

Chamadas de Sistema

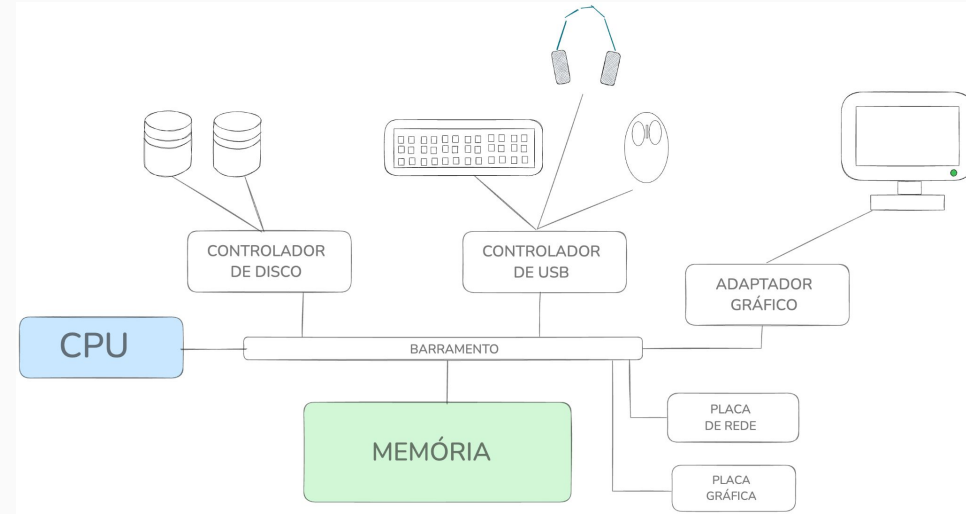
Sistemas Operacionais

Prof. Pedro Ramos
pramos.costar@gmail.com

ANTERIORMENTE: SO E ARQUITETURA

- O SO faz a interface entre a arquitetura e o usuário

. . . mas exige alguns detalhes do hardware.



ANTERIORMENTE: SO E ARQUITETURA

SERVIÇO	SUORTE NA ARQUITETURA
<i>Proteção</i>	Modo kernel/ modo usuário; instruções protegidas; registradores limite e base
<i>Interrupções</i>	Vetores de interrupções
<i>Chamadas de Sistema</i>	Vetores de instruções e instruções de interrupção
<i>I/O</i>	Interrupções e mapeamento em memória
<i>Escalonamento, recuperação de erros, contabilidade de recursos</i>	Temporizador
<i>Sincronização</i>	Instruções atômicas
<i>Memória Virtual</i>	TLB - Translation Lookaside Buffers (Buffers de tradução de endereços)

NESTA AULA: CHAMADAS DE SISTEMA

- Mais sobre syscalls (chamadas de sistema)

NESTA AULA: ESTRUTURAÇÃO DO SISTEMA OPERACIONAL

- Mais sobre syscalls (chamadas de sistema)
- Organização de componentes em um SO
- 4 Exemplos de organização em SO:
 - Kernel monolítico
 - Arquitetura em camadas
 - Micro-kernel
 - Modular

SO's modernos utilizam elementos das 4 abordagens.

Vamos visualizar os *tradeoffs* entre elas!

EXERCÍCIO EM SALA

Em 2013: LANÇAMENTO DO **iPhone 5s**

- Primeiro smartphone 64-bits
- iOS 7.0: primeiro SO 64-bits

1. Quais benefícios e/ou desvantagens ao implementar SO's em 64 bits, em relação aos SO's de 32 bits, em smartphones?

- Adição de um co-processador (M7) além do processador principal (A7)

Descarrega trabalho (processamento de dados de sensores - acelerômetro, giroscópio, biometrias, etc) da CPU para esse co-processador.

2. Quais benefícios e/ou desvantagens em adicionar um co-processador para realizar processamento de dados de sensores?

CHAMADAS DE SISTEMA

- É uma **interface** (API) para serviços providos pelo SO
- Tipicamente escrito em C ou C++
- Programas acessam a API em alto nível, não fazem chamadas de sistema diretamente.
- As 3 API's de sistema mais famosas
 - Win32
 - POSIX (Unix, Linux, Mac OS X)
 - Portable Operating System Interface (X -> portabilidade)
 - Java API (JVM)

PORQUÊ UTILIZAR UMA API AO INVÉS DE CHAMADAS DE SISTEMA DIRETAS?

