

Algoritmos de Substituição de Páginas

Prof. Dr. Márcio Castro
marcio.castro@ufsc.br



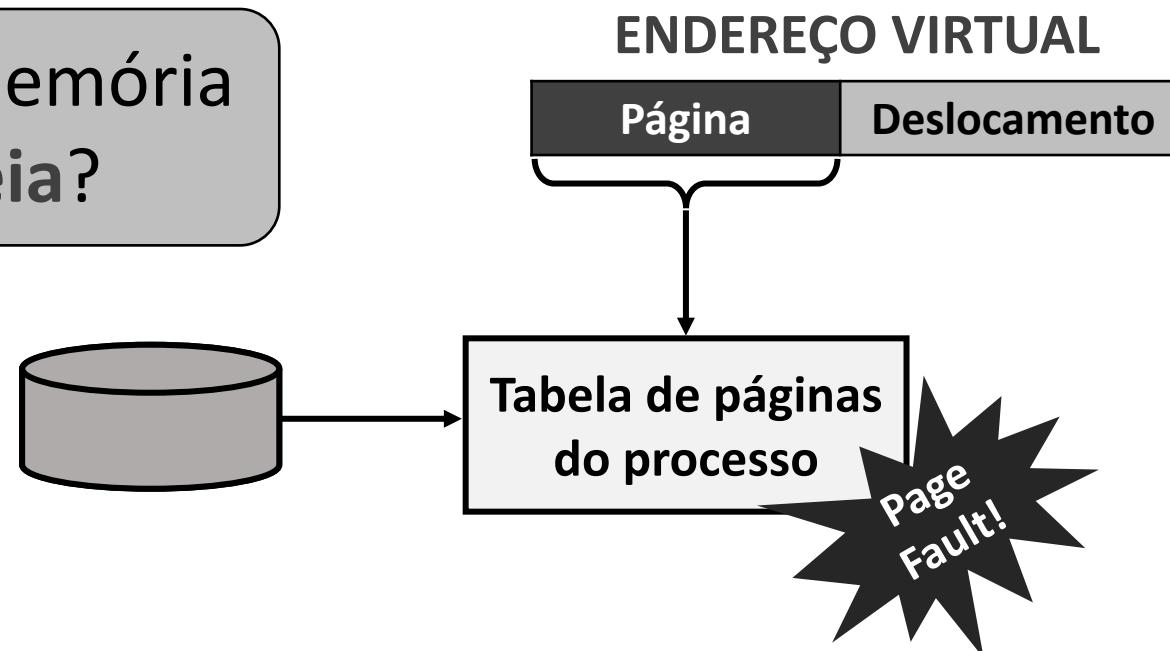
1

Introdução

Introdução

- Na técnica de **paginação**, quando a tradução entre página virtual e moldura de página não existe na **tabela de páginas do processo**, ocorre uma falta de página (*page fault*)
 - Nesse caso, a página é trazida do disco para a memória

Mas... E se a memória estiver **cheia**?



Introdução

- **Solução simples**

- Escolher aleatoriamente uma página para ser substituída

E se a página escolhida estiver sendo **muito utilizada** pelo processo?



Terá que ser **trazida de volta para a memória rapidamente**

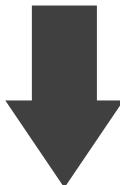


Sobrecusto!

Introdução

- **Algoritmo ótimo de substituição de páginas**
 - Associa a cada página em memória uma “etiqueta” contendo o número de instruções que serão executadas antes dela ser acessada novamente
 - Substitui a página que possui a etiqueta com o maior número

Como implementar esse algoritmo?



O sistema operacional **não tem como saber quando** cada página será acessada na próxima vez

Introdução

- Em sistemas com **memória virtual**, dois bits (**R** e **M**) são associados à cada página para que o SO esteja ciente das páginas físicas em uso
 - **R** é colocado em 1 sempre que a página é **referenciada** (lida ou escrita)
 - **M** é colocado em 1 sempre que a página é **modificada**
- Ambos os bits são colocados em 1 pelo próprio **hardware**, permanecendo assim **até que o SO os reinicialize (coloque em zero)**

Entrada da tabela de páginas

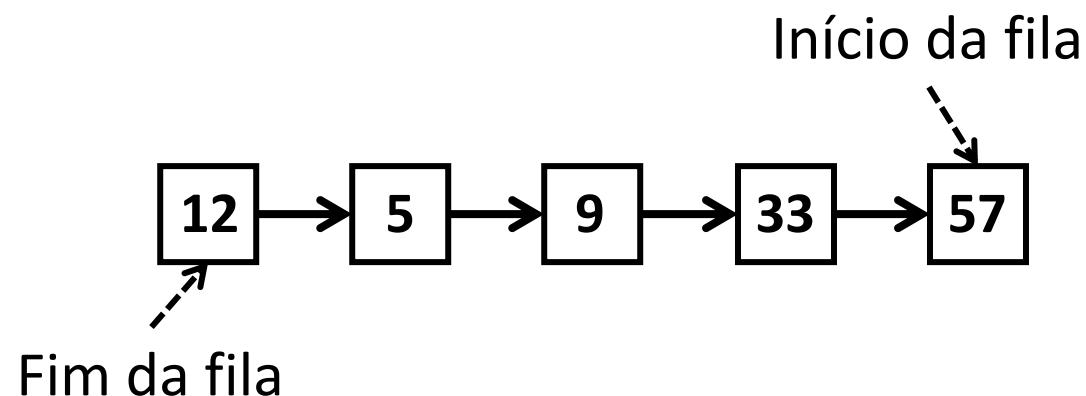


Bits **R** e **M** são usados pelos **algoritmos de substituição de páginas** para tomadas de decisão mais **inteligentes**

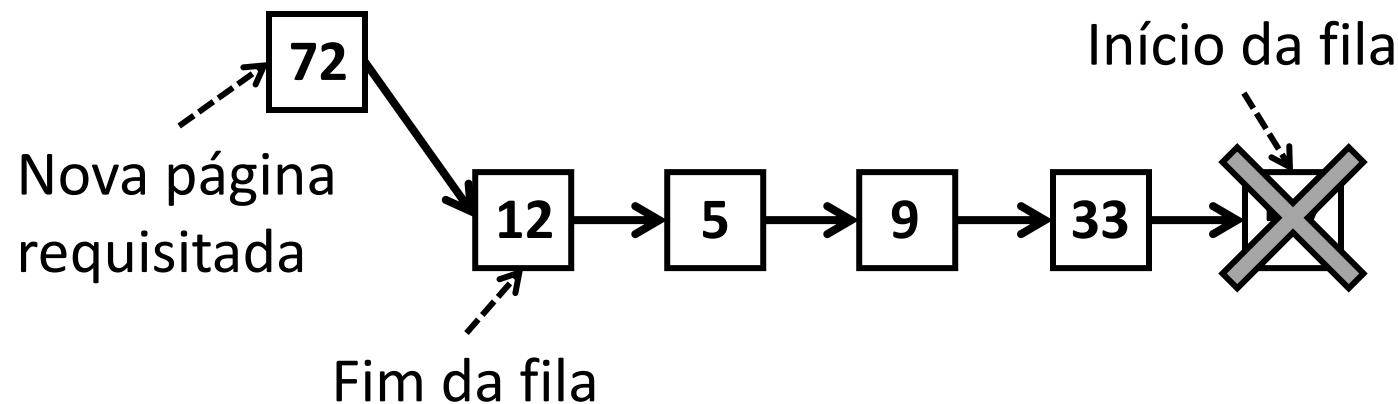
2

First-In, First-Out (FIFO)

