

Conceitos de Sistemas Operativos

Componentes de um S. O.

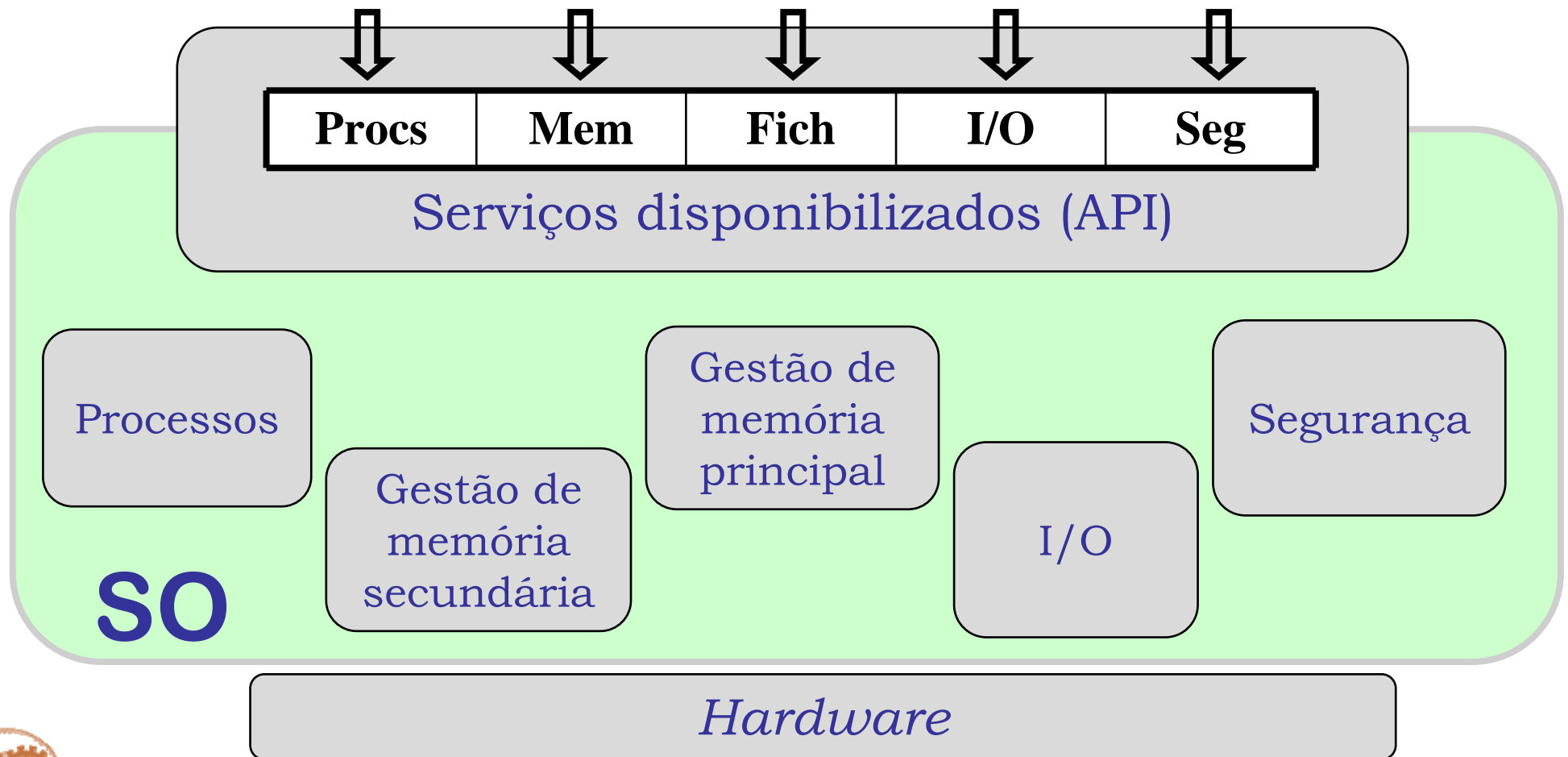
Vamos abordar os vários componentes de um S.O. e seus principais conceitos:

- Serviços disponibilizados pelo SO (a sua API)
- Gestão de processos
- Gestão de memória principal
- Gestão de memória secundária (sistema de ficheiros)
- Gestão de *Input/Output*
- Gestão de segurança



Serviços disponibilizados pelo S.O.

