

# Memória Virtual

Prof. Gustavo Girão

# Introdução

- Uma aplicação, quando executada, deve estar na memória principal
  - ✧ Se existirem várias aplicações executando, todas devem estar na memória principal
- O SO também deve estar na memória principal
- Aplicação: instruções + dados

# COMO GERENCIAR A MEMÓRIA PRINCIPAL DE FORMA A COLOCAR TODAS AS APLICAÇÕES E SO NA MEMÓRIA?



No Windows: 440 MBytes



2013 - No Windows: 3 GBytes



No Windows: 4 MBytes

# Princípio I

## a memória precisa ser particionada

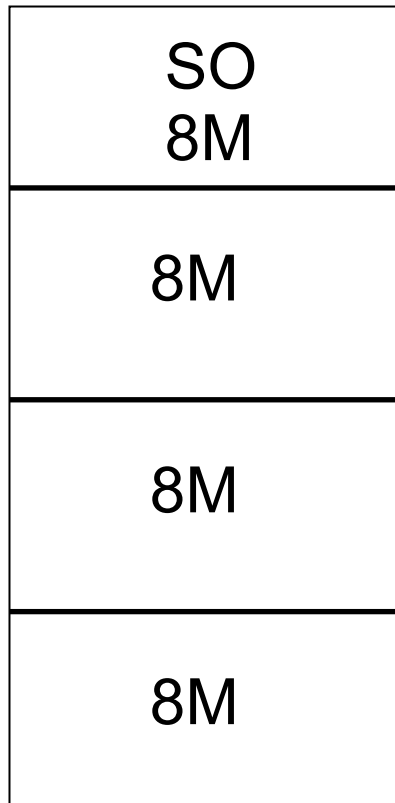
# Particionamento

- Dividir a memória principal em partições
  - "pedaços" da memória que podem ser gerenciados isoladamente.
- Cada partição irá armazenar parte do conteúdo de uma aplicação ou SO

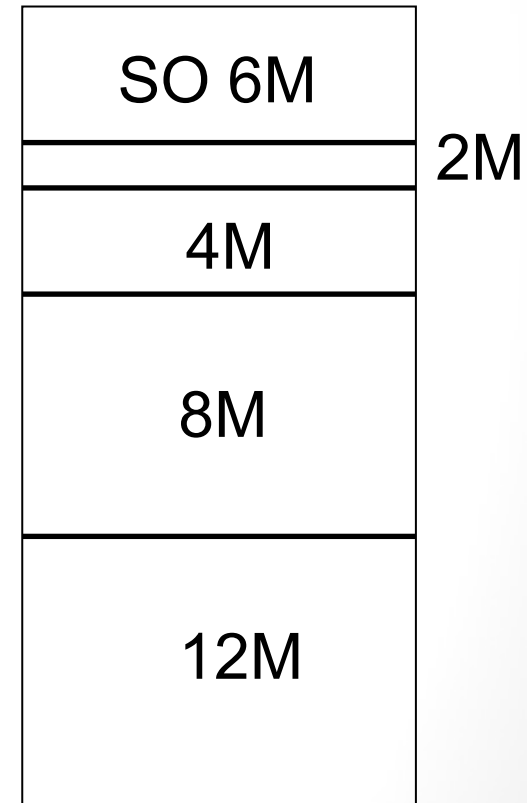
# Particionamento

- **Partições de tamanho fixo**

- Todas as partições têm tamanho fixo, mas não precisam ser do mesmo tamanho



Partições de mesmo tamanho

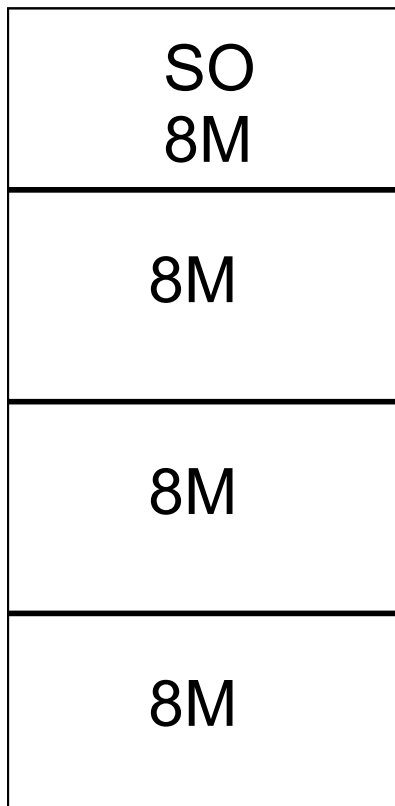


Partições de tamanhos diferentes

# Particionamento

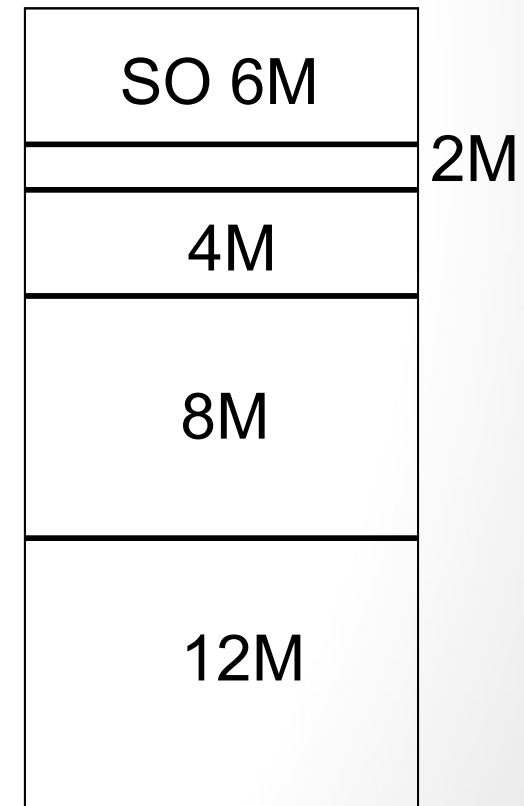
- **Partições de tamanho fixo**

- Todas as partições têm tamanho fixo, mas não precisam ser do mesmo tamanho



Na maior parte dos casos, a parte da aplicação não exigirá tanta memória quanto a fornecida pela partição

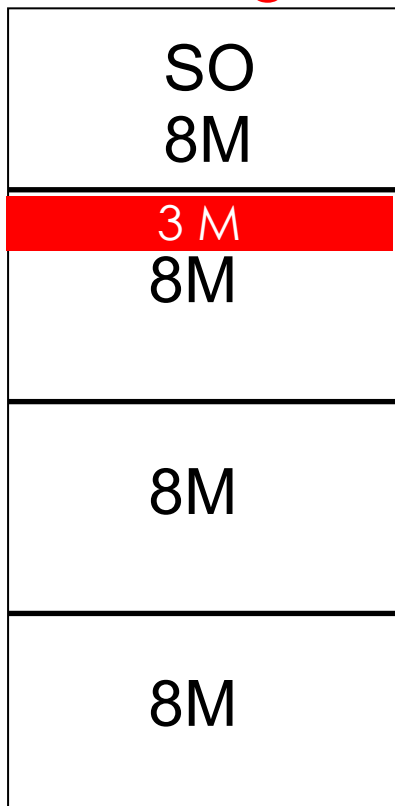
Ex.: 3MB será colocado em 4MB



# Particionamento

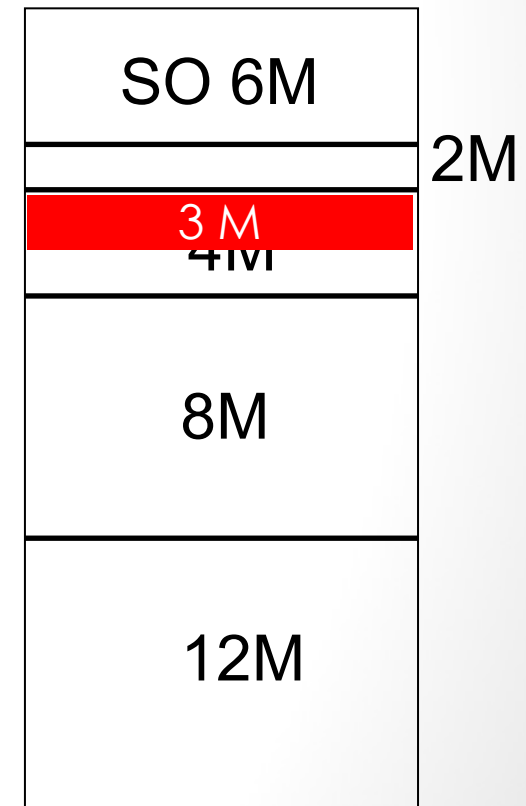
- Partições de tamanho fixo

- Todas as partições têm tamanho fixo, mas não precisam ser do mesmo tamanho. Pode acarretar em **Fragmentação Interna**



Na maior parte dos casos, a parte da aplicação não exigirá tanta memória quanto a fornecida pela partição

