

Marisangila Alves, MSc
marisangila.alves@udesc.br
marisangila.com.br



JOINVILLE
CENTRO DE CIÊNCIAS
TECNOLOGICAS

UDESC
Universidade do Estado de Santa Catarina

2025/2

Sistemas Operacionais

Gerencia de Memória Virtual

Sumário

- 1 Memória Virtual
- 2 FIFO
- 3 ÓTIMO
- 4 LRU
- 5 RELÓGIO
- 6 NRU
- 7 ENVELHECIMENTO
- 8 WORKING SET
- 9 WSCLOCK
- 10 Belady
- 11 Thrashing
- 12 Bibliografia

Memória Virtual

Conceito

- › **Memória virtual** é uma técnica que permite que programas utilizem mais memória do que a fisicamente disponível.
- › O sistema operacional utiliza o **disco** como extensão da memória principal.
- › Cada processo possui um **espaço de endereçamento lógico contínuo**, independentemente da fragmentação física da RAM.
- › A memória virtual é implementada através de três mecanismos principais:
 - 1 **Overlay**;
 - 2 **Swap**;
 - 3 **Paginação sob demanda**.

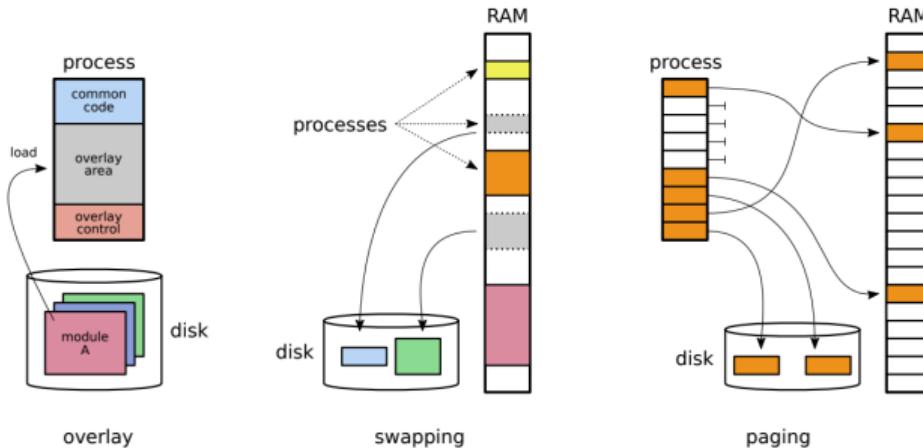


Figura 1: Abstração de memória virtual (Maziero, 2019).

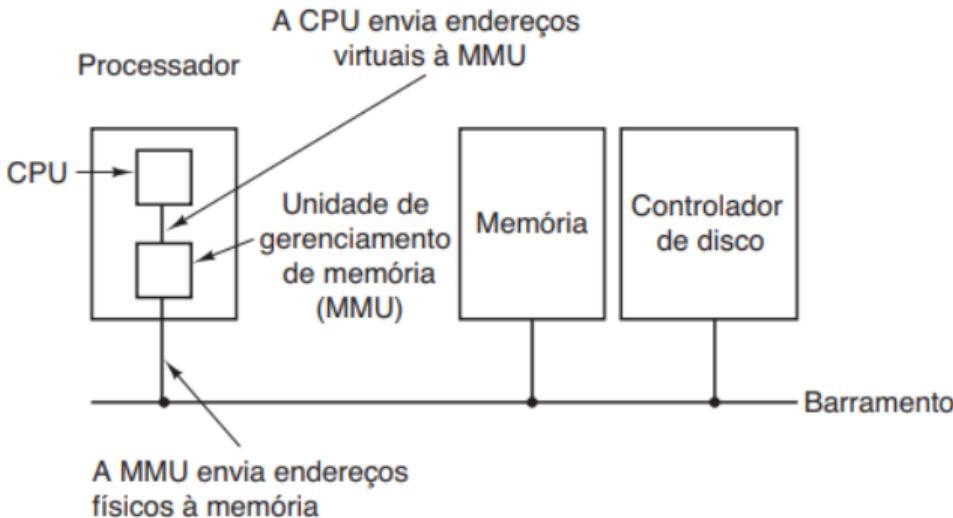


Figura 2: MMU gerencia a paginação (TANENBAUM, 2010).

Carregamento Parcial de Programas

- › Técnica antiga, usada quando a memória era muito limitada.
- › O programa é dividido em partes (módulos) que são carregadas apenas quando necessárias.
- › Quando uma parte não é mais usada, ela é substituída por outra no mesmo espaço de memória.
- › **Controle manual** feito pelo programador.

Troca entre memória e disco

- › Extensão do conceito de overlay, mas automática.
- › O sistema operacional copia **páginas** ou **processos inteiros** entre a RAM e o disco.
- › Ocorre quando há falta de espaço na memória principal.
- › O processo temporariamente removido é restaurado quando volta a ser necessário.

