

Sistemas Hardware-Software

Sinais: envio e notificação de finalização

Engenharia

Fabio Lubacheski
Maciel Vidal
Igor Montagner

Interação do Kernel com processos

- Os **processos** interagem com o **Kernel** por meio de chamadas de sistema (**Syscall**);
- Entretanto, eventos esporádicos, assíncronos, ou inesperadas, levam à necessidade do **Kernel** interagir com o **processo**.
- Como fazer o **Kernel** interagir com o **processo** ?

ex1.slides.c

```
#include <stdio.h>
int b=0;
int main()
{
    int i = 3/b;
    printf("fim do programa.\n");
    return 0;
}
```

Qual a saída nesse exemplo ?

O que é impresso ?

O que acontece nesse processo ?

ex2.slides.c

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

```
int main() {  
    int *px = (int*) 0x01010101;  
    *px = 0;  
    printf("fim do programa.\n");  
    return 0;  
}
```

Qual a saída nesse exemplo ?

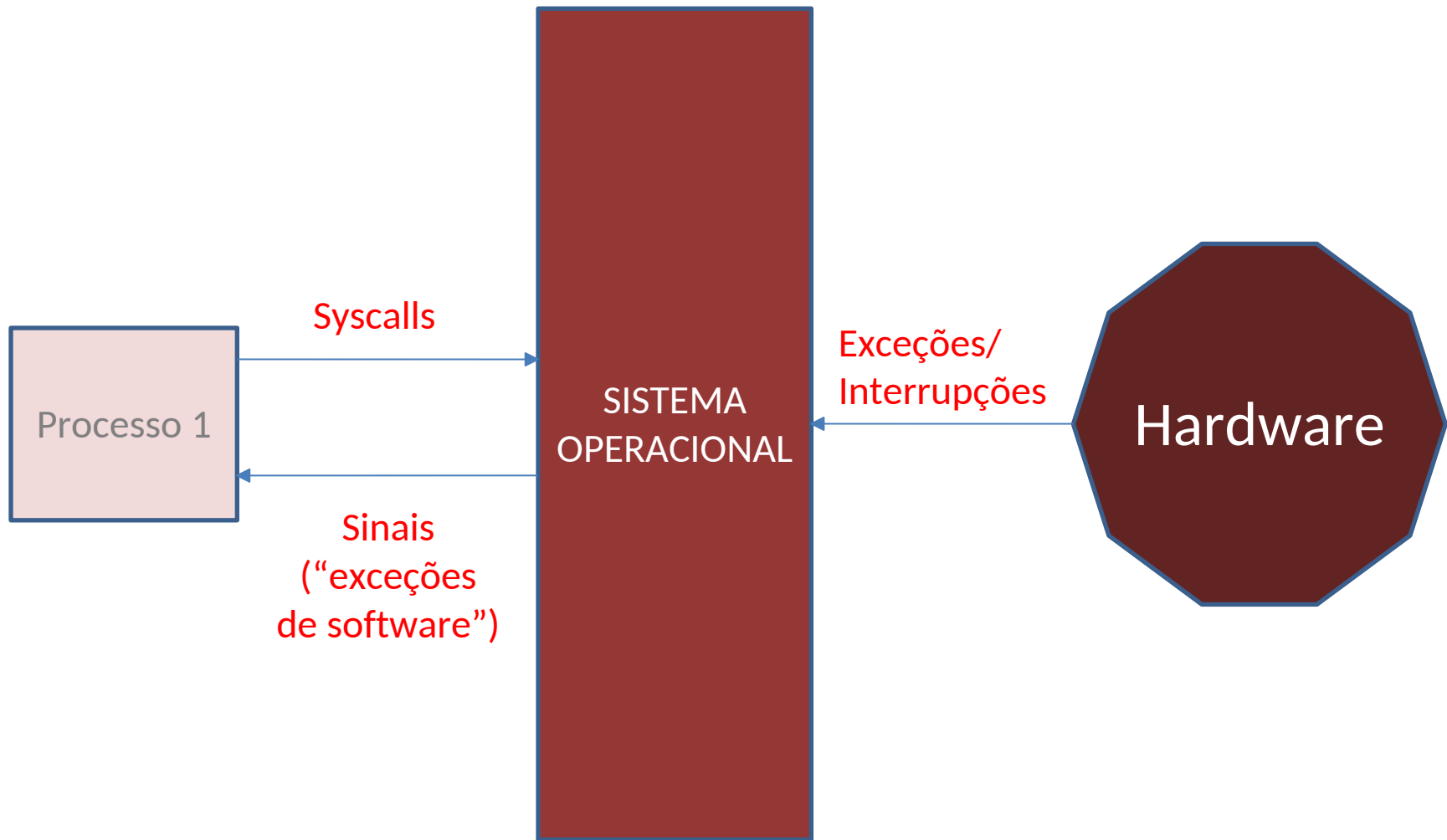
O que é impresso ?