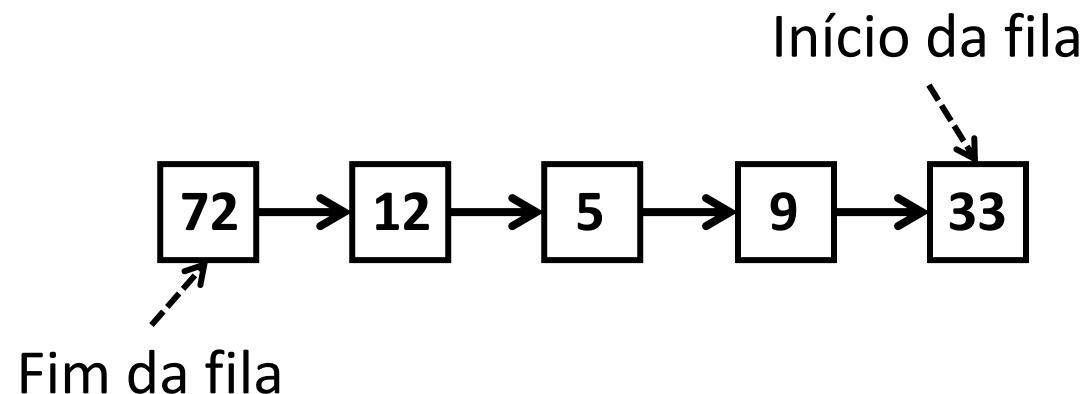
- Mantém as páginas do processo que estão atualmente em memória em uma **lista ordenada pela ordem de chegada (fila)**
- **Substituição de página**
 - Remove a página que está no **íncio da fila**
 - Insere a nova página no **fim da fila**



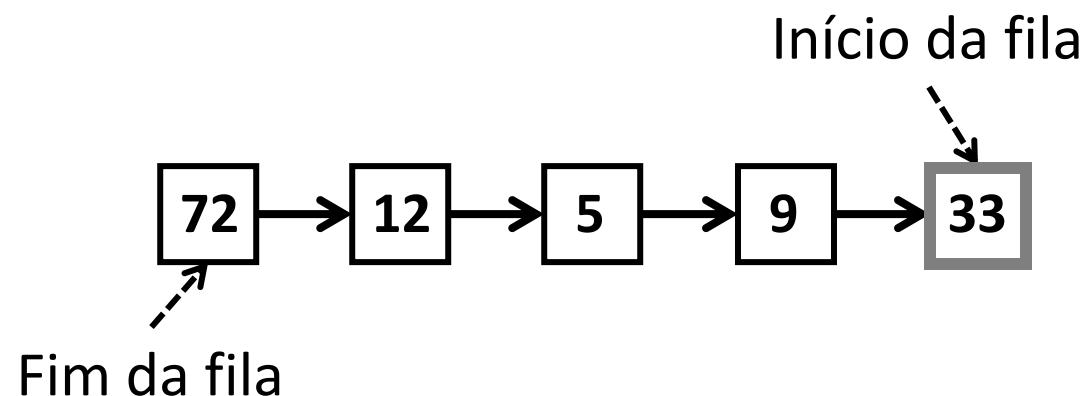
- Mantém as páginas do processo que estão atualmente em memória em uma **lista ordenada pela ordem de chegada (fila)**
- **Substituição de página**
 - Remove a página que está no **íncio da fila**
 - Insere a nova página no **fim da fila**



- Mantém as páginas do processo que estão atualmente em memória em uma **lista ordenada pela ordem de chegada (fila)**
- **Substituição de página**
 - Remove a página que está no **íncio da fila**
 - Insere a nova página no **fim da fila**



- **Problema:**
 - Pode remover páginas que estão sendo referenciadas **frequentemente**
- **Exemplo:** se a página 33 é acessada frequentemente deve-se evitar removê-la para não gerar *page faults*



3

Clock

Clock

- Organiza logicamente as páginas do processo que estão atualmente na memória em uma **lista circular**
- Um **ponteiro** aponta para a página mais antiga (**cabeça da lista**)
 - Páginas são **inseridas no final da lista**



Clock

- Organiza logicamente as páginas do processo que estão atualmente na memória em uma **lista circular**
- Um **ponteiro** aponta para a página mais antiga (**cabeça da lista**)
 - Páginas são **inseridas no final da lista**



Clock

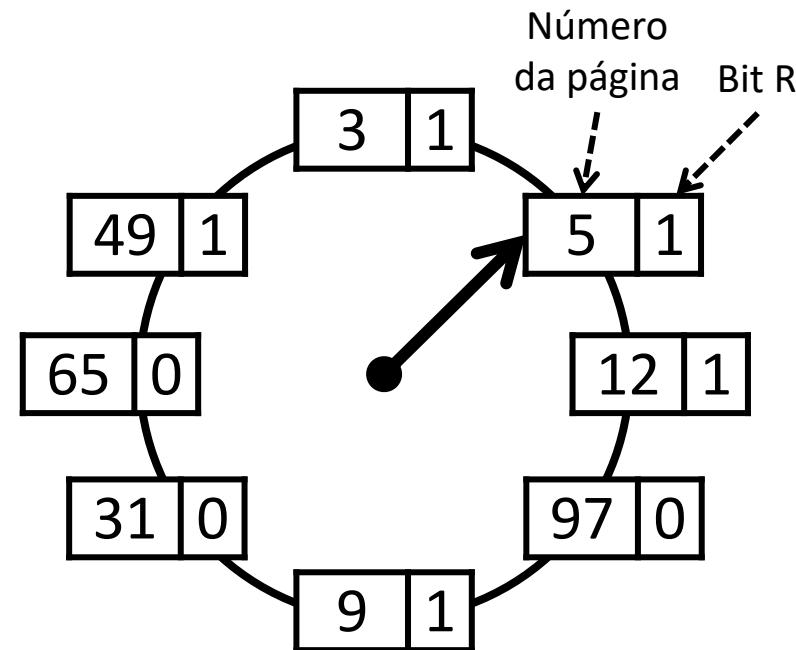
- Organiza logicamente as páginas do processo que estão atualmente na memória em uma **lista circular**
- Um **ponteiro** aponta para a página mais antiga (**cabeça da lista**)
 - Páginas são **inseridas no final da lista**



Clock

- Organiza logicamente as páginas do processo que estão atualmente na memória em uma **lista circular**
- Um **ponteiro** aponta para a página mais antiga (**cabeça da lista**)
 - Páginas são **inseridas no final da lista**

