Programmation Système
TP no 2 : Signaux

Guillaume Mercier
email : mercier@enseirb.fr

22 Octobre 2018

1 Masquage de signaux

Un processus ne prend pas forcément en compte les signaux dans l'ordre où ils ont été reçus. À l'aide des masques, faites en sorte de forcer cet ordre. Effectuez des tests avec `SIGUSR1`, `SIGUSR2` et `SIGSEGV` et vérifiez que votre programme fonctionne correctement : par exemple vous allez envoyer `SIGSEGV` en premier, mais vous voulez qu'il soit pris en compte en dernier, après `SIGUSR1` et `SIGUSR2`.

2 Signaux : communications en Morse

Cet exercice a pour but de réaliser une communication utilisant le code *morse* (ou un quelconque code similaire) entre deux processus — un client et un serveur — au moyen de signaux. Un résumé du codage morse est disponible dans le fichier `morse.txt` du répertoire `/net/ens/mercier/TP/Prog_Sys`. Pour information, il faut y ajouter les particularités suivantes :

— les signes morses sont séparés par un blanc ;
— les lettres sont séparées par trois blancs ;
— les mots sont séparés par sept blancs.

Questions

— écrivez un programme `tstsig_1.c` qui boucle infiniment et affiche :
  — `'.'` lorsqu'il reçoit un signal `SIGUSR1` ;
  — `'-'` lorsqu'il reçoit un signal `SIGUSR2` ;
  — `' '` lorsqu'il reçoit un signal `SIGALRM`.
  Vous utiliserez la fonction `sigaction(3)` pour gérer la mise en place des gestionnaires (*handler*) de signaux. Vous pourrez utiliser la commande `kill(1)` pour envoyer des signaux au processus depuis un terminal.

— écrivez un programme `tstsig_2.c` à partir de `tstsig_1.c` qui — en plus des fonctionnalités de `tstsig_1.c` — se termine en affichant le message `"[over]"` lorsqu'il reçoit un signal `SIGTERM`.

— écrivez un programme `serveur.c` et un programme `client.c` avec les fonctionnalités suivantes :

serveur Le serveur itère en demandant un message (directement en morse, ou en format texte si vous avez le temps de programmer la conversion en fin de TD) et un numéro de `pid`, puis envoie le message au processus client correspondant en utilisant les 4 signaux ci-dessus avec les mêmes conventions de signification.

client Le client est une version étendue de `tstsig_2.c`. Il prend le `pid` du serveur en argument de ligne de commande puis reçoit et affiche un message envoyé par le serveur.

Note : il est indispensable que le message soit transmis de manière fiable.

3 Signaux et zombies

Faites le TD du chapitre 7 (i.e section 7.8).