# Memória Virtual

## Ciclo 4 – AT2

Prof. Hermes Senger

# Nota

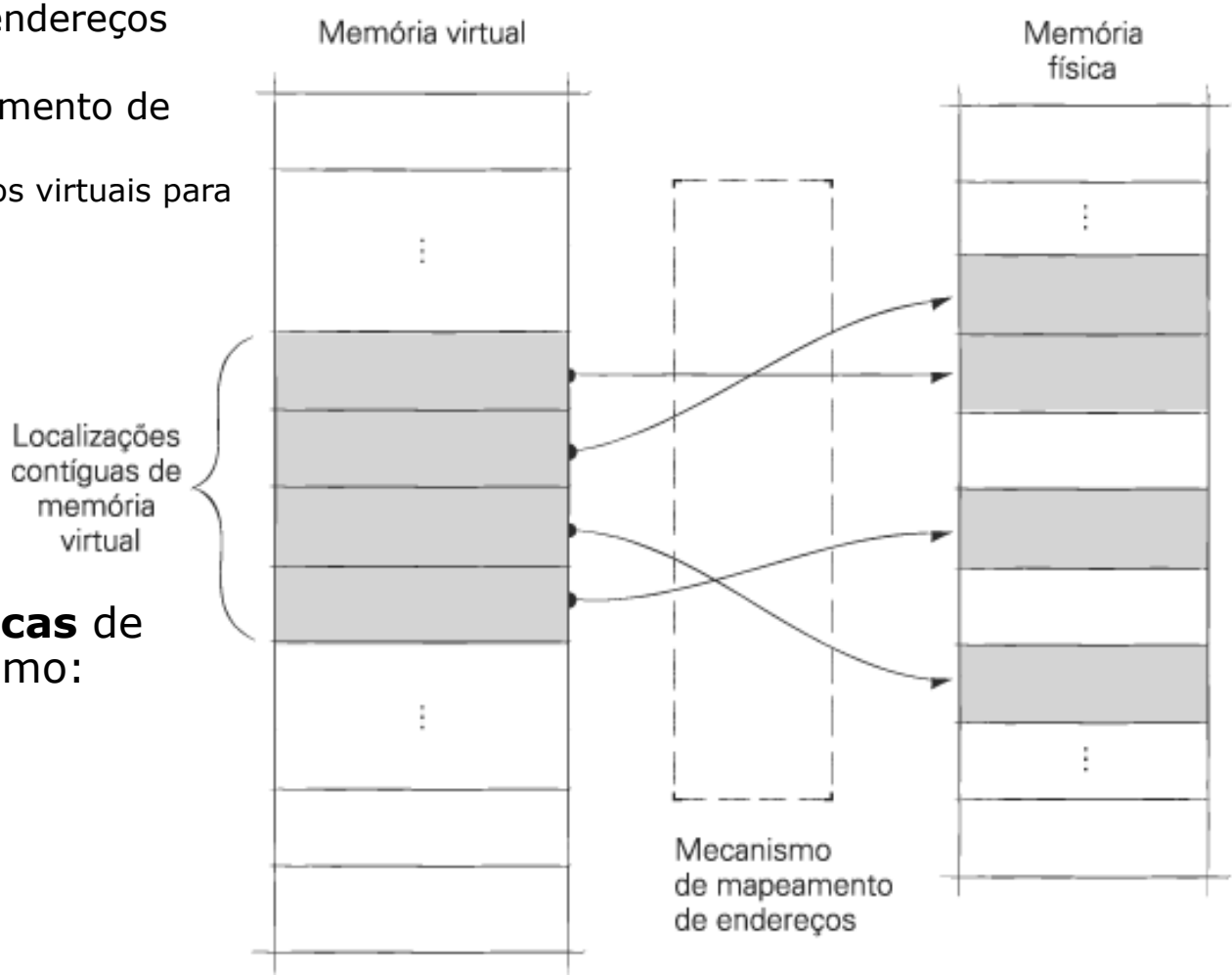
O presente material foi elaborado com base no material didático do livro *Sistemas Operacionais, 3ª edição, de H.M.Deitel, P.J. Deitel, D.R. Choffnes, Editora Pearson Prentice Hall, São Paulo, 2005*, disponibilizado pela editora.

# Objetivos

- **Esta videoaula apresenta:**
  - Breve revisão sobre
    - o conceito de memória virtual
  - O conceito de paginação sob demanda
  - Os desafios da substituição de páginas.
  - Diversas estratégias populares de substituição de páginas e como se comparam à substituição ideal de páginas.
  - O impacto do tamanho da página sobre o desempenho da memória virtual.
  - O comportamento do programa sob paginação.

# Revisão - Memória Virtual

- Soluciona o problema de **pouco espaço** de memória.
- Cria a **ilusão** de que existe mais memória do que a disponível no sistema.
- **Contiguidade artificial**
  - Baseada no uso de endereços virtuais
  - Unidade de gerenciamento de memória (MMU)
    - Traduz os endereços virtuais para endereços físicos.



