 09:03:11 PM IST
*
```

Delete the last line of the file and write the changes to the file

```
~/Documents/week5 ed test.txt
158
,p
line-1 hello world
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of file
Tuesday 01 February 2022 09:03:11 PM IST
$
Tuesday 01 February 2022 09:03:11 PM IST
.d
P
*,p
line-1 hello world
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of file
*w
117
*q
```


Append a line to the file (press . then enter to exit)

```
~/Documents/week5 ed test.txt
117
P
*,p
line-1 hello world
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of file
*1
line-1 hello world
*a
appended this line after line-1
.
*,p
line-1 hello world
appended this line after line-1
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of file
*|
```

Search and Replace

```
*2
appended this line after the first line
*s/appended/Appended
Appended this line after the first line
*,p
line-1 hello world
Appended this line after the first line
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of file
```

File name

```
*f
test.txt
```

Print the current line

```
*p
line-4 end of file
```

Append to the current line

```
*a
This line is appended at the end of the file
.
```

Join line 5 and 6 (The command is `5,6j`)

```
*5
line-4 end of file
*5,6j
*p
line-4 end of fileThis line is appended at the end of the file
*.
line-4 end of fileThis line is appended at the end of the file
```

Move the current line after the given line # (We are moving the current line below line 1 here)

```
*m1
*,p
line-1 hello world
line-4 end of fileThis line is appended at the end of the file
Appended this line after the first line
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
```

Undo the previous operation


```
*u
*,p
line-1 hello world
Appended this line after the first line
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of fileThis line is appended at the end of the file
```

Add a prefix to all the lines and then modify the prefix on some lines

```
~/Documents/week5 > ed test.txt
201
P
*,p
line-1 hello world
Appended this line after the first line
line-2 welcome to line editor
line-3 ed is perhaps the oldest editor out there
line-4 end of fileThis line is appended at the end of the file
*%s/\(.*\) /PREFIX \1/
*,p
PREFIX line-1 hello world
PREFIX Appended this line after the first line
PREFIX line-2 welcome to line editor
PREFIX line-3 ed is perhaps the oldest editor out there
PREFIX line-4 end of fileThis line is appended at the end of the file
*3,5s/PREFIX/prefix/
*,p
PREFIX line-1 hello world
PREFIX Appended this line after the first line
prefix line-2 welcome to line editor
prefix line-3 ed is perhaps the oldest editor out there
prefix line-4 end of fileThis line is appended at the end of the file
```

ed/ex commands

| | |
|---|-------------------------------------|
| f | show name of file being edited |
| p | print the current line |
| a | append at the current line |
| c | change the line |
| d | delete the current line |
| i | insert line at the current position |
| j | join lines |
| s | search for regex pattern |
| m | move current line to position |
| u | undo latest change |

readlink

Prints the resolved symbolic links or canonical file names, *but what does that mean?*

If we have file which is a symbolic link to file which in turn is another symbolic link to a file and so on ...

The `readlink` command will display the actual file the initial symbolic link is referring to

Example usage