Acesso aos Serviços dos vários componentes
por interfaces bem definidas



Gestão de processos

■ Principais conceitos:

- **Processo** - é uma instância de um programa em execução
- **Tarefa** (thread) – um fio de execução dentro de um processo

■ Acções da gestão de processos:

- Suporte a Processos:
 - estruturas: espaço de endereçamento próprio, contendo o código executável, dados e stacks; contextos de execução (registos do cpu); controlo dos recursos em uso
 - acções: criar, remover, suspender, reactivar
- Suporte a Threads:
 - estruturas: implementação user ou kernel level
 - acções: criar, remover, suspender, reactivar
- CPU Scheduling – algoritmo (política) de scheduling
- Sincronismo entre threads: exclusão mútua
 - Secções críticas, semáforos, monitores, transacções atómicas
- Comunicação entre processos: sinais, pipes, sockets, Remote Procedure Calls



Gestão de memória principal

- **Espaço de endereçamento dos processos**
 - Noção de endereço físico e lógico
 - Endereçamento linear e/ou segmentado
 - Acesso à memória linear e/ou segmentada
- **Protecção de memória** – limites, tipo de acesso
- **Memória virtual:**
 - Swaping de processos
 - Swaping de páginas
 - Relocalização de páginas de memória
 - Algoritmos de selecção de páginas para swapping, noção de set de páginas
- **Dynamic loading of libraries (DLLs)**
- **Memória partilhada entre processos**

