Algoritmos de Substituição de Páginas

Igor Gustavo Hoelscher
Renan Arend
Rogério Corrêa Medeiros

Introdução

- No momento em que ocorre uma *page fault* o sistema operacional precisa escolher uma página a ser removida da memória a fim de liberar espaço para que uma nova página seja carregada.
- Outros problemas na computação são análogos à esse: como por exemplo os blocos carregados na memória cache de um computador ou as páginas armazenadas na cache de um servidor web.
- O algoritmo ótimo que resolveria esse problema precisa saber quando as páginas serão referenciadas novamente, o que implica em um sistema não causal e é impossível de se realizar.

O Algoritmo NRU

- NRU (*Not Recently Used* – não usada recentemente).
- Usa dois bits de status: bit R (referenciado) e bit M (modificado).
- Quando o processo inicia, suas páginas ainda não estão presentes na memória. Assim que uma delas é referenciada, o bit R é colocado em 1.
- Em seguida, se esta página é modificada, o bit M é colocado em 1.
- Ao ocorrer uma *page fault* o sistema operacional separa todas as páginas em quatro categorias:
  - **Classe 0:** não referenciada, não modificada.
  - **Classe 1:** não referenciada, modificada.
  - **Classe 2:** referenciada, não modificada.
  - **Classe 3:** referenciada, modificada.

O Algoritmo NRU

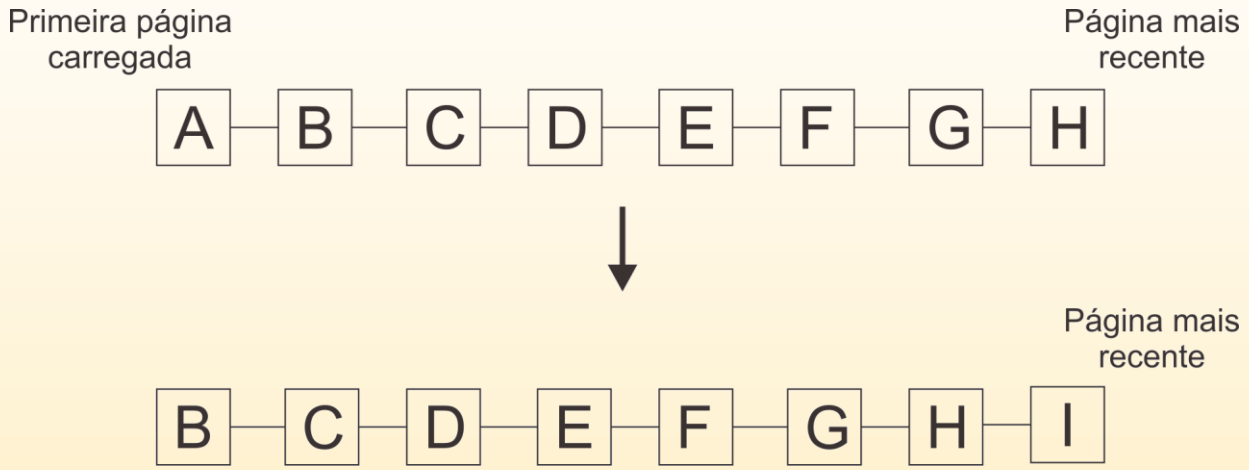
Como uma página pode ter sido modificada mas não referenciada (Classe 1)? Por que essa página é de uma classe de ordem menor a uma página referenciada, mas não modificada?

- O NRU então remove uma página aleatória da classe mais baixa que não esteja vazia.
- Entre as vantagens está a baixa complexidade de entendimento e implementação e a boa aproximação para o algoritmo ótimo.

