

**Université de Nice Sophia-Antipolis**  
**Licence de *Mathématiques Appliquées et Sciences Sociales*, deuxième année**

**Enseignement de *Linux, vecteur de culture informatique***

Pierre Crescenzo (Pierre.Crescenzo@unice.fr),  
Jean-Pierre Lips (Jean-Pierre.Lips@unice.fr) et  
Isabelle Mirbel (Isabelle.Mirbel@unice.fr)

**Travaux Dirigés 9 de la semaine du 30 novembre 2009 : Scripts Shell**

## 1. Introduction

- a. Connectez-vous. Créez votre répertoire de séance `~/Linux-TD/TD09`. Rendez-vous dans ce répertoire.
- b. Durant tout ce TD, appliquez-vous à utiliser la complétion, l'historique des commandes et les couper/copier/coller au clavier et à la souris.

## 2. Premiers scripts Shell

Ces exercices sont les mêmes que ceux terminant le TD 8. Reprenez là où vous en étiez sans sauter de question.

- a. À l'aide de l'éditeur `emacs` puis de la commande `chmod`, créez un script Shell appelé `affiche-params1`. Celui-ci affiche sur une unique ligne le nom du script puis le premier paramètre positionnel.
- b. Testez ce script `affiche-params1` comme suit et expliquez les résultats obtenus par les commandes suivantes :
  1. `affiche-params1`
  2. `affiche-params1 +23`
  3. `affiche-params1 toto`
  4. `affiche-params1 ~crescenzo`
  5. `affiche-params1 $DISPLAY`
  6. `affiche-params1 '$DISPLAY'`
  7. `affiche-params1 --help +23`
  8. `affiche-params1 --help+23`
  9. `affiche-params1 bonjour tout le monde`
  10. `affiche-params1 'bonjour tout le monde'`
- c. Écrivez un script `affiche-params2` qui affiche sur une ligne son nom, le nombre de paramètres positionnels et les deux premiers de ses paramètres puis, sur une deuxième ligne, la totalité des paramètres positionnels. Testez ce script avec divers appels, en vous inspirant des tests de `affiche-params1`.
- d. Écrivez un script `affiche-params3` qui affiche pour chaque paramètre positionnel, sur une ligne : le caractère '`<`' puis la valeur du paramètre puis '`>`'.
- e. Dans un terminal, et non dans un script Shell, en utilisant une boucle `for`, affichez chaque nom de fichier du répertoire `/bin` sur une ligne entre les caractères '`<`' et '`>`'.
- f. En utilisant une boucle `for` et une expression arithmétique, écrivez un script `affiche-params4` qui affiche pour chaque paramètre positionnel, son numéro d'indice (1, 2, 3, etc.) et sa valeur.

### 3. Une simple corbeille

Dans cet exercice, vous allez programmer quelques scripts Shell qui vous permettront de disposer d'une corbeille simpliste en ligne de commande. Pour qu'ils puissent s'exécuter depuis n'importe quel répertoire, placez vos scripts dans votre répertoire `~/bin`.

- a. Créez un répertoire `~/MaCorbeille`. C'est dans ce répertoire que seront placés, au moins temporairement, les fichiers effacés, pour pouvoir les récupérer (manuellement) en cas d'effacement erroné.
- b. Créez et exportez une variable `CORBEILLE` qui contient le nom du répertoire de votre corbeille.
- c. Programmez puis testez un script `effacer` qui prend un ou plusieurs noms de fichier en paramètre et :
  1. Vérifie l'existence de la variable `CORBEILLE` et, sinon, affiche un message d'erreur adéquat et termine le script sur un code d'erreur.
  2. Vérifie l'existence du répertoire donné par la variable `CORBEILLE` et, sinon, le crée.
  3. Vérifie que vous avez, en tant que propriétaire, tous les droits sur le répertoire donné par la variable `CORBEILLE` et, sinon, ajoute ces droits.
  4. Vérifie l'existence des fichiers dont le nom est passé en paramètre : pour ceux qui existent, les efface (c'est-à-dire les déplace dans le répertoire donné par la variable `CORBEILLE`) ; pour ceux qui n'existent pas, le signale par un message significatif.
- d. Programmez puis testez un script `vider` qui ne prend pas de paramètre et vide le répertoire donné par la variable `CORBEILLE` en faisant, comme le script `effacer`, les tests préliminaires utiles.