Ex.: 3MB será colocado em 4MB



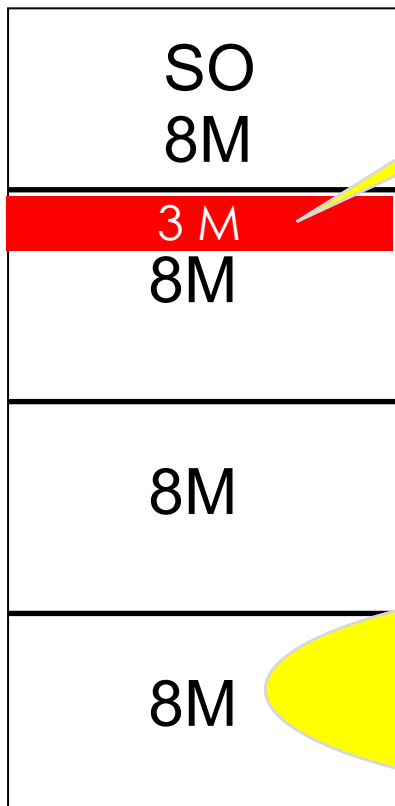


# Particionamento

- **Partições de tamanho fixo**

- Todas as partições têm tamanho fixo, mas não precisam ser do mesmo tamanho

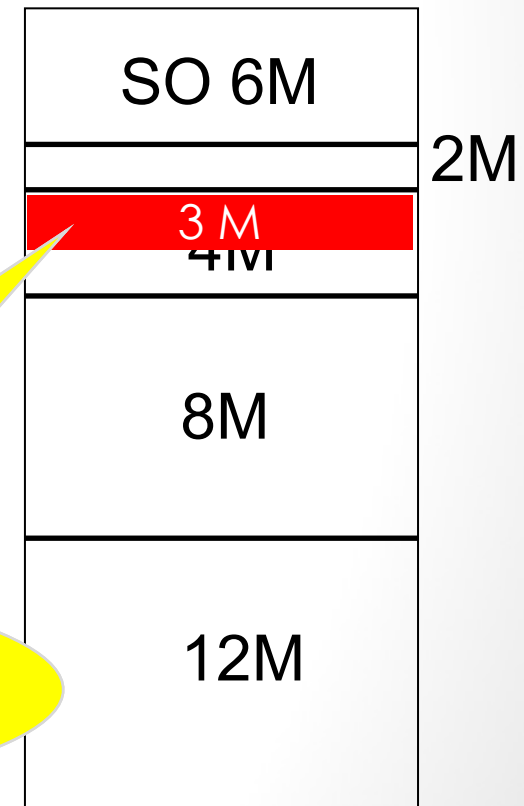
Uso de apenas 3M em uma partição de 8M



Na maior parte dos casos, a parte da aplicação não exigirá tanta memória quanto a fornecida pela partição

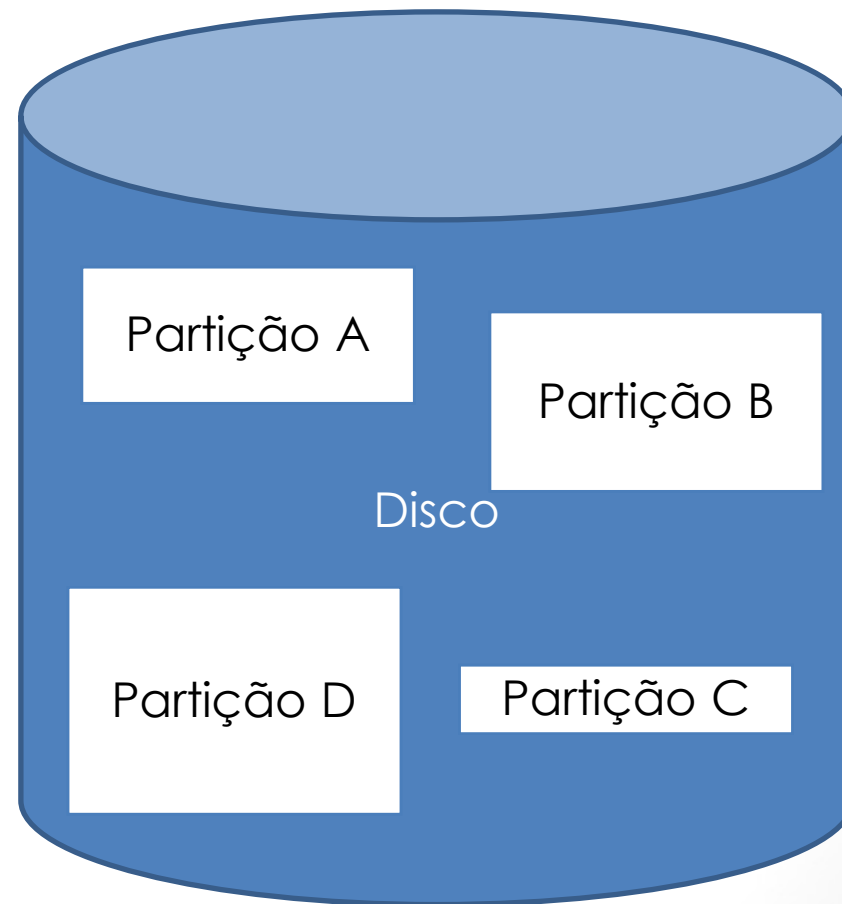
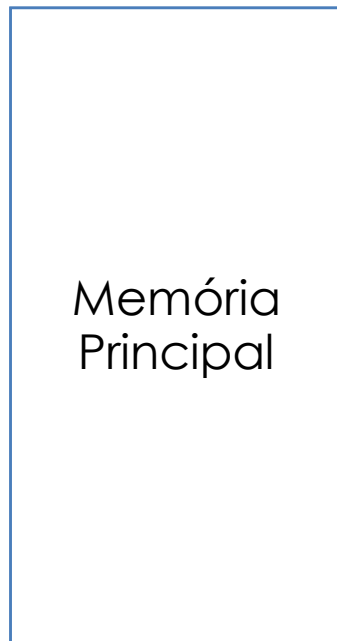
Ex.: 3MB será colocado em 4M

Uso de apenas 3M em uma partição de 4M



# Particionamento

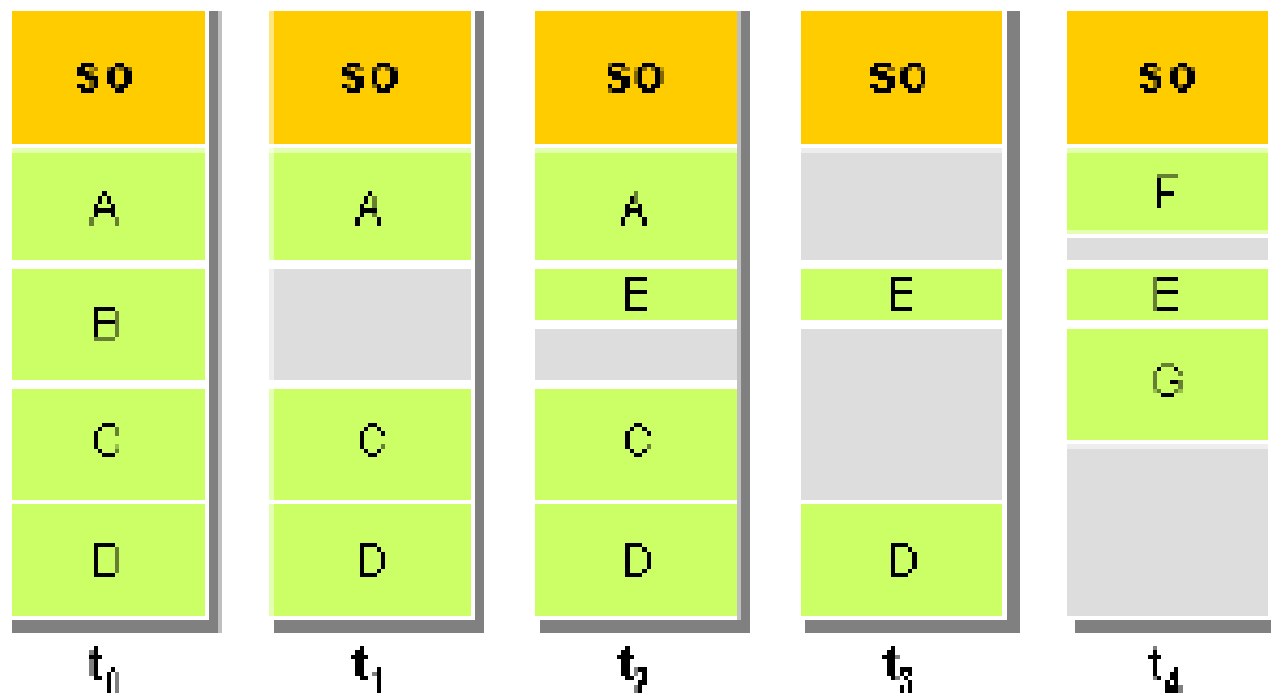
- **Partições de tamanho variável**
  - Dividir a memória principal em partições
  - Cada partição armazena parte do conteúdo de uma aplicação ou SO



# Particionamento

- Partições de tamanho variável

- Quando parte de uma aplicação é levada para a memória, ela recebe somente a quantidade de memória exigida. Pode acarretar em **Fragmentação Externa**