O Algoritmo Segunda Chance

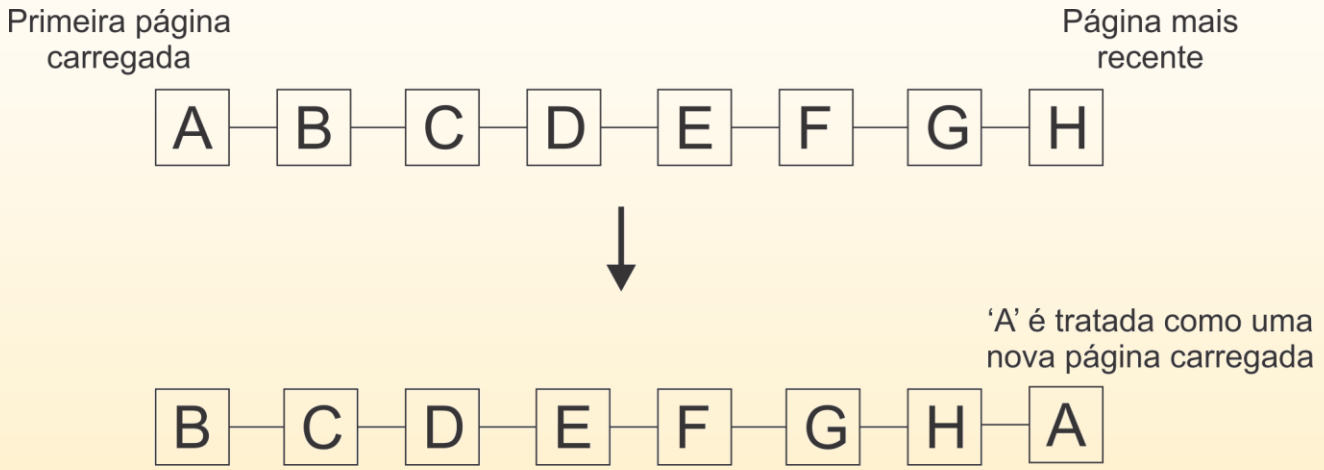
- **FIFO + bit R.**
- **Inspeciona o bit R da página mais velha.**
- Se for 0, ela é velha e não foi usada recentemente: é trocada.
- Se for 1, foi utilizada recentemente: o bit é feito 0 a página é colocada no final da fila, seu tempo de carga é modificado fazendo parecer que recém chegou na memória (recebe uma segunda chance).
- A busca continua.

O Algoritmo Segunda Chance



- Ocorre page fault no tempo 20 e Ra = 0.
- "A" é removido e novo elemento é inserido no final da fila.

