

Introduction aux systèmes d'exploitation (IS1)

TP 9 – Scripts

Semaine du 23 octobre 2006

Attention : la commande `script` vue au TP5 n'a rien à voir avec la notion de script `bash` que nous allons étudier dans ce TP.

Exercice 1 – Quelques scripts utiles

1. créez un répertoire `bin` dans votre répertoire maison. Modifiez-en les droits d'accès de manière à ce que seul vous puissiez en lire le contenu (le groupe et les autres ayant tout de même la possibilité d'accéder aux fichiers s'y trouvant).

Ce répertoire contiendra entre autres tous les scripts de cet exercice.

2. De manière à pouvoir lancer les scripts depuis n'importe quel répertoire courant, il faut modifier la variable d'environnement `PATH`, pour qu'elle explore votre répertoire `bin` en dernier. Quelle commande devez-vous taper ?

Devoir taper la commande précédente dans chaque nouvelle fenêtre n'est pas très pratique. Le fichier `.bashrc` de votre répertoire maison est un script spécial qui est exécuté à chaque fois que vous lancez un `SHELL`.

3. Mettez donc la ligne de commande précédente dans votre fichier `.bashrc`. Faites en sorte de ne *pas* taper la valeur complète de `PATH` en réutilisant son ancienne valeur.

Cela rendra votre code plus "portable" entre les différentes architectures.

Nous allons écrire un script qui permet de "faire le ménage". En effet, de nombreux programmes génèrent des fichiers temporaires, des fichiers de sauvegardes, etc ... Par exemple :

- si vous ouvrez un fichier `fic` (déjà existant) avec `emacs`, ce dernier commence par le recopier en un fichier `fic~` qui est une copie de sauvegarde. Après avoir modifié votre fichier et fermé `emacs`, vous n'avez en principe plus besoin de ces vieilles versions de fichiers. De même `emacs` génère parfois (lors de plantages) des fichiers `#fic#`.
 - `ltx` (qui est un outil recommandé pour faire votre rapport), génère des fichiers ayant l'extension `.log`.
4. Quelle ligne de commande pouvez-vous taper pour supprimer tous les types de fichiers précémmment cités se trouvant *dans le répertoire courant* ? On donnera la préférence à une ligne de commande courte, utilisant un maximum d'expansions de `bash`.
 5. Écrivez un script `nettoie` qui, en utilise une boucle `for` et les tests appropriés, descend récursivement dans les répertoires et supprime tous les fichiers en question.

6. Votre script serait-il aussi efficace s'il n'était pas à un endroit référencé dans PATH ? Pouvez-vous descendre dans les liens symboliques (par ex. dans rep1bis, obtenu par `ln -s rep1 rep1bis`) ? Est-ce une bonne idée ?
7. Que se passe-t-il lorsque vous rencontrez un répertoire dont vous n'avez pas les droits en exécution ? S'il y a un problème, corrigez-le.

Nous allons maintenant rendre le script plus "souple" en lui rajoutant des options.

8. on souhaite rajouter pour commencer un argument qui indique quels sont les types de fichier à effacer. Par exemple `nettoiePlus em cs` supprime uniquement les fichiers du type `fic~` et `#fic#`. `log` indiquera tous les fichiers du type `fic.log`, tout indiquera tous les fichiers précédemment cités.
9. ensuite, on souhaite rajouter un deuxième argument optionnel (c'est à dire qu'on est pas obligés de l'indiquer) qui prend les valeurs `-i` ou `-f`. L'option `-i` indique que l'on veut confirmer à la main la suppression de chaque fichier, l'option `-f` indique le contraire : dans tous les cas, on veut supprimer le fichier en question.
10. enfin, dans le cas où l'utilisateur n'aurait pas donné les bons arguments à votre script, affichez un message d'erreur qui indique comment utiliser ce script.