UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Tópico 07: Gerência de Memória

Conceitos, Paginação e Segmentação

Prof. Rodrigo Campiolo
Prof. Rogério A. Gonçalves[1]

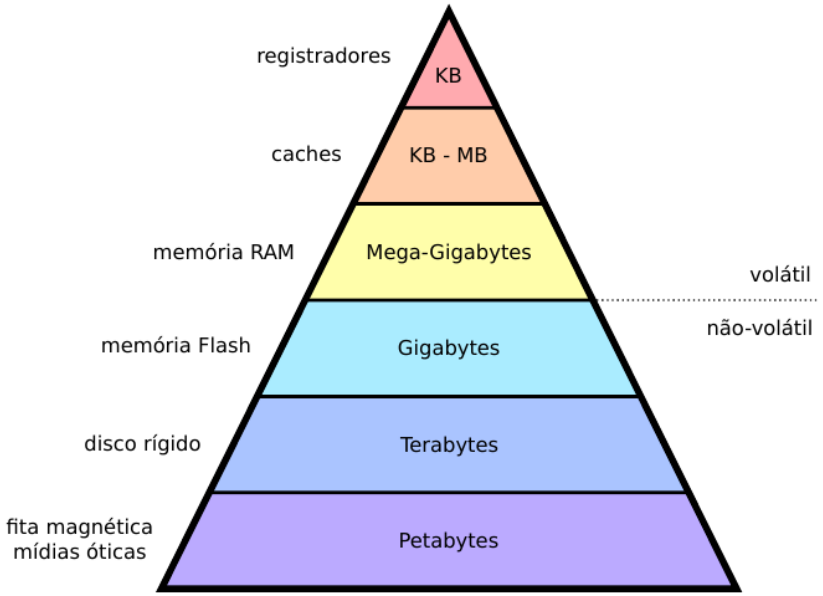
[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento de Computação (DACOM)
Campo Mourão, Paraná, Brasil

Ciência de Computação

BCC34G - Sistemas Operacionais

Introdução

- Dois principais tipos de memória
  - Primária: volátil (RAM).
  - Secundária: persistente (discos).
- Os programas são armazenados na memória secundária e carregados na memória primária (processos) para a execução.
- A gerência de memória é responsável pelo gerenciamento da memória primária e depende das facilidades providas pelo hardware.



Fonte: [1]

Endereço Lógico x Físico

- Endereços referem-se a uma posição de memória.
- **Endereços lógicos (virtuais)** são gerados pela CPU durante a execução de um programa.
- **Endereços físicos (reais)** indicam localizações na unidade de memória.
- Endereços lógicos são traduzidos para endereços físicos.
- Espaço de endereçamento lógico (processo) x espaço de endereçamento físico (hardware).

Memória de um Processo

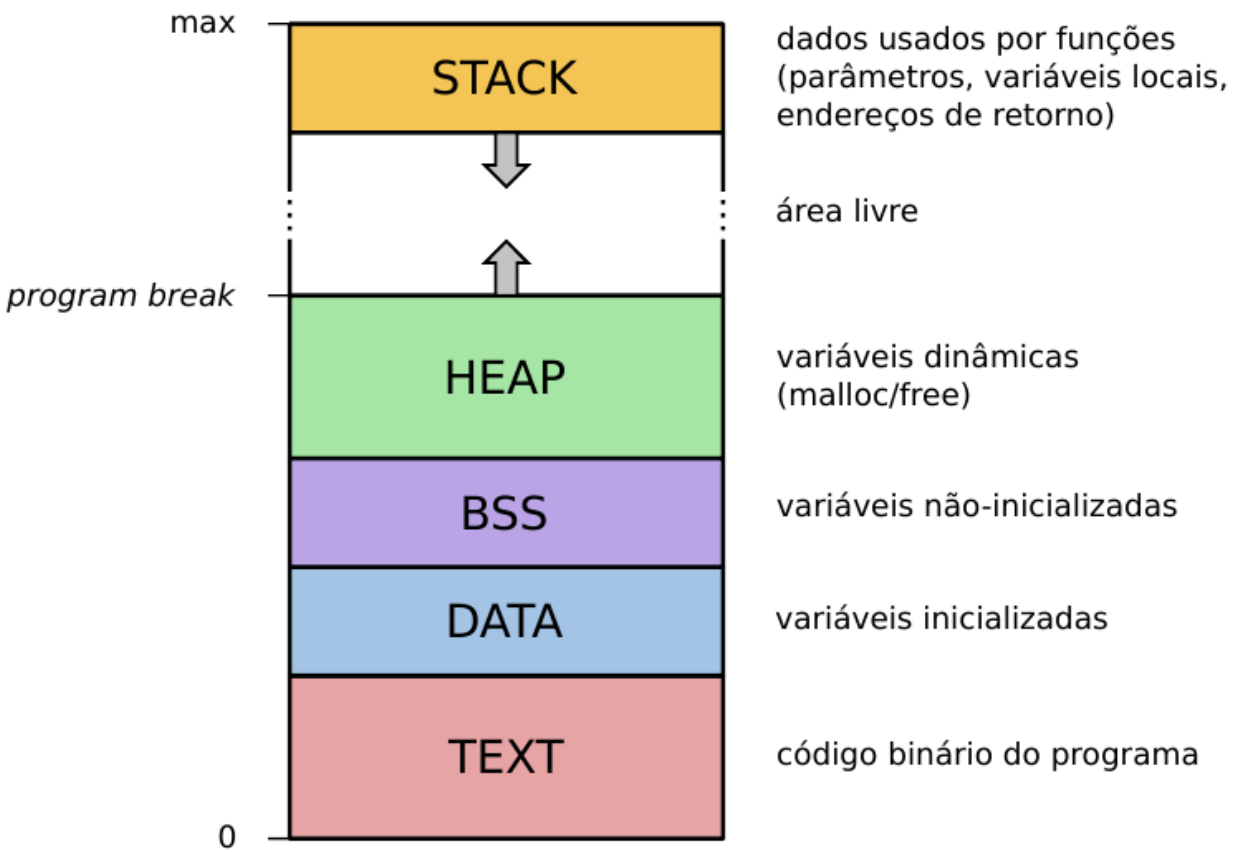
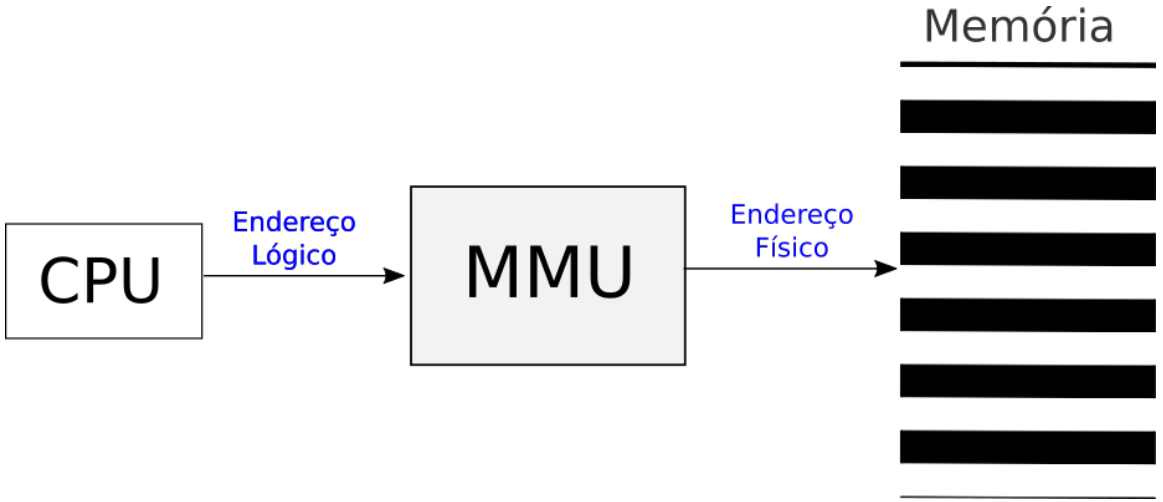


Figura 1: Organização de memória de um processo. [1]

MMU

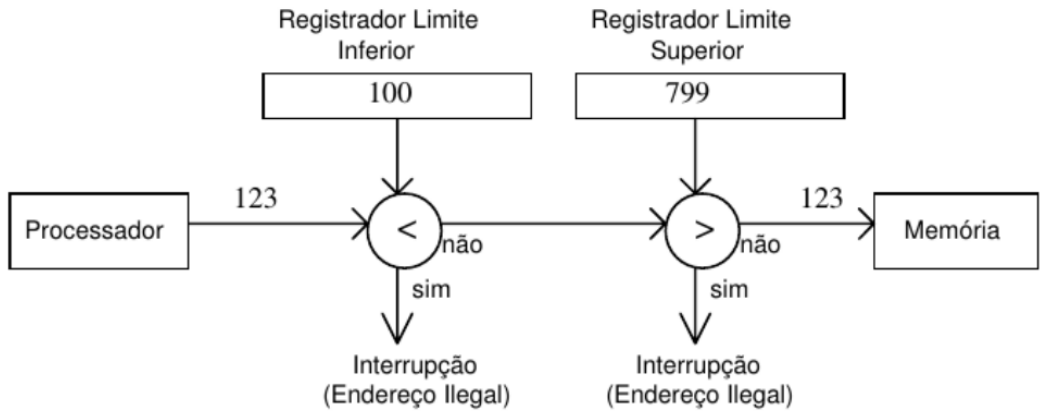
- *Memory Management Unit (MMU)*.
- Hardware responsável pelo mapeamento entre endereço lógico e endereço físico.



- Algumas das funcionalidades são: tradução de endereços, proteção de memória, controle de cache.

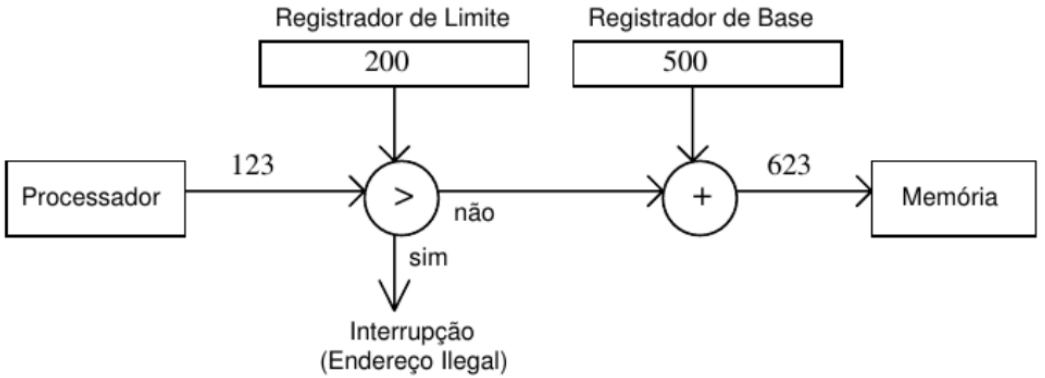
Proteção de Memória

- Proteção de memória usando registradores de limite.



