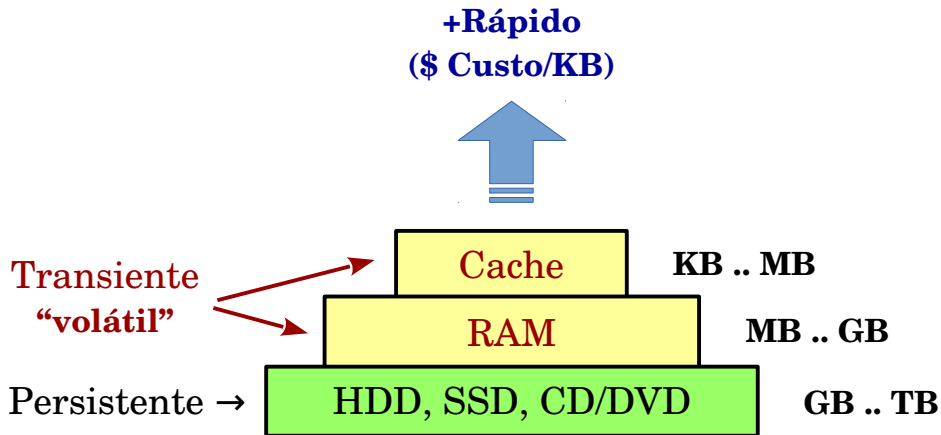
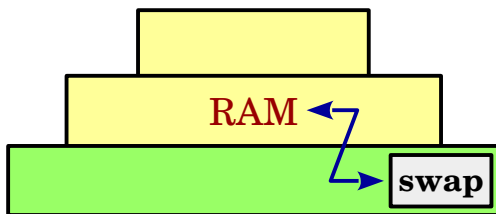


Hierarquia de Memórias



É função do SO abstrair a hierarquia de memórias em um modelo computacional mais simples e gerenciar esse modelo: gerenciador de memória...

Processos (programas em execução) não precisam, **necessariamente**, permanecer “por completo” o tempo todo na memória principal.



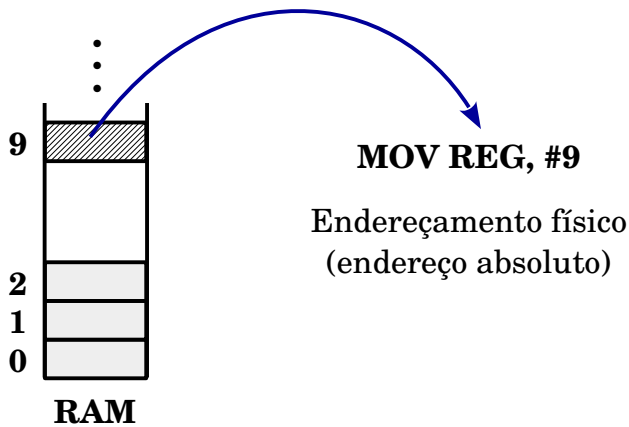
Swap...

Área de troca que recebe segmentos da memória principal (RAM) “temporariamente” fora de uso.

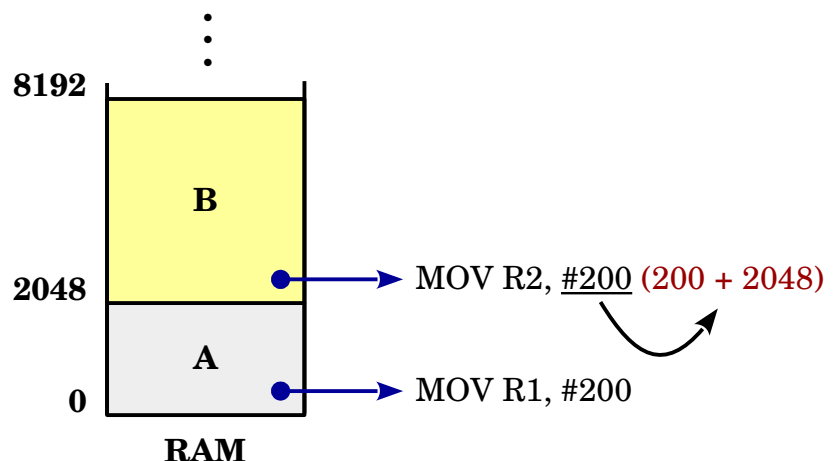
O gerenciador de memória do SO controla os segmentos de memória em uso e lida com a alocação e liberação de memória para os processos.

