vi editor

Pronounced: `vee eye''

Agenda

- Describe the background of vi Editor
- Use vi editor to:
  - create text files
  - edit text files

What is vi ?

- The *visual editor* on the Unix.
- Before vi the primary editor used on Unix was the line editor
  - User was able to see/edit only one line of the text at a time
- The vi editor is not a text formatter (like MS Word, Word Perfect, etc.)
  - you cannot set margins
  - center headings
  - Etc…

Vi History

- Although other stories exist, the true one tells that vi was originally written by **Bill Joy** in 1976.
- Who is Bill Joy you ask**?**
  - He co-founded Sun Microsystems in 1982 and served as chief scientist until 2003.
- Joy's prowess as a computer programmer is legendary, with an oft-told anecdote that he wrote the vi editor in a weekend. Joy denies this assertion.

Some Vi Ports

- All Unix OS's
- MS-Dos
- Windows 3.x
- Windows 9x/2k/NT/XP
- OS/2

- Macintosh
- Atari
- Amiga
- OpenVMS/Alpha
- OpenVMS/VAX

Characteristics of vi

- The vi editor is:
  - a very powerful
  - but at the same time it is cryptic
  - It is hard to learn, specially for windows users
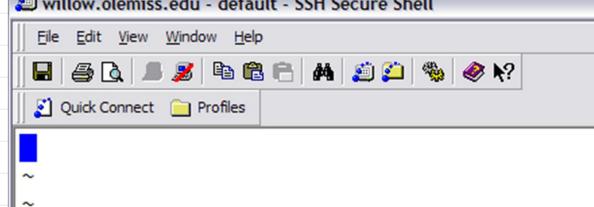- The best way to learn vi commands is to use them
- So Practice…

Vim equals Vi

◆ The current iteration of **vi** for Linux is called **vim**

  ▪ **Vi** I**m**proved
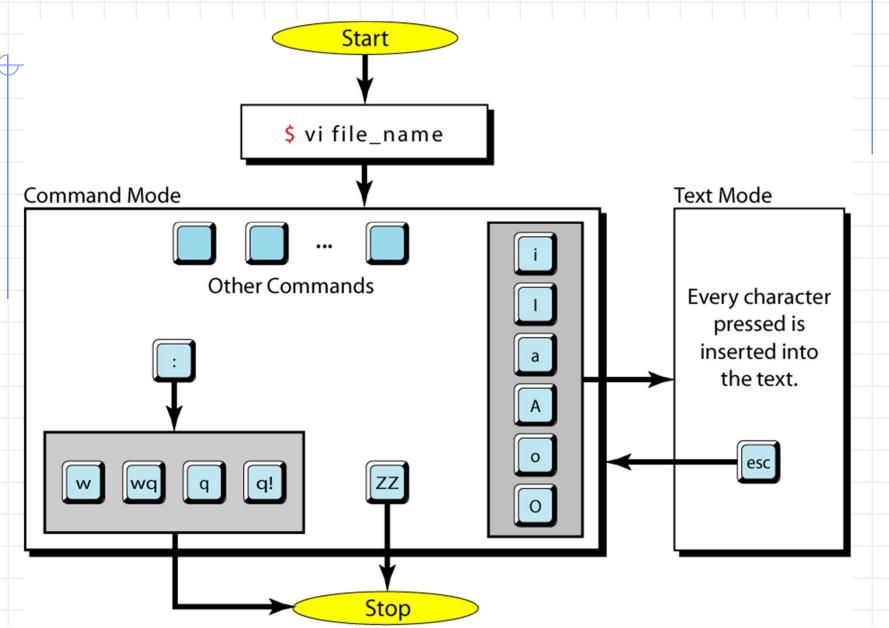
  ▪ http://www.vim.org

Starting vi

◈ Type **vi** <filename> at the shell prompt

◈ After pressing enter the command prompt disappears and you see tilde(**~**) characters on all the lines

◈ These tilde characters indicate that the line is blank

willow.olemiss.edu - default - SSH Secure Shell

File   Edit   View   Window   Help

Quick Connect   Profiles

~

~

Vi modes

- There are two modes in vi
  - Command mode
  - Input mode
- When you start vi by default it is in command mode
- You enter the input mode through various commands
- You exit the input mode by pressing the Esc key to get back to the command mode

vi Modes

How to exit from vi

◆ First go to command mode
  ■ press **Esc** There is no harm in pressing **Esc** even if you are in command mode. Your terminal will just beep and/or or flash if you press **Esc** in command mode

◆ There are different ways to exit when you are in the command mode

How to exit from vi
(comand mode)

◆ **:q** <enter> is to exit, if you have not made any changes to the file

◆ **:q!** <enter> is the forced quit, it will discard the changes and quit

◆ **:wq** <enter> is for save and Exit

◆ **:x** <enter> is same as above command

◆ **ZZ** is for save and Exit (Note this command is uppercase)

