Travaux Dirigés n°5

Cours de Langage C

—Licence Informatique—

Ce TD porte sur l'utilisation de la ligne de commande d'un programme.

▸ **Exercice 1.** (2 en 1) *Que dire des deux exécutables* `compress` *et* `uncompress` *en voyant le résultat de la commande suivante :*

```
$ ls -i /usr/bin/compress /usr/bin/uncompress
  7557 /usr/bin/compress      7557 /usr/bin/uncompress
```

▸ **Exercice 2.** (ligne de commande) *La structure contenant la ligne de commande est initialisée par le système et manipulable grâce à deux paramètres de la fonction* `main`*. Le premier paramètre (souvent appelé* `argc`*) est le nombre d'arguments de la commande plus un, le deuxième argument (usuellement nommé* `argv`*) est un tableau de chaînes de caractères (correspondant à la ligne de commande) terminé par* `NULL`*.*

1. *Ecrire un programme affichant la ligne de commande par laquelle il a été lancé.*
2. *Ecrire une fonction qui effectue la copie exacte de la structure qui contient les arguments de la ligne de commande.*
3. *Ecrire un programme libérant la copie des arguments de la ligne de commande.*

▸ **Exercice 3.** (variables d'environnement) *Il est aussi possible de récupérer, au niveau d'un programme C, la valeur des variables d'environnement du shell appelant. Ceci se fait au moyen d'un troisième paramètre du* `main` *(usuellement appelé* `env`*) du même type que* `argv`*. Ecrire un programme affichant la valeur des variables passées en argument de la ligne de commande du programme.*

▸ **Exercice 4.** (tri à bulles) *Ecrire un programme (contenant uniquement la fonction* `main`*) récursif réalisant un tri à bulles du nom par lequel il a été appelé. A chaque étape on réordonnera (le cas échéant) deux lettres consécutives puis on fera un appel récursif pour continuer le travail. Le programme s'arrêtera dès lors que son nom sera ordonné (par ordre croissant des codes ASCII). Il affichera alors le nombre d'étapes du tri et le nom trié. Par exemple, notre programme* `tri` *donnera :*

```
$ tri
irt obtenu par 3 swaps
$ mv tri trilleur_a_bulles
$ trilleur_a_bulles
__abeeillllrrstuu obtenu par 70 swaps
```