



Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Computação
Bacharelado em Ciência da Computação

Sistemas Operacionais

Gerência da Memória

Prof. Rodrigo Campiolo

22/04/20

Memória Virtual

- Introdução
 - **Memória virtual** possibilita a execução de um processo sem que esteja totalmente na memória.
 - Princípios básicos:
 - carregar sob demanda páginas/segmentos.
 - manter somente páginas/segmentos necessários na memória.
 - Vantagens:
 - executar processos maiores que a memória.
 - aumentar o grau de multiprogramação.
 - diminuir o número de operações E/S e *swap*.

Memória Virtual

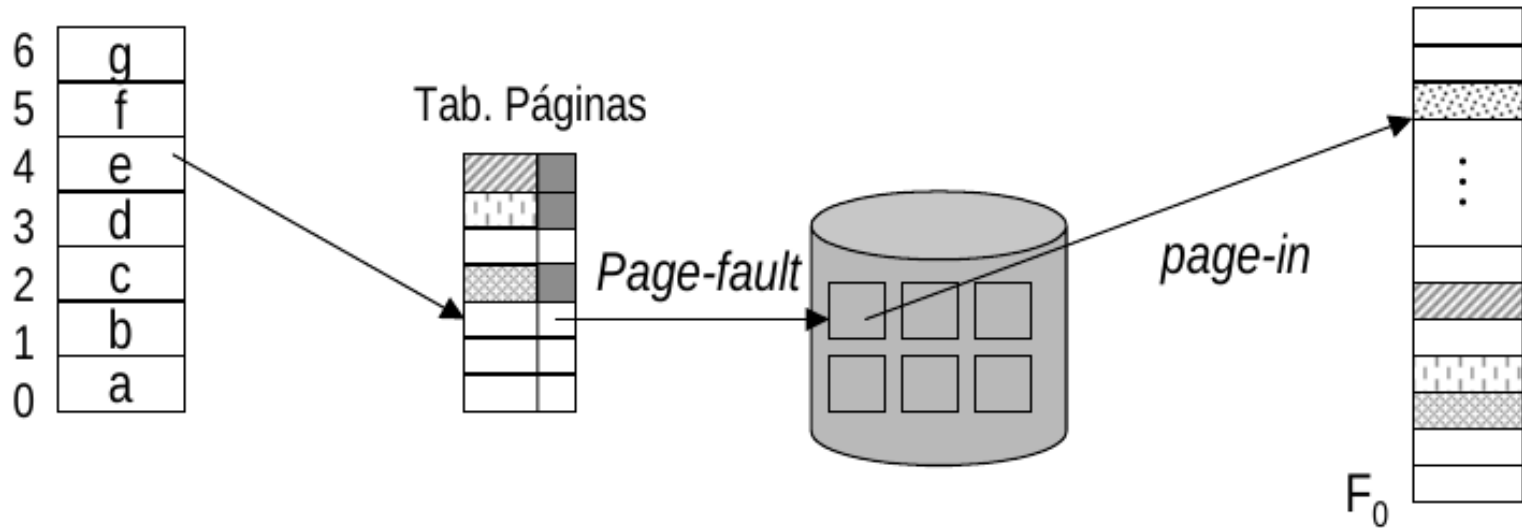
- Princípio da Localidade de Referência
 - Processos executam acessos a instruções e dados limitados a sequências e trechos.
 - sequência de instruções (**i, i+1, i+2, ..., n**).
 - laços de repetição (**localização temporal**).
 - acessos a elementos de vetores (**localização espacial**).
 - Implicações:
 - Trechos de código e dados precisam estar na memória somente quando necessários.
 - Previsibilidade dos próximos trechos.

Memória Virtual

- Paginação por demanda
 - A página é carregada para a memória somente quando é necessária.
 - Associar uma referência a cada página de presente ou não na memória primária.
 - uso de um bit de controle em cada entrada da tabela de páginas.
 - Uma exceção de **page-fault** é gerada se a página não está na memória primária.

