Sistemas Operacionais

4º período

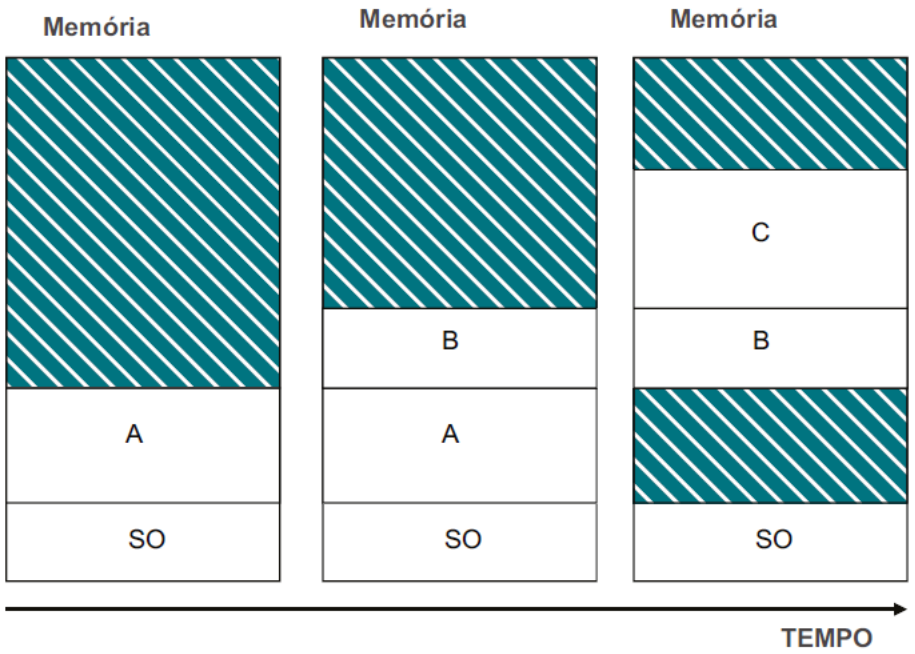
Professora: Michelle Hanne

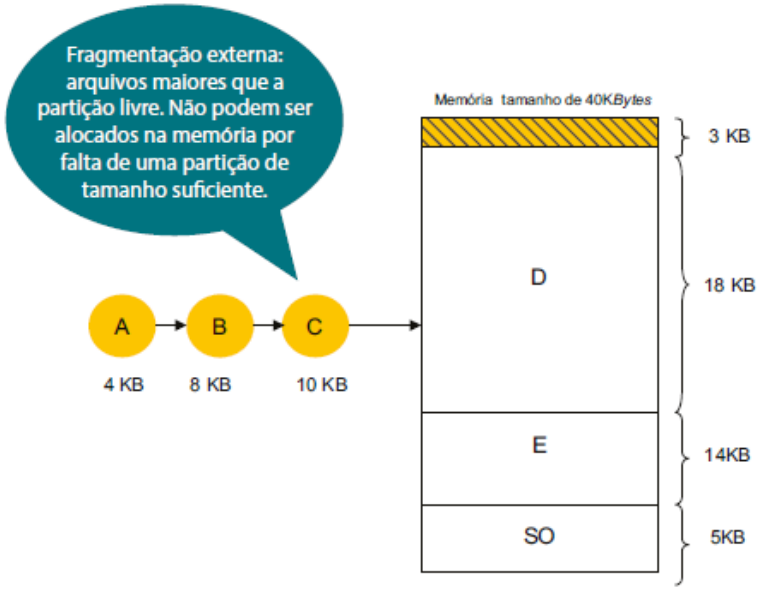
PUC Minas

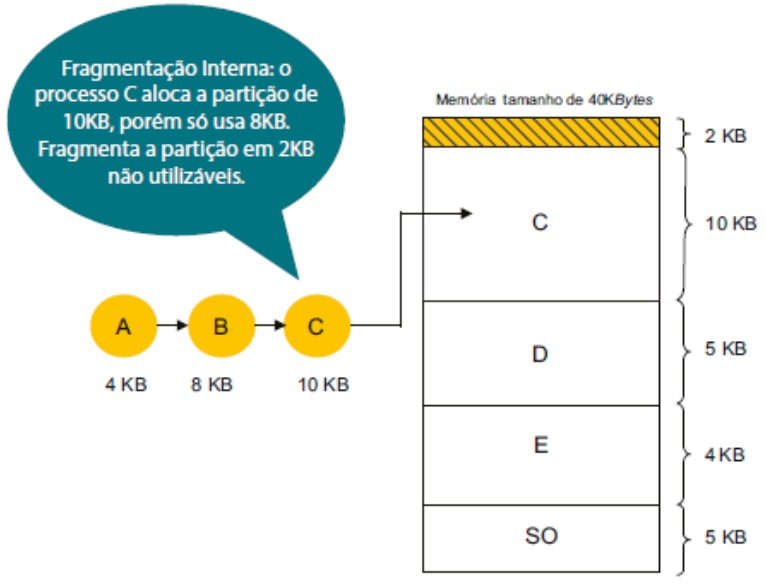
Fragmentação de Memória

Fragmentação de Memória

- A troca de processo **gera fragmentações na memória.** Fragmentação são as áreas de memória livre que não são utilizadas pelos programas por serem menores de tamanho que esses programas em fila.
- **A Fragmentação pode ser interna ou externa**. Ela é **interna** quando o processo não ocupa 100% do espaço livre na memória, e **externa** quando a partição livre tem tamanho menor que o processo a ser executado.







Fragmentação de Memória

PUC Minas

O sistema operacional faz uso de realocação de partição com o objetivo de minimizar o problema da fragmentação.

Esse mecanismo, também conhecido como realocação dinâmica, reduz o problema da fragmentação com alto consumo de recurso, disco (memória secundária) e processador (TANENBAUM, 2016).

Estratégias para Escolha da Partição

PUC Minas

Nos sistemas operacionais, por exemplo, **interativos como sistema para jogos,** diversos processos **concorrentes disputam um espaço de tempo para serem executados na CPU**. Para garantir um melhor desempenho, os Sos implementam algumas técnicas de escolhas de alocação de memória onde os processos ficam armazenados durante o seu estado de pronto.

- Basicamente, três estratégias (**a partição mais justa, a pior partição, a melhor partição**) são utilizadas para determinar em qual partição livre um programa será carregado para execução. O objetivo principal dessas estratégias é **minimizar os problemas causados de fragmentação**.

Estratégias para Escolha da Partição

PUC Minas

Pode-se adotar diversas estratégias para implementar as técnicas, mas a principal é **o tamanho do processo** que está na **fila de processamento**. O sistema operacional possui uma lista de partições (áreas livres), *free list*, que armazena o endereço e o tamanho de cada uma das partições livres

Há algumas técnicas que permitem o sistema operacional escolher a partição para alocação do programa em execução: *best-fit*, *worst-fit*, *first-fit* (MACHADO; MAIA, 2013).