End. Lógico == End. Físico

- Proteção de memória usando registradores de base e limite.



End. Lógico: 0 a 200
End. Físico: 500 a 700

Fonte: [2]

Carregador Relocador x Absoluto

Carregador relocador

- Realiza a correção das referências de memória segundo a posição de carga do programa.
- Usado quando o esquema de proteção é registradores de limite.
- Correção via software.

Carregador absoluto

- Realiza a tradução das referências de memória durante a execução.
- Usado quando o esquema de proteção é registradores base e limite.
- Correção via hardware.

Gerência de Memória Livre

- Mapa de bits x Lista encadeada

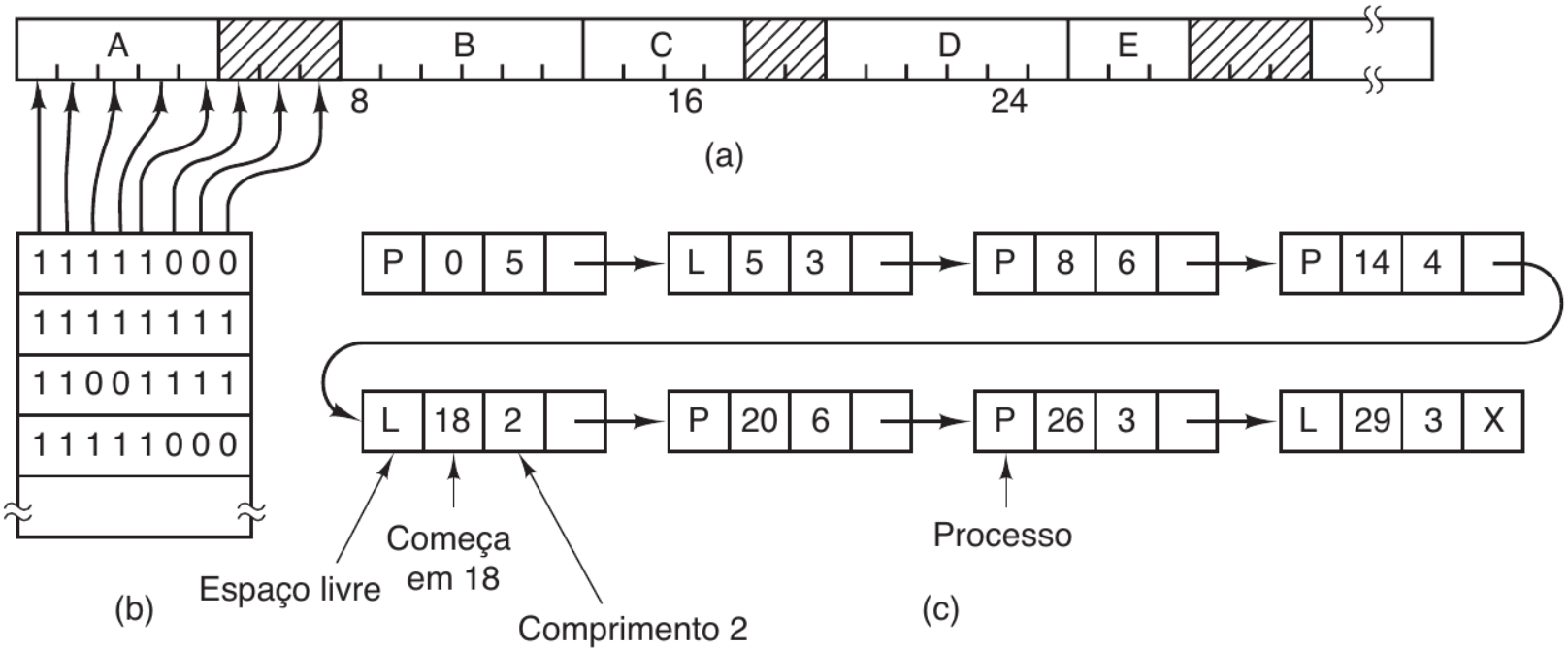


Figura 2: (a) memória com 5 processos e três lacunas livres. (b) mapa de bits. (c) lista encadeada. [3]

Alocação de Memória

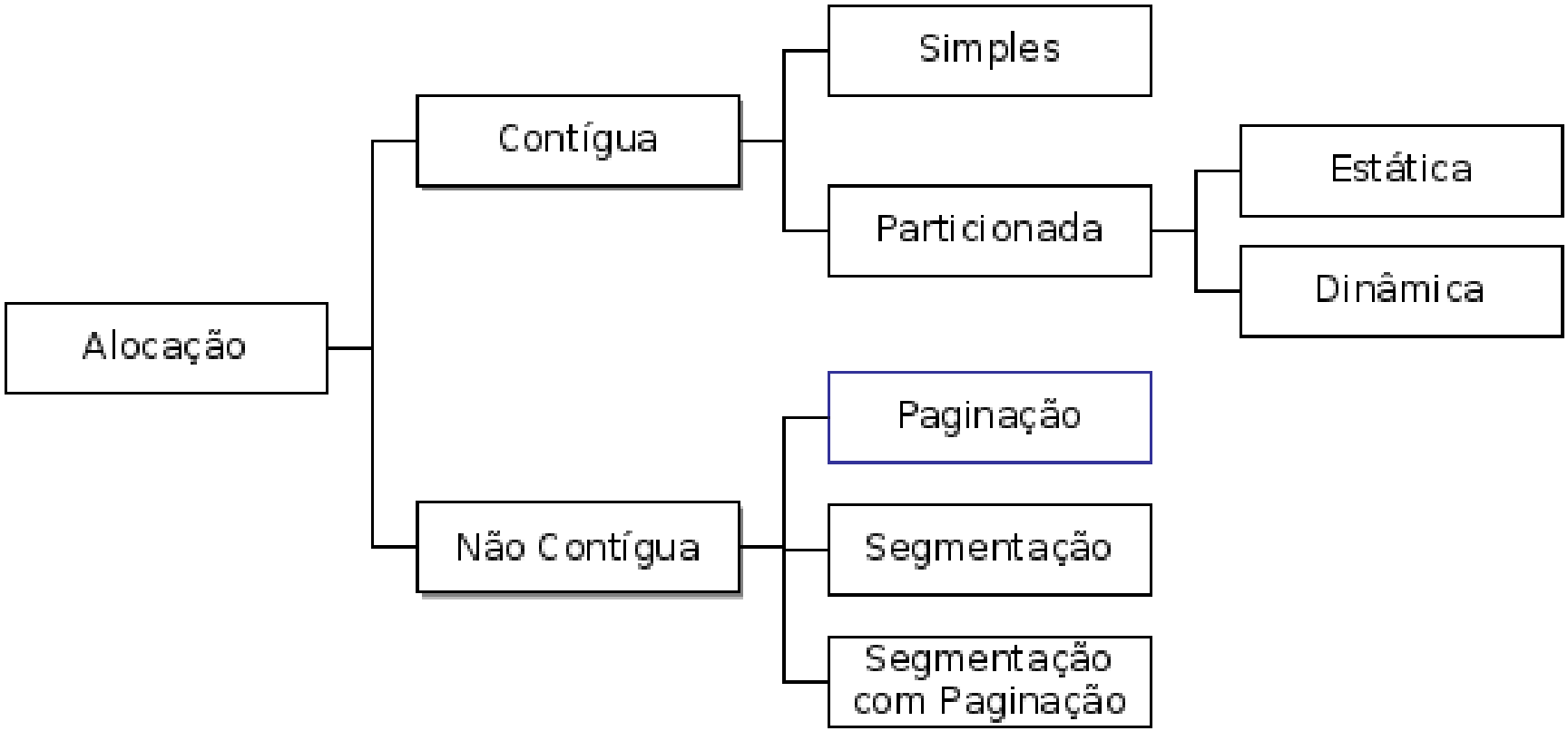
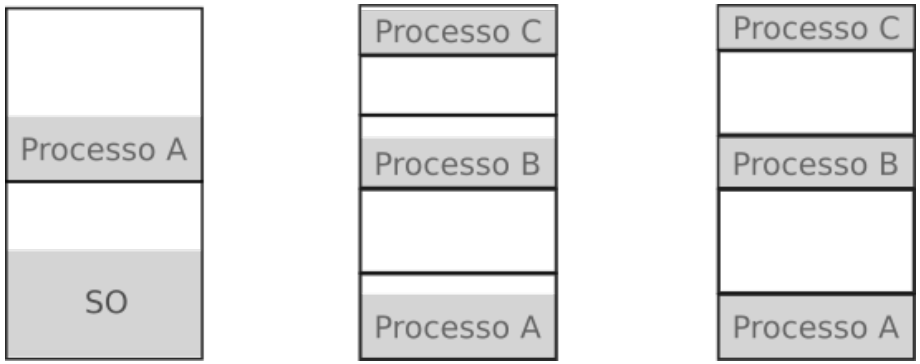


Figura 3: Formas de alocação de memória.

Alocação de Memória

Alocação contígua

- Simples
  - Divide em duas partes: SO e usuário.
- Particionada Estática
  - Partições de tamanho variável e estáticas.
- Particionada Dinâmica
  - Partições de tamanho variável e dinâmicas.
  - Adaptam-se ao tamanho do processo.



Alocação de Partição

- Algoritmos para alocação
  - *First-fit*: aloca primeira lacuna.
  - *Best-fit*: aloca lacuna com menor sobra.
  - *Worst-fit*: aloca lacuna com maior sobra.
  - *Circular-fit* ou *Next-fit*: aloca próxima lacuna a partir da última alocação.
  - *Quick-fit*: possui listas de espaços livres para os tamanhos mais utilizados.

