

# DCA-0125 Sistemas de Tempo Real

Luiz Affonso Guedes  
[www.dca.ufrn.br/~affonso](http://www.dca.ufrn.br/~affonso)  
[affonso@dca.ufrn.br](mailto:affonso@dca.ufrn.br)



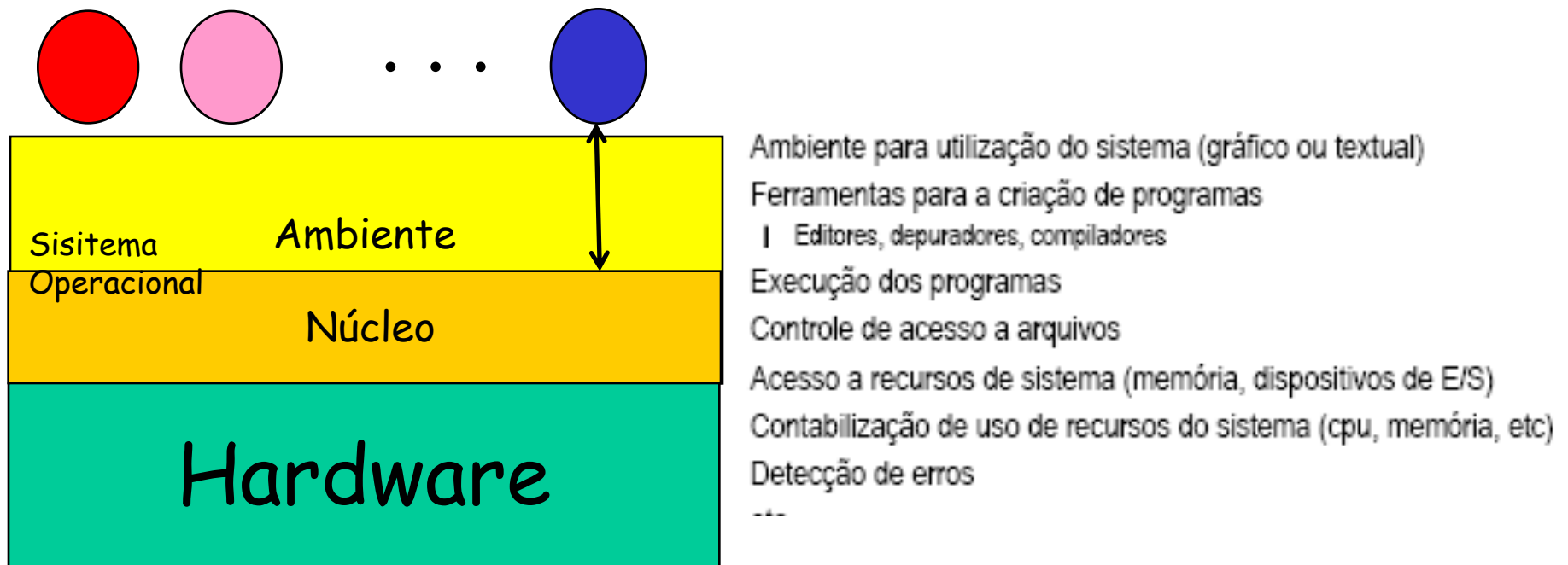
# Conceitos de Sinais

# Conteúdo

- ❑ Definição de sinais entre processos no Sistema Operacional
- ❑ Tipos de Sinais
- ❑ Exemplos em C/C++