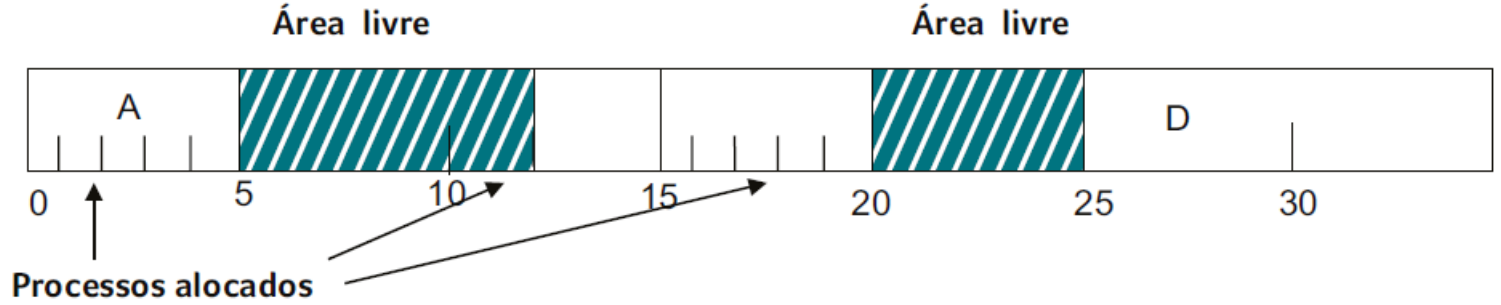


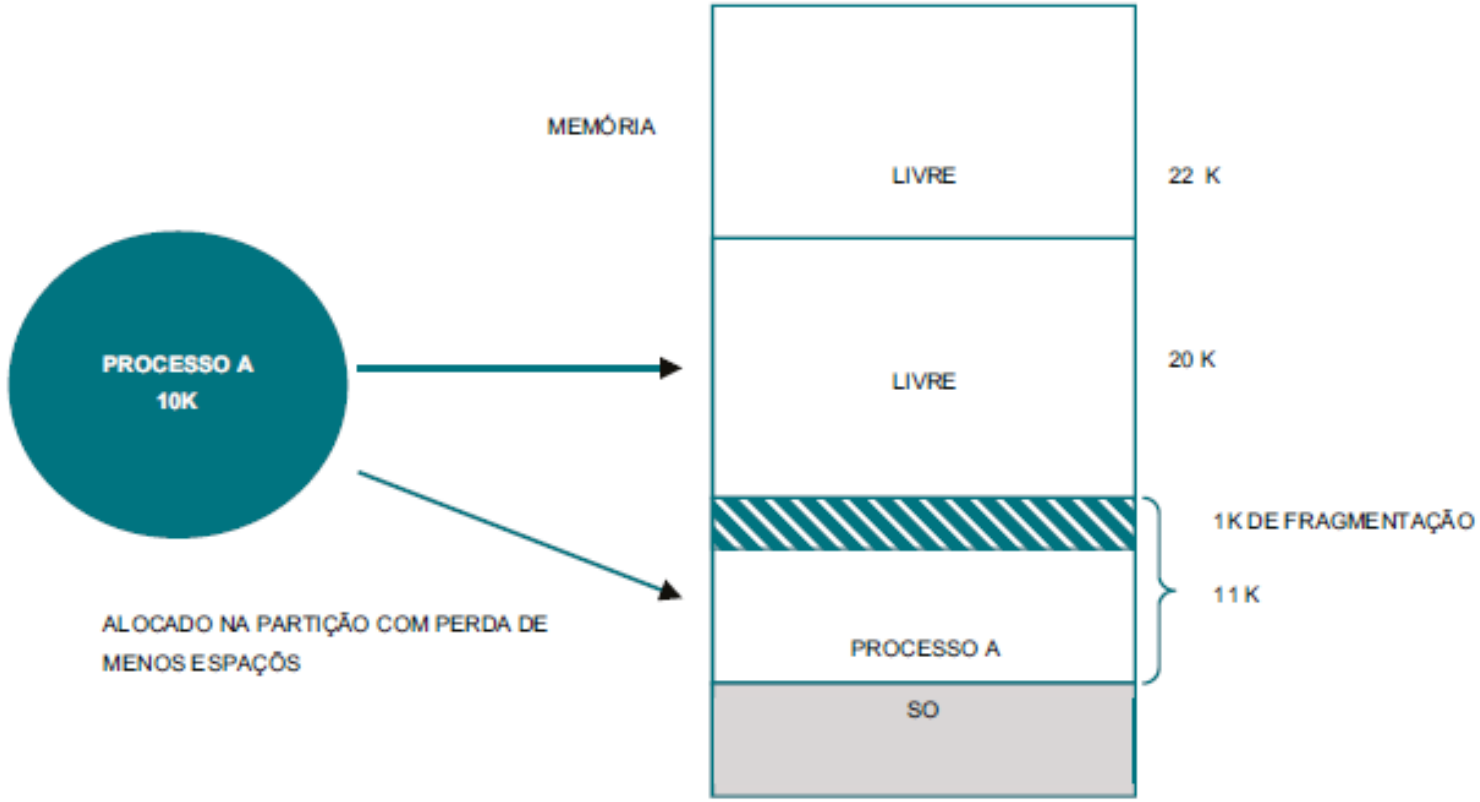
Tabela de endereços de segmento e tamanho

Endereço	Partição
0 – 5	5 K
5 - 12	7 K
12 - 14	2 K
15 - 20	5 K
20 - 25	5 K
25 - 35	10 K

Best-Fit

PUC Minas

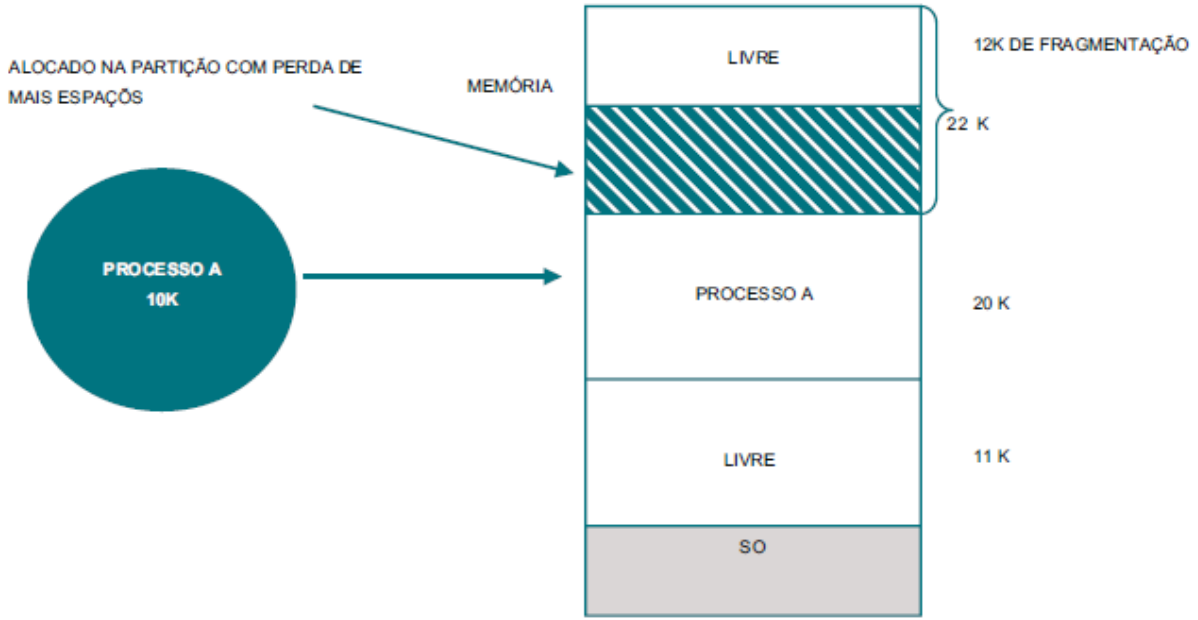
As áreas livres são definidas por meio da escolha da **melhor partição (*Best-fit*)** para armazenamento do programa a ser executada na CPU. A ideia é escolher o programa alocado que desperdice menos espaço interno, ou seja, espaço interno sem utilização.
A ordenação desse algoritmo é por tamanho. **Ele gera problemas de micro fragmentação interna com tendência de existir na memória um grande volume de área não continua livre.**



Worst-Fit

PUC Minas

- As partições escolhidas são aquelas onde os programas deixam mais espaços livres para que outros programas sejam alocados nessas áreas livres, ou seja, **a pior partição (*worst-fit*).**
- Apesar do algoritmo escolher grandes partições que deixam enormes áreas livres para armazenamento, ele diminui o **problema da fragmentação interna**. O processo de 10k chega para ser alocado numa partição, e o sistema operacional determina a maior partição 22k para ser atribuída ao processo. Este algoritmo gera um problema para o caso de ter que alocar processos grandes após a alocação de alguns processos menores.

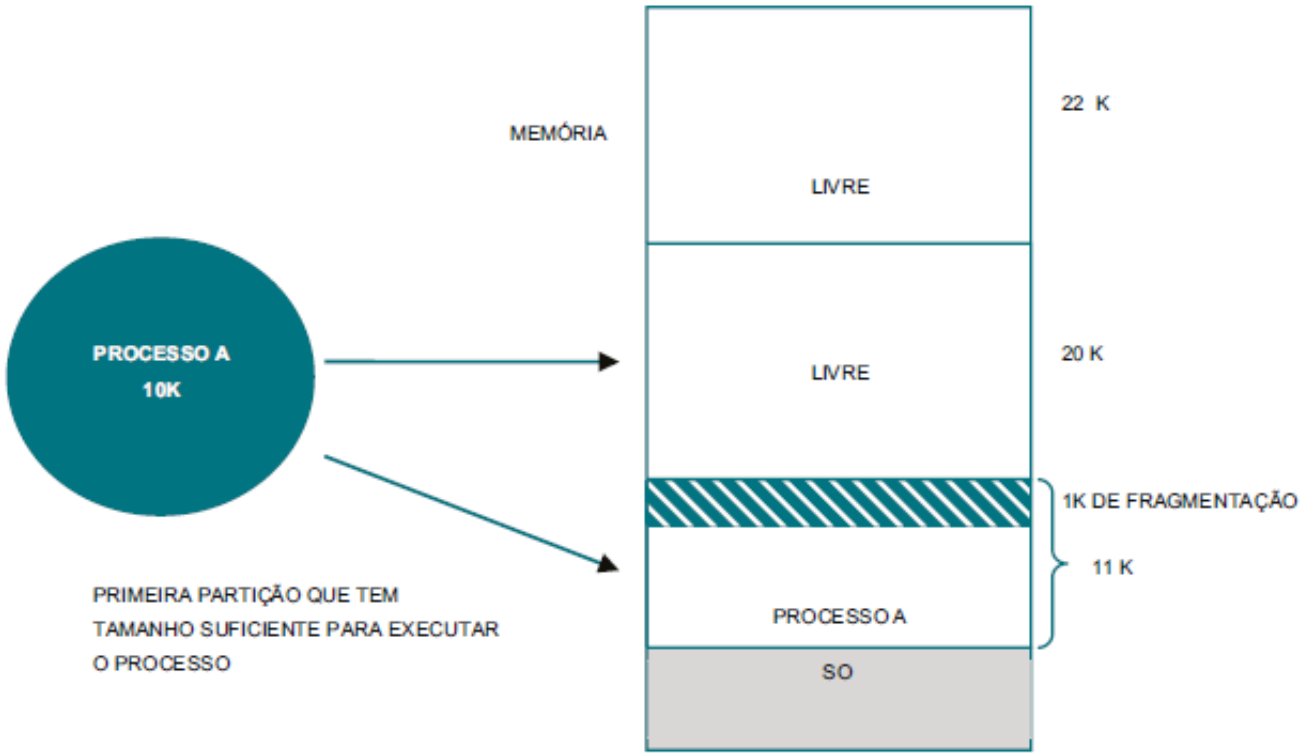


- Neste cenário, provavelmente, não haverá partições grandes o suficiente para comportar estes processos. **Isso pode gerar fragmentação externa**.