O que acontece nesse processo ?

Interação do Kernel com processos

- Em sistemas POSIX a interação entre o Kernel e os processos é realizada através de **Sinais**, ou seja, uma **notificação de software** informando a ocorrência de um evento.
- O **Sinal** é gerado pelo **Kernel** quando o evento acontece e entregue ao processo que pode:
 - 1) Tratamento default (definido pelo Kernel); ou
 - 2) Capturado: neste caso, é executada a função definida no tratador do sinal definido pelo usuário (handler); ou
 - 3) Ignorado: neste caso, nada acontece. Funciona para todos os sinais.

Interação do Kernel com processos



Tipos de sinais

- O padrão POSIX define códigos inteiros para um número fixo de sinais, cada um é caracterizado por um nome simbólico iniciado com **SIG**
- Nos exemplos apresentados foram gerados os sinais de exceções de erro, **SIGFPE** e **SIGSEGV**, respectivamente.
- **SIGFPE**: ocorre em um erro de uma divisão por zero, e tem ação o termino do processo. (Floating point exception)
- **SIGSEGV**: ocorre quando é feito uma referência inválida à memória, e o processo também é finalizado (segmentation fault).

(Alguns) Sinais POSIX

Signal	Default Action	Description
SIGABRT	Terminate (core dump)	Process abort signal
SIGALRM	Terminate	Alarm clock
SIGCHLD	Ignore	Child process terminated, stopped, or continued.
SIGFPE	Terminate (core dump)	Erroneous arithmetic operation.
SIGSEGV	Terminate (core dump)	Segmentation fault.
SIGINT	Terminate	Terminal interrupt signal. (Ctrl+C)
SIGKILL	Terminate	Kill (cannot be caught or ignored).
SIGTERM	Terminate	Termination signal.

Exemplos de usos de sinais

- Ctrl+C envia um sinal **SIGINT** para o processo.
 - Ele pode ser capturado e fazer com que o programa feche conexões e arquivos abertos, por exemplo.
- O sinal **SIGSTOP (SIGTSTP)** é usado para deixar um processo em background. Ele fica parado até ser resumido por **SIGCONT**
- O sinal **SIGKILL** interrompe um processo imediatamente. Ele não pode ser ignorado.

Gerando sinais

- A partir do **Shell** é possível testar alguns sinais
O comando **kill** permite o envio de sinais a partir da **Shell**.

```
$ kill -9 9750
```

Onde **9750** é pid do processo, e o que seria o **-9** ?

Consulte a documentação do comando no manual do sistema **man 1 kill**.



Atividade prática

Recuperação de erros via wait (20 minutos)

1. Macros para checar sinais recebidos
2. Mensagem "amigável" de finalização



Atividade prática

Enviando sinais(20 minutos)

1. O programa kill e sua chamada de sistema



Atividade prática

Enviando sinais II (20 minutos)

1. A chamada de sistema kill

Insper

www.insper.edu.br