Chapitre 2: Les signaux

Cours 2

Plan

1) Définition
2) Liste des signaux
3) Envoie d’un signal
4) Prise en compte d’un signal
    - Ignorer un signal

Définition

Définition 1:

Un signal est un moyen de communication entre processus.

Définition 2:

Un signal est un message Synchrone ou Asynchrone envoyé par le noyau à un processus pour indiquer l'occurrence d'un événement survenu au niveau du système.

Définition

- La prise en compte du signal par le processus oblige celui-ci à exécuter une fonction de gestion du signal.
- De façon simplifiée, quand un signal arrive, le noyau interrompt le processus qui déclenche alors son gestionnaire du signal pour réagir de façon appropriée.

Remarque:

Nous avons déjà rencontré le signal SIGCHLD qui permet au noyau d'avertir un processus père de la mort de son fils.

Liste des signaux

Num. Signal	Nom du signal	Événement associé
2	SIGINT	Interruption du clavier (frappe de « Ctrl+C »)
3	SIGQUIT	Caractère « quit » frappé depuis le clavier (Frappe de « Ctrl \ »)
8	SIGFPE	Erreur mathématique virgule flottante
9	SIGKILL	Terminaison forcée du processus
10	SIGUSR1	Signal à disposition de l'utilisateur
11	SIGSEGV	Référence mémoire invalide
12	SIGUSR2	Signal à disposition de l'utilisateur
13	SIGPIPE	Écriture dans un tube sans lecteur
14	SIGALRM	Horloge temps réel, fin de temporisation
15	SIGTERM	Terminaison du processus
17	SIGCHLD	Processus fils terminé
18	SIGCONT	Reprise de l'exécution d'un processus stoppé
19	SIGSTOP	Stoppe l'exécution d'un processus

Envoie d'un signal

Un processus peut recevoir un signal de deux façons différentes :

- C'est un autre processus qui lui envoie un signal via l'appel système **kill()**.

  *Exemple: La commande **$ kill -9 23** permet au shell d'envoyer le signal de code 9, c-à-d SIGKILL au processus de pid "23"*

- Ou, en exécutant une instruction non autorisée: division par 0, le gestionnaire d'exception associé, positionne un signal pour signaler l'erreur.

  Par exemple, la division par zéro amène à positionner (envoyer) le signal SIGFPE.

Envoie d'un signal

Un signal peut être envoyé soit à partir du Shell soit par un processus en appelant la primitive kill() dont le prototype est:

```
#include <sys/types.h>
#include <signal.h>
int kill(pid_t pid, int sig);
```

- Si pid > 0, le signal **sig** est envoyé au processus **pid**.
- Si pid = 0, le signal est envoyé à tous les processus du groupe du processus émetteur.
- La primitive kill retourne 0 en cas de succès et −1 en cas d'erreur.

Envoie d'un signal

Exemple 1: Que fait le programme signal1.c

```
int main(){
int pid;
pid=fork();
if(pid>0){
      printf("processus père=%d\n", getpid());
      printf("processus fils=%d\n", pid());
      kill(pid,SIGUSR1);
      printf("Envoie du signal au fils");
      }
exit(0);
}
```

Envoie d'un signal

Exercice:

1) Remplacer les paramètres de kill par getpid() et SIGKILL. Que se passe-t-il ?
2) Modifier le code du fils pour qu'il affiche le message « coucou » avec compteur de manière continue, pendant cet affichage le père est endormi pendant 2 secondes.
3) Après le réveil du père, ce dernier doit stopper l'exécution de son fils pendant 10 secondes, au bout de ce timing il le réveille.
4) Quand le compteur atteint la valeur 1000000, le fils tue son père, que devient le fils ? par qui est-il adopté ?
5) Comment tuer ce nouveau fils ?

Envoie d'un signal

Remarques:

- Si deux signaux identiques sont envoyés au même processus à un court intervalle de temps, un seul signal est délivré.
- Le signal ainsi délivré mais pas encore pris en compte par le processus destinataire est qualifié de signal pendant. Au bout d'un certain temps le système (noyau) délivre tous les signaux pendants pour ce processus.
- L'arrivée d'un signal sur un processus endormi après un appel tel que wait(), waitpid(), read(dans un tube) et pause() entraine le système à réveiller le processus et le faire passer à l'état prêt, à son élection le signal sera délivré.
- Les signaux n'ont aucun effet sur les processus zombies.

Prise en compte d'un signal

La prise en compte d’un signal par un processus s’effectue:

- Lorsque celui-ci s’apprête à quitter le mode noyau pour repasser en mode utilisateur.
- Avant de passer dans un état bloqué.
- En sortant de l’état bloqué.

Prise en compte d'un signal

Une fois que le signal est délivré au processus. Le processus peut donc indiquer au noyau ce qui doit se passer à la réception d'un signal, pour cela trois types d'action peuvent être réalisés :

1) **Ignorer le signal,**
2) Exécuter l'action par défaut,
3) Exécuter une procédure spécifique.

Prise en compte d'un signal

1) **Ignorer le signal:**

Lorsque le signal est ignoré, aucune action n'est entreprise au moment de sa prise en compte.

Cependant, une exception existe concernant le signal SIGCHLD; Lorsque ce signal est ignoré, le noyau force le processus à consulter les informations concernant ces fils zombies de manière à ce que leur prise en compte soit réalisée et que les blocs de contrôle associés soient détruits.