Gerenciamento de Memória

1 – Sem abstrações



- Processos “enxergam” a memória RAM física, ou seja, o endereçamento é feito de forma direta (endereço absoluto).
- Apenas um processo por vez, embora seja possível alocar processos consecutivamente na memória e **“realocar estaticamente”** os endereçamentos feitos no momento da carga do programa.

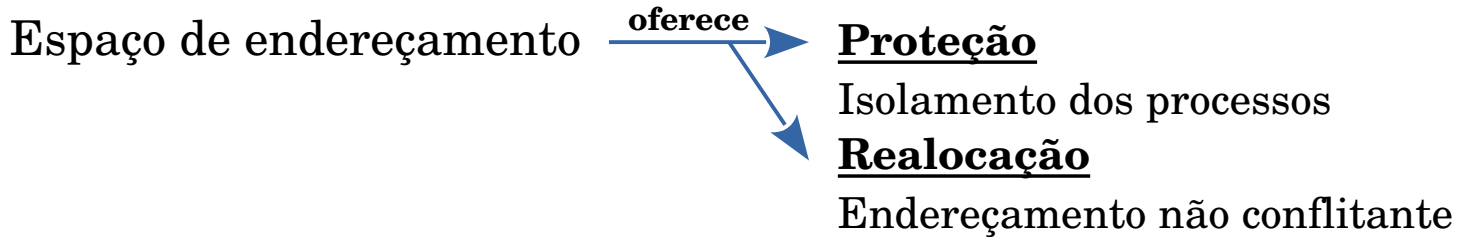


Problemas...

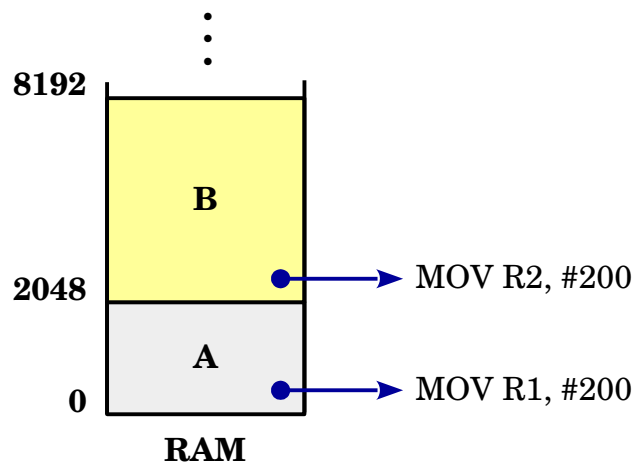
- Processos podem, ainda que involuntariamente, acessar regiões de memória não permitidas, a não ser que haja um mecanismo de proteção por hardware.
- Difícil manter mais de um programa em execução.

2 – Espaços de endereçamento

Permitem que diversos programas estejam em execução ao mesmo tempo sem que um interfira com o outro.