O Algoritmo Segunda Chance

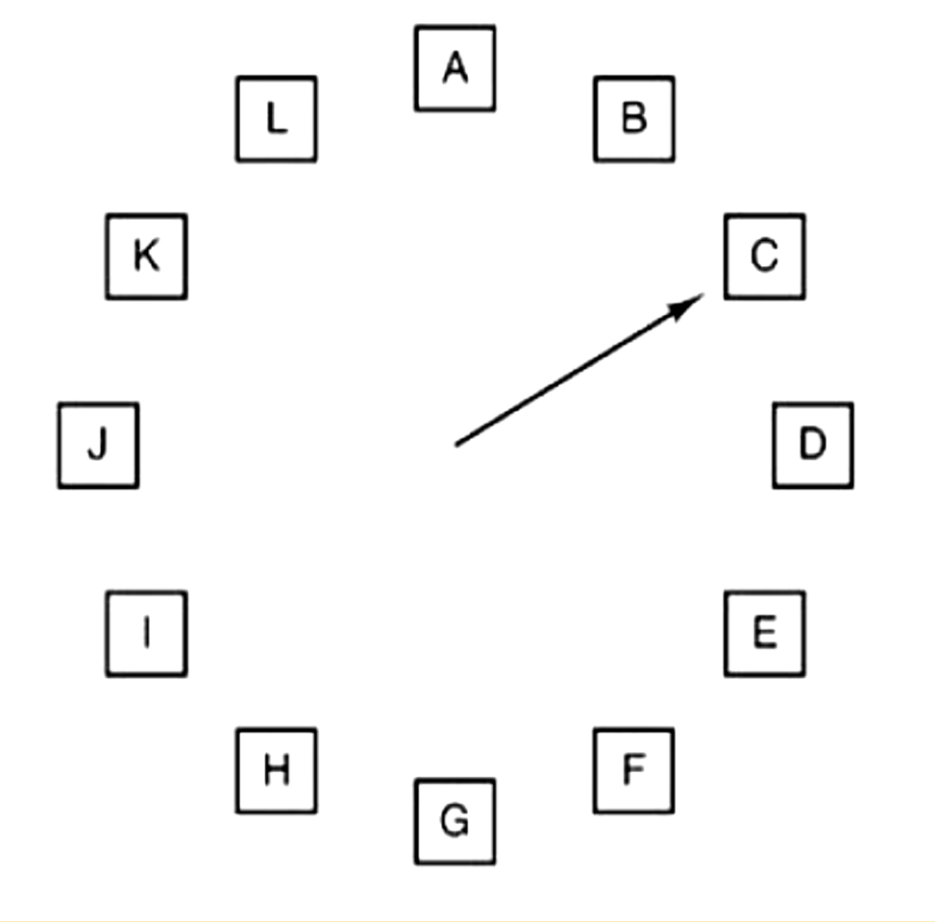


- Ocorre page fault no tempo 20 e Ra = 1.
- "A" é tratado como mais recente pagina carregada, recebendo uma "segunda chance" (Ra = 0).

O Algoritmo do Relógio

- O algoritmo segunda chance é ineficaz. Visando melhorar a técnica, surgiu o algoritmo de paginação do relógio.
- A única mudança se deve ao fato da lista estar em formato circular.
- Assim, quando houver *page fault*, a página indicada pelo ponteiro é examinada. Se o bit R estiver em 0, ela é substituída e o ponteiro avança. Se for 1, o bit é zerado e o ponteiro avança fazendo uma nova busca.
- Quando chega ao fim sem encontrar bit R = 0, o ponteiro retorna ao início, e essa página então é eliminada.

O Algoritmo do Relógio



- Quando ocorre uma falta de página, a página indicada pelo ponteiro é inspecionada. A ação executada depende do bit R.
- R = 0 : Substituir página
- R = 1 : Zerar e avançar

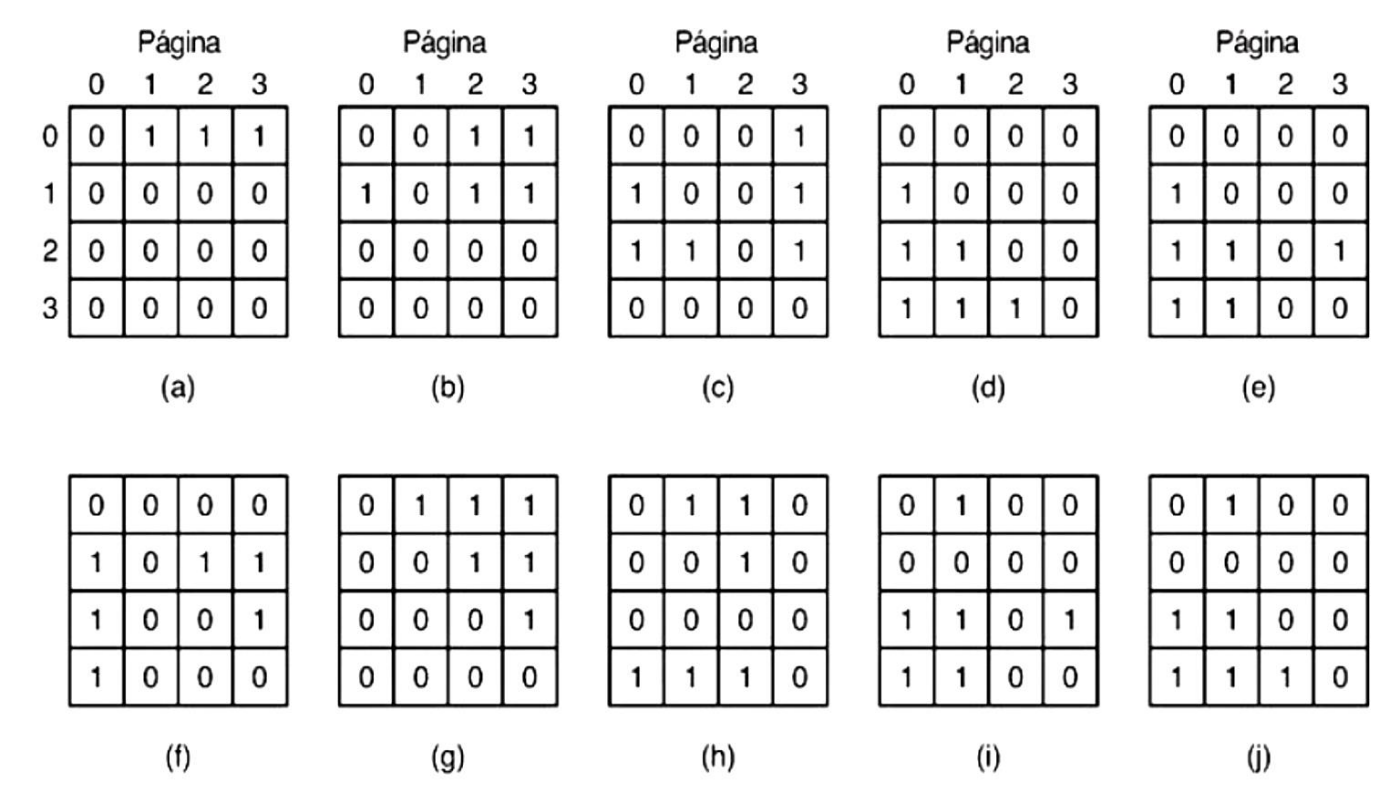
O Algoritmo LRU e sua simulação em software