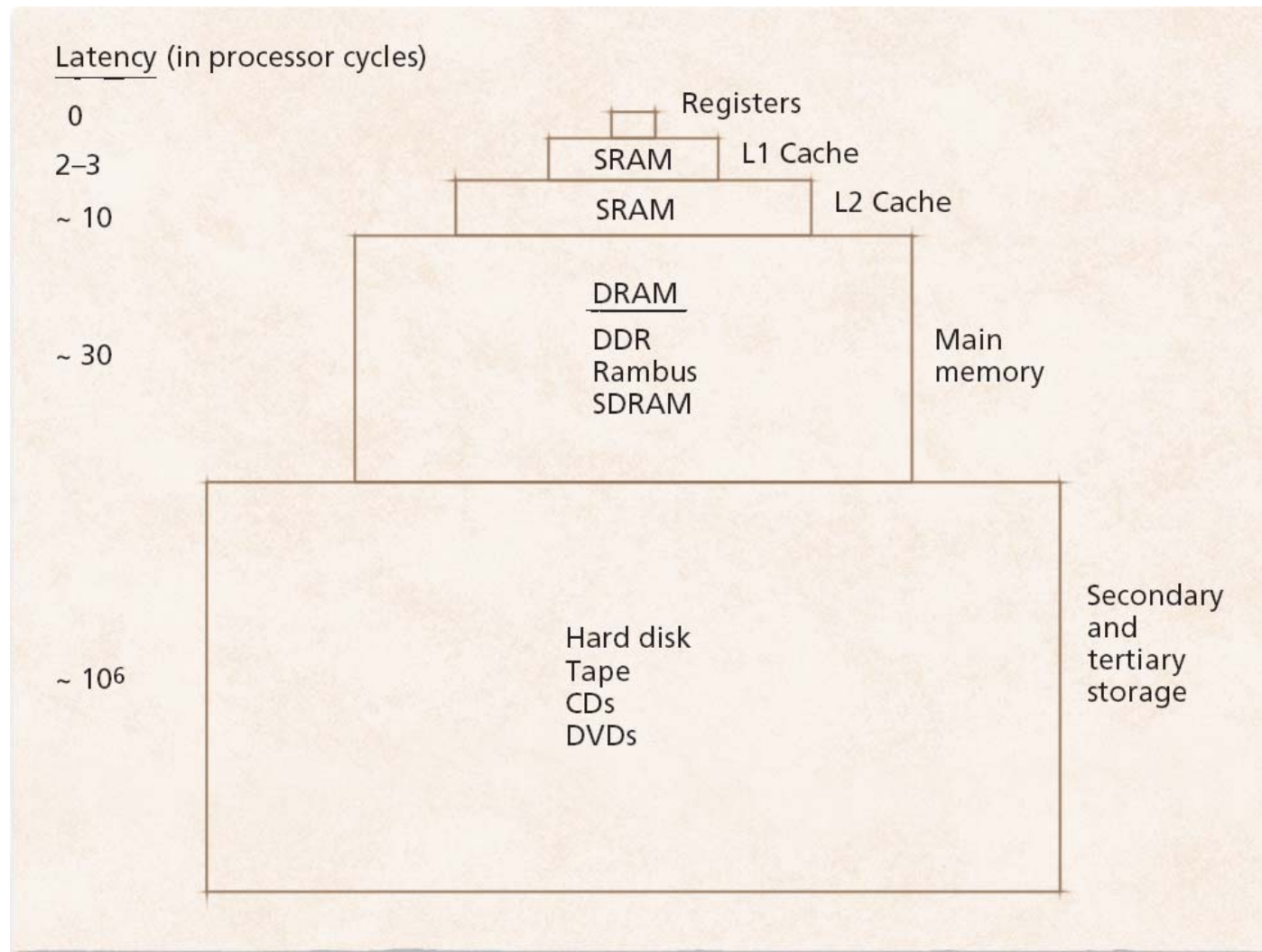


Gestão de memória secundária

- **Designa-se por memória secundária todos os dispositivos de armazenamento de memória que não estão ligados “directamente” ao processador**
- **Acesso (interface) ao Sistema de Ficheiros**
 - Conceito de Ficheiro:
 - Atributos, ex.: name, size, owner, protection, access, times and dates
 - Operações, ex: creat, write, read, position within a file, delete, read and change attributes
 - Conceito de Directoria:
 - seus atributos e operações
 - File Buffering, locking and sharing
- **Organização (estrutura) do Sistema de Ficheiros**
 - Noção de partição e volume
 - Estruturas de suporte a volumes, directorias e ficheiros
 - Gestão de espaço livre e ocupado no volume
 - Algoritmos de alocação de espaço livre no volume
 - Algoritmos de optimização de acesso ao disco (disc scheduling)



Hierarquia da memória



Gestão de *input/output*

As principais funções da gestão de Input/ Output (I / O) são:

- **Gerir os dispositivos de I / O**

- I / O Hardware (como se organiza)
- Controladores de I / O (device controller)
- Acoplamento entre os controladores e o SO: Device Drivers (Device Driver interface)
- Acesso básico a I / O: pooling, interrupts, DMA
- I / O Buffering, I / O scheduling

- **Gerir e controlar as operações de Input / Output**

- Acesso a dispositivos orientados ao bloco e ao caracter
- Acesso bloqueante e não bloqueante
- Acesso com e sem bufferização
- Spooling e Device reservation
- Tratamento de erros