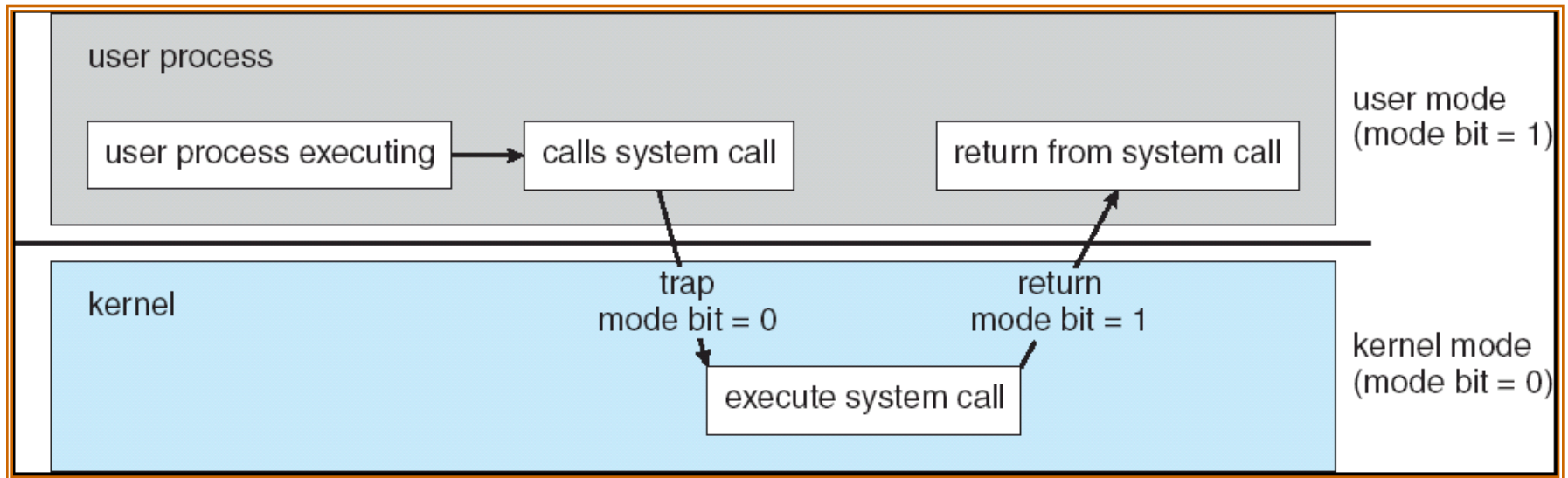
# Recordando SO

## Estrutura e funcionalidades básicas



# Recordando SO

## Interação entre núcleo ambiente



# Exemplos de Chamadas do Sistema

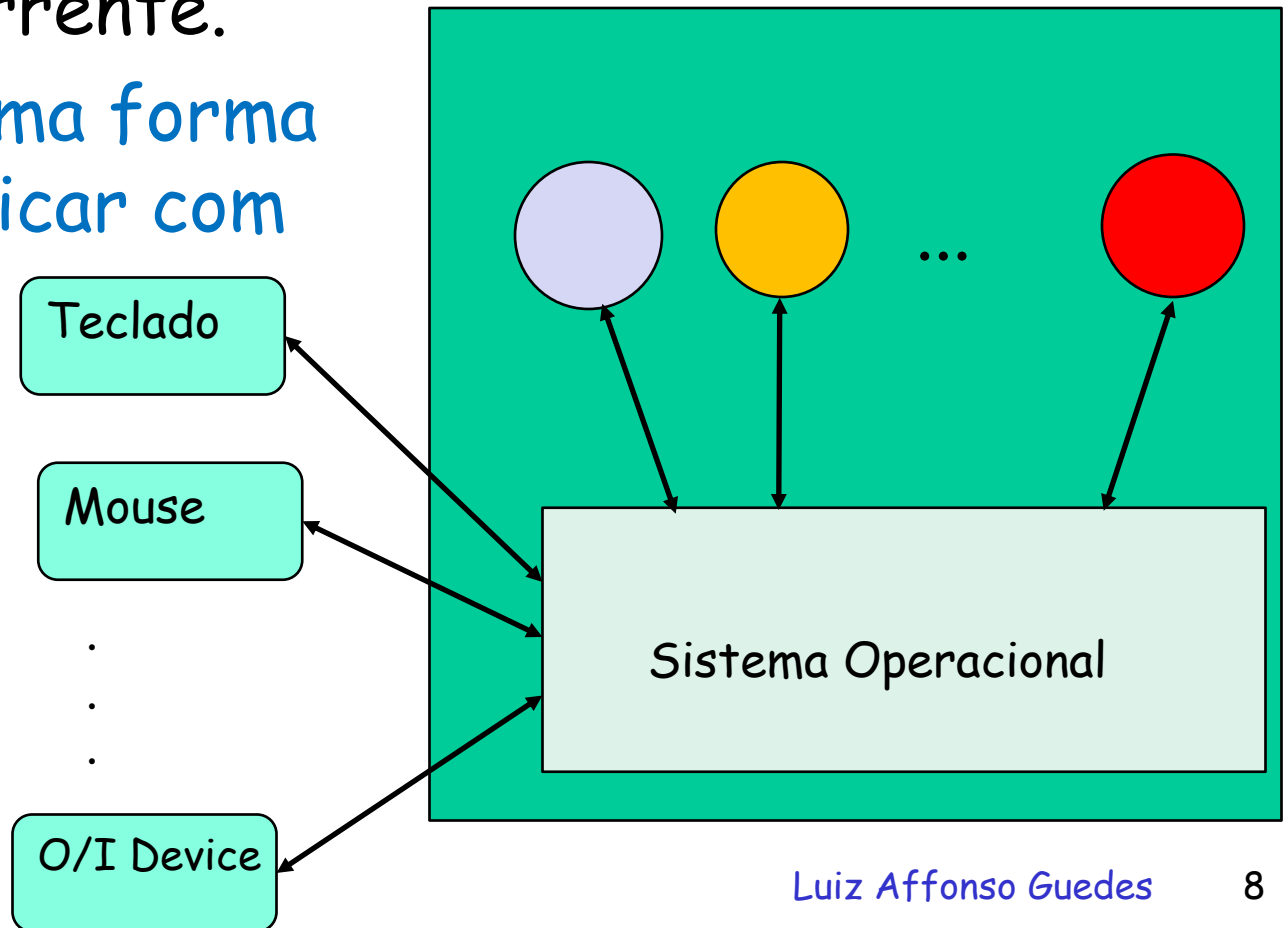
UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

# Conceitos Fundamentais

- ❑ Processos
- ❑ Interrupção
- ❑ Sincronização, **comunicação**, controle e proteção entre processos

# Sinais em Sistemas Operacionais

- ❑ São uma forma de interrupção do processo corrente.
- ❑ Também é uma forma de se comunicar com processos.