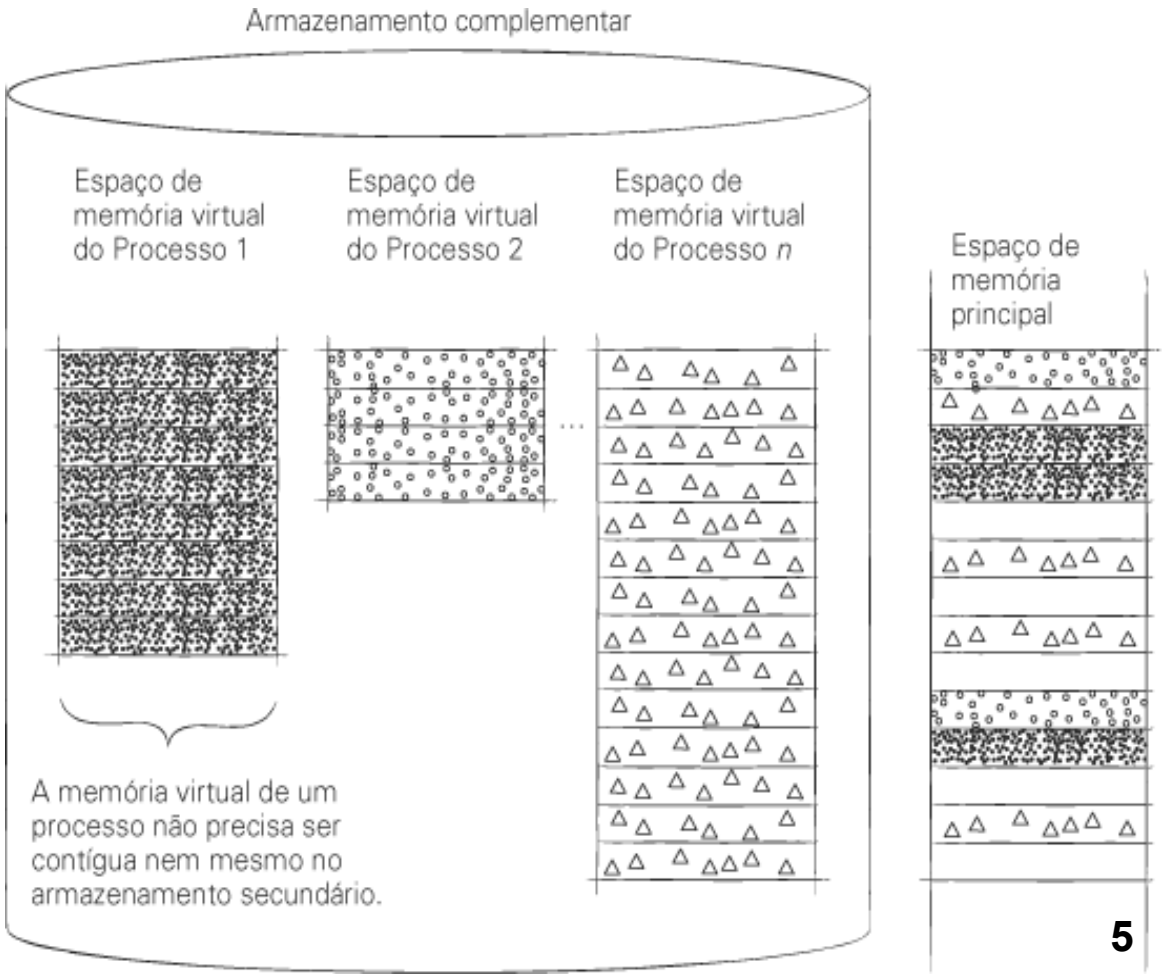

- Geralmente usa **técnicas** de gerenciamento tais como:
  - **Paginação**
  - **Segmentação**
  - ou combinações dessas …

# Revisão: Memória virtual

- **Espaço de endereçamento virtual** de um processo
  - Reside fisicamente no disco - pode estar em espaço não-contíguo
  - Uma parte (subconjunto) reside na memória principal – não-contígua
- **Contiguidade** apenas artificial

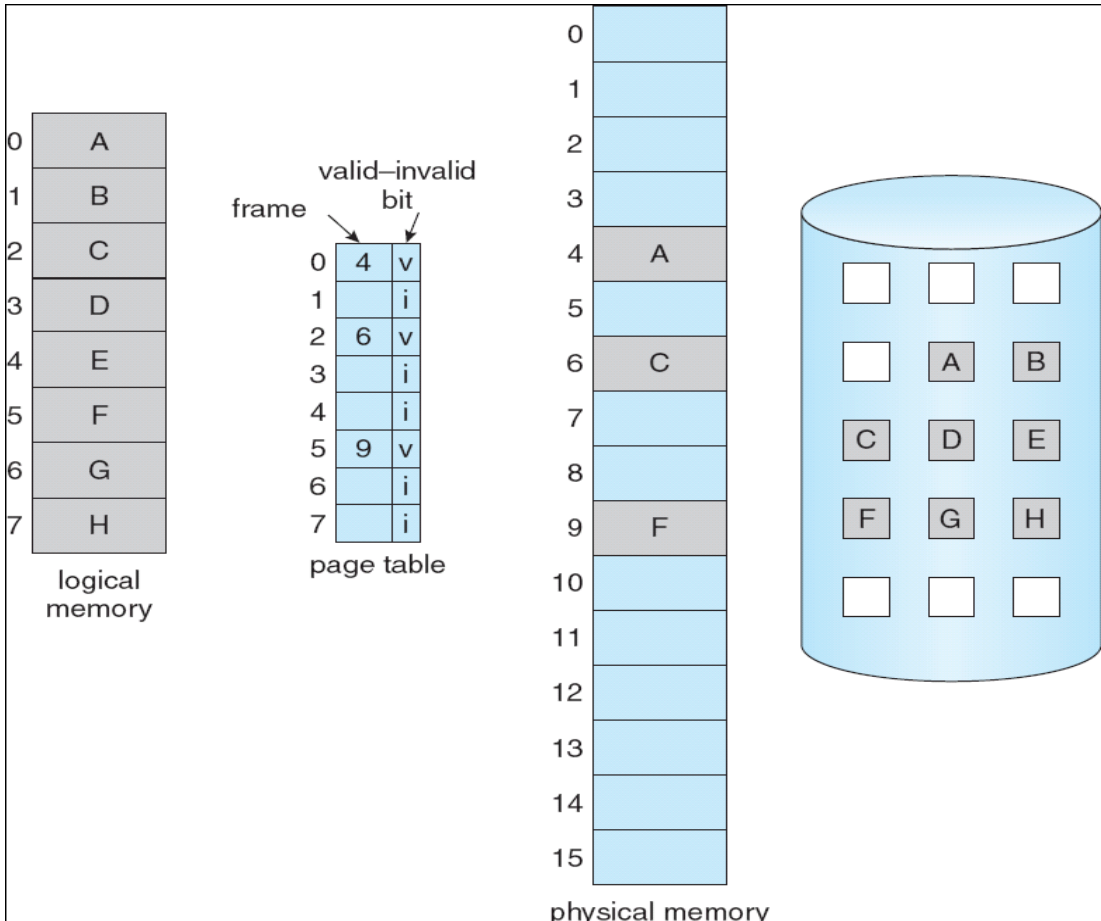
Obs.: **Página**

- Pode conter **dados e instruções**
- Só precisa ser carregada na memória física no momento em que vai ser acessada (leitura/escrita)
- Se estiver na memória
  ⇒ **acesso é rápido** (ex: nanossegundos)
- Mas, se estiver no disco
  ⇒ o **acesso** é muito mais **lento** (ex: milissegundos)



# Revisão: Paginação e Memória Virtual

- Normalmente, a memória virtual é maior do que a memória física
  - ⇒ Páginas possuem uma cópia armazenada em disco
  - ⇒ **Nem todas as páginas ficam na memória física**
- O que acontece no caso de uma página ausente ser requisitada pelo processador?
  - Ex.: O que aconteceria se a pág. 1 (que contém a informação B) for requisitada?
  - E se todos os quadros da memória física estivessem ocupados e uma nova página for requisitada?