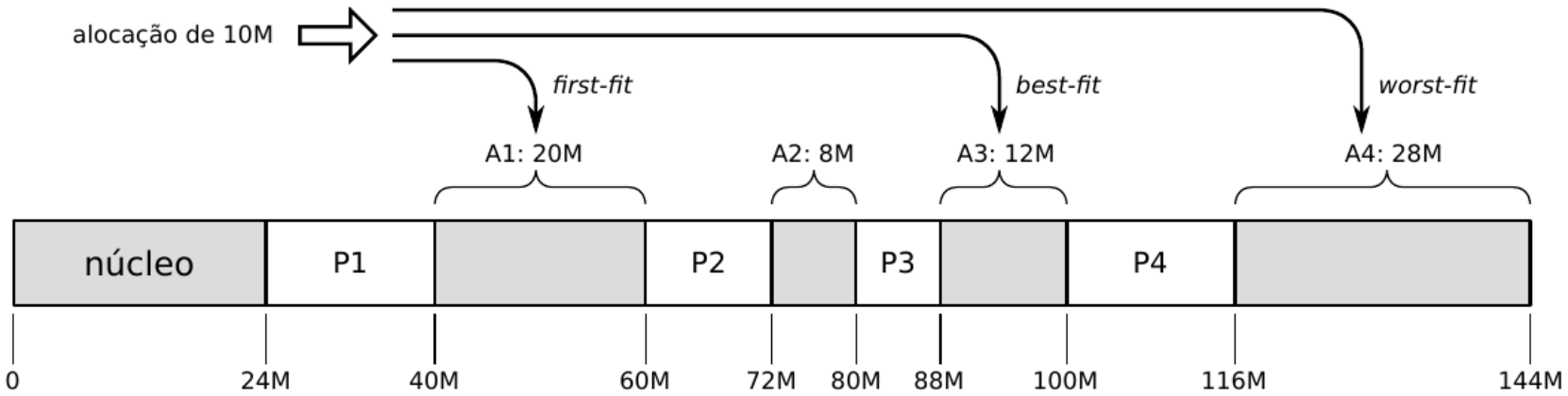


Figura 4: Algoritmos de alocação de partição. [1]

Framentação Interna x Externa

- Fragmentação interna: espaço interno à partição não usado.
- Fragmentação externa: espaços disponíveis, mas não contíguos para caber processos.

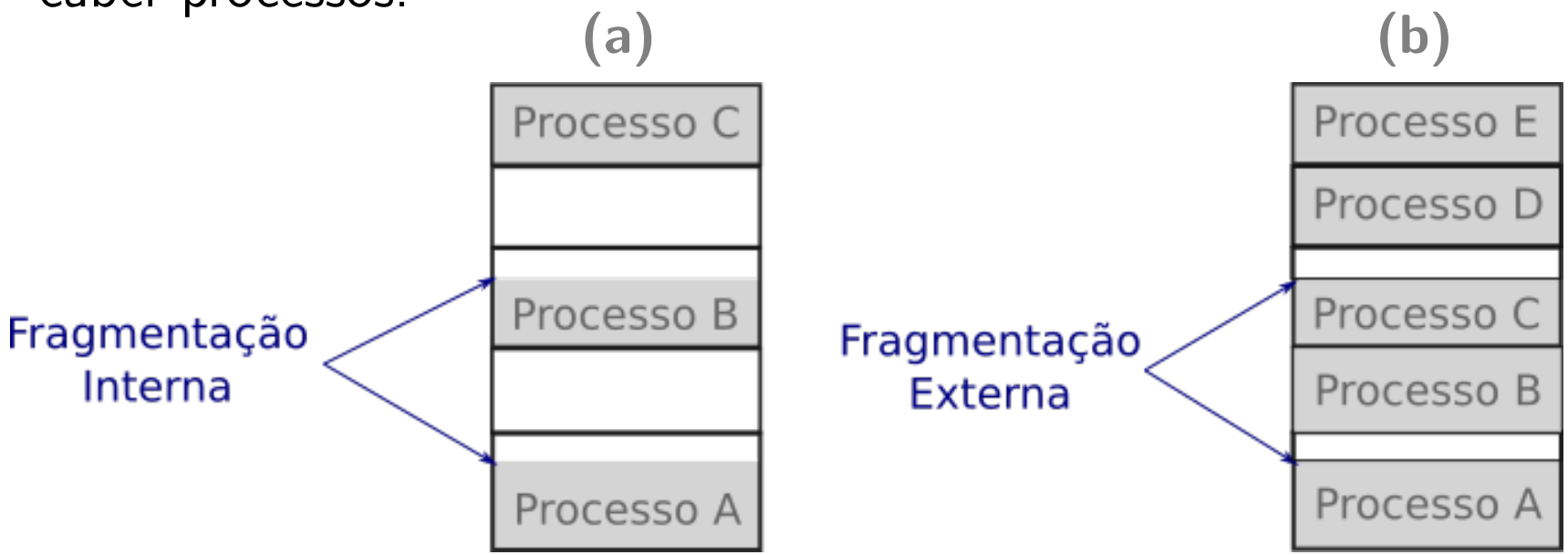
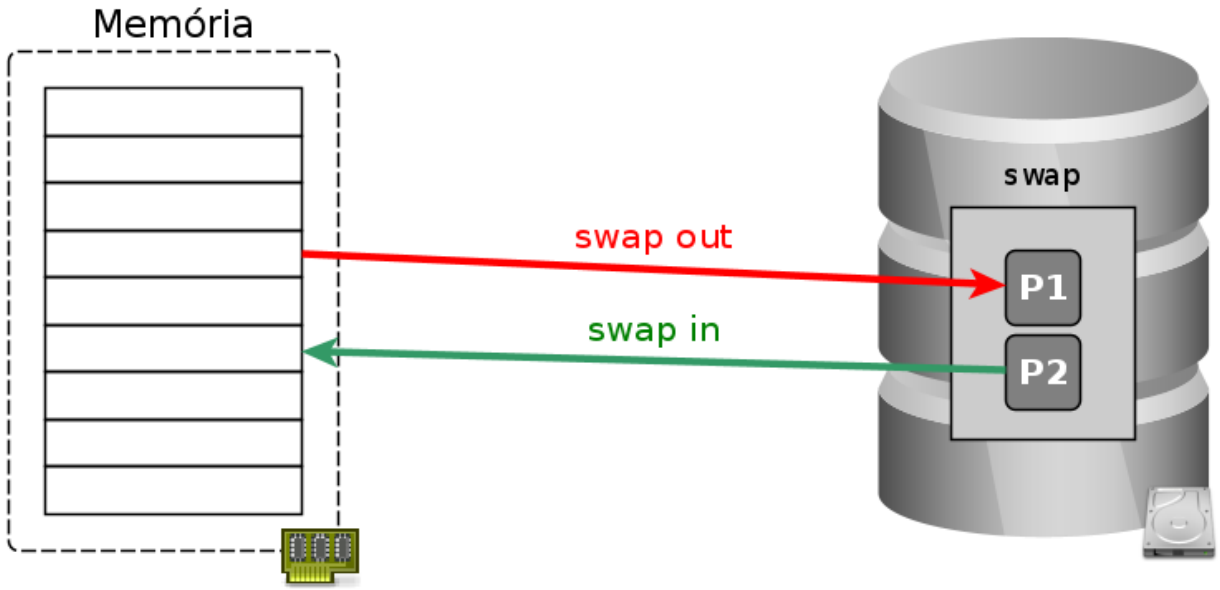


Figura 5: (a) fragmentação interna: 5 partições e 3 processos. (b) fragmentação externa: 7 partições e 5 processos.

Swapping

- O swapping consiste em mover processos entre a memória principal e a secundária.



- **swap out** move para a memória secundária.
- **swap in** move para a memória primária.

Windows:
- swapfile.sys
- pagefile.sys
- hiberfil.sys

Linux:
- partição de swap

Paginação

- Consiste em dividir a memória física (sistema) e a memória lógica (processo) em blocos de tamanho fixo e idênticos.
  - memória física dividida em blocos fixos denominados **quadros** (*frames*).
  - memória lógica dividida em blocos fixos denominados **páginas** (*pages*).
- Cada processo possui uma **Tabela de Páginas** com as suas páginas.
- Toda página está mapeada para um quadro.
- Tamanho da página é igual ao tamanho do quadro.

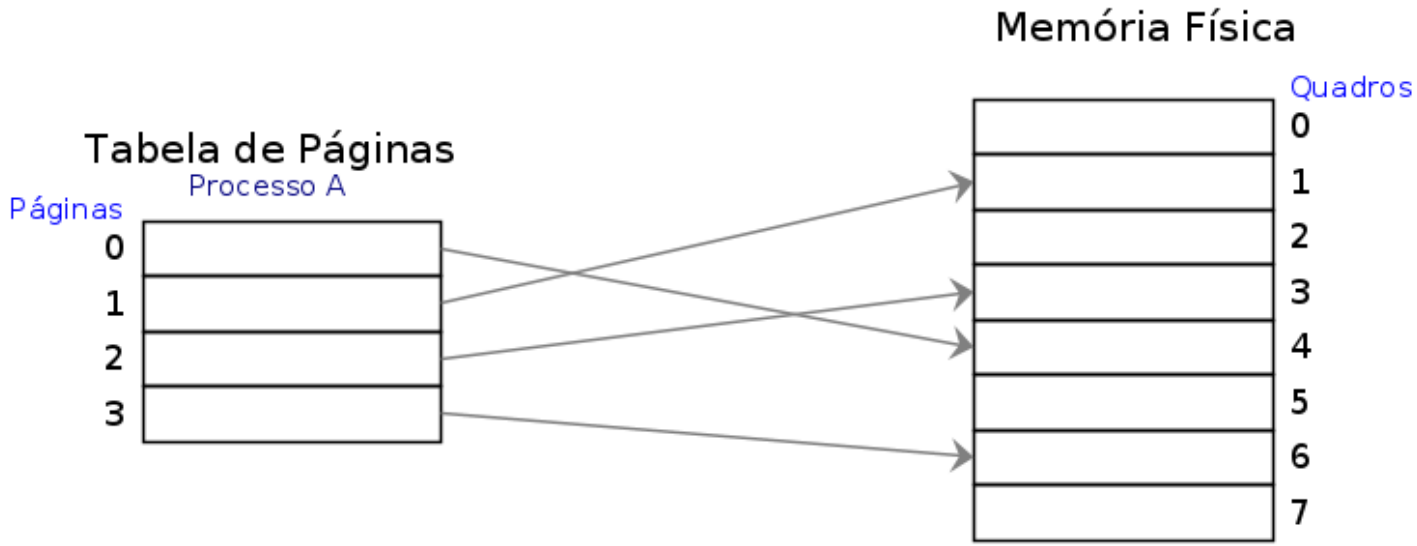
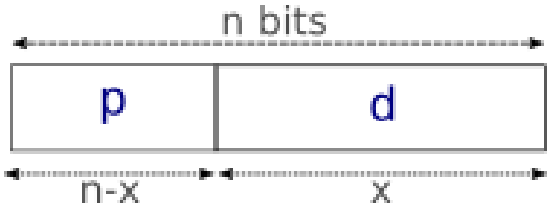


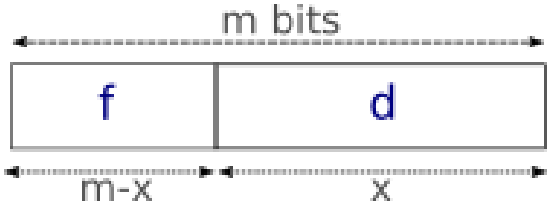
Figura 6: Mapeamento entre páginas e quadros.

Paginação

- O endereço lógico é dividido em duas partes:
  - número da página (p).
  - deslocamento na página (d).



- O endereço físico é dividido em duas partes:
  - número do quadro (f).
  - deslocamento no quadro (d).