◆ The **!** Character forces over writes, etc. **:wq!**

◆ **:w** Write/Save changes, but don't quit

How to exit from vi
(comand mode)

◆ **:q** <enter> is to exit, if you have not made any changes to the file

◆ **:q!** <enter> is the forced quit, it will discard the changes and quit

```
~
~
~
:q!
bash-3.00$ 
```
Connected to willow.olemiss.edu

Saving Changes in vi
(comand mode)
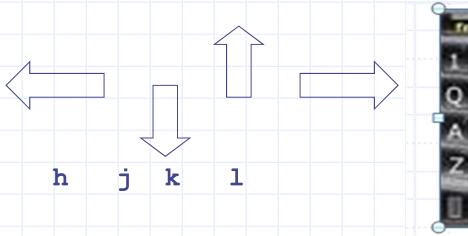
◆ **:wq** Write/save changes and quite

◆ **:w** Write/Save changes, but don't quit

```
~

~

"mysong" 1 line, 13 characters
```
Connected to willow.olemiss.edu   SSH2 - aes128-cbc - hmac-

Moving Around

◈ You can move around only when you are in the command mode

◈ Arrow keys usually works(but may not)

◈ The standard keys for moving cursor are:

h j k l

Moving Around

- **w** - to move one word forward
- **b** - to move one word backward
- **e** - to move end of the word
- **$** - takes you to the end of line
- <**enter**> takes the cursor the the beginning of next line
- **^,0** - start of line

Moving Around

- **-** - (minus) moves the cursor to the first character in the current line
- **H** - takes the cursor to the beginning of the <u>current screen</u>(Home position)
- **L** - moves to the Lower last line
- **M** - moves to the middle line on the current screen

Moving Around

- **f** - (find) is used to move cursor to a particular character on the current line
  - For example, **fa** moves the cursor from the current position to next occurrence of '**a**'
- **F** - finds in the reverse direction

Moving Around

- ◆ **)** - moves cursor to the next sentence
- ◆ **}** - move the cursor to the beginning of next paragraph
- ◆ **(** - moves the cursor backward to the beginning of the current sentence
- ◆ **{** - moves the cursor backward to the beginning of the current paragraph
- ◆ **%** - moves the cursor to the matching parentheses

Moving Around

- **Control-d** scrolls the screen down (half screen)
- **Control-u** scrolls the screen up (half screen)
- **Control-f** scrolls the screen forward (full screen)
- **Control-b** scrolls the screen backward (full screen).

Miscellanous

◆ **:!<command><Enter>** execute *<command>*

Entering text

◆ To enter the text in vi you should first switch to input mode

- To switch to input mode there are several different commands

- **a** - Append mode places the insertion point after the current character

- **i** - Insert mode places the insertion point before the current character

Entering text

- **i** inserts text before the current character

- **I** inserts text at the beginning of the current

Before

```
Insert before the cursor.
```

Before

```
beginning of line.
```

Command `i text <esc>`

After

```
Insert text before the cursor.
```

Cursor

(a) Insert Before Cursor

Command `I Insert at <esc>`
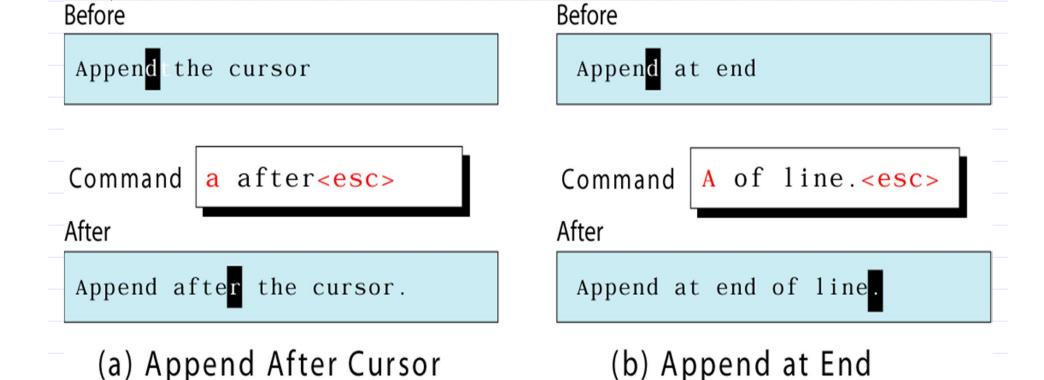
After

```
Insert at beginning of line.
```

Cursor

(b) Insert at Beginning

Insert Text After the Cursor and Line

- **a** appends text after the current character.

- **A** appends text at the end of the current line.