- O LRU (*Least Recently Used* - usada menos recentemente) parte do princípio que as páginas usadas com mais frequência nas ultimas execuções provavelmente serão muito utilizadas novamente.
- Desse modo, quando houver *page fault* o algoritmo elimina a página não utilizada pelo período de tempo mais longo.
- Necessita que a lista seja atualizada para cada execução, além do que, uma página será buscada, se encontrada ela será excluída e posicionada na frente da lista. Um custo de operação alto, mesmo em hardware.

O Algoritmo LRU e sua simulação em software

- Há outros modos de se implementar em hardware.
- Entre eles, o mais simples é a construção de um contador, que é incrementado a cada instrução. A página referenciada necessita de um novo campo, que recebe o valor do contador no momento de sua referência. Quando houver *page fault*, o hardware examina as páginas em busca do menor valor de contador, a página "usada menos recentemente".
- Outro modo seria o auxílio de uma matrix n x n (onde n é o número de molduras de página). Quando uma página k for referenciada, as posições da linha k recebem 1 e da coluna k recebem 0.

O Algoritmo LRU e sua simulação em software

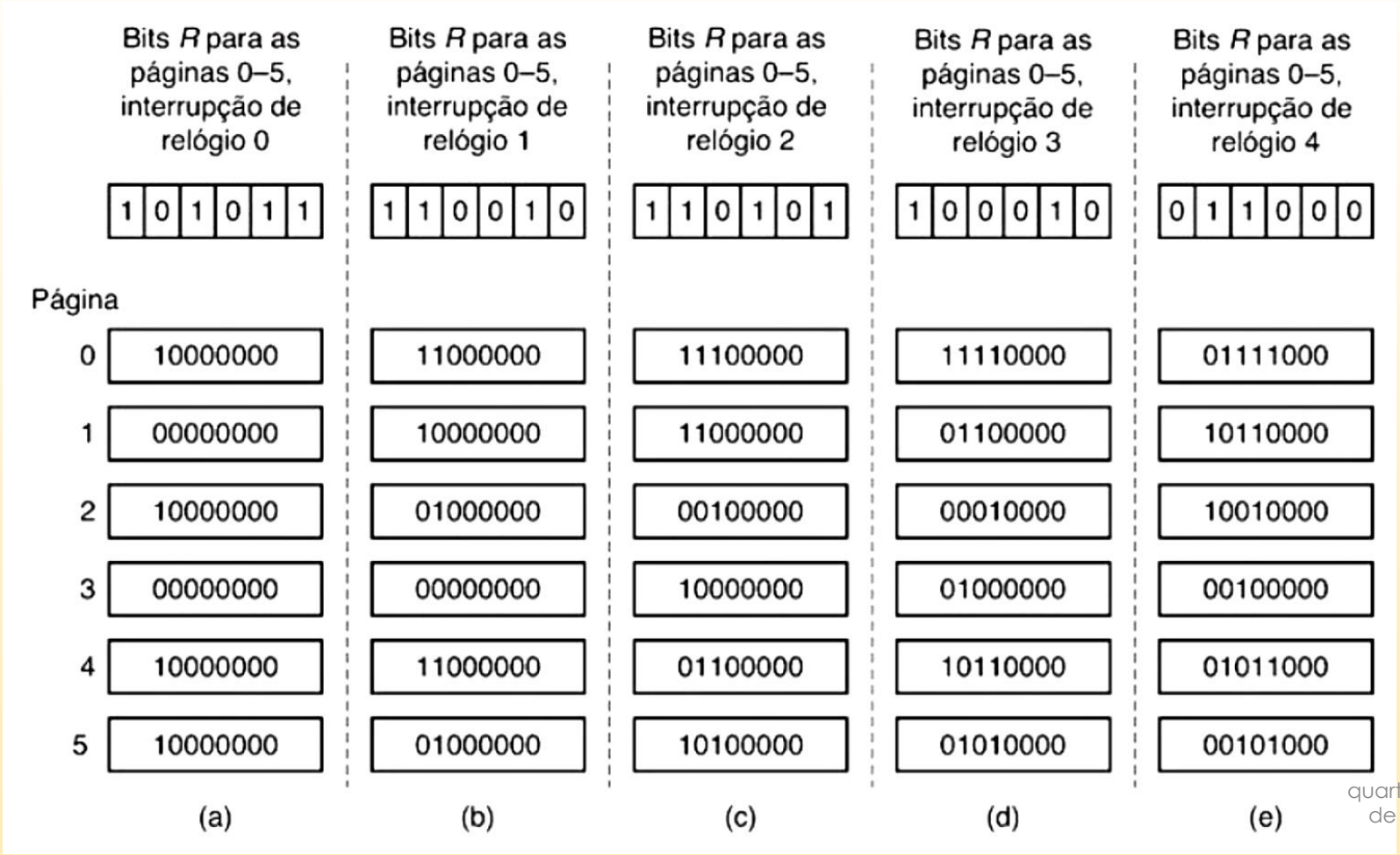


- String de referências: 0 1 2 3 2 1 0 3 2 3

O Algoritmo LRU e sua simulação em software

- Não se encontram muitos computadores com hardwares auxiliares como os que eram necessários nos dois exemplos.
- Desse modo, uma solução em software é necessária. Entre elas está o NFU (*Not Frequently Used*). Esse algoritmo requer contadores em software, cada um associado a uma página e iniciados com 0.
- Em cada interrupção, o valor do bit R é adicionado ao contador. Quando houver *page fault*, a página com menor valor em contador será eliminada.
- No entanto existem problemas...
- Uma pequena modificação (gerando o algoritmo conhecido como Algoritmo de Envelhecimento) pode aproximar a simulação do algoritmo LRU.

O Algoritmo LRU e sua simulação em software



