# **MINITALK**

Fonctions autorisées :

* Write

**Signal :**

* + #include <signal.h>
  + Un signal est un message émit à destination d'un processus (ou d'un groupe de processus) pour l'informer que quelque chose vient de survenir ou pour lui demander de réaliser une tâche particulière. La fonction signal permet de définir le gestionnaire de signal à invoquer en cas de réception d'un signal particulier.

Une image contenant table

Description générée automatiquement

SIGACTION

**#include <[signal.h](https://linux.die.net/include/signal.h)>**

**int sigaction(int sig, const struct sigaction \*gestion,  
 struct sigaction \*ancien**

sigaction - Examiner et modifier l'action associée à un signal 

L'appel système **sigaction**() sert à modifier l'action effectuée par un processus à la réception d'un signal spécifique.

*signum* indique le signal concerné, à l'exception de **SIGKILL** et **SIGSTOP**.

Si *act* est non nul, la nouvelle action pour le signal *signum* est définie par *act*. Si *oldact* est non nul, l'ancienne action est sauvegardée dans *oldact*.

La structure *sigaction* est définie par quelque chose comme :

struct sigaction {

void (\*sa\_handler) (int);

void (\*sa\_sigaction) (int, siginfo\_t \*, void \*);

sigset\_t sa\_mask;

int sa\_flags;

void (\*sa\_restorer) (void);

};

Examinons ici les paramètres de sigaction

sig

Il s'agit là, bien entendu, du numéro du signal auquel nous allons associer un gestionnaire.

gestion

Pointeur vers une structure de type [struct sigaction](http://brunogarcia.chez.com/Unix/Docs/Signaux.html" \l "Traitement:gestionnaire:structure) désignant le gestionnaire ainsi qu'un éventuel masque de signaux à bloquer

anciens

Si ce paramètre est différent de 0 (NULL), alors la structure pointée est renseignée avec l'ancien gestionnaire présent, ce qui permet, par exemple, de le réinstaller ensuite.

**SIGEMPTYSET**

**sigemptyset**() initializes the signal set given by *set* to empty, with all signals excluded from the set.

SIGADSET

**sigaddset**() and **sigdelset**() add and delete respectively signal *signum* from *set*.

**sigemptyset**(), **sigfillset**(), **sigaddset**(), and **sigdelset**() return 0 on success and -1 on error.

SIGUSR1 and SIGUSR2 are user-defined signal

Short for **process identifier**, a PID is a unique number that identifies each running processes in an operating system, such as Linux, Unix, macOS, and Microsoft Windows. ... The output below shows a few of the processes running in Windows and their associated PIDs listed in the PID column.