

Programmation Système

TP n° 2

1^{er} et 2 février 2016

L'utilisation de la fonction `system()` n'est pas autorisée dans ce TP.

Exercice 1 : premiers pas avec `exec`

1. Quel est l'effet de la ligne de commande « `exec ls` » ?
2. écrire un programme qui exécute la commande « `ls` ». Votre programme devra seulement exécuter `/bin/ls` — il n'est pas nécessaire de créer un nouveau processus.
3. Modifier votre programme pour qu'il exécute la commande « `ls -l` ».
4. Modifier votre programme pour qu'il exécute la commande dans un processus fils (sans oublier d'attendre la fin de l'exécution du fils).

Exercice 2 : `exec` avec des vecteurs

1. écrire un programme « `please` » qui exécute la commande passée en paramètre. Par exemple, « `please ls -l` » doit exécuter la commande « `ls -l` ».
2. Modifier votre programme pour que, une fois la commande exécutée, il affiche comment elle s'est terminée — soit `terminaison normale avec resultat n`, soit `commande tuee par le signal s`, soit `bizarre`.
3. écrire un programme « `please5` » qui exécute la commande passée en paramètre cinq fois d'affilée — « `please5 ls -l` » doit exécuter cinq fois la commande « `ls -l` ».

Exercice 3 : exécution conditionnée

écrire un programme « `si` » qui se comporte (presque) comme la commande « `if` » du `shell`. Votre programme prendra deux ou trois noms de commande en paramètres : un test, une conséquence et optionnellement une alternative. Il commencera par exécuter le test ; si celui-ci retourne 0 (succès), il exécutera la conséquence ; sinon, si une alternative est proposée, il exécutera cette dernière.

Votre programme ne devra créer qu'un seul processus fils.

Par exemple, la commande « `./si true emacs` » doit exécuter « `emacs` », tandis que « `./si false emacs vi` » doit exécuter « `vi` » (on rappelle que les commandes « `true` » et « `false` » ne font rien et retournent 0 et 1 respectivement).

Exercice 4 : un peu de rangement

écrire un programme qui simule la commande :

```
« find . -size +100000 -exec gzip {} ; ».
```