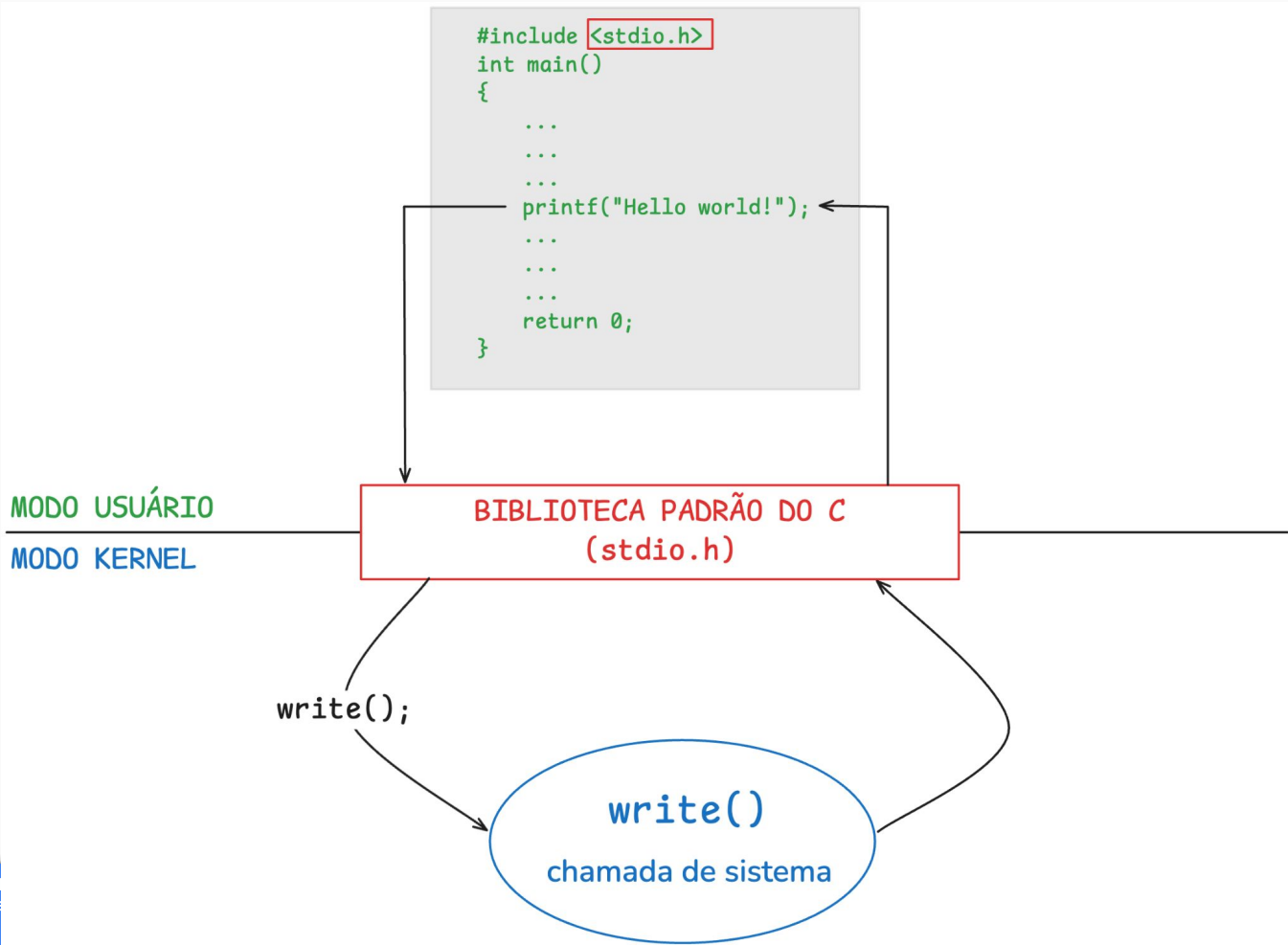
CHAMADAS DE SISTEMA

- É uma **interface** (API) para serviços providos pelo SO
- Tipicamente escrito em C ou C++
- Programas acessam a API em alto nível, não fazem chamadas de sistema diretamente.