# Revisão: Tradução de endereços

Ex.: memória física:

- 32 bytes = 8 páginas de 4 bytes

Exemplos:

0 → (5x4+0) = 20

(end.lógico) (end físico)

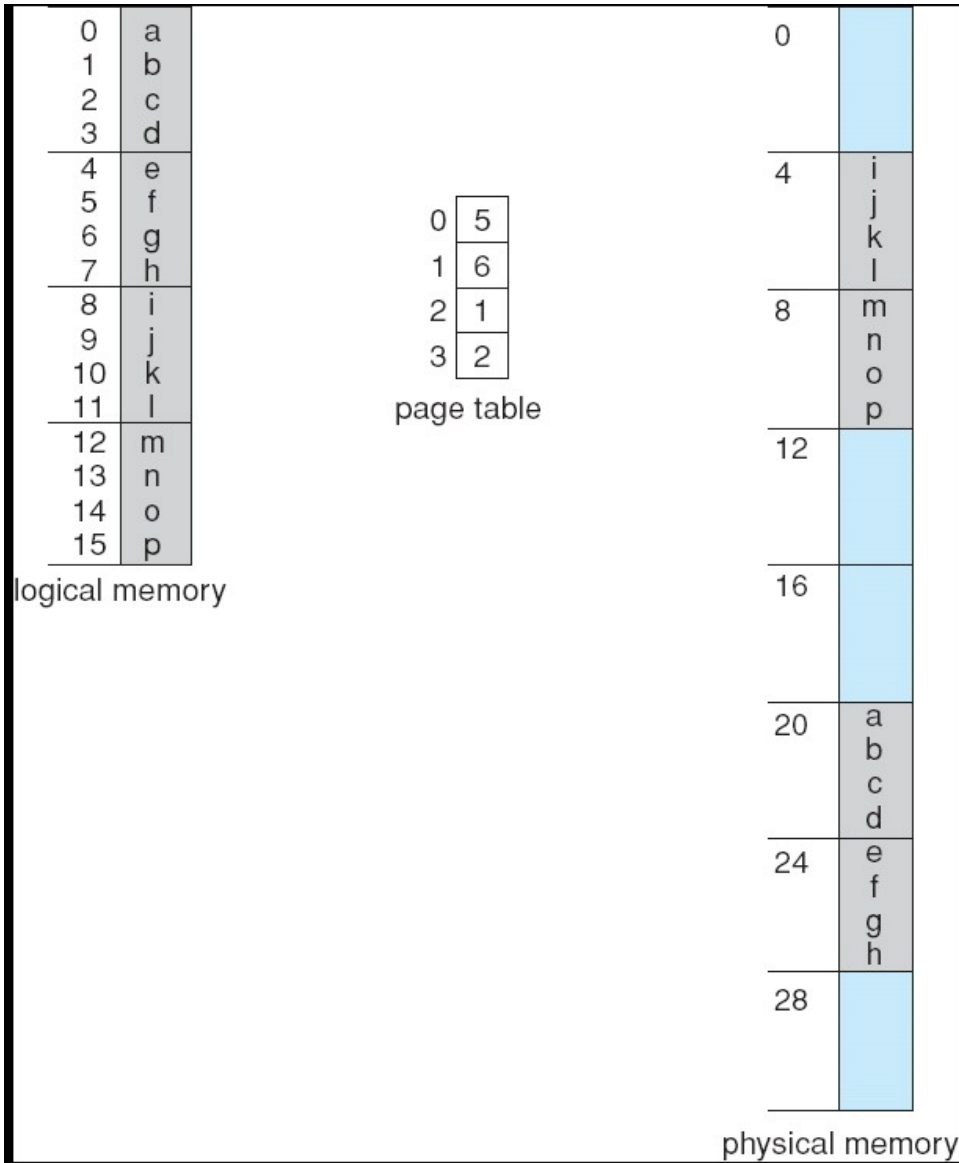
| 000 | 00 | → | 101 | 00 |
|---|---|---|---|---|

Calcule:

End lógico 3 =

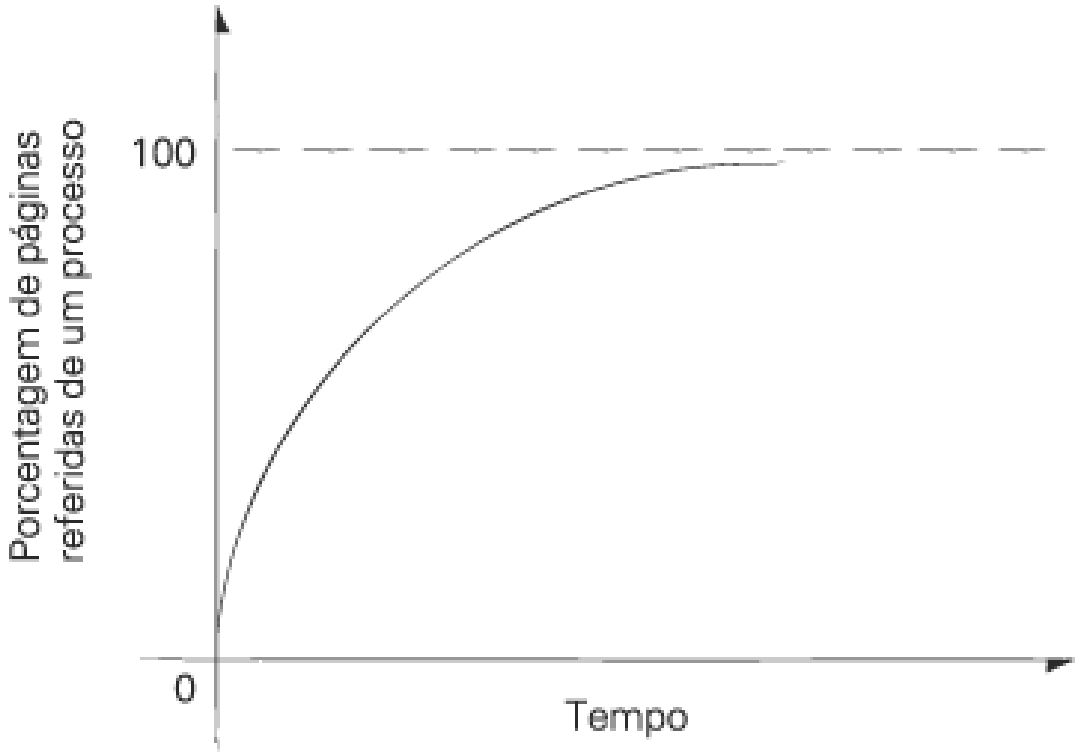
End lógico 4 =

End lógico 13 =



# Paginação por demanda

- **Paginação por demanda**
  - Quando um processo é executado pela primeira vez, o sistema **transfere** para a memória principal a página que contém sua **primeira instrução**.
  - Depois disso, cada vez que o processo referenciar uma **nova página** que está armazenada no disco, o SO deverá **transferí-la** para a memória principal.



