Introduction to dotfiles

And some tips

chanfr_v - delaho_h - sidore_m

December 7, 2017

Outline

Introduction

Hi!

Introduction

- On the PIE dotfiles are symlinks from `afs/.confs/`

Example

`.vimrc` is located at `afs/.confs/vimrc`

- These dotfiles are copied in `~/` thanks to `afs/.confs/install.sh`, which you must fill
- Be careful, dotfiles not copied in `afs/.confs/` will be removed at exit!

X Window System (X11)

- The X Window System (X11) is a windowing system for bitmap displays, common on UNIX-like computer operating systems.
- i3 is designed for X11.
- To load i3 you have the choice:
  - Boot on tty*, then execute `~/.xinitrc` with startx.
  - Display Manager.

Xresources

- Where you set your colours, your font, and more.
- Location: `~/.Xresources`

Naming

Xresources is just a naming convention, you can rename it according to your fantasies, then you will have to run:

```
42sh$ xrdb dotfile
```

Useful links

Links

- Arch wiki
- Nerd fonts: Gives an indecent amount of glyphs

Man pages

- X(1)
- xrdb(1)
- xset(1)

i3 Window Manager

- What is a window manager?
  - A graphical interface for the user.
  - Controls the placement and appearance of windows.
  - Might result in a desktop environment.
- What about i3?
  - Native, i3 is among the most ugly of them.
  - Light, highly customisable, he has a great potential.

Now, let's have a look at the dotfile.

.config/i3/config

- Where i3 configuration (fonts, borders. . .) is set
- Location of all the useful shortcuts, learn them!

Example

Try `$mod + R` to start the resize mode, then resize your window with `$mod + arrows`

- Useful place to start daemons with `exec --no-startup-id`

Some recommendations (not for the PIE)

- redshift: Schedule a screen redshift to rest your eyes
- numlockx: Activate numlock on bootup
- mons: Automatic monitors
- xautolock: Automatic lock screen
- udiskie: Automount USB devices

i3 bar

- Configured in `~/.config/i3/config`
- Change the content and the bar behaviour

Polybar

Polybar is a github project which replaces i3 bar. More customizable, allows click effects... Have a look!

Useful links

- The i3 reference manual
- Polybar

What is base 16?

"An architecture for building themes [...] using a base of 16 colors"
Templates are available for a bunch of IDE and programs, among them:

- i3
- Shell
- Xresources
- Vim
- Xcode
- Jetbrains (Rider, CLion, Pycharm...)

How to get/use Base 16?

First, install a builder (Many are available)
Then download and build the colorschemes

```
1       pip3 install pybase16-builder
2       mkdir -p $HOME/.dotfiles/base16 && cd $HOME/.dotfiles/base16
3       pybase16 update && pybase16 build
```

Now all the color schemes are downloaded
Read their README to know how to apply them

Base16 Shell

For instance, to get base16 in your shells, in your `.bashrc/.zshrc`:

```
1  BASE16_SHELL=$HOME/.dotfiles/base16/templates/shell/
2  [ -n "$PS1" ] && [ -s $BASE16_SHELL/profile_helper.sh ] &&\
3    eval "$($BASE16_SHELL/profile_helper.sh)"
```

Vim

- Written by Bram Moolenaar in C and Vim script
- Vi-IMproved, is a text editor program based on Vi.
- Released in 1991
- Free and OpenSource & GNU General Public License
- One of the most popular text editor on Linux

.vimrc

The `.vimrc` file is the main configuration file for vim

- Use vim script language
- YOU will learn how to fill it up
- This configuration file is used for:
  - ui
  - colors
  - indentation
  - tabulation behaviour
  - plugins
  - any vim script command
  - and much more...

.vimrc

Vimscript Syntax

- Most of the **options** are *setted* by the token **set**.
- A single " are used to mark comments.
- We recommend writing a single configuration modification per line. And to set up parts of your vimrc configuration. As it could expand to hundred of lines.

.vimrc

How to modify vim resources

- Directly in vim, in the command bar.
- **:vimscript**
- Like :
  - **:set number**
  - **:syntax enable**
- By writting vimscript in `~/.vimrc` or `/etc/vimrc` or `/etc/vim/vimrc`. But without the **:** of course.

