

Retour sur le shell

Patrick MARCEL, Université d'Orléans

L2 Outils du développeur — S3

Philosophie UNIX

Keep
It
Simple
Stupid

Briques de base :

les **processus** et les **fichiers**

Système de fichiers

Le système de fichiers est présenté sous la forme d'une **arborescence unique**.

À chaque processus est associé un **utilisateur**, ainsi qu'un **répertoire courant**.

Nommage des fichiers

Un **chemin** est une chaîne constituée de noms de fichiers séparés par des caractères `/`.

`/usr/bin/sh`

chemin absolu

`atelier12/slides/1-shell.tex`

chemin relatif

Les éléments intermédiaires sont des **répertoires** ou des **liens symboliques vers des répertoires**.

`.`

répertoire courant

`..`

répertoire parent

Types de fichiers

Fichier régulier. Séquence d'octets quelconque, adressable, de longueur variable

```
\documentclass[smaller]{beamer}\n\usepackage[franc...
```

0

F

1F

2F

Répertoire. Associe des noms à des fichiers

(définit toujours . et ..)

```
. →  
.. →  
hello.c →  
slides →  
hello.o →
```

Types de fichiers (suite)

Lien symbolique. Alias défini par un chemin
(*fichier contenant un chemin*)

Tube nommé. mécanisme de communication entre processus
(*aucune donnée associée*)



Fichier spécial. mécanisme d'E/S vers les périphériques
(*aucune donnée associée*)
(voir /dev)

Données et méta-données

Selon son type, un fichier contient des **données**.

À tout fichier sont associées des **méta-données** :

- type de fichier ;
- propriétaire ;
- droits d'accès ;
- taille (*si ça a un sens*) ;
- dates de création/modification/accès ;
- ...

consulter ces informations : `ls` (et notamment `ls -l`)

Processus

Programme Ensemble de fichiers *texte* décrivant dans un langage donné un ensemble d'opérations à effectuer.

Binaire exécutable Fichier *binaire* contenant le code machine correspondant au programme.

Processus Une instance d'un exécutable en cours d'exécution dans le système.

Shell

Le **shell** `sh` est un interprète de commandes.

```
# date  
lundi 1 septembre 2014, 10:26:38 (UTC+0200)  
#
```

Il exécute les commandes qu'il lit soit depuis l'entrée standard soit depuis un fichier.

Une commande simple

```
$ tr "[a-z]" "[A-Z]"  
Bonjour  
BONJOUR  
echo  
ECHO
```

L'utilisateur interagit avec le processus à travers :

- les **arguments**, séparés par des blancs ;
- les **entrées** ;
- les **sorties**.

Expansion

Avant de lancer le programme, le shell modifie les arguments en procédant à une **expansion** :

- il remplace les variables par leur valeur ;
- il exécute les commandes entre accent grave (`) ;
- il remplace `~` par la valeur de la variable **HOME** ;
- il remplace les **patterns** par les noms de fichiers correspondants.

Patterns

- `?` désigne un unique caractère ;
- `*` désigne une suite (potentiellement vide) de caractères
- `[` et `]` servent à désigner un caractère parmi un ensemble (ex : `[aei]`, `[a-z]`, `[!a]`).

Protection

- les guillemets doubles " servent à grouper un ensemble de mots et empêchent l'expansion des patterns ;
- les guillemets simples ' servent à complètement empêcher l'expansion ;
- les accents graves remplacent le contenu par la sortie standard de son exécution.

Il est également possible de protéger un symbole en le précédant d'un backslash (\).

Environnement

```
$ date  
lundi 1 septembre 2014, 10:26:38 (UTC+0200)
```

Pourquoi la commande répond-elle en français ?

Il existe d'autres sources d'information pour un processus :

- Les fichiers de configurations ;
- Les variables d'environnement.

Variables d'environnement

Les **variables d'environnement** sont des variables contenant des valeurs qui sont passées aux programmes lors de leur démarrage.

Usuellement, ces variables sont écrites intégralement en majuscule.

Quelques variables courantes

- **PATH** : répertoires où chercher les programmes ;
- **HOME** : emplacement du *répertoire maison* ;
- **PWD** : répertoire courant ;
- **LANG** : **locale** utilisée ;
- **USER** : nom de l'utilisateur ;
- **PROMPT** : format du prompt ;
- **EDITOR** : éditeur de texte par défaut ;
- **PAGER** : défileur de texte par défaut ;
- ...

VARIABLES D'ENVIRONNEMENT EN SHELL

Accès : Il suffit de préfixer la variable par le caractère `$`.

```
$ echo $HOME  
/Users/nopid
```

En cas de doute, ajouter des accolades...

Modification : Avec une affectation (=) et en utilisant la commande `export`.

```
$ export LANG=C  
$ TZ=Asia/Tokyo date  
Mon Sep 1 17:33:26 JST 2014  
$ date  
Mon Sep 1 10:33:26 CEST 2014  
$ export PATH=$PATH:/sbin
```

Utilisez **efficacement** votre shell