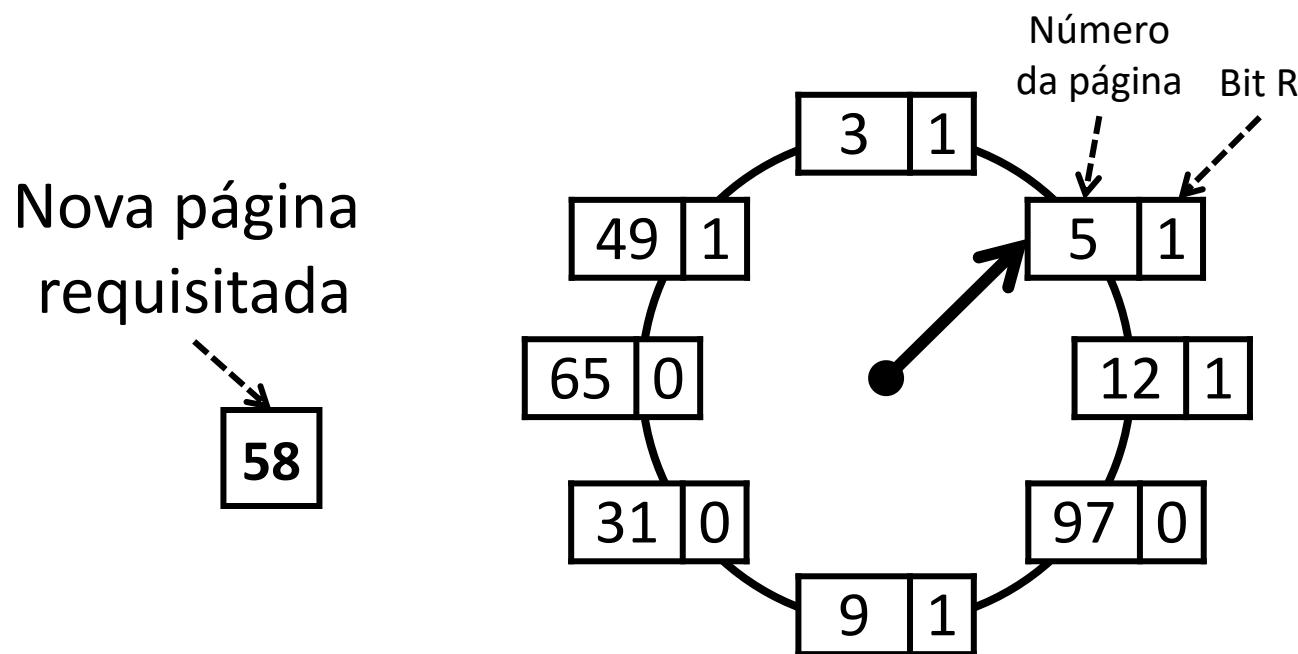
**Após várias
requisições**



Clock

■ Substituição de página

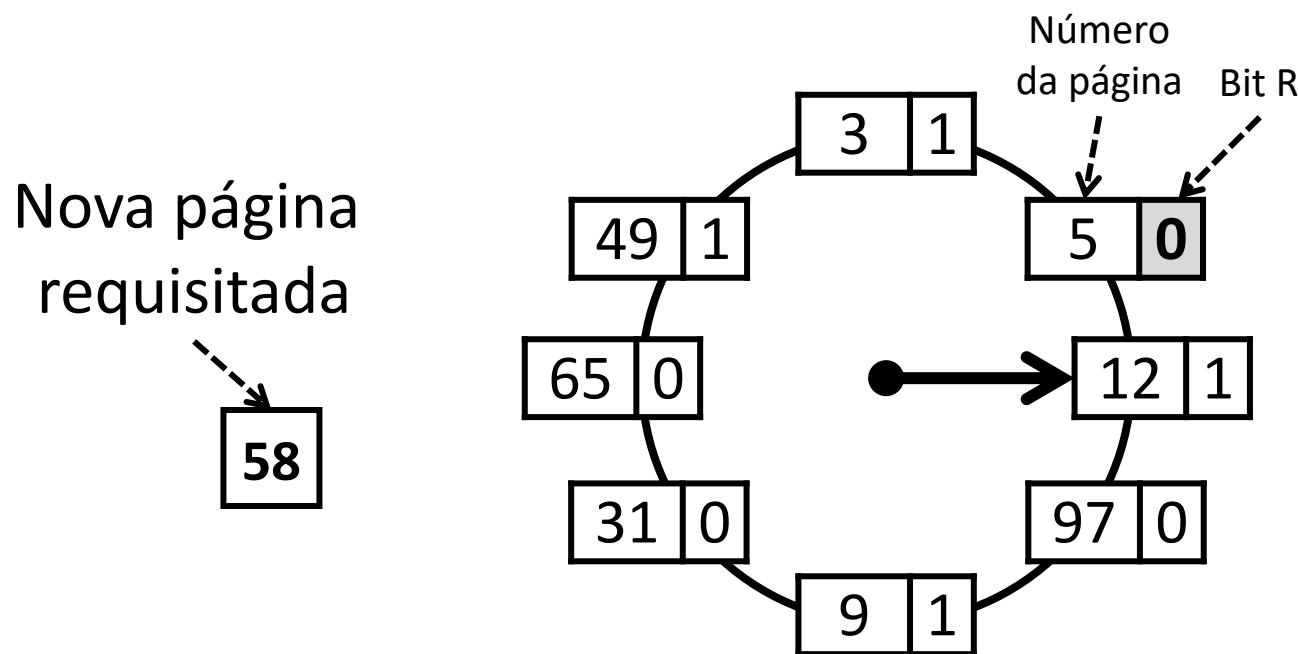
- Se $R = 1$, coloca $R = 0$ e avança o ponteiro
- Se $R = 0$, remove a página, insere a nova página em seu lugar e avança o ponteiro



