# angel-PS1

15 juin 2013

Journées Perl 2013

# Olivier Mengué (DOLMEN)

- Développeur Perl professionnel depuis 2009
  - Perl embarqué dans la box domotique IJENKO
  - AnyEvent..
  - Plus de 300 tickets rapportés sur rt.cpan.org
  - Contributeur Dist::Zilla cœur et plugins
  - 21 distributions sur le CPAN
- Développeur shell Unix depuis 1993
  - 4 ans où **`ksh`** était mon langage principal au boulot
  - Journées Perl 2008 : « Mieux programmer en Shell »
  - Commit bit sur **Liquid Prompt**

# PS1 ?

# PS1 ?

Définit l'affichage de l'invite du shell Unix

```
dolmen:~ $ _
```

# PS1

## Montre l'état

de l'environnement dans lequel vous allez exécuter une commande

# PS1 : exemple bash

```
PS1='\u:\w \$ '
```

Affiche :

- le nom de l'utilisateur
- le chemin abrégé (~ pour $HOME)
- '$' ou '#' selon si utilisateur normal ou root

Exemple :

```
Dolmen:~/Bureau $ _
```

# PS1 dynamique

- Utilisation de variables du shell :

```
PS1='$RANDOM \w \$'
```

- Appel de commandes (bash) :

```
PS1='$(jobs | wc -l) \w \$'
```

```
PS1='\w$(__git_ps1) \$ '
```

- Appel d'une fonction shell qui modifie PS1 avant son évaluation :
  **`PROMPT_COMMAND`** (bash) / **`precmd`** (zsh)

# angel-PS1

- Générateur de PS1 en Perl
  - Framework de construction de votre PS1
  - Plugins
- Un ~~démon~~ ange compagnon du shell

# angel-PS1

```
dolmen:~/Code/angel-PS1 $ eval `./angel-PS1`
angel installed.
16:16:40 pts/0 :~/Code/angel-PS1 devel* $
```

Configuration par défaut version pré-alpha

# Angel-PS1 : objectifs

- Framework pour construire son prompt
- Plugins
- Perl & CPAN !
- Couleurs faciles
- Distribution via Github et CPAN
- Suite de tests
- Accessible aux non-perleux !

# Configuration

~/.config/angel-PS1/PS1.pl

Exemple :

```
use AngelPS1::Plugin::Git;

(
        '\w',
        ' ',
        \&GitInfo,
        '\$',
        ' ',
)
```

API
pre-Alpha !

# Implémentation

# angel-PS1

- Créer des pipes pour la communication
- Générer le code shell qui utilisera ces pipes

# Communication

- Deux pipes nommés :

```
my $FIFO_PATH = tmpnam().".$NAME-$$";
my $FIFO_IN = "$FIFO_PATH.in";
my $FIFO_OUT = "$FIFO_PATH.out";
mkfifo($FIFO_IN, 0600);
mkfifo($FIFO_OUT, 0600);
```

# Côté shell

```
-angel-PS1()
{
  local err=$?;
  printf '%s\0%s' "?=$err" "PWD=$PWD"
      > $FIFO_IN
  read PS1 < $FIFO_OUT
}
PROMPT_COMMAND='-angel-PS1'
```

Code simplifié sans gestion d'erreur

# Côté Perl

- Imprimer le code shell
- fork()
- Boucle, tant que le shell parent existe
  - Lit le pipe d'entrée
  - Construit PS1
  - Écrit sur le pipe de sortie

# Côté Perl

```
for(;;) {
  open my $in, '<', $FIFO_IN or die;
  my %state =
    map { ( m/^(.*?)=(.*)/s ) }
    split /\0/
    do { local $/; <$in> };

  $PS1 = process(\%state);

  open my $out, '>', $FIFO_OUT or die;
  print $out $PS1;
}
```

Pseudo code

# Contraintes

- Cible : large base d'utilisateurs Unix
  - Non familiers avec Perl
  - Non familiers avec le CPAN
    - Installer angel-PS1 à partir du CPAN ? Trop compliqué !
    - Installer des plugins à partir du CPAN ? Trop compliqué
  - Vieux Perl : Perl système
  - Objectif : installation facile
- Mais on veut garder la modularité pour le développement
  - Espaces de noms : AngelPS1::Plugin, AngelPS1::Util...
  - Séparer en plusieurs fichiers pour le développement, pour les tests unitaires
  - Profiter du chargement dynamique : ne pas compiler le code qui ne sera jamais exécuté

# Installation facile

- Objectif : fichier unique
  - Téléchargeable rapidement avec git/curl/wget
  - Disséminable simplement sur une nouvelle machine avec scp
- Solution : construire un source complet à partir d'un script et de modules
  - App::FatPacker !
  - Embarquer les modules AngelPS1::
  - Permettra d'embarquer des modules pur-Perl du CPAN

# Distribution

- Angel-PS1 installable à partir de :
  - Github : git clone
  - CPAN : pour les perleux !

# Distribution via Github

- La branche **`devel`** est la branche « source »
  - `bin/angel-PS1` : le script « main »
  - `lib/AngelPS1/*.pm` : les modules
- **`build.pl`** construit `/angel-PS1` avec FatPacker
- **`build.pl -r`** le commite dans la branche **`release`**
- La branche **`release`** :
  - branche visible sur Github (branche « par défaut »)
  - ne contient que angel-PS1 construit et la doc
  - la branche par défaut avec **`git clone`**

# Vieux Perl

- Restriction à 5.8.3 pour le cœur
  - Au niveau langage
  - Au niveau modules (vérification avec Module::CoreList)
- Pas de restriction pour les plugins, sauf ceux de la configuration par défaut
- Profiter *éventuellement* des modules installés (dépendances optionnelles) : chargement dynamique
- Le build FatPack nous permettra éventuellement d'inclure des modules pur Perl

# L'API des plugins

Désolé, c'est pas prêt :(

# L'API des couleurs

Désolé, c'est pas prêt :(

# Contribuer

Pour les patchs, c'est pas encore mûr :(

Pour les bugs, envoyez !

# Conclusion

- Projet jeune à suivre
- → OSDC.fr
- Jettez un œil au code de build.pl
- Utilisez LiquidPrompt pour l'instant

github.com/dolmen/angel-PS1

L'ange gardien de votre shell