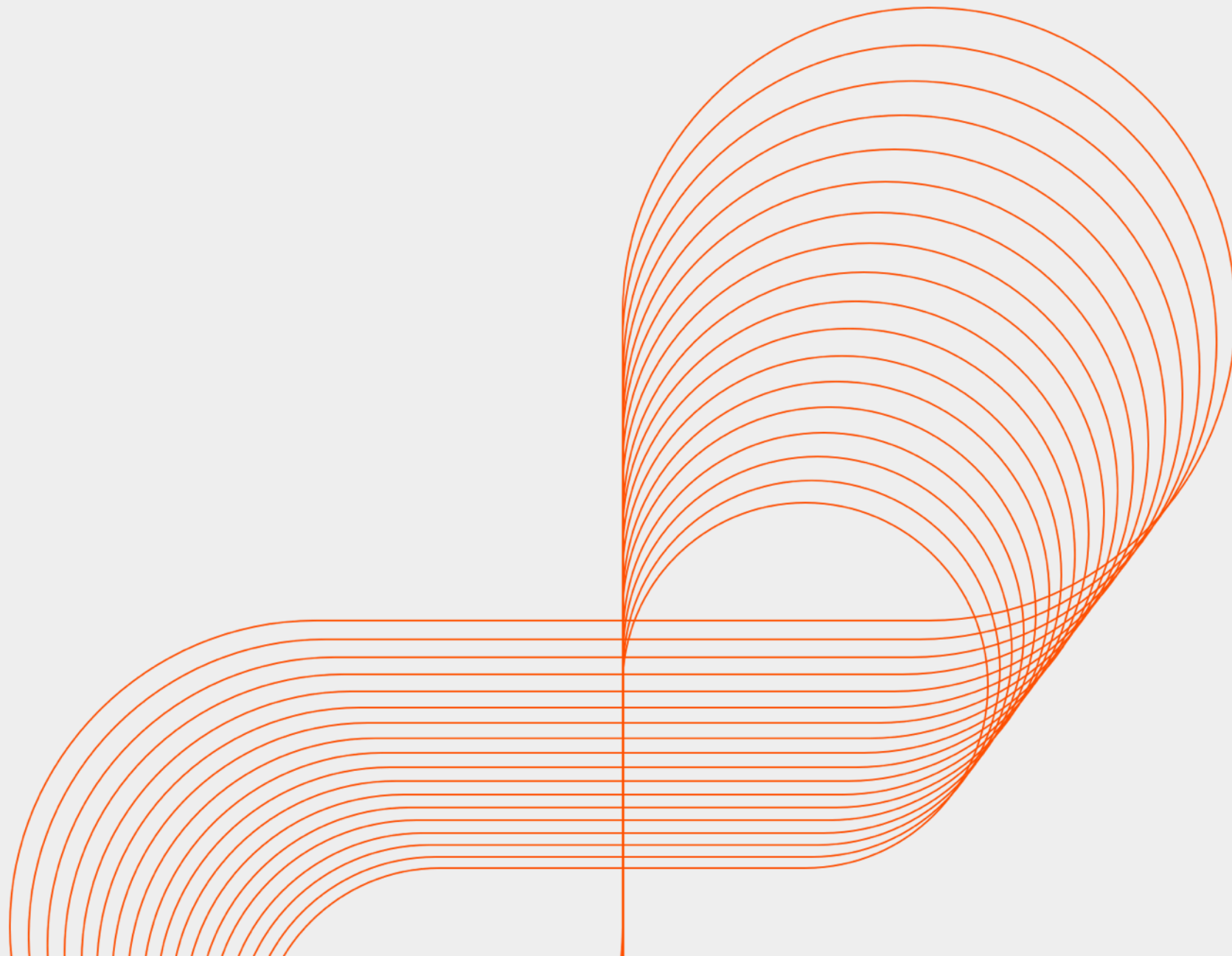




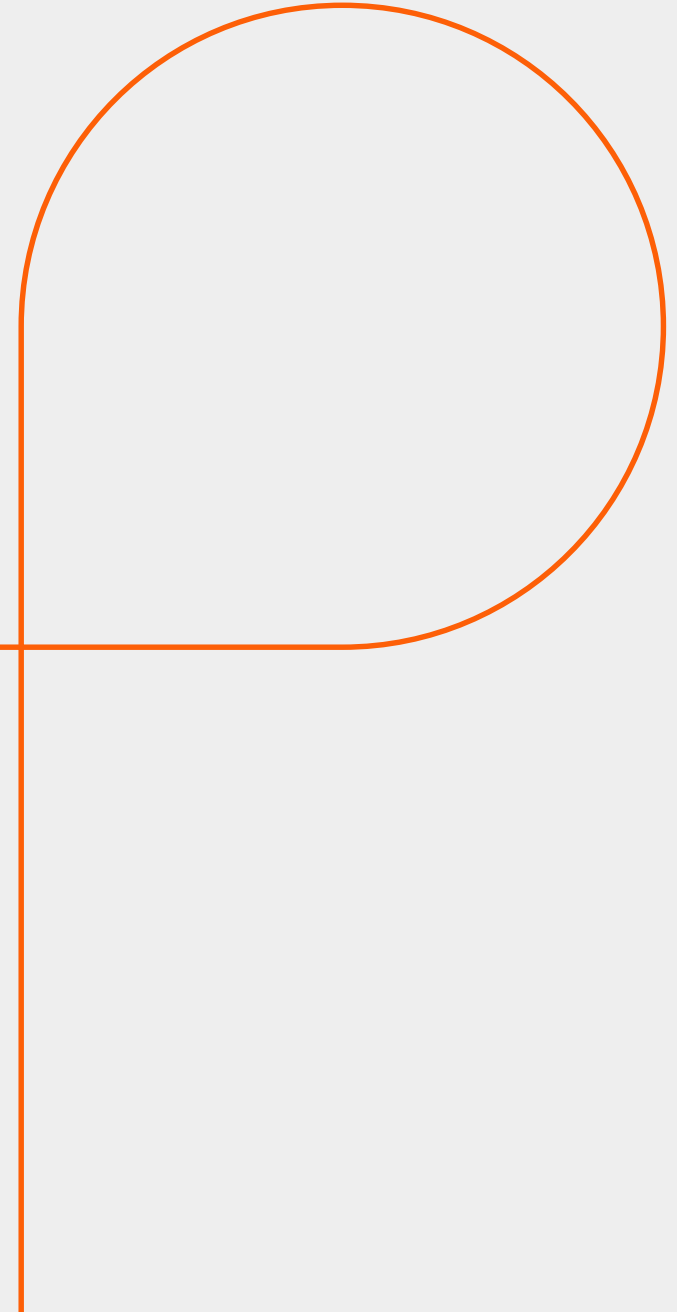
Persistent

UNIX



Editing Files

vi Editor



Vi Editor

- Vi is considered as the standard editor in Unix editors because it is available on all Unix systems
- There are lots of implementations of Vi but the original Vi was written by Bill Joy in 1976 for a BSD Unix release
- Vi implementations are very similar across major Unix variants
- Vi requires fewer resources than other editors

Vi Editor Concepts

- Vi editor operates in three modes - command mode, command line(or last line) mode and insert mode
 - Command mode - everything that is typed is interpreted as a command by the editor
 - Insert mode - By typing the letter *i*, you go into the insert mode and can start typing the text. Pressing *ESC* key will take you back to the command mode
 - Command Line mode - Lets you enter commands that appear on the bottom line of the screen, and can affect the editing environment or interact with the external environment

Vi Command Mode

- The command mode commands are normally in the following format
[*count*] command [*where*]
- Arguments in the brackets are optional
- Most commands are one character long, including those which use control characters
- The count is entered as a number beginning with any character from 1 to 9. e.g. 23x while in command mode, will delete 23 characters
- Some commands use an optional *where* parameter, where you can specify how many lines or how much of the document the command affects

Vi Command Line Mode

- This mode is used to execute commands that are used to change environment or interact with the external environment
- The commands always start with a :
- Examples are :w, :q, :set nu, etc
- These commands can also change certain options of the editor