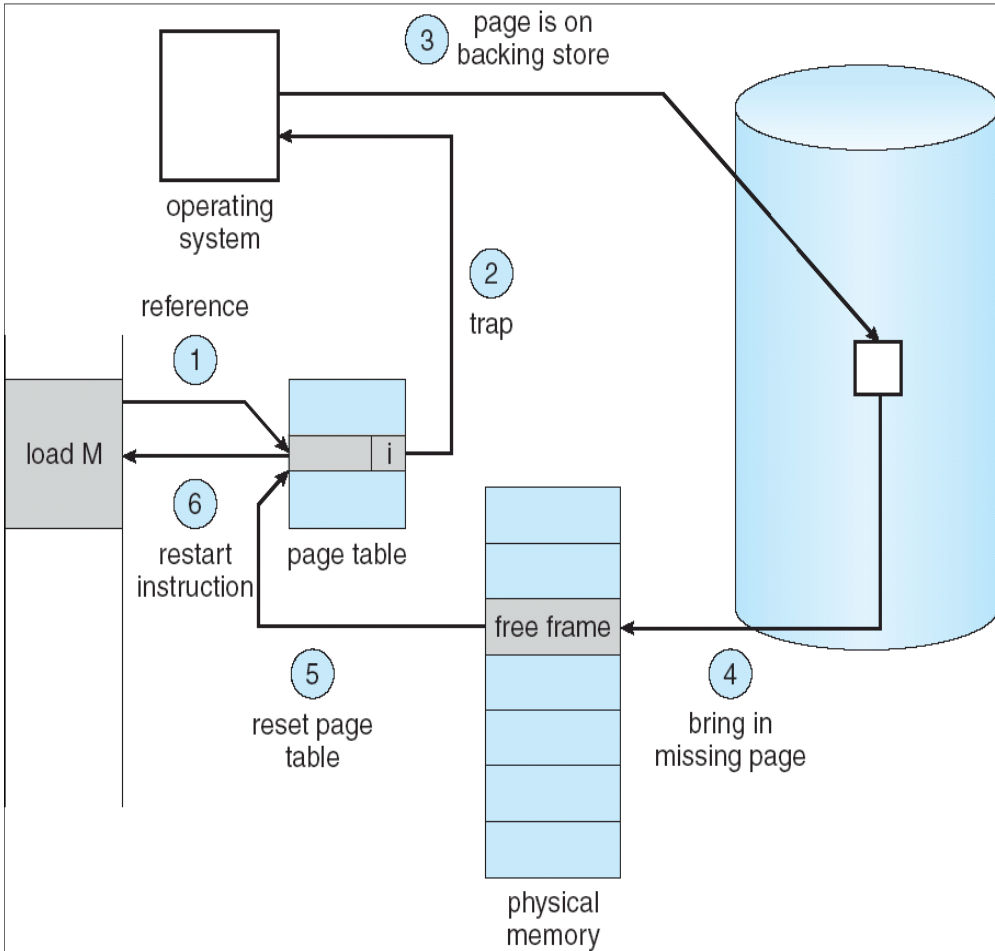
# Falta de Página

- Quando uma página é **referenciada pela primeira vez**, ela precisa ser carregada

  Dizemos que ocorreu uma falta de página (page fault)

**Ex: MOV AX, [1000]**

1. É gerada uma interrupção (instr. MOV é **paralisada**)
2. O SO então verifica a referência:
   1. Se for **inválida,** aborta o processo
3. Procura um **quadro** (*frame*) livre
4. **Carrega** (*swap in*) a página no quadro
5. **Atualiza** a tabela de páginas
6. Marca **bit** de página válida (presente)
7. **Recomeça** a instrução que causou o *page fault*



# Substituição de Páginas

- **E se não houver quadros livres para carregar uma página na memória ?**
  - É preciso liberar quadros
  - O SO então **escolhe** um quadro a ser substituído
- **Estratégia de substituição**
  - Visa  selecionar páginas para substituição quando a memória está cheia.
  - Determina **onde** se deve colocar na memória principal um segmento que está entrando.
- **Estratégia de busca**
  - Determina **quando** as páginas ou os segmentos devem ser carregados na memória principal.
  - Estratégia de busca **antecipada**
    - Usa heurísticas para **prever** a quais páginas um processo vai se referenciar em breve e carrega essas páginas ou segmentos.

# Desempenho da Paginação Sob Demanda

- Memória: tempo de acesso ~ nanossegundos ($10^{-9}$ s)
- Disco: tempo de acesso ~ milissegundos ($10^{-3}$ s)
- **Taxa de falta de páginas**: $0 <= p <= 1.0$
  - se $p = 0$ não há *page faults*
  - se $p = 1$, então 100% das referências são *page faults*
- Tempo de acesso efetivo (TAE)

TAE = (1 – *p*) x tempo de acesso à memória
+ *p* (tempo de *page fault*
+ tempo de *swap out*
+ tempo de *swap in*
+ tempo de *restart* da instrução)

# Exemplo

- Tempo de acesso à mem = 200 ns (nanossegundos)
- Tempo de *page-fault* = 8 (millissegundos)
- TEA = (1 – p) x 200 + p (8 ms)
  = (1 – p ) x 200 + p x 8.000.000
  = 200 + p x 7.999.800 ns

Se um a cada 1.000 acessos causa um *page fault*, então

TEA = 8.2 microssegundos.

O atraso será 41 vezes mais lento !!!

