

Unix : les SIGNAUX

Laure Danthony

1 Introduction

Les signaux sont les supports du système UNIX pour la notification d'événements. Notons qu'un signal est un numéro.

2 Différents types de signaux

Les signaux peuvent venir ...

- d'une exception matérielle : SIGSEGV (segmentation fault notifiée par le processeur), SIGBUS (bus error : on a essayé d'accéder à une adresse invalide), SIGFPE (floating point error : ex division par 0), SIGILL (illegal instruction : en effet, certains binaires ne peuvent pas être lus par le processeur), ...
- d'une exception logicielle : SIGPIPE (voir les tubes), ...
- d'une notification du système : SIGALRM (time), SIGCHLD (envoyé à un père quand son fils meurt), ...
- d'une émission explicite : SIGKILL (`kill -KILL`), SIGTERM, SIGSTOP (`^Z`), SIGCONT, SIGUSR1, SIGUSR2 (non utilisés par le système), ...
- d'événements "utilisateur" : SIGINT (`^C`), ...

3 Gestion des signaux

- On dispose d'une table où il y a trois cases par signal : la première case dit si le signal est masqué ou pas, la deuxième case s'il est en suspend ou pas, la troisième case est le traitement de signal.

- Pour modifier la première ligne, on peut utiliser les primitives *C* : `sigprocmask`, `sigemptyset` (met tout à 0) et `sigaddset`.
- Dans la troisième ligne, un traitant peut être :
 - `SIGDFL` : `SIGCHLD` est ignoré, `SIGSTOP` entraîne la suspension du processus, `SIGSEGV` et `SIGKILL` entraînent la terminaison et la génération de fichiers “core” (c’est-à-dire une photo prise par le système au moment du problème).
 - `SIGIGN` : ignoré.
 - une fonction *C* : `void (*handler) (int sig)`.