O Algoritmo LRU e sua simulação em software

O Algoritmo de Envelhecimento simula exatamente o comportamento do algoritmo de paginação LRU?

- Se notarmos a execução anterior, deve se escolher entre a página 3 e a página 5 para eliminação. No entanto não temos como decidir qual das duas foi referenciada primeiro, já que a contagem acontece somente no final de cada ciclo.
- Outra diferença é que o algoritmo de envelhecimento tem contadores finitos (no exemplo, de 8 bits). Assim, quando dois contadores estiverem zerados no momento no momento de uma *page fault*, temos que eliminar aleatoriamente uma delas.

Algoritmo Working Set (Conjunto de trabalho)

- Paginação por demanda: páginas são carregadas na memória somente quando são necessárias;
- Pré-paginação → Working set
- Conjunto de páginas que um processo está efetivamente utilizando em um determinado tempo t;
- Objetivo principal: reduzir a falta de páginas
- Um processo só é executado quando todas as páginas
- necessárias no tempo t estão carregadas na memória;

Algoritmo Working Set (Conjunto de trabalho)

- Até então, gerará page faults.
- A idéia é determinar o working set de cada processo e certificar-se de tê-lo na memória antes de rodar o processo.
- Modelo de Conjunto de Trabalho ou pré-paginação working set w(k,t).
- Conjunto consistindo, em um dado instante t, de todas as páginas usadas pelas k referências mais recentes à memória.

Algoritmo Working Set (Conjunto de trabalho)

- O working set varia lentamente com o tempo.
- Podemos estimar o número de páginas necessárias quando o programa é trazido do disco com base em seu working set de quando foi interrompido.
- Prépaginação: consiste em carregar essas páginas antes de rodar novamente o processo.
- Implementação:
  - O SO precisa manter registro de que páginas estão no working set.
  - Quando ocorrer um page fault, encontre uma página fora do working set e a remova, caso não haja mais nenhuma página livre.

