



UNIX Signal

Fabiano Arruda / Jorge Pereira



- O que são signals(sinais)?
- Tipos de sinais.
- O padrão POSIX.
- Como funciona?
- Como manipular um sinal?
- Dicas para manipulação de sinais.

O que são signals(sinais)?



Um sinal é uma notificação assíncrona transmitido entre um processo e outro, pelo OS.

O que são signals(sinais)?



- Sinais podem ser iniciados por programas, usuarios ou adiministradores.
- Sinais são normalmente usados pelo sistema operacional para notificar o processo, que algum evento aconteceu, os sinais então deveriam ser handled(manipulado) para poder criar um mecanismo de notificação para a aplicação.

Tipos de sinais.



- Todo sinal tem um numero(valor) atrelado ao mesmo. ex: SIGINT – 2, SIGKILL - 9
- Está definido no header `<signal.h>`
- Existe diferença entre os tipos de sinais dependendo do sistema operacional.

Tipos de Sinais.



SUS - Single Unix Specification

- SIGABRT - process aborted
- SIGALRM - signal raised by alarm
- SIGBUS - bus error "access to undefined portion of memory object"(SUS)
- SIGCHLD - child process terminated, stopped (*or continued)
- SIGCONT - continue if stopped
- SIGFPE - floating point exception -- "erroneous arithmetic operation"(SUS)
- SIGHUP - hangup
- SIGILL - illegal instruction
- SIGINT - interrupt
- SIGKILL - kill
- SIGPIPE - write to pipe with no one reading
- SIGQUIT - quit
- SIGSEGV - segmentation violation
- SIGSTOP - stop executing temporarily
- SIGTERM - termination
- SIGTSTP - terminal stop signal
- SIGTTIN - background process attempting to read ("in")

Tipos de Sinais.



- SIGTTOU - background process attempting to write ("out")
- SIGUSR1 - user defined 1
- SIGUSR2 - user defined 2
- *SIGPOLL - pollable event
- *SIGPROF - profiling timer expired
- *SIGSYS - bad syscall
- *SIGTRAP - trace/breakpoint trap
- SIGURG - urgent data available on socket
- *SIGXCPU - CPU time limit exceeded
- *SIGXFSZ - file size limit exceeded

O padrão POSIX.



Os sinais SIGBUS, SIGEMT, SIGIOT, SIGTRAP, e SIGSYS foram omitidos do POSIX.1 porque o seu comportamento depende da implementação e não pôde ser adequadamente categorizado. Implementações em conformidade podem emitir estes sinais, mas devem documentar as circunstâncias sob as quais eles são emitidos e salientar quaisquer restrições relativas à sua emissão.

Como Funciona?



O UNIX provê um mecanismo pra catching(capturar) um sinal(como uma exceção)

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

A função “handler” será invocada no sinal sig, quando o OS enviar para o processo que está tratando o respectivo sinal.

Como manipular um sinal?



- Os sinais podem ser manipulados de três maneiras:
- Default – faz o que o OS acha que deve fazer.
 - Ignore – ignora a interrupção.
 - Signal handler – o desenvolvedor prove uma função de callback para manipular o sinal.
 - O signal system call registra a função de callback que será executada quando o processo receber a interrupção.
 - Nem todos os sinais podem ser manipulados (SIGKILL and SIGSTOP for example).

DEMOS



Dicas para manipulação de sinais.



- ✓ Make it short - faça as funções de callback sempre curtas.
- ✓ Careful with "fault" signals – não tente mudar o comportamento padrão de alguns sinais, como (SIGBUS, SIGSEGV, SIGFPE), ou seja não de continuidade ao processo se o mesmo receber um desses sinais, a não ser que você saiba realmente o que está fazendo, que é raro o caso, no máximo fazer um cleanup e sair, de preferencia com um core dump(usando a função abort()).
- ✓



DUVIDAS???



Obrigado!!!

Contatos
Jorge Pereira <jorge@jorgepereira.com.br>
Fabiano Arruda <fabiano.arruda@hotmail.com>