

## Programmation Système & Réseau TD n° 3

### Notions abordées :

- Exécution de commandes (primitives de la famille exec)
- Duplication de descripteurs de fichier (dup et dup2)
- Tubes nommés (mkfifo)

### Exercice 1

Écrire un programme C qui lance la commande `ls` sur un répertoire passé en argument.

### Exercice 2

Écrire un programme C qui exécute une commande Unix passée en argument.  
Exemple :

```
$ ./a.out ls -al
total 320
drwxr-xr-x 2 violard nfs 4096 2011-10-06 15:14 .
drwxr-xr-x 4 violard nfs 4096 2011-10-06 15:04 ..
-rwxr-xr-x 1 violard nfs 7252 2011-10-06 15:14 a.out
...
$
```

### Exercice 3

Écrire un programme C qui exécute au plus 3 fois (dans des processus séparés) une commande Unix passée en argument. Si l'une des exécutions provoque une erreur, il ne faut pas réaliser les exécutions suivantes.

## Exercice 4

Écrire un programme C équivalent à la commande shell `ps aux | grep root | wc -l`.

## Exercice 5

Écrire 2 programmes C distincts : le premier lit des données sur l'entrée standard et les passe par un tube nommé au deuxième qui les affiche sur sa sortie standard.

## Exercice 6

Écrire 2 programmes C : un client et un serveur qui communiquent par des tubes nommés. Le serveur crée 2 tubes nommés `question` et `reponse` ; chaque fois qu'il reçoit un caractère sur le tube `question`, il applique une fonction `traduction` sur ce caractère et renvoie le résultat (un autre caractère) sur le tube `reponse`. Le client prend un mot en argument, envoie toutes les lettres du mot au serveur et affiche les lettres rendues par le serveur.