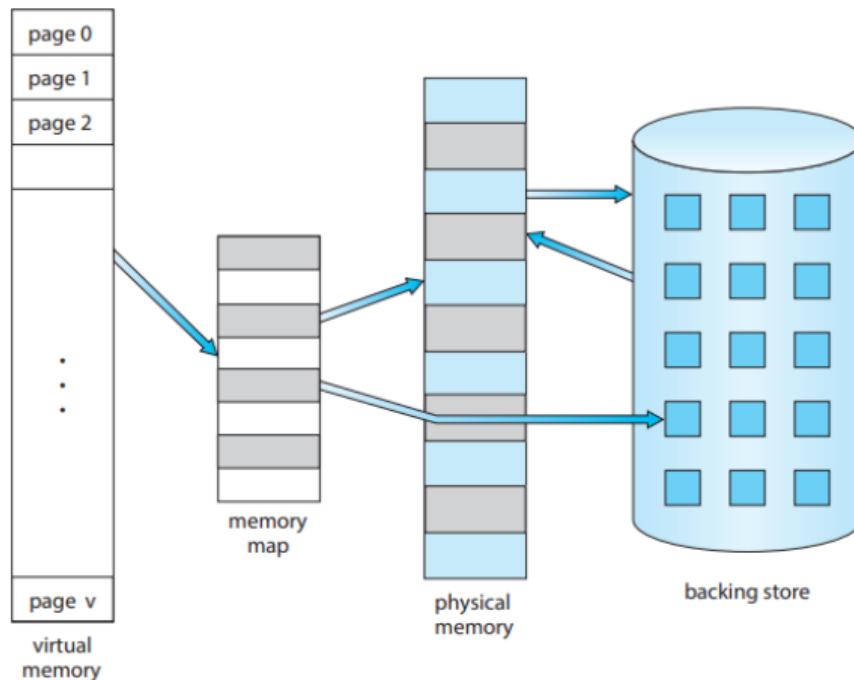


Figura 3: MMU gerencia a paginação (SILBERSCHATZ; GALVIN; GAGNE, 2001).

Carregamento de páginas quando necessário

- › As páginas são carregadas apenas quando acessadas pela primeira vez.
- › Se a página não está na memória, ocorre uma **falha de página**.
- › O sistema busca a página no disco e a coloca na RAM.
- › Se não há espaço disponível, é necessário substituir uma página existente.

Paginação Sob Demanda II

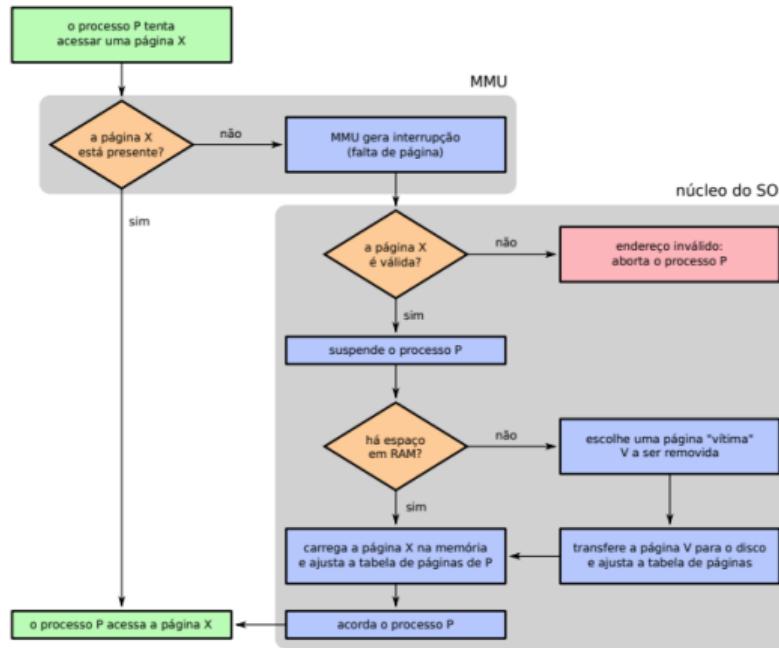


Figura 4: Fluxograma de Paginação em Disco (Maziero, 2019).

