

Le Shell ou les shells ?

L'univers culturel du shell

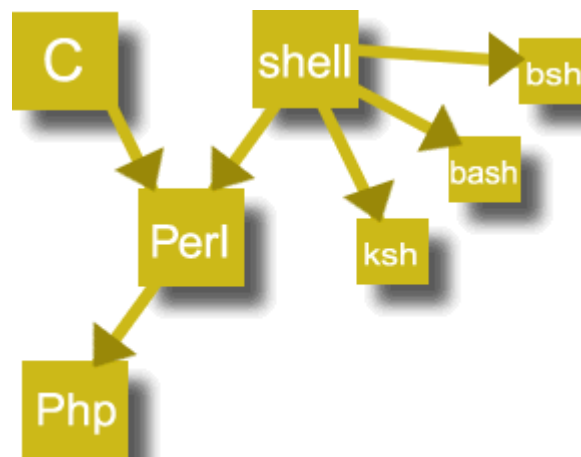
Le shell script est-il un langage ?

On est en droit de se poser la question de savoir si un langage commandes est à proprement parler un langage de programmation. Si le shell est considéré seulement comme une syntaxe de passage d'ordres à une ligne de commande, il y a bien sûr la notion de langage (induite par la notion de syntaxe), mais ce langage ne serait pas à proprement parler un langage de programmation.

A partir du moment où on rajoute des variables et des structures de contrôle, il est légitime de parler de langage de programmation. Le "shell script" n'est pas un langage "complet", en ce sens qu'il ne permet pas de "tout programmer".

L'ensemble des langages basés sur le shell

Le shell fait partie d'un ensemble de langages *text-oriented*, c'est-à-dire, dont le type principal des données manipulées est du texte (potentiellement à valeur numérique). Le langage est forgé sur ce principe. Il appartient par certains aspects à la famille des langages non typés qui contient le Perl et le Php. Cette famille de langages peut être représentée ainsi :



- Le C conçu par Brian Kernighan et Dennis Ritchie est le langage d'origine du système UNIX, puis LINUX.
- Le shell est le langage de commande interprété par un interpréteur C.
- Le Perl est une fusion des deux langages. Il hérite du C sa syntaxe de blocs et du shell la gestion des variables, le caractère interprété et un bon nombre de syntaxes d'opérateurs et de variables prédéfinies.
- Le PHP est une simplification du Perl qui transforme en fonctions simples (C-like) les syntaxes complexes du Perl. On peut dire que Php élimine le shell du Perl.

Le shell en général

shell et shell script : bien faire la différence

Le **shell** désigne l'interpréteur de commandes qui décrypte et exécute les commandes du langage de shell. Le "shell" désigne donc à la fois un programme exécutable et une syntaxe. L'essentiel du shell est déjà connu sous forme des commandes UNIX. Il existe cependant un certain nombre de syntaxes supplémentaires qui permettent d'en faire un langage d'écriture de scripts : le shell script.

Le **shell script** est l'écriture de langage shell dans un fichier source, qui peut être interprété par le programme shell pour exécuter une séquence de commandes système. Le shell script est donc un "source" contenu dans un fichier.

Le shell de base : sh

Le shell est un langage destiné aux administrateurs système, pour automatiser la gestion du système et de ses ressources. Il est donc prévu :

- pour l'efficacité syntaxique : écrire peu, donc écrire vite
- pour la puissance des instructions : écrire peu, faire beaucoup

La conséquence est que les scripts shell sont rarement verbeux, donc difficiles à lire. La syntaxe est très compacte et se base sur un jeu d'opérateurs extrêmement puissants et des "raccourcis" syntaxiques très condensés.

Les différentes versions

Plusieurs shells sont apparus sur le marché, de façon concurrente. Ils diffèrent tous plus ou moins par certaines constructions syntaxiques et des commandes "natives". Tous les shells ne sont pas disponibles sur tous les SE. Le choix du "shell" peut être effectué par l'administrateur système et attribué "par défaut" à tout ou partie des utilisateurs.

Cshell (csh)

Le CShell a été écrit par Bill Joy à l'Université Californienne de Berkeley. L'intention principal était de construire un interpréteur de commande dont la syntaxe se rapproche le plus possible de langage C originel.

Les fonctionnalités principales du C shell sont

- Environnement modifiable.
- Commandes abrégées. (Aliases.)
- Historique. (Se rappelle des commandes tapées.)
- Contrôle des tâches. (Fait tourner des programmes en avant-plan ou en arrière-plan.)
- Programmation shell. (One can write programs using the shell.)
- Raccourcis claviers.
- Structures de contrôle. (par exemple, boucles et structures conditionnelles.)
- Vérification d'existence et des droits sur les fichiers.
- Assignation de variables.
- Variables prédéfinies.

Bourne SHell (bsh)

Le Bourne shell ou sh, était le shell par défaut d'Unix, il remplaçait le [Thompson shell](#) dont le fichier exécutable avait le même nom : sh. Le Bourne shell a été programmé par Stephen [Bourne](#) (d'où le nom) des laboratoires Bell et a fait son apparition pour la première fois en 1977 sur la version 7 d'Unix qui était distribué aux universités et académies.

Le principal apport du Bourne Shell est le système de "tubes" (pipe) qui permet d'enchaîner une sortie d'une commande sur l'entrée d'une autre commande.

Korn SHell (ksh)

Le korn shell regroupe les fonctions du C shell et du bourne shell, tout en apportant de nouvelles propriétés, afin d'obtenir un shell plus convivial, plus puissant et plus rapide.

Le korn shell a les possibilités supplémentaires suivantes :

- un historique des commandes peut-être mis en place et utilisé (vi ou emacs).
- une compatibilité bourne shell.
- des variables d'environnements supplémentaires (par exemple la définition de directories).
- des commandes Bourne avec de nouvelles fonctionnalités (test, expr, echo).
- des sélections par menu possible.
- des messages d'erreur plus significatifs.
- des alias peuvent être créés.

Les différences entre le Korn Shell et le Bash sont suffisamment faibles pour envisager des scripts communs.

Bourne Again SHell (bash)

Le bash est aujourd'hui le plus populaire, même s'il n'est pas le plus puissant. Il est la référence de base des distributions open source de Linux. Le bash améliore le Bourne Shell.

L'une des différences importantes est qu'il est une implémentation GNU, donc libre de droits de bsh, qui est propriété des laboratoires Bell.

Il est porté sur Windows (Cywin) et est également le shell standard sous MacOSx.

L'équivalence sur d'autres systèmes

Il existe sur Windows un équivalent très "light" du shell : le langage DOS qui permet d'écrire des scripts .BAT. La puissance du DOS est nettement plus faible que la puissance du shell. On ne peut plus parler réellement de langage de programmation, bien qu'il contienne certaines structures (conditionnelles, notamment).

Le DOS est plutôt fait pour exécuter des scripts linéaires (séquences de commandes) plutôt qu'une stratégie complexe de manœuvre du système.

Quelques définitions finales

Un "script"	une séquence d'instructions interprétables , en général avec un point d'entrée et une fin , qui automatise le déroulement de commandes.
Le "shell"	un programme (exécutable système) qui exécute des commandes écrites en langage de script (shell script).
Le "shell script"	un langage de programmation système interprétable.
Un "script shell"	un source écrit en shell script <=> un fichier.

Clefs Wikipedia collectées :

<http://fr.wikipedia.org/wiki/Stephen+Bourne>