Algoritmo Working Set (Conjunto de trabalho)

- Implementação:
  - Contar as k referências mais recentes é custos.
  - Para simplificar o working → set pode ser visto como o conjunto de páginas que o processo referenciou durante os últimos t segundos de sua execução.
  - Conta o tempo individual do processo, descontando escalonamento → seu tempo virtual corrente.
  - Utiliza bit R e o tempo de relógio (tempo virtual) da última vez que a página foi referenciada.

Algoritmo Working Set(Conjunto de trabalho)

- Algoritmo:
- Pressupostos:
- O hardware define o bit R. Em cada ciclo do clock, o bit de referência é limpo.
- O tempo do working set se estende por vários ciclos do clock.
- Em cada page fault, a tabela de páginas inteira é buscada.
- À medida que cada entrada é processada, examine R.
- Se 1, escreva o tempo virtual corrente no campo Tempo do último Uso (TLU), indicando que a página estava em uso no instante da page fault, ou seja, estava no working set → não é candidata

Algoritmo Working Set (Conjunto de trabalho)

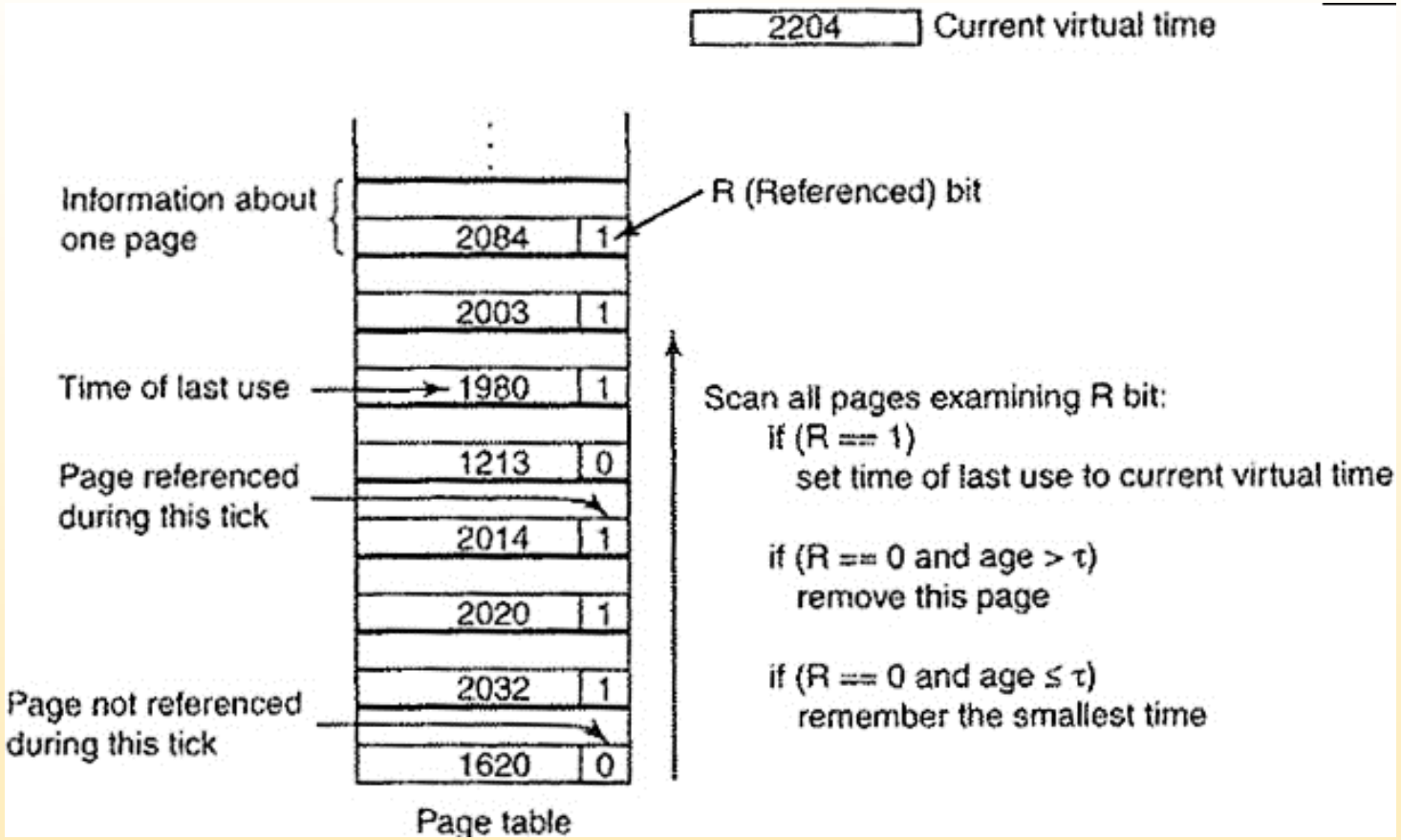
- Algoritmo:
  - Em cada page fault, a tabela de páginas inteira é buscada
  - À medida que cada entrada é processada, examine R
  - Se R=0, a página não foi referenciada no ciclo atual, e pode ser uma candidata
  - Nesse caso, se sua idade for maior que o intervalo t do working set, ela não está nele, e pode ser removida