UI Basic Settings

```
1  " Basic UI Config
2
3  set number          " set line numbers
4  set nu              " same
5
6  set showcmd         " show command in bottom bar
7  set ruler           " show cursor positon
8  set cursorline      " highlight current line
9  set wildmenu        " show all autocomplete of the vim commands
10 set showmatch       " highlight brackets, parenthesis.
```

Tabs & Spaces Basic Settings

```
1  " Basic tabs \& space config
2
3  set tabstop=2      " set numbers of spaces per TAB
4  set softabstop=2   " set number of spaces in tab when editing
5  set shiftwidth=2   " set tab to 2 spaces
6  set expandtab      " convert tabs to spaces
```

Indentation Basic Settings

```
1  " Basic indent options
2
3  set autoindent   " copy indentation of the previous line
4  set ai           " same
5
6  set smartindent  " more levels of indentation, nice on C files
7  set si           " same
8
9  set cindent      " C indentation, strict but customizable
10
11 " enable file type detection and associate its good indentation
12 filetype plugin indent on
```

More Simple & Cool Stuff

```
" More Vim options \& protips

set list          " by default, replace '\n' with visible '£'

set background=dark

set colorcolumn=80 " colorized the 80 column with a color
" the color is set by the colorscheme, or red by default

set spell    " check for English spelling

colorscheme your-favorite-colorscheme
" override vim default colors with custom colors

" Setting keymaps
map <KEY> :vimscript<CR>

" optimize vim output, to keep some static elements unchanged
set lazydraw
```

Survival Vim for Exams

SVE : for Newbies

```
1  " Survival VIM for Exams
2
3  set number
4  set expandtab
5  set softtabstop=2
6  set shiftwidth=2
7  syntax enable
8  set list
9  set listchars=trail:@
10 set autoindent
11 set smartindent
12 set colorcolumn=80
```

autocmd

autocmd

This feature of vim allow you to automate execution of vimscripts.

```
1  " Example of use of autocmd
2  autocmd bufnewfile *.example source /path/to/a/txt/file
3  " autocmd with bufnewfile keyword will execute the content of source
4  " at the creation of a new file, matching the globing '*.example'
```

file.txt

```
1  :insert
2  The content of this file will be written each time, you
3  create a new file with vim.
4  Typing the command:
5  42sh$ vim file.example
6  or
7  42sh$ vim xx.example
8  This is what the `insert' command do, it is quite useful.
```

Plugins

Multiple Package managers available:

- Vundle
- Pathogen
- vim-plug
- neobundle

Today we are going to present Vundle

Vundle

Simply run

`git clone http://github.com/VundleVim/Vundle.vim ~/.vim/bundle/Vundle.vim`

In your vimrc:

```
1     set nocompatible              " be iMproved, required
2     filetype off                  " required
3
4     set rtp+=~/.vim/bundle/Vundle.vim
5     call vundle#begin()
6
7     Plugin 'VundleVim/Vundle.vim'    " Plugin Manager
8
9     call vundle#end()            " required
10    filetype plugin indent on    " required
```

Plugin

Now you are able to install plugins, simply add them to your `.vimrc`

```
1   set nocompatible              " be iMproved, required
2   filetype off                  " required
3
4   set rtp+=~/.vim/bundle/Vundle.vim
5   call vundle#begin()
6   Plugin 'VundleVim/Vundle.vim'      " Plugin Manager
7   Plugin 'vim-airline/vim-airline'   " Statusbar
8   Plugin 'vim-syntastic/syntastic'   " Syntax Checking
9   Plugin 'chriskempson/base16'       " Base16 Colorscheme
10  if filereadable(expand("~/.vimrc_background"))
11    let base16colorspace=256
12    source ~/.vimrc_background
13  endif
14  let g:syntastic_c_compiler_options= \
15    '-Wall -Werror -Wextra -pedantic -std=c99'
16  call vundle#end()             " required
17  filetype plugin indent on     " required
```

Recap

Our githubs (contains the dotfiles)

- chanfrv/dotfiles
-
-

Questions

Questions?