Vi Commands

- We shall view the commands for the following operations:
 - Starting Vi
 - Stopping Vi
 - Insertion
 - Navigation
 - Deletion
 - Searching
 - Undo
 - Copy & paste

Starting Vi

| Command | Meaning |
|------------------------------------|--|
| <i>vi filename</i> | Edits filename |
| <i>vi -r filename</i> | Edits last save version of filename after a crash |
| <i>vi + n filename</i> | Edits filename and places cursor at line n |
| <i>vi + filename</i> | Edits filename and places cursor on last line |
| <i>vi +/string filename</i> | Edits filename and places cursor on first occurrence of string |
| <i>vi -R filename</i> | Opens the file in readonly mode, although the changes can be written using :w! command |

Ending Vi

| Command | Meaning |
|-------------------------------|---|
| <i>:w</i> | Save changes |
| <i>:w new_filename</i> | Write buffer to new filename |
| <i>:wq</i> | Save changes and quit vi |
| <i>:x</i> | Same as :wq |
| <i>:q</i> | Quits VI and may prompt if you need to save |
| <i>:q!</i> | Quits VI and without saving |
| <i>:e!</i> | Edits file discarding any unsaved changes (starts over) |

Insertion Commands

| Command | Meaning |
|-----------------|--|
| <i>a</i> | Append text after the cursor |
| <i>A</i> | Append text at the end of the line |
| <i>i</i> | Insert text before the cursor |
| <i>I</i> | Insert text at the beginning of the line |
| <i>o</i> | Open a new line below the cursor |
| <i>O</i> | Open a new line above the cursor |

Positioning Commands

| Command | Meaning |
|--|---|
| <i>h, Left Arrow, or Back Space</i> | Move left one character |
| <i>j or Down Arrow</i> | Move down one line |
| <i>k or Up Arrow</i> | Move up one line |
| <i>l, Right Arrow or Spacebar</i> | Move left (forward) one character |
| <i>O</i> | Move to beginning of current line |
| <i>\$</i> | Move to end of current line |
| <i>w</i> | Move forward one word (including punctuation) |
| <i>W</i> | Move forward one word (past punctuation) |
| <i>b</i> | Move back one word (including punctuation) |
| <i>B</i> | Move back one word (past punctuation) |
| <i>e</i> | Move to end of current word |
| <i>E</i> | Move to end of word (past punctuation) |

Positioning Commands

| Command | Meaning |
|----------------------|-----------------------------|
| <i>gg</i> | Go to first line of file |
| <i>G</i> | Go to last line of file |
| <i>#G</i> | Go to # line of file |
| <i>:#</i> | Go to line # |
| <i>CTRL-f</i> | Page forward one screen |
| <i>CTRL-b</i> | Page back one screen |
| <i>CTRL-d</i> | Scroll down one-half screen |
| <i>CTRL-u</i> | Scroll up one-half screen |
| <i>CTRL-E</i> | Scroll window down one line |
| <i>CTRL-Y</i> | Scroll window up one line |

Deleting Text

| <i>Command</i> | Meaning |
|------------------------|--|
| <i>x or dl</i> | Delete character at the cursor |
| <i>X or dh</i> | Delete character to the left of the cursor |
| <i>dw</i> | Delete word (or part of word to left of cursor) |
| <i>#dw</i> | Delete # of words |
| <i>dd</i> | Delete line containing the cursor |
| <i>#dd</i> | Delete # of lines |
| <i>D or d\$</i> | Delete line to right of cursor (from cursor position to the end of the line) |
| <i>dG</i> | Delete to the end of the file |

UndoingText

| Command | Meaning |
|-------------------------------|---|
| <i>u</i> | Undo last few actions |
| <i>U</i> | Undo the previous actions in the current line |
| <i>Ctrl+R or :redo</i> | redo |

