# Quelques informations sur *perl*

Olivier Aubert, `Olivier.Aubert@enst-bretagne.fr` 18 Juillet 1996

Ce document donne un certain nombre de points de départ concernant *perl*. C'est une introduction générale au langage, mais pas à sa programmation. Des références vers d'autres documents complémentaires sont données tout au long de l'article.

## Contents

# 1 Introduction

## 1.1 Présentation

*perl* est un langage interprété (avec une phase interne de pré-compilation) optimisé pour traiter des fichiers texte, mais qui peut également être utilisé pour diverses tâches d'administration-systèeme.

Sa syntaxe s'inspire très largement de celles de *C*, *awk*, *sed* et *sh*, ce qui rend son apprentissage très facile pour toute personne ayant un peu d'expérience avec ces outils. Néanmoins, pas d'affolement si vous ne les connaissez pas, vous pourrez apprécier certains concepts avec un oeil neuf.

Il n'y a aucune limitation sur la taille des données ou sur leur contenu (une chaîne peut contenir le caractère nul, et la gestion de la mémoire qu'elle utilise n'est pas à la charge du programmeur).

Une grande quantité de modules permet de lui ajouter rapidement diverses extensions (CGI, Tk, Msql, POSIX, Curses, NNTP, etc).

Un des ses avantages est qu'il permet d'écrire rapidement des applications puissantes qui peuvent tourner immédiatement sur plusieurs plates-formes différentes. Son utilisation va donc du prototypage rapide d'applications au développement complet de programmes divers et portables.

## 1.2 Historique

*perl* a été conçu en 1987 par Larry Wall. C'est à la même personne que l'on doit le lecteur de news `rn` ainsi que l'utilitaire `patch`, bien connu des Linuxeurs/iens/istes. Il avait besoin à l'époque de générer des rapports sur le trafic engendré par un système analogue aux News actuelles, et les outils dont il disposait à l'époque étaient soit pas assez puissants, soit pas assez souples.

Il a donc conçu un langage adapté à ses besoins, c'est-à-dire capable d'analyser des données textuelles pour concevoir des rapports, d'où l'acronyme utilisé: *Practical Extraction and Report Language*. Cette signification *politiquement correcte* n'est apparue, d'après le folklore, qu'après le terme plus précis: *Pathologically Eclectic Rubbish Lister*.

Ayant mis sa création à disposition d'autres personnes, il a ensuite intégré de nombreuses fonctionnalités (appels systèmes, etc) qui lui étaient demandées, ce qui a conduit le langage a être utilisé par de nombreux administrateurs-système.

## 1.3 Problème de version ?

Ces derniers temps ont vu une nouvelle source de problèmes émerger dans la communauté *perl*, à l'occasion de la sortie de la version 5 du langage. La version précédente, généralement 4.036, était alors utilisée depuis quelques années, quand la nouvelle version est sortie. Celle-ci apporte de nombreuses améliorations, directement visibles ou plus internes, tout en restant compatible avec les anciennes versions du langage (il faut noter que certains scripts écrits pour *perl* version 1 fonctionnent encore sans changement avec la version actuelle).

La compatibilité n'est cependant pas totale : un certain nombre d'incompatibilités mineures ont été introduites et peuvent obliger à reprendre certains scripts. L'exemple le plus courant est le fait que le caractère

`@` doit maintenant être précédé d'un *backslash* à l'intérieur d'une chaîne de caractères, afin que le `@` ne soit pas interprété comme un préfixe de tableau.

L'ensemble des incompatibilités, ainsi que les pièges les plus courants pour les personnes utilisant *C* ou *awk* sont présentés dans la page de manuel `perltrap`

Cet article est consacré principalement à *perl5*, et plus spécifiquement *perl5.003* ou supérieur. Cette dernière version est en effet celle qui est la plus pratique à utiliser ou installer, et la seule activement maintenue.

### 1.4 Qu'apporte perl5 par rapport à la version précédente ?

Voilà quelques unes des améliorations les plus évidentes:

- des structures de données complexes, grâce à l'introduction des références ;
- une approche orientée-objet, dont l'intérêt n'est pas simplement académique: elle a permis en effet le développement de divers 4.1 rendant l'écriture des scripts encore plus simple et plus rapide ;
- les expressions régulières sont maintenant plus puissantes, et la possibilité est offerte de placer des commentaires à l'intérieur, afin d'expliquer leur fonctionnement ;
- la possibilité d'ajouter facilement de nouvelles fonctionnalités venant de librairies C grâce au langage d'extension *XS* (voir les pages de manuel `perlxs` et `perlxstut`).

## 2 Où trouver les fichiers, indispensables ou superflus ?

### 2.1 Sur le WWW

Le principal site WWW sur *perl* est maintenu par Tom Christiansen, un des gourous de *perl*. Son adresse est `<http://www.perl.com/>`. Il contient de nombreux documents sur tous les aspects de *perl*, du plus technique au plus général en passant par les inévitables guerres de religion entre les partisans de différents langages (*perl*, *python*, *tcl*, etc).