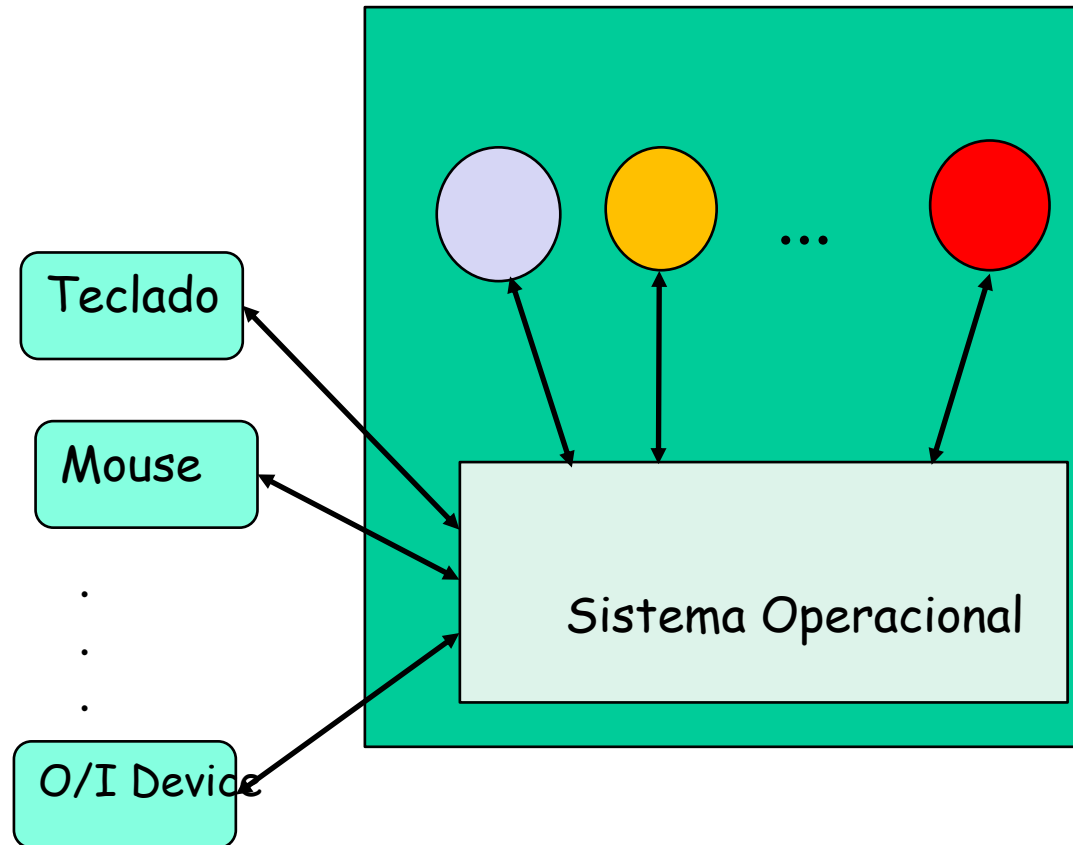


# Sinais em Sistemas Operacionais

□ Há sinais síncronos e Assíncronos.

□ Exemplos:

- Divisão por zero
  - Acesso inválido à memória
  - Interrupção do programa
  - Término de um processo filho - Alarme



# Sinais em Sistemas Operacionais

□ include <signal.h>. SIG\* → são números de ID dos sinais

Nome	Descrição	Origem	Acção por defeito
SIGABRT	Terminação anormal	abort()	Terminar
SIGALRM	Alarme	alarm()	Terminar
SIGCHLD	Filho terminou ou foi suspenso	S.O.	Ignorar
SIGCONT	Continuar processo suspenso	shell (fg, bg)	Continuar
SIGFPE	Excepção aritmética	hardware	Terminar
SIGILL	Instrução ilegal	hardware	Terminar
SIGINT	Interrupção	teclado (^C)	Terminar
SIGKILL	Terminação ( <i>non catchable</i> )	S.O.	Terminar
SIGPIPE	Escrever num <i>pipe</i> sem leitor	S.O.	Terminar
SIGQUIT	Saída	teclado (^I)	Terminar
SIGSEGV	Referência a memória inválida	hardware	Terminar
SIGSTOP	Stop ( <i>non catchable</i> )	S.O. (shell - stop)	Suspender
SIGTERM	Terminação	teclado (^U)	Terminar
SIGTSTP	Stop	teclado (^Y, ^Z)	Suspender
SIGTTIN	Leitura do teclado em <i>backgd</i>	S.O. (shell)	Suspender
SIGTTOU	Escrita no écran em <i>backgd</i>	S.O. (shell)	Suspender
SIGUSR1	Utilizador	de 1 proc. para outro	Terminar
SIGUSR2	Utilizador	idem	Terminar

# Sinais em Sistemas Operacionais

## □ Exercício:

- Compile e execute o programa sinal1.cpp
  - `g++ sinal1.cpp -o sinal1`
  - `./sinal1`
- Tente terminar o programa (processo) com CTRL+C
- Analise o resultado
- Modifique o programa para que se imprima o valor de SIGINT

# Sinais em Sistemas Operacionais

## □ Exercício:

- Compile e execute o programa sinal2.cpp
  - `g++ sinal2.cpp -o sinal2`
  - `./sinal2`
- Tente terminar o programa (processo) com CTRL+C
- Analise o resultado
- Modifique o programa para que o processo consiga terminar após a terceira tentativa de CTRL+C.

