

VI editor

Vi editor works in two different modes

1) Insert mode

2) Command mode

1) Insert mode keys

Press I to insert in Insert mode

I = Insert text before the current cursor position

I = Insert text at the beginning of the current line

a = Append text after current cursor position

A = Append text after the end of the current line

O = Create new line below the current line

O = Create new line above the current line

j = To move down

k = To move up

h = To move left

I = To move right

G = Will direct you at the last line of the file

3. i = Move paragraph to paragraph.

2 = Move beginning of paragraph

press **esc** To exit from Insert mode

Command Mode

yy = Copy (yank) the current line

3yy = Copy three lines (starting with current line)

yw = Copy one word starting at the cursor position

2yw = Copy two words starting at the cursor position

p = paste after the current cursor position

P = paste before the current cursor position

dd = delete on the current line

4dd = delete 4 lines (Starting with the current line)

dw = delete one word line

x = delete the character at the cursor position

u = undo the last change

U = undo all changes to the line

/word = Search for the word in the file

:%os/zed/blue = Replace the word zed with blue

: Se nu = Show line numbers

: Se neru = Remove line numbers

: 8 = Go to line number 8

. b = move one back word

Saving and Exiting

: W = Save the file

: Wq = Save the file and quit vi

: ZZ = Save the file and quite vi

: xe = Save the file and quite vi

: q = quit vi without saving

: q! = Forcefully quite without saving

WC = To count the number of lines, words, and characters in a file.

WC - l = To display the number of lines

WC - w = To display the number of words

WC - c = To display the number of character

examples :-

1) cat /etc/passwd | wc -l
To view file location of file Pipe To display number of lines.

2) cat /etc/passwd | wc -w
To display number of words

3) cat /etc/passwd | wc -c
To display number of character

Head or Tails

Head = Display the ^{first} few lines of a file
(it shows first 10 lines of a file)

examples :-

1) head -n5 etc | lessd

(n = To specify the number of lines you wish to view)

2) head -n15 etc | lessd > test

(> = it will direct data)

3) head etc | lessd | wc - l = ^{Display} no. of line

4) head etc | lessd | wc - w = Display no. of words

5) head etc | lessd | wc - c = Display characters

Tails = Display last few lines of a file

(it shows last 10 lines of a file)

examples :-

1) tail etc | lessd | wc - l = ^{Display last} no. of lines.

2) tail -n5

grep (Global Search for regular Expression & print out)

Grep: grep can use to print the lines of Text that match a specific pattern (Searching and matching text files contained within the regular expression)

Syntax / example

- 1) `grep 'root' /etc/passwd` (it will print all the lines which contains root)
- 2) `grep -o 'bin' /etc/passwd` (" -o " print only the matched parts of matching line)
- 3) `grep 'bin' /etc/passwd | wc -l` (it search bin & show number of lines.)
- 4) `grep -i 'System' /etc/passwd` (" -i " Ignores, case for matching)
- 5) `grep -io 'System' /etc/passwd` (" -o " it display ignores case sensitive and matched parts of matching line)

" ^ = caret "

- c) grep 'Na' | tac | passwd
(" ^ To matched the lines which start with the given string or pattern)
- d) grep -i 'Nap' | tac | passwd
("-i 'Nap'" it display ignores case sensitive & match the lines which started with given word | string)
- e) grep "ich \$" | tac | passwd
("ich \$" it specifies the end of a line , to match the lines which ends with given word)
- f) grep -B 2 'seviers' | tac | passwd
("-B" it print the search line and n lines before the result)
- g) grep -A 2 'seviers' | tac | passwd
("-A" it prints the searched lines and n lines after the result)
- h) grep -c = it prints the searched line and n lines after and before the result)
grep -v > . it display lines without root (it is not display root). Skip root.

Camlin

* **"More"** :- Display the content of a file one page at a time.
(it also shows how much data was read)

"Less" :- It allows you to view text files one page at a time
excerpts :-

1) more [file] [option]

2) less [file] [option]

options :-

-d : displays the file in decimal form

-e : displays the file in octal form

-o : displays the file in hex form

-r : displays the file in reverse order

-s : displays the file in sorted order

-t : displays the file in tabular form

Cut Command

basically "first command or tool" or "cat" or "tail"

Editing Filters

Horizontal

Vertical

Horizontal

- In horizontal filter, it edits a row in the file.
- examples of horizontal filter

- head
- tail
- grep

Vertical

- In vertical filter, it edits a column in the file.

Examples :

- cut

If you want to view only a part of a file (or a section), then "cut" command can be useful.

example:-

1) `Cut -d ';' -f test`

Annotations:

- delimeter
- view only
- field
- filename
- command
- Separation symbol
- as Special character in the file.

* -d = delimiter, separation of columns

2) `Cut -d ';' -f 1,3 test`

To view the

first and third word of each line

3) `Cut -d ';' -f 2 -5 , 7 test`

To view the second, ^{seventh}, fourth & fifth word of each line

(2-5)
with the help of
it consider all numbers
2 4 5

Sort = To sort text files.

examples:-

delimiter

filename

pipe (to connect commands)

1) `cut -d ';' -f 1-5 test.txt | sort -n`

Command

To view 1 to 5th

To sort the file

word of each

by numbers (-n)

line

cmd / delimiter cmd / filename cmd / delimiter (sorted by numbers)

2) `cut -d ';' -f 1-5 test | sort -t ';' -k 3 -n`

To view only 1 to 5th

word of each line

(-k 3 it is a column number)

3) `cut -d ';' -f 1-5 test | sort -r -e`

[-r sort in reverse order]

* `| -c` = it cut the characters

* `free -m | cut -c 1-20 | grep mem -i`

↑
it shows memory

↑
cut 1 to 20 characters

↑
find memory word

↑
ignore case option