Memória Virtual

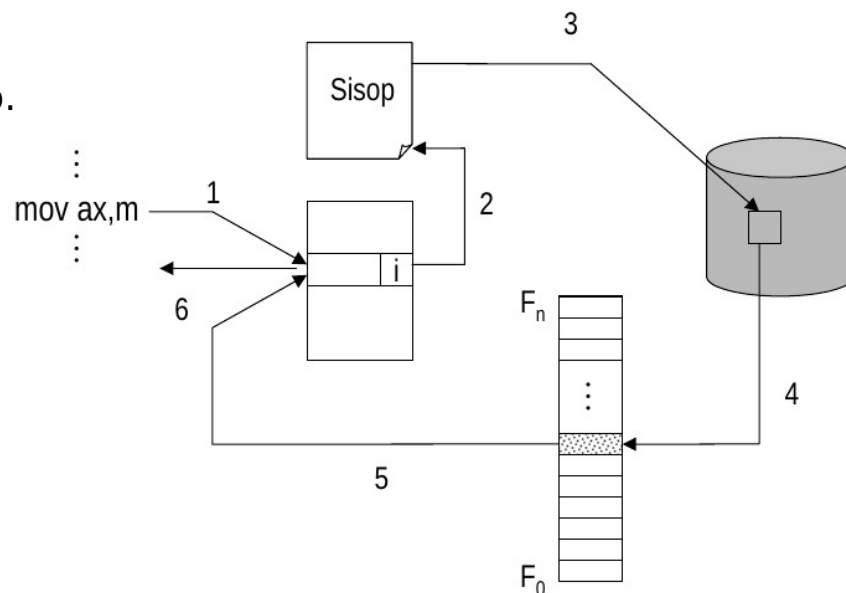
- Paginação por demanda



Memória Virtual

■ Paginação por demanda

1. Acesso a um endereço de memória.
2. Ocorre uma exceção de *page-fault*.
3. Verifica se referência é válida e determina a localização da página no disco.
4. Solicita a leitura da página do disco para o quadro.
Bloqueia o processo e escalona outro.
Interrupção de término de leitura.
5. Atualiza a tabela de páginas.
Processo transita de bloqueado para pronto.
6. Processo executa novamente o acesso.



Memória Virtual: Implementação

- Requisitos
 - Política de carga de página.
 - Política de localização de página.
 - Política de substituição de página.
 - Partição de *swap*.
 - área específica do disco para armazenar páginas.
 - organizada para otimizar o acesso.

Memória Virtual: Implementação

- Política de carga de página
 - Carrega uma página para o um quadro.
 - Situações:
 - quadro livre: carregar página no quadro.
 - não há quadro livre:
 - liberar espaço transferindo da memória para o disco (*page-out*).
 - selecionar a página vítima usando a política de substituição.
- Otimizações:
 - carregar mais de uma página na memória.
 - páginas não modificadas e somente leitura não precisam ser armazenadas no disco.

Memória Virtual: Implementação

- Política de localização de página
 - Localizar as páginas de um processo na memória primária.
 - Realizada pela MMU e transparente para o sistema operacional.

Memória Virtual: Implementação

- Política de substituição de páginas
 - Não há quadros disponíveis para carregar novas páginas.
 - Liberar um quadro para atender à falta de página.
 - **Algoritmos de substituição de páginas** são responsáveis por selecionar uma página vítima.

Critérios:

- ☠ idade da página.
- ☠ frequência de acessos à página.
- ☠ marca de tempo do último acesso.
- ☠ prioridade do processo (p. ex. tempo real).
- ☠ conteúdo da página (p. ex. executável).
- ☠ páginas especiais (p. ex. chaves criptográficas).

Memória Virtual: Implementação

- Substituição de página: Bits auxiliares
 - Facilitam e otimizam a implementação do mecanismo de substituição de páginas.
- bit de referência (*reference bit*)
 - indica se a página foi acessada em um intervalo de tempo.
- bit de sujeira (*dirty bit*) ou modificação
 - indica se a página foi modificada durante a execução.
- bit de trava (*lock bit*)
 - indica que uma página não pode ser vítima.

Algoritmos de Substituição de Páginas

- Algoritmo Ótimo
 - Retira da memória as páginas que demorarão mais para ser referenciadas.
 - Problema: prever as páginas que serão referenciadas primeiro.
 - Interessante para comparar com outros algoritmos de paginação.



Next pages are

Fonte: <https://vignette.wikia.nocookie.net/>

Algoritmos de Substituição de Páginas

- NRU (Not Recently Used)
 - Substitui as páginas não usadas recentemente.
 - Utiliza os bits R e M para selecionar as páginas vítimas.
 - R: página referenciada
 - M: página modificada
 - O bit R é limpo periodicamente (p. ex. interrupção de relógio)

Algoritmos de Substituição de Páginas

- NRU (Not Recently Used)
 - Faz uso de 4 classes:
 - Classe 0: não referenciada, não modificada.
 - Classe 1: não referenciada, modificada.
 - Classe 2: referenciada, não modificada.
 - Classe 3: referenciada, modificada.
 - Substitui página aleatória de classe mais baixa.
 - Vantagens:
 - fácil de compreender
 - moderadamente eficiente de implementar
 - proporciona um desempenho satisfatório.