# Particionamento

- **Partições de tamanho variável**

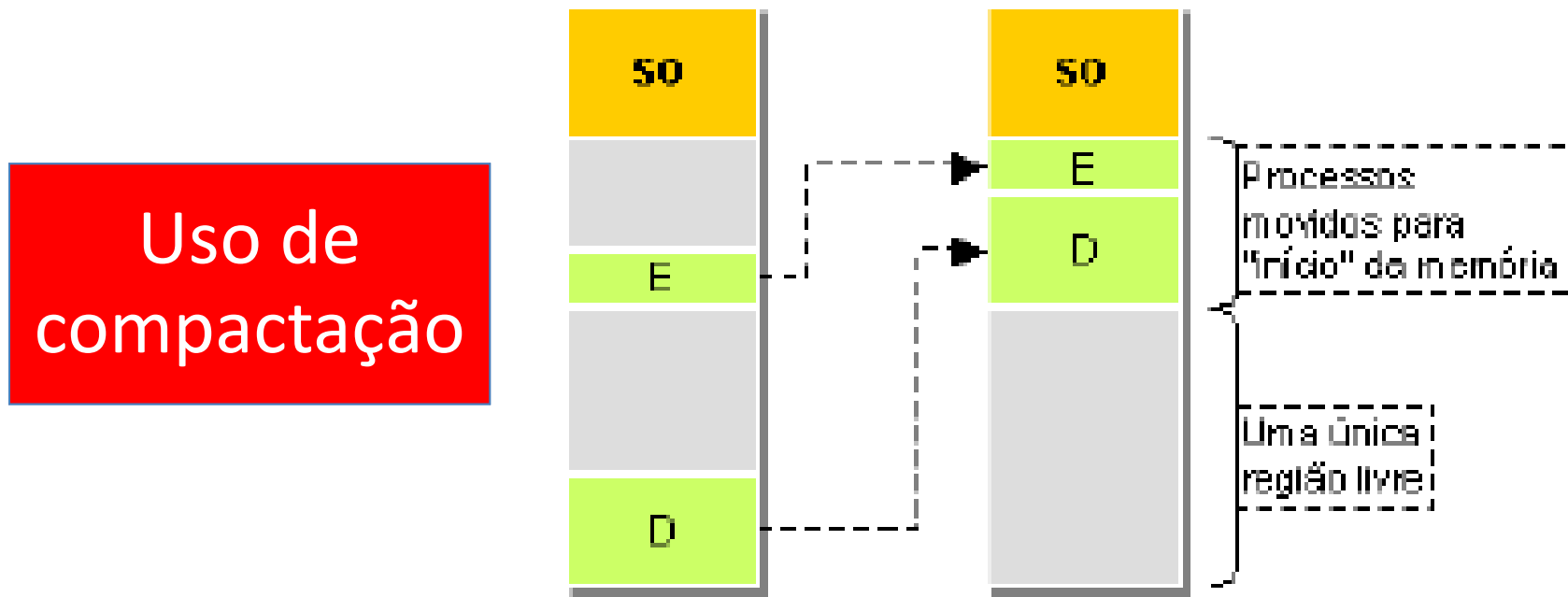
- Quando parte de uma aplicação é levada para a memória, ela recebe somente a quantidade de memória exigida



# Particionamento

- **Partições de tamanho variável**

- Quando parte de uma aplicação é levada para a memória, ela recebe somente a quantidade de memória exigida

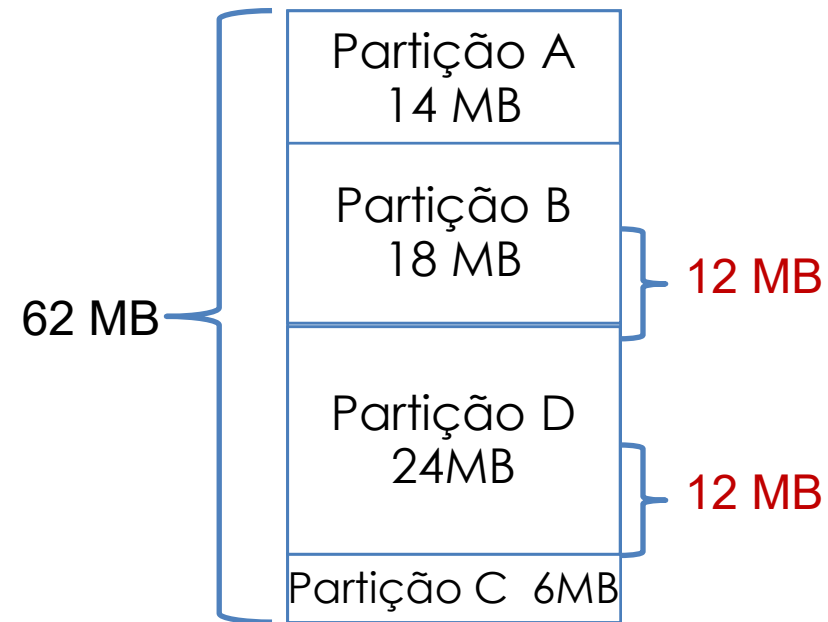


# Fragmentação

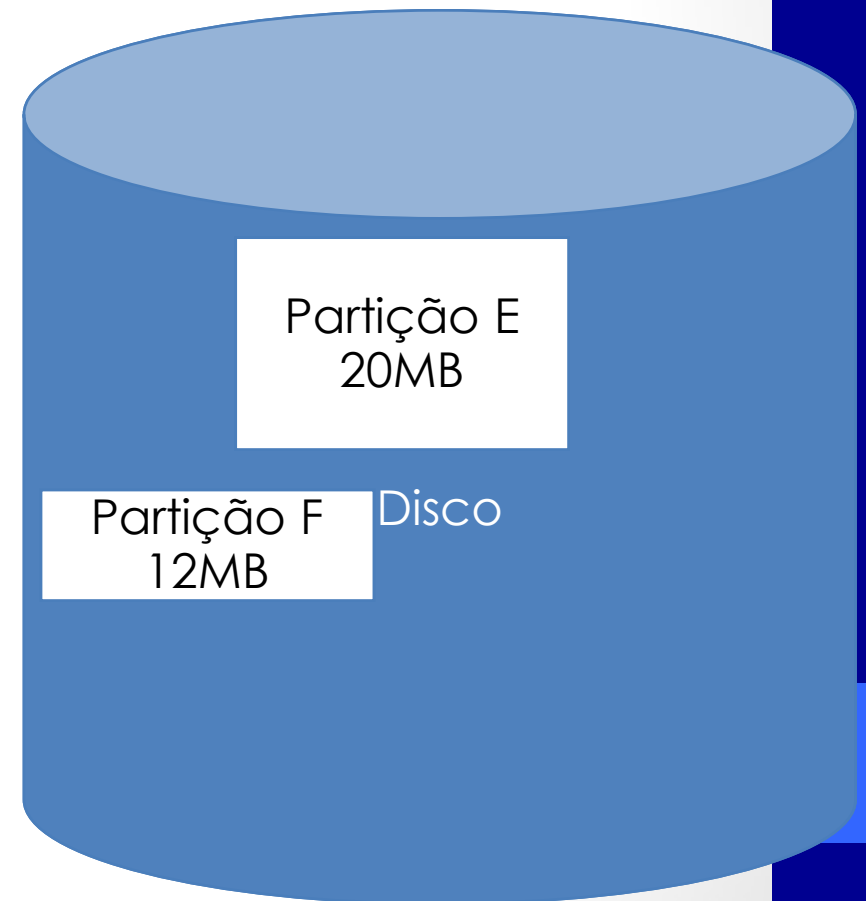
- Fragmentação

- A movimentação

**Uso de compactação** espaço não utilizável



Espaço disponível: 24MB



# Particionamento

- Não é possível garantir que um mesmo programa sempre estará na mesma posição da memória sempre que for executado
- Assim, não dá para usar o endereço físico da memória como referência ao programa, pois, em um outro momento, o programa pode estar em outro endereço

**COMO ENCONTRAR OS  
ENDEREÇOS SE A APLICAÇÃO  
NÃO ESTÁ EM UMA POSIÇÃO  
FIXA DA MEMÓRIA?**



# Princípio II

## USO DE ENDEREÇOS LÓGICOS PARA REFERENCIAR A MEMÓRIA

# Endereço Lógico e Físico

- **Endereço lógico**
  - Local relativo ao início do programa
- **Endereço físico**
  - Local real na memória principal
- **Existe uma conversão de endereço lógico em um físico.**

Princípio III

PARTICIONAR A MEMÓRIA DE  
MODO A EVITAR O DESPERDÍCIO

# Paginação

- Partições desiguais de tamanho fixo e de tamanho variável são inefícazes
- Solução:
  - Dividir a memória em pedaços menores de tamanho fixo – **Frames**
  - Dividir a aplicação em pedaços de mesmo tamanho dos pedaços da memória – **Páginas**
  - No máximo, o espaço desperdiçado na memória para esse processo é uma fração da última página

# Paginação

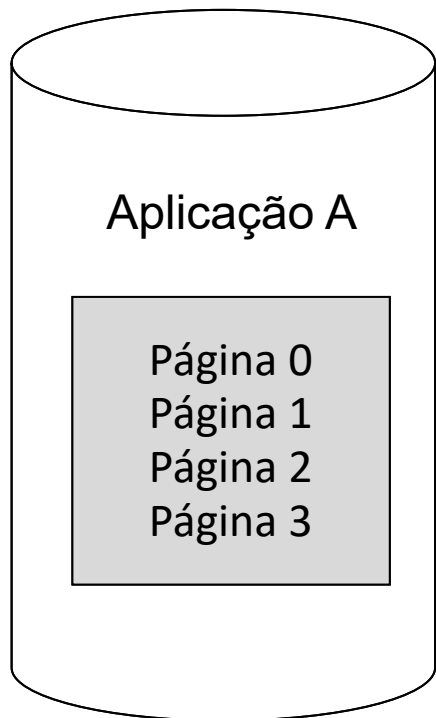


Tabela de página da Aplicação A

18
13
14
15

13	Página 1 de A
14	Página 2 de A
15	Página 3 de A
16	Em uso
17	Em uso
18	Página 0 de A
19	Em uso

# Paginação por Demanda

- Cada página de uma aplicação é trazida apenas quando necessária, ou seja, por demanda
  - **Princípio da localidade**

# Paginação por Demanda

- Se precisar de outra página: FALTA DE PÁGINA (*page fault*)
  - Busca no disco
- Possível mapear mais aplicações na memória
- Páginas não usadas não precisam ser mantidas na memória
- Uso de política de substituição de página
- Alta taxa de falta de página: ***thrashing***

# Paginação por Demanda

- Se precisar de outra página: FALTA DE PÁGINA (*page fault*)

- Busca

- Possível  
memória

- Páginas  
mantidas na memória

- Política de substituição de página

- Alta taxa de falta de página: ***thrashing***

ONDE JÁ VIMOS ISSO?