Before

```
Append the cursor
```

Before

```
Append at end
```

Command | `a after<esc>`

Command | `A of line.<esc>`

After

```
Append after the cursor.
```

After

```
Append at end of line.
```

(a) Append After Cursor

(b) Append at End

Entering text

- **I** - places the insertion point at the beginning of current line
- **o** - is for open mode and places the insertion point after the current line
- **O** - places the insertion point before the current line
- **R** - starts the replace(overwrite) mode

Editing text

- **x** - deletes the current character
- **d** - is the delete command but pressing only d will not delete anything you need to press a second key
  - **dw** - deletes to end of word
  - **dd** - deletes the current line
  - **d0** - deletes to beginning of line
- There are many more keys to be used with delete command

The change command

- **c** - this command deletes the text specified and changes the vi to input mode. Once finished typing you should press <**Esc**> to go back to command mode
- **cw** - Change to end of word
- **cc** - Change the current line
- There are many more options

Structure of vi command

- The vi commands can be used followed by a number such as

  n<command key(s)>

  - For example **dd** deletes a line **5dd** will delete five lines.

- This applies to almost all vi commands

- This how you can accidentally insert a number of characters into your document

Undo and repeat command

- **u** - undo the changes made by editing commands
- **.** (dot or period) repeats the last edit command

Copy, cut and paste in vi

- **yy** - (yank) copy current line to buffer
- **nyy** - Where **n** is number of lines
- **p** - Paste the yanked lines from buffer to the line below
- **P** - Paste the yanked lines from buffer to the line above

(the paste commands will also work after the **dd** or **ndd** command)

Stupid vi Tricks

◆ Indent four lines: 4>>

◆ Will delete the character under the cursor, and put it afterwards. In other words, it swaps the location of two characters: xp

◆ Similar to xp, but swapping lines: ddp

◆ Remove all lines that start with #

  ▪ :g/^#/d

◆ Remove all empty lines, assumes no spaces or tabs:

  ▪ :g/^$/d

Some Practice

◆ Using a command line utility called wget let's pull down a copy of the Gettysburg Address

wget –U " " http://wildbill.org/rose/gettysburg.txt

◆ Now we will run a few commands against the text (note: I placed several blank lines and lines starting with the #)

Gettysburg.txt

- Run the following commands:
  - **vi gettysburg.txt**
  - **:g/^#/d**
  - **:g/^$/d**

Gettysburg.txt

◈ Invoke vi's edit mode by pressing the **Esc** key, then a colon (**:**), and enter:
1,$s/oldstring/newstring/g

◈ This will change oldstring into newstring wherever it occurs throughout the entire text. The 1 (the number one) in the above command means "start the search on the first line". The $ means "end the search on the last line". The g at the end executes the change globally on each line. If you omit the g , the search will stop after finding the first occurrence of oldstring.

Vi References

◈ **The Vi Lovers Home Page**
   http://thomer.com/vi/vi.html

◈ **The Editor War**
   http://en.wikipedia.org/wiki/Editor_war

For the vi Lover

World War VI

HotGames Entertainment apps

Side-scrolling shooter

Word War Vi

Version 0.09 **Web** http://wordwarvi.sourceforge.net

Old-school arcade games are good, but put a geek twist on it and they become irresistible. *Word War Vi* is a side-scrolling shoot 'em up that puts an urgent focus on the old *Vi* vs *Emacs* debate.

You must pilot your craft through core memory to rescue lost SWP files. On your mission of mercy you must avoid the OS defences and wipe out "memory hogging *Emacs* processes." It's clear from this that the author has a strong position in the editor wars. Fortunately he also has a sense of humour, as do most of the *Emacs*-lovers at LXF Towers – it must have caught us on a good day.

You start the game by hitting Q (for quarter, apparently – a 25c coin to start an arcade game). In fact the keyboard controls are all odd – it uses *Vi*'s traditional HJKL direction keys (you can use arrows instead), Z or space bar to fire signal number -9 (laser), B to inject NULL pointer into execution stream (drop a bomb), and C to forks to evade pursuers (throw chaff).

To install, untar the source code package (**tar xvf packagename.tar.gz**), change into the directory with **cd** then type **./configure** followed by **make install** as root – see page 76 for more on installing software if you're unsure.

This is very old-school arcade, with noisy sound effects of gunfire and explosion. The rock-based soundtrack, recorded by the author and his friends,

> **Flying above the guns can get you dangerously near Kernel Space.**

"It's clear that the author has a strong position in the editor wars."

adds to the fun. As you go up through the levels things become even more frenzied.

Development is rapid, with bugfixes appearing in CVS every week. If you mis-spent your youth in arcades, or on 8-bit PCs and consoles, you'll need no more prompting to try this. If you missed out on 80s games, you might want to dive in now – at least you'll have an answer if anyone asks: "What did you do in the Editor Wars, daddy?"

Cave exploration game

http://wordwarvi.sourceforge.net/

World War VI

◆ **Word War vi is your basic side-scrolling shoot 'em up '80s style arcade game.**

◆ **You pilot your "vi"-per craft through core memory, rescuing lost .swp files, avoiding OS defenses, and wiping out those memory hogging emacs processes.**

◆ **When all the lost .swp files are rescued, head for the socket which will take you to the next node in the cluster.**