

# Livin' the Vida Loca on the Command Line

*(Or How I Learned to Stop Worrying and Love the Shell)*

Sameer Khan  
[skhan156@gmail.com](mailto:skhan156@gmail.com)  
June 1, 2019

# Overview

- Simplify your life using textual tools

1. Shell (bash)
2. Editor (emacs)
3. Terminal (tmux)
4. Browser (elinks)
5. Markup languages (rst)
6. Character Encoding (ASCII)

# Shell

- Sourcing vs. Execution
  - Same shell vs. subshell ("shelling out"; `SHLVL`, `BASH_SUBSHELL`)
- Subshells/Command Grouping are your friend
  1. `( mycommand1; mycommand2; ) >out.txt 2>err.txt`
    - ★ to reduce whitespace confusion with `{}`, `[[`, `((`, etc. stick to convention
  2. Treated as Single Unit: File descriptor redirection, Signal handling, etc.
  3. Pipelines (`set -o pipefail` to debug)
  4. Handle `STDIO`, `STDOUT`, `STDERR` ( e.g. `0<&-`, `1>/dev/null`, `2>&1`)

# Shell comparisons

- `[[ ]]` preferred over `test`, `[`, etc.
  - ★ e.g. `if [[ $input =~ [0-9]+ ]]; then...`
  - ★ can use `==`, `!=`, etc. for both numeric and alpha comparisons, i.e. no need for `-gt` vs. `>`, etc.
  - ★ keyword, not built-in i.e. no word-splitting
  - ★ no need for double-quotes around variables if undefined
- `$ ( ( ) )` for arithmetic expressions

# Command Line Editing

- readline
  - ★ comes with bash
  - ★ rlwrap for your own REPL (`~/inputrc`)
  - ★ emacs (default) vs. vi bindings
- tab completion
  - ★ `/etc/bash_completion.d`
- history
  - ★ HISTIGNORE, HISTTIMEFORMAT, GLOBIGNORE, `set -o,`  
`shopt, ...`

# Directory navigation

- Visual representation (`tree`, etc.)
- Dired mode, ido-mode, `dirtree`, NeoTree (emacs), NerdTree (vim)
- `find` command (`-maxdepth`, pipe to `xargs`, `parallel`, etc.)
- Ad-hoc stack (`dirs; cd ~4`)
- Persistent stack (<http://code.google.com/p/go-tool/>)
  - ★ Best to rename/refactor if you have go lang installed!
- Symlinks (e.g. 'current' to point to your active project, version, etc.)
- CDPATH

# Prompts & Debugging

- Prompt
  - ★ PS1 (DSL: `\u`, `\w`, etc.; "add bling": Powerline, etc.)
  - ★ PS4 (tracing; e.g. `$LINENO`, etc.)
- Debuggability
  - ★ `set -vx` (Verbose tracing), `-errexit`, `-pipefail`
- Read man page (over and over again!; good at bedtime)

# Editor

- vim or emacs (what else?!)
  - ★ Grammar
    - ★ sed/awk, Perl, etc.
  - ★ Very rich featureset for anything textual
- DSL
  - ★ Emacs Lisp
    - ★ First-class objects (files, buffers, selections, etc.; Programmatic API: C-h r)
    - ★ Can also embed standalone run like sed (batch mode) in scripts
  - ★ vimscript



# Terminals

- Multiplexer
  - ★ `tmux`, `screen`
  - ★ Powerline integration
  - ★ Complex workspaces
    - ★ Pre-defined Session, Window, Pane layouts
  - ★ `tmuxinator`, `tmuxp`, `teamocil`, `etc.`

# Web!

- Major news sites within last 2 years (CNN, NPR, etc.)
- elinks
  - --dump for text; scriptable
- eww (emacs)
- curl, wget (Swiss-Army knives; ~/.curlrc, ~/.wgetrc)
- Filtering news sources
  - RSS (emacs elfeed, etc.)
  - Platform owners cutting back/shutting down programmatic access!

# Authoring

- Markup editors
  - Unobtrusive metacharacters retain human readability
  - Smart editors (only one command in emacs for rst)
  - `rst2pdf`, `rst2s5`, `rst2html`, etc.
  - Smart conversion tools, e.g. convert ‘--’ to emdash
- Full suite of text tools
  - `agrep` (approximate grep)
  - Fuzzy matching (Soundex, Levenshtein, etc.)

# Character Encodings

- ASCII (Take advantage of the country you're in!)
- `col -b`
- `unicode`
- Heuristically “normalize”/transliterate

```
# cat myfile.txt
```

```
Klüft “skräms” inför på fédéral
```

```
# cat myfile.txt | unicode
```

```
Kluft "skrams" infor pa federal
```

# Productivity

- Org Mode
  - ★ Organize your life in text!
  - ★ TODO, Agenda mode, MobileOrg,...
  - ★ Toolbox
- Works with Source Code and config files
  - ★ Literate Programming

# Parting Thoughts & Links

- Wisdom of UNIX Philosophy
  - ★ "Write programs to handle text streams, because that is a universal interface." - Doug McIlroy (Pipes)
- *Art of UNIX Programming* (<http://www.catb.org/esr/writings/taoup/html/> ; esp. Ch. 5 - Textuality)
- [https://en.wikipedia.org/wiki/Unix\\_philosophy](https://en.wikipedia.org/wiki/Unix_philosophy)
- Is the shell a "generative pattern language" (emergent behavior)?
  - ★ Christopher Alexander - Patterns in Architecture - YouTube