# MEMÓRIA VIRTUAL

# Memória Virtual

- **Princípio:** Não é necessário carregar todo o programa na memória
- Se o programa for maior que a memória, é necessário criar maneiras de estruturá-lo em partes que possam ser carregadas uma de cada vez
  - Com a paginação por demanda, quem faz isso é SO e hardware
  - Para o programador existe uma memória imensa (a troca de páginas é transparente)

# Memória Virtual

- **Princípio:** Não é necessário carregar todo o programa na memória

O QUE ACONTECERIA SE  
NÃO HOUVESSE A TROCA  
DE PÁGINA E A MEMÓRIA  
PRINCIPAL ESTIVESSE  
CHEIA?

PAGE\_FAULT\_IN\_NONPAGED\_AREA



If this is the first time you've seen this error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:



\*\*\* STOP: 0x00000050 (0x8872A990, 0x00000001, 0x804F35D7, 0x00000000)

\*\*\* ati3diag.dll - Address ED80AC55 base at ED88F000, Date Stamp 3dcb24d0

Beginning dump of physical memory  
Physical memory dump complete.

Desculpe, mas você não pode carregar mais nenhum aplicativo. Por favor, feche um dos programas abertos para poder abrir um novo

#### Windows

A fatal exception 8E has occurred at 0020:C00060F0 in UxD UHM(81) + 000059F8. The current application will be terminated.

- \* Press any key to terminate the application.
- \* Press CTRL+ALT+DEL to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue

# Memória Virtual

- ▶ **Funcionamento**
- ▶ **Desempenho**
- ▶ **Papel na hierarquia de memória**

# Memória Virtual

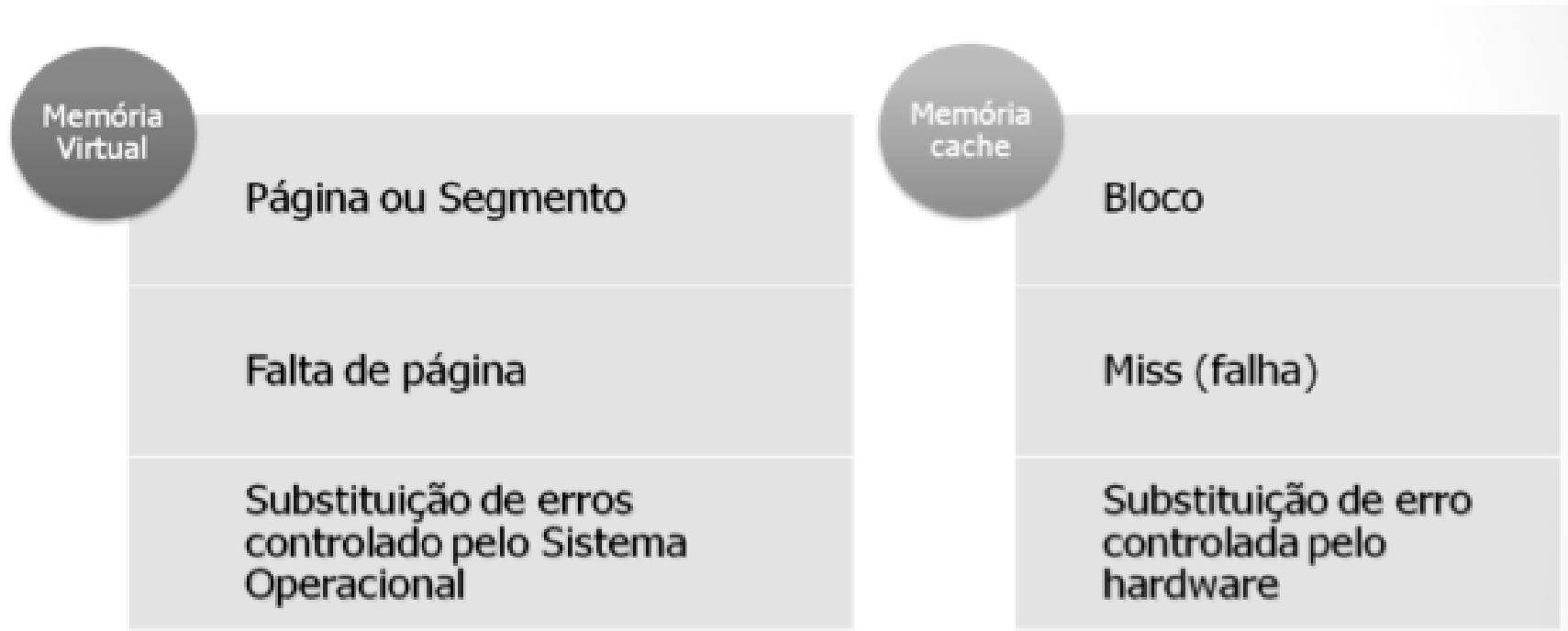
- ▶ Memória Virtual é a técnica que dá ao programador a ilusão de poder acessar rapidamente um grande espaço de endereçamento.
- ▶ Objetivos da técnica:
  - Permitir que haja um meio seguro e eficiente de se compartilhar informações, armazenadas na memória, entre vários programas
    - Máquinas virtuais!
  - Minimizar os problemas causados aos programas pela existência de uma pequena quantidade de memória principal

# Memória Virtual

- ▶ Os **programas** que compartilham a memória de determinada máquina **mudam dinamicamente** durante o processo de execução.
- ▶ Cada programa deve ser **compilado** usando seu próprio **espaço de endereçamento** (ou seja, em uma região da memória acessível somente a esse programa).
- ▶ A técnica de memória **virtual** realiza a **tradução** do **espaço de endereçamento** de um programa para seus **endereços reais**.