Gestão de segurança

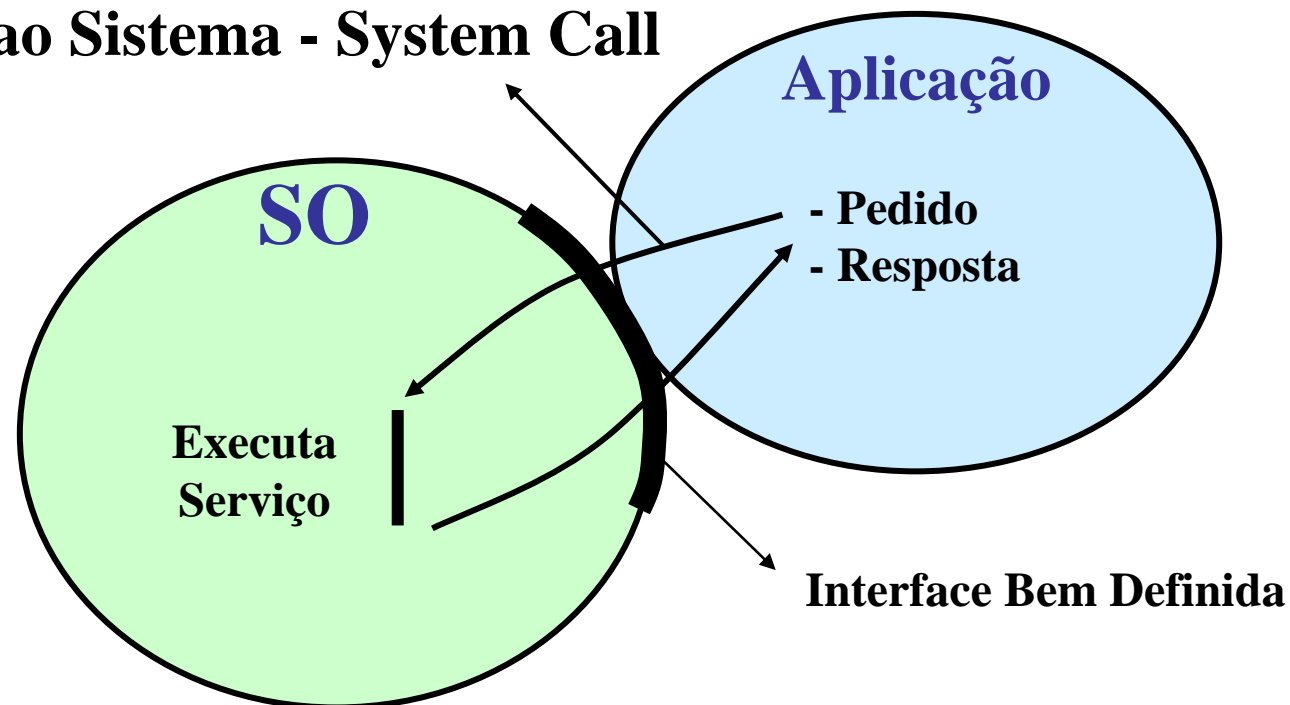
- **Controlar o acesso aos recursos do SO**, por exemplo:
 - CPU, memória, ficheiros, processos, modems, canais de I/O, impressoras e serviços
- **O sistema de protecção deve:**
 - Providenciar forma de autenticação do utilizador
 - Permitir definir permissões de acesso
 - O que é autorizado fazer
 - Quem o pode fazer
 - Distinguir acesso autorizado de não autorizado
- **Tópicos**
 - Permissões de acesso, domínios de protecção (grupos), listas de controle de acessos
 - Passwords, encriptação, stack overflow, viruses, denial of service, Intrusion detection, logging, ...