Paginação Sob Demanda III

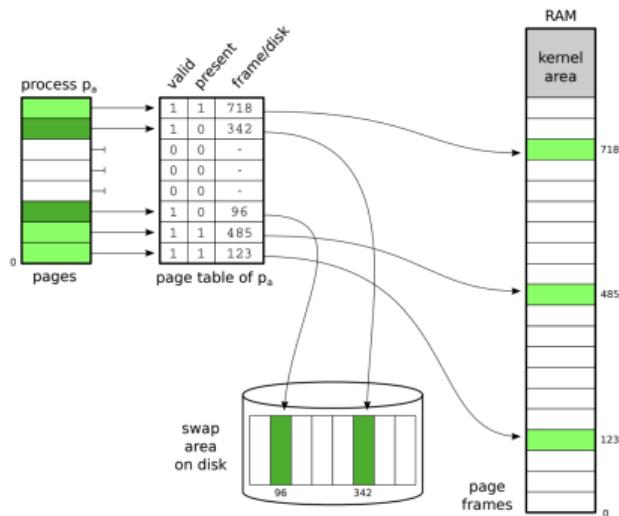
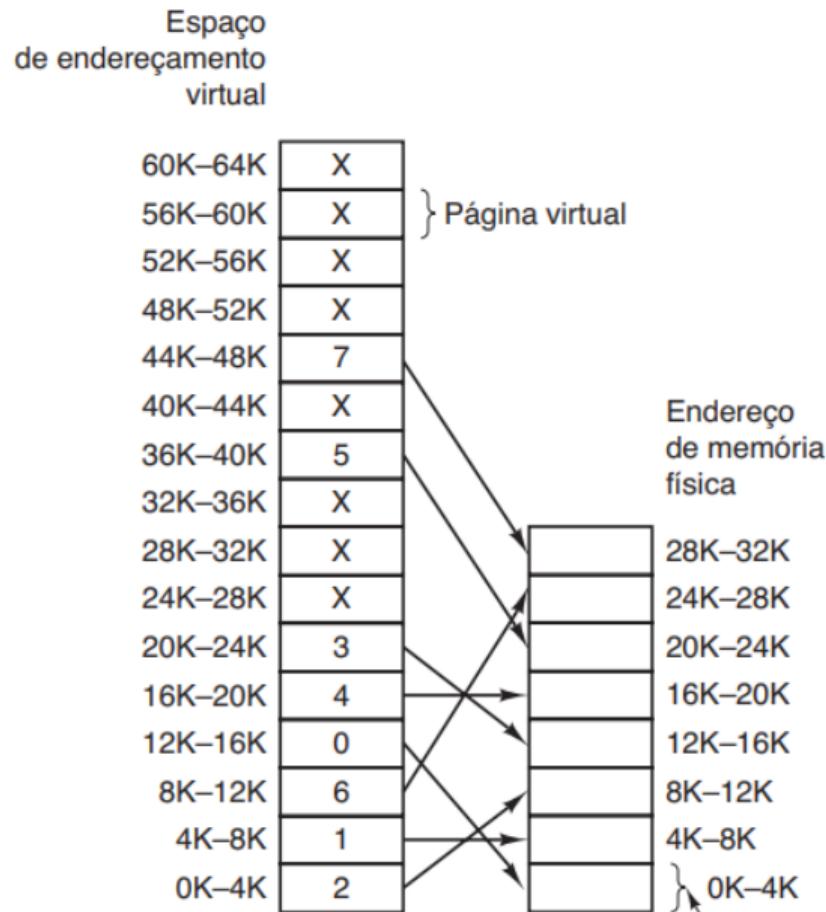


Figura 5: Paginação sob demanda (Maziero, 2019).

Paginação Sob Demanda IV



- › **Cadeia de referências:** sequência de páginas acessadas durante uma execução.
- › Usada para **estudar algoritmos de substituição de páginas.**
- › Cadeias reais são muito longas (*milhões de referências por segundo*).
- › Exemplo de cadeia usada neste texto:

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

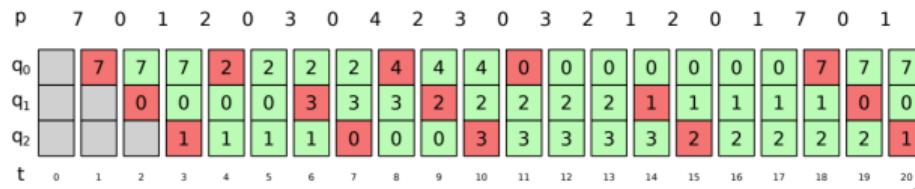
Observação

Essa sequência será utilizada para demonstrar o funcionamento dos algoritmos de substituição, como **FIFO**, **LRU** e **Ótimo**.

FIFO

First-In, First-Out

- › A página mais antiga (carregada há mais tempo) é a primeira a ser substituída.
- › Implementação simples usando uma fila.
- › Pode causar a **anomalia de Belady**.



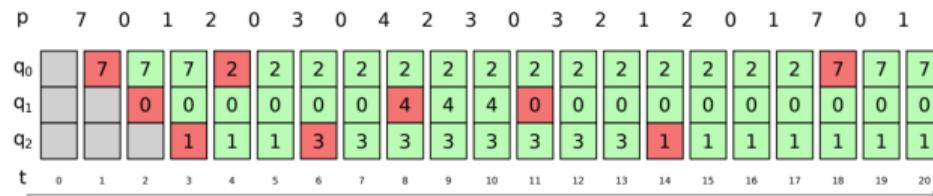
O algoritmo FIFO gera 15 faltas de página.

Figura 8: Substituição FIFO (OLIVEIRA; CARISSIMI; TOSCANI, 2001).