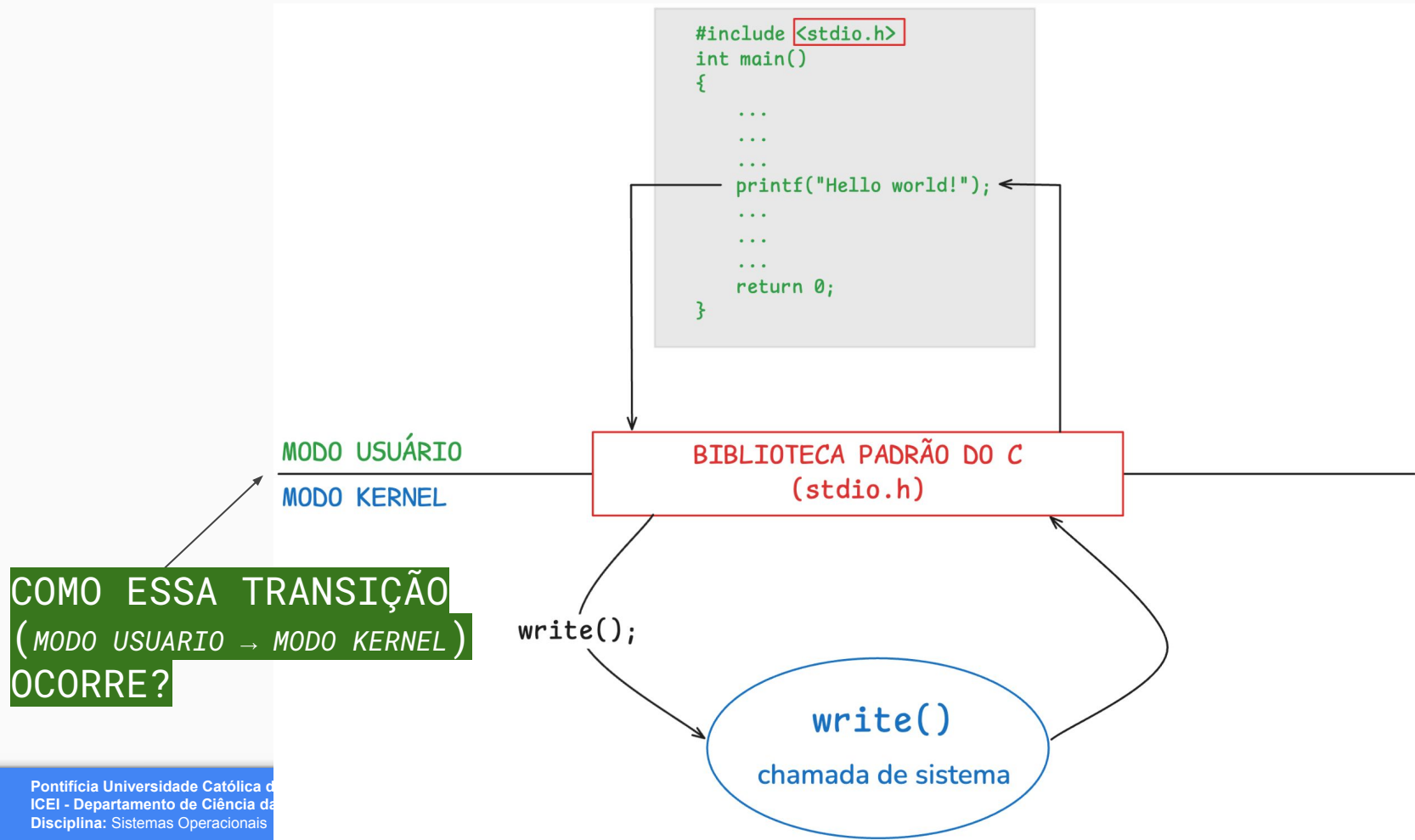
PORQUÊ UTILIZAR UMA API AO INVÉS DE CHAMADAS DE SISTEMA DIRETAS?

SEGURANÇA / SIMPLICIDADE / PRATICIDADE /
PORTABILIDADE / ... e o quê mais ?

EXEMPLO: BIBLIOTECA PADRÃO DO C



EXEMPLO: BIBLIOTECA PADRÃO DO C



EXEMPLO: BIBLIOTECA PADRÃO DO Win32

valor de retorno



Não é necessário decorar todas as chamadas de sistema.