Clock

■ Substituição de página

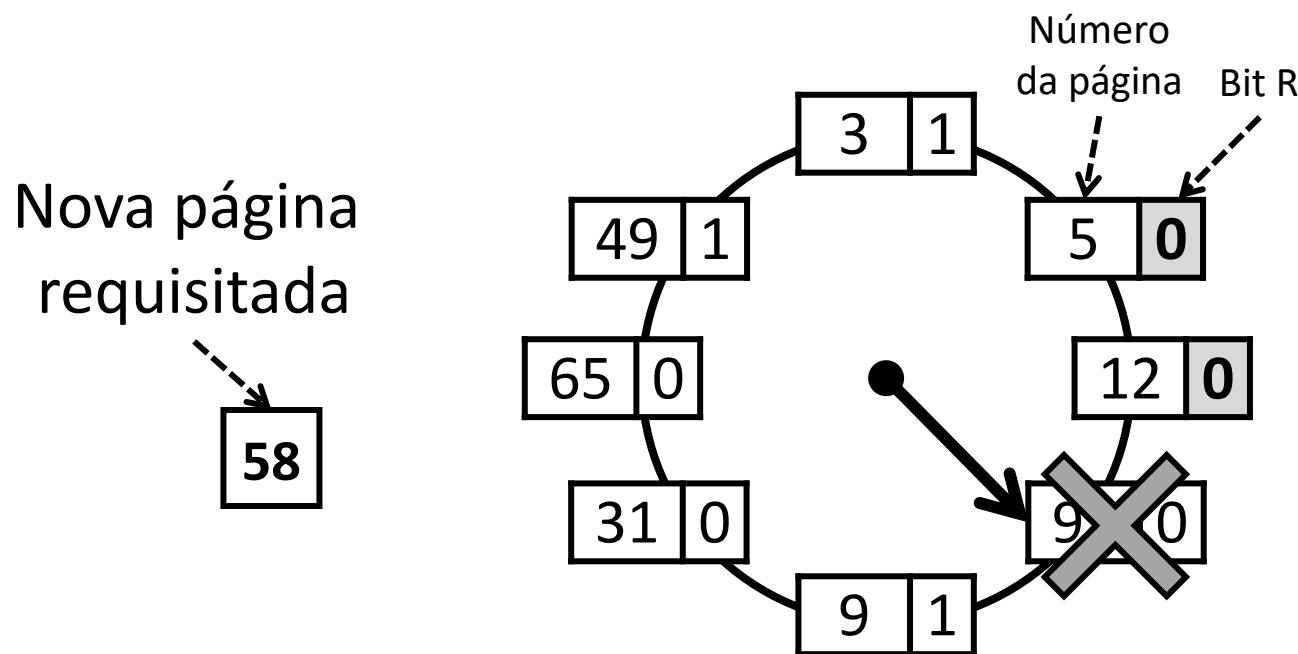
- Se $R = 1$, coloca $R = 0$ e avança o ponteiro
- Se $R = 0$, remove a página, insere a nova página em seu lugar e avança o ponteiro



Clock

■ Substituição de página

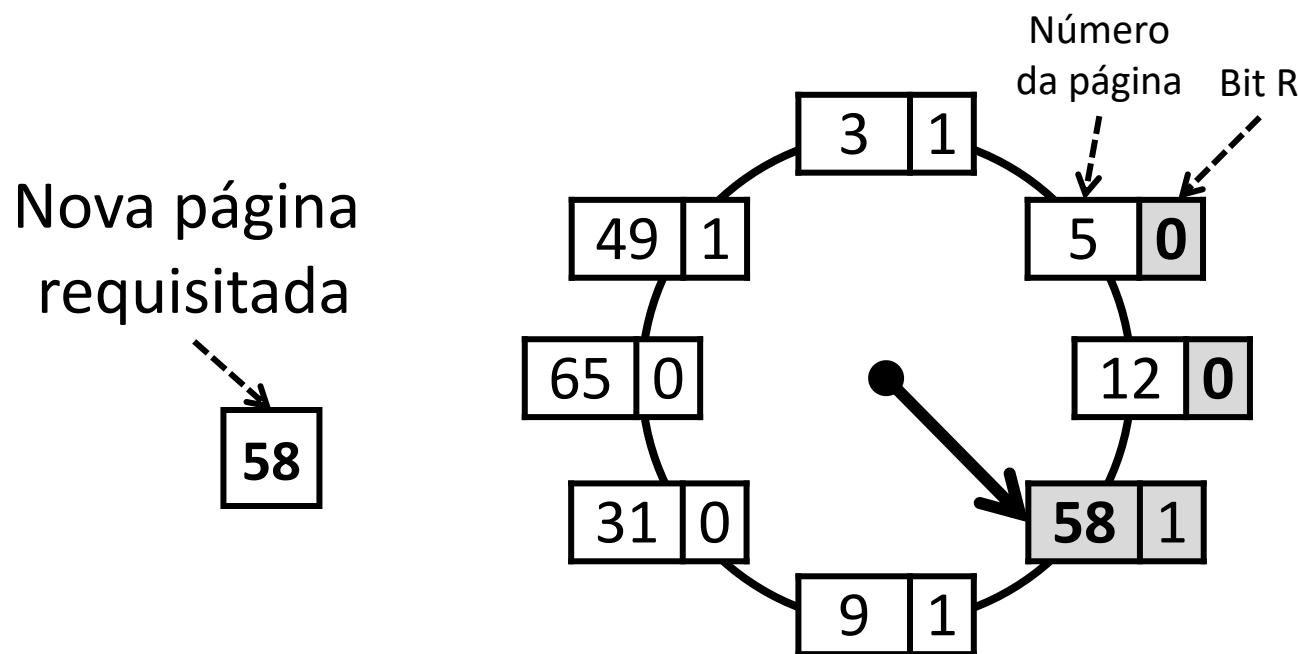
- Se $R = 1$, coloca $R = 0$ e avança o ponteiro
- Se $R = 0$, remove a página, insere a nova página em seu lugar e avança o ponteiro



Clock

■ Substituição de página

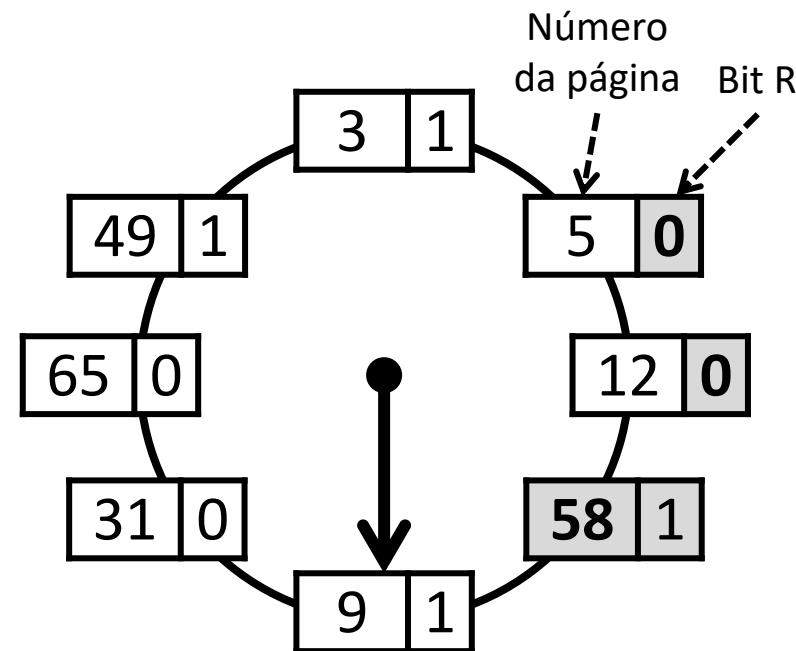
- Se $R = 1$, coloca $R = 0$ e avança o ponteiro
- Se $R = 0$, remove a página, insere a nova página em seu lugar e avança o ponteiro