First-Fit

PUC Minas

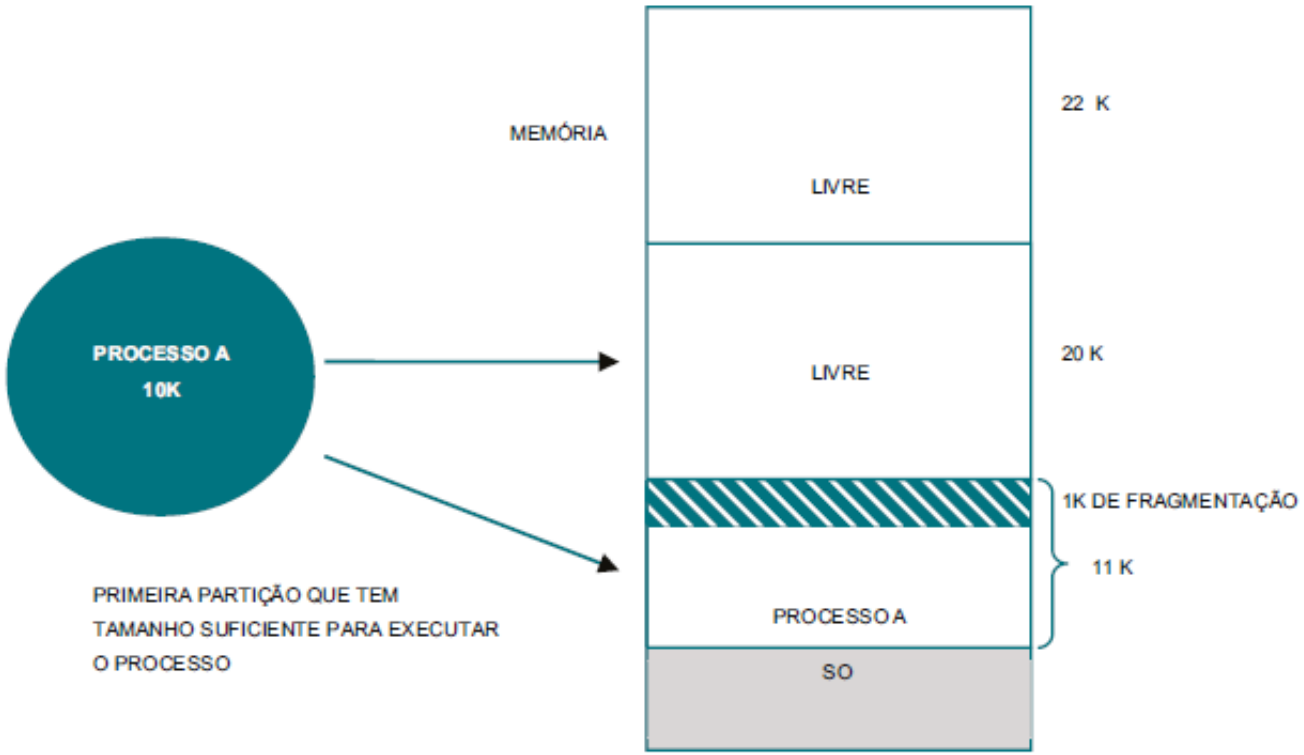
- **Escolhe a primeira partição livre (*first-fit*)** que seja de tamanho suficiente para alocação do programa a ser executado na CPU.
- A ordenação é por endereços crescentes. O método tenta primeiro escolher as partições mais baixas, então, as partições mais altas tendem a ter partições maiores livres.
- **Esse algoritmo é o mais rápido entre eles**, e consome menos recurso do sistema. O sistema aloca a primeira partição encontrada de tamanho suficiente para o processo de 10k.



First-Fit

PUC Minas

- **Escolhe a primeira partição livre (*first-fit*)** que seja de tamanho suficiente para alocação do programa a ser executado na CPU.
- A ordenação é por endereços crescentes. O método tenta primeiro escolher as partições mais baixas, então, as partições mais altas tendem a ter partições maiores livres.
- **Esse algoritmo é o mais rápido entre eles**, e consome menos recurso do sistema. O sistema aloca a primeira partição encontrada de tamanho suficiente para o processo de 10k.



Memória Virtual

PUC Minas

Embora os sistemas multitarefas tenham sido eficientes, **as vezes a lentidão ou travamento ocorriam devido à alocação máxima de memória,** ou seja, programas não eram executados devido à falta de partições livres.

- Isso devido ao tratamento dos programadores em gerenciar/tratar programas com grandes blocos muito maiores que as memórias principais disponíveis para serem executadas.

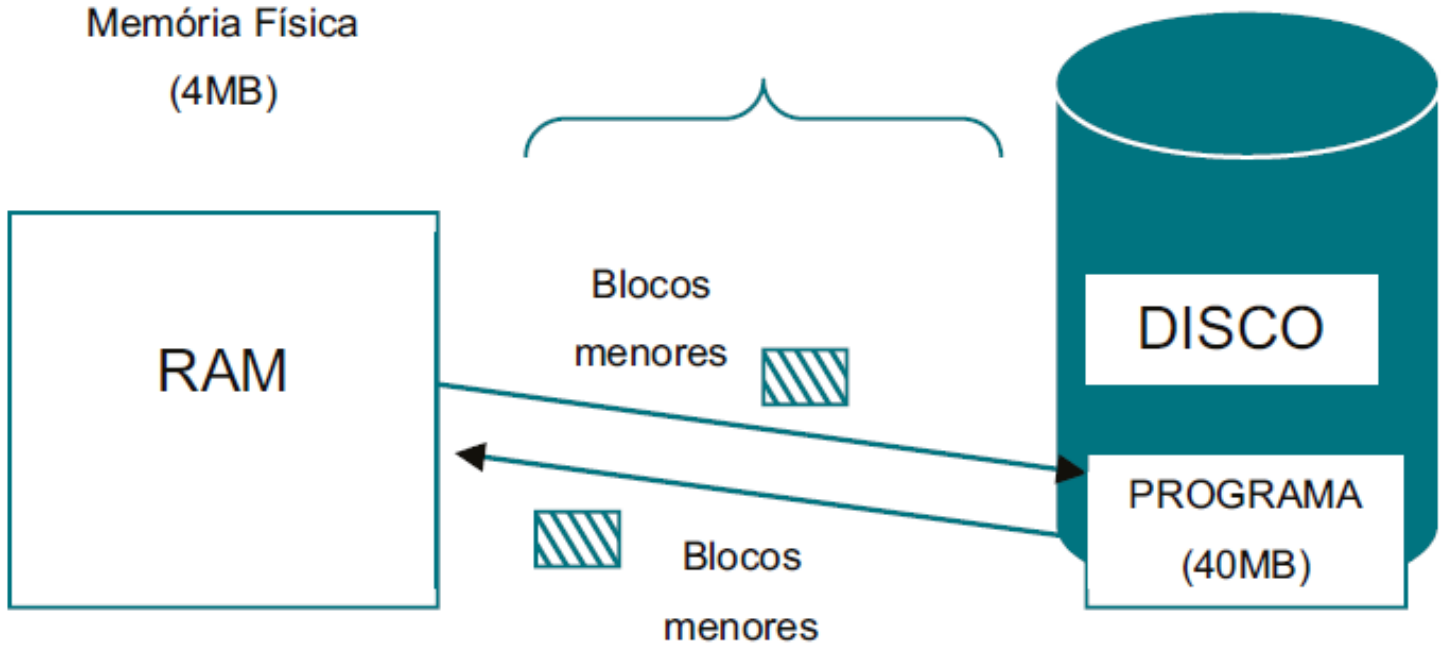
Uma solução inicial foi a técnica de *overlay*. O programa era dividido em módulos, e cada módulo era executado na memória por vez.

Uma melhora da técnica foi transferir para o sistema operacional a responsabilidade de fazer essa troca de memória principal para memória secundária ou memória de disco. Método proposto por John Fotheringham (1961) conhecido por **Memória Virtual.**

Memória Virtual

PUC Minas

Imagine um programa de **40MB usando uma memória de 4MB**. Esse programa tem que ser dividido em partes menores que possam ser alocadas na memória. Então, há troca constante de processos entre a memória e o disco físico.



Todos os tipos de sistemas de computador atual, de *datacenters* a dispositivos *wearable* e biomédicos, dependem da memória virtual.

Memória Virtual

PUC Minas

O processo pela qual o **sistema operacional retira o segmento do programa de memória física e coloca-o em memória de disco é chamado de swapping (troca).** A memória virtual minimiza o problema de fragmentação de memória física (BHATTACHARJEE, 2017).

