

# Sistemas Hardware-Software

Aula 16 - Sinais: envio e notificação de finalização

**Engenharia**

Fabio Lubacheski

Maciel C. Vidal

Igor Montagner

Fábio Ayres

# Correção da aula passada

## **Exercício copy\_file.c**

1. Abrir dois arquivos e copiar o conteúdo de um arquivo para outro

# 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>
```

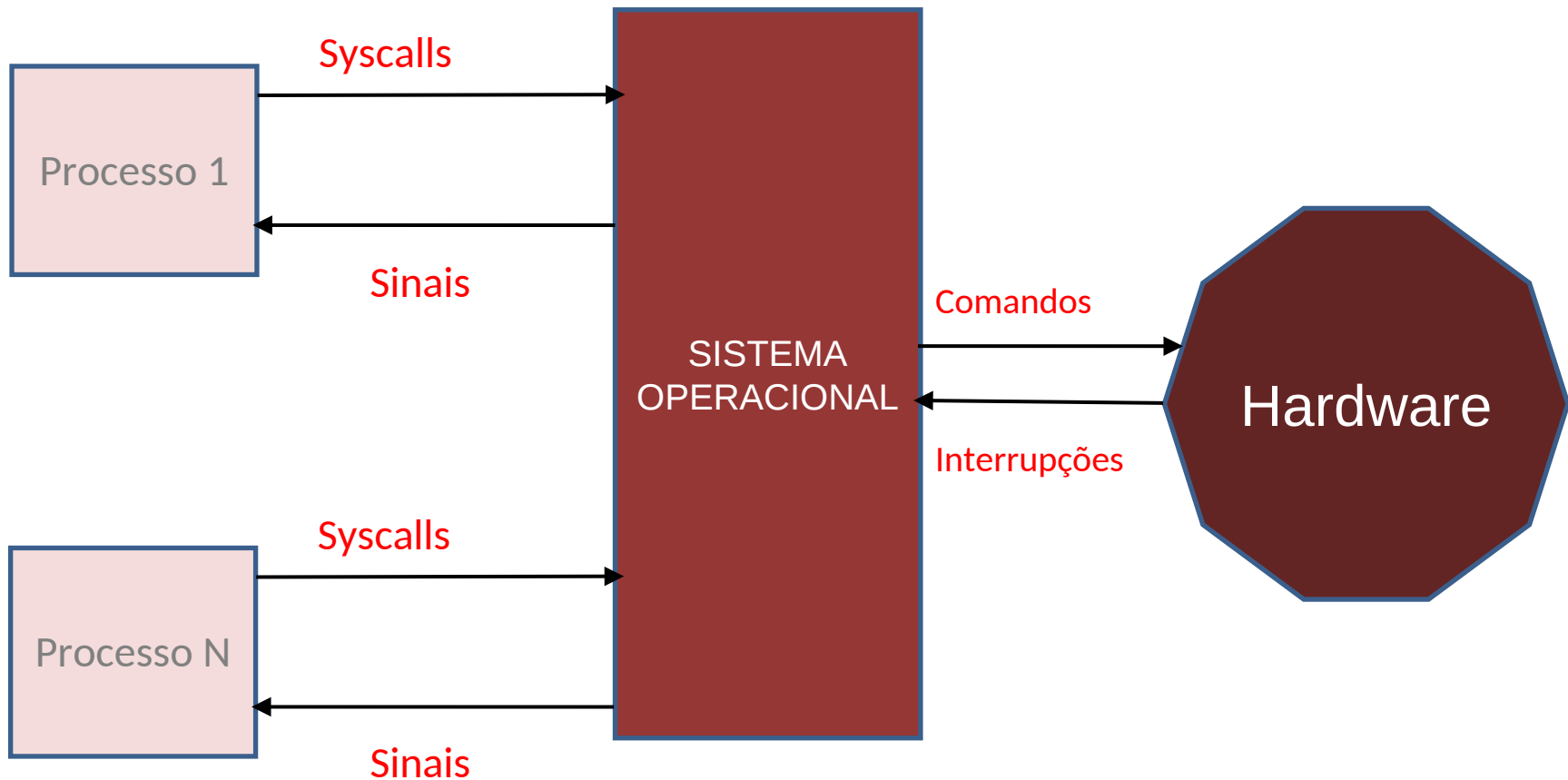
```
int main() {  
    int *px = (int*) 0x01010101;  
    *px = 0;  
    printf("fim do programa.\n");  
    return 0;  
}
```

**Qual a saída nesse exemplo ?**

**O que é impresso ?**

**O que acontece nesse processo ?**

# Interação do Kernel com processos



# 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**
- No exemplo apresentado foi gerado o sinal de erro **SIGSEGV**, que 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.

# Outros exemplos de usos de sinais

- Quando um usuário pressiona Ctrl+C no terminal é enviado um sinal **SIGINT** (interrupção de sinal) para o processo.
  - O sinal pode ser capturado e fazer com que o programa feche conexões e arquivos abertos, antes de encerrar o processo.
- O sinal **SIGTERM** (terminação de sinal) parecido ao **SIGINT**, mas o **SIGTERM** é enviado por outros processos ou pelo próprio sistema.
- O sinal **SIGKILL** interrompe um processo imediatamente. **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, execute **man 7 signal** e procure por **Signal numbering for standard signals**.



# Atividade prática

**Recuperando informações de erros usando wait (20 minutos)**

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



# Atividade prática

**Envio de sinais via terminal (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](http://www.insper.edu.br)