Cours THL & Révisions C

Introduction

Rappels sur le Shell et Appels Système

Fork: Le fils a le même groupe de processus que son père.

```
pid_t p = fork();

if (p == 0)
    // fils
else
    // père
```

Pipe: Passer la sortie std d'un processus à celui d'un autre processus.

Autre:

- **getpgrp( )**: récupérer son process group.
- **tcgetpgrp(*STDOUT_FILENO*)**: récupérer le process group du terminal.
- **setpgid(*cpid* , *cpid*)**: permet de set un process group
- **tcsetpgrp(*STDOUT_FILENO* , *cpid*)**: set le process group du terminal à un autre (ici, celui du fils: défini juste au-dessus).
  Info: Quand j'associe à mon terminal un autre process group, ça va mettre ce dernier en foreground et mettre l'ancien en background.
- **waitpid()**: attendre que le fils finisse.

Typiquement, en liant le terminal au group process du fils, ce dernier étant déjà lié à son père, si un SIG touche le groupe en foreground, le père est averti que son/ses fils ont été tués mais il est toujours en vie!

Exemple: Application des fonctions listées au-dessus.

```c
int status = 0;
int err = setpgid(cpid, cpid);

if (err < 0)
    perror("setpdgid fail");

err = tcsetpgrp(STDOUT_FILENO, cpid)

if (err < 0)
    perror("tcset out");

pid_t w = waitpid(-cpid, &status, 0);

if (WIFEXITED(status))
    perror("exit");
if (WIFSIGNALED(status))
    perror("signal");
```

SigAction (`man sigaction` better)

Appel système reçoit le nom du signal, une structure contenant les nouvelles informations ainsi qu'un pointeur sur une structure vide dans laquelle sera entrée l'ancienne version.