Les Signaux

Définition

Définition

Un signal est une interruption logicielle asynchrone.

Génération d'un signal

- par le terminal `CTRL^D`
- Exception matériel (division par 0)
- fonction `kill()`
- commande `kill)`
- logiciels

Quelques signaux

SIGABRT	SIGALRM	SIGBUS
SIGCHLD	SIGCONT	SIGHUP
SIGINT	SIGKILL	SIGPIPE
SIGQUIT	SIGSEGV	SIGSTOP
SIGTERM	SIGUSR1	SIGUSR2

Mécanisme

Le traitement du signal s'effectue de la manière suivante :
`void(*signal(int signo, void (*func)(int)))(int);`
modifie le traitement du signal `sig` par la fonction `func`. `func` peut prendre comme valeurs :

- `SIG_DFL` : réinitialise le traitement du signal par la valeur par défaut.
- `SIG_IGN` : ignore le signal.
- fonction : fonction appelée lors de la réception du signal.

Exemple

```
 void handler(int signo)
{
    signal(signo,handler);/*reposition du signal*/
    printf("Signal recu %d",signo);
    return;
}
int main()
{
    signal(SIGUSR1, handler);
    for(;;)
    {
       pause();
   }
}
```

Envoi de signaux

Fonctions d'envoi

- `int kill(pid_t pid, int signo)` : envoie le signal `signo` au processus de PID `pid`.
- `int raise(int signo)` : équivalent à `kill(getpid(), signo)`.

Autres fonctions

- `int alarm(int nsec)` : envoie un signal `SIGALRM` apres `nsec` secondes.
- `pause(void)` : suspend l'execution jusqu'à la prochaine reception d'un signal.

Ensemble de signaux

Unix permet la gestion d'ensembles de signaux. Les différentes fonctions sont les suivantes :

- `sigset_t my_sigs` : jeu de signaux.
- `int sigemptyset(sigset_t *set)` : effacer le jeu de signaux.
- `int sigfillset(sigset_t *set)` : ajouter tous les signaux dans un jeu de signaux.
- `int sigaddset(sigset_t *set, int signum)` : ajoute un signal dans un jeu.
- `int sigdelset(sigset_t *set, int signum)` : supprime un signal dans un jeu.
- `int sigismember(const sigset_t *set, int signum)` : renvoie la valeur 1 si le signal est dans le jeu, 0 sinon.

Ensemble de signaux

- `int sigaction(int signum, const struct sigaction *act, struct sigaction *oldact)` : définit les actions d'un signal.
- Structure utilisée pour définir les actions d'un jeu de signaux :

```
struct sigaction {
    void (*sa_handler)();
    sigset_t sa_mask;
    int sa_flags;
};
```

Ensemble de signaux

- `int sigprocmask(int how, const sigset_t *set, sigset_t *oldset)` : permet d'empêcher la mise en place de certains signaux. `how` vaut :
  - `SIG_BLOCK` : le jeu spécifié indique les signaux supplémentaires à bloquer (désactiver).
  - `SIG_UNBLOCK` : le jeu spécifié indique les signaux à débloquer (activer).
  - `SIG_SETMASK` : le jeu spécifié remplace le masque courant représentant les signaux bloqués.
- `int sigpending(sigset_t *set)` : récupére les signaux bloqués par `sigprocmask`.
- `int sigsuspend(const sigset_t *mask)` : débloque les signaux bloqués du masque.