ÓTIMO

Substituição Ideal (Teórica)

- Substitui a página que **não será usada por mais tempo** no futuro.
- Fornece o número mínimo de falhas de página possível.
- Impossível de implementar na prática — usado como **referência teórica**.



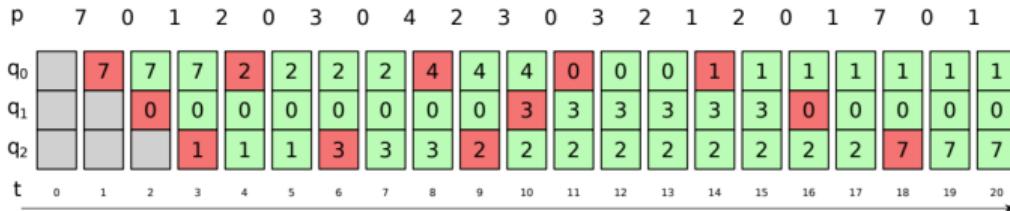
O algoritmo ótimo gera 9 faltas de página.

Figura 9: Algoritmo ótimo (referência teórica) (Maziero, 2019).

LRU

Least Recently Used

- › Substitui a página **menos recentemente utilizada**.
- › Requer rastrear o uso recente das páginas.
- › Pode ser implementado por contador, pilha ou bits de referência.



O algoritmo LRU gera 12 faltas de página.

Figura 10: Algoritmo LRU (SILBERSCHATZ; GALVIN; GAGNE, 2001).

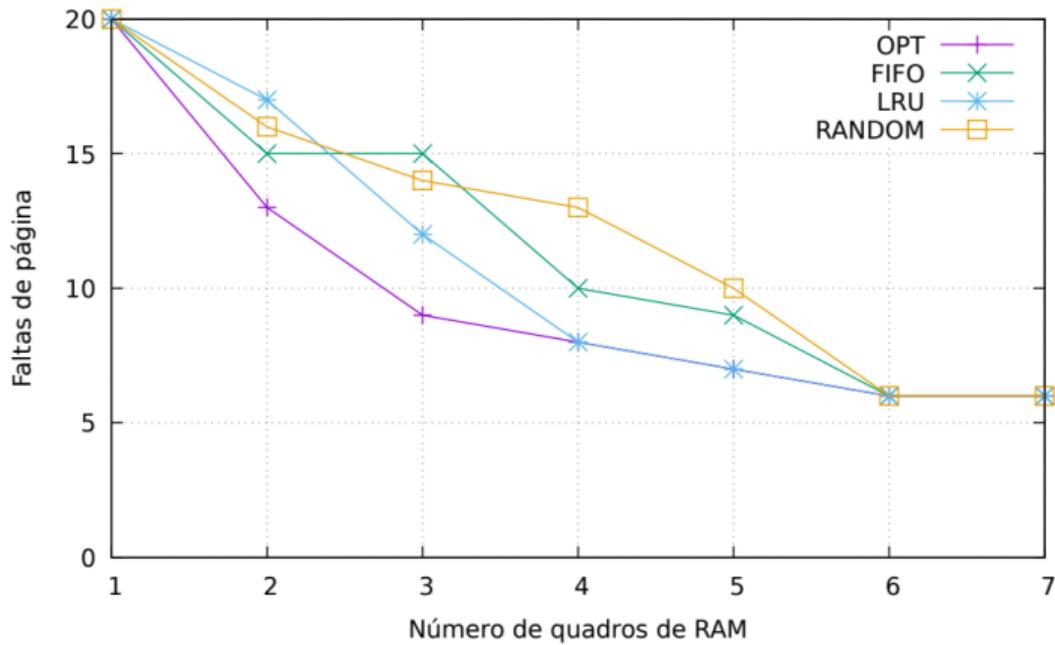


Figura 11: Comparação entre os algoritmos (Maziero, 2019).