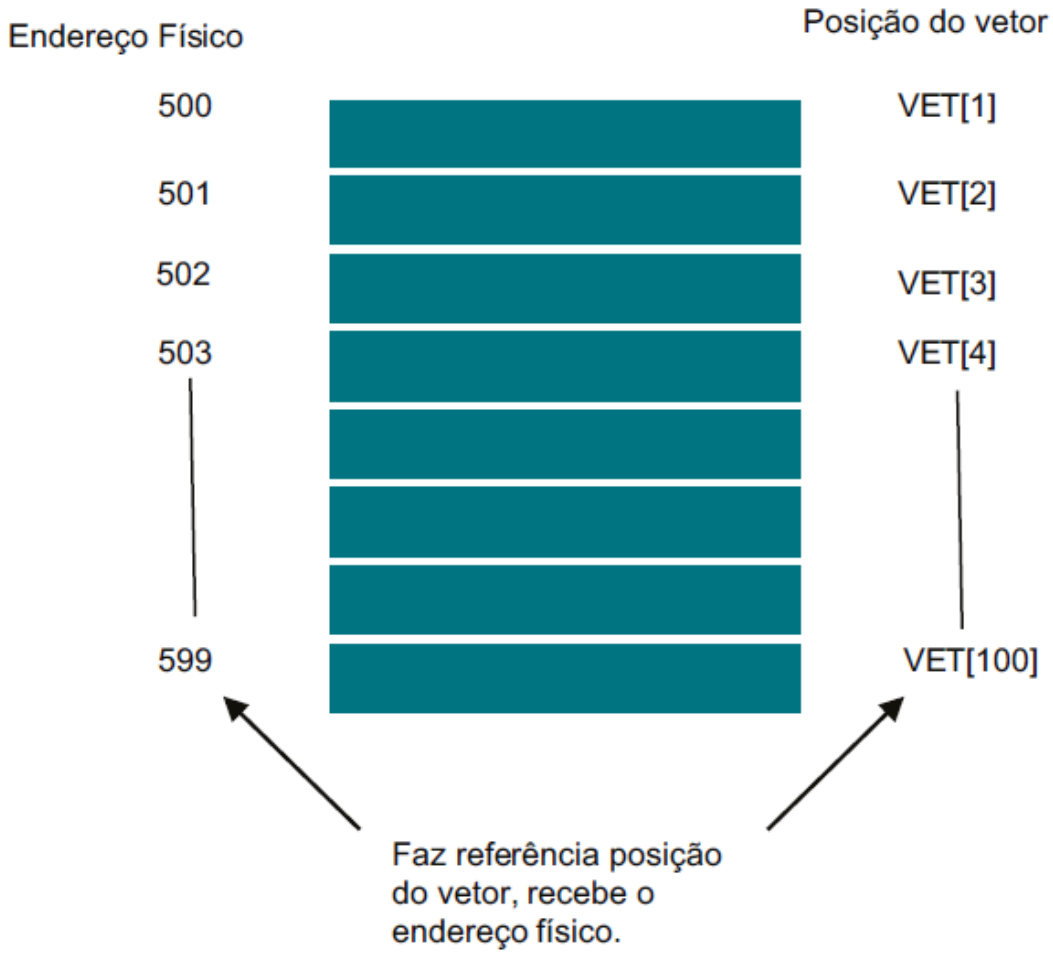
Endereço Virtual

PUC Minas

- A memória virtual faz uma referência parecida como um vetor, ou seja, busca numa matriz os endereços do programa e os dados. Programas virtuais não fazem referência aos endereços físicos de memória principal, mas aos endereços virtuais.

- Há uma tradução no momento da compilação de um endereço virtual para o físico, porque o processador somente acessa posições de memória RAM. **A tradução de endereços lógicos para endereços físicos é chamada de mapeamento.**

- Devido ao fato de termos dois tipos de endereços: lógico e físico, o conjunto desses endereços é chamado de **endereçamento virtual e endereçamento real.**



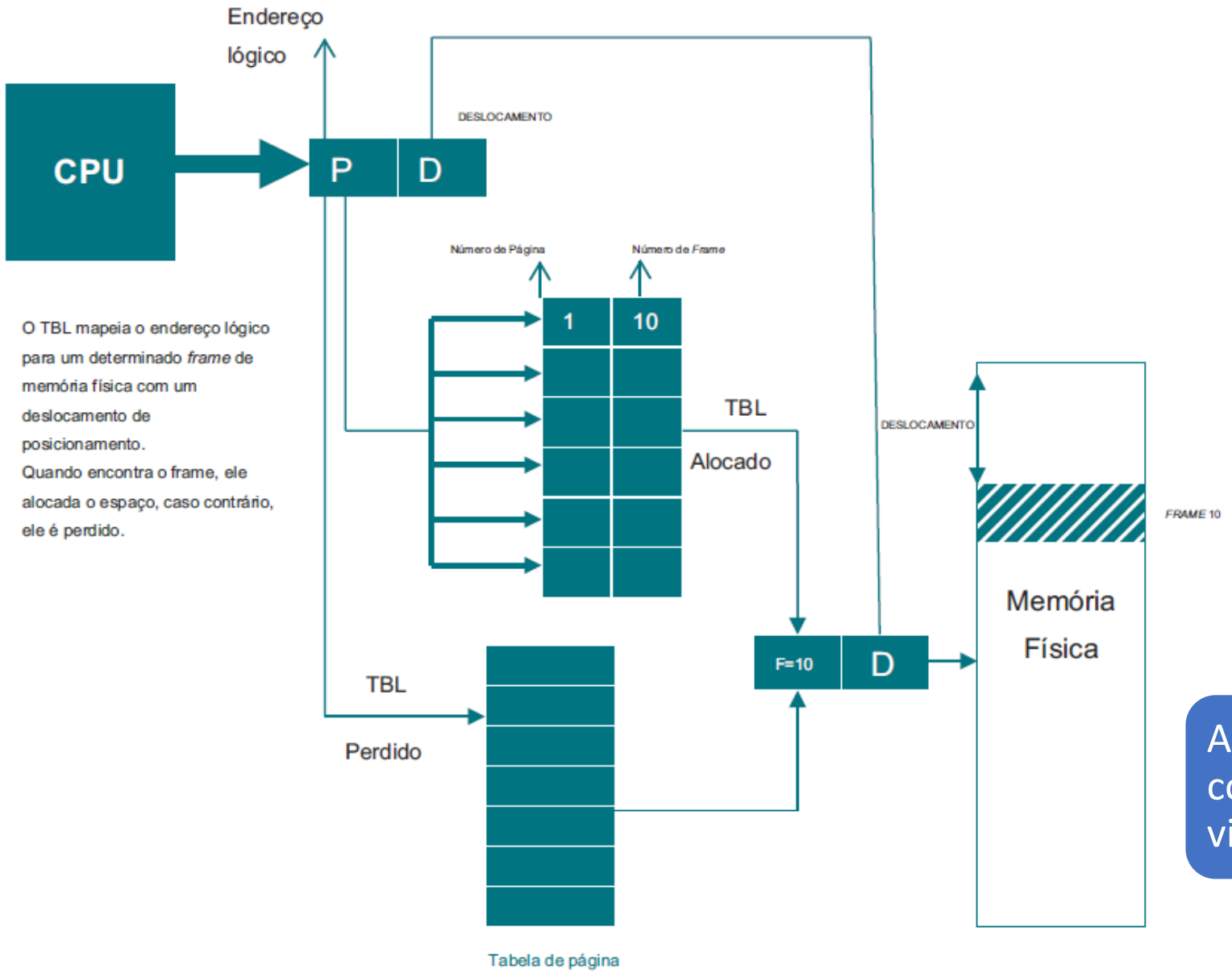
Mapeamento de Endereço Virtual

PUC Minas

O sistema operacional cuida da tradução por meio do mapeamento, ou seja, **traduzir endereços lógicos em endereços reais**. Essas operações são realizadas de tal forma que não comprometa o desempenho e que tenha transparência para o usuário e aplicações.

No apoio ao sistema operacional com objetivo de melhorar o desempenho dessa tradução, um hardware especial, denominado **memória associativa** ou ***translation lookaside buffer*** *(tbl)* foi introduzido para mapear os endereços virtuais para endereços físicos (CRUZ *et al*., 2015).

Mapeamento de Endereço Virtual



- Há uma tabela de tradução que relaciona o endereço lógico com o número da página (*frame*). O *frame* juntamente com o deslocamento determina a posição na memória RAM, ou seja, o endereço físico.
- O deslocamento é a posição de memória que inicia a partição, ou seja, a posição inicial da área de memória que será alocada ao processo.

A cada acesso de memória, uma comunicação entre endereços de memória virtual e endereços físicos deve existir

Mapeamento de Endereço Virtual

PUC Minas

Na operação de tradução de endereços, o sistema operacional **armazena as tabelas de páginas na memória principal**. No entanto, para reduzir a sobrecarga imposta pelos acessos à memória na tabela de páginas, uma **memória *cache* especial, chamada de TLB, é responsável por armazenar as entradas de conversão da tabela de páginas para as páginas acessadas mais recentemente.**

O número de blocos é a divisão do tamanho de endereçamento virtual pelo o tamanho de blocos.
Exemplo: 2^32 / 2^9 = 2^23
2^64/2^16 = 2^48

Espaço de endereçamento virtual	Tamanho de bloco	Número de blocos	Número de entradas na tabela de páginas
2^{32}	512 Bytes	2^{23}	2^{23}
2^{32}	4 Kbytes	2^{20}	2^{20}
2^{64}	4 Kbytes	2^{52}	2^{52}
2^{64}	64Kbytes	2^{48}	2^{48}