Clock

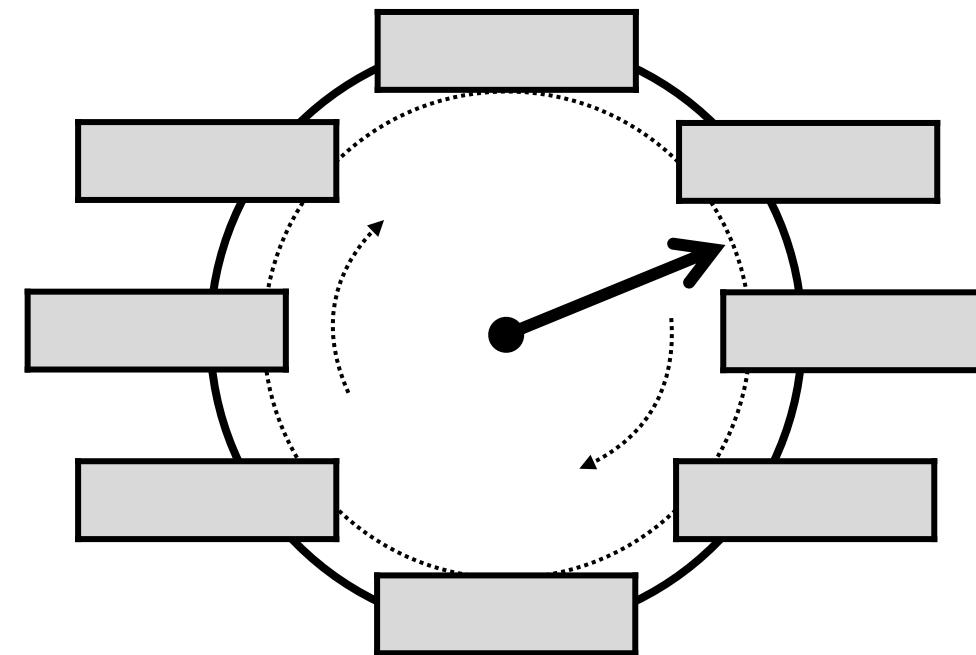
■ Substituição de página

- Se $R = 1$, coloca $R = 0$ e avança o ponteiro
- Se $R = 0$, remove a página, insere a nova página em seu lugar e avança o ponteiro



O que acontece se todas as páginas tiverem $R = 1$?

- Isso significa que todas as páginas estão sendo acessadas frequentemente
- Nesse caso, o Clock apresenta o **mesmo comportamento** do algoritmo FIFO



Vantagens do algoritmo Clock

- Páginas referenciadas frequentemente terão $R = 1$
- Páginas com $R = 1$ ganham uma **segunda chance**, não sendo eliminadas diretamente
- Evita remover páginas referenciadas **frequentemente**

4

Not Recently Used (NRU)

- Periodicamente (e.g., a cada x interrupções de relógio), coloca $R = 0$ em todas as páginas do processo que estão na memória
- **Substituição de página**
 - Separa as páginas em 4 classes com base nos bits R e M
 - Remove aleatoriamente uma página da classe de ordem mais baixa

| Classe | R | M |
|--------|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |

NRU: exemplo com 4 molduras de página

Moldura de página

| | R | M | A/P |
|---|----|---|-----|
| 7 | 10 | 0 | 0 |
| 6 | 11 | 0 | 0 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 00 | 0 | 0 |
| 2 | 00 | 0 | 0 |
| 1 | 00 | 0 | 1 |
| 0 | 01 | 1 | 1 |

Página

Tabela de páginas

NRU: exemplo com 4 molduras de página

Moldura de página

| | R | M | A/P |
|---|----|---|-----|
| 7 | 10 | 0 | 0 |
| 6 | 11 | 0 | 0 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 00 | 0 | 0 |
| 2 | 00 | 0 | 0 |
| 1 | 00 | 0 | 1 |
| 0 | 01 | 1 | 1 |

Tabela de páginas

Página

Página requisitada: 3

| Classe | Páginas |
|--------|---------|
| 0 | 6, 7 |
| 1 | 1 |
| 2 | — |
| 3 | 0 |

Página escolhida: 6

NRU: exemplo com 4 molduras de página

Moldura de página

| | R | M | A/P |
|---|----|---|-----|
| 7 | 10 | 0 | 0 |
| 6 | 11 | 0 | 0 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 00 | 0 | 0 |
| 2 | 00 | 0 | 0 |
| 1 | 00 | 0 | 1 |
| 0 | 01 | 1 | 1 |

Tabela de páginas

Página

Página requisitada: 3

| Classe | Páginas |
|--------|---------|
| 0 | 6, 7 |
| 1 | 1 |
| 2 | — |
| 3 | 0 |

Página escolhida: 6

NRU: exemplo com 4 molduras de página

Moldura de página

| | R | M | A/P |
|---|----|---|-----|
| 7 | 10 | 0 | 0 |
| 6 | 11 | 0 | 0 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 11 | 1 | 0 |
| 2 | 00 | 0 | 0 |
| 1 | 00 | 0 | 1 |
| 0 | 01 | 1 | 1 |

Tabela de páginas

Página

Página requisitada: 3

| Classe | Páginas |
|--------|---------|
| 0 | 6, 7 |
| 1 | 1 |
| 2 | — |
| 3 | 0 |

Página escolhida: 6

5

Least Recently Used (LRU)

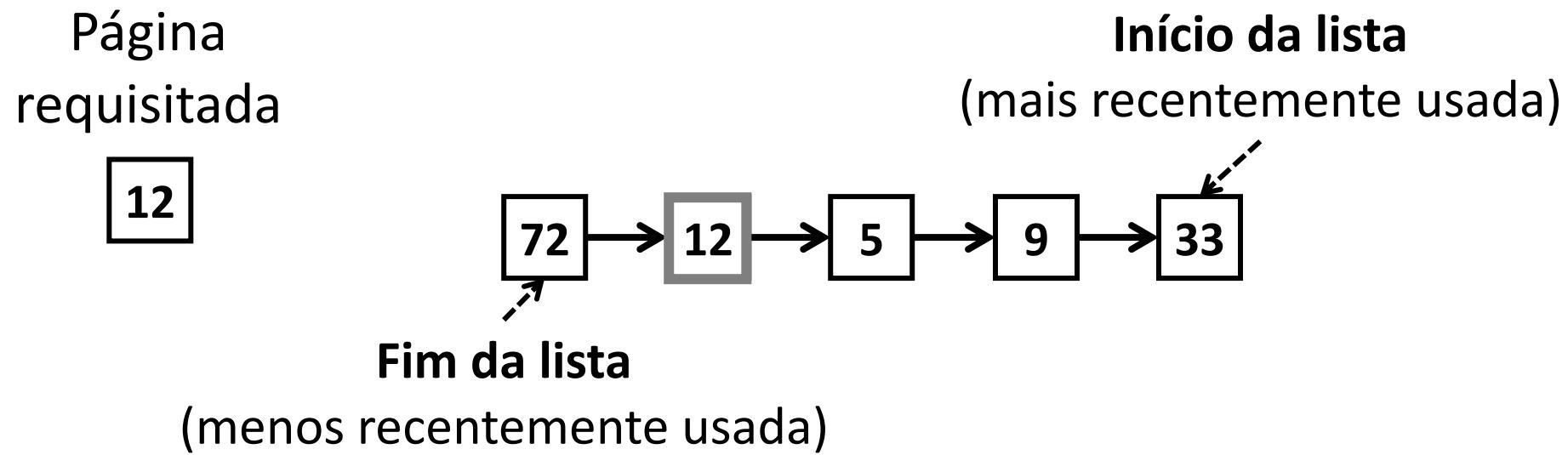
- **Baseado nas seguintes observações:**