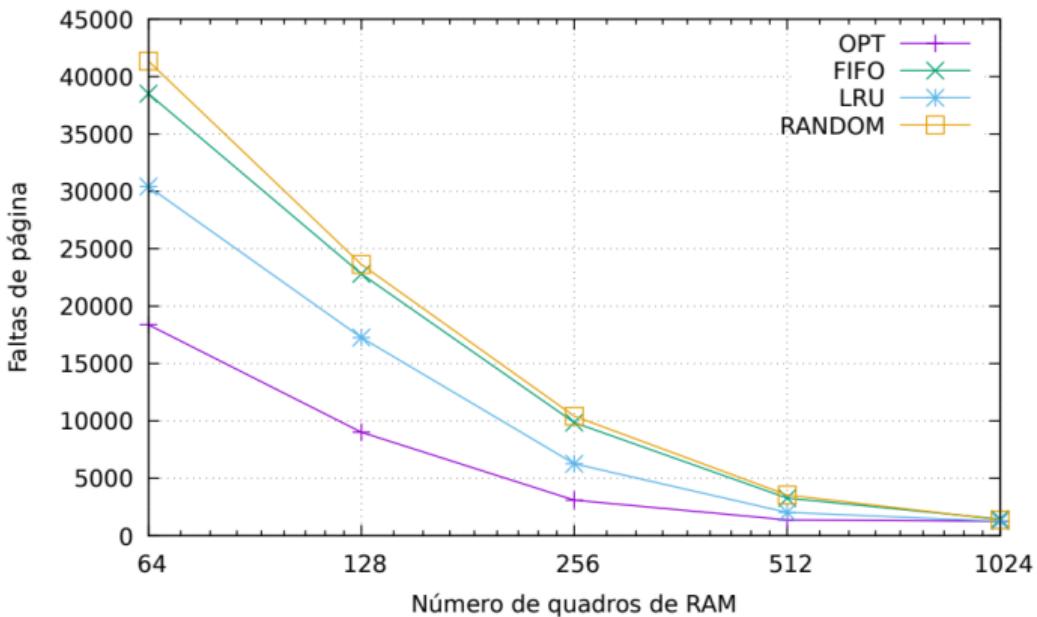


Figura 12: Cadeia de referência real (GCC) (Maziero, 2019).

RELÓGIO

Clock / Second Chance

- › Versão aprimorada do FIFO.
- › Cada página possui um **bit de uso**.
- › Quando uma página é usada, o bit é definido como 1.
- › O ponteiro do “relógio” percorre as páginas:
 - » Se o bit = 0, a página é substituída;
 - » Se o bit = 1, ele é zerado e a página recebe uma “segunda chance”.

Segunda Chance (Relógio) II

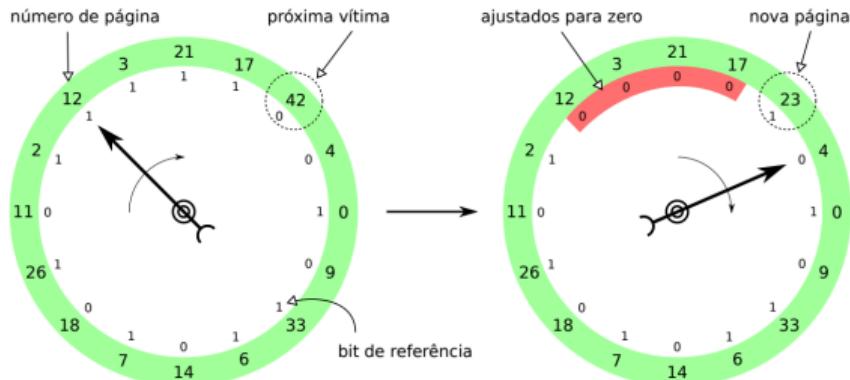
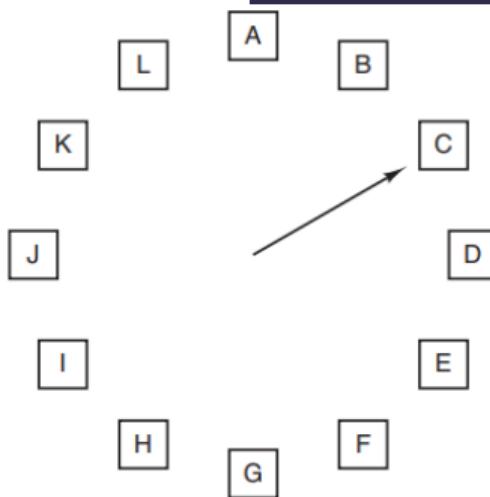


Figura 13: Algoritmo do Relógio (Maziero, 2019).

Segunda Chance (Relógio) III



Quando ocorre uma falta de página,
a página indicada pelo ponteiro
é inspecionada. A ação executada
depende do bit R:

R = 0: Remover a página

R = 1: Zerar R e avançar o ponteiro

Figura 14: Algoritmo do Relógio (TANENBAUM, 2010).