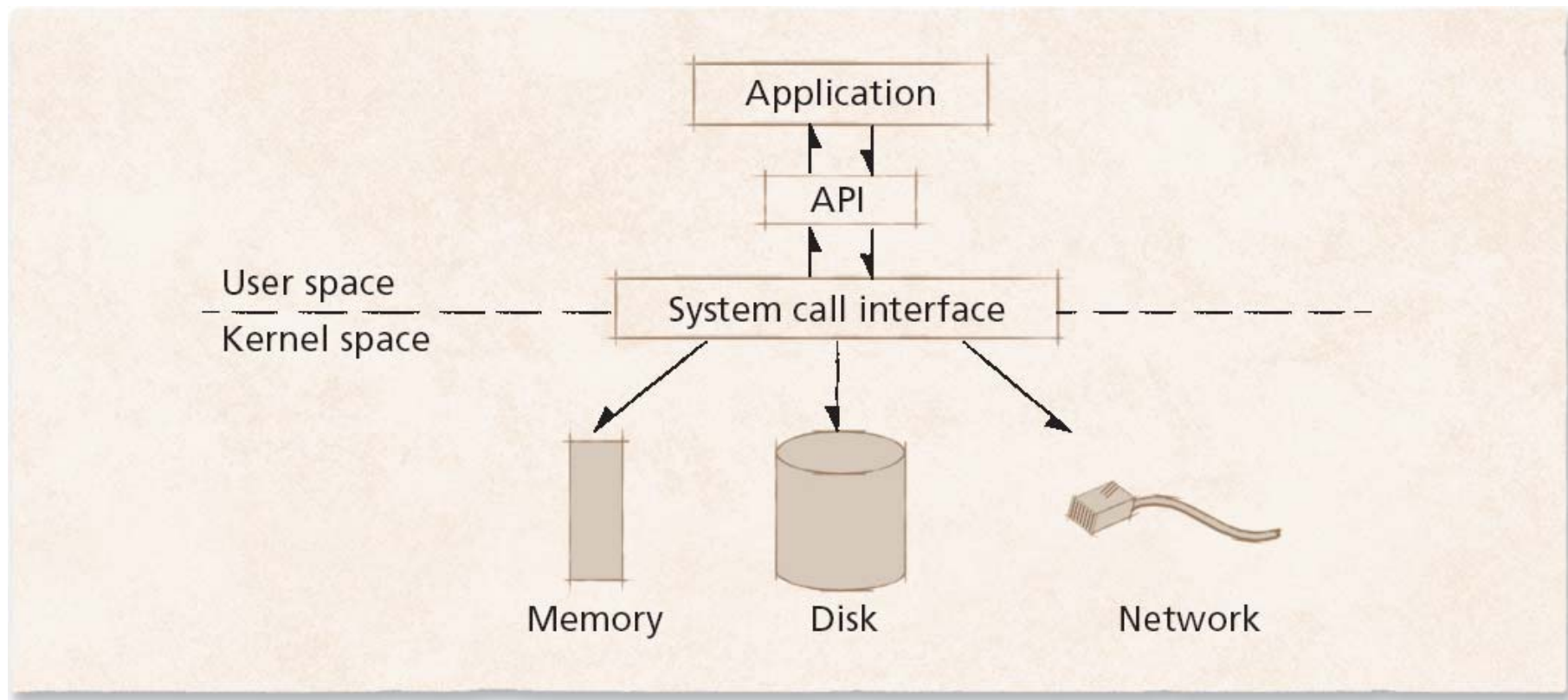
Aplicações e o Sistema Operativo

Os serviços são disponibilizados pelo sistema Operativo na forma de uma interface bem definida, em que cada evocação de um serviço se designa de **System Call (Chamada ao Sistema)**.

Chamada ao Sistema - System Call



Interface do S.O.



Serviços disponibilizados por um S.O. (alguns exemplos)

- **Execução de programas (processos)**
 - Criar, terminar, suspender, reactivar processos e tarefas
 - Criar, destruir e utilizar mecanismos de sincronismo
 - Criar, destruir e utilizar mecanismos de comunicação
- **Memória**
 - Alocar e libertar memória
 - Mapear memória em ficheiro
 - carregar e libertar dinamicamente DLL's
- **Sistemas de ficheiros**
 - Abrir, ler escrever, fechar, consultar e alterar atributos, de ficheiros



Chamada ao sistema / *System call*

Geralmente segue os seguintes passos:

- **Colocação dos parâmetros nos devidos locais**
 - Parâmetros nos registos
 - Parâmetros num tabela, sendo passado o seu endereço
 - Parâmetros em *stack*
- **Execução de um *interrupt* especial (*trap*),**
 - que indexa ao ponto de entrada do SO na tabela de *interrupts*, e
 - que muda o modo do processador para modo *kernel*
- **O SO verifica se os parâmetros estão correctos**
- **Efectua a chamada à rotina correspondente ao serviço requisitado**
- **Após a execução do serviço solicitado, passa para modo utilizador, e salta para a instrução seguinte à evocação do *interrupt* (*trap*)**

Normalmente as aplicações não fazem uso directo das chamadas ao sistema, pois usam bibliotecas que disponibilizam esses mesmos serviços, geralmente com mais funcionalidade, na linguagem de programação em que a aplicação é desenvolvida



Bibliotecas e chamadas ao sistema

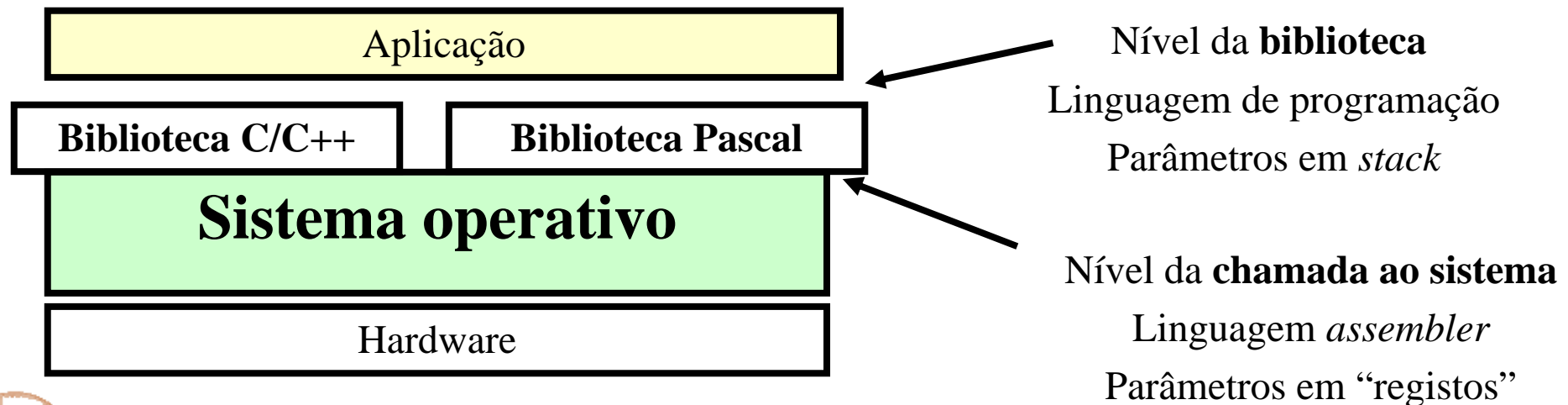
Normalmente as aplicações não fazem uso directo das chamadas ao sistema, pois usam bibliotecas que disponibilizam esses mesmos serviços, geralmente com mais funcionalidade, na linguagem de programação em que a aplicação é desenvolvida