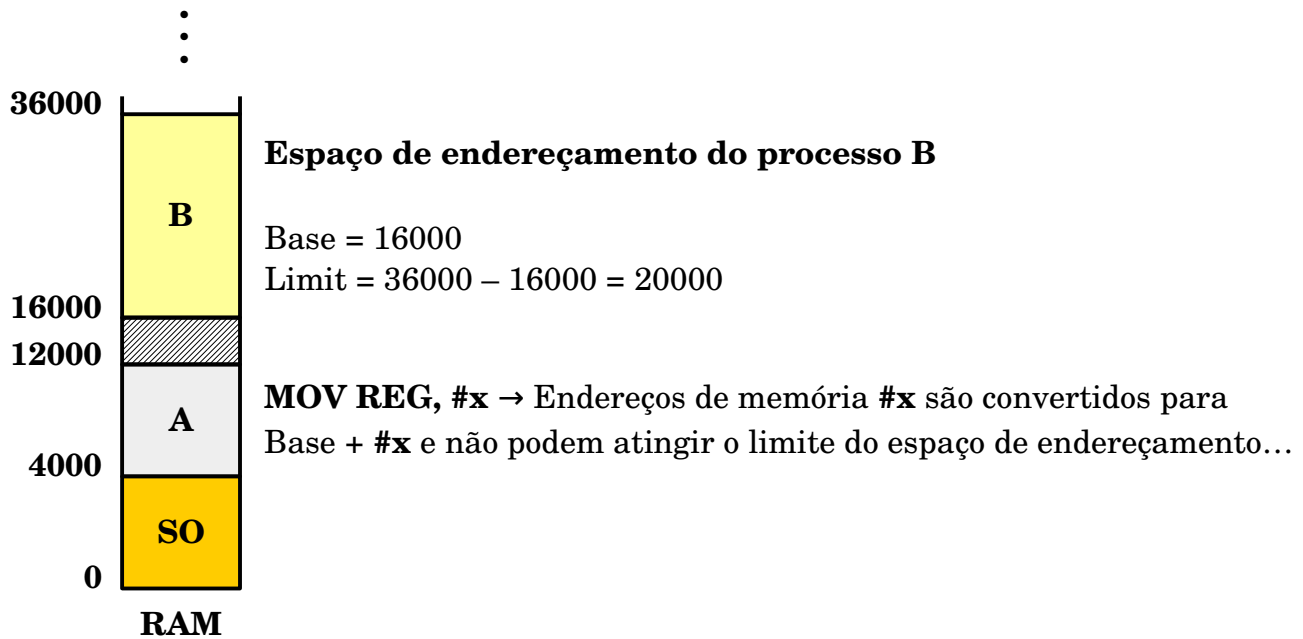


O espaço de endereçamento é um conjunto de endereços utilizado por um processo para hospedar código, dados e pilha. Cada processo possui seu próprio espaço de endereçamento.



O endereço lógico #200 no processo A e o endereço lógico #200 no processo B são mapeados, respectivamente, para os endereços físicos #200 e #2248 ($200 + 2048$) somente durante a execução e antes de qualquer acesso a memória RAM...

Realocação dinâmica → Registradores **Base** e **Limit**



Problema: Cada programa em execução possui seu próprio espaço de endereçamento, mas este deve ser contíguo e pode não haver memória suficiente para todos os processos.

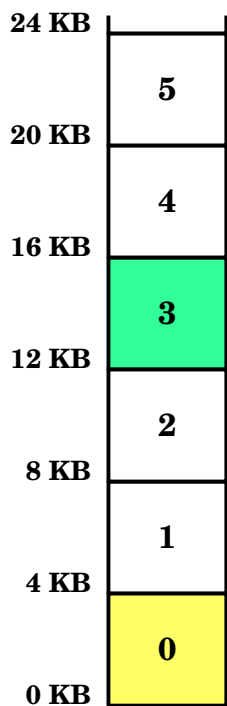
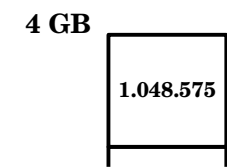
Solução...

- **Swapping:** Consiste em liberar o espaço de memória alocado para o processo “descarregando” o segmento em disco e carregando novamente quando for necessário.
- **Memória virtual:** Permite que programas ocupem áreas não-contíguas da memória e possam ser executados mesmo quando estão apenas parcialmente na memória RAM.

3 – Memória virtual

Espaço de endereçamento lógico dividido em unidades de alocação de tamanho fixo (páginas) que são mapeadas sob demanda na memória física.

Espaço de endereçamento virtual (32 bits)

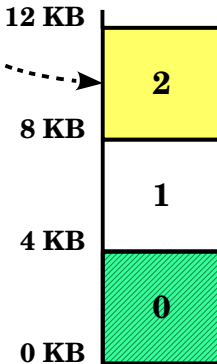
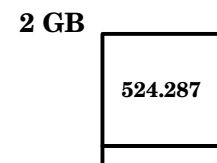


↑
Páginas

Pág.	Bits		
	Present	Dirty	Frame
0	1	0	2
1	0	0	-
2	0	0	-
3	1	1	0
⋮	⋮	⋮	⋮
1.048.575	0	0	-

Tabela de Paginação – MMU
“Processo A”

Espaço de endereçamento físico (RAM)



↑
Frames

Endereço virtual...

0000.0000 – 0000.0000 – 0000.0000 – 0000.0000 (32 bits)

└─ $2^{20} = 1.048.576$ páginas ─┘ └─ $2^{12} = 4$ KB ─┘

Exemplos...

