UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Licence 3 Informatique
Info0601

TP n°5

Les signaux

Le but de ce TP est de manipuler les appels systèmes relatifs à la gestion des signaux. Concentrez-vous en priorité sur les questions non subsidiaires.

1 Positionnement de gestionnaire

Dans cet exercice, nous souhaitons vérifier le comportement de la fonction `sigaction`.

Questions

1. Écrivez un programme qui se met en pause pendant 10 secondes à l'aide de la fonction `alarm`. Une fois les 10 secondes passées, le programme affiche un message puis termine "normalement".
2. Écrivez un programme qui se met en pause pendant 10 secondes et qui compte le nombre de signaux `SIGINT` reçus. Une fois la pause terminée, le programme affiche le nombre de signaux reçus.

Lorsqu'un gestionnaire est exécuté dans votre programme à la suite de la réception d'un signal, même si le programme doit s'arrêter, faites en sorte qu'il reprenne le cours normal d'exécution (dans le `main`, par exemple) avant de s'arrêter.

Questions subsidiaires

1. Écrivez un programme qui positionne un gestionnaire sur les signaux `SIGUSR1` et `SIGUSR2` qui permet d'afficher un message personnalisé pour chacun de ces signaux. Le programme continue son exécution (qui correspond à une boucle infinie) tant que le signal `SIGINT` n'est pas reçu. Bien sûr, le programme doit quitter normalement sur le return du `main` une fois le signal `SIGINT` reçu. Les signaux doivent être envoyés depuis le terminal à l'aide de la commande `kill`.
2. Modifiez le programme précédent pour que les gestionnaires des signaux `SIGUSR1` et `SIGUSR2` mettent en pause le processus pendant 10 secondes.
   (a) Que se passe-t-il si un signal est reçu avant la fin de la pause ?
   (b) Remplissez ou videz l'ensemble correspondant au champ `sa_mask` pour vérifier les différences de comportement.

2 Prise en main des ensembles de signaux

L'objectif de cet exercice est de tester le fonctionnement de `sigprocmask`, `sigpending` et `sigaction`, ainsi que les fonctions de gestion des ensembles de signaux.

Questions

1. Écrivez un programme qui bloque tous les signaux puis se met en pause pendant 20 secondes.
   (a) Peut-on utiliser la fonction `sleep` pour la pause ?
   (b) Et la fonction `alarm` ?
   (c) Vérifiez que les signaux sont bien bloqués pendant la pause (augmentez la durée si nécessaire).
   (d) Que se passe-t-il si le signal `SIGKILL` est envoyé ?
2. Après les 20 secondes de pause, le programme doit afficher les signaux reçus pendant sa pause.
   (a) Modifiez le programme pour afficher le code des signaux reçus.
   (b) Est-il possible de savoir si plusieurs signaux du même type ont été reçus ?

Questions subsidiaires

1. Modifiez le programme pour qu'un gestionnaire soit positionné sur le signal `SIGINT` puis qui attend 20 secondes avant le blocage de tous les signaux (le code de la question précédente). Le gestionnaire affiche simplement un message à chaque réception de signal.
   (a) Écrivez les modifications demandées.
   (b) Pourquoi ne peut-on pas utiliser `sleep` pour l'attente de 20 secondes ?
   (c) Le gestionnaire est-il encore appelé lorsque les signaux sont bloqués ?

3 Signaux temps-réel

L'objectif de cet exercice est de manipuler les signaux temps-réel.

Questions

1. Dans cette question, nous souhaitons écrire deux programmes. Le premier programme attend la réception du signal `SIGRTMIN` et retourne ce même signal au processus qui lui a envoyé. Le deuxième programme prend en argument le PID du processus correspondant au premier programme et lui envoie un signal `SIGRTMIN`. Il se met ensuite en attente d'un signal `SIGRTMIN`.
   (a) Comment le premier programme peut-il récupérer le PID du deuxième programme à la réception du signal ?
   (b) Comment faire en sorte que l'attente soit une attente passive ?
   (c) Écrivez les programmes.
2. Modifiez le deuxième programme pour qu'il envoie en plus un entier spécifié en argument (en plus du PID du premier programme). Le premier programme affiche l'entier reçu et renvoie le double. Le deuxième programme affiche la valeur de l'entier reçu.