Paginação

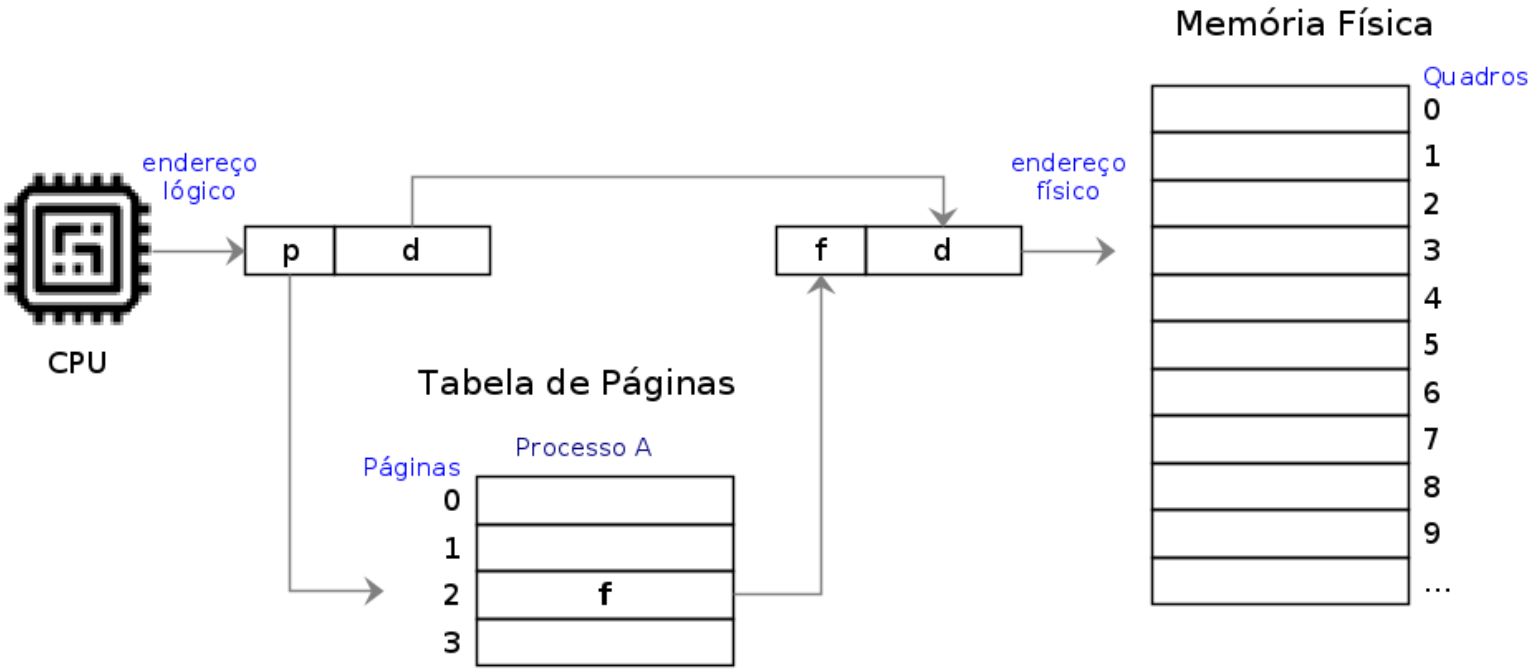
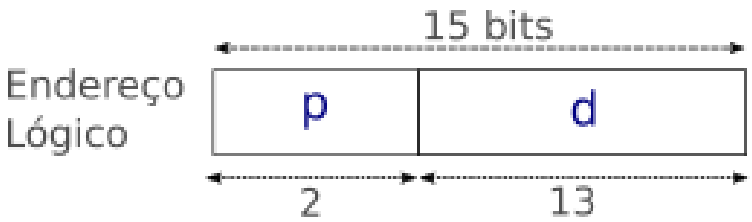
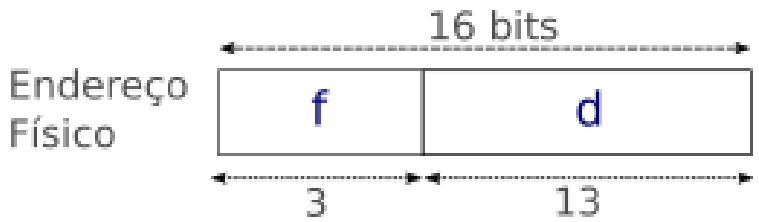


Figura 7: Tradução de endereço lógico para endereço físico.

Paginação: Exemplo

- Considere o sistema:
  - memória física: 64 KiB (16 bits).
  - tamanho máximo do processo: 32 KiB (15 bits).
  - tamanho de páginas: 8 KiB (13 bits).
- Paginação:
  - número de quadros: 64 KiB / 8 KiB = 8 (3 bits) - quadros de 0 a 7.
  - número de páginas: 32 KiB / 8 KiB = 4 (2 bits) - páginas de 0 a 3.
  - deslocamento: 8 KiB (13 bits).





Paginação: Exemplo

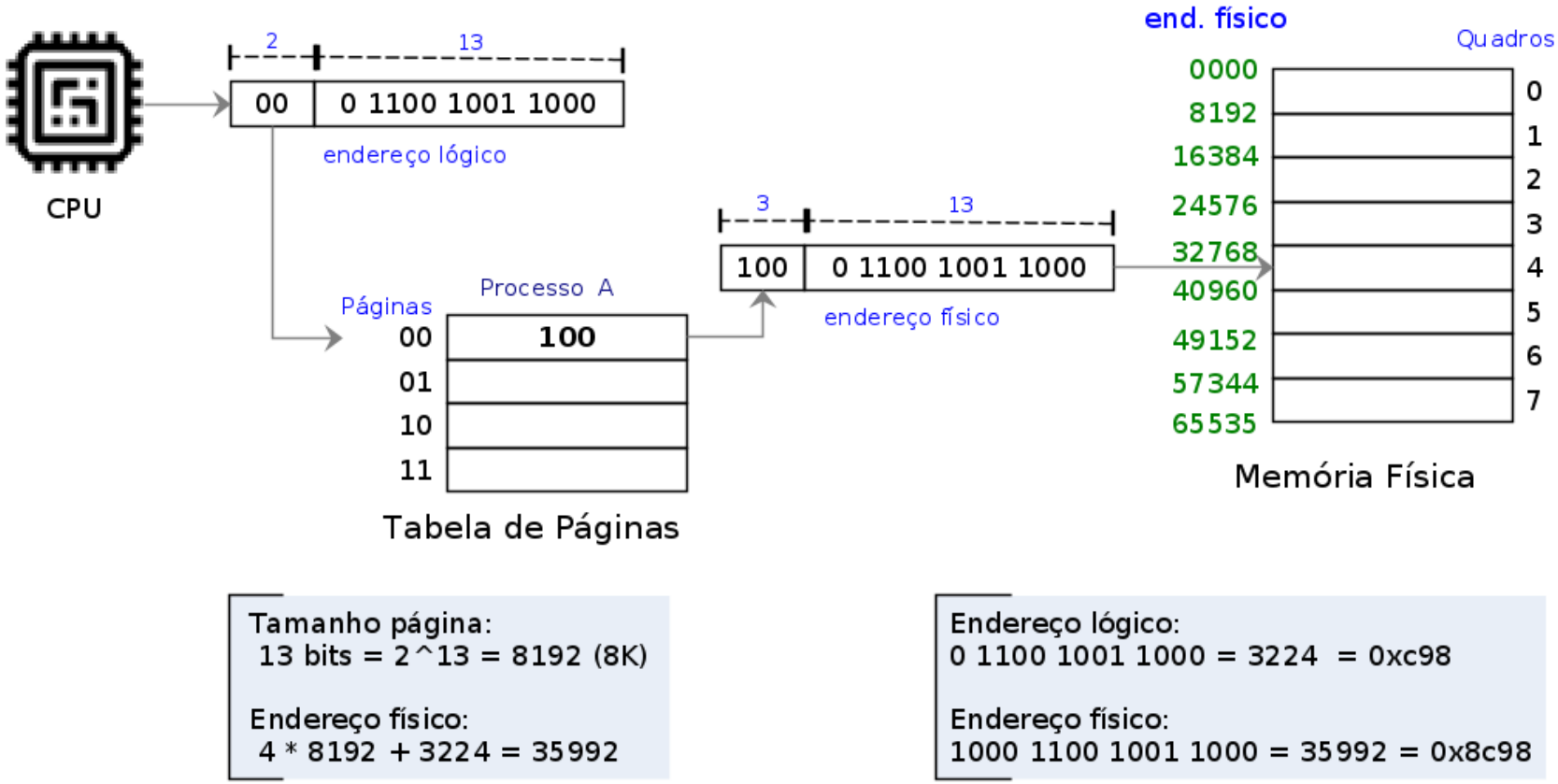


Figura 8: Tradução do endereço lógico 0xC98 para o endereço físico 0x8c98.[2]

Paginação

- A paginação elimina a fragmentação externa e diminui a fragmentação interna.
- Como processos são de diferentes tamanhos, logo o número usado de páginas por processos varia e, por consequência, o tamanho das tabelas de páginas também.
- Questões:
  - Como dimensionar o tamanho da tabela de páginas: fixo ou variável?
  - Como armazenar a tabela de páginas: contíguo em memória (fragmentação externa) ou fazer a paginação na própria tabela?
- Solução: **Paginação Multinível**.

Paginação Multinível

- A paginação multinível divide a tabela em uma estrutura de níveis:
  - Diretórios (níveis).
  - Tabela de páginas
- As entradas dos diretórios apontam para outros diretórios ou para tabela de páginas.
- Cada nível do diretório e cada tabela de páginas deveriam ocupar no máximo um quadro na memória física.
- Os processos devem manter o endereço do diretório global (nível 1).