- Páginas muito utilizadas nas últimas instruções provavelmente serão muito utilizadas nas próximas instruções
- Páginas não utilizadas por um longo período de tempo provavelmente permanecerão inutilizadas por muito tempo

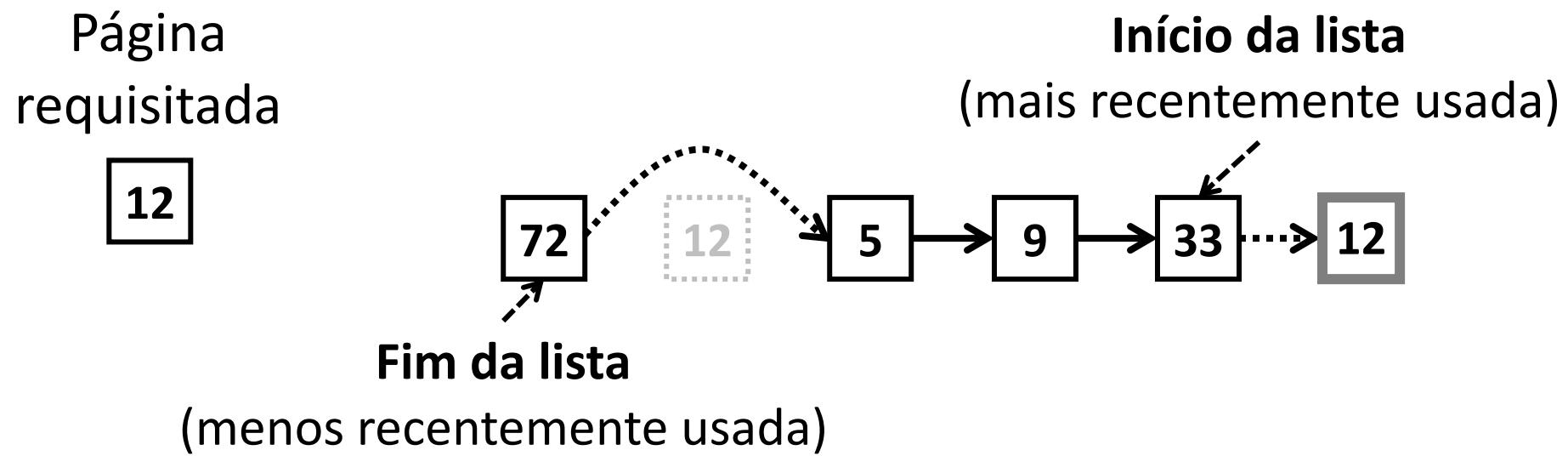
- **Portanto, a ideia básica do LRU é:**

- Substituir a página não utilizada pelo período de tempo mais longo

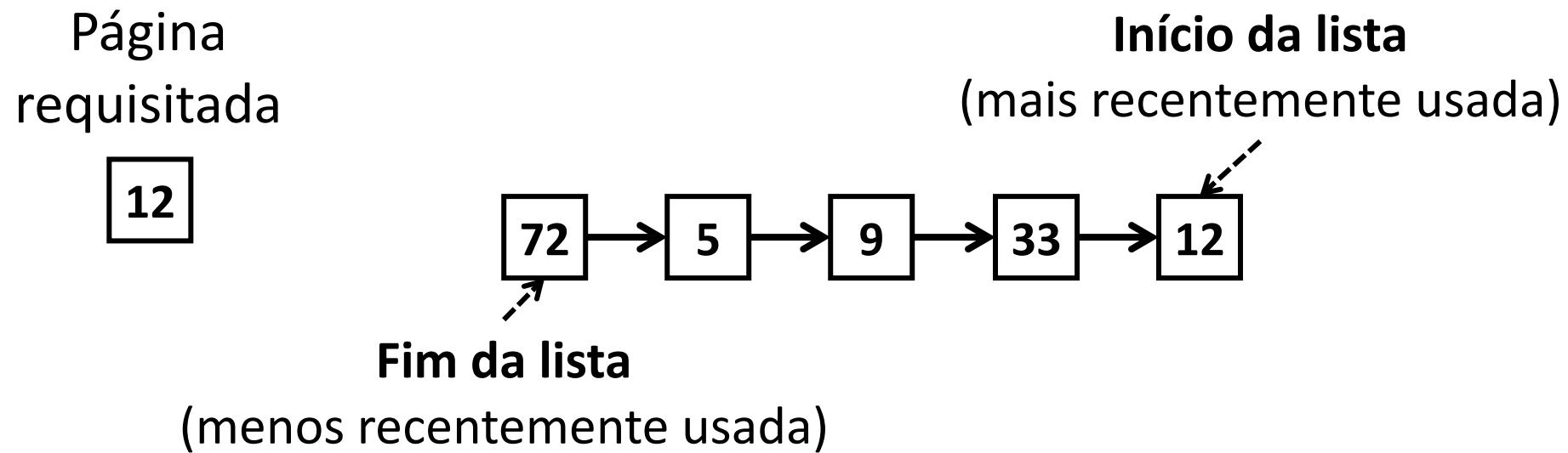
- Mantem as páginas em uma lista ordenada pelo instante de último uso
 - Mais recente (**início da lista**) e menos recente (**fim da lista**)
- A lista é atualizada em cada referência à memória



- Mantem as páginas em uma lista ordenada pelo instante de último uso
 - Mais recente (**início da lista**) e menos recente (**fim da lista**)
- A lista é atualizada em cada referência à memória



- Mantem as páginas em uma lista ordenada pelo instante de último uso
 - Mais recente (**início da lista**) e menos recente (**fim da lista**)
- A lista é atualizada em cada referência à memória

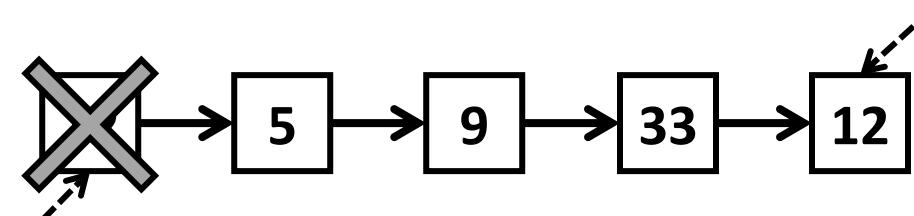


■ Substituição de página

- Remove a página que está no **fim da lista**
- Insere a nova página no **início da lista**