Uma chamada ao sistema (*system call*) :

em Linguagem C: `write(buff, size, file)`

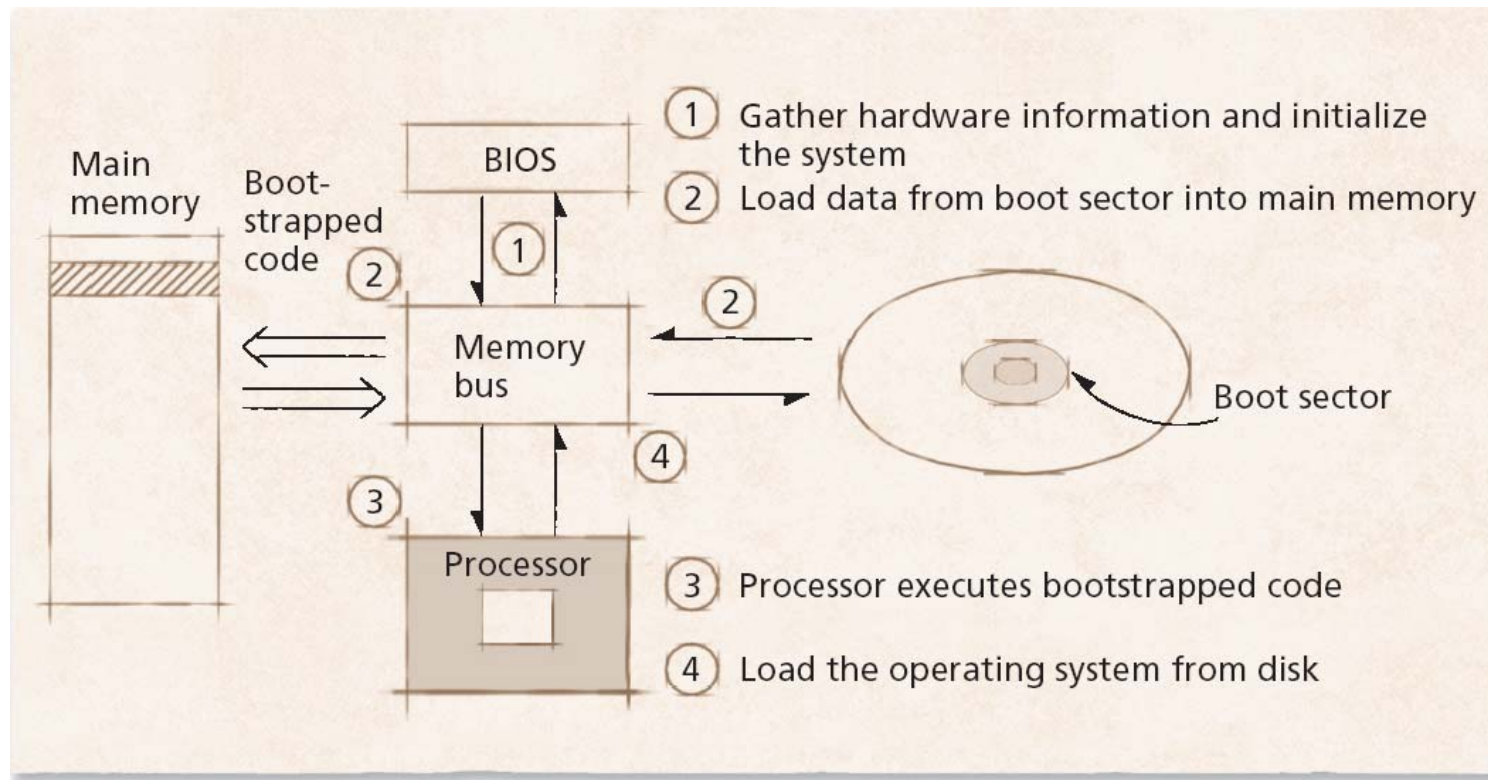
no MSDOS : `int 21h, serviço..., parâmetros...`

