

Principe des systèmes d'exploitation

TD1

1 Exercice 1 : argv et getenv()

Programmez les exemples du cours et testez les :

- echoargv.c (diapositive 21, coté gauche, Unix seulement)
- echoargenviron.c (diapositive 23)
- exemple_environ_2.c (diapositive 24)

On rappelle que pour compiler un programme C simple en ligne de commande, il suffit de lancer (nous détaillerons les options du compilateur gcc dans un prochain exercice) :

```
$ gcc -std=gnu11 -Wall -o echoargv echoargv.c
```

Pour le premier exemple on doit fournir un script bash test_echoargv.sh qui lance au moins les exécutions suivantes :

- echoargv
- echoargv PATH HOME PWD
- echoargv TOTO
- export TOTO="mon test de TOTO"
- echoargv TOTO

2 Exercice 2 : Makefile

En vous inspirant du fichier Makefile de la diapositive 26, écrivez le Makefile qui permet de compiler les trois exemples précédents (fichier à inclure dans le même répertoire que les sources).

Dans ce répertoire, testez les commandes :

- make
- touch *
- make clean
- make all

3 Exercice 3 : appels systèmes

Exécuter la commande `strace` sur l'exécutable `echoargenv`. Que fait la commande `strace` ? Quel est le premier appel système ? Que deviennent les `"printf"` de votre code ?

4 Exercice 4 : `id`

Dans une console (terminal de ligne de commandes), testez la commande `bash id` : quel est le résultat ?

Comparez ce résultat avec le contenu du fichier `/etc/passwd`.

Écrivez le programme C `myid.c` qui imite cette commande en utilisant les appels systèmes `getuid()` et `getgid()` (voir le manuel avec la commande `man`). Pour le nom de l'utilisateur, utiliser la famille de `getpwnam()` (plus précisément `getpwuid()`).