vcsh

manage config files in $HOME via fake bare git repositories

Richard Hartmann,
RichiH@{freenode,OFTC,IRCnet},
richih.mailinglist@gmail.com

2012-02-04

Outline

Outline

Who am I?

- Project & Network Operations Manager at Globalways AG
- freenode & OFTC staff
- Passionate about FLOSS
- Author of vcsh

What is git?

- Version control system
- Distributed
    - No need for central repository
    - Allows you to commit while offline
- Full history in every checkout
- Best version control system available (imo...)

Outline

What is vcsh?

- Implemented in POSIX shell; portable
- ”version control shell” or ”version control system `$HOME`”
- Based on git
  - git unable to maintain several working copies in one directory
  - Sucks if you want to keep your configs in git
- vcsh uses fake bare git repositories to work around this
- Think of it as an extension to git

fake bare.. what?

- Normal git repo:
  - working copy in `$GIT_WORK_TREE`
  - git data in `$GIT_WORK_TREE/.git` aka `$GIT_DIR`
- Bare git repo:
  - git data in `$GIT_DIR`
  - no `$GIT_WORK_TREE`
- Fake bare git repo:
  - working copy in `$GIT_WORK_TREE`
  - git data in `$GIT_DIR`
  - `$GIT_WORK_TREE == $HOME`
  - `$GIT_DIR == $XDG_CONFIG_HOME/vcsh/repo.d/$repo.vcsh`
  - `core.bare = false`

Problems with fake bare git repos

- Fake bare repos are messy to set up and use
- Reason why git disallows shared `$GIT_WORK_TREE`: complexity due to context-dependency
- Mistakes lead to confusion or data loss; imagine `$GIT_WORK_TREE` set and
    - `git add`
    - `git reset --hard HEAD~1`
    - `git checkout -- *`
    - `git clean -f`

Solution: vcsh

- Wraps around git
- Hides complexity and does sanity checks
- Several git repos checked out into `$HOME` at once
  - One repo for zsh, vim, mplayer, etc
  - Enables specific subsets of repos per host
- Manages complete repo life-cycle

Outline

Create new repo

```
# create new repo
vcsh init vim
# add files to it
vcsh run vim git add .vim .vimrc
# commit using shorthand form
vcsh vim commit
# push using longhand form
vcsh run vim git push
```

Made-up life-cycle

```
# clone repo into new name zsh
vcsh clone git://github.com/RichiH/zshrc.git zsh
# optionally update legacy repos
vcsh setup zsh
# display all files managed by this repo
vcsh run zsh git ls-files
# rename repo just because
vcsh rename zsh zshrc
# delete repo
vcsh delete zshrc
```

run vs enter

```
# do everything from outside
vcsh run zsh git add .zshrc
vcsh run zsh git commit
vcsh run zsh git push
# the same, but from within
vcsh enter zsh
git add .zshrc
git commit
git push
exit
```

Playing nice with others

- shells can display exported `ENV` in `$PROMPT`
- vcs_info
- mr via plugin, mainline soon
- git-annex to manage non-configuration files
- Simple but powerful hook system

Outline

Future work

- More unit tests
- Get vcsh into more distributions
- Spread awareness to reach critical mass
- Maybe extend support to subversion, mercurial, etc

Outline

Where to get it

- `git clone git://github.com/RichiH/vcsh.git`
- Native packages for
  - Debian
  - Ubuntu
  - Arch Linux (AUR)
- Small bug in README.md, use v0.20120203 or git

Thanks!

Thanks for listening!

Questions? Follow me outside when my time-slot is over.

See slide footer for further contact Information.