```
readlink -f /usr/bin/pico
```

Output

```
/usr/bin/nano
```

.bashrc

It is a config file that is read by the bash shell everytime it opens

nano

`nano` is a text editor, example syntax is `nano filename`

It also does syntax highlighting

File handling

- Ctrl+S Save current file
- Ctrl+O Offer to write file ("Save as")
- Ctrl+R Insert a file into current one
- Ctrl+X Close buffer, exit from nano

Editing

- Ctrl+K Cut current line into cutbuffer
- Alt+6 Copy current line into cutbuffer
- Ctrl+U Paste contents of cutbuffer
- Alt+T Cut until end of buffer
- Ctrl+] Complete current word
- Alt+3 Comment/uncomment line/region
- Alt+U Undo last action
- Alt+E Redo last undone action

Search and replace

| | |
|--------|-------------------------------|
| Ctrl+Q | Start backward search |
| Ctrl+W | Start forward search |
| Alt+Q | Find next occurrence backward |
| Alt+W | Find next occurrence forward |
| Alt+R | Start a replacing session |

Deletion

| | |
|----------|--------------------------------|
| Ctrl+H | Delete character before cursor |
| Ctrl+D | Delete character under cursor |
| Alt+Bsp | Delete word to the left |
| Ctrl+Del | Delete word to the right |
| Alt+Del | Delete current line |

Operations

| | |
|--------|--------------------------------|
| Ctrl+T | Execute some command |
| Ctrl+J | Justify paragraph or region |
| Alt+J | Justify entire buffer |
| Alt+B | Run a syntax check |
| Alt+F | Run a formatter/fixer/arranger |
| Alt+: | Start/stop recording of macro |
| Alt+; | Replay macro |

Moving around

| | |
|-----------|------------------------|
| Ctrl+B | One character backward |
| Ctrl+F | One character forward |
| Ctrl+← | One word backward |
| Ctrl+→ | One word forward |
| Ctrl+A | To start of line |
| Ctrl+E | To end of line |
| Ctrl+P | One line up |
| Ctrl+N | One line down |
| Ctrl+↑ | To previous block |
| Ctrl+↓ | To next block |
| Ctrl+Y | One page up |
| Ctrl+V | One page down |
| Alt+\ | To top of buffer |
| Alt+/ | To end of buffer |

Special movement

| | |
|-------|-----------------------------|
| Alt+G | Go to specified line |
| Alt+] | Go to complementary bracket |
| Alt+↑ | Scroll viewport up |
| Alt+↓ | Scroll viewport down |
| Alt+< | Switch to preceding buffer |
| Alt+> | Switch to succeeding buffer |

Information

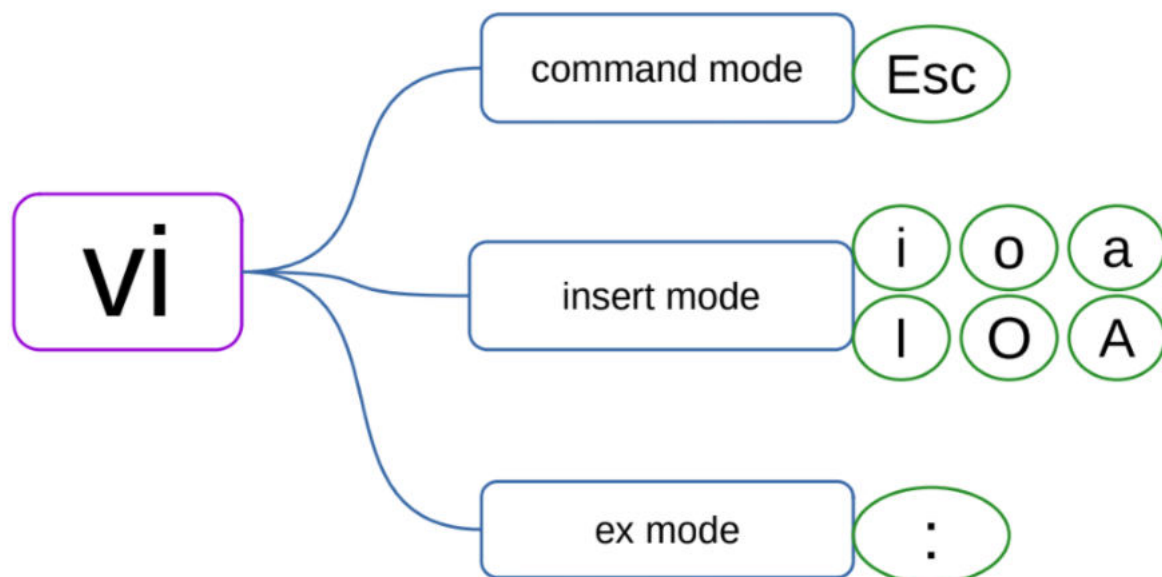
| | |
|--------|----------------------------------|