Segunda Chance (Relógio) IV

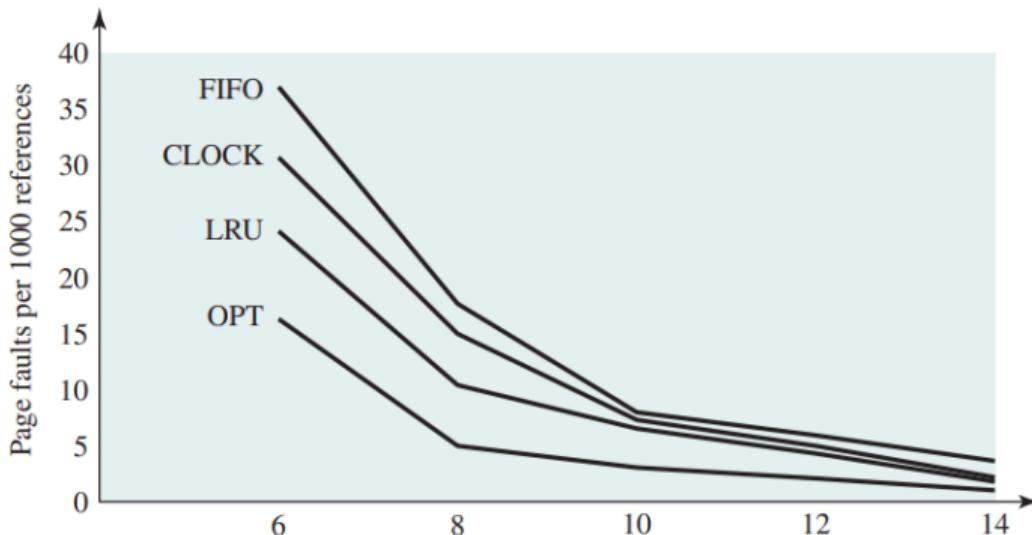


Figura 15: Comparação entre os algoritmos (STALLINGS, 2009).

NRU

Not Recently Used

- › Classifica as páginas com base nos bits de **referência (R)** e **modificação (M)**.
- › A substituição ocorre com prioridade para a classe 1.
- › O sistema divide as páginas em quatro classes:

Tabela 1: Combinações dos bits R (Referência) e M (Modificação)

R	M	Status
0	0	Nem usada nem modificada; melhor escolha!
0	1	Sem uso, mas alterada (<i>salvar antes de remover</i>).
1	0	Usada recentemente.
1	1	Usada recentemente e alterada; pior escolha!

ENVELHECIMENTO

Aproximação do LRU

- › Mantém um contador para cada página.
- › Periodicamente, o contador é deslocado para a direita e o bit de referência é inserido na esquerda.
- › Páginas mais usadas têm contadores maiores.
- › A menor contagem indica a página a ser substituída.

Algoritmo de Envelhecimento II

$$\begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \left[\begin{array}{c} R \\ 0 \\ 1 \\ 0 \\ 1 \end{array} \right] \left[\begin{array}{c} contadores \\ 0000 \ 0011 \ (3) \\ 0011 \ 1101 \ (61) \\ 1010 \ 1000 \ (168) \\ 1110 \ 0011 \ (227) \end{array} \right] \Rightarrow \left[\begin{array}{c} R \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \left[\begin{array}{c} contadores \\ 0000 \ 0001 \ (1) \\ 1001 \ 1110 \ (158) \\ 0101 \ 0100 \ (84) \\ 1111 \ 0001 \ (241) \end{array} \right]$$

Figura 16: Algoritmo de envelhecimento (Maziero, 2019).

WORKING SET

Localidade e Eficiência

- › O **conjunto de trabalho (*working set*)** é o conjunto de páginas usadas recentemente por um processo.
- › O algoritmo **WSClock** combina o conceito de relógio com o conjunto de trabalho.
- › Evita substituir páginas que ainda fazem parte do conjunto ativo do processo.

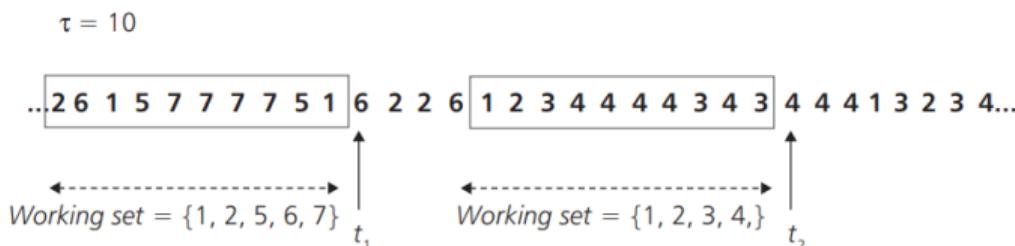


Figura 17: Algoritmo WSClock (OLIVEIRA; CARISSIMI; TOSCANI, 2001).

WSCLOCK

Localidade e Eficiência

- › O **conjunto de trabalho (working set)** é o conjunto de páginas usadas recentemente por um processo.
- › O algoritmo **WSClock** combina o conceito de relógio com o conjunto de trabalho.
- › Evita substituir páginas que ainda fazem parte do conjunto ativo do processo.

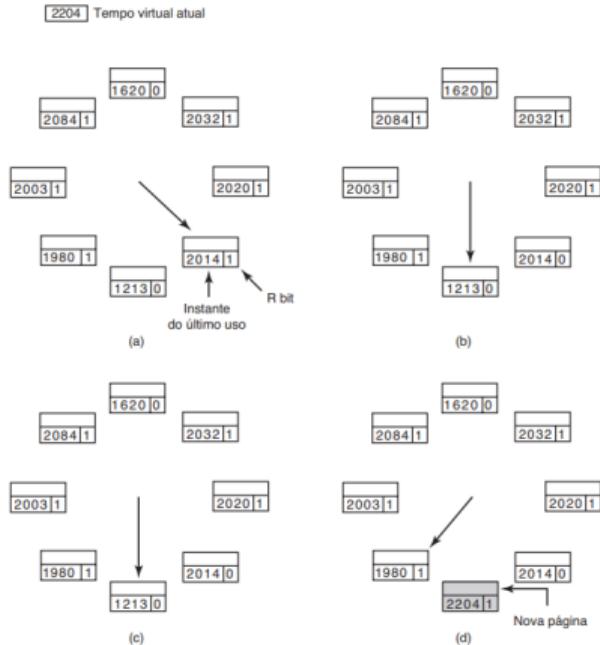


Figura 18: Algoritmo WSClock (TANENBAUM, 2010).