Nova página
requisitada

15



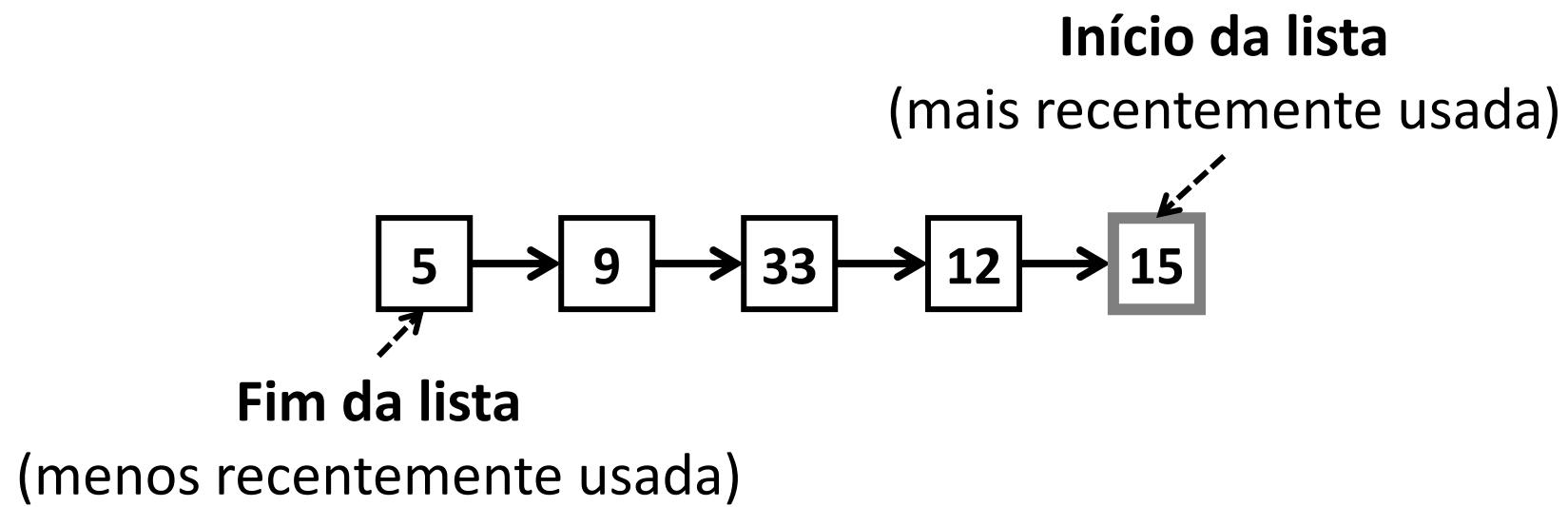
Início da lista
(mais recentemente usada)

Fim da lista

(menos recentemente usada)

■ Substituição de página

- Remove a página que está no **fim da lista**
- Insere a nova página no **início da lista**



- Embora teoricamente realizável, o LRU apresenta um grande problema
 - Atualizar a lista em cada referência à memória ➔ péssimo desempenho
- Simulação do LRU em software
 - Not Frequently Used (NFU)
 - Aging

6

Not Frequently Used (NFU)

- Assoca um **contador** à cada página
- Periodicamente (e.g., a cada x interrupções de relógio), realiza as seguintes operações nas páginas do processo que estão atualmente em memória:
 - Soma o valor do bit R ao **contador associado à página**
 - Coloca R = 0
- Substituição de página
 - Substitui a página com menor valor de contador
 - Escolhe aleatoriamente em caso de empate

NFU: exemplo com 4 molduras de página

Moldura
de página

R A/P Contador

| | | | | |
|---|----|---|---|---|
| 7 | 00 | 0 | 0 | 0 |
| 6 | 00 | 0 | 0 | 0 |
| 5 | 00 | 0 | 0 | 0 |
| 4 | 00 | 0 | 0 | 0 |
| 3 | 00 | 0 | 0 | 0 |
| 2 | 00 | 0 | 0 | 0 |
| 1 | 00 | 0 | 0 | 0 |
| 0 | 00 | 0 | 0 | 0 |

Tabela de páginas

Estado inicial

NFU: exemplo com 4 molduras de página

Moldura
de página

R A/P Contador



| | R | A/P | Contador |
|---|----|-----|----------|
| 7 | 00 | 0 | 0 |
| 6 | 10 | 1 | 1 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 00 | 1 | 1 |
| 2 | 11 | 1 | 1 |
| 1 | 00 | 0 | 0 |
| 0 | 01 | 1 | 1 |

Tabela de páginas

Período: 1

Páginas referenciadas

0 2 3 6

NFU: exemplo com 4 molduras de página

Moldura
de página

R A/P Contador

| | R | A/P | Contador |
|---|----|-----|----------|
| 7 | 00 | 0 | 0 |
| 6 | 10 | 1 | 1 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 00 | 1 | 1 |
| 2 | 11 | 1 | 1 |
| 1 | 00 | 0 | 0 |
| 0 | 01 | 1 | 1 |

Tabela de páginas

Período: 1

Páginas referenciadas



NFU: exemplo com 4 molduras de página

Moldura
de página

| | R | A/P | Contador |
|---|----|-----|----------|
| 7 | 00 | 0 | 0 |
| 6 | 10 | 0 | 1 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 00 | 0 | 1 |
| 2 | 11 | 0 | 1 |
| 1 | 00 | 0 | 0 |
| 0 | 01 | 0 | 1 |

Tabela de páginas

Período: 1

NFU: exemplo com 4 molduras de página

Moldura
de página

R A/P Contador

| | | | | |
|---|----|---|---|---|
| 7 | 00 | 0 | 0 | 0 |
| 6 | 10 | 0 | 1 | 1 |
| 5 | 00 | 0 | 0 | 0 |
| 4 | 00 | 0 | 0 | 0 |
| 3 | 00 | 1 | 1 | 1 |
| 2 | 11 | 0 | 1 | 1 |
| 1 | 00 | 0 | 0 | 0 |
| 0 | 01 | 1 | 1 | 1 |

Tabela de páginas

Período: 2

Páginas referenciadas

0 3

NFU: exemplo com 4 molduras de página

Moldura
de página

R A/P Contador

| | R | A/P | Contador |
|---|----|-----|----------|
| 7 | 00 | 0 | 0 |
| 6 | 10 | 0 | 1 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 00 | 1 | 1 |
| 2 | 11 | 0 | 1 |
| 1 | 00 | 0 | 0 |
| 0 | 01 | 1 | 2 |

Tabela de páginas

Período: 2

Páginas referenciadas

0 3

NFU: exemplo com 4 molduras de página

Moldura
de página



| | R | A/P | Contador |
|---|----|-----|----------|
| 7 | 00 | 0 | 0 |
| 6 | 10 | 0 | 1 |
| 5 | 00 | 0 | 0 |
| 4 | 00 | 0 | 0 |
| 3 | 00 | 0 | 1 |
| 2 | 11 | 0 | 1 |
| 1 | 00 | 0 | 0 |
| 0 | 01 | 0 | 2 |

Tabela de páginas

Período: 2

NFU: exemplo com 4 molduras de página

Moldura
de página

R A/P Contador

| | | | | |
|---|----|---|---|----|
| 7 | 00 | 0 | 0 | 0 |
| 6 | 10 | 1 | 1 | 10 |
| 5 | 00 | 0 | 0 | 0 |
| 4 | 00 | 0 | 0 | 0 |
| 3 | 00 | 1 | 1 | 5 |
| 2 | 11 | 0 | 1 | 3 |
| 1 | 00 | 0 | 0 | 0 |
| 0 | 01 | 0 | 1 | 8 |