Paginação Multinível

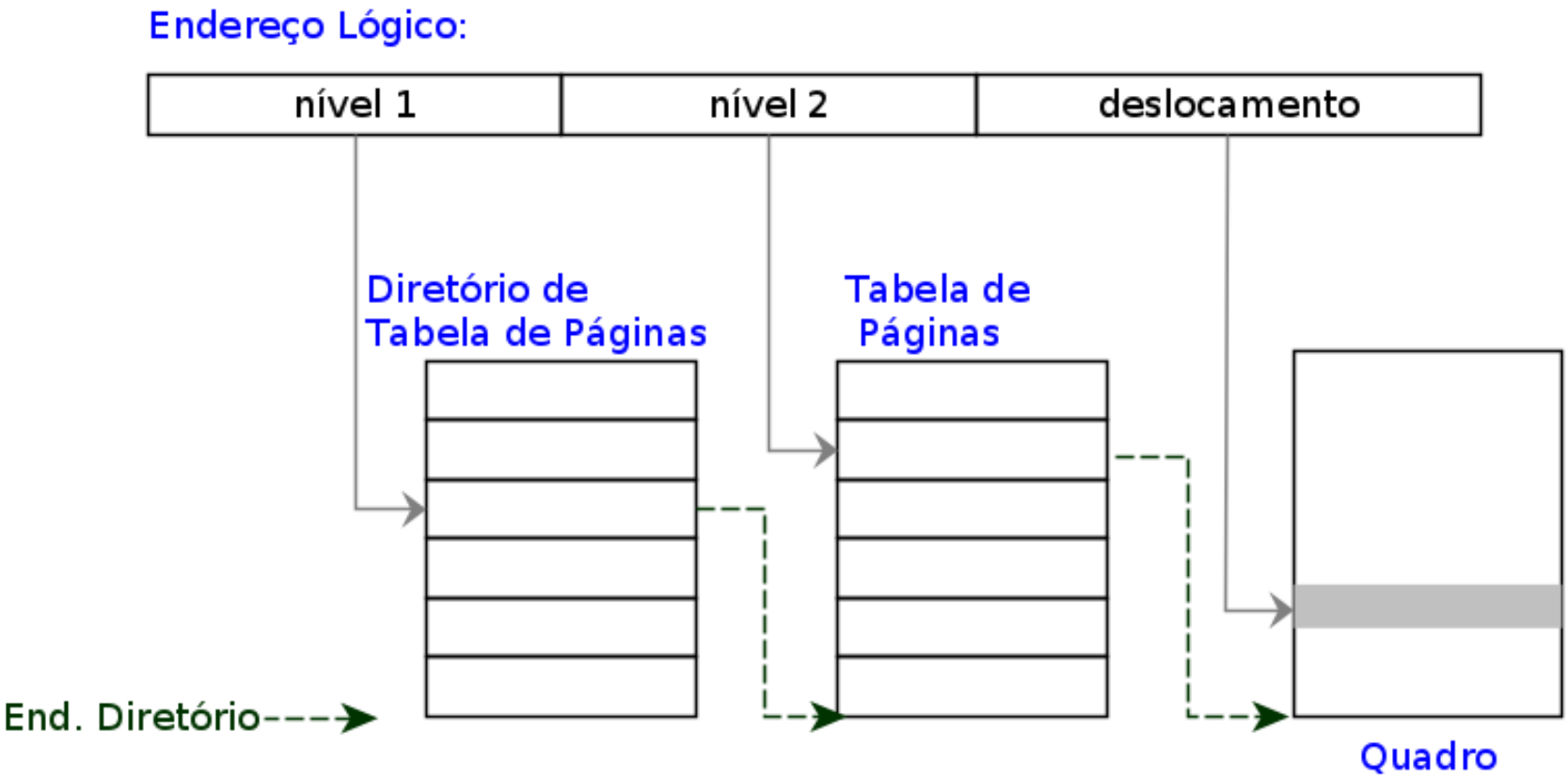


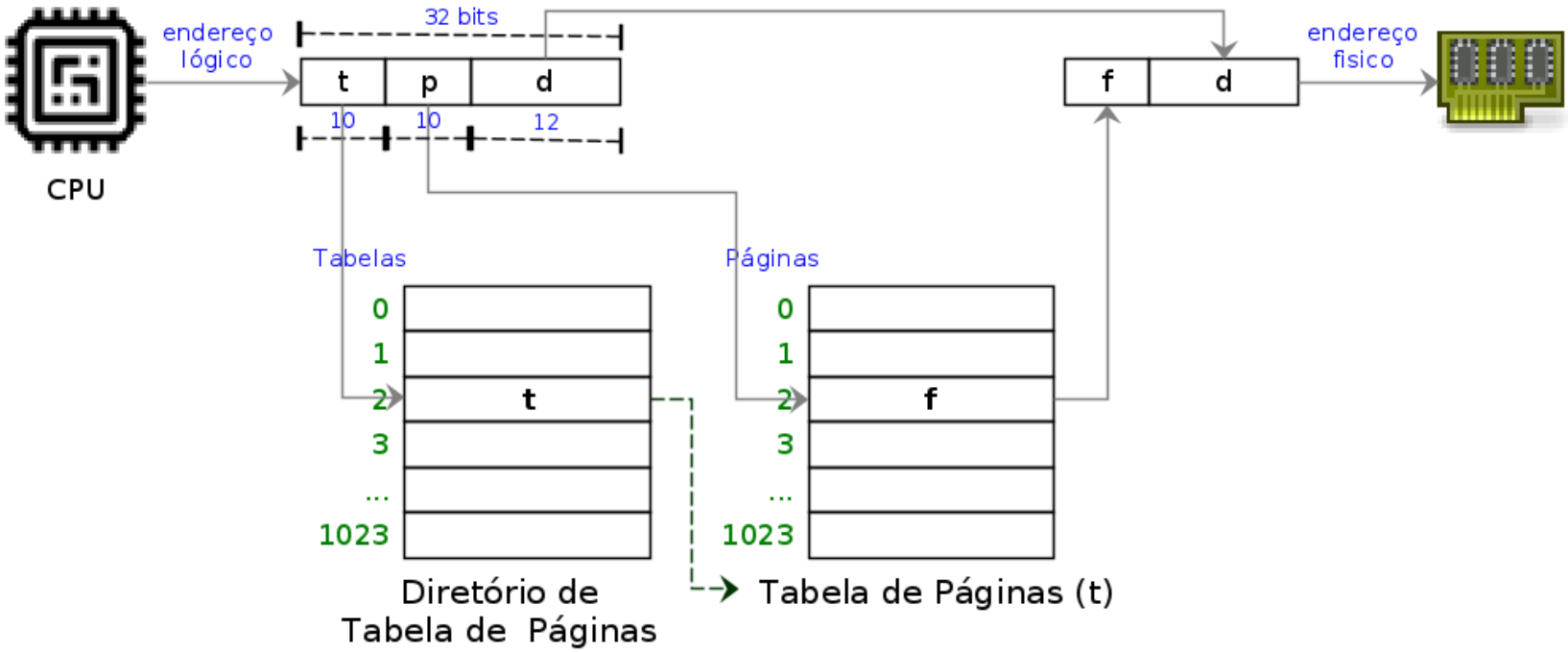
Figura 9: Paginação multinível com dois níveis.

Paginação Multinível: Exemplo

- Considere o sistema com as seguintes características (80x86):
  - Endereço lógico: 4 GiB (32 bits).
  - Tamanho de páginas: 4 KiB (12 bits).
  - Quantas entradas possuiria uma tabela de páginas?
    $\frac{4GiB}{4KiB} = \frac{4 \times 2^{30}B}{4 \times 2^{10}B} = 2^{20} = 1.048.576$
  - Qual o tamanho da tabela de páginas por processo?
    considerando cada entrada: 4 bytes (32 bits)
    número de entradas: $2^{20} = 1M$
    $2^{20} \times 4B = 1M * 4B = 4MiB$
- E com paginação em dois níveis?

Paginação Multinível: Exemplo

- Usando paginação em dois níveis:
  - Endereço lógico: 32 bits
  - Tamanho página: 4 KiB
  - Tabela de páginas: 1024 entradas
  - Diretório de tabela de páginas: 1024 entradas (4 bytes)



Paginação Multinível

- Paginação 3 níveis (arquiteturas 64 bits).

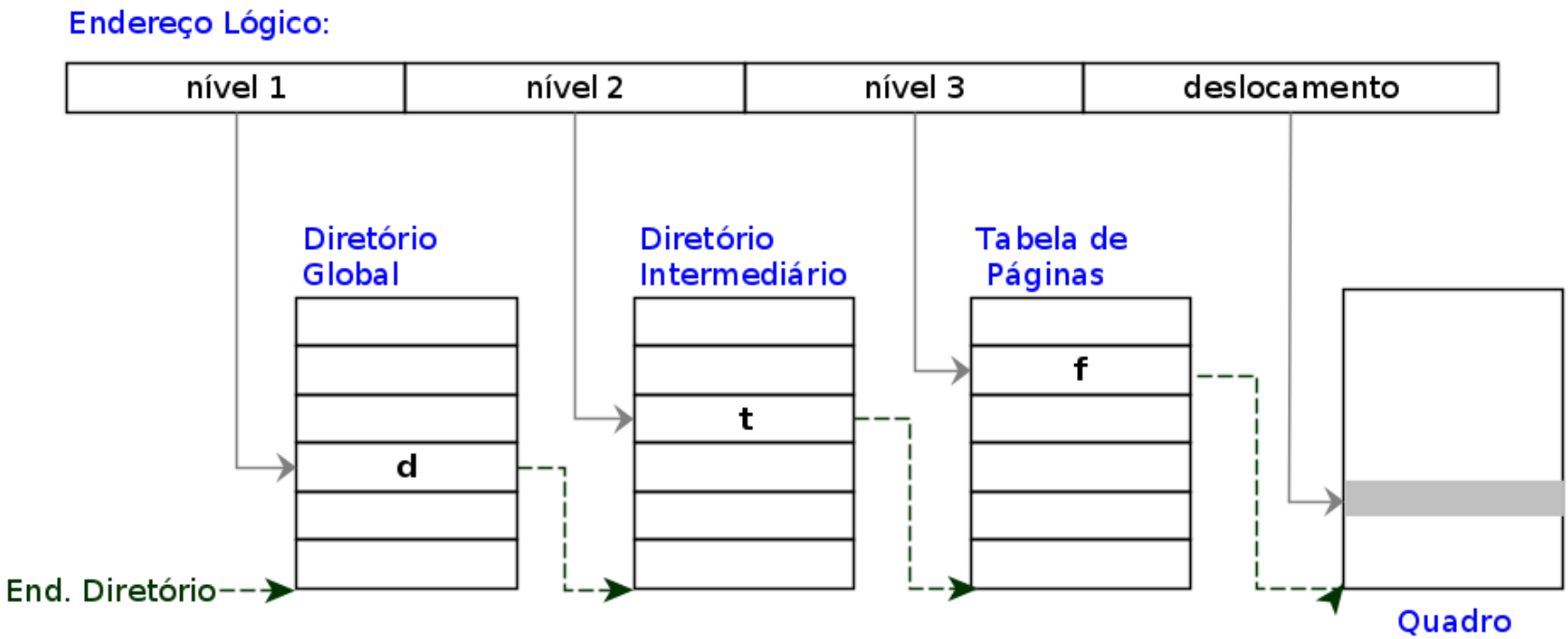


Figura 10: Paginação em três níveis.

Atividade

1. Considere um sistema de memória virtual com memória física de 8GiB, tamanho de página 8KiB e endereço lógico de 46 bits. Suponha que toda tabela de página deve ser armazenada em um único quadro (*frame*). Se o tamanho da entrada de cada tabela é 4 Bytes, então quantos níveis de paginação seriam necessários.[1]

* Dica: Avalie as configurações para 1, 2 e 3 níveis.