MOV REG, #0 → **Página 0 presente** → Frame 2

MMU insere no barramento o endereço #8192 (8192 + 0)

MOV REG, #100 → **Página 0 presente** → Frame 2

MMU insere no barramento o endereço #8292 (8192 + 100)

MOV REG, #12290 → **Página 3 presente** → Frame 0

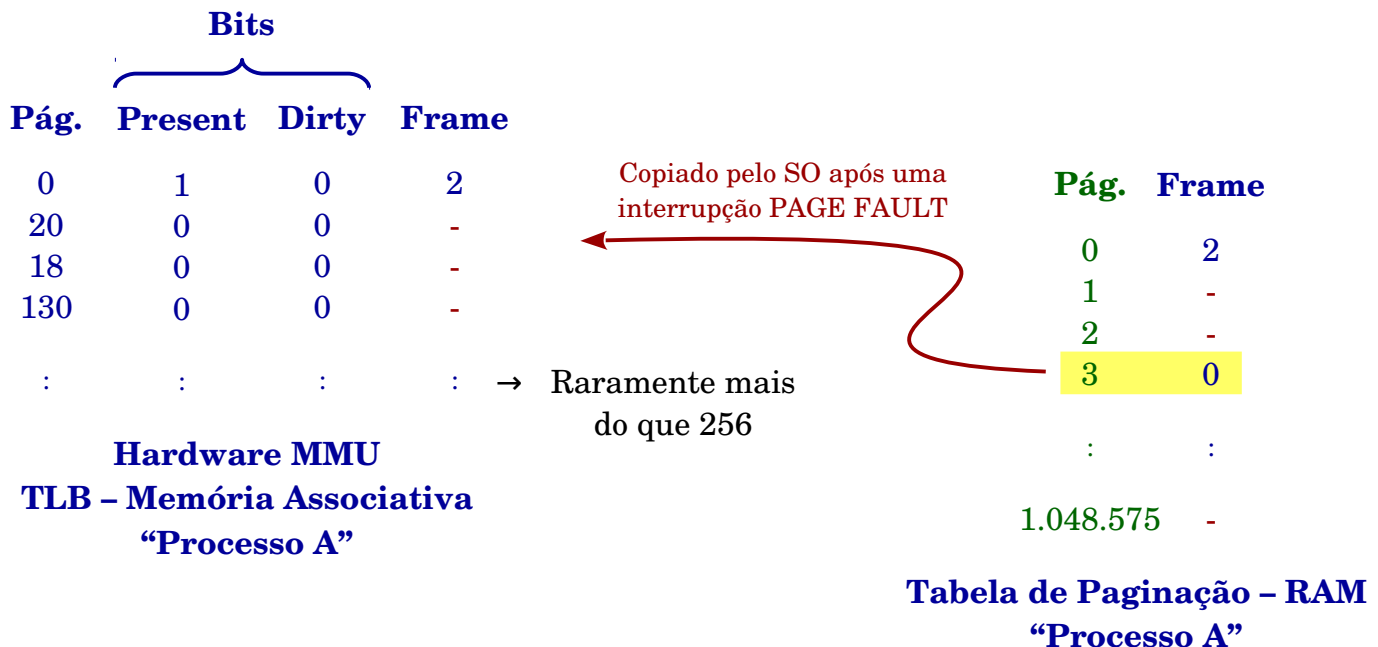
MMU insere no barramento o endereço #2 (0 + 2)

MOV REG, #5000 → **Página 1 ausente**

MMU lança uma interrupção **PAGE FAULT**

Problema: Tabela de paginação demasiadamente grande para ser mantida na MMU (Memory Management Unit).

Solução: Translation Lookaside Buffer – TLB.



Problema: Tabela de paginação ainda demasiadamente grande para ser mantida na memória RAM.