Algoritmos de Substituição de Páginas

- FIFO (First in, First out)
 - Substitui a página mais antiga (primeira na fila)
 - Problema: a página mais antiga pode ainda estar sendo usada.
 - Raramente é usado esse algoritmo.

Sistema com 8 páginas e 3 frames:

cadeia de referência
(reference string)

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
f_1	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
f_2	-	0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
f_3	-	-	1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1

15 Page-faults

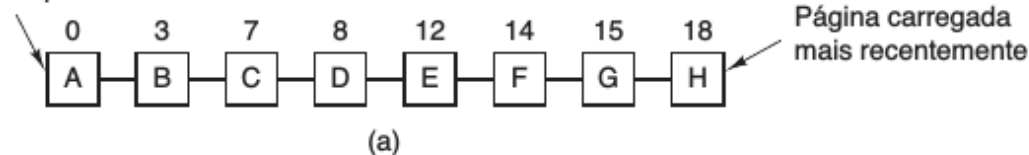
Página 0 foi substituída e em seguida referenciada!!

* Anomalia de Belady (aumentar a memória e obter mais faltas de páginas)

Algoritmos de Substituição de Páginas

- Segunda Chance
 - FIFO + bit R
 - Substitui a página mais velha não referenciada
 - Implementado como uma lista encadeada de páginas na memória:
Se $R=0$, então remove página;
Senão, $R \leftarrow 0$, então coloca no final da lista.

Página carregada primeiro



(a)



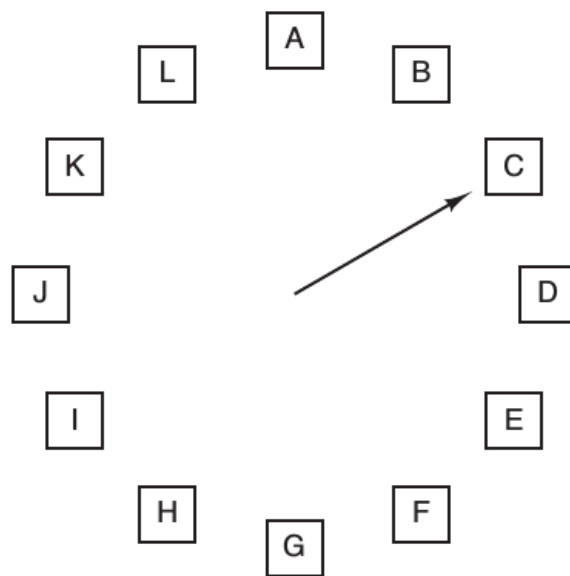
(b)

Algoritmos de Substituição de Páginas

- Relógio
 - Otimização do Segunda Chance
 - Implementado como uma lista circular de páginas na memória.
 - Ponteiro aponta para a página mais antiga.
Se $R == 0$
troca de página e desloca ponteiro;
Senão
 $R \leftarrow 0$, desloca ponteiro, continua a busca.

Algoritmos de Substituição de Páginas

- Relógio



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

Algoritmos de Substituição de Páginas

- LRU (Least Recently Used)
 - Substitui as páginas menos usadas recentemente.
 - Implementado como uma lista de páginas ordenadas pelo uso recente.
 - Problema: Custo para atualizar a lista a cada acesso à memória.
 - Solução: Hardware especial, por exemplo, contador **C** que é incrementado e armazenado para cada entrada. Percorrer a tabela de páginas ativas e localizar o menor valor de **C**.

Algoritmos de Substituição de Páginas

- LRU pode ser implementado em software
 - NFU – Not Frequently Used
 - Aging - Envelhecimento

Algoritmos de Substituição de Páginas

- NFU – Not Frequently Used
 - Substitui a página menos frequentemente usada
 - Contador de software associado a cada página
 - Percorrer todas as páginas a cada interrupção do relógio.
 - Então, somar o valor do bit R (0 ou 1) ao contador.
 - Durante a falta de página, seleciona o menor contador.
 - Problema: memória de elefante