Uma API de chamada de sistema (*syscall*) é uma função que tem tipo, parâmetros, e escreve algum valor de retorno no endereço de retorno. O SO provê a implementação dessas API's.

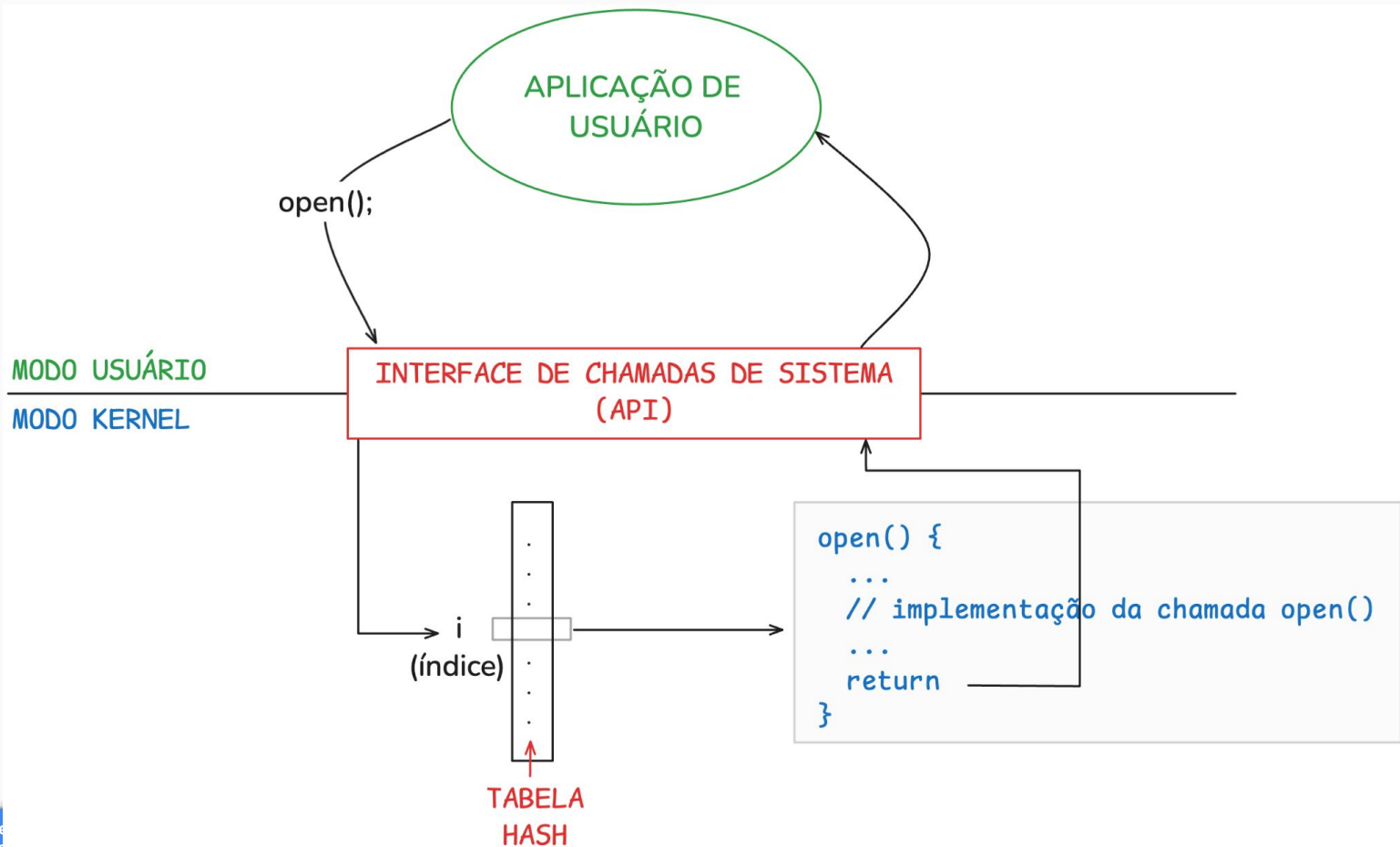
IMPLEMENTAÇÃO DE CHAMADA DE SISTEMA

- Como implementar?