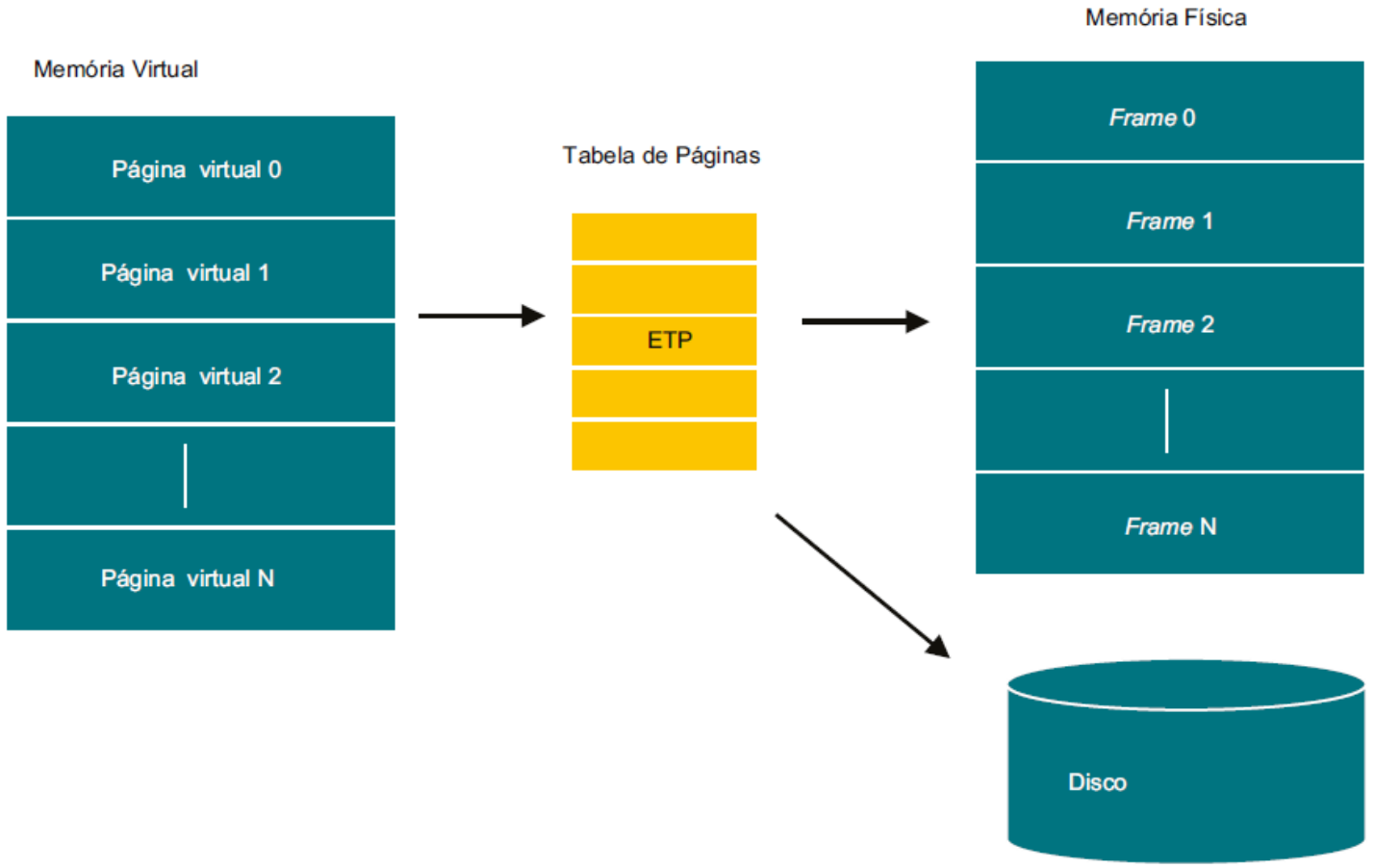
PUC Minas

Paginação

Paginação

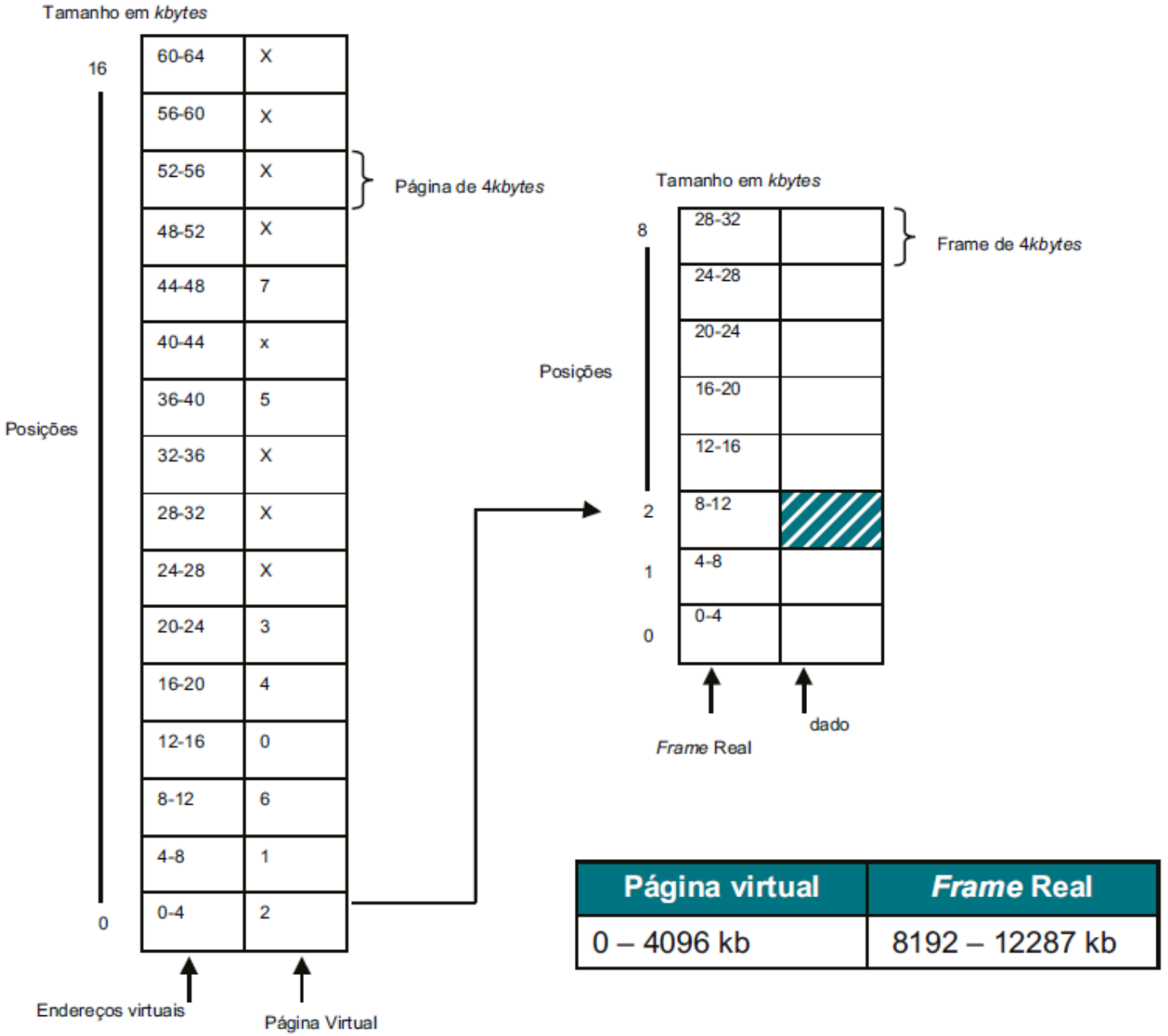
- Essa técnica divide **o espaço de endereçamento virtual e espaço de endereçamento real em blocos de mesmos tamanhos chamados de página.**
- Páginas **no espaço virtual são chamadas de páginas virtuais** e no espaço real são chamadas de ***frames* ou páginas reais.**

- Todo o mapeamento é realizado por meio da tabela de páginas. **Cada página virtual de cada processo possui uma página real** correspondente na tabela de páginas, **Entrada na Tabela de Páginas (ETP).**



Ao executar um programa, as páginas virtuais são transferidas de disco para memória RAM e colocadas em *frames*.

Paginação



Página virtual	*Frame* Real
0 – 4096 kb	8192 – 12287 kb

Suponha uma memória de 32KbBytes e uma memória virtual de 64*KBytes* e cada frame ou página de memória de 4*kBytes*.

Então, quando o sistema operacional precisa escrever algo na página 0, ele verifica a referência do endereço físico (*frame* 2) na entrada da tabela de página e escreve o dado na posição correta de memória física
(STUART, 2011)

Page Fault

PUC Minas

Quando o processo faz referência a uma de suas páginas, e esta não está mapeada pelo gerenciamento de memória de endereço virtual, cria-se uma falha de exceção gerada pelo hardware do computador, denominado ***page fault (falha de páginas)***.

A taxa de *page fault* gerada por cada programa depende do modo que a aplicação foi desenvolvida e da gerência de política de memória implementada pelo sistema operacional. O ideal é **manter o número mínimo de páginas em memória principal para que o número da taxa de paginação seja reduzido ao máximo e não cause problemas para os demais processos** (STUART, 2011).

Page Fault

PUC Minas

Working set

- é um conceito que originou da análise da taxa de paginação dos processos e tem por definição o conjunto de páginas referenciadas por um processo durante determinado intervalo de tempo.

Princípio da Localidade

- Localidade é a tendência de o programa fazer referência a posições de memória de forma uniforme, ou seja, o processo concentrará suas referências no mesmo conjunto de páginas.

Exemplo:

- Imagine um loop, a tendência de executar no mesmo conjunto de páginas é enorme.
- Um dos maiores problemas na gerência de memória por paginação não é decidir que página carregar na memória, mas quais serão as páginas removidas. Qualquer estratégia de realocação de páginas deve considerar se uma página foi ou não modificada, senão pode haver perdas de dados armazenados.

PUC Minas

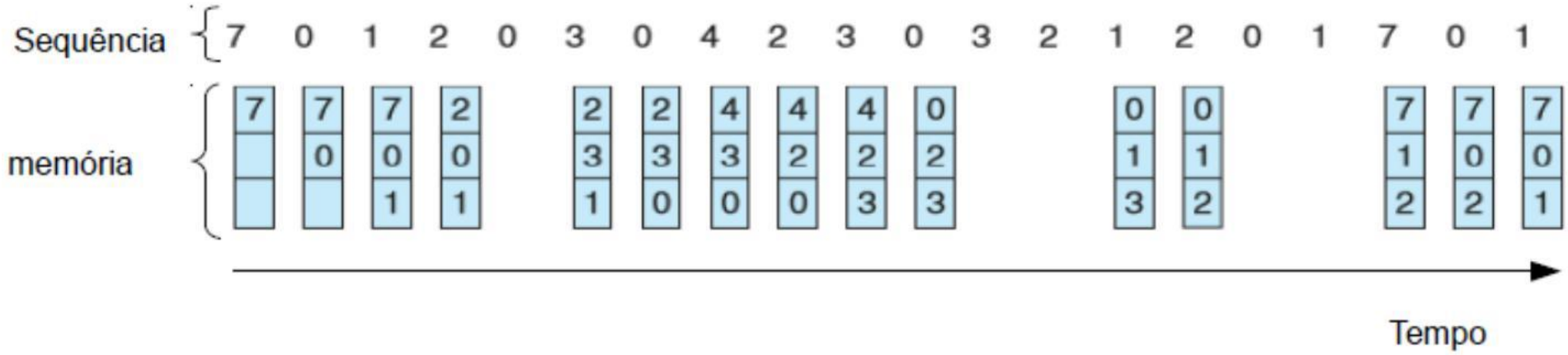
Algoritmos de Paginação

Substituição de Páginas

PUC Minas

FIFO - **A página que primeiro for utilizada será a primeira a ser descartada**. Pode descartar páginas importantes.