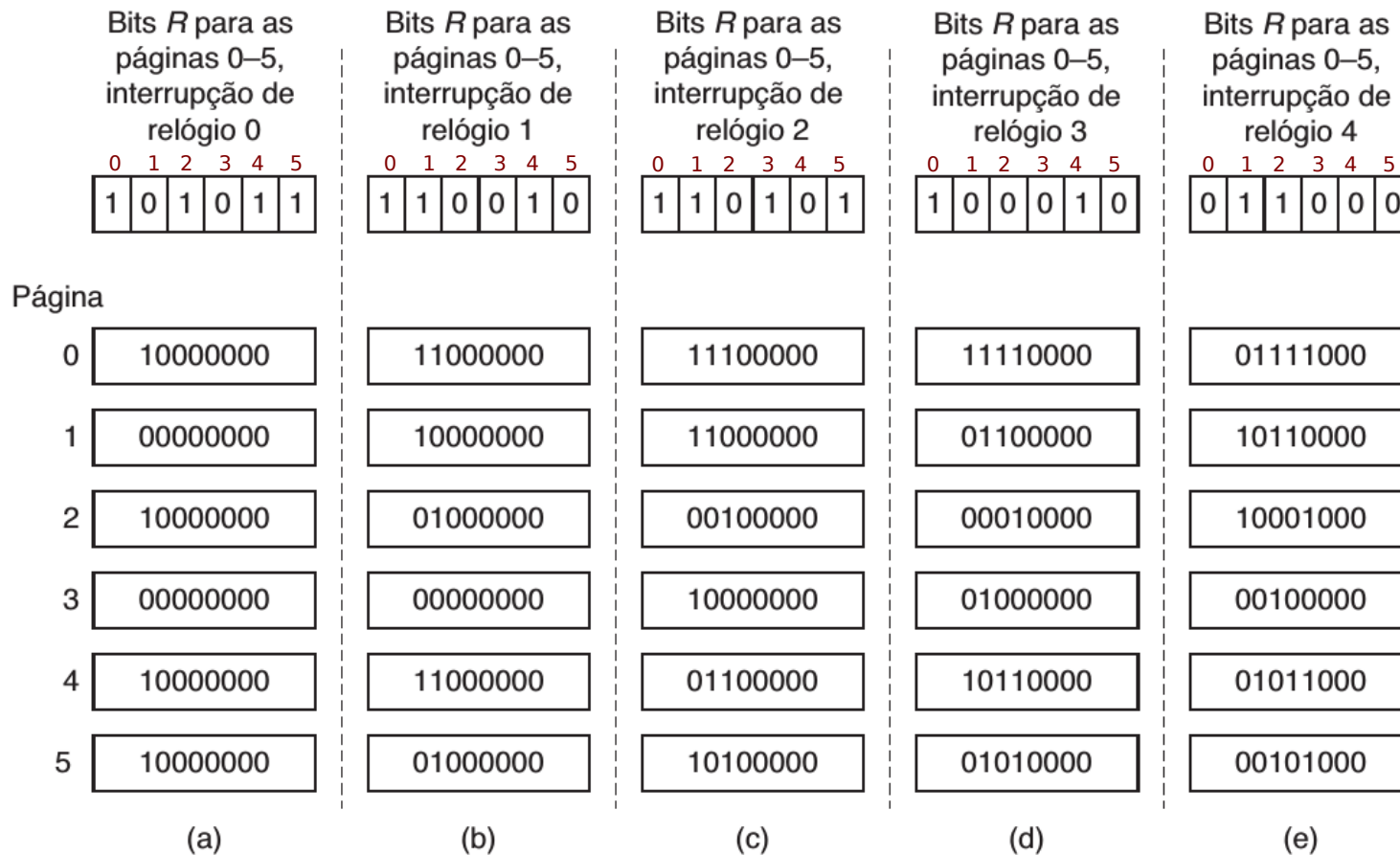
Algoritmos de Substituição de Páginas

- Aging – Envelhecimento
 - Modificação do NFU
 - A cada interrupção do relógio:
 - Contadores são deslocados 1 bit a direita
 - Acrescenta-se R ao bit mais a esquerda
 - Duas principais diferenças ao LRU:
 - Limite de passado (8 bits)
 - Entre as interrupções, não é possível determinar a ordem de acesso.

Algoritmos de Substituição de Páginas

■ Aging – Envelhecimento

As cinco interrupções de relógio são representadas por (a) a (e).



