

Signaux

Positionnement d'un gestionnaire

Cet article présente comment mettre en place un gestionnaire de réception de signal à l'aide de l'appel système `sigaction`. De plus, il présente comment envoyer un signal à un autre processus à l'aide de l'appel système `kill`.

1 Le programme client

Le programme client doit envoyer deux signaux au programme serveur. Le PID du serveur est passé comme argument. Pour envoyer le signal, le client utilise l'appel système `kill`.

```
pidServeur = atoi(argv[1]);
if(kill(pidServeur, SIGUSR1) == -1) {
    perror("Erreur_lors_de_l'envoi_du_signal");
    exit(EXIT_FAILURE);
}
```

Il envoie ensuite le deuxième signal de la même manière.

2 Le programme serveur

Le programme serveur doit attendre de recevoir les signaux `SIGUSR1` et `SIGUSR2`. Nous mettons en place un *handler* qui est une procédure appelée lors de la réception d'un message. Il met à jour la variable globale `cpt`.

```
int cpt = 0;    /* Permet de marquer les signaux recus */

void handler(int signum) {
    if(signum == SIGUSR1) {
        printf("Signal_1_reçu\n");
        cpt = cpt | 1;
    }
    if(signum == SIGUSR2) {
        printf("Signal_2_reçu\n");
        cpt = cpt | 2;
    }
}
```

La mise en place du gestionnaire est réalisée à l'aide de l'appel système `sigaction`.

```
struct sigaction action;

action.sa_handler = handler;
sigemptyset(&action.sa_mask);
action.sa_flags = 0;
if(sigaction(SIGUSR1, &action, NULL) == -1) {
    perror("Erreur_lors_du_positionnement_");
    exit(EXIT_FAILURE);
}
if(sigaction(SIGUSR2, &action, NULL) == -1) {
    perror("Erreur_lors_du_positionnement_");
    exit(EXIT_FAILURE);
}
```

Le serveur peut maintenant se mettre en attente des signaux. Pour cela, nous utilisons l'appel système `pause` dans une boucle infinie.

```
while(cpt != 3) {
    pause();
}
```

3 Compilation et exécution

Le `makefile` fourni permet de compiler les programmes précédents. Tapez la commande suivante :

```
make
```

Pour tester ces programmes, il faut d'abord lancer le serveur qui affiche son PID.

```
./signauxServeur
```

On peut lancer ensuite le client en spécifiant le PID affiché (en remplaçant les XXXX).

```
./signauxClient XXXX
```