| Ctrl+C | Report cursor position |
| Alt+D | Report line/word/character count |
| Ctrl+G | Display help text |

Various

| | |
|-----------|--------------------------------|
| Alt+A | Turn the mark on/off |
| Tab | Indent marked region |
| Shift+Tab | Unindent marked region |
| Alt+V | Enter next keystroke verbatim |
| Alt+N | Turn line numbers on/off |
| Alt+P | Turn visible whitespace on/off |
| Alt+X | Hide or unhide the help lines |
| Ctrl+L | Refresh the screen |

Source: <https://www.nano-editor.org/dist/latest/cheatsheet.html>

Modes in **vi** editor



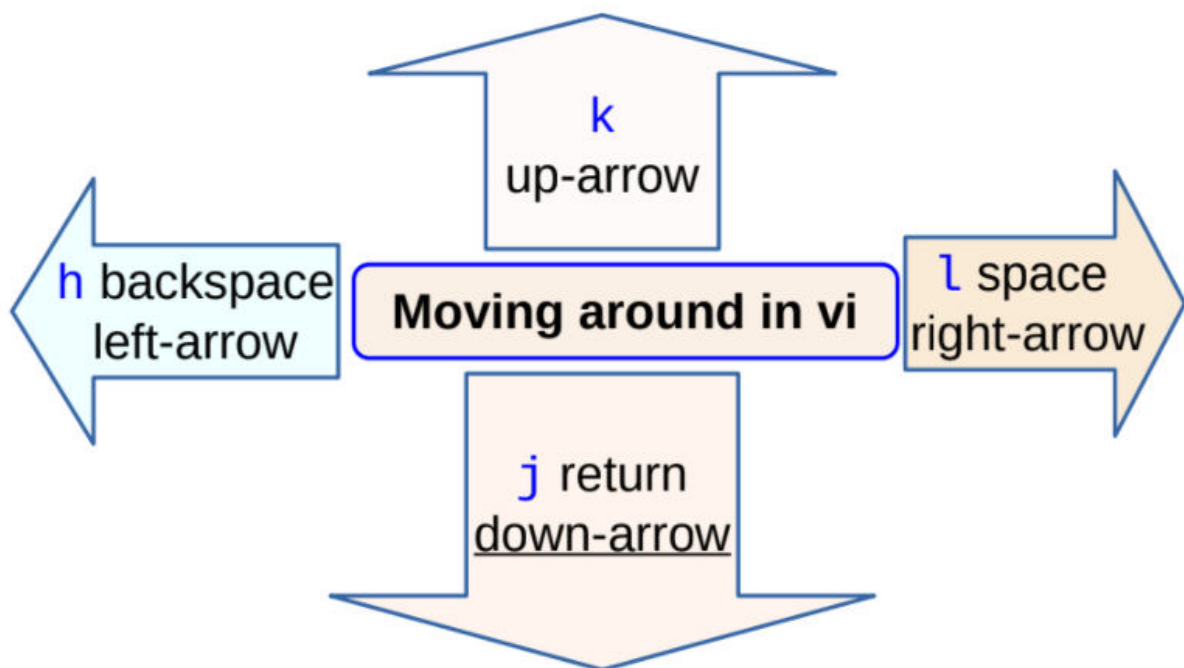
i will insert the characters from the current position of the cursor

o will insert a new line

`a` will append the text

`vi` help

- **These work only in the Command mode**
 - Press `Esc` to enter this mode
- To exit out of `vi`
 - `:w` → write out
 - `:x` → write out and quit
 - `:wq` → write out and quit
 - `:q` → quit (if write out is over)
 - `:q!` → ignore changes and quit
 - *If nothing else works, cry and smash your keyboard, and then rage quit*



`vi` command mode

- Screen manipulation
 - `Ctrl + F` → Scroll forward one screen
 - `Ctrl + B` → Scroll backward one screen

- `Ctrl + D` → Scroll down half screen
- `Ctrl + U` → Scroll up half screen
- `Ctrl + L` → Redraw screen
- `Ctrl + R` → Redraw screen removing deleted stuff
- Moving around
 - `0` → Start of the current line
 - `$` → End of the current line
 - `w` → Beginning of the next word
 - `b` → Beginning of the preceding word
 - `:0` → First line in the file
 - `1G` → First line in the file
 - `:n` → n-th line in the file
 - `nG` → n-th line in the file
 - `:$` → Last line in the file
 - `G` → Last line in the file
- Changing text
 - `r` → Replace a single character under the cursor
 - `R` → Replace characters from the cursor till `Esc`
 - `cw` → Change word under the cursor, from the current character till `Esc`
 - `cNw` → Change **N** words, from the current character till `Esc`
 - `C` (CAPITAL C) → Change characters in the current line till `Esc`
 - `cc` (small c) → Change the line till `Esc`
 - `Ncc` → Change the next **N** lines, starting from the current till `Esc`
- Deleting text
 - `x` → Delete a single character under the cursor
 - `Nx` → Delete **N** characters from the cursor
 - `dw` → Delete one word, from the character under the cursor