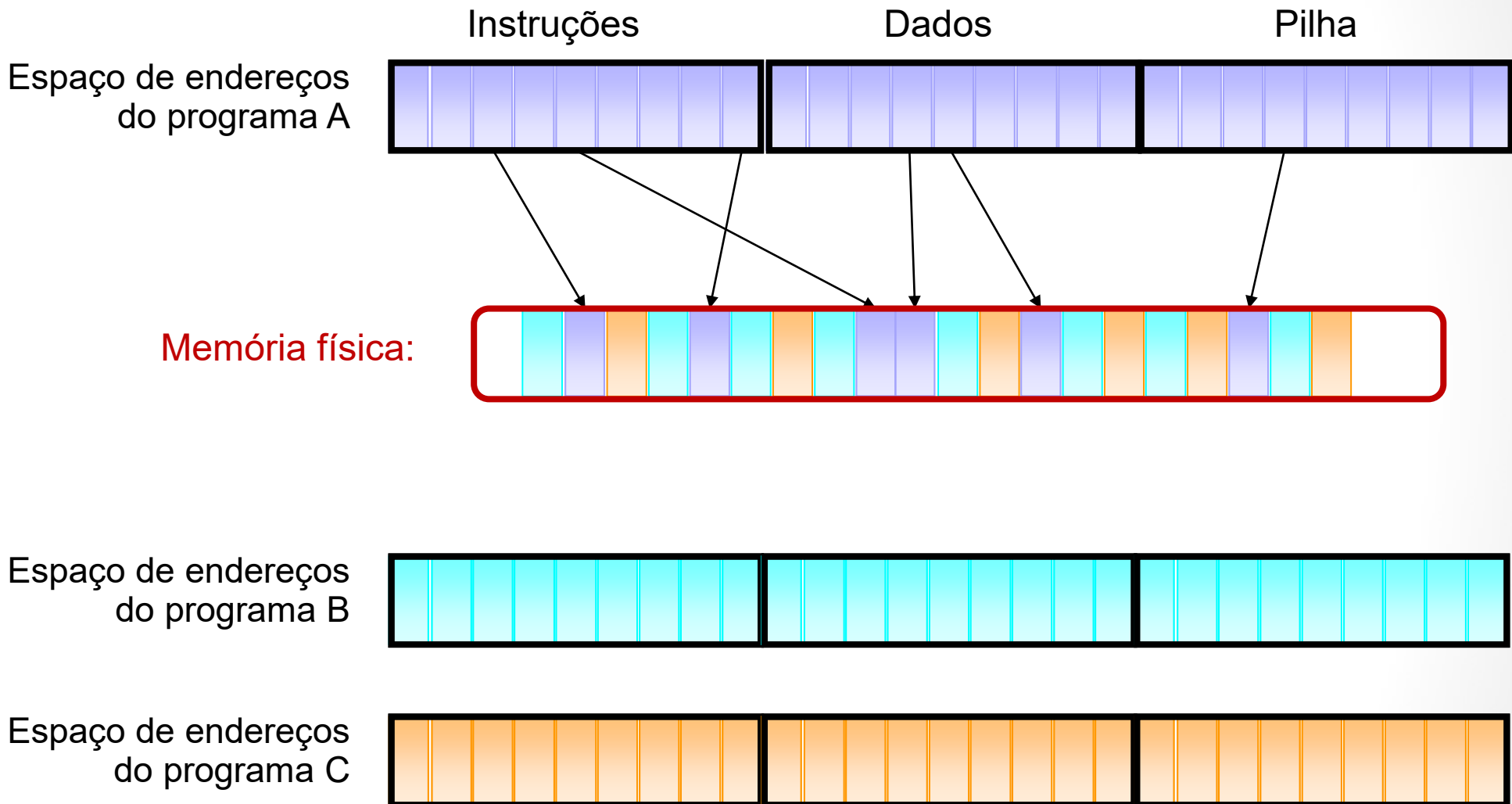
# Memória Virtual x Memória Cache



# Memória Virtual x Memória Cache

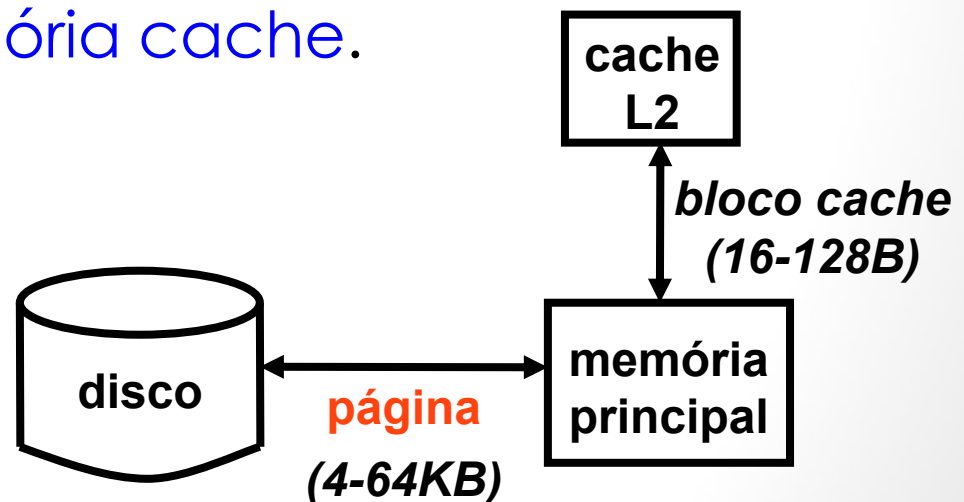
Parâmetro	Cache L1	Memória virtual
Tamanho do bloco (página)	16-128 bytes	4.096-65.536 bytes
Tempo de acerto (hit)	1-3 ciclos	100-200 ciclos
Penalidade de falha	8-200 ciclos	1 – 10 M ciclos
Taxa de falha	0,1-10%	0,00001-0,001%

# Memória Virtual – tradução de endereços

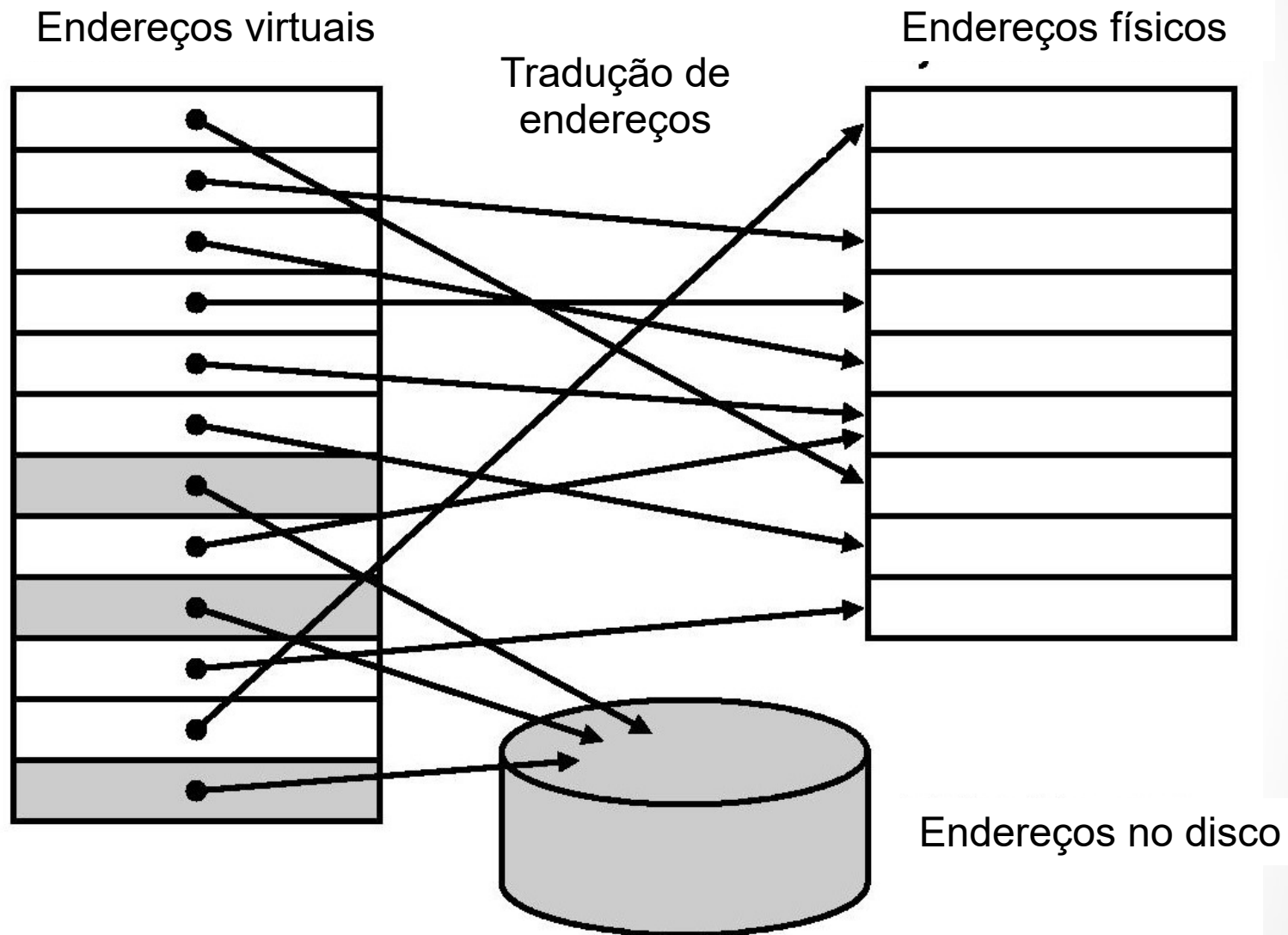


# Funcionamento

- ▶ Transferência entre disco e memória ocorre em **páginas**.
- ▶ Tamanho da página é grande para amortizar o alto custo de acesso ao disco.
- ▶ Compromisso (*tradeoff*) no aumento do tamanho das **páginas** na **memória virtual** é similar ao compromisso no aumento dos **blocos** para a **memória cache**.



