TD n°6

Signaux : envoi et capture

Exercice 1 [Envoi d'un signal] On reprend l'exercice coucou-hibou de la première feuille. On rappelle que coucou (le pere) affiche coucou toutes les secondes jusqu'à un nombre de secondes donné en paramètre, tandis que hibou (le fils) affiche hibou toutes les 1 à 5 secondes, indéfiniment. On souhaite à présent interrompre hibou dès que coucou s'arrête.

Exercice 2 [Reception d'un signal] On souahite écrire un programme se déroulant de la façon suivante. Le père crée un processus fils. Chacun des deux processus installent une fonction de capture pour un signal `SIGUSR1` (le père) ou `SIGUSR2` (le fils). Le fils doit ensuite entrer dans une boucle infinie, puis s'il reçoit le signal `SIGUSR2` il affiche "signal !" puis envoie au père le signal `SIGUSR1` et se remet dans sa boucle infinie. Le père doit attendre quelques secondes, puis envoyer le signal `SIGUSR2` au fils, puis se mettre à afficher tous les entiers en partant de 0. Dès qu'il reçoit le signal `SIGUSR1`, il affiche "fin !" puis envoie au fils le signal `SIGINT` et se termine.

Exercice 3 [Communication inter-processus et signaux] On suppose qu'on dispose de deux fonctions de tri sur des tableaux d'entiers, appelées `tri1` et `tri2`. Ces fonctions ont le prototype `void triX(int *tab);`, et elles travaillent directement sur le tableau en mémoire. On veut écrire un programme qui teste ces fonctions afin de savoir laquelle est la plus rapide, en les faisant faire le calcul en même temps. Plusieurs processus devront donc s'exécuter en parallèle. Lorsque l'une des méthodes aura términé le tri, elle devra prévenir tout de suite l'arbitre puis lui communiquer le tableau trié. L'autre concurrent sera notifié de son échec et devra envoyer à l'arbitre le tableau en cours de tri dans son état actuel.

1. Proposer une architecture de programme utilisant des tubes et les signaux.
2. Écrire le code des concurrents.
3. Écrire le code de l'arbitre.