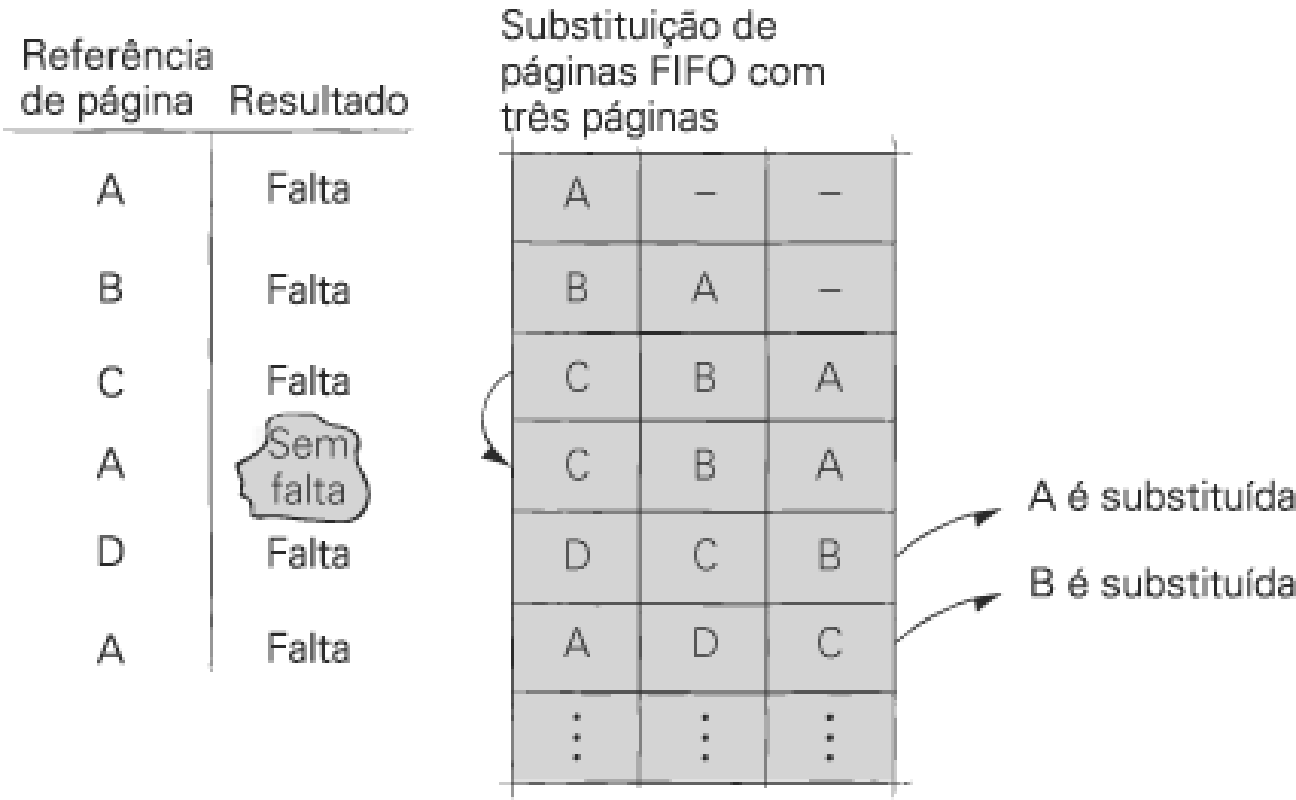
Então, como podemos manter a taxa de *page-faults* baixa?

# Localidade

- **A paginação funciona bem graças a um princípio observado empiricamente:**
  - **Princípio de localidade**
- Localidade temporal:
  - Se um item foi referenciado num dado instante, então, é bastante provável que ele seja referenciado novamente em breve
- Localidade espacial:
  - Se um item foi referenciado, então é bastante provável que os itens próximos a ele também sejam acessados em breve

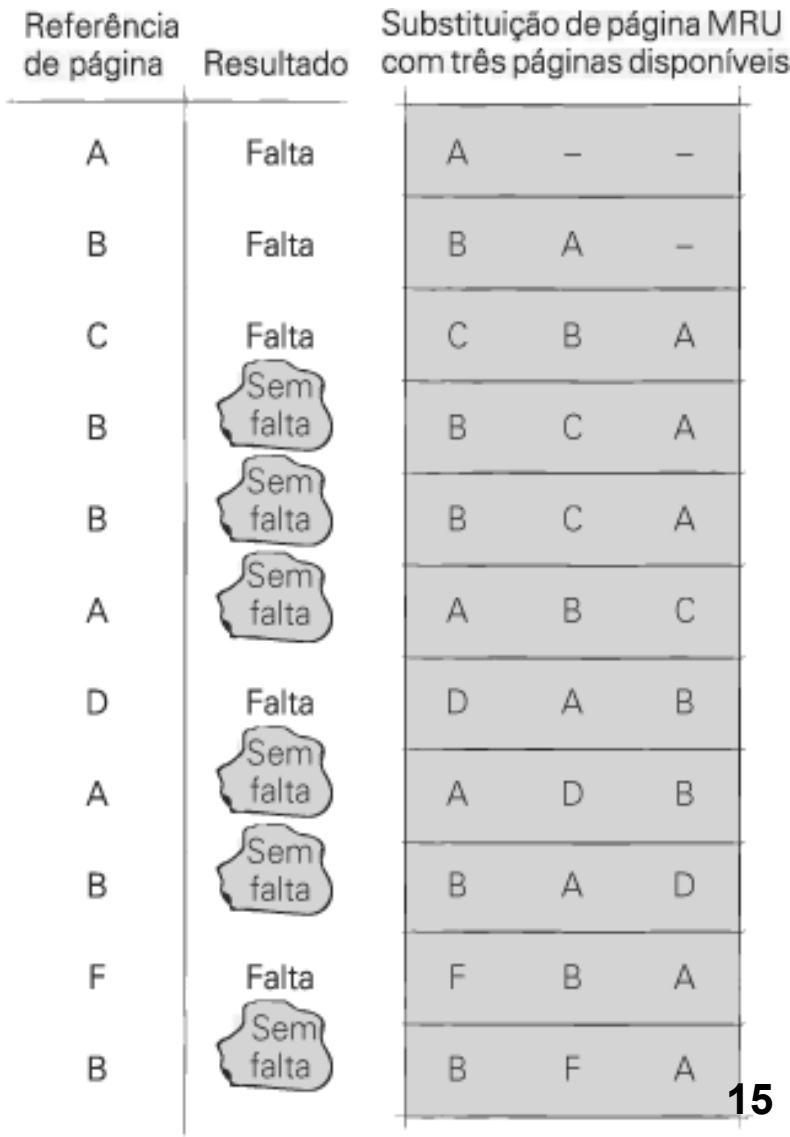
# Estratégia de substituição de páginas FIFO (primeira a entrar, primeira a sair)

- **Substituição de páginas FIFO**
  - Substitui a página que está há mais tempo no sistema.
  - Tende a substituir páginas que são intensamente usadas.
  - Pode ser implementada com uma sobrecarga relativamente baixa.
  - Não é prática para a maioria dos sistemas.



# Substituição de página menos recentemente usada (MRU)

- **Substituição de página MRU**
  - Melhor desempenho que a FIFO
  - Porém, maior sobrecarga
  - Explora a **localidade temporal,** substituindo a página que ficou mais tempo na memória sem ser referenciada.
  - O desempenho da substituição MRU pode ser inadequado se a página menos recentemente usada for a página seguinte a ser referenciada por um laço que referencia diversas páginas.



# *Substituição de página menos frequentemente usada (MFU)*

- **Substituição de página MFU**
  - Substitui a página menos intensamente referenciada.
  - Ideia básica:
    - **a página que não é referenciada com frequência provavelmente não será referenciada no futuro.**
  - Problema:
    - Pode escolher uma página errada para substituição.
    - Uma página referenciada intensamente no passado **talvez nunca mais** venha a ser referenciada, mas permanecerá na memória enquanto as páginas novas e ativas são substituídas.