- Memória com 3 frames (quadros)
- Frame tem tamanho igual a uma página virtual



Total de requisições = 20
Total de falhas = 15
Taxa de falha = 15/20 = 0,75

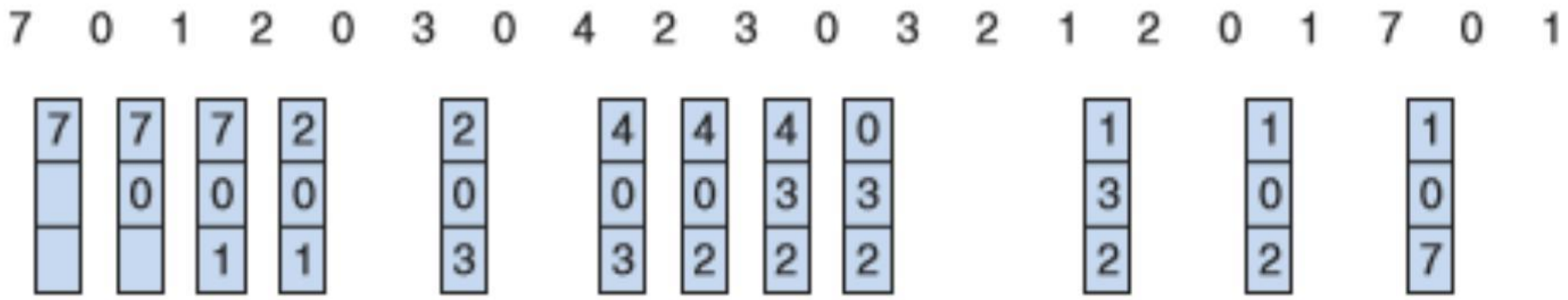
https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/9_VirtualMemory.html

Substituição de Páginas

PUC Minas

LRU - *Least-recently-used* - A página que está a mais tempo sem ser referenciada.

- FIFO como critério de desempate
- Memória com 3 frames (quadros)
- Frame tem tamanho igual a uma página virtual



- Total de requisições = 20
- Total de falhas = 13
- Taxa de falha = 13/20 = 0,65

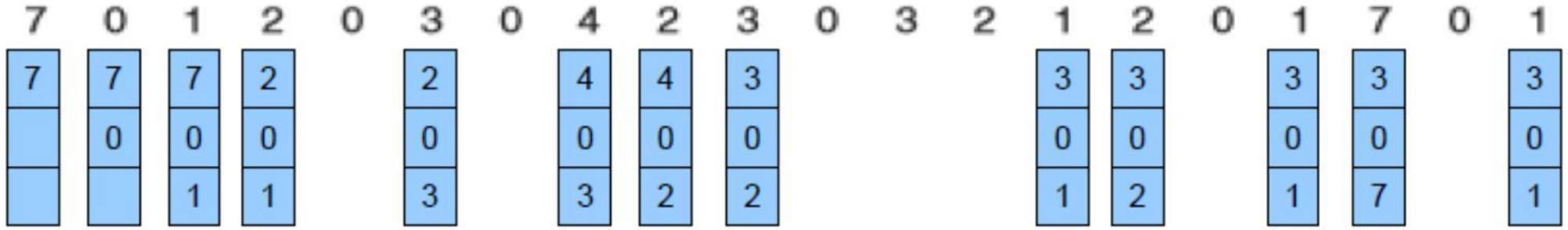
https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/9_VirtualMemory.html

Substituição de Páginas

PUC Minas

LFU - *(Least Frequently Used - Página menos frequentemente utilizada*) escolhe a página que foi menos acessada dentre todas as que estão carregas em memória

- FIFO como critério de desempate
- Memória com 3 frames (quadros)
- Frame tem tamanho igual a uma página virtual



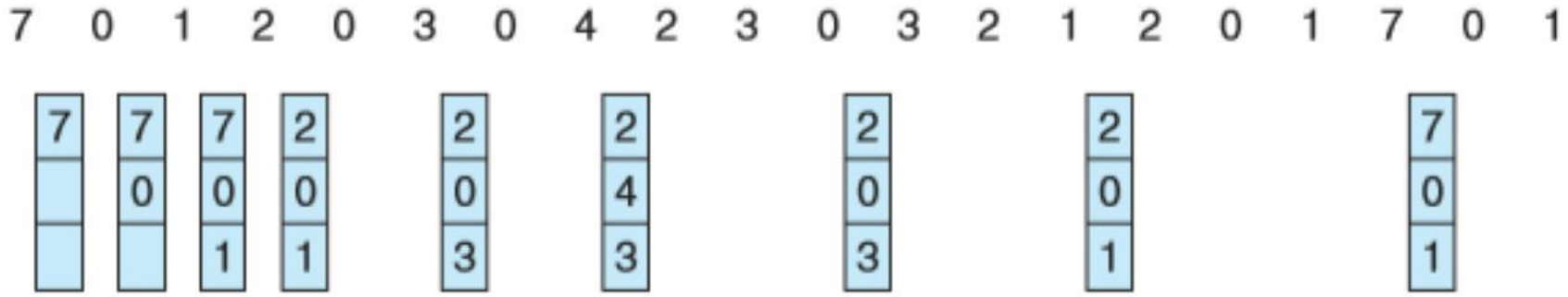
- Total de requisições = 20
- Total de falhas = 13
- Taxa de falha = 13/20 = 0,65

https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/9_VirtualMemory.html

Substituição de Páginas

PUC Minas

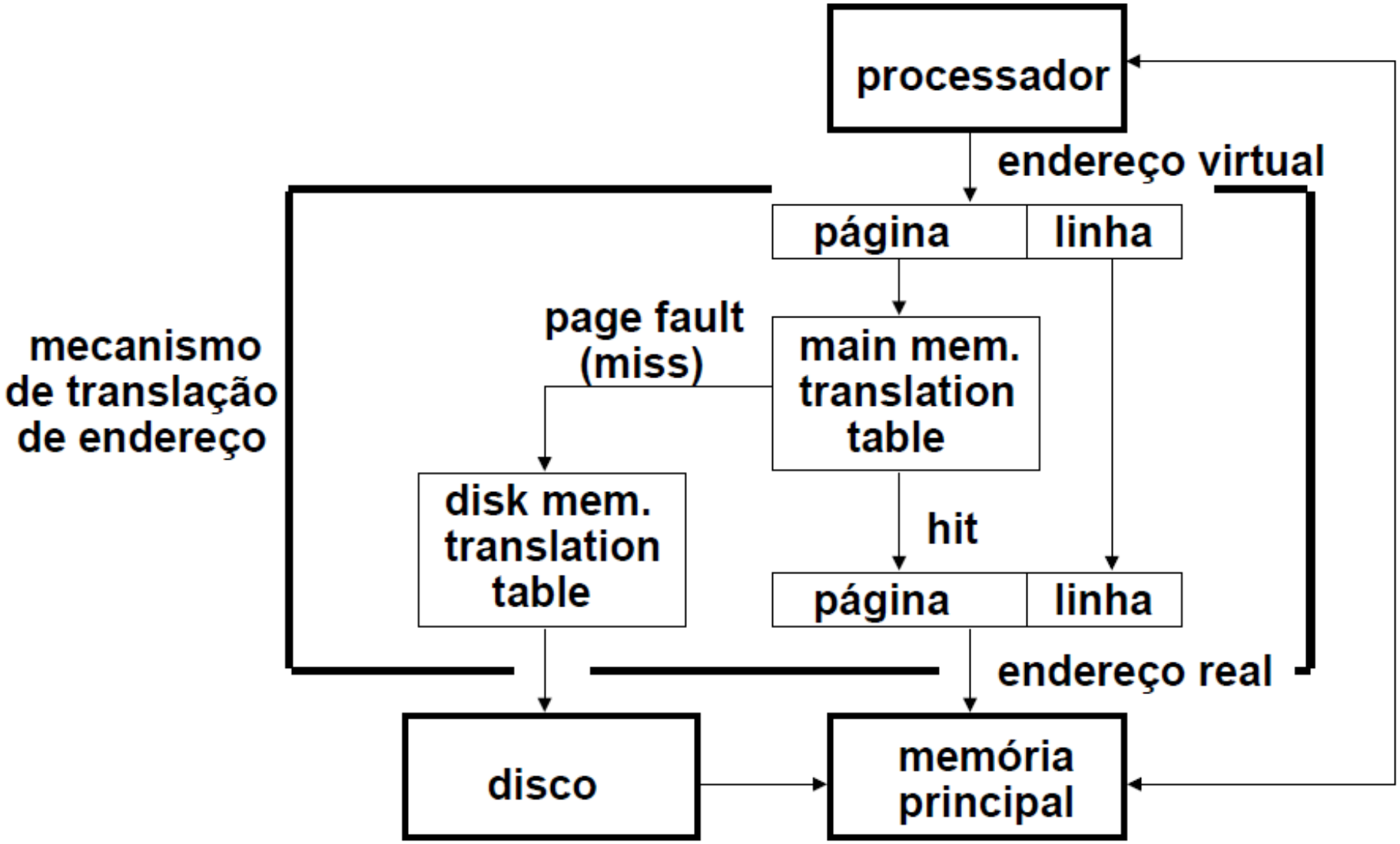
- **OPT (solução ótima)**
  - Substitui a página que não será usada por um tempo mais longo (prever futuro)
- **Memória com 3 frames (quadros)**
  - Frame tem tamanho igual a uma página virtual