[1] https://www.geeksforgeeks.org/multilevel-paging-in-operating-system/ (2019)

Paginação: Proteção

- Proteção de acesso
  - processos acessam somente suas páginas.
  - endereços inválidos somente na última página (fragmentação interna).
- Uso de bits de controle e validade na tabela de páginas.
  - indicação se a página é leitura, escrita, código executável.
  - página pertence ao espaço de endereçamento do processo.

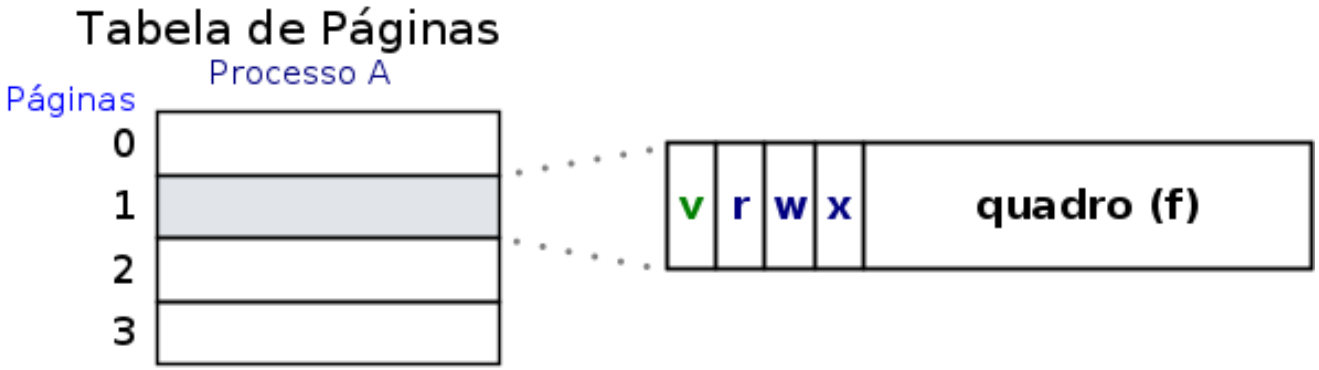


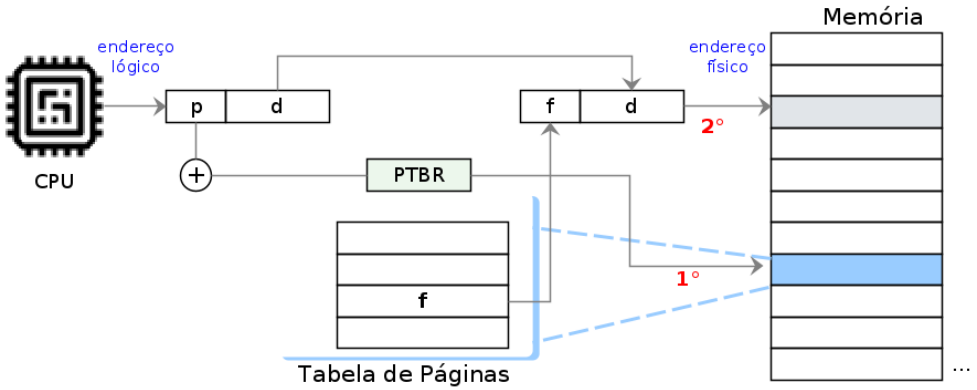
Figura 11: Entrada da tabela de páginas com bits de validade (v), leitura (r), escrita (w) e execução (x).

Paginação: Implementação

- **Forma 1:** Registradores
  - A tabela de páginas é implementada por um conjunto de registradores.
  - Cada página é referenciado por um registrador.
  - BCP deve possuir cópias dos registradores para troca de contexto.
  - Vantagens: acesso rápido para tradução.
  - Desvantagens: número de registradores.

Paginação: Implementação

- **Forma 2:** Memória
  - A tabela de páginas é implementada em memória.
  - *Page Table Base Register (PTBR)*: início da tabela de páginas.
  - *Page Table Length Register (PTLR)*: número de entradas na tabela de páginas.
  - BCP deve manter os dois registradores para troca de contexto.
  - Vantagens: suporta grande número de entradas para tradução.
  - Desvantagens: para cada acesso, são necessários no mínimo dois acessos à memória.



Paginação: Implementação

- **Forma 3:** Translation Lookaside Buffer (TLB)
  - A tabela de páginas é implementada em memória e faz uso de cache (TLB).
  - A cache é composta pr um banco de registradores (memória associativa).
  - Consiste em manter a tabela de páginas em memória com uma parte na cache.
  - A cache é consultada primeiro para a tradução.
  - Página na TLB (hit) e página não está na TLB (miss).
  - Vantagens: melhora o desempenho de acesso a tabela de páginas.
  - Desvantagens: cache de tamanho limitado, memória cara, a TLB é compartilhada entre os processos.

Paginação: Implementação

- **Forma 3:** Translation Lookaside Buffer (TLB) - Exemplo

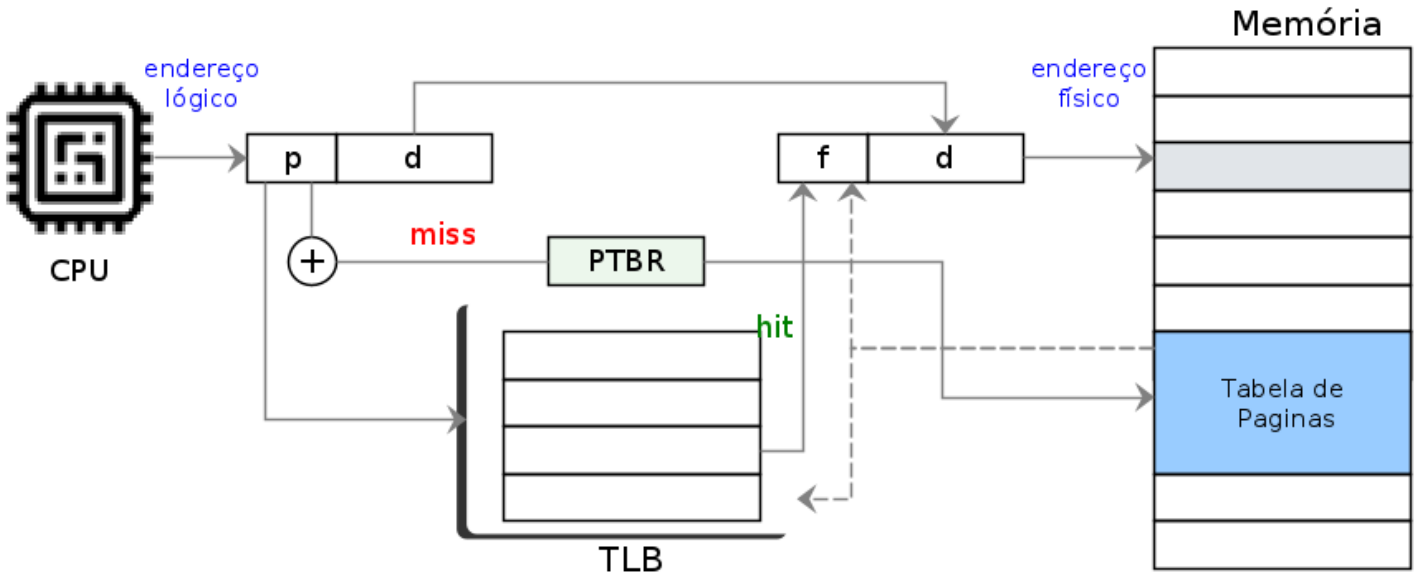


Figura 12: Implementação de paginação com TLB.

Paginação: Implementação

- **Forma 3:** Translation Lookaside Buffer (TLB) - Tempo de acesso
  - Taxa de acerto (*hit ratio* - h): endereço está na TLB.
  - Taxa de erro (*miss ratio*): endereço não está na TLB.
  - Tempo médio de acesso:
    $h(t_{atlb} + t_{amem}) + (1 - h)(t_{atlb} + 2\,t_{amem})$
  - Considere o exemplo:
    $t_{acesso\_tlb} = 20ns$
    $t_{acesso\_mem} = 100ns$
    $t_{acesso(hit)} = 20 + 100 = 120ns$
    $t_{acesso(miss)} = 20 + 100 + 100 = 220ns$
  - Para um hit ratio 80%:
    $t_{medio} = 0.8 \times 120 + 0.2 \times 220 = 140ns$
  - Para um hit ratio 98%:
    $t_{medio} = 0.98 \times 120 + 0.02 \times 220 = 122ns$

Atividade

1. Considere um sistema com gerência de memória que faz uso de TLB. Neste sistema, o tempo de acesso à TLB é de 10 ns e de acesso à memória é 80 ns. Considerando uma taxa de 90% de acerto na TLB, qual o tempo médio de acesso à memória?

Tabela de Páginas Invertida

- Uma tabela para todo o sistema (não por processo).
- Possui uma entrada para cada quadro.
- O endereço lógico é composto por **PID**, **página** e **deslocamento**.
- Cada entrada possui PID e página.
- O indíce da tabela refere-se a um quadro.
- Uma consulta à tabela de páginas devolve o índice associado à entrada.

