UNIVERSITÉ DE STRASBOURG

Département d'Informatique
LICENCE INFORMATIQUE 2ème année

Programmation Système
TD n° 2

Notions abordées :

- **Processus** (`fork()`, `getpid()`, `getppid()`, `exit()`, `wait()`)
- **Tubes** (`pipe()`)

Exercice 1

Écrire un programme générant un processus fils :
- le processus fils affiche son numéro (*pid*) ainsi que le numéro du père, puis termine avec un code de retour égal au dernier chiffre du *pid*.
- le processus père, quant à lui, affiche le *pid* du fils, puis attend sa terminaison et affiche son code de retour.

Exercice 2

Qu'affiche chacun des programmes suivants ? Quel est le nombre de processus créés dans chaque cas ?

a)
```
for (i=1; i<=4; i++)
      {
        pid = fork();
        if (pid != 0) printf("%d\n", i);
      }
```

b)
```
for (i=1; i<=4; i++)
      {
        pid = fork();
        if (pid == 0) break;
        else printf("%d\n", i);
      }
```

Exercice 3

Écrire un programme qui lance n processus fils, puis attend leur terminaison. À chaque fois qu'un processus se termine, le père affiche son pid et son numéro d'ordre.

Exercice 4

Écrire un programme composé de deux processus : le premier lit des données sur l'entrée standard et les passe par un *tube* au deuxième qui les affiche sur sa sortie standard.

Exercice 5

Généraliser la question précédente à n processus : le premier passe les données depuis l'entrée standard au second, qui les passe au troisième, et ainsi de suite jusqu'au $(n-1)$-ième qui les passe au n-ième, qui les écrit sur sa sortie standard.