

Programação Concorrente – Aula 09

Gilmar Gimenes Rodrigues

1 Sinais UNIX

- São interrupções de software.
- São totalmente assíncronos.
- Em versões antigas do Unix, os sinais eram “nao confiáveis”.
- POSIX1 padronizou o suporte para sinais confiáveis.
- Nomes de sinais comecam com SIG.
 - SIGINT
 - SIGKILL
 - SIGSEGV
 - SIGSTOP
 - SIGALARM
 - SIGFPE
 - SIGTERM

Alguns sao gerados pelo prprio hardware.

SIGBUS SIGTRAP SIGEMT

Dependetes da implementação do Unix

SIGUSR1 SIGUSR2

São sinais disponíveis para os usuários

O processo pode escolher quando um sinal é recebido:

- Ignora o sinal (com excessão de SIGKILL e SIGSTOP)
- Tratar o sinal (com excessão de SIGKILL e SIGSTOP)
- Ficar com o tratamento default (geralmente determinar o processo)

A interface mais simples é a função signal:

```
#include <signal.h>
void *signal(int signo, void (*func) (int));

typedef void sighandler_t (int);

sighandler_t * sinal (int signo, sighandler_t * func);
```

func pode ser:

- Ponteiro para a função tratadora do sinal.
- SIG_IGN para ignorar o sinal.
- SIG_DFL para tratamento default.

Em caso de erro, func retorna SIG_ERR.

exemplo:

```
#include <signal.h>

static void sig_usr (int signo) {
    if (signo == SIGUSR1)
        printf("Recebeu SIGUSR1\n");
    else if (signo == SIGUSR2)
        printf("Recebeu SIGUSR2\n");
    else {
        printf("Recebi o sinal %d\n", signo);
        exit (1);
    }
}

int main (void) {
    if (signal(SIGUSR1, sig_usr) == SIG_ERR) {
        printf ("Erro de chamada a signal\n");
        exit (1);
    }

    if (signal(SIGUSR2, sig_usr) == SIG_ERR) {
        printf ("Erro de chamada a signal\n");
        exit (1);
    }
    for (;;)
        pause (); // fica esperando um sinal, sem ocupar a CPU!
}
```

Voce pode testar esse programa enviando sinais para ele. Para isso, utilize o comando “kill”.

```
>$ kill -USR1 12345
Recebido SIGUSR1
>$ kill -USR2 12345
Recebido SIGUSR2
>$ kill 12345 # o mesmo que kill -TERM 12345
12345 terminated
```

Onde 12345 eh o PID do processo.

Comandos uteis:

- kill: manda um sinal especificado para um processo (ou process group) especificado.
- raise: manda um sinal para si mesmo.

- pause: espera sinal.
- alarm: ?

Para resolver o problema de sinais não confiáveis, POSIX1 definiu outra interface:

- sigaction (substitui o signal).
- sigprocmask.
- sigpending.
- sigsuspend.