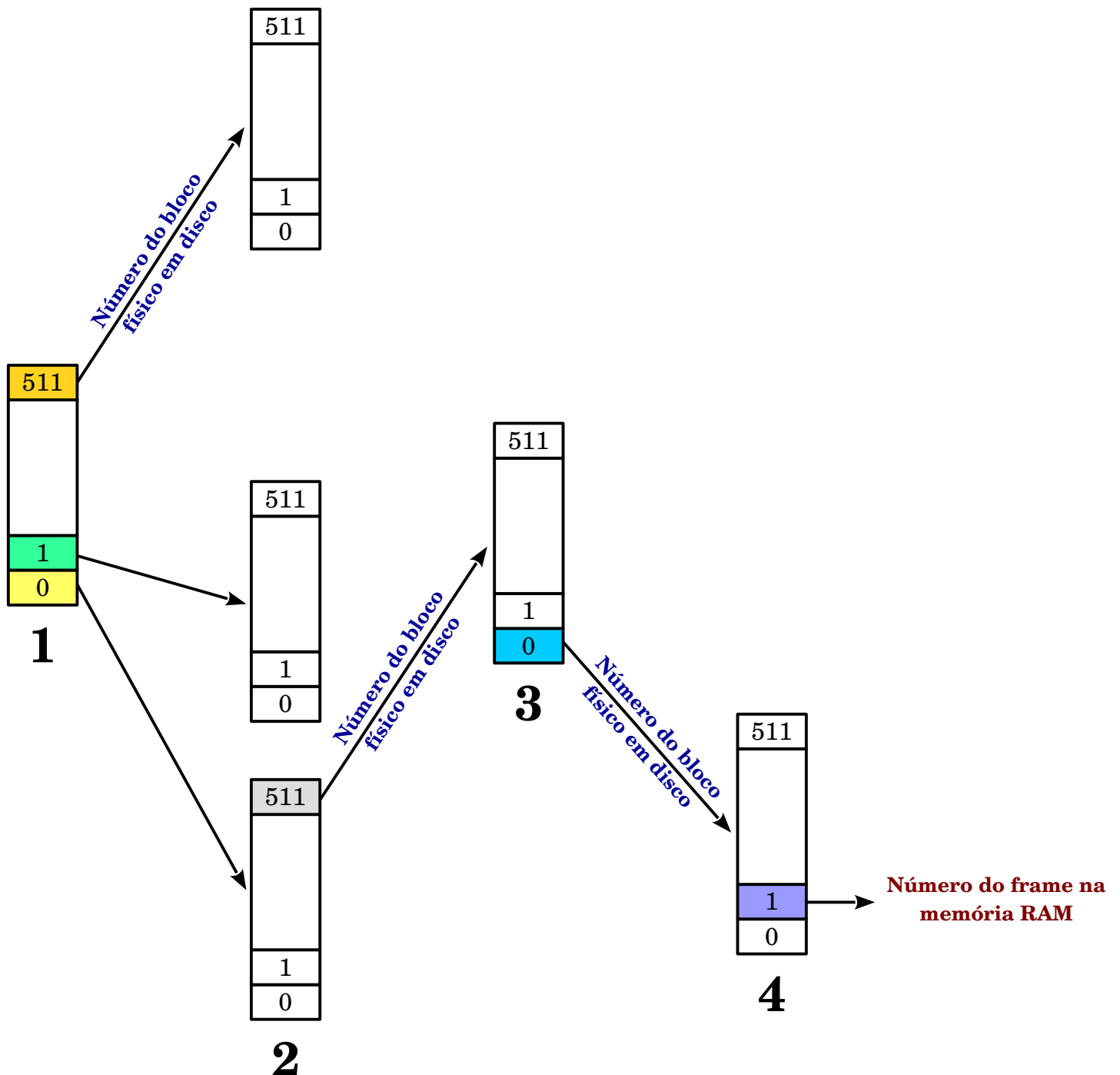
Solução: Paginação multinível.

Paginação multinível

Endereço virtual...

$$\frac{9 \text{ bits}}{512} \cdot \frac{9 \text{ bits}}{512} \cdot \frac{9 \text{ bits}}{512} \cdot \frac{9 \text{ bits}}{512} \cdot \frac{12 \text{ bits}}{4096} = 48 \text{ bits} = 256 \text{ TB}$$

┌───────────┐
 512 512 512 512 4096
 └───────────┘
 Paginação de 4 níveis
 2³⁶ páginas de 4 KB
 (4 KB)



Algoritmos de substituição de páginas...

Not Recently Used – Não Usada Recentemente

Remove uma página qualquer (modificada ou não) que não tenha sido acessada desde o último ciclo de clock (tick) → tipicamente 20ms.

Least Recently Used – Usada Menos Recentemente

Remove a página mais antiga considerando o instante de seu último acesso.

First-In, First-Out (FIFO) – Primeira a Entrar, Primeira a Sair

Remove a página mais antiga a entrar na memória RAM, mesmo que altamente requisitada.

Second Chance – Segunda Chance

Combina a técnica FIFO com a técnica NRU de modo a não remover da memória uma página antiga mas ainda muito requisitada.

Outros...

Aging (Envelhecimento), Working-Set (Conjunto de Trabalho), Clock (Relógio), WSClock (Working-Set Clock).

