

Le shell : ZSH et divers

Bruno Bonfils, <asyd@asyd.net>

Mars 2008

Disclaimer

- Basé sur mon expérience personnelle
- Pas de troll pendant la présentation, merci !

Au programme

- Le shell en général (retour d'expérience)
- ZSH :
 - Le scripting
 - La complétion
 - ZLE (ZSH Line Editor)

Rappel

- Le shell n'est pas uniquement destiné à exécuter des commandes mais également (et surtout ?) à traiter des données
 - Extraction de colonnes,
 - Modification de masse
 - Calculs

Organisation

- ~/ : une hiérarchie simple mais efficace
 - download
 - tmp
 - travail
 - docs
 - bin
 - tools
- Utilisation d'un SCM

Utilisation

- Outils indispensables
 - screen
- Le cœur du shell : l'interpréteur (zsh !)

Les outils

- Les plus évidents
 - grep, find, xargs
 - sed, awk, perl
- Les moins connus
 - socat
 - dict

Les outils

- screen
 - multiplexage / splitting
 - recherche dans buffer
 - hardstatus / title bars
 - copier/coller sans souris

ZSH

- Shell axé interactif
- Maintenant disponible de base sur la plupart des Unix (sauf Linux...)

ZSH

- Quelques points forts de zsh
 - pattern matching
 - substitution

zsh : Pattern matching

Oubliez find !

- Afficher l'ensemble des éléments d'un répertoire et de ses sous répertoires
 - `ls **/*`
- Même chose, mais uniquement pour les fichiers
 - `ls **/*(.)`
- Finalement non, je ne veux que les répertoires
 - `ls -d **/*(/)`
- Autre exemples
 - `ls *s*`
 - `ls *s*(#i)`
 - `ls *~*.(#i)(png|jpg)`

Substitution

```
% var=/tmp/toto.txt
```

```
% echo ${var:flag}
```

h	Équivalent à dirname (/tmp)
r	Suppression de l'extension : \$ for file in **/*.jpg ; do convert \$file \${file:r}.png ; done
t	Équivalent à basename (toto.txt)
l	Miniscule
u	Majucule

Substitution

C: Capitalize
U: Upper case
L: Lower case

```
% echo ${(flag)variable}
```

o, O, n	Tri
u	Unicité
C,U, L	Modification de la casse
s:pattern:	Splitting
j:pattern:	Joining

zsh : Scripting

Un exemple pratique

- Obtenir la liste des alias définis dans le fichier `~/.muttrc` pour en faire un tableau trié (exemple d'utilisation : fonction de complétion `_mutt`)

alias asyd Bruno Bonfils <asyd@asyd.net>

- Équivalent à :
 - `# grep '^alias' ~/.muttrc | awk { print $2 } | sort -u`

zsh : Scripting

Un exemple pratique

- Créer un tableau de lignes à partir d'un fichier : `buffer=(${(f)"$(< ~/.muttrc)"})`
- `list=(${(M)buffer##alias})`
- `list=(${(o){buffer/alias /}%% *})`

zsh : Scripting un second exemple

```
% var="abc" ; print ${(j::){(Oas::)var}}
```

cba



Oa : reverse order

zsh : la complétion

- La complétion vue par zsh :
 - réutilisable : la complétion ne se fait pas uniquement commande par commande, mais surtout par type de paramètres (hôtes, utilisateurs, groupes, paquets, etc.)
 - configurable à un très haut niveau

zsh : la complétion

- Une définition pour plus de 1500 commandes (tout OS confondus)
- UNIX, Windows, OS-X

zsh : la complétion

- Une complétion parfois très poussée :
 - Liste des modules perl (perl -M<tab>, man <tab>, etc.)
 - Liste des paquets (distinction entre paquets disponibles/installés)
 - target Make/ant
- Utilisation d'un cache possible

zsh : la complétion

- La complétion se configure d'une manière vraiment souple, via l'utilisation des `zstyle` :
 - `zstyle ':completion:*' use-ip true`
 - `zstyle ':completion:*' use-cache on`
 - `zstyle ':completion:*' cache-path ~/.zsh/cache`

zsh : la complétion

- `zstyle ':completion:*:(all-|)files' ignored-patterns '(|*/)CVS'`
- `zstyle ':completion:*:cd:*' ignored-patterns '(*/*)#CVS'`
- `zstyle ':completion:*' ignore-parents parent pwd`

zsh : la complétion

- `zstyle ':completion:*:*:gqview:*' file-patterns '*(-/):directories :(#i)*.(png|jpeg|jpg):all-files'`
- Pas de fonction de complétion pour `gqview`, mais je limite à la complétion des répertoires et des fichiers images, tolérant à la casse
- `zstyle ':completion:*:*:vi*:*' file-sort modification`
- le premier fichier proposé lors d'un `vi <tab>` sera le dernier fichier modifié

zsh : la complétion

- L'apparition des commandes aptitude-bundle-* m'a rendu fou (complétion multiple sur sudo apt<tab>) donc :
- `zstyle ':completion:*:complete:-command-::commands' ignored-patterns 'aptitude-*`
- `zstyle ':completion:*:*:sudo:*:commands' ignored-patterns 'aptitude-*`

zsh : la complétion

- `zstyle :completion:<widget>:<completer>:<context>:<argument>:<tag> <style>`
- 7 widgets
- 11 completer (complete, approximate, match, etc.)
- ~90 tags
- ~100 styles (file-patterns, file-sort, list, users, etc.)

zsh : la complétion

- Définir une complétion très simple pour un programme (ex: script nécessitant un hôte en paramètre)
- `#compdef _hosts backup.sh`

zsh : zle

- Les interactions entre l'utilisateur et le moteur se font via la ZLE (ZSH command Line Editing)
- Des modes vi, emacs
- Très facile de modifier les bindkeys de base

zsh : zle

- Quelques widgets à connaître :
 - push-line
 - insert-last-word
 - history-search-{backward|forward}

zsh : zle

- Il est même très facile de créer ses propres widgets
 - Création d'une fonction
 - Associer cette fonction à un widget
 - Binder ce widget

zsh : zle

```
regex-edit () {  
    local REPLY words  
    words=(${=BUFFER})  
    read-from-minibuffer "Regexp:"  
    echo $REPLY  
    if [ -n $REPLY ]; then  
        BUFFER="$words[1] $(echo $words[2,${#words}] | sed ${REPLY})"  
    fi  
}  
  
zle -N regex-edit  
  
bindkey "^[e" regex-edit
```

zsh

- Quelques autres fonctionnalités en vrac
 - suffix alias (./fichier.jpg)
 - complétion colorée
 - pile pour la gestion des répertoires
 - complétion intelligente sur les URL, gestion des bookmarks

zsh : exemple de configuration

```
% svn co http://svn.asyd.net/svn/zsh/trunk .zsh
```

```
% ln -s .zsh/zshrc .zshrc
```

```
% vi .zsh/README
```

- Un fichier de configuration par
 - OS
 - Hôte

zsh : exemple de configuration

- Mise à jour dynamique de la hardstatusbar, titlebar de screen (ssh, telnet, etc.)
- Quelques hacks sur des fonctions de complétion

Plus d'infos

- Doc de ZSH (23000 lignes de man, contre à peine plus de 5000 pour bash)
- <http://zshwiki.org/home/>
- <http://cli.asyd.net/home/>
- echo subscribe | mail shell-request@asyd.net