- Total de requisições = 20
- Total de falhas = 9
- Taxa de falha = 9/20 = 0,45

https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/9_VirtualMemory.html

Translação de Endereço



Translação de Endereço

PUC Minas

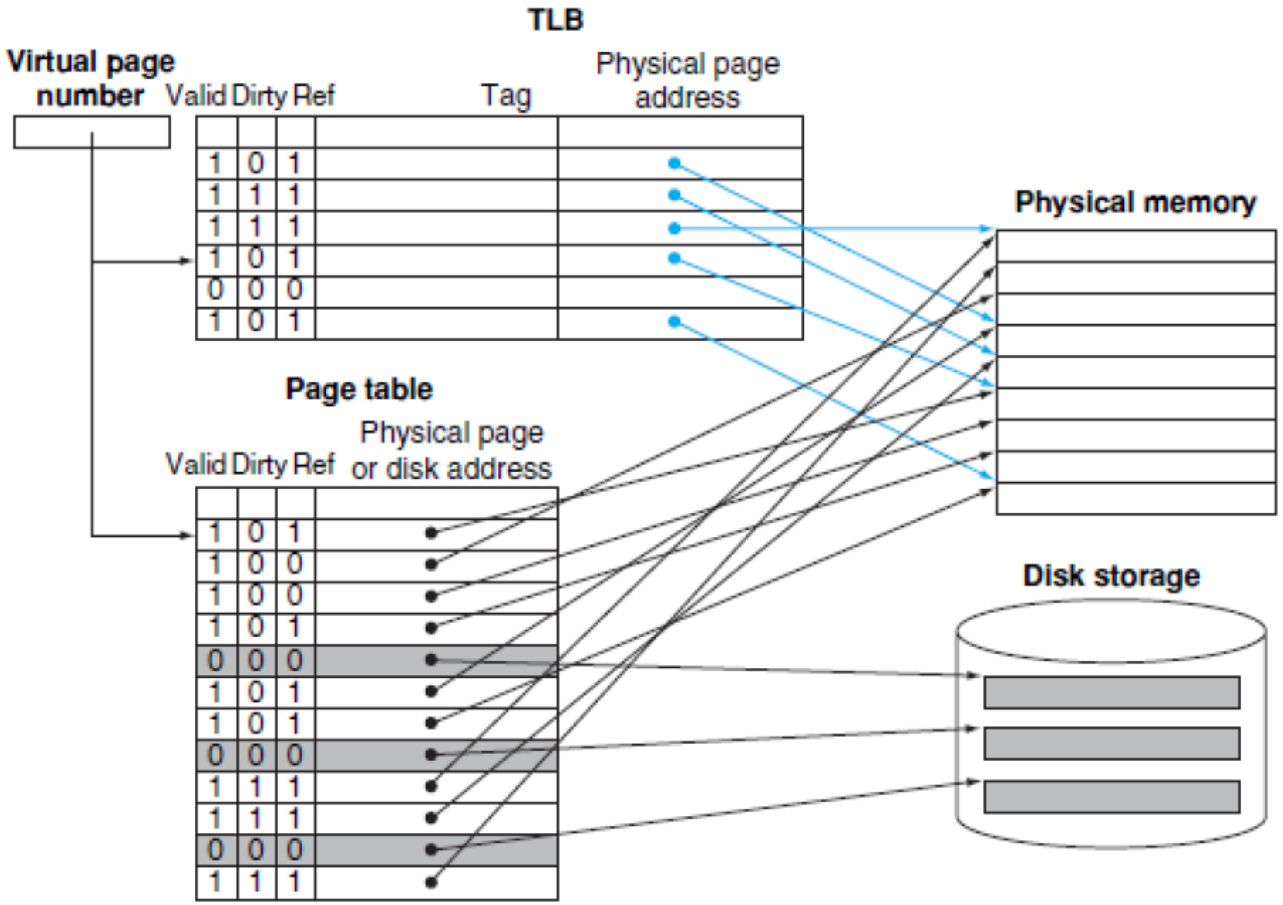
- nº de páginas na memória secundária é muito grande
  - espaço virtual de 2^{32}bytes, páginas de 4k bytes, 4 bytes por entrada na tabela
  - 4 MBytes apenas para a tabela de páginas!!!
  - tamanho excessivo da main memory translation table
- se tabela ficar na memória principal => dois acessos à memória a cada cache miss

- working set = conjunto de páginas mais prováveis de serem acessadas num dado momento, devido ao princípio de localidade

- Translation Look-Aside Buffer (TLB)
  - implementado em hardware
  - traduz endereços virtuais para endereços reais
  - só inclui páginas do working set
  - pode ser considerado como uma "cache" da MMTT

- Main Memory Translation Table (MMTT)
  - implementada em **software**

Translação de Endereço



PATTERSON, David A. Organização e projeto de computadores a interface hardware/software

PUC Minas

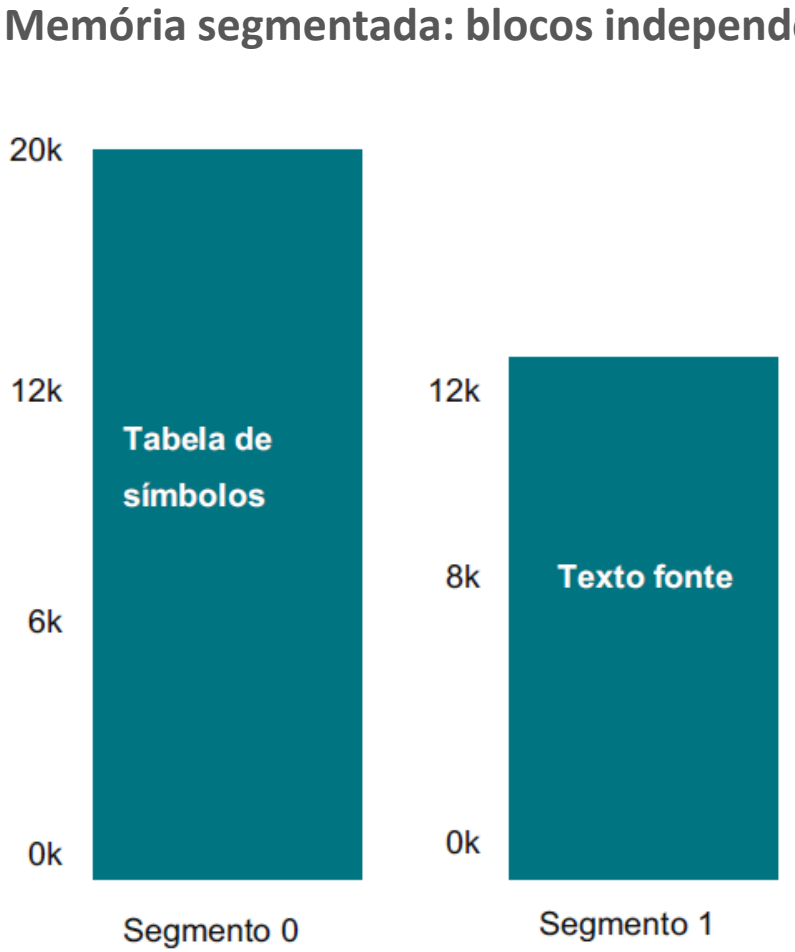
Segmentação

Segmentação

PUC Minas

- Nessa técnica de gerência de memória, os programas são **divididos em sub-rotinas e estrutura de dados, sendo colocados em blocos de informações na memória.**

- Esses blocos ou segmentos possuem tamanhos variados. Esse tamanho pode ser de 1 ao máximo permitido. Segmentos diferentes podem ter tamanhos diferentes (20k, 12k). Esses segmentos crescem e diminuem independentemente.



Segmentação

PUC Minas

A segmentação **facilita o compartilhamento de procedimentos ou dados entre vários processos**. Uma biblioteca compartilhada, por exemplo, pode ser acessada por vários processos. Esta é armazenada num segmento e poderá ser compartilhada entre os diversos processos (MACHADO; MAIA, 2013).

O **processo é muito semelhante ao de paginação**, mas nessa técnica uma tabela de mapeamento de segmentos é utilizada para **mapear as áreas livres.** Os endereços são compostos pelo número de segmento e um deslocamento dentro do segmento.

Segmentação

PUC Minas

O número de segmento identifica unicamente a **entrada na tabela de segmentos (ETS)**, onde estão as informações sobre o segmento de memória real. O endereço absoluto é calculado **a partir do endereço inicial do segmento mais o deslocamento dentro do segmento.**

O sistema operacional trabalha mantendo uma tabela de áreas livres e ocupadas na memória.