  A busca continua atualizando as demais entradas

Algoritmo Working Set (Conjunto de trabalho)

- Se, contudo, a idade for menor que t, a página é poupada.
- Contudo, a página com maior idade é marcada.
- Se nenhum candidato for encontrado (todas as páginas estão no working set), substitua a página mais velha, dentre as com R=0.

Algoritmo Working Set (Conjunto de trabalho)

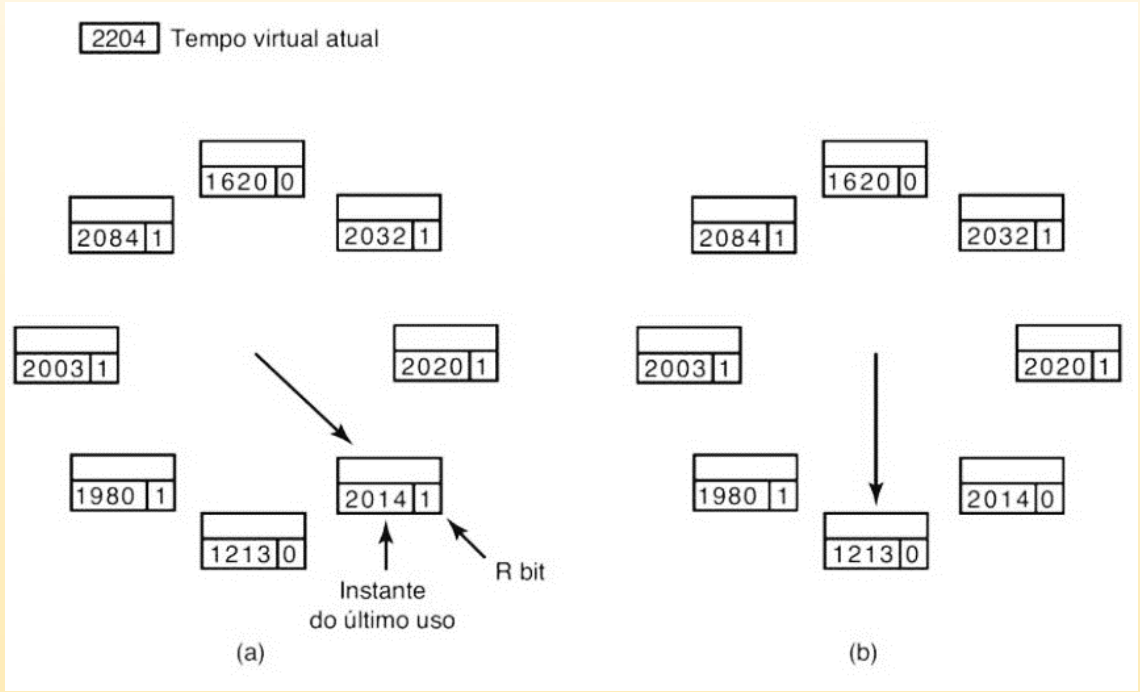


O Algoritmo WSClock

- **Relógio + Working Set.**
- Amplamente usado, devido à sua simplicidade e performance
- Utiliza lista circular de páginas;
- Inicialmente vazia, à medida que mais páginas são carregadas, entram na lista, formando um anel;
- Cada entrada contém o tempo de último uso, além dos bits R e M.