no Windows : `int 2eh, ...`



Bootstrapping – Arranque do S.O.

- **BootStrapping – Carregar inicialmente os componentes do S.O. Para memória**
 - Executado pela *Basic Input/Output System* (BIOS)
 - Inicia o Hardware
 - Carrega o código de arranque (do S.O.) do Boot sector para memória
 - Esse código providencia o carregamento da restante parte do S.O.



Plug and Play

■ Tecnologia *Plug and Play*

- Permite que o S.O. configure *hardware* novo, automaticamente e sem interacção com o utilizador
- Para suportar *plug and play* os dispositivos de *hardware* têm de:
 - Identificar-se univocamente ao S.O.
 - Comunicar com o S.O. para indicar os recursos e serviços que necessitam para funcionar correctamente
 - Identificar o *driver* que suporta o dispositivo e permitir que o software configure o dispositivo (ex: atribuir o dispositivo a um canal de DMA)



Requisitos de *Hardware* dos S.O.

O sistema operativo deve assegurar a correcta operação do sistema computacional garantindo a sua integridade e a dos processos. Para tal necessita das seguintes características disponibilizadas pelo hardware:

- **Dois modos de operação (*kernel/user*)**
- **Instruções privilegiadas**
- **Protecção de memória**
- ***Interrupts***



Protecção do S.O.

- **A execução concorrente de programas requer que o sistema operativo garanta que a execução de um programa não afecte o sistema operativo nem qualquer outro programa.**
- **Restrições à execução de um programa:**
 - Acesso indiscriminado à memória
 - Acesso directo ao I/O
 - Acesso ao estado do processador (que é independente do processo, ex. DI)
 - Uso indiscriminado do CPU (ex. ciclo infinito)



Protecção *Hardware* do S.O.

- **Para atingir os objectivos enunciados os sistemas operativos utilizam protecções disponibilizadas pelo hardware:**
 - Existência de dois (ou mais) modos de funcionamento do CPU:
 - Modo sistema / kernel / monitor / supervisor / privilegiado
 - Modo em que o sistema operativo executa o seu código
 - Modo sem restrições
 - Modo utilizador / user
 - Modo em que os programas do utilizador são executados
 - Modo em que as restrições estão activas
 - Existência de instruções (privilegiadas) que só podem ser executadas no modo kernel – ex. DI, IN, OUT,
 - Uso do *interrupt* do *timer* para garantir que o SO adquire o controle do CPU
 - Protecção de acesso à memória



Protecção da memória

- **O sistema operativo não deverá permitir que um processo aceda:**
 - À tabela de *interrupts*
 - À memória com código do sistema
 - A outra memória que não seja à sua devidamente “alocada”

- **Para tal deve existir um esquema de protecção disponibilizado pelo *hardware*:**
 - baseado em espaços de endereçamento lógicos
 - Cada endereço lógico tem de ser menor que o valor do registo limite
 - A cada endereço lógico válido é adicionado um valor base resultando no endereço físico de acesso à memória
 - Um acesso fora deste espaço de endereçamento resulta numa interrupção do sistema operativo que o interpreta como um erro fatal