Changing Text

| Command | Meaning |
|------------------------------------|---|
| <i>cw</i> | Change word (or part of word) at cursor location to the end of the word |
| <i>#cw</i> | Change # number of words |
| <i>c#b</i> | Change backwards # of words |
| <i>cc or S</i> | Replace entire current line |
| <i>C or c\$</i> | Change from cursor to end of line |
| <i>c0</i> | Change from cursor to beginning of line |
| <i>R</i> | Overwrite or replace characters on line |
| <i>r</i> | Replace character at cursor with one other character |
| <i><< or >></i> | Shifts the line left or right (respectively) by one shift width (a tab) |

Cut, Copy & Paste Commands

| Command | Meaning |
|--------------------------------------|--|
| <i>yy</i> | Yank the current line |
| <i>:y</i> | Yank the current line |
| <i>nyy or nY</i> | Places n lines in the buffer-copies |
| <i>"(a-z)nyy or "(a-z)ndd</i> | Copies or cuts (deletes) n lines into a named buffer a through z; |
| <i>p</i> | Put after the position or after the line |
| <i>P</i> | Put before the position or before the line |
| <i>"(a-z)p or "(a-z)P</i> | Pastes text from a named buffer a through z after or before the current line |

Shell Functions

| Command | Meaning |
|-----------------------|---|
| <i>:! cmd</i> | Executes shell command cmd; you can add these special characters to indicate:% name of current file# name of last file edited |
| <i>:!!</i> | Executes last shell command |
| <i>:r! cmd</i> | Reads and inserts output from cmd |
| <i>:w !cmd</i> | Sends currently edited file to cmd as standard input and execute cmd |
| <i>:cd dir</i> | Changes current working directory to dir |
| <i>:sh</i> | Starts a sub-shell (CTRL-d returns to editor) |

Search Commands

| Command | Meaning |
|-------------------------|--|
| /string | Search forward for string |
| ?string | Search backward for string |
| n | Search for next instance of string |
| N | Search for previous instance of string |
| :set ic | Ignores case when searching |
| :set noic | Pays attention to case when searching |
| :%s/old/new/g | Substitutes old with new throughout the file |
| :s/old/new/flags | Replace old with new in the current line according to flags. |
| | G Flag - Replace all occurrences of pattern |
| | C Flag - Confirm replaces. |
| & | Repeat last :s command |

Vi options

- The set command is used to turn on & off vi options such as indenting, line numbers, etc
- There are two types of options that can be changed
 - toggle options, which are either on or off
 - options that take a numeric or string value
- Toggle options may be on or off by default. To set an option, the command is: *set option*
- To toggle off an option: *set nooption*
- Examples: *set ic*, *set cindent*, *set nu*

Common Vi Options

| Command | Meaning |
|-----------------------------|-------------------------------------|
| <i>:set ai</i> | Turns on auto indentation |
| <i>:set all</i> | Prints all options to the screen |
| <i>:set eb</i> | Precedes error messages with a bell |
| <i>:set nu</i> | Shows line numbers |
| <i>:set ro</i> | Changes file type to "read only" |
| <i>:set nooption</i> | Turns off option |

Vi Configuration Files

- The options that are set in Vi are applied to that session only
- If you want the changes to be permanent, the options can be set in a file called *.vimrc*, that is read by Vi whenever it is started
- A global configuration file exists at */etc/vim/vimrc* that can be modified only by the root
- You can also setup a configuration file in your home directory as */home/username/.vimrc*
- The global *vimrc* file is executed first, then the one in your home directory

Links for objective multiple choice questions.

- <http://www.sanfoundry.com/linux-command-mcq-1/>
- <http://www.sanfoundry.com/linux-command-mcq-2/>
- <http://www.sanfoundry.com/linux-command-mcq-3/>
- <http://www.indiabix.com/computer-science/unix/>
- <http://www.avatto.com/computer-science/test/mcqs/questions-answers/unix/153/1.html>
- <http://www.gkseries.com/computer-engineering/unix/multiple-choice-questions-and-answers-on-unix-and-shell-programming>
- http://www.withoutbook.com/online_test.php?quiz=38&quesNo=10&subject=Top%2010%20UNIX%20Online%20Practice%20Test%20%7C%20Multiple%20Choice