

Consolidation : accès à l'environnement à partir d'un programme C

Les programmes exécutables applicatifs, programmés en C ou en tout autre langage savent dialoguer avec le système d'exploitation. En particulier, ils savent accéder à ces variables d'environnement. Parler des variables d'environnement est un "abus de langage" : il faudrait parler des variables de "leur" environnement.

Dans cet article, nous consolidons notre compréhension de cet environnement d'exécution d'un programme en proposant une petite démonstration sur l'usage que peut en faire un programme écrit en langage C.

Nous démontrerons comment les commandes exécutées dans l'univers du shell ont indirectement une influence sur le déroulement d'un tel programme.

Le programme exemple `myvar.c`

Voici un programme TRES simple qui lit une variable dans l'environnement d'exécution et réagit en conséquence. On remarque qu'aucune librairie exotique n'est nécessaire pour accéder à la fonction de lecture. Il s'agit donc d'une fonction TRES standard :

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

void main(void){
    char *uneVarEnv = getenv("EDITEUR");
    printf("La variable d'environnement d'éditeur vaut %s\n", uneVarEnv);
}
```

Ce programme peut être enregistré dans un fichier `monenv.c`, puis compilé en utilisant le compilateur en ligne de commande :

```
$>gcc monenv.c -o monenvcmd
```

Il est maintenant possible de tester notre programme en le lançant à partir d'un shell de commandes classique (une console) :

```
$>./monenvcmd
La variable d'environnement d'éditeur vaut (null)
```

puisque nous n'avons pas défini cette variable dans l'environnement. Définissons-la donc.

```
$>EDITEUR='eclipse'
$>./monenvcmd
La variable d'environnement d'éditeur vaut (null)
```

La réponse est toujours la même. Que se passe-t-il ? La variable que nous avons mise en place est bien effectivement connue du shell courant. Pour preuve une lecture subséquente :

```
$>echo EDITEUR  
eclipse
```

Mais elle n'a pas été exportée. Elle est donc connue que du shell, mais pas des autres processus qui suivent. L'export de la variable corrige cette situation :

```
$>export EDITEUR  
$>./monenvcmd  
La variable d'environnement d'éditeur vaut eclipse
```