Instructor: Nicolas Savva

February 6, 2015

---

1 based on slides by Hussam Abu-Libdeh, Bruno Abrahao and David Slater over the years
Remaining CSUGLab accounts activated
A2 is due Saturday 02/07
A3 coming out within the next 24hrs
Today

- Text editor: vim
- Multiplexing terminals: screen / tmux
Vim = Awesome!

- Vim is a powerful lightweight text editor.
- The name “Vim” is an acronym for “Vi IMproved”
  - vi is an older text editor
- Ports of Vim are available for all systems
  - including Microsoft Windows

Vim allows you to perform text editing tasks much faster than most other text editors!
- Though it does have a learning curve
One of the reasons that Vim allows you to perform tasks quickly is because it works in modes.

Without modes, users would have to either use command menus (with a mouse or keyboard), or use complex/long command shortcut keys involving the control key (ctrl) or the alt key (alt).

Vim uses modes to speed up editing by not relying on command keys or menus.

You can do all of your editing by just using the keyboard which is super fast!!
The 3 main modes of Vim

- **Normal mode:**
  - Launching pad to issue commands or go into other modes
  - Allows you to view the text but not edit it
  - Vim starts in normal mode
  - You can jump to normal mode by pressing the Escape (Esc) key on your keyboard

- **Visual mode:**
  - Used to highlight text and perform operations on selected text
  - You get to visual mode from normal mode by pressing the `v` key on your keyboard

- **Insert mode:**
  - Used to type text into the buffer (file)
  - This probably what you’re used to from your text editor
  - You get to the insert mode by pressing the `i` key on your keyboard
You can issue “command-line commands” from inside Vim to perform some functionalities
- write to disk, quite, get help, split screen, ...etc

To issue a command, go to normal mode and the type: followed by your command

Launching Vim help

:help
Moving around

**Fast**

You can use your mouse to move around in Vim (assuming a graphical interface as in gVim).

**Faster**

However it is much faster to just use your keyboard, and for that you can just use the arrow keys to move up/down/left/right.

**Fastest**

You can even be more efficient by not leaving the main area of the keyboard and using “h” to go left, “j” to go down, “k” to go up, and “l” to go right.

To start off, I recommend you just use the arrow keys.
To get started, launch Vim and go through the built in help.

`:help`

You can search for help on a specific topic as well.

`:help [topic to search for]`
Helpful commands

I can’t possibly teach you about all the power of Vim in a few minutes, however here are a few commands to help you get started.

**Getting help**

:help

**Entering normal mode**

<Esc>

**Entering insert mode (from normal)**

<i>

**Entering visual mode (From normal)**

<v>
Helpful commands

Save text to filename.txt
:w filename.txt

Exit
:q

Quit without saving
:q!

Open another file
:e [filename]
Helpful commands

Turn on syntax highlighting
:syntax on

Turn on line numbering
:set number

Turn on spell check
:set spell
Helpful commands

Split screen horizontally
:sp

Split screen vertically
:vsp

Move between split regions
<ctrl-w w>
The Most Helpful Command By Far

:help
Vim can run right in your shell, but there are also implementations of it that run in a nice GUI window (with menus, toolbars, and mouse)

- Use gVim for that
- MacVim for Mac OS X
Useful links

- **Vim project website**

- **Vim tips and tricks**

- **Vim recipes**
  - [http://vim.runpaint.org/toc/](http://vim.runpaint.org/toc/)
Bash isn’t Perfect

There are a few problems with your basic BASH session. Some of these you may even have encountered already:

- Your session isn’t preserved if you close your ssh connection
- It’s a pain to switch back and forth between files/the prompt.
- Sometimes using two or three shells at once would be really convenient!

All of these complaints can be resolved by using screen.
The `screen` command

`screen` - a screen manager with terminal emulation

Generally `screen` can be used just as you would normally use a terminal window. However, special commands can be used to allow you to save your session, create extra shells, or split the window into multiple independent panes.

Passing Commands to `screen`

Each `screen` command consists of a CTRL-a (hereafter referred to as C-a) followed by another character.
Using Screen

Attach a screen

```
screen [options]
```
- Opens a new screen for use
- `-a` : include all capabilities

Resume a screen

```
screen -r [pid.tty.host]
```
- Resumes a detached screen session

```
screen -x [pid.tty.host]
```
- Attach to a non-detached screen session

If you only have one screen, the `[pid.tty.host]` string is unnecessary.
Identifying Screen Sessions

**Screen Listing**

```
screen -ls  or screen -list
```

- Lists your screen sessions and their statuses

These screen sessions are the [pid.tty.host] strings required for resuming

**Resuming a screen**

If `screen -ls` returns `15829.pts-9.rumman (Detached)`

- `screen -r 15829.pts-9.rumman` to resume the screen

**Note:** You only need to specify the full “name” of the session if you have multiple sessions open. If you just have one session, just use `screen -r`
Creating More Shells

Creates a New Shell Window

C-a  c
- Creates a new shell in a new window and switches to it
- Useful for opening multiple shells in a single terminal
- Similar to tabbed browsing/tabbed IMs

But how do we switch between windows? (hint: every window is numbered by order of creation)

Window Selection

C-a  1  - switches to window 1
C-a  9  - switches to window 9
Split Screen Computing

C-a S - splits your terminal area into multiple panes
C-a tab - changes the input focus to the next pane

- The ‘S’ is case-sensitive
- Each split results in a blank pane
- Use C-a c to create a new shell in a pane
- Use C-a <num> to move an existing window to a pane