Algoritmos de Substituição de Páginas

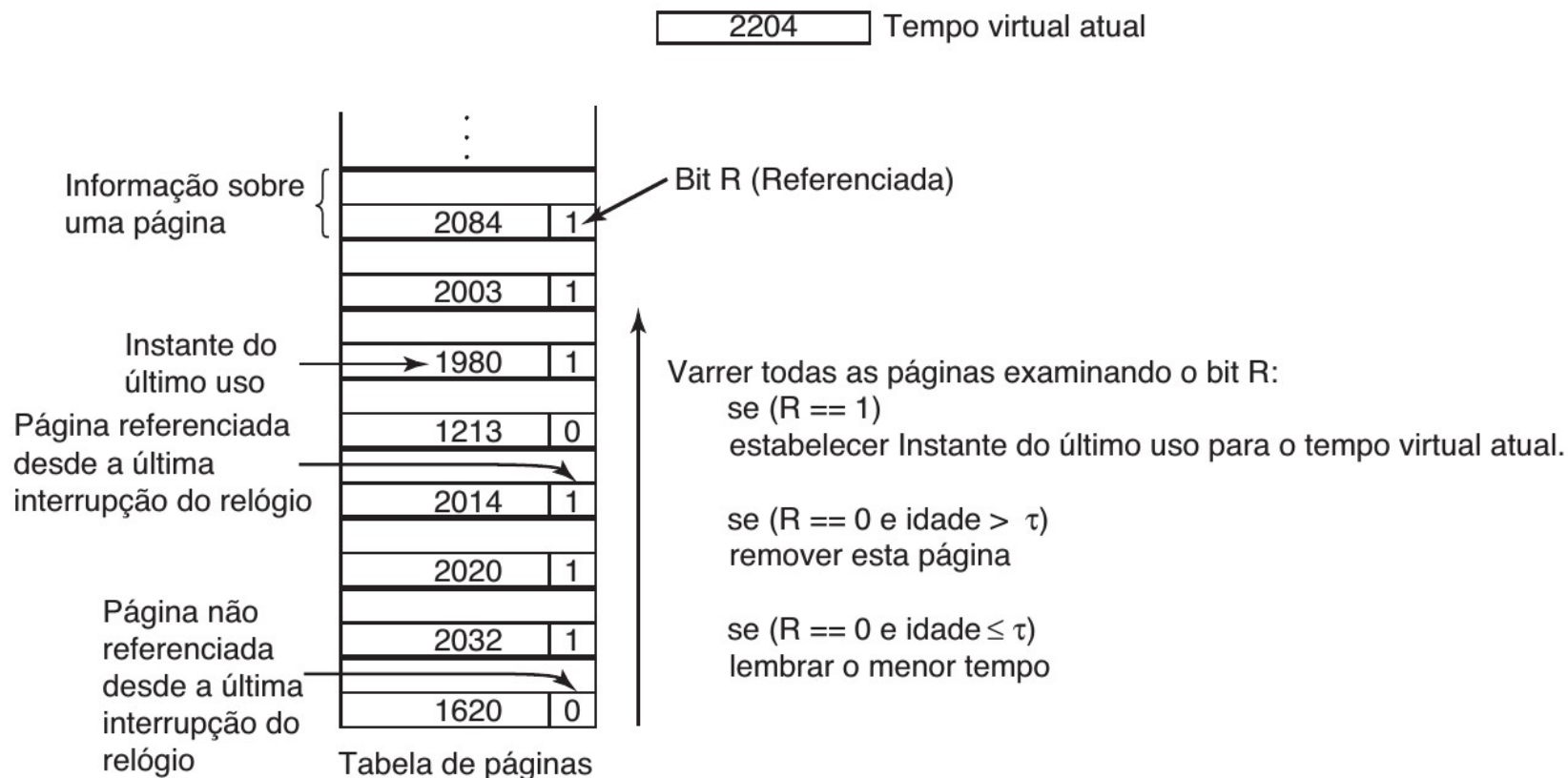
- Algoritmo do Conjunto de Trabalho
 - Conjunto de trabalho é o conjunto de páginas que um processo está usando.
 - *Thrashing* é quando ocorre a todo instante falta de páginas.
 - Tentar manter o conjunto de trabalho para não causar tantas faltas de páginas.
 - O algoritmo deve procurar remover páginas que não estão no conjunto de trabalho.

Algoritmos de Substituição de Páginas

- Algoritmo do Conjunto de Trabalho
 - Uma solução para determinar o conjunto de trabalho é usar o *tempo de execução*.
 - Por exemplo, definir o conjunto de trabalho como as páginas acessadas nos últimos X **ms** de execução.
 - Para cada processo, apenas seu tempo de execução conta.

