

Atelier n°1 : (Rappel de C) Récupération des arguments de la ligne de commande.

Utilité.

Dans le cadre de nos différentes applications, il sera important d'être capable de récupérer les « arguments » passés dans la ligne de commande (la ligne qui est entrée par l'intermédiaire de l'interpréteur de commande, généralement appelé «*shell*»). Par exemple, lorsqu'il est invoqué, un interpréteur python est capable de récupérer le nom du script à lancer, ainsi que diverses options. Par exemple :

```
$ python mon_script.py
```

Il faut bien que, d'une manière ou d'une autre, le programme interpréteur python (l'exécutable), puisse récupérer le nom du fichier à interpréter (ici `mon_script.py`), qui constitue ici « l'argument ».

Le langage C offre un mécanisme de base qui permet de récupérer ces informations. En l'occurrence, c'est la « signature » de la fonction principale (fonction `main`) qui permet de remplir cet objectif : au lieu de se contenter d'une signature « classique », indiquant que la fonction `main` ne récupère pas de paramètres, à savoir :

```
int main(void)
{
...
}
```

On va plutôt préciser ceci :

```
int main( int argc, char *argv[] )
{
...
}
```

Supposons que nous ayons écrit un programme nommé `programme1` faisant appel à ce type de signature et que nous l'invoquions dans la ligne de commande suivante :

```
$ programme1 UN DEUX TROIS 4
```

Celle-ci comporte un total de 5 chaînes de caractères, qui sont «`programme1`», «`UN`», «`DEUX`», «`TROIS`» et «`4`» (attention, il s'agit bien ici d'une chaîne et non d'un simple caractère). Lors de la validation de la ligne par la touche **return**, le *shell* va donc en analyser le contenu et positionner la variable `argc` à 5, qui représente donc le nombre de chaînes total dans la ligne de commande. Ces chaînes elles-mêmes seront copiées dans `argv`. La première chaîne, à savoir «`programme1`», sera stockée dans `argv[0]`, la seconde («`UN`») dans `argv[1]` et ainsi de suite. La chaîne «`4`» sera ainsi rangée dans `argv[argc-1]`.

A faire.

Pour commencer, il va s'agir d'écrire un petit programme simple se contentant d'afficher à l'écran les chaînes de caractères qui lui ont été passées comme argument. Le code correspondant pourrait commencer de la manière suivante :

```
#include <stdio.h>
```

```

#include <stdlib.h>
#include <ctype.h>

int main( int argc, char *argv[] )
{
    int i ;    /* ->compteur de boucle */
    /* dans la boucle ci-dessous on affiche les chaines */
    /* passees en argument */
    /* A FINIR */
    return( 0 ) ;
}

```

Dans le cadre de l'exemple cité avec :

```
$ programme1 UN DEUX TROIS 4
```

On souhaite que l'affichage produit soit :

```

Nombre de chaines dans la ligne de commande = 5
argv[0] = programme1
argv[1] = UN
argv[2] = DEUX
argv[3] = TROIS
argv[4] = 4

```