Note:
When you reattach a split screen, the split view will be gone. Just re-split the view, then switch between panes and reopen the other windows in each with C-a <num>
Now let's put this together to do something useful

Suppose you are doing some serious scientific computing and want to run it on a remote server. We can put together what we have learned to do this efficiently:

- `ssh` into the remote machine
  
  ```bash
  ssh slater@boom.cam.cornell.edu
  ```

- `start screen`
  
  ```bash
  screen
  ```

- `start mathematica`
  
  ```bash
  math < BatchJob.m
  ```

- `renice the math kernel so other uses can use the machine`
  
  ```bash
  renice -20 PID
  ```

- Detach the screen, logout, and come back 8 hours later when it is done
Terminal Multiplexer tmux

- Concurrently view/manage multiple programs in one terminal
- Read documentation (key-binding / configuration)
- Edit .tmux.conf
tmux - what it looks like
### tmux cheat sheet v1.0

http://www.clintoncurry.net/tmux-cheatsheet

**Sessions**

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>Rename the session</td>
</tr>
<tr>
<td>s</td>
<td>Go to a window in a different session.</td>
</tr>
<tr>
<td>t</td>
<td>Switch to the last session.</td>
</tr>
<tr>
<td>( or )</td>
<td>Cycle through sessions.</td>
</tr>
<tr>
<td>r</td>
<td>Refresh the client.</td>
</tr>
<tr>
<td>d</td>
<td>Detach yourself from the session.</td>
</tr>
<tr>
<td>D</td>
<td>Detach someone else from the session.</td>
</tr>
</tbody>
</table>

**Misc**

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-b</td>
<td>Send the prefix on through to the current pane's process.</td>
</tr>
<tr>
<td>C-z</td>
<td>Sends the tmux process to the background.</td>
</tr>
<tr>
<td>:</td>
<td>Enter a command directly.</td>
</tr>
<tr>
<td>?</td>
<td>Show a list of all key bindings.</td>
</tr>
<tr>
<td>~</td>
<td>Show tmux messages you might have missed.</td>
</tr>
<tr>
<td>t</td>
<td>Show a clock in the current pane.</td>
</tr>
<tr>
<td>i</td>
<td>Displays information about the current window.</td>
</tr>
</tbody>
</table>

**Copy/Paste (Emacs mode)**

See man page for vi mode commands.

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>[]</td>
<td>Enter copy mode</td>
</tr>
<tr>
<td>C-Spc</td>
<td>Start selection</td>
</tr>
<tr>
<td>C-w</td>
<td>Copy selection</td>
</tr>
<tr>
<td>v or R</td>
<td>Toggle rectangle select</td>
</tr>
<tr>
<td>#</td>
<td>Paste buffer select menu</td>
</tr>
<tr>
<td>]</td>
<td>Paste into the current buffer.</td>
</tr>
<tr>
<td>-</td>
<td>Delete the current buffer.</td>
</tr>
<tr>
<td>PgUp</td>
<td>Enter copy mode, and scroll up.</td>
</tr>
</tbody>
</table>

**Panes**

**Pane navigation and management**

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>Select the next pane</td>
</tr>
<tr>
<td>&quot;</td>
<td>Splits the area occupied by the current pane into two panes.</td>
</tr>
<tr>
<td>%</td>
<td>Splits the area occupied by the current pane into two panes.</td>
</tr>
</tbody>
</table>

**Arrows**

- Move to pane in that direction

**C-g**

- Rotates the panes around in the current window

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Creates a new window and moves the current pane there.</td>
</tr>
<tr>
<td>q</td>
<td>Briefly label the panes with their numbers and dimensions</td>
</tr>
<tr>
<td>x</td>
<td>Kill the current pane</td>
</tr>
<tr>
<td>{</td>
<td>Swap this pane with the previous one</td>
</tr>
<tr>
<td>}</td>
<td>Swap this pane with the next one</td>
</tr>
</tbody>
</table>

**Windows**

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Create a new window</td>
</tr>
<tr>
<td>0...9</td>
<td>Choose window 0, 1, ..., 9.</td>
</tr>
<tr>
<td>?</td>
<td>Select a window (prompt for index)</td>
</tr>
<tr>
<td>.</td>
<td>Give the current window a new index</td>
</tr>
<tr>
<td>f</td>
<td>Finds a window whose title matches the selected glob.</td>
</tr>
<tr>
<td>,</td>
<td>Rename the current window.</td>
</tr>
<tr>
<td>&amp;</td>
<td>Kill the current window.</td>
</tr>
<tr>
<td>M-n</td>
<td>next window that has seen activity</td>
</tr>
<tr>
<td>M-p</td>
<td>previous window that has seen activity</td>
</tr>
<tr>
<td>l</td>
<td>Go to the last window</td>
</tr>
<tr>
<td>n</td>
<td>Go to the next window</td>
</tr>
<tr>
<td>p</td>
<td>Go to the previous window</td>
</tr>
<tr>
<td>w</td>
<td>Choose a window from an interactive list.</td>
</tr>
</tbody>
</table>

**On the command line**

- `tmux new-session` -s `session-name`
- `tmux list-sessions`
- `tmux kill-session` -t `target`

**Tips**

- The default prefix is C-b. Some folks change it to C-a, but don’t you use that one?
- After C-b ?, use / to search for a command; n and N to navigate among results.
- Many folks can attach to the same session.
- Highly configurable. Check the man page.
- Some keys can be repeated without prefixing again.
tmux - check the website

http://tmux.sourceforge.net/
Next Time