IMPLEMENTAÇÃO DE CHAMADA DE SISTEMA

- Como implementar? **TABELA HASH**
- 1 número associado à cada `syscall`
A API de `syscall` tem acesso a essa tabela indexada de acordo com esses números
- Programa de usuário não sabe detalhes de implementação da chamada de sistema, apenas obedece ao `S0`.

RELACIONAMENTO ENTRE API - CHAMADA DE SISTEMA - SO



RELACIONAMENTO ENTRE API - CHAMADA DE SISTEMA - SO

OK, RESOLVEMOS COMO O SO ENCONTRA O CÓDIGO DE UMA CHAMADA DE SISTEMA A SER EXECUTADO. BASTA INDEXAR AS CHAMADAS EM UMA TABELA HASH.

LINHAS DA TABELA: [ÍNDICE DA CHAMADA, CÓDIGO DA CHAMADA]

RELACIONAMENTO ENTRE API - CHAMADA DE SISTEMA - SO

OK, RESOLVEMOS COMO O SO ENCONTRA O CÓDIGO DE UMA CHAMADA DE SISTEMA A SER EXECUTADO. BASTA INDEXAR AS CHAMADAS EM UMA TABELA HASH.

LINHAS DA TABELA: [ÍNDICE DA CHAMADA, CÓDIGO DA CHAMADA]

PORÉM, ALGUMAS CHAMADAS DE SISTEMA RECEBEM PARÂMETROS.

COMO PASSAR OS PARÂMETROS PARA A CHAMADA DE SISTEMA?

RELACIONAMENTO ENTRE API - CHAMADA DE SISTEMA - SO

COMO PASSAR OS PARÂMETROS PARA A CHAMADA DE SISTEMA?

Métodos tradicionais: passagem por valor ou por referência.

Qual o problema?

Atenção: O CÓDIGO DE USUÁRIO É UM ESPAÇO DE MEMÓRIA (processo) SEPARADO DO CÓDIGO DE KERNEL.

RELACIONAMENTO ENTRE API - CHAMADA DE SISTEMA - SO

COMO PASSAR OS PARÂMETROS PARA A CHAMADA DE SISTEMA?

SERVIÇO:
→ PROTEÇÃO

Porquê o SO protege os espaços de memória do programa de usuário em relação aos programas de kernel? SEGURANÇA.

Não podemos permitir que o programa de usuário passe uma referência na memória do programa de usuário para ser lida pelo programa do kernel.

RELACIONAMENTO ENTRE API - CHAMADA DE SISTEMA - SO

COMO PASSAR OS PARÂMETROS PARA A CHAMADA DE SISTEMA?

- **Simple:** passar os parâmetros em registradores
 - Pode não haver registradores o suficiente. Queremos passar strings de parâmetros, vamos precisar usar a memória principal.
- **Parâmetros são copiados em blocos (arrays) ou tabela** para outra região de memória e o endereço dessa região é passado para a chamada de sistema.
 - Linux e Solaris
- **Parâmetros colocados na pilha** pelo usuário e desempilhados pelo kernel

Quais os tradeoffs envolvidos nos 3 métodos?

CHAMADAS DE SISTEMA - UNIX vs WINDOWS

Categoria	Windows	Unix
Controle de Processos	CreateProcess()	fork()
	ExitProcess()	exit()
	WaitForSingleObject()	wait()
Manipulação de Arquivos	CreateFile()	open()
	ReadFile()	read()
	WriteFile()	write()
	CloseHandle()	close()
Manipulação de Dispositivos	SetConsoleMode()	ioctl()
	ReadConsole()	read()
	WriteConsole()	write()
Manutenção de Informação	GetCurrentProcessID()	getpid()
	SetTimer()	alarm()
	Sleep()	sleep()
Comunicação	CreatePipe()	pipe()
	CreateFileMapping()	shmget()
	MapViewOfFile()	mmap()
Proteção	SetFileSecurity()	chmod()
	InitializeSecurityDescriptor()	umask()
	SetSecurityDescriptorGroup()	chown()

PERGUNTAS?

REFERÊNCIAS

- **TANENBAUM, Andrew.** Sistemas operacionais modernos.
- **SILBERSCHATZ, Abraham et al.** Fundamentos de sistemas operacionais: princípios básicos.
- **MACHADO, Francis; MAIA, Luiz Paulo.** Arquitetura de Sistemas Operacionais.
- **CARISSIMI, Alexandre et al.** Sistemas operacionais.