De nombreux pointeurs vers d'autres sites y sont également présents. C'est donc une mine de renseignements, que l'on peut également visiter uniquement par curiosité.

Il propose également un service permettant d'accéder de manière transparente au site 2.2 le plus près de chez vous.

### 2.2 Par `ftp`

Fin 1995, le besoin s'est fait sentir de regrouper les efforts de la communauté *perl* afin d'éviter d'avoir à réinventer la roue à chaque fois. De cette idée est né le CPAN, pour *Comprehensive Perl Archive Network*, d'un principe analogue au CTAN pour TeX.

Le site principal `<ftp://ftp.funet.fi/pub/languages/perl/CPAN/>` centralise les sources, la documentation ainsi que les nombreux modules existants. Il est ensuite *mirrorisé* sur l'ensemble des sites constituant le CPAN, assurant ainsi un temps d'accès convenable.

En France, il existe deux sites CPAN:

- `<ftp://ftp.ibp.fr/pub/perl/CPAN/>`
- `<ftp://ftp.pasteur.fr/pub/computing/unix/perl/CPAN/>`

La lecture du fichier ROADMAP présent dans le répertoire principal de ces sites est un bon point de départ pour se retrouver dans son architecture.

Si vous cherchez un module particulier, allez voir la liste des modules, postée régulièrement dans le groupe `comp.lang.perl.modules` et disponible sur les sites 2.2.

## 2.3 Par Usenet

*perl* est également présent dans l'arboresence Usenet. Le groupe initial, `comp.lang.perl` a subi en 1995 une réorganisation, suite au trafic important qu'il supportait, et les groupes suivants sont maintenant disponibles:

`comp.lang.perl.announce`
: Groupe modéré à faible trafic, destiné à recevoir les annonces concernant *perl*, ses modules ainsi que les FAQs ;

`comp.lang.perl.misc`
: Groupe général sur *perl* ;

`comp.lang.perl.modules`
: Utilisation et développement des modules ;

`comp.lang.perl.tk`
: À propos du module Tk de *perl*.

# 3 Où trouver de la documentation ?

## 3.1 En ligne

### 3.1.1 Accès

Depuis la version 5 de *perl* un effort particulier a été porté sur la documentation en ligne. L'unique page de manuel `perl` a été réécrite et séparée en plusieurs fichiers, afin de faciliter sa consultation. Pour voir les différentes pages disponibles, il suffit de rentrer la commande

```
% man perl
```

ou encore, si les pages de manuel n'ont pas été installées correctement sur votre système

```
% perldoc perl
```

### 3.1.2 Organisation

Si l'on a l'intention de lire l'ensemble des pages (ce qui se fait très bien), il est conseillé de les lire dans l'ordre proposé. Si l'on cherche la syntaxe d'une commande particulière, il faut consulter le document `perlfunc`. Pour une référence complète sur les expressions régulières, consulter le document `perlre`

### 3.1.3 Documentation des modules

*POD*, un nouveau format de documentation a été introduit dans *perl5*. Il signifie *Plain Old Documentation* et sa syntaxe est décrite dans le document `perlpod`.

C'est un format très simple, ne disposant pas de grandes possibilités mais largement suffisant pour écrire des pages de manuel. De nombreux convertisseurs existent pour le convertir ensuite dans d'autres formats: texte, nroff, LaTeX, html. On dispose ainsi d'une documentation unique et facilement consultable quel que soit le moyen dont on dispose.

Un autre de ses avantages est qu'il est prévu pour être directement intégré dans du code *perl*. Ainsi, la plupart des modules contiennent leur propre documentation. Pour accéder à la documentation du module `File::Copy` par exemple, il suffit d'exécuter la commande

```
% perldoc File::Copy
```

Il est possible d'appliquer les convertisseurs du format *POD* (`pod2txt`, `pod2html`, `pod2man`, `pod2html`) directement sur les modules ou les scripts, afin d'en extraire cette documentation.

## 3.2 Sur le WWW

J'ai déjà parlé du site `<http://www.perl.com/>`, qui contient des pointeurs vers de la documentation en ligne ou vers divers livres consacrés à *perl*.

De nombreux tutoriaux, le plus souvent en anglais, sont également disponibles. Il suffit de rentrer les termes de recherche *perl* et *tutorial* dans un moteur de recherche pour les obtenir. Je ne peux pas vraiment en conseiller un, étant donné que la documentation standard de *perl* m'a toujours suffi.

## 3.3 De vrais livres

De nombreux livres existent, du livre de référence au simple tutorial. La liste la plus à jour de ces ouvrages est disponible sur le site WWW `<http://www.perl.com/>`, à l'URL `<http://www.perl.com/perl/info/books.html>`.

