Université Paris 7
Licence
Système
Année 2003-2004

TD no 6
Signaux : envoi et capture

Exercice 1 [ ]

Motivation : fonction `kill` et action par défaut à la réception de signaux (exple : SIGUSR1 termine le processus).

Exercice 2 [ ]

Motivation : L'utilisateur fournit une fonction pour traiter le signal. Utilisation de `sigaction` et `struct sigaction`.

kill

L'appel système `kill` peut être utilisé pour envoyer un signal à un processus ou à un groupe de processus.

```
KILL(2)                    Linux Programmer's Manual                    KILL(2)

NAME
       kill - send signal to a process

SYNOPSIS
       #include <sys/types.h>
       #include <signal.h>

       int kill(pid_t pid, int sig);

DESCRIPTION
       The  kill  system  call  can  be used to send any signal to any
       process group or process.

       If pid is positive, then signal sig is sent to pid.

       If pid equals 0, then sig is sent to every process in the
       process group of the current process.

       If  pid equals -1, then sig is sent to every process except for
       process 1 (init), but see below.

       If pid is less than -1, then sig is sent to every process in
```

```
       the process group -pid.

       If  sig  is 0, then no signal is sent, but error checking is still
       performed.

RETURN VALUE
       On success, zero is returned.  On error, -1 is returned, and  errno
       is set appropriately.
```

On peut aussi envoyer des signaux à partir d'un autre terminal en utilisant `kill -s signal pid`

signal

Pour positionner l'action, il y a:

```
#include <signal.h>

int (*signal(int sig,void (*func)(int)))(int);
```

(variante: void au lieu de int) La fonction peut tre SIG_DFL, SIG_IGN ou l'adres

```
    #include <signal.h>
    #include <stdio.h>

    int i = 0;

    void onsig(int code) {
            fprintf(stderr,"\nSignal %d, i=%d\n",code,i);
            if (code == SIGQUIT) exit(1);
            signal(code,onsig);
    }

    main() {
            signal(SIGINT,onsig);
            signal(SIGTERM,onsig);
            signal(SIGQUIT,onsig);
            while (1) {
                    if (++i % 100000 == 0) fprintf(stderr,".");
            }
            exit(0);
    }
```

Le signal SIGINT correspond l'interruption du terminal (CTRL-C).

Son action par dfaut est la terminaison.

sigaction

SYNOPSIS

```
#include <signal.h>

int sigaction(int signum, const struct sigaction *act, struct sigaction
*oldact);

int sigprocmask(int how, const sigset_t *set, sigset_t *oldset);

int sigpending(sigset_t *set);

int sigsuspend(const sigset_t *mask);
```

DESCRIPTION

The sigaction system call is used to change the action taken by a process on receipt of a specific signal.

signum specifies the signal and can be any valid signal except SIGKILL and SIGSTOP.

If act is non-null, the new action for signal signum is installed from act. If oldact is non-null, the previous action is saved in oldact.

The sigaction structure is defined as something like

```
struct sigaction {
    void (*sa_handler)(int);
    void (*sa_sigaction)(int, siginfo_t *, void *);
    sigset_t sa_mask;
    int sa_flags;
    void (*sa_restorer)(void);
}
```