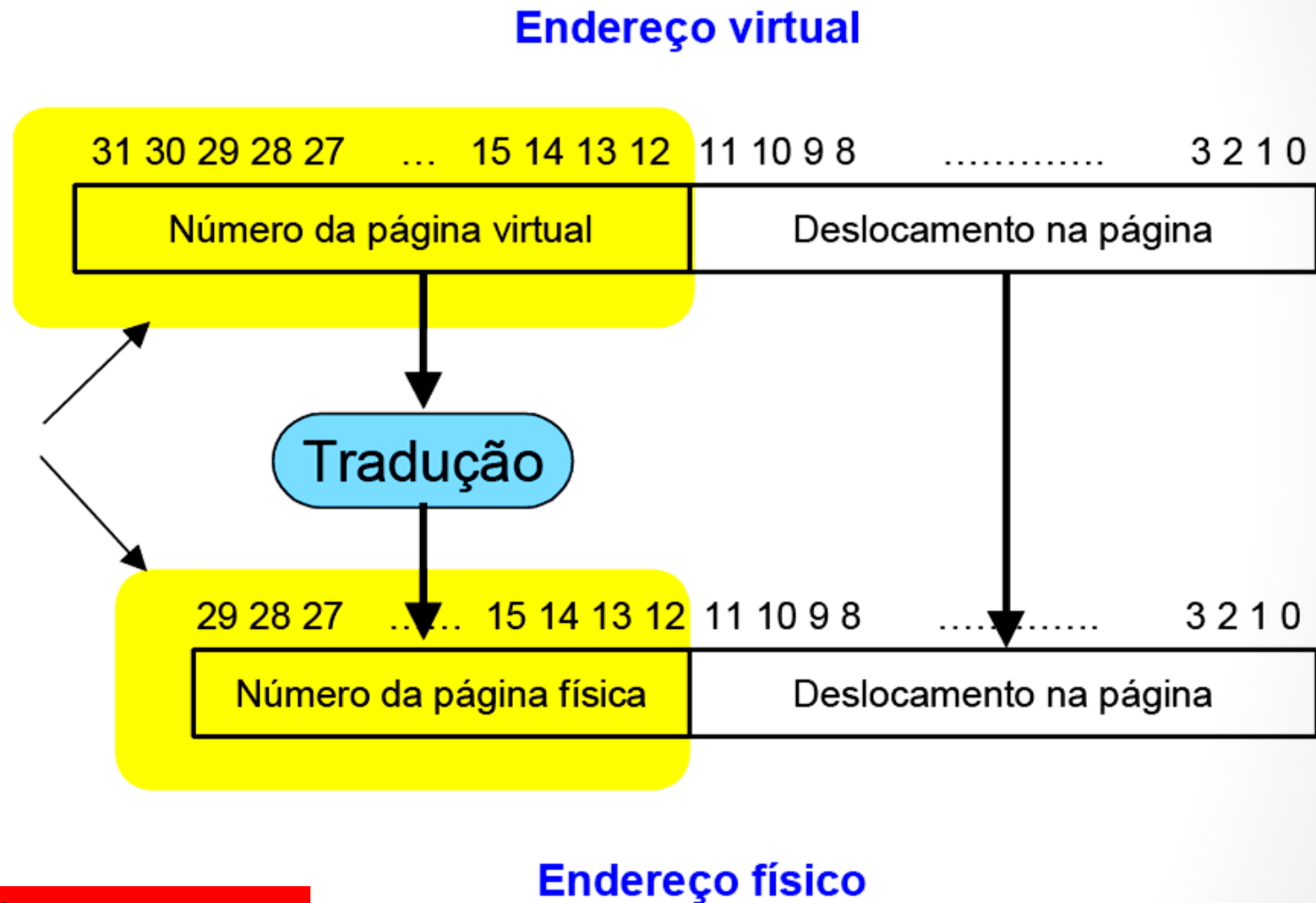
# Memória Virtual – tradução de endereços



# Memória Virtual – tradução de endereços

- ▶ Tradução de endereços é feita pelo hardware e pelo sistema operacional (SO).
- ▶ SO mantém para cada programa:
  - Quais páginas estão associadas a ele;
  - Onde fica cada página no disco;
  - Quais páginas estão residentes na memória;
  - O nº de cada página física associada com o nº da página virtual residente na memória.

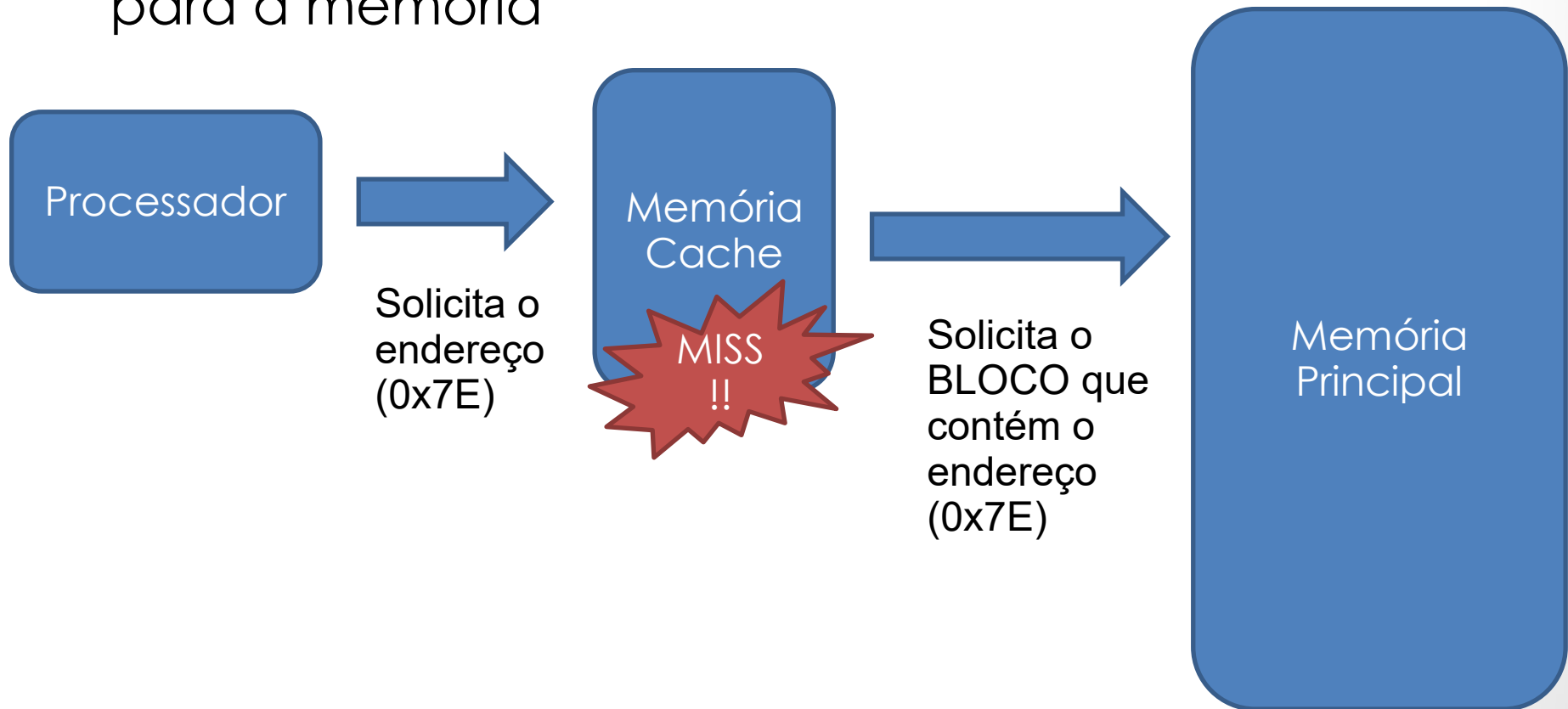
# Memória Virtual – tradução de endereços



Dá a **ilusão** de se ter uma quantidade de memória maior à disposição

# Exemplo de tradução

- O que acontece se nesse sistema processador solicitar o endereço 126 (0x7E) e houver um miss na cache?
- A cache deve solicitar o bloco do endereço 0x7E para a memória





# Exemplo de tradução

- Uma vez que a memória principal recebe a solicitação, se faz necessária uma tradução
  - Até então, processador e memória cache estavam lidando com um ENDEREÇO VIRTUAL (mas não sabiam disso...)
- **Neste sistema em particular** a tradução ocorre de modo que:
  - Para endereçar uma palavra dentro de uma páginas são precisos **4 bits** (cada página tem **16 palavras**)
  - Endereço Virtual tem **8 bits**
  - Endereço Físico tem **6 bits**

# Exemplo de tradução

- Para realizar a tradução, o sistema de memória virtual conta com uma tabela de tradução
  - É indexada pelo número da página no endereço virtual (**4 bits**)
  - Cada entrada tem largura de um número de página no endereço físico (**2 bits**)

Indica se aquela entrada é válida ou não

2 bits de endereço físico

Válido	Endereço Físico

# Tabela de tradução

- A tabela de tradução é preenchida conforme as páginas são solicitadas (em função de misses da cache e page faults)

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	0		
0011	0		
0100	0		
0101	0		
0110	0		
0111	0		
1000	0		
1001	0		
1010	0		
1011	0		
1100	0		
1101	0		
1110	0		
1111	0		

$2^4 = 16$  entradas

# Tabela de tradução

- A tabela de tradução é preenchida conforme as páginas são solicitadas (em função de misses da cache e page faults)

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	0		
0011	0		
0100	0		
0101	0		
0110	0		
0111	0		
1000	0		
1001	0		
1010	0		
1011	1	1	0
1100	0		
1101	0		
1110	0		
1111	0		

2<sup>4</sup> = 16 entradas

# Tabela de tradução

- A tabela de tradução é preenchida conforme as páginas são solicitadas (em função de misses da cache e page faults)

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	1	0	1
0011	0		
0100	0		
0101	0		
0110	0		
0111	0		
1000	0		
1001	0		
1010	0		
1011	1	1	0
1100	0		
1101	0		
1110	0		
1111	0		

2<sup>4</sup> = 16 entradas

# Tabela de tradução

- A tabela de tradução é preenchida conforme as páginas são solicitadas (em função de misses da cache e page faults)

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	1	0	1
0011	0		
0100	0		
0101	0		
0110	0		
0111	0		
1000	0		
1001	0		
1010	0		
1011	1	1	0
1100	0		
1101	1	0	0
1110	0		
1111	0		

2<sup>4</sup> = 16 entradas

# Tabela de tradução

- A tabela de tradução é preenchida conforme as páginas são solicitadas (em função de misses da cache e page faults)

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	1	0	1
0011	0		
0100	0		
0101	0		
0110	0		
0111	1	1	1
1000	0		
1001	0		
1010	0		
1011	1	1	0
1100	0		
1101	1	0	0
1110	0		
1111	0		

$2^4 = 16$  entradas

# Exemplo BEM MAIS simples

Endereço Virtual

0 1 1 1 1 1 1 0

DIRETO

Endereço Físico?