# *Substituição de página não usada recentemente (NUR)*

- **Substituição de página NUR**
  - Parecida com a MRU
  - Pouca sobrecarga
    - usa um bit referenciado e um bit modificado para determinar que página não foi usada recentemente e pode ser substituída rapidamente.
  - Pode ser implementada em máquinas que não dispõem de bit referenciado e/ou bit modificado.

# *Modificações da FIFO: substituições de página 'segunda chance' e 'relógio'*

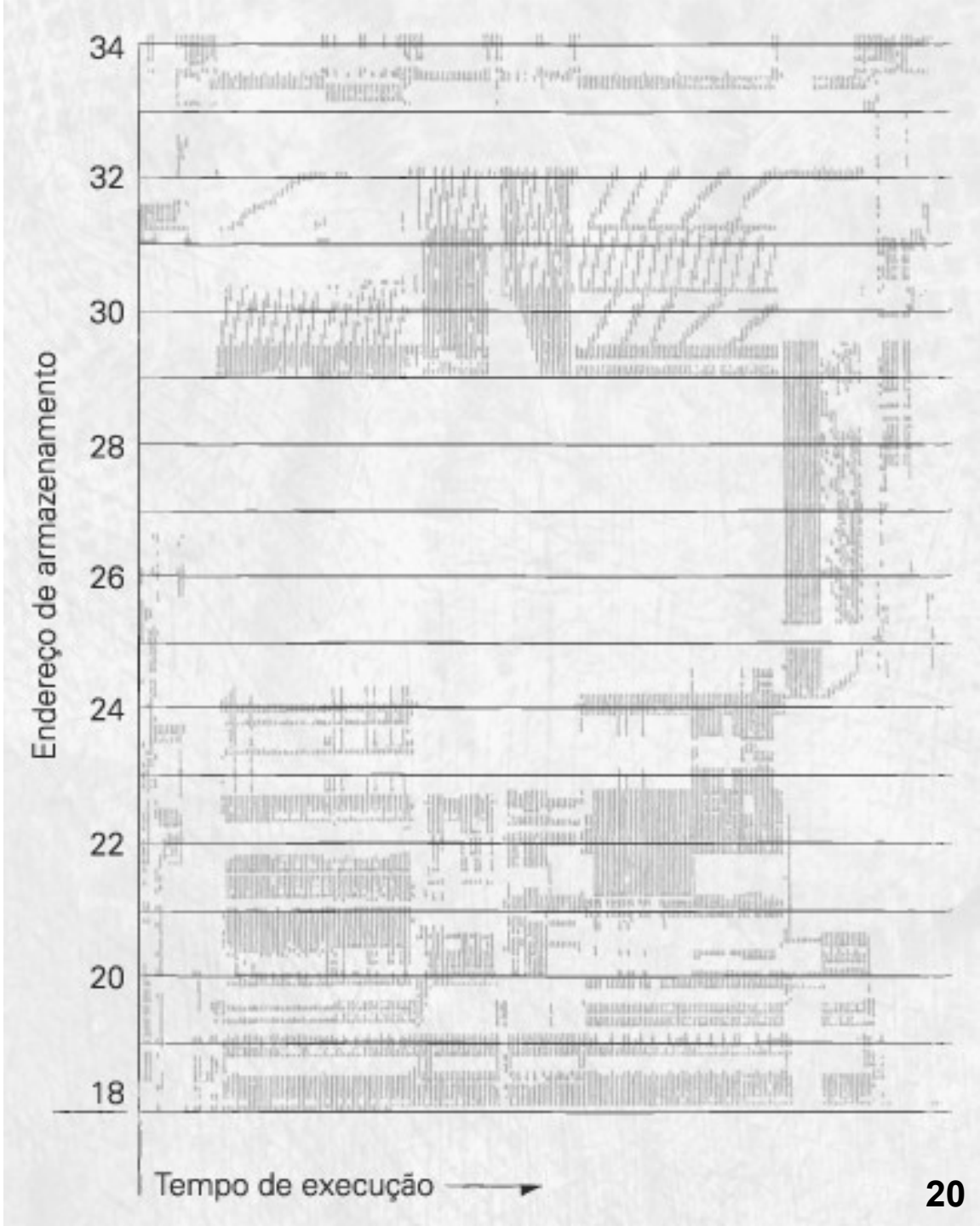
- **Substituição de página segunda chance**
  - Examina o bit referenciado da página mais antiga.
    - Se estiver desligado,
      → a estratégia seleciona essa página para substituição.
    - Se estiver ligado,
      → a estratégia desliga o bit e move a página para o final da fila FIFO.
  - Garante que as páginas ativas sejam as que menos têm probabilidade de ser substituídas.
- **Substituição de página relógio**
  - Semelhante à segunda chance, mas organiza as páginas em uma lista circular, e não em uma lista linear.

# Modelo de conjunto de trabalho

- **Para que um programa seja executado eficazmente:**
  - O sistema tem de manter na memória principal o subconjunto de páginas favorecido desse programa.
- **Do contrário:**
  - Pode haver uma atividade de paginação excessiva no sistema, diminuindo, assim, a utilização do processador. Essa paginação excessiva (*trashing*) ocorre porque o programa solicita páginas repetidamente ao armazenamento secundário.