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU (não usado recentemente)	Aproximação muito rudimentar do LRU
FIFO (primeiro a entrar, primeiro a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
LRU (usada menos recentemente)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não frequentemente usado)	Aproximação bastante rudimentar do LRU
Envelhecimento (<i>aging</i>)	Algoritmo eficiente que aproxima bem o LRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

Figura 19: Algoritmos de substituição de páginas (TANENBAUM, 2010).

Belady

Mais quadros, mais falhas

- › Em certos algoritmos, como o FIFO, aumentar o número de quadros pode aumentar o número de falhas de página.
- › Esse fenômeno é chamado de **Anomalia de Belady**.
- › Não ocorre em algoritmos como LRU e Ótimo.

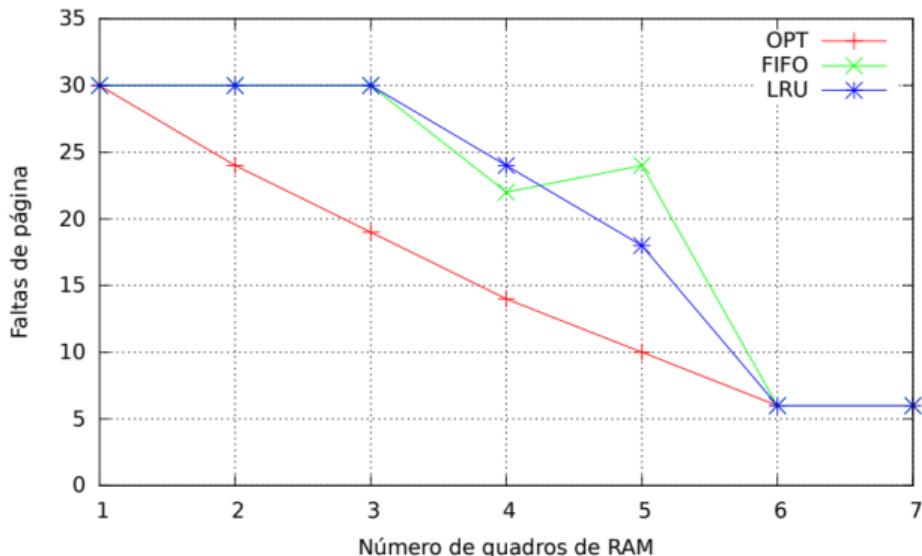


Figura 20: Anomalia de Belady (Maziero, 2019).

Thrashing

Efeito do Excesso de Falhas de Página

- › Ocorre quando o sistema passa mais tempo trocando páginas do que executando processos.
- › Sintoma de sobrecarga da memória virtual.
- › O desempenho do sistema cai drasticamente.
- › Soluções possíveis:
 - » Ajustar o número de quadros por processo;
 - » Suspender processos temporariamente (reduzir multiprogramação);
 - » Utilizar o algoritmo de conjunto de trabalho (WS).

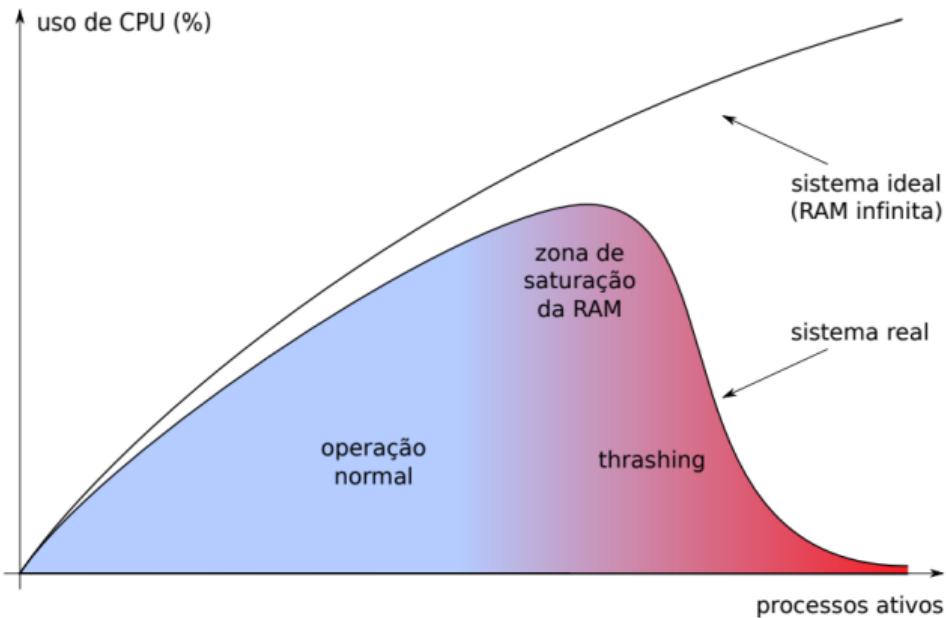


Figura 21: Efeito do thrashing (Maziero, 2019).