Les ouvrages de référence, car écrits par Larry Wall, l'auteur du langage, et d'autres spécialistes (Tom Christiansen, Randal L. Schwartz), sont:

**Programming Perl**

aussi appelé le *Camel Book* à cause de l'image de dromadaire présente sur sa couverture. C'est l'ouvrage de référence pour *perl*. Une version est en cours de réécriture pour intégrer les nouvelles fonctionnalités de *perl5*. Sa date de sortie est prévue pour septembre 1996.

**Learning Perl**

C'est un tutorial, couvrant uniquement les possibilités de *perl4*. Une traduction française est disponible.

*(Attention, pub)* J'ai également écrit un document d'introduction à perl en français, dont les versions Postscript et dvi sont disponibles à l'adresse `<http://www-eleves.enst-bretagne.fr/ aubert/perl/>`.

# 4 Informations complémentaires

Dans cette section se trouvent - pour l'instant - des informations pouvant être utiles aux personnes désirant installer des modules. D'autres informations plus spécifiques pourront y être ajoutées, sur demande.

## 4.1 Installation de modules

L'introduction des modules et du langage d'extension *XS* a grandement facilité la vie des programmeurs ainsi que des utilisateurs. Un mécanisme (`MakeMaker`) a été développé afin de rendre l'installation de nouveau modules la plus simple possible. J'en parlerai ici du point de vue de la personne qui doit installer un nouveau module. Ce que je décris est valable surtout pour les systèmes gérant les librairies dynamiques. La marche à suivre pour les autres systèmes (librairies statiques) est presque identique mais impose de recompiler une nouvelle version de l'interpréteur `perl`.

### 4.1.1 Démarche générale

Voyons dans un premier temps la démarche générale, adaptée pour tous les modules simples. Après avoir détarré l'archive du module et être allé dans le répertoire ainsi créé, il suffit de procéder ainsi:

```
% less README
% less INSTALL
% perl Makefile.PL
% make
% make test
% make install
```

### 4.1.2 Un exemple plus développé

Prenons l'exemple du module *Curses* qui procure des routines permettant de gérer la gestion d'un écran en mode texte. Il ne vient pas avec la distribution standard de *perl*, il faut donc aller le récupérer sur un des sites 2.2 ; il se trouve dans la catégorie *User Interfaces*.

Il faut ensuite détarrer le fichier archive rapatrié dans un répertoire quelconque. Rappelons aux utilisateurs de Linux qu'il est déconseillé de compiler quoi que ce soit en étant root : mieux vaut le faire en tant qu'utilisateur normal, puis passer root pour faire l'installation uniquement.

Dans le répertoire `Curses-1.0` ainsi créé se trouvent plusieurs fichiers et répertoires dont nous allons voir les plus importants.

En premier lieu, bien lire les fichiers `README` et `INSTALL`. La marche à suivre y est indiquée. Dans le cas de *Curses* qui est un paquetage variant beaucoup d'une machine à l'autre, il faut copier à la main le fichier du répertoire `hints` correspondant à son architecture dans le fichier `c-config.h` et éditer le fichier `Makefile.PL` pour décommenter les lignes relatives à son système d'exploitation.

Je vous rassure, *Curses* fait partie des modules impliquant le plus de manipulations. Dans la plupart des cas, comme par exemple pour *GD* ou *Msql*, il suffit d'avoir auparavant installé les librairies adéquates sur son système si elles n'y étaint pas déjà.

Il faut ensuite lancer la commande

```
% perl
Makefile.PL
```

qui va faire un certain nombre de tests et créer le fichier `Makefile`. Un simple `make` doit ensuite suffire à construire l'extension.

`% make test` lance une série de tests afin de vérifier que le module est correctement compilé. Il suffit enfin de passer root pour faire le `% make install` qui ira installer les fichiers nécessaires dans les bons répertoires (généralement, `/usr/local/lib/perl5/site_perl/` ou `/usr/lib/perl5/site_perl/`).

## 4.2 Comment installer un module sans avoir les droits root ?

Les modules sont installés par défaut dans le répertoire standard des librairies de *perl*. Si vous avez besoin d'un module mais que vous n'avez pas les droits sur l'arborescence `/usr/local`, il est toujours possible d'installer des modules dans votre répertoire personnel, en précisant pour cela le répertoire adéquat lors de la génération du fichier `Makefile`:

```
% perl Makefile.PL PREFIX=~
```

ce qui va installer les modules dans le répertoire `$HOME/lib`. Pour y accéder, il suffit de commencer le script par:

```
#!/usr/local/bin/perl

use lib "$ENV{HOME}/lib";
```

La documentation complète des attributs d'installation pouvant être fixés est disponible par la commande:

```
% perldoc ExtUtils::MakeMaker
```