# Exemplo BEM MAIS simples

Endereço Virtual

0 1 1 1 1 1 1 0



Endereço Físico?

1 1 1 0

DIRETO

# Exemplo BEM MAIS simples

TABELA DE TRADUÇÃO!

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	1	0	1
0011	0		
0100	0		
0101	0		
0110	0		
0111	1	1	1
1000	0		
1001	0		
1010	0		
1011	1	1	0
1100	0		
1101	1	0	0
1110	0		
1111	0		

Endereço Virtual

0 1 1 1 1 1 1 0

Tradução

DIRETO

Endereço Físico?

1 1 1 0

# Exemplo BEM MAIS simples

TABELA DE TRADUÇÃO!

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	1	0	1
0011	0		
0100	0		
0101	0		
0110	0		
0111	1	1	1
1000	0		
1001	0		
1010	0		
1011	1	1	0
1100	0		
1101	1	0	0
1110	0		
1111	0		

Endereço Virtual

0 1 1 1 1 1 1 0

Tradução

DIRETO

Endereço Físico?

1 1 1 1 1 0

# Exemplo BEM MAIS simples

## TABELA DE TRADUÇÃO!

Válido	Endereço Físico
--------	-----------------

PERGUNTA:  
E se a entrada correspondente  
estivesse INVÁLIDA??

1110	0		
1111	0		

Endereço Físico

1 1 1 1 1 0

# Exemplo BEM MAIS simples

TABELA DE TRADUÇÃO!

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	1	0	1
0011	0		
0100	0		
0101	0		
0110	0		
0111	0	1	1
1000	0		
1001	0		
1010	0		
1011	1	1	0
1100	0		
1101	1	0	0
1110	0		
1111	0		

Endereço Virtual

0 1 1 1 1 1 1 0

Tradução

DIRETO

Endereço Físico

? ? 1 1 1 0

# Exemplo BEM MAIS simples

## TABELA DE TRADUÇÃO!

Válido	Endereço Físico
--------	-----------------

RESPOSTA:

Isso significa que a página solicitada NÃO se encontra na memória principal – PAGE FAULT

1110	0		
1111	0		

Endereço Físico

? ? 1 1 1 0

# Exemplo BEM MAIS simples

## TABELA DE TRADUÇÃO!

Válido	Endereço Físico
--------	-----------------

E como sabemos onde a página está no disco?

1110	0		
1111	0		

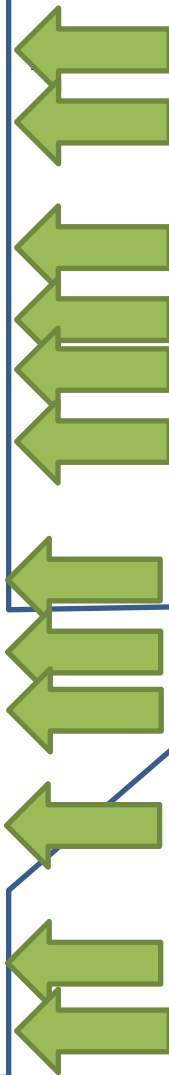
Endereço Físico

? ? 1 1 1 0

# Exemplo BEM MAIS simples

TABELA DE TRADUÇÃO!

	Válido	Endereço Físico	
0000	0		
0001	0		
0010	1	0	1
0011	0		
0100	0		
0101	0		
0110	0		
0111	0	1	1
1000	0		
1001	0		
1010	0		
1011	1	1	0
1100	0		
1101	1	0	0
1110	0		
1111	0		



Endereço V

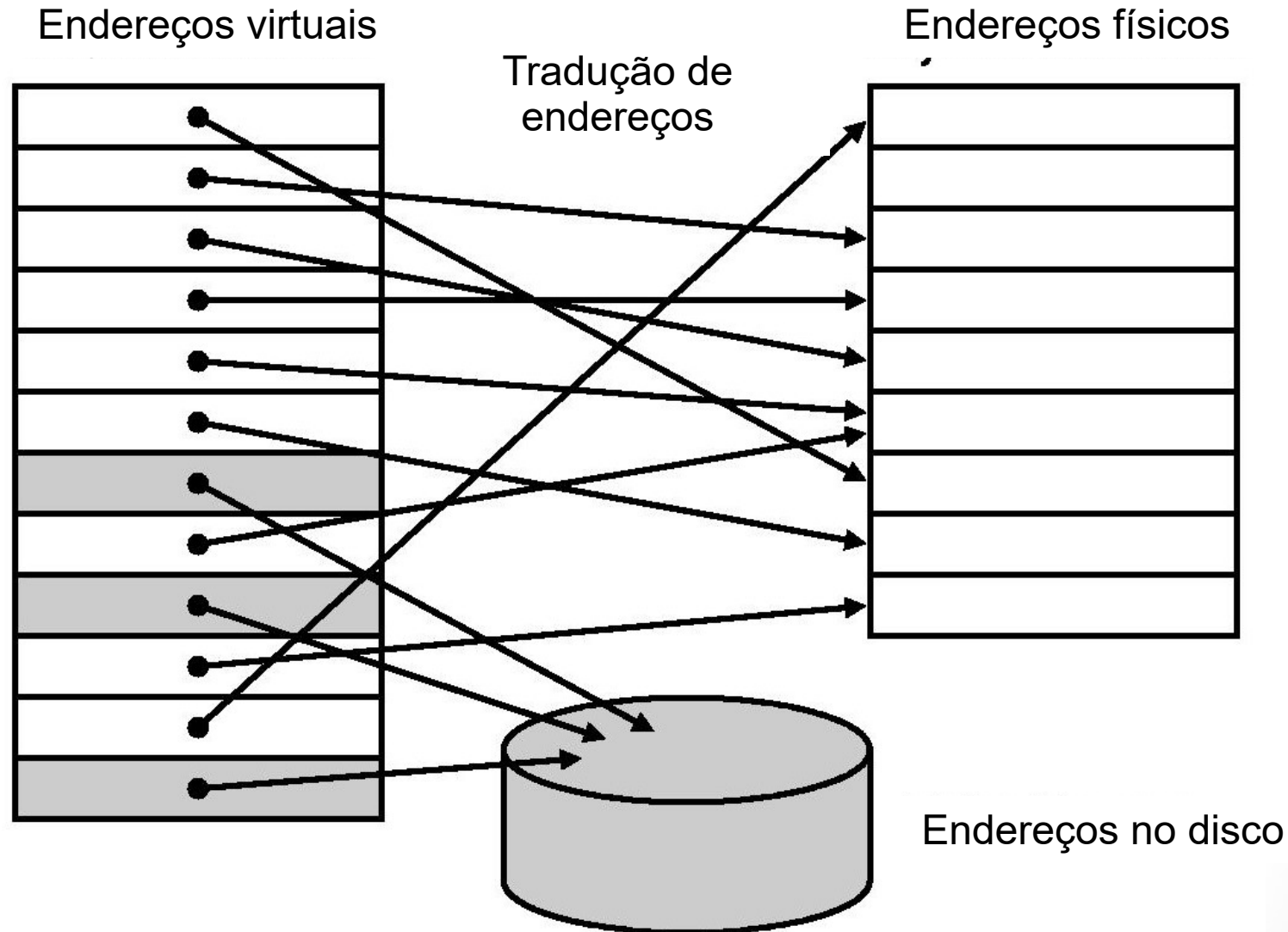
As entradas  
“inválidas”  
armazenam o  
endereço no  
disco

Endereço Físico

? ? 1 1 1 0



# Exemplo BEM MAIS simples



## I. Posicionamento da página.

Onde a página deve ser colocada na memória principal?

## II. Identificação da página.

Como a página é encontrada na memória principal?

## III. Substituição de página.

Quais páginas serão trocadas em uma falta?

## IV. Estratégia de gravação.

O que acontece em uma escrita de página?

# I – Posicionamento da Página

- ▶ A penalidade de erro para a memória virtual é  **muito alta**, pois envolve o acesso a um dispositivo de armazenamento  **magnético e mecânico**.
- ▶ Em razão disso, para reduzir a frequência de faltas de páginas, os sistemas operacionais utilizam o esquema de posicionamento  **totalmente associativo**.