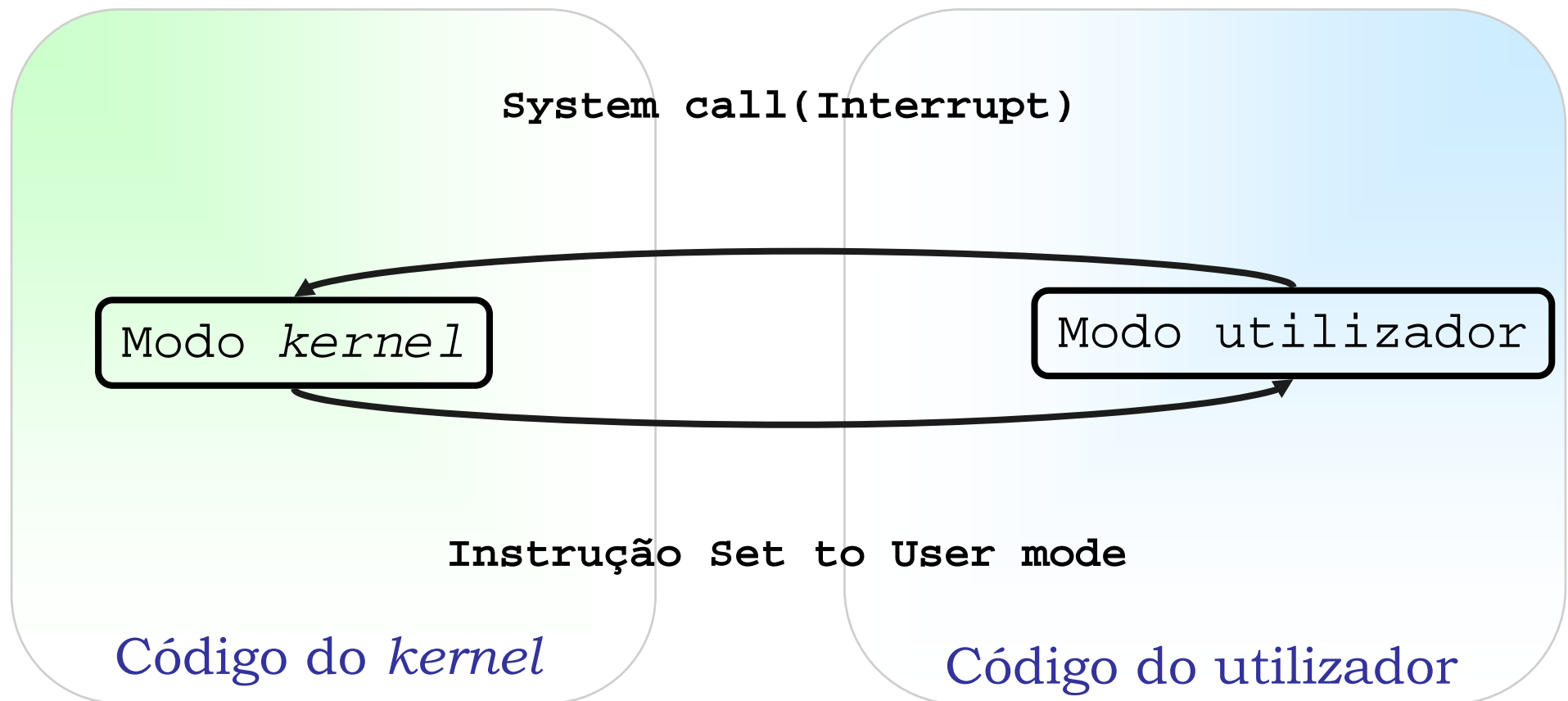
Acesso ao hardware

- **O SO controla o acesso ao *hardware* de forma a que os programas se executem correctamente**
 - Controla todo o acesso ao *hardware*, de forma a proporcionar uma utilização segura e eficiente.

- **Como acedem as aplicações ao *hardware* ?**
 - As aplicações acedem aos recursos de hardware invocando um serviço do “sistema operativo” por intermédio de uma chamada ao sistema (*system call*).
 - O código do serviço é parte integrante do Sistema operativo e executa-se em modo *Kernel*



Mudança de Modo de Funcionamento



Chamada ao sistema / *System call* e *modo de funcionamento*

- **Chamada ao sistema / *system call*** - é uma invocação de um serviço do sistema operativo
- **Geralmente segue os seguintes passos:**
 - Execução de um *interrupt* especial (*trap*), que indexa ao ponto de entrada do SO na tabela de *interrupts*, e que muda o modo para ***kernel***
 - O SO verifica se os parâmetros estão correctos
 - Efectua a chamada à rotina correspondente ao serviço requisitado
 - Após a execução do serviço solicitado, passa para modo **utilizador**, e salta para a instrução seguinte à evocação do *interrupt*