# Sinais em Sistemas Operacionais

- ❑ Um processo pode programar a resposta a dar a cada tipo de sinal. As respostas permitidas são:
  - ignorar o sinal,
  - tratar o sinal com uma rotina do programador (*catch*) e
  - deixar que o processo tenha a resposta por defeito (*default action*) para esse tipo de sinal.

# Sinais em Sistemas Operacionais

## □ Exercício:

- Compile e execute o programa sinal3.cpp
  - `g++ sinal3.cpp -o sinal3`
  - `./sinal3`
- Espere o programa terminar por si só
- Analise o resultado

## □ Exercício:

- Compile e execute o programa sinais\_help.cpp
  - `g++ sinais_help.cpp -o sinais_help`
  - `./sinais_help`

# Sinais em Sistemas Operacionais

- ❑ Para ignorar um sinal utilize:
  - `signal(SIGNAL, SIG_IGN)`
  
- ❑ Exercício para casa:
  - Reescreva o programa `signal1.cpp` para que ele ignore o sinal `SIGCONT`

# Sinais em Sistemas Operacionais

## ❑ Comando kill do Shell:

### Descrição

Para enviar qualquer sinal para um processo a partir da linha de comando, use `kill` .

Para listar todos os sinais disponíveis, use a opção `-l` (L minúsculo). Também fornecemos uma lista de sinais do Linux para você usar como referência rápida.

Os sinais usados com frequência incluem `HUP` , `INT` , `KILL` , `STOP` , `CONT` e `0` .

Os sinais podem ser especificados de três maneiras:

- por número (por exemplo, `-9` )
- com o prefixo "SIG" (por exemplo, `-SIGKILL` )
- sem prefixo "SIG" (por exemplo, `-KILL` ).

❑ No terminal1, execute: `kill -l`

❑ No terminal1, execute `./sinal1`

❑ No terminal 2, execute: `kill -2 <PID de sinal1>`

❑ O que ocorre se for executado `kill -9 <PID_Process>?`

❑ O que faz a função `sigkill(PID, SIG)?`



# Sinais em Sistemas Operacionais

## ■ Exercício:

- Compile e execute o programa sinal3.cpp
  - `g++ sinal4.cpp -o sinal4`
  - `./sinal4`
- Espere o programa terminar por si só
- Analise o resultado

# Sinais em Sistemas Operacionais

## ■ Exercício:

- No terminal 1:
  - Compile e execute o programa `vitima.cpp`
  - `g++ vitima.cpp -o vitima`
  - `./vitima`
- No Terminal 2:
  - Compile e execute o programa `matador.cpp`
  - `g++ matador.cpp -o matador`
- Analise o resultado

## ■ Exercício para casa:

- Reescreva o programa `matador.cpp` para que ele possa também possa pausar e continuar um processo.

# Sinais em Sistemas Operacionais

## ❑ Sinal de Alarme:

- É um sinal assíncrono.
- Envia um sinal do SIGALRM para o processo após alguns segundos.

## ❑ Estrutura

- `unsigned int alarm(unsigned int seconds);`

# Sinais em Sistemas Operacionais

## ■ Exercício:

- Compile e execute o programa sinal5.cpp
  - `g++ sinal5.cpp -o sinal5`
  - `./sinal5`
- Analise o resultado

## ■ Exercício para casa:

- Modifique o programa sinal5.cpp para que ele possa ser terminado por CTRL+C somente após 5 segundos de execução.

# Sinais em Sistemas Operacionais

## ■ Exercício:

- Compile e execute o programa sinal6.cpp
  - `g++ sinal6.cpp -o sinal6`
  - `./sinal6`
- Analise o resultado

## ■ É possível ignorar um sinal de alarme?

- `signal(SIGALRM, SIG_IGN)`