Introduction to vi Editor

TEKNOTURF

vi Editor

- The **vi** editor lets a user to create new files or edit existing files
- **Syntax:**
  - vi <filename>
  - If the file doesn't exist, a new file is created. If the file exists then the file is opened for editing
  - When vi command is given without an argument, the editor is opened for inserting data, which can be saved later
- **Modes of vi editor:**
  - Command mode
  - Input mode
  - Last line mode

Modes of vi

TEKNOTURF

- **command mode**
  - When given vi, the file is opened in the command mode by default
- Some of the available options are
  - r - replace one character
  - x - delete text at cursor
  - dd - delete entire line
  - 5dd - delete 5 lines
  - yy - copy a line
  - *n*yy - copy *n* lines
  - P - paste above current line
  - p - paste below current line

Input Mode

TEKNOTURF

Options to switch from command mode to input or text mode:

- i- insert text at cursor
- a- insert text after cursor
- A- append text at the end

Input mode

- This mode allows the user to insert or modify or append text.
- To change to command mode press **<Esc>**

Last Line Mode

TEKNOTURF

Last line mode

- This is invoked from the command mode
- When the user types : the cursor moves to the last line of the screen

options given in last line mode:

- :w - save
- :wq - save and quit
- :q! - quit without save
- :w <filename> - saves a copy of the file (save as in windows)
- :set nu - sets line number
- :set ai - set auto indent

Cursor Movement Commands

TEKNOTURF

j or <return> [or down-arrow]	Move cursor down one line
k[or up-arrow]	Move cursor up one line
h or<backspace> [or left-arrow]	Move cursor left one line
l or<space>[or right-arrow]	Move cursor right one line
0(zero)	Move cursor to start of current line(the one with the arrow)
$	Move cursor to end of the current line
w	Move cursor to beginning of next word
b	Move cursor back to beginning of preceding word
:0<return> or 1G	Move cursor to first line in file
:n<return>or nG	Move cursor to line n in file
:$<return>or G	Move cursor to last line in file

Paging Functions

TEKNOTURF

:.=	Returns line number of current line at bottom of screen
:=	Returns the total number of lines at bottom of the screen
^g	Provides the current line number, along with the total number of lines,in the file at the bottom of the screen

Searching Commands

/string	Search forward for occurence of string in text
?string	Search backward for occurence of string in text
n	Move to next occurence of search string
N	Move to next occurence of search string in opposite direction

Introduction to SED

- sed command is commonly used to replace string in Unix or UNIX based OS. That is why it is more commonly used as 'sed replace'
- sed is stream editor in Unix. It is used to parse and transforms texts
- sed was developed in the year 1973
- sed uses compact programming language; it was built for command line processing
- sed is also used to make programs which can change files

SED Commands

SED with &
- The option & is used to append to the search pattern

Syntax:
- sed 's/search_pattern/& append_pattern/' file.txt

Example:
- The below sed command will find the search pattern '123' and append '456' to it
- $ sed 's/123/&456/' file.txt

SED Commands

TEKNOTURF

SED with s

- S stands for substitution. It replaces the searched string with the new string

Syntax:

- sed 's/search_pattern/replaced_pattern/' file.txt

Example:

- The below command will replace the string 'ABC' with 'ZYX' In the given sed command, sed will replace only the first occurrence of the pattern in each line
- $ sed 's/ABC/ZYX/' file.txt

SED Commands

TEKNOTURF

SED with g

g works as global replacement when used with sed command. The following command will replace all the searched pattern with the replaced pattern

Syntax:

- $ sed 's/searched_pattern/replaced_pattern/'g file.txt