TP 3 : Rudiments de programmation en shell

Février 2014

Objectifs :
- utiliser les outils de base d'Unix et leur composition ;
- découvrir la programmation shell.

Ecriture d'un outil de test

On se propose d'écrire un outil permettant de vérifier de façon automatique le bon fonctionnement d'un programme.
On supposera que chaque programme à tester lit des données sur son entrée standard et affiche ses résultats sur sa sortie standard.
On obtiendra alors un comportement tel que le suivant (partiel) :

```
autotest -v -p tri-selection
Test avec entree ./auto_test/001.input et sortie /tmp/001-at1413 compare a ./auto_test/001.output ...OK
      ...
Test avec entree ./auto_test/011.input et sortie /tmp/011-at1413 compare a ./auto_test/011.output ...OK
Test avec entree ./auto_test/012.input et sortie /tmp/012-at1413 compare a ./auto_test/012.output ...ERREUR
      ...
Test avec entree ./auto_test/999.input et sortie /tmp/999-at1413 compare a ./auto_test/999.output ...OK

3 echecs sur 999 effectues
Les fichiers d'echec sont dans /tmp
```

1. Écrire une première version simple de ce programme de test qui prenne en paramètre un programme à tester, un fichier d'entrée et un fichier de sortie et qui affiche si le test est réussi ou non.
2. Enrichir le programme pour qu'il soit capable d'itérer le test sur un ensemble de fichiers (ces fichiers seront par exemple situés dans un répertoire et nommés *fichier*`.input` et *fichier*`.ouput`.
3. Gérer des options de sorte à obtenir un outil avec un comportement par défaut mais paramétrable, par exemple :

```
./autotest <option> ...          Automatisation  du test d'un pgm
Options
   -h          cette aide
   -v          verbeux
   -k          conserve les fichiers de sortie (seuls les fichiers
               d'erreur sont conserver par defaut)
   -p <pgm>    test <pgm> (./a.out par defaut)
   -d <dir>    repertoire a tester (./auto_test par defaut)
   -n          ne pas faire reellement les tests
```