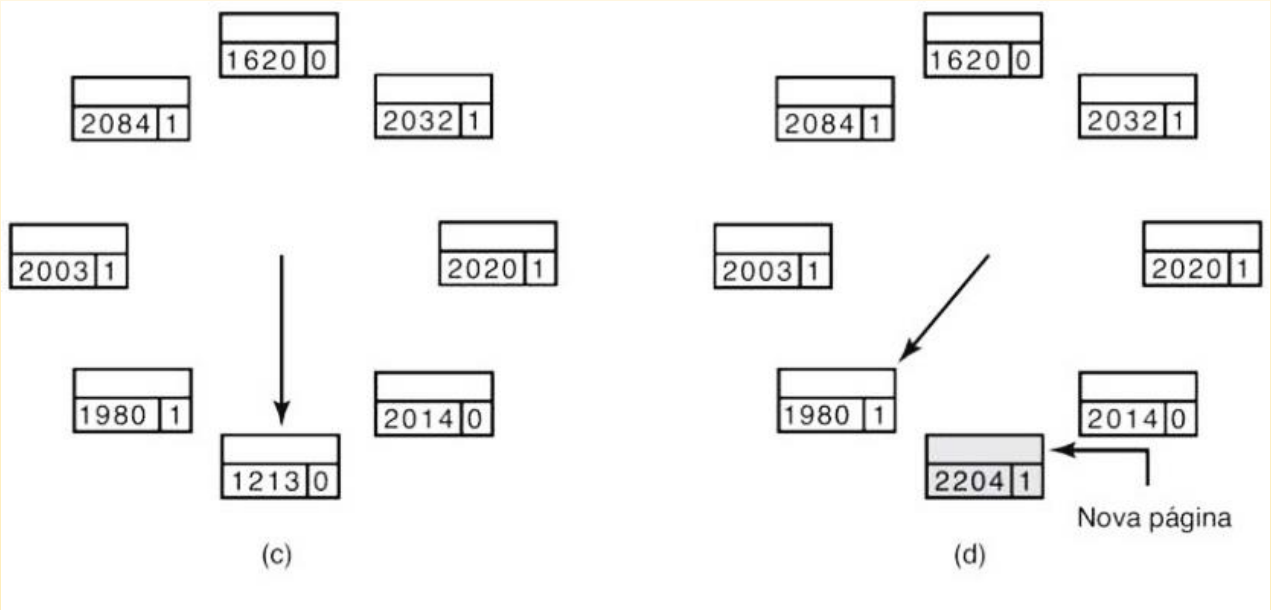
O Algoritmo WSClock

- A cada *page fault*, a página da cabeça é examinada primeiro.
- **Se R=1:**
- A página foi usada durante o ciclo de clock corrente e não é candidata a remoção. R = 0 e avança a cabeça à próxima página, repetindo o algoritmo para esta página.



O Algoritmo WSClock

- **Se R=0:**
- Se a idade for maior que o tamanho do working set t e a página estiver limpa (M=0) não está no working set e uma cópia válida existe no disco.
- A página é substituída e a cabeça da lista avança.



O Algoritmo WSClock

Se, contudo, a página estiver suja (não possui cópia válida no disco)?

- Agenda uma escrita ao disco, evitando troca de processo;
- Avança a cabeça da lista, prosseguindo da página seguinte.

Se a cabeça der uma volta completa na lista sem substituir?

- **E pelo menos uma escrita no disco foi agendada:**
  - A cabeça continua se movendo, em busca de uma página limpa
  - Em algum momento a escrita agendada será executada, marcando a página como limpa
- **E nenhuma escrita foi agendada:**
  - Todas as páginas estão no working set
  - Na falta de informação adicional, substitua qualquer página limpa
  - Se nenhuma página limpa existir, escolha qualquer outra e a escreva no disco

Referências

- TANENBAUM, Andrew S. **Sistemas Operacionais Modernos.** Pearson Prentice Hall, 3ª Edição, 2010.
- ROMAN, Norton T. MORANDINI, Marcelo. UEYAMA, Jó. **Gerenciamento de Memória Virtual: Algoritmos de Paginação.** Disponível em: *http://wiki.icmc.usp.br/images/d/dc/Aula12.pdf*.
- BRANDÃO, Humberto. **Gerência de Memória: Algoritmos de Substituição de Página.** Disponível em: *http://www.dcc.ufmg.br/~humberto/unifal/*.