- `dNw` → Delete **N** words, from the character under the cursor
- `D` → Delete the rest of the line, from the character under the cursor
- `dd` → Delete the current line
- `Ndd` → Delete the next N lines, starting from the current one
- Copy and Paste text
 - `yy` (small y) → Copy the current line to the buffer
 - `Nyy` → Copy the next N lines, including the current, into the buffer
 - `p` → Paste buffer into the file after the current line
 - `u` → Undo the previous action
- Searching text
 - `/string` → Search forward for the given `string`
 - `?string` → Search backward for the given `string`
 - `n` → Move the cursor to the next occurrence of the `string`
 - `N` → Move the cursor to the previous occurrence of the `string`

`:se nu` → Set line numbers

`:se nonu` → Unset the line numbers

To copy a file using secure copy `scp`

Syntax → `scp <username>@<IP_ADDRESS>:<path/to/file/on/that/machine> <where/to/save>`

Example → `scp gphani@10.17.0.167:Documents/code3d.tar .`

To untar (extract) a `.tar` file

Syntax → `tar -xvf filename.tar`

`emacs`

C-x means `Ctrl + x`

M-x means `Alt + x`

- Moving around
 - `C-p` → Move up by one line
 - `C-b` → Move left by one character
 - `C-f` → Move right by one character

- `C-n` → Move down by one line
- `C-a` → Go to the beginning of the current line
- `C-e` → Go to the end of the current line
- `C-v` → Move forward one screen
- `M-<` → Move to the first line of the file
- `M-b` → Move left to the previous word
- `M-f` → Move right to the next word
- `M->` → Move to the last line of the file
- `M-a` → Move to the beginning of the current sentence
- `M-e` → Move to the end of the current sentence
- `M-v` → Move back one screen

Source: <https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf>

`emacs` commands

- Exiting `emacs`
 - `C-x C-s` → Save buffer to the file
 - `C-z` → Exit emacs but keep it running
 - `C-x C-c` → Exit emacs and stop it
- Searching a text
 - `C-s` → Search forward
 - `C-r` → Search backward
 - `M-x` → Replace string
- Copy and Paste
 - `M-backspace` → Cut the word before the cursor
 - `M-d` → Cut the word after the cursor
 - `C-k` → Cut from the cursor to the end of the line
 - `M-k` → Cut from the cursor to the end of the sentence
 - `C-y` → Paste the content at the cursor

Syntax for emacs → `emacs -nw filename`

Make sure to pass the `-nw` flag to make it open in terminal mode instead of the GUI