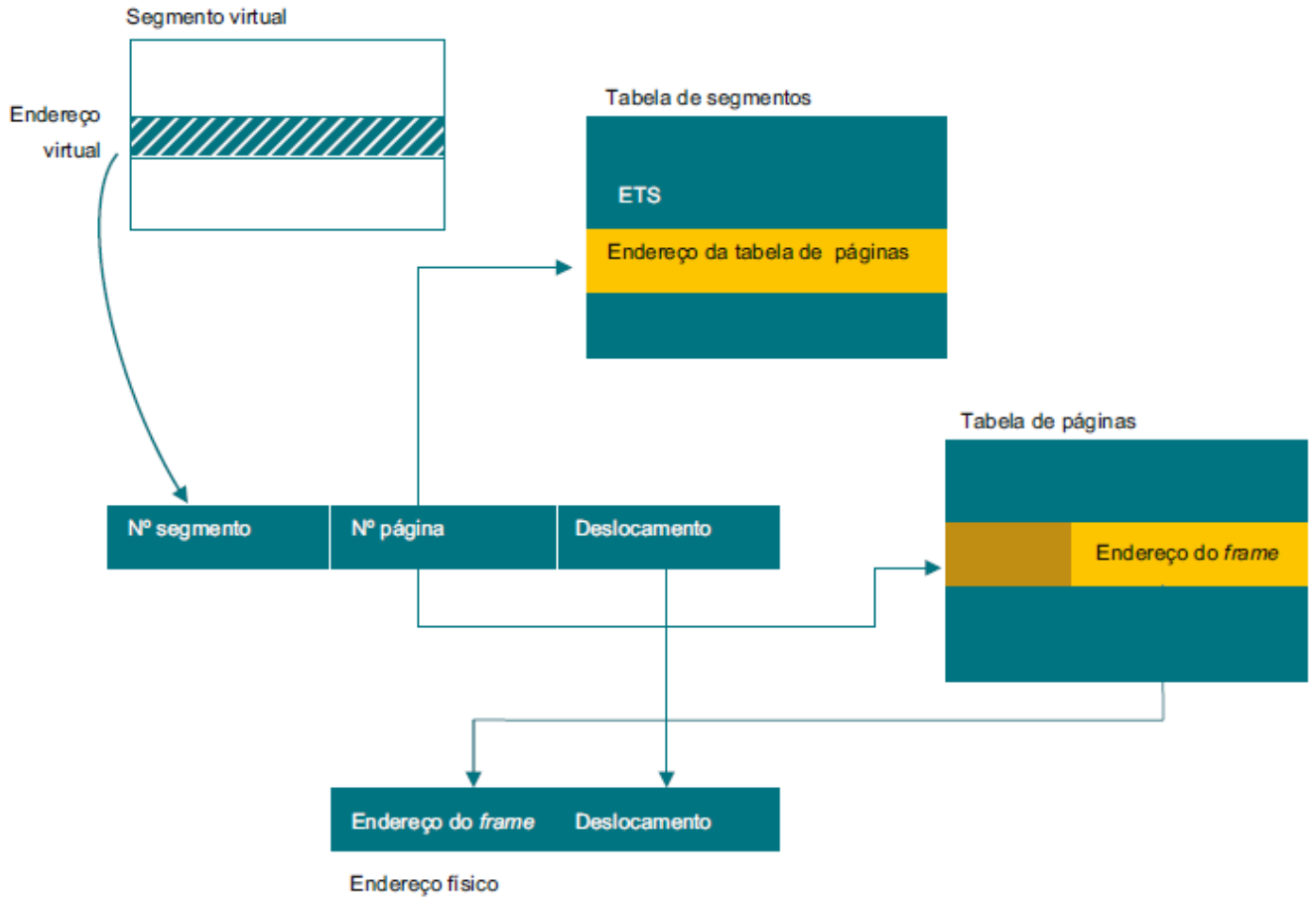
Ao iniciar um novo processo carregado para memória, o sistema localiza um espaço livre para sua alocação. ***Best-fit*, *worst-fit* ou *first-fit***, são estratégias de alocação dinâmicas que podem ser usadas para determinar as áreas livres mais adequadas.

Segmentação por Paginação

PUC Minas

Combinação das duas técnicas, **paginação e segmentação.**

- Os programas são divididos em segmentos e esses segmentos são divididos em páginas (Figura 79). **Nesse sistema, o endereço é formado pelo número do**
- **segmento, um número de página dentro desse segmento e um deslocamento dessa página**.
- Por meio do número de segmento encontra o número da página de segmentos. Com o número da página, obtém uma entrada na tabela de páginas do segmento, e por fim, com esse número entrada na tabela de páginas com informações de endereço. Adiciona a posição inicial e o deslocamento ao endereço físico (STUART, 2011).



Proteção

PUC Minas

Deve haver segurança em sistemas multitarefas, onde **há vários processos compartilhando a memória concorrentemente.** No sistema de memória virtual, **cada processo possui sua própria tabela de mapeamento e a tradução de endereços é realizada pelo sistema**.

A proteção é necessária mesmo assim para impedir que um determinado processo acesse uma página/segmento do sistema. Essa proteção é implementada por meio de dois *bits* (***Read, Write***) de acesso que determinam a segurança. A combinação binária determina a proteção.

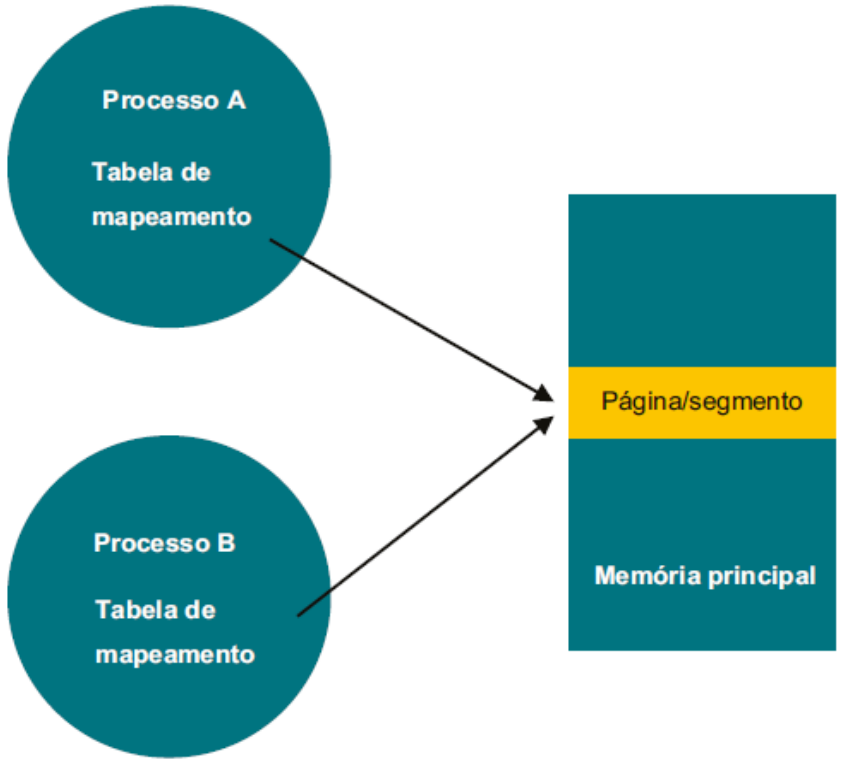
R W	Explicação
0 0	Sem acesso
1 0	Acesso de leitura
0 1	Acesso de gravação
0 1	Acesso leitura/gravação

Compartilhamento de Memória

PUC Minas

Aplicações como editores de textos, aplicações de usuários (agendas, *waze*, jogos) em sistemas **multiprogramáveis** fazem uso constante de memória, e por motivos de eficiência seria desnecessário ocupar a memória por cópias desses programas simultaneamente.

O sistema de memória virtual faz **uso de compartilhamento de memória utilizando a mesma página/segmentação**. Ele somente faz com que as tabelas de entradas de ambos apontem para as mesmas páginas/segmentos na memória principal, ou seja, fazem uso da mesma posição na memória (SILBERSCHATZ, 2015).



A diferença entre paginação e segmentação está no tamanho. Enquanto segmentação é tamanho variado, a paginação é de tamanho fixo. No compartilhamento, a grande vantagem é a forma que os programas são divididos. Devido a estrutura lógica, sub-rotinas e estrutura de dados, o compartilhamento é mais simples do que o de páginas. Enquanto a tabela de vetor necessita de várias entradas, a tabela de segmentos necessita somente de uma entrada

Diferença entre Paginação e Segmentação

PUC Minas

A diferença entre paginação e segmentação está no tamanho. Enquanto segmentação é tamanho variado, a paginação é de tamanho fixo.

No compartilhamento, a grande vantagem é a forma que os programas são divididos. Devido a estrutura lógica, sub-rotinas e estrutura de dados, o compartilhamento é mais simples do que o de páginas. Enquanto a tabela de vetor necessita de várias entradas, a tabela de segmentos necessita somente de uma entrada

Referências

PUC Minas

- BHATTACHARJEE, A. Preserving Virtual Memory by Mitigating the Address Translation Wall. IEEE Micro, v. 37, n. 5, 2017. 6-10 p. DOI: 10.1109/MM.2017.3711640
- CRUZ , E M et al. Communication-aware thread mapping using the translation lookaside buffer. Concurrency and Computation: Practice and Experience, v. 27, n. 17, 2015. 4970-4992 p. DOI: 10.1002/cpe.3487.
- MACHADO, Francis Berenger; MAIA, Luiz Paulo. Arquitetura de sistemas operacionais. 5a ed. Rio de Janeiro: LTC, 2013. 263 p. ISBN:9788521622109.
- SILBERSCHATZ, Abraham. Fundamentos de Sistemas Operacionais. 9ª ed. LTC, 2015. 524 p. ISBN: 9788521629399. ISBN digital 9788521630012.
- STUART, Brian L.. Princípios de sistema operacionais: projetos e aplicações. São Paulo: Cenage Learning, 2011. 637 p. ISBN: 9788522107339.
- TANENBAUM, Andrew S. Sistemas Operacionais Modernos. 4a ed. São Paulo: Pearson, 2016. 864 p. ISBN: 9788543005676.