Tabela de Páginas Invertida

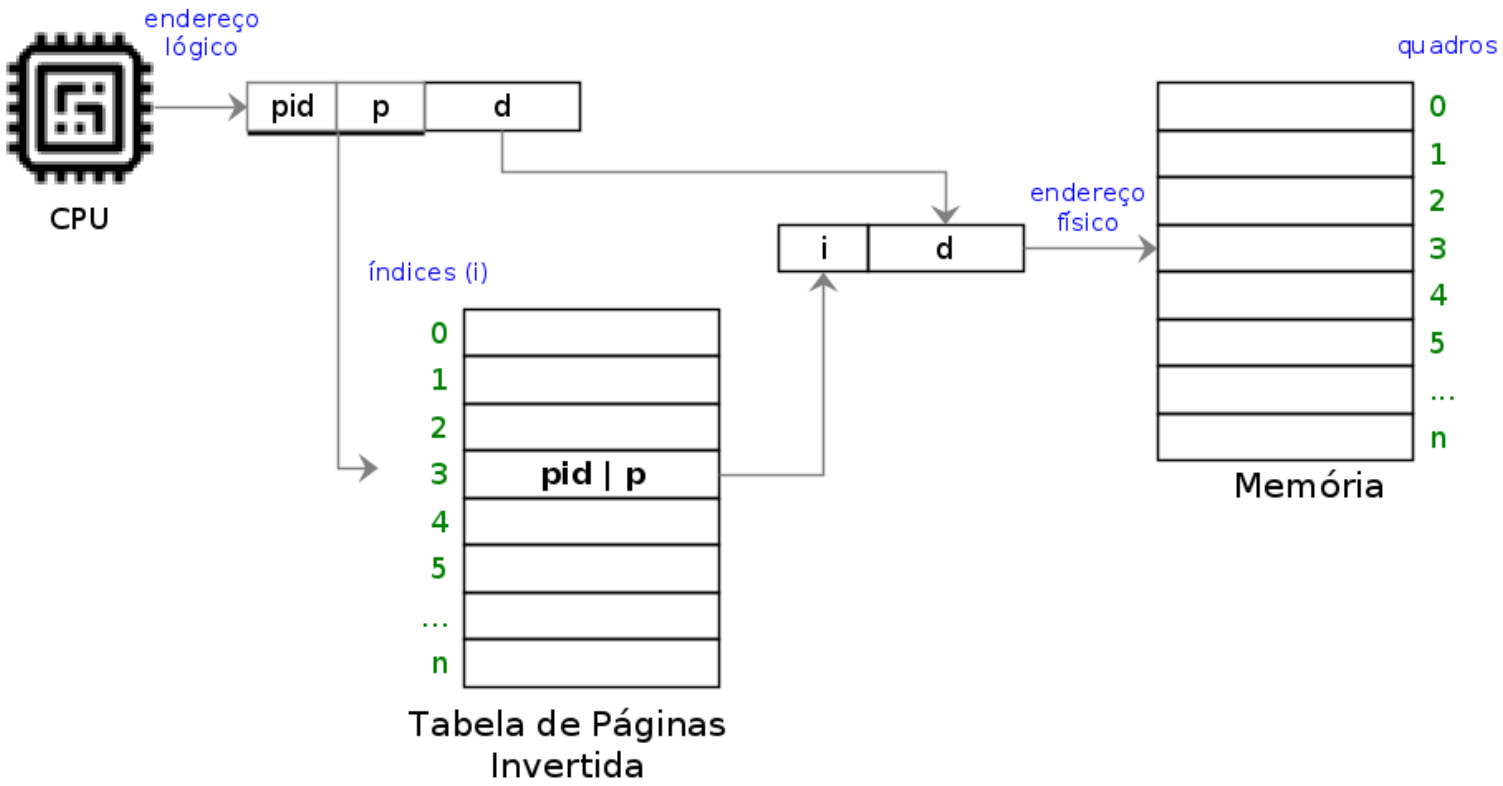
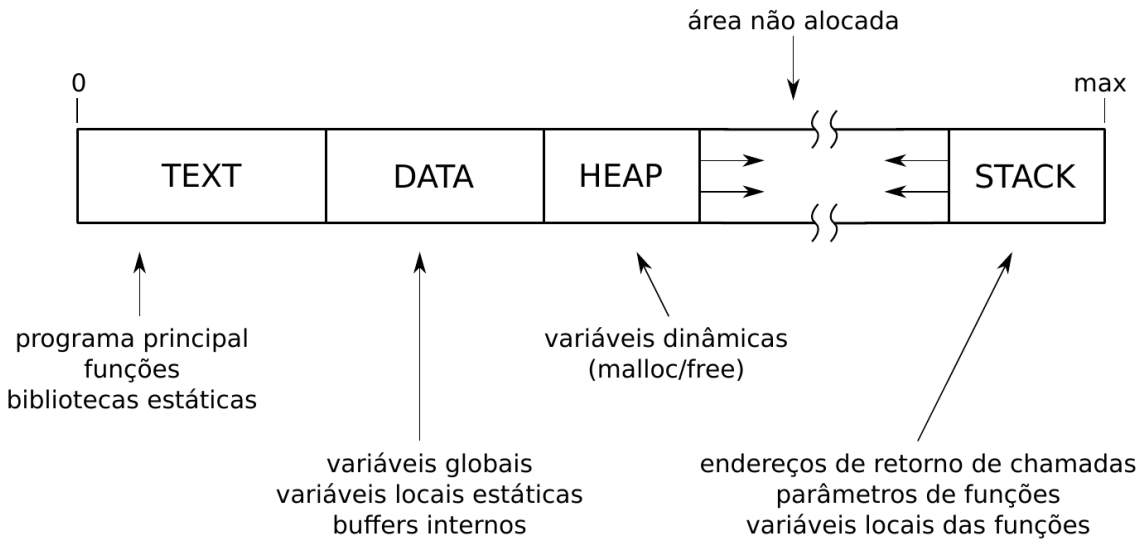


Figura 13: Tabela de páginas invertida para memória dividida em n quadros.

Segmentação

- Espaço de endereço de um processo é dividido em várias regiões (segmentos).
  - Facilita o gerenciamento da memória, pois cada região possui uma finalidade específica.
  - Facilita o compartilhamento de áreas de memória.
- A Gerência de Memória suporta o conceito de Segmentos.



Fonte: [1]

Segmentação

- Endereço lógico composto por duas partes:
  - Número do segmento.
  - Deslocamento no segmento.
- Segmentos podem ser de tamanhos diferentes (há um limite máximo).
- Tradução similar a paginação por meio de uma **Tabela de Segmentos**.
- Cada entrada possui:
  - base: início do segmento (endereço físico).
  - limite: tamanho do segmento.
- Deve-se verificar a cada acesso a validade (via hardware).

Segmentação

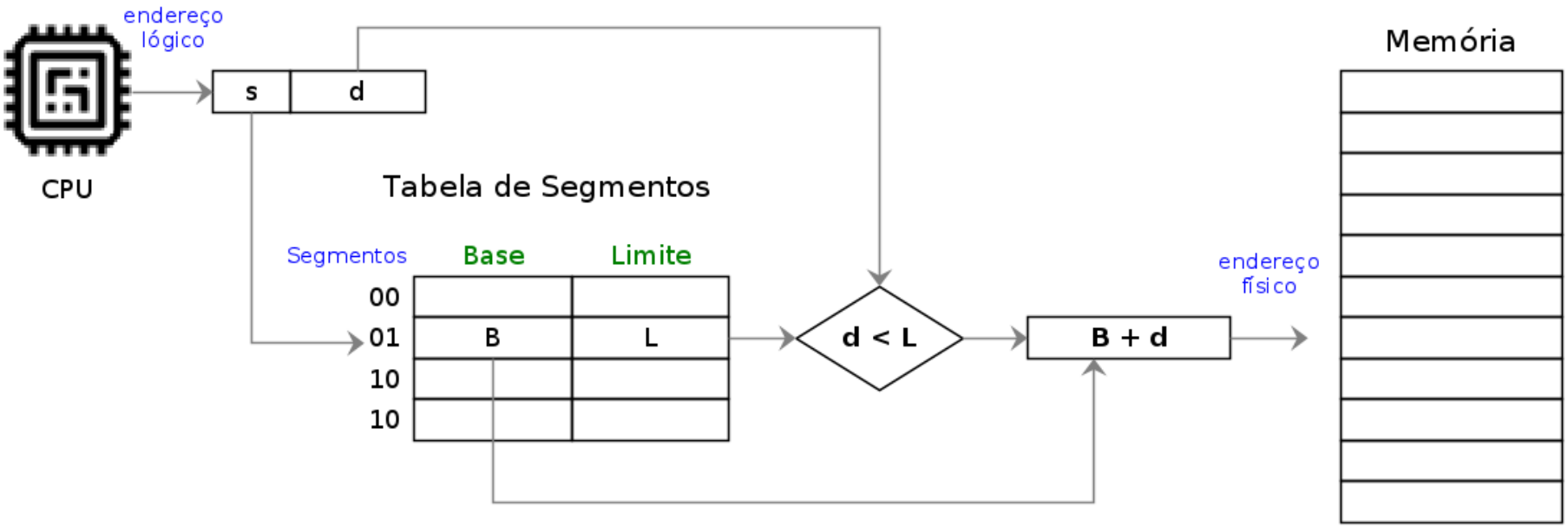


Figura 14: Tradução de endereço lógico para físico.

Segmentação: Exemplo

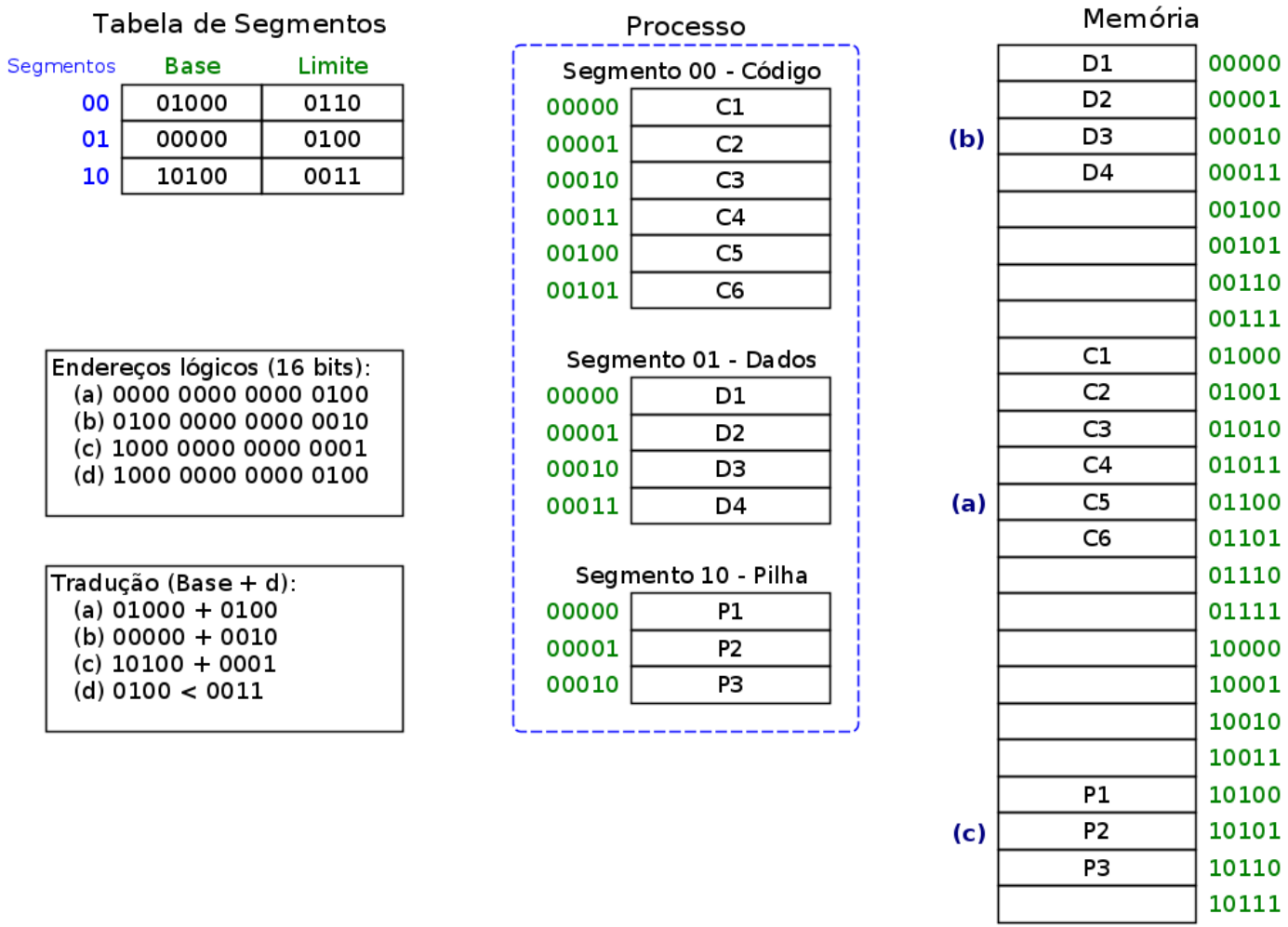


Figura 15: Exemplos de tradução de endereços lógicos para físicos. [2]