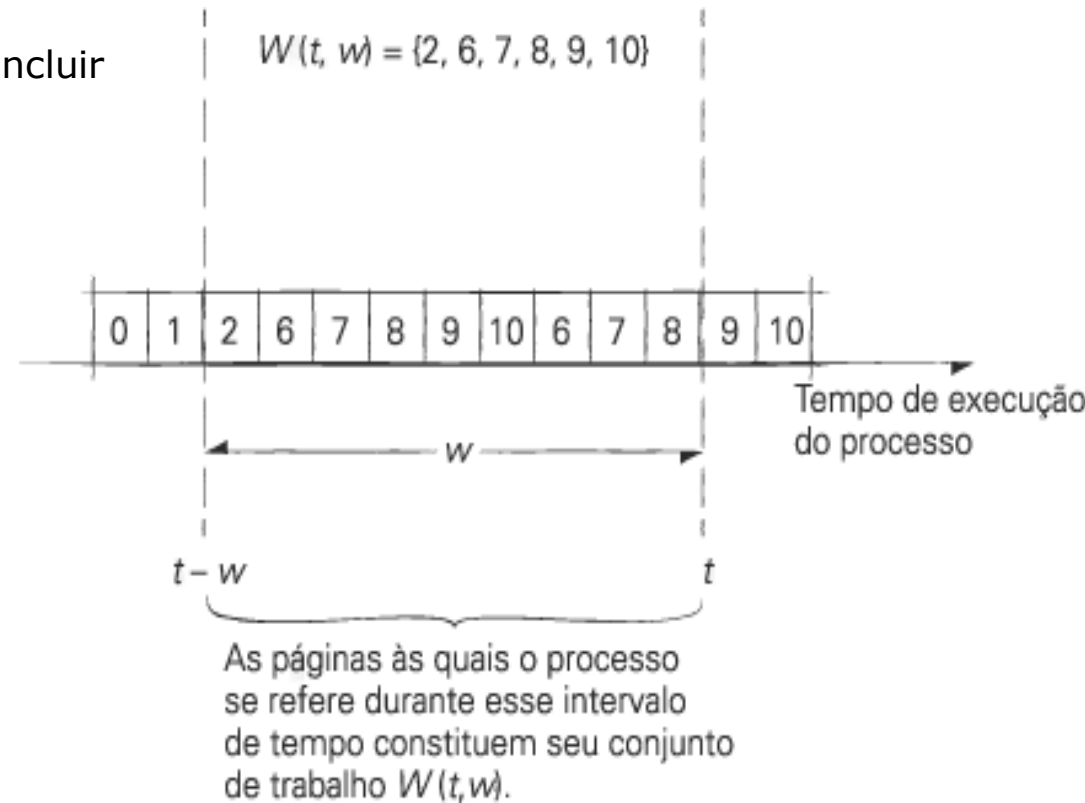
# Modelo de conjunto de trabalho

Padrão de referência de armazenamento exibindo localidade. (IBM Systems Journal, 1971)



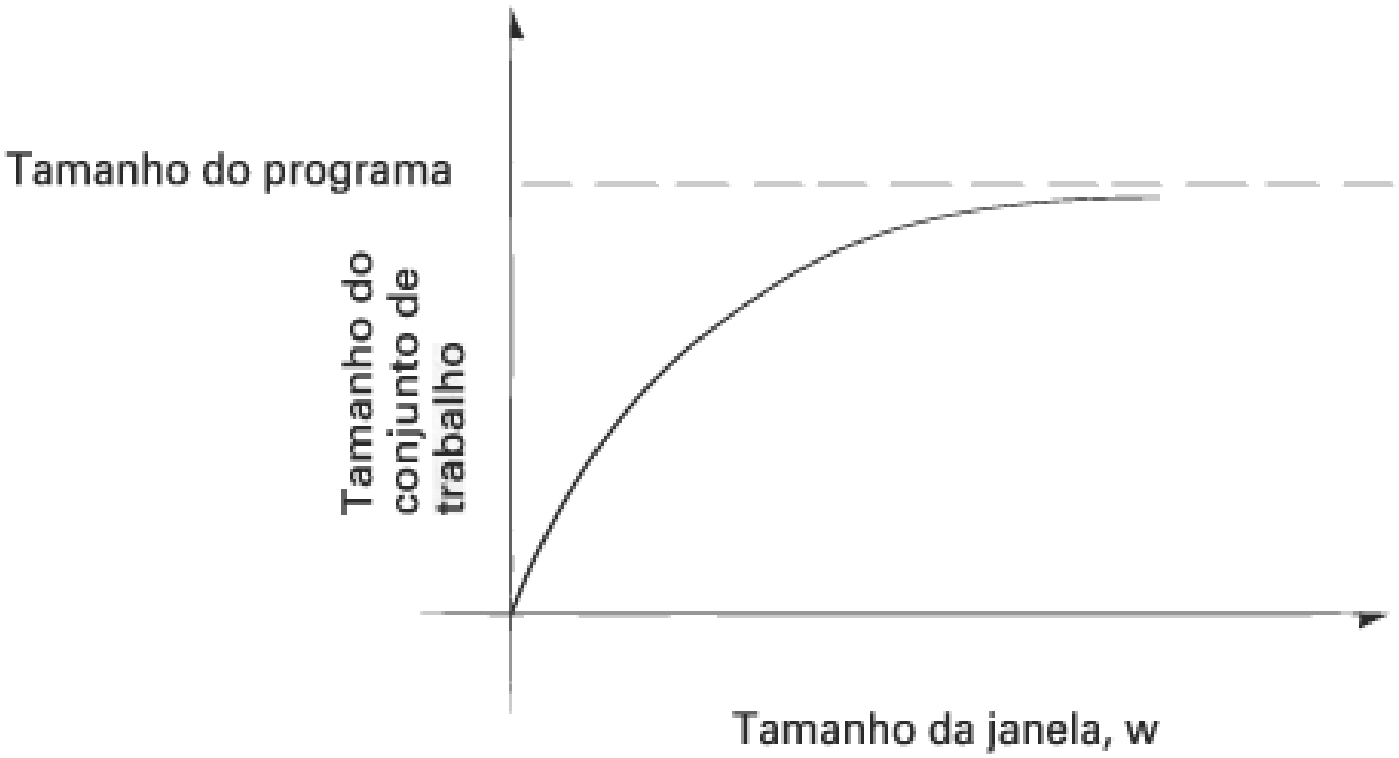
# Modelo de conjunto de trabalho

- Tenta identificar o **conjunto de páginas mais utilizadas** por um processo no passado recente
- Analisa os últimos acessos à memória feitos do **intervalo *t – w*** até ***t***
- No exemplo abaixo **W(*t, w*)** inclui as páginas: **2, 6, 7, 8, 9, 10**
- Qual o tamanho ideal para *w*?
  - Se **w** for muito pequeno: não vai incluir a localidade toda
  - Se **w** for muito grande: vai incluir várias localidades (ruim)



# Modelo de conjunto de trabalho

- O tamanho do conjunto de trabalho aumenta à medida que o tamanho da janela (w) aumenta.