Algoritmos de Substituição de Páginas

■ Algoritmo do Conjunto de Trabalho



τ = Supõe-se que seja múltiplas execuções da interrupção do relógio.

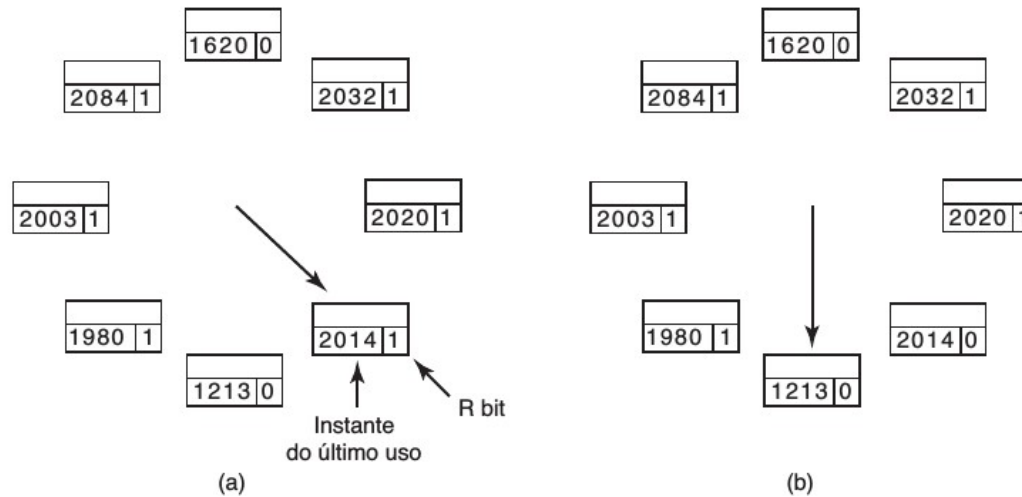
Algoritmos de Substituição de Páginas

- Algoritmo WSClock
 - Combinação do algoritmo do Conjunto de Trabalho e Relógio.
 - Simples de implementar e bom desempenho.
 - Se $R==1$: $R \leftarrow 0$ e avança.
 - Se $R==0$ e idade $> \tau$ e página limpa: substitui.
 - Se $R==0$ e idade $> \tau$ e página suja: escalona para escrita e continue a procurar.

Algoritmos de Substituição de Páginas

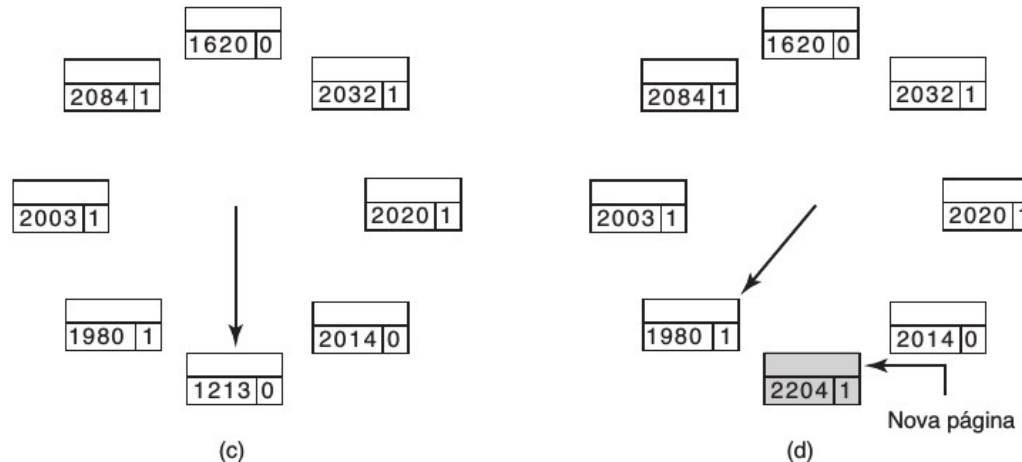
2204 Tempo virtual atual

$R = 1$



$R = 0$

WSClock



Algoritmos de Substituição de Páginas

■ Resumo

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

Referências

- OLIVEIRA, R. S. et al. **Sistemas Operacionais**. 4ª Edição e Slides online. Bookman. 2010.
- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de Sistemas Operacionais**. 9. ed. LTC, 2015.
- TANENBAUM, A. S.; BOS, H.. **Sistemas Operacionais Modernos**. 4a ed. Pearson, 2016. págs 65-58.