Segmentação: Implementação

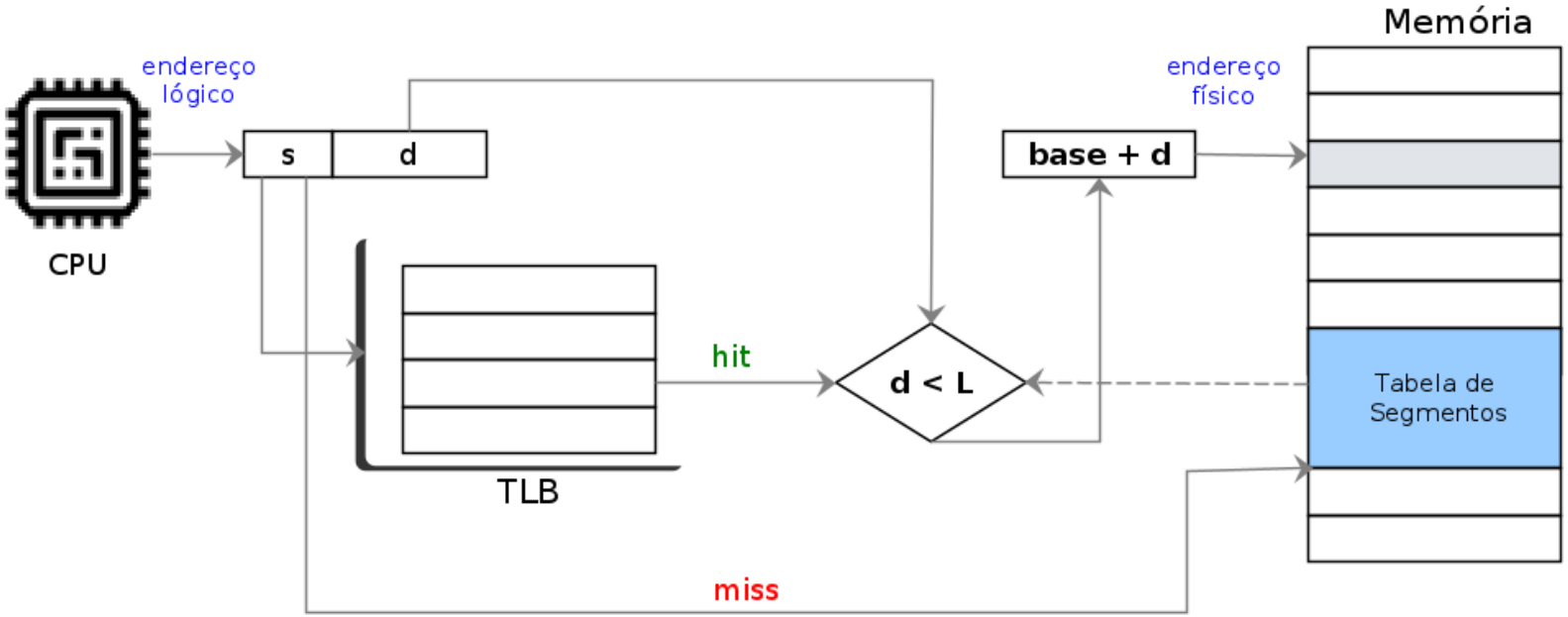


Figura 16: Tradução de endereços lógicos para físicos com a TLB.

Segmentação Paginada

- Cada segmento possui uma tabela de páginas.
- O endereço lógico é composto por:
  - segmento (s): segmento da tabela.
  - página (p): número da página.
  - deslocamento (d): deslocamento na página.

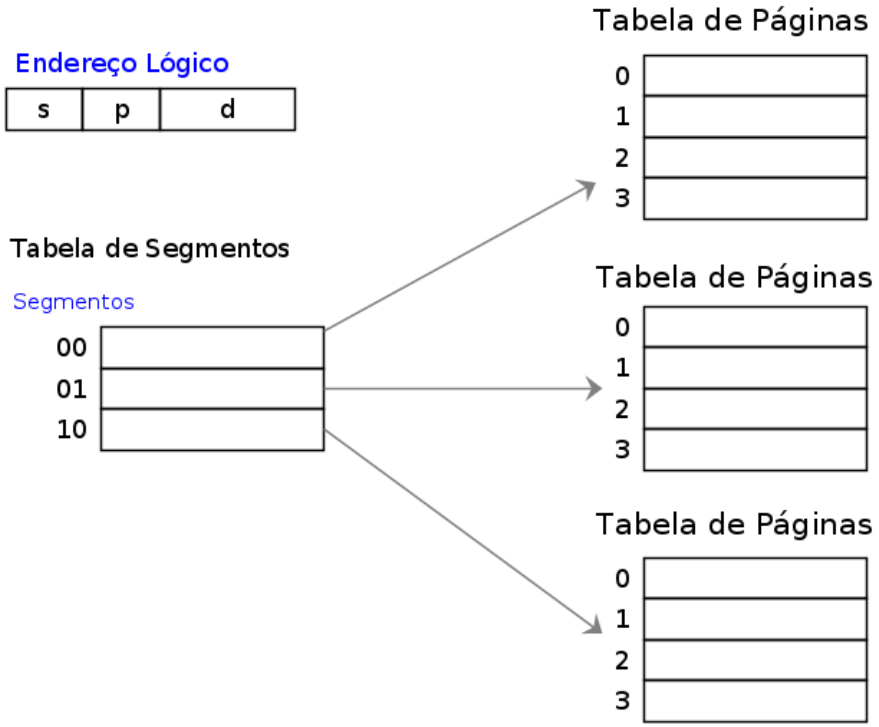


Figura 17: Visão geral e endereço lógico com segmentação paginada.

Segmentação Paginada

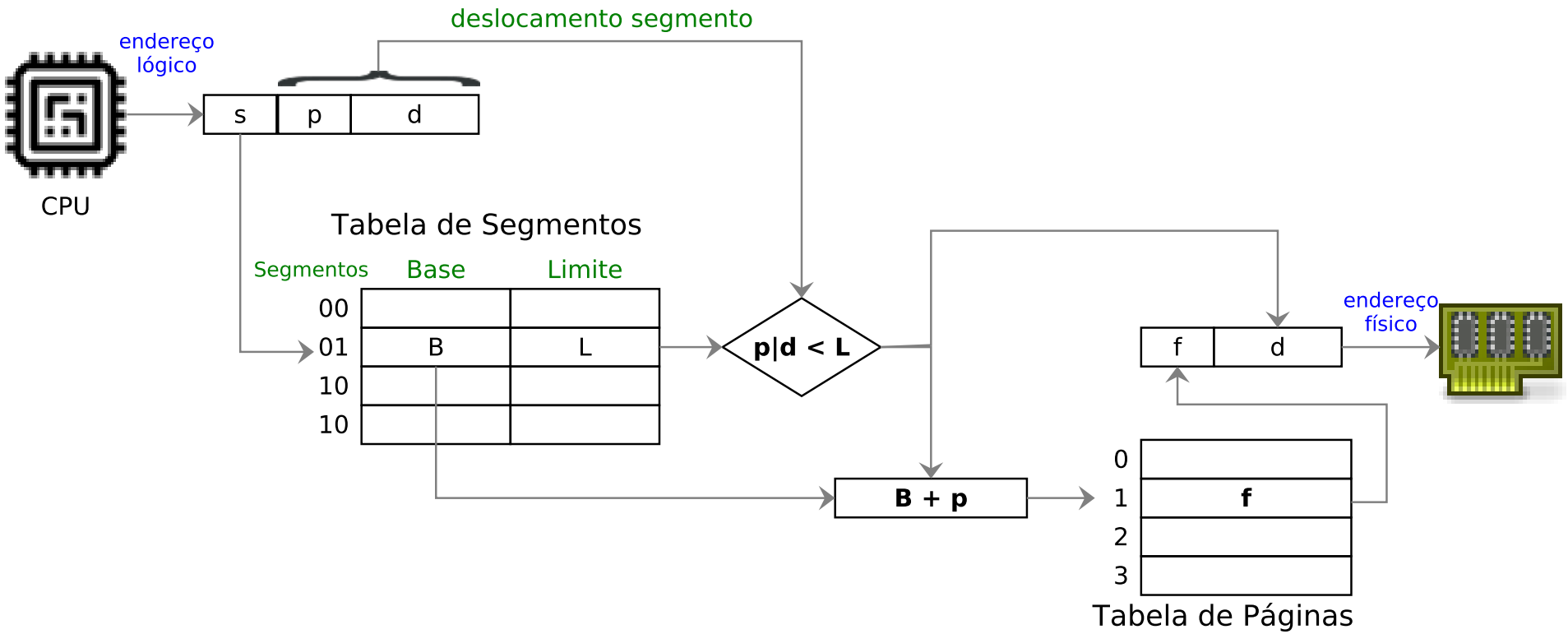


Figura 18: Tradução de endereço lógico para físico.[2]

[2] Fonte: Fonte:https://www.javatpoint.com/os-segmented-paging

Segmentação Paginada

- Vantagens:
  - reduz uso memória.
  - tabela de páginas limitada pelo tamanho do segmento.
  - não há fragmentação externa.
- Desvantagens:
  - ainda há fragmentação interna.
  - tabelas de páginas devem estar contíguas na memória.

Atividades

1. Fazer a lista de exercícios *L07 - Gerência de Memória* disponível na plataforma Moodle.
2. Sugestões de leitura:
   - Capítulos 14 a 16. Sistemas Operacionais: Conceitos e Mecanismos. Maziero [1].
   - Capítulo 3 (Seções 3.1 a 3.3). Sistemas Operacionais Modernos. Tanenbaum [3].

Referências I

[1] Maziero, C. A. (2019). *Sistemas operacionais: conceitos e mecanismos*. online. Disponível em http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=so:so-livro.pdf.

[2] Oliveira, R. S. d., Carissimi, A. d. S., and Toscani, S. S. (2010). *Sistemas operacionais*. Série Livros Didáticos. Bookman.

[3] Tanenbaum, A. S. and Bos, H. (2016). *Sistemas operacionais modernos*. Pearson Education do Brasil, 4 edition.