

TP 2 : Processus et signaux

Objectifs :

- Comprendre et manipuler les images de processus avec la commande `exec`.
- Comprendre et manipuler la réplication des processus avec la commande `fork`.
- Utiliser les signaux comme mécanisme de communication entre processus.

Exercice 1 : Image de processus

Question 1 : Ouvrez deux terminaux (T_1 et T_2). Affichez dans T_2 la liste des processus en exécution et repérez le pid de T_1 . Exécutez dans T_1 la commande : `exec sleep 15`. Affichez de nouveau dans T_2 la liste des processus en exécution. Que constatez-vous ? Attendez la fin des 15 secondes. Que se passe-t-il ? Déduisez-en la fonctionnalité de la commande `exec`.

En C, il existe une famille de fonctions qui correspond à la commande `exec`. Chaque fonction permet de remplacer le code exécutable du processus courant par celui qu'elle est chargée d'exécuter. Pour les quatre fonctions `execl()`, `execlp()`, `execv()` et `execvp()` :

- le premier argument de ces fonctions est le chemin absolu de l'exécutable. Pour les fonctions se terminant par un `p`, si seul le nom de la commande est donné, elle est alors recherchée en utilisant la variable d'environnement `PATH` ;
- le deuxième argument des fonctions `execl()` et `execlp()` doit être la valeur de `argv[0]` ; c'est-à-dire le nom du programme lui-même. Les arguments suivants sont les pointeurs sur des chaînes de caractères qui correspondent chacune à un paramètre que l'on passe à la commande à exécuter. Il faut terminer par le pointeur `NULL` pour clore la liste des paramètres ;
- les fonctions `execv()` et `execvp()` n'acceptent que deux arguments, le deuxième étant un tableau de chaîne de caractères. Chaque élément du tableau correspond aux paramètres de la ligne de commande.

Question 2 : Ecrivez un programme en C qui utilise l'une des fonctions précédentes et qui exécute la commande `du -s -h`. (La commande `du` se trouve dans le répertoire `/usr/bin`.)

Question 3 : Ecrivez un programme en C qui utilise une autre des fonctions précédentes et qui lance un programme exécutable que vous avez précédemment réalisé.

Exercice 2 : « Je suis ton père... »

Testez le programme « Je suis ton père... » donné en TD.

Exercice 3 : Partie de ping pong

Vous devez écrire un programme qui permette à deux processus de faire une partie de ping pong. L'échange de la balle entre les deux processus doit être réalisé avec une alarme.

Exercice 4 : Horloge

Ecrire un programme qui utilise trois processus *Heure*, *Minute* et *Seconde* qui incrémentent les trois aiguilles d'une horloge. *Seconde* reçoit un signal `SIGALRM` chaque seconde et émet un signal à *Minute* quand son compteur passe de 59 à 0. Quand *Minute* reçoit un signal, il incrémente son compteur. Quand le compteur de *Minute* passe de 59 à 0, il envoie un signal à *Heure*. Les arguments passés au programme principal correspondent aux valeurs d'initialisation des compteurs.