4 – Segmentação com paginação

Oferece um espaço de endereçamento bidimensional com os benefícios da paginação.

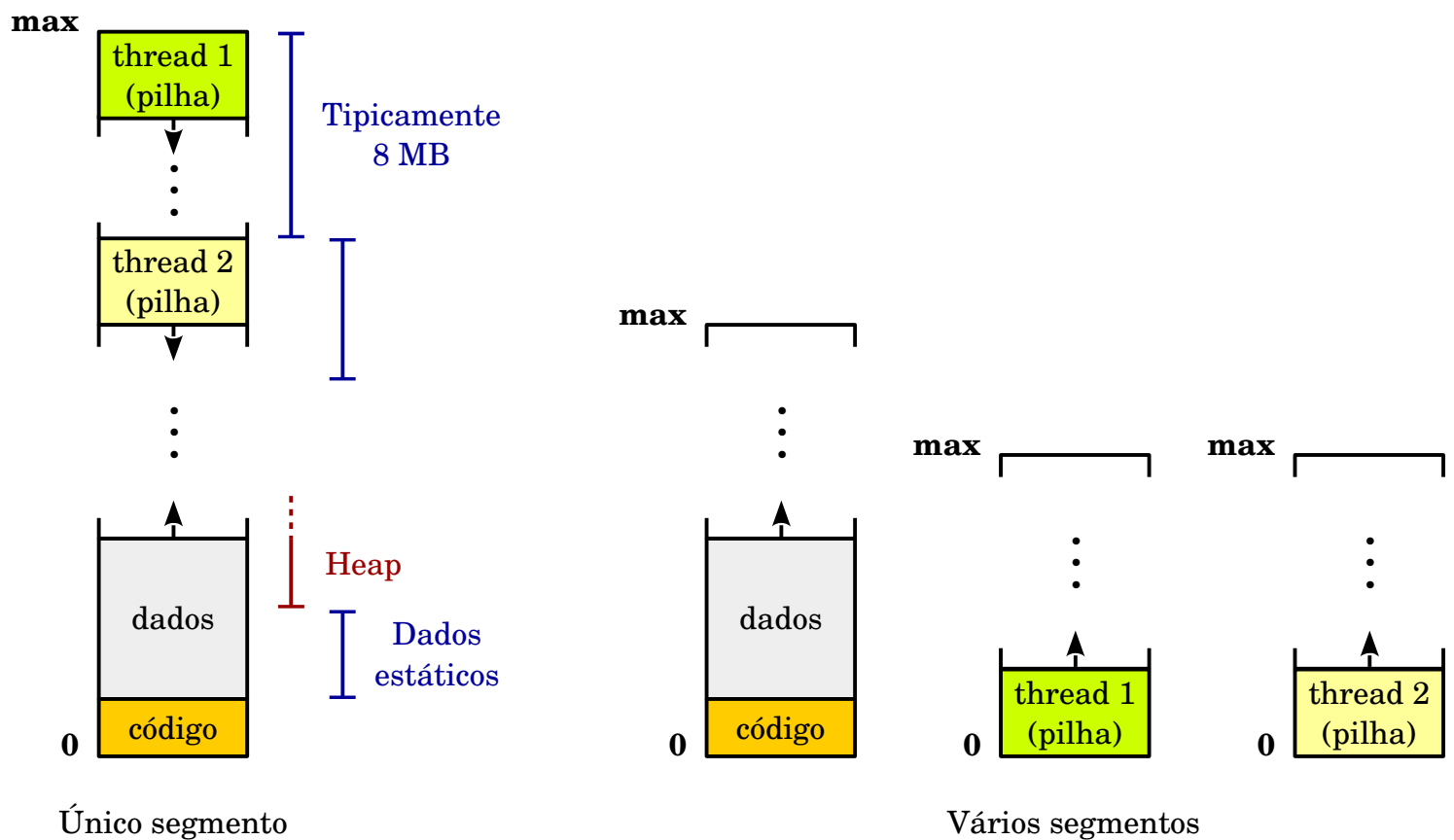
Espaço unidimensional

endereço = <offset>

Espaço bidimensional

endereço = <segmento, offset>

Nota → UNIX/Linux e Windows não utilizam



Intel x86

32 bits: 16384 segmentos de até 4 GB.

64 bits: Segmentação obsoleta → Suporte removido em razão dos principais sistemas operacionais não utilizarem.