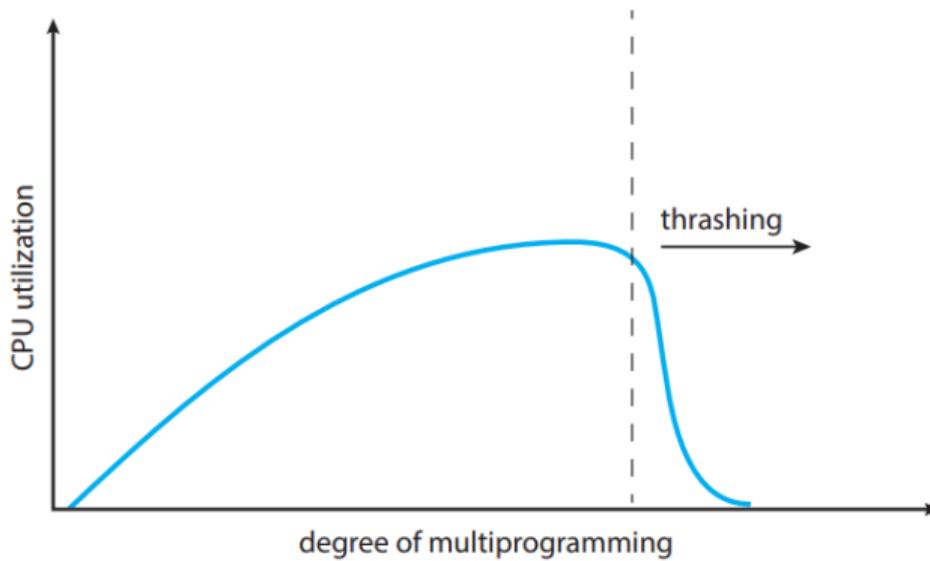
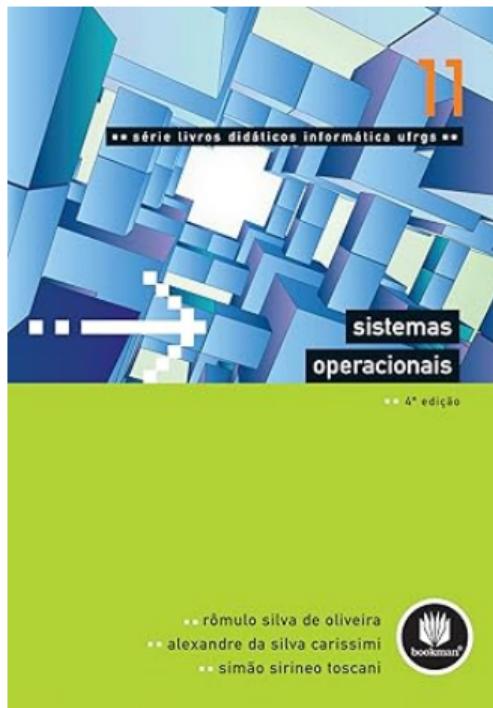


Figura 22: Efeito do thrashing (SILBERSCHATZ; GALVIN; GAGNE, 2001).

Capítulo 7 (OLIVEIRA; CARISSIMI; TOSCANI, 2001)



Bibliografia

-  OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. S. **Sistemas Operacionais**. 2. ed. Porto Alegre: Sagra-Luzzatto, 2001.
-  STALLINGS, William. **Operating Systems: Internals and Design Principles**. 6. ed. Upper Saddle River, NJ: Prentice-Hall, 2009.
-  TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.

 SILBERSCHATZ, Abraham; GALVIN, Peter; GAGNE, Greg. **Sistemas Operacionais: Conceitos e Aplicações**. Rio de Janeiro: LTC, 2001.

 TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas Operacionais: Projeto e Implementação**. 2. ed. Porto Alegre: Bookman, 2000.

 MAZIERO, Carlos Alberto. **Sistemas Operacionais: Conceitos e Mecanismos [recurso eletrônico]**. Curitiba: DINF - UFPR, 2019. ISBN 978-85-7335-340-2.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Marisangila Alves, MSc
marisangila.alves@udesc.br
marisangila.com.br



JOINVILLE
CENTRO DE CIÊNCIAS
TECNOLOGICAS

UDESC
Universidade do Estado de Santa Catarina

2025/2

Sistemas Operacionais

Gerencia de Memória Virtual