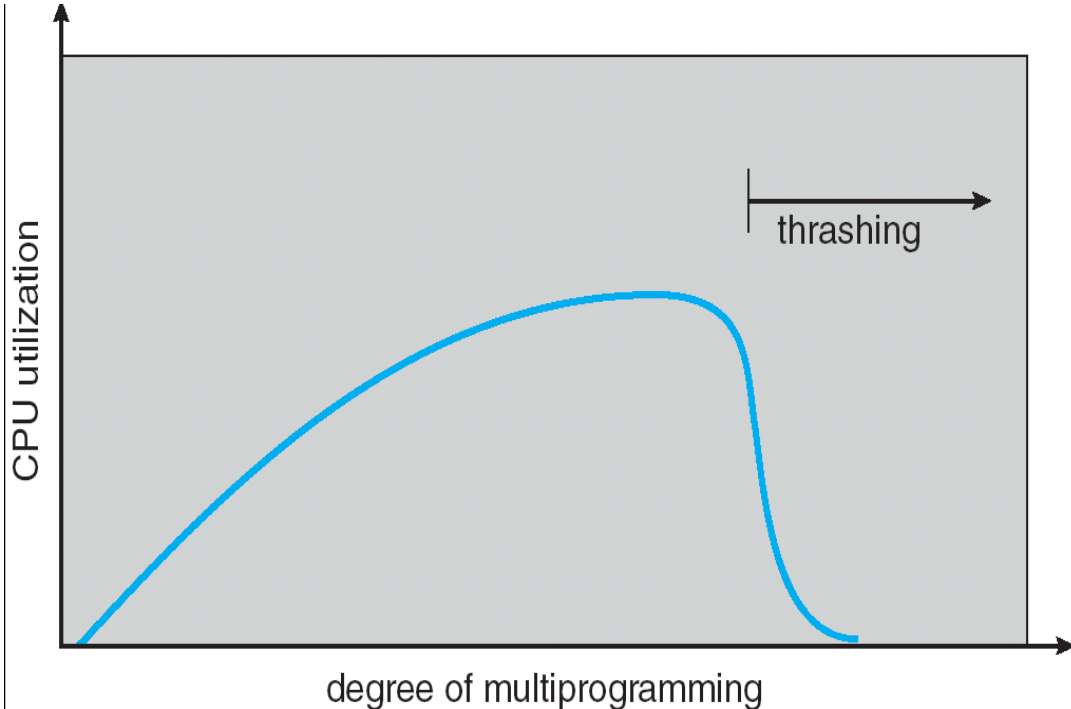


# Tamanho de página

| Fabricante | Modelo | Tamanho de página | Tamanho do endereço real |
|---|---|---|---|
| Honeywell | Multics | 1KB | 36 bits |
| IBM | 370/168 | 4KB | 32 bits |
| DEC | PDP-10 e PDP-20 | 512 bytes | 36 bits |
| DEC | VAX 8800 | 512 bytes | 32 bits |
| Intel | 80386 | 4KB | 32 bits |
| Intel/AMD | Pentium 4/Athlon | 4KB ou 4MB | 32 ou 36 bits |
| Sun | UltraSparc II | 8KB, 64KB, 512KB, 4MB | 44 bits |
| AMD | Opteron/Athlon 64 | 4KB, 2MB e 4MB | 32, 40 ou 52 bits |
| Intel-HP | Itanium, Itanium 2 | 4KB, 8KB, 16KB, 64KB, 256KB 1MB, 4MB, 16MB, 32MB, 64MB, 128MB, 256MB | entre 32 e 63 bits |
| IBM | PowerPC 970 | 4KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, 8MB, 16MB, 32MB, 64MB, 128MB, 256MB | 32 e 64 bits |

# Trashing

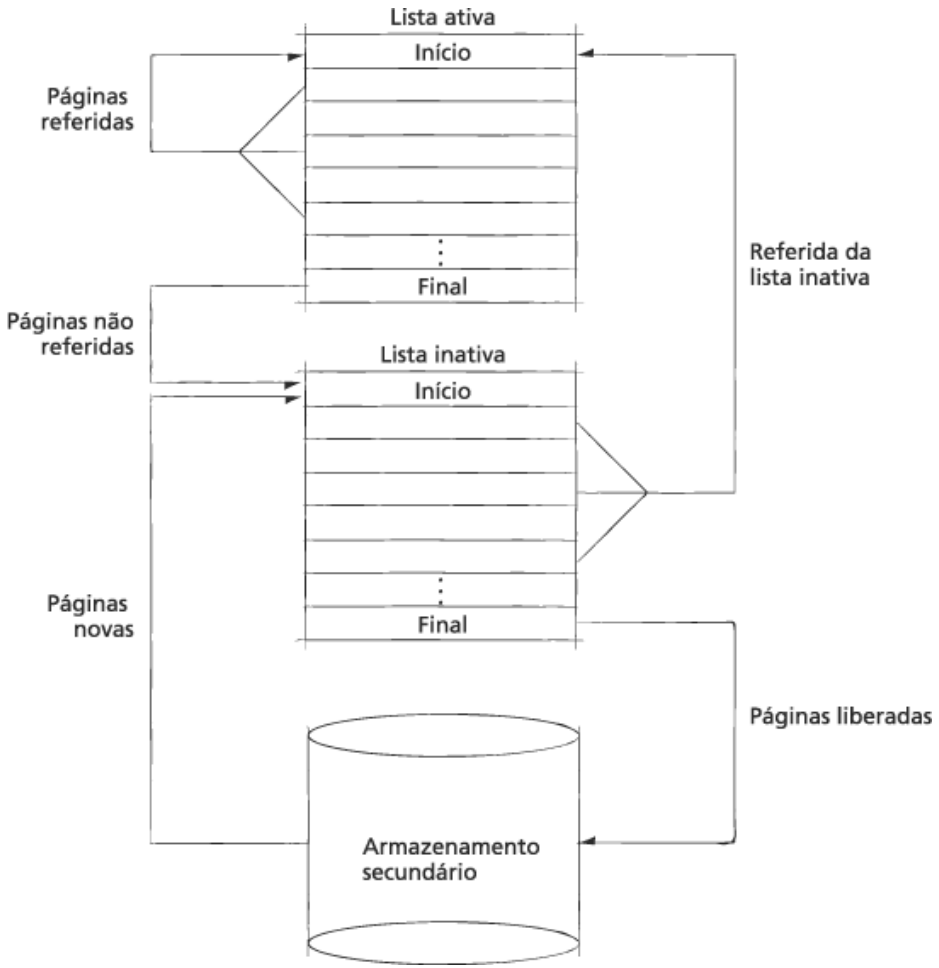
- CPU com baixa utilização
  - O sistema operacional decide aumentar a o grau de multiprogramação
    - Coloca mais processos para executar
- **Thrashing** ≡ processo passa a maior parte do tempo fazendo *swap* de páginas



# *Estudo de caso: substituição de páginas no Linux*

- **O Linux usa uma variação do algoritmo relógio para se aproximar da estratégia de substituição de páginas MRU.**
- **O gerenciador de memória usa duas listas vinculadas.**
  - **Lista ativa**
    - Contém as páginas ativas.
    - As páginas usadas mais recentemente ficam quase no topo da lista ativa.
  - **Lista inativa**
    - Contém as páginas inativas.
    - As páginas menos recentemente usadas ficam no final da lista inativa.
  - Somente as  páginas da lista inativa é que serão **substituídas**

# Estudo de caso: substituição de páginas no Linux



*Fim*