Insper

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

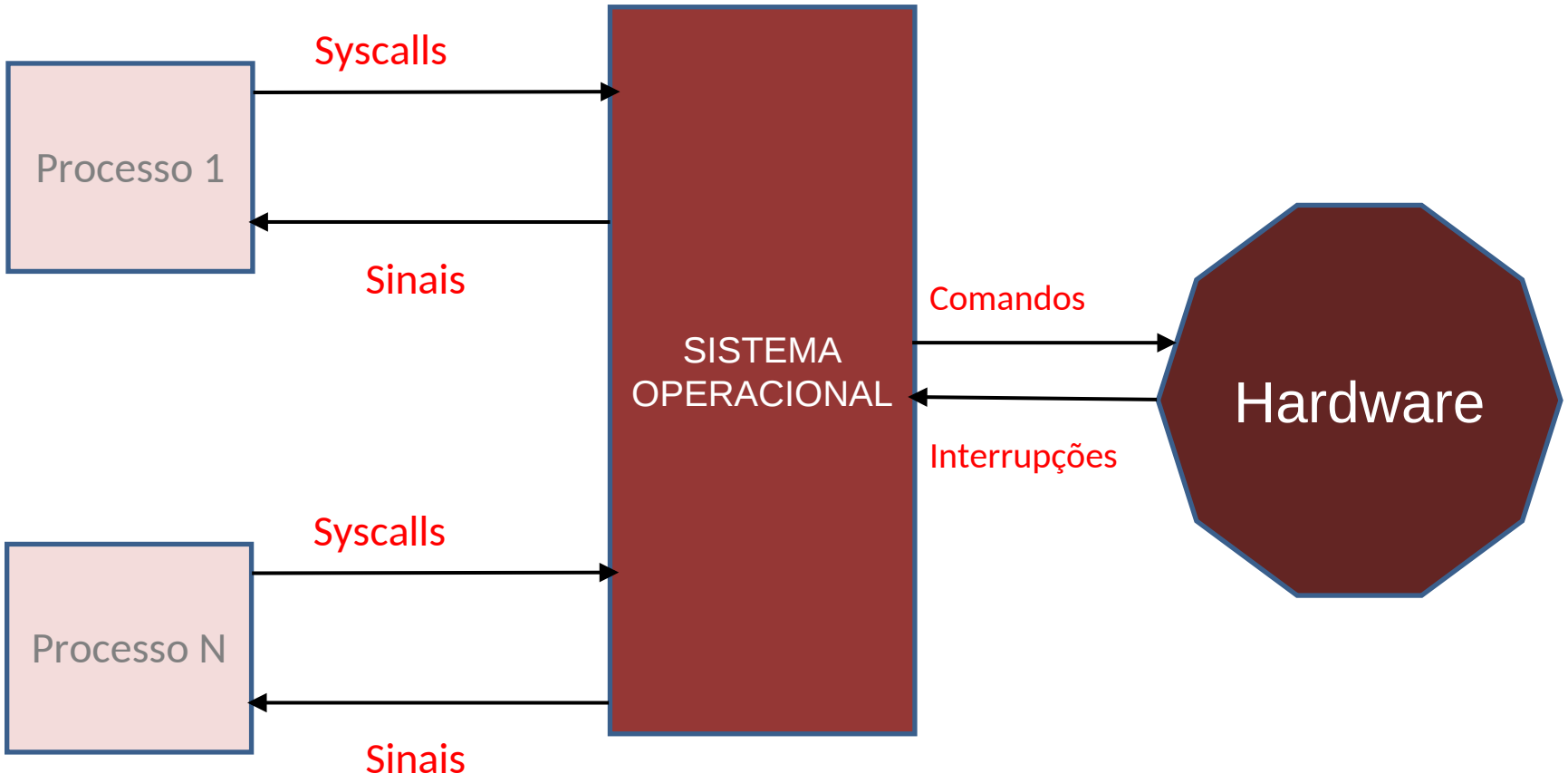
```c
#include <stdio.h>

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



Insper

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).

Sinais POSIX (alguns)

Sinal	Código	Ação Padrão	Descrição
SIGINT	**2**	Terminar	Interrupção (Ctrl+C).
SIGFPE	**8**	Core Dump	Erro aritmético: divisão por zero, overflow, underflow.
SIGKILL	**9**	Terminar	**Terminação forçada, não pode ser capturado ou ignorado.**
SIGSEGV	**11**	Core Dump	Violação de segmento (acesso inválido à memória).
SIGPIPE	**13**	Terminar	Escrita em pipe sem leitor.
SIGALRM	**14**	Terminar	Alarme de timer expirado.
SIGTERM	**15**	Terminar	Pedido de término gracioso.
SIGCHLD	**17**	Ignorar	Filho terminou ou parou.
SIGSTOP	**19**	Parar	**Pausa forçada — não pode ser capturado ou ignorado.**
SIGTSTP	**20**	Parar	Pausa interativa (Ctrl+Z).

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