## II - Identificação da Página

- ▶ A **desvantagem** da escolha do posicionamento **totalmente associativo** está em **localizar uma entrada**, já que ela pode estar em **qualquer lugar** da memória virtual.
- ▶ Como o **espaço** ocupado pela memória virtual é **maior** que aquele ocupado pela memória cache, o **tempo de busca** aumenta substancialmente.
- ▶ Para contornar essa desvantagem, utiliza-se uma **tabela de páginas**, uma estrutura que **indexa as traduções** de endereços **virtuais** para endereços físicos

## II - Identificação da Página

- ▶ Tabela de páginas está armazenada na memória principal
- ▶ É indexada com o número da página extraído do endereço virtual e contém o número da página física correspondente

## II - Identificação da Página

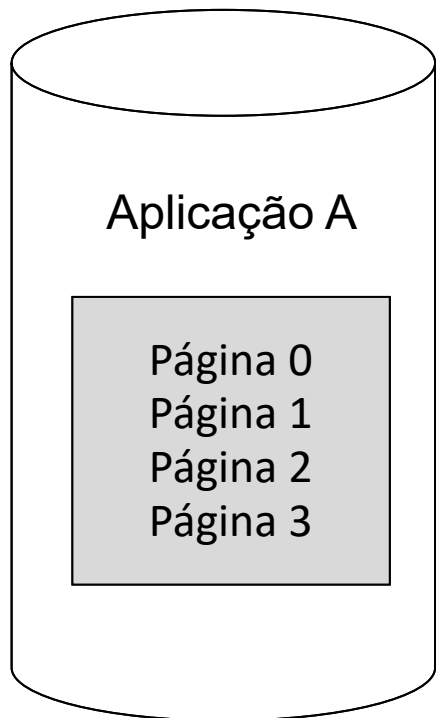
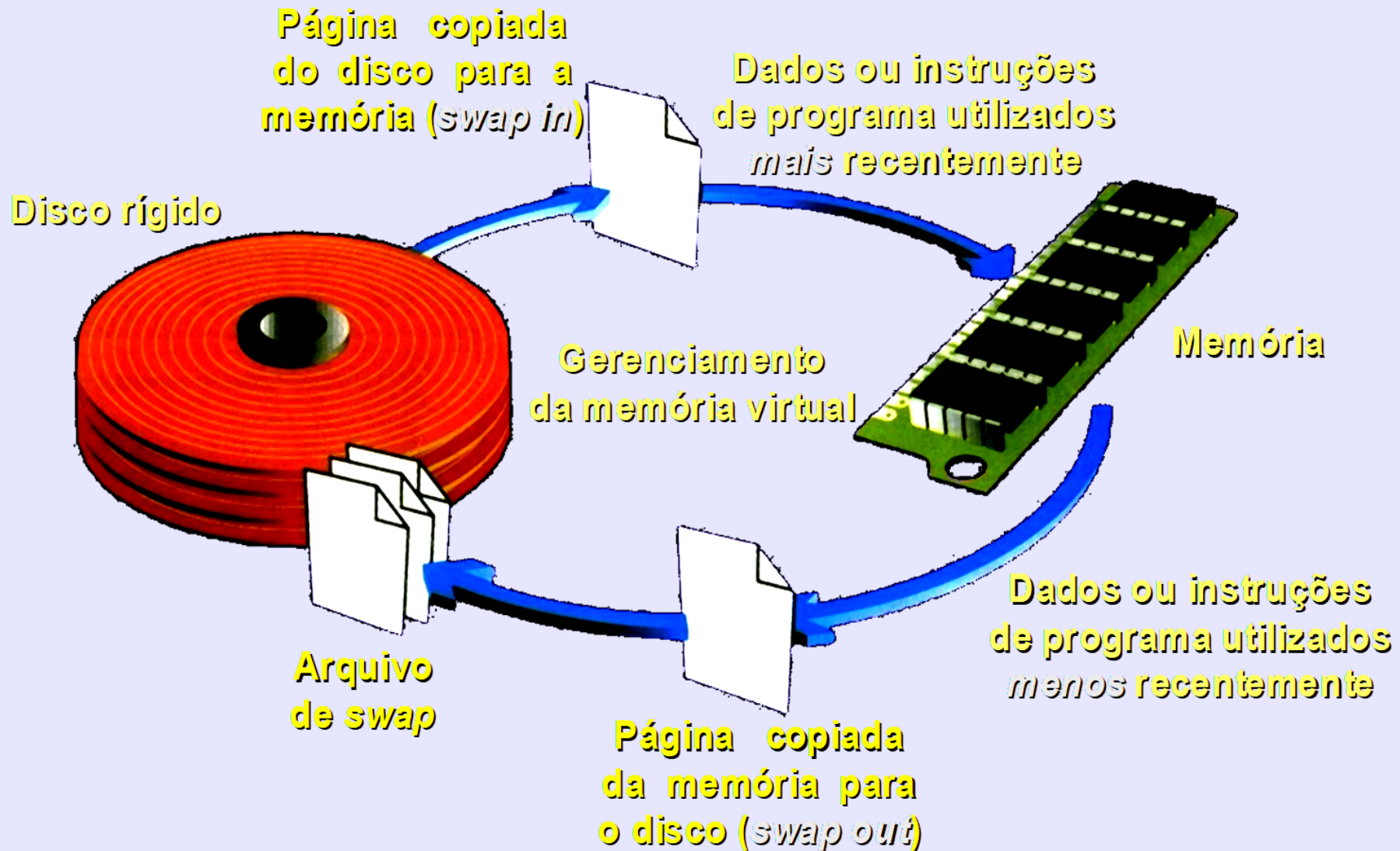


Tabela de página da  
Aplicação A

18
13
14
15

13	Página 1 de A
14	Página 2 de A
15	Página 3 de A
16	Em uso
17	Em uso
18	Página 0 de A
19	Em uso

# III – Substituição da Página



# III – Substituição da Página

- ▶ Quando ocorre falta de página:
  - ▶ O sistema operacional precisa
    - Encontrar a página faltante no nível hierárquico inferior (geralmente, no HD)
    - Decidir em que lugar da memória principal deve ser colocada a página requisitada
- ▶ O endereço virtual em si não informa em que posição do HD está a página que gerou a falta de página.



# III – Substituição da Página

- Em uma memória totalmente associativa, todos os blocos são candidatos à substituição.

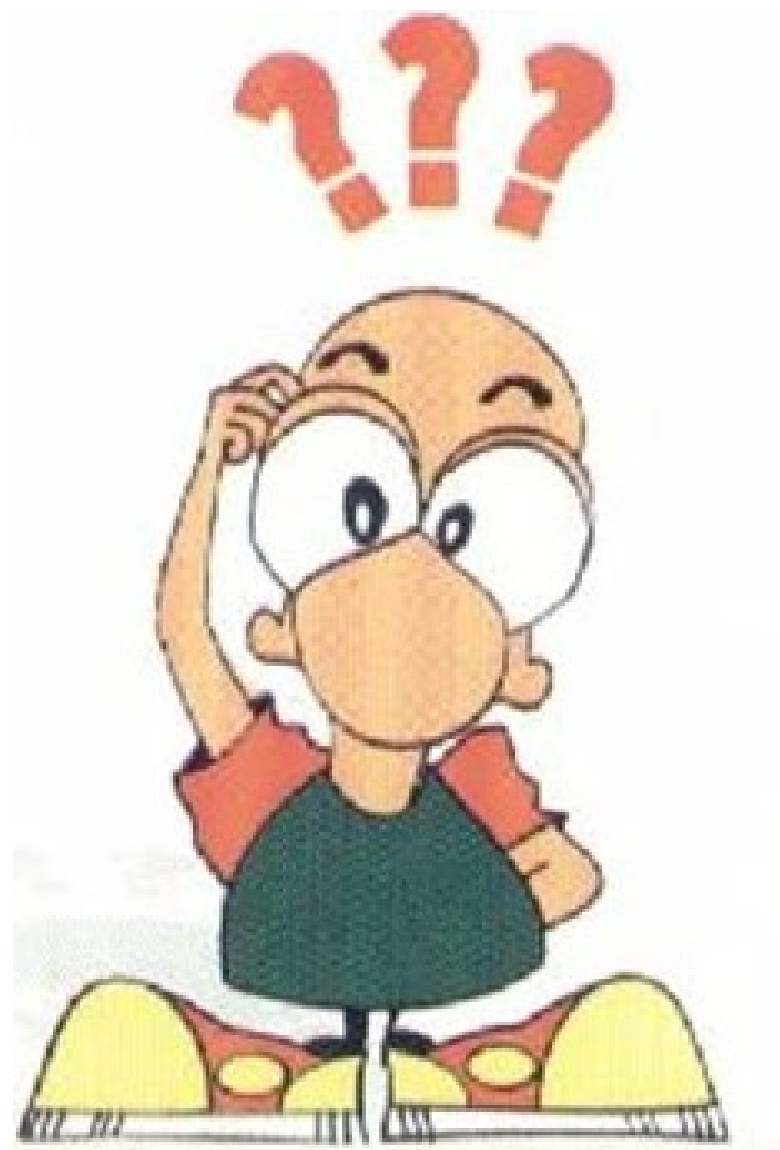
Estratégias para a substituição de blocos:

**Aleatória:** os blocos candidatos à substituição são escolhidos ao acaso, possivelmente contando com algum auxílio de hardware.

Bloco menos usado recentemente (**LRU**): o bloco substituído é aquele menos utilizado recentemente.

# IV – Estratégia de Gravação

- ▶ A escrita no disco consome de 1 milhão a 10 milhões de ciclos de clock.
- ▶ Esquema **write-through** não funciona para memória virtual.
- ▶ Esquema **write-back** é usado:
  - Página é copiada para o disco no momento em que for substituída (nomenclatura: “**copy-back**”)



## Para saber mais ...

- ▶ PATTERSON, D.A. & HENNESSY, J. L.  
**Organização e Projeto de Computadores -**  
A Interface Hardware/Software. 3ª ed. Campus, 2005.
- ▶ William Stallings, “Arquitetura e Organização de Computadores,” 8ª Edição
  - ▶ **Capítulo 8 – (sessão 8.3)**