Tabela de páginas

Período: 10

Páginas referenciada

5

Página a ser removida

NFU: exemplo com 4 molduras de página

Moldura
de página

| | R | A/P | Contador | |
|---|----|-----|----------|----|
| 7 | 00 | 0 | 0 | 0 |
| 6 | 10 | 1 | 1 | 10 |
| 5 | 11 | 1 | 1 | 1 |
| 4 | 00 | 0 | 0 | 0 |
| 3 | 00 | 1 | 1 | 5 |
| 2 | 11 | 0 | 0 | 3 |
| 1 | 00 | 0 | 0 | 0 |
| 0 | 01 | 0 | 1 | 8 |

Tabela de páginas

Período: 10

Informações atualizadas

- **Problema: o NFU “nunca se esquece do passado”**
 - Páginas muito referenciadas durante um longo período terão **valores muito altos em seus contadores**
 - Páginas antigas não serão eliminadas durante muito tempo
 - NFU substituirá páginas sendo usadas recentemente

7

Aging

Aging

- Associa um **contador** à cada página
- Periodicamente (e.g., a cada x interrupções de relógio), realiza as seguintes operações nas páginas do processo que estão atualmente em memória:
 - Desloca o **contador** em 1 bit à direita (contador $\gg 1$)
 - Coloca o valor do bit R no **bit mais significativo do contador**
 - Coloca R = 0
- **Substituição de página**
 - Substitui a página com menor valor de contador
 - Escolhe aleatoriamente em caso de empate

Aging: exemplo com 4 molduras de página

| Período 0 | Período 1 | Período 2 | Período 3 | Período 4 | Período 5 |
|-----------------------|-----------|-----------|-----------|-----------|-----------|
| Página -----> 0 1 2 3 | 0 1 2 3 | 0 1 2 3 | 0 1 2 3 | 0 1 2 3 | 0 1 2 3 |
| Bits R -----> 1 0 0 0 | 1 0 1 0 | 0 0 1 1 | 1 1 1 0 | 1 0 1 1 | 1 0 1 0 |
| Página 0 10000000 | 11000000 | 01100000 | 10110000 | 11011000 | 11101100 |
| Página 1 00000000 | 00000000 | 00000000 | 10000000 | 01000000 | 00100000 |
| Página 2 00000000 | 10000000 | 11000000 | 11100000 | 11110000 | 11111000 |
| Página 3 00000000 | 00000000 | 10000000 | 01000000 | 10100000 | 01010000 |

Contador ----->

- **Vantagens**

- Simula muito bem o LRU e com custo muito menor
- Melhor que o NFU, pois penaliza páginas não acessadas recentemente

- **Divergências com relação ao LRU**

- Não é possível saber a ordem em que as páginas foram referenciadas dentro de um mesmo período
- Contadores tem tamanho limitado, logo há perda de informação passada

Working Set Clock (wsclock)

- Na forma mais básica de paginação, processos são inicializados sem nenhuma página na memória
 - Assim que a CPU tenta buscar a primeira instrução há uma **falta de página**
 - Em seguida, surgem **faltas de página** relativas à **pilha** e aos **dados** do programa
 - Esta técnica é denominada **paginação por demanda** (*demand paging*)
- **Localidade de referência**
 - Processos possuem **fases de execução**
 - Referenciam uma **fração** de suas páginas durante uma **determinada fase**

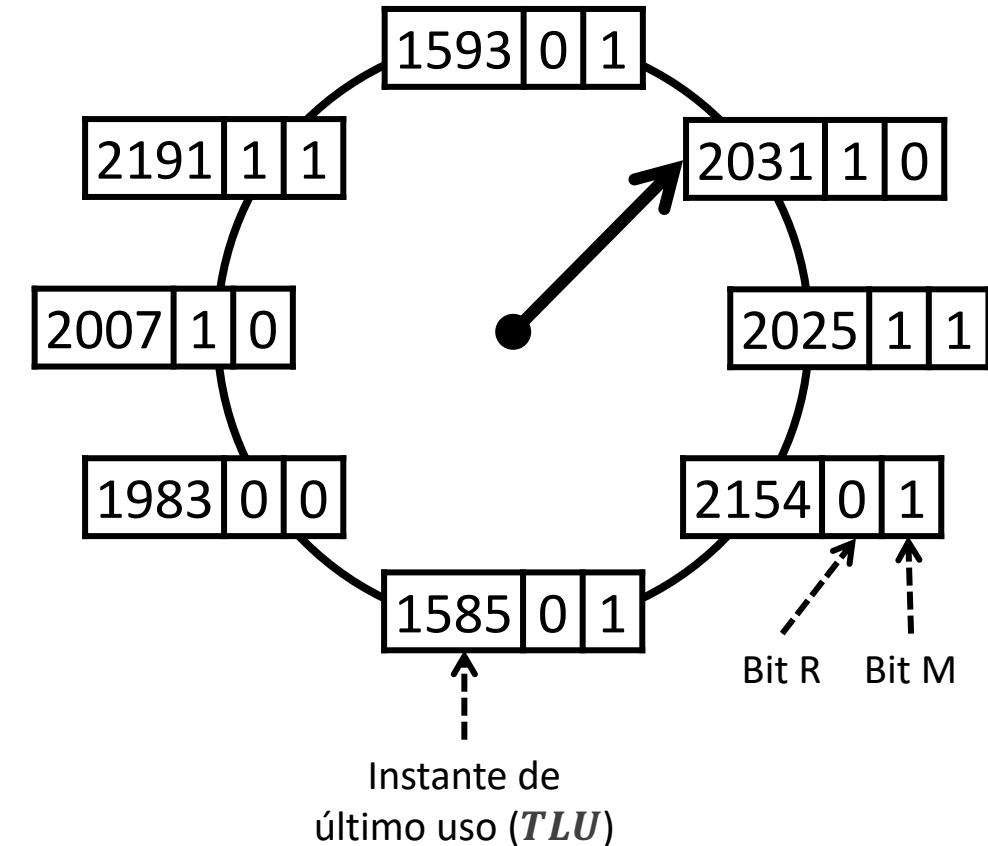
- **Conjunto de trabalho (*working set*)**
 - Conjunto de páginas que o processo está usando **atualmente**
 - Se **todo o conjunto** estiver em memória ➔ **poucas faltas de página**
 - Se **não há memória** suficiente ➔ **perda de desempenho**
 - ***Trashing***: quando faltas de página ocorrem **frequentemente e continuamente**
- **Aproximação para determinar o conjunto de trabalho**
 - Tempo de CPU usado pelo processo (**tempo virtual atual**)
 - **Conjunto de trabalho**: conjunto de páginas usadas pelo processo durante as últimas \mathcal{T} unidades de tempo de seu **tempo virtual**

WSClock

- Se $R = 1$:
 - Coloca $R = 0$, copia o CVT para o TLU e avança o ponteiro

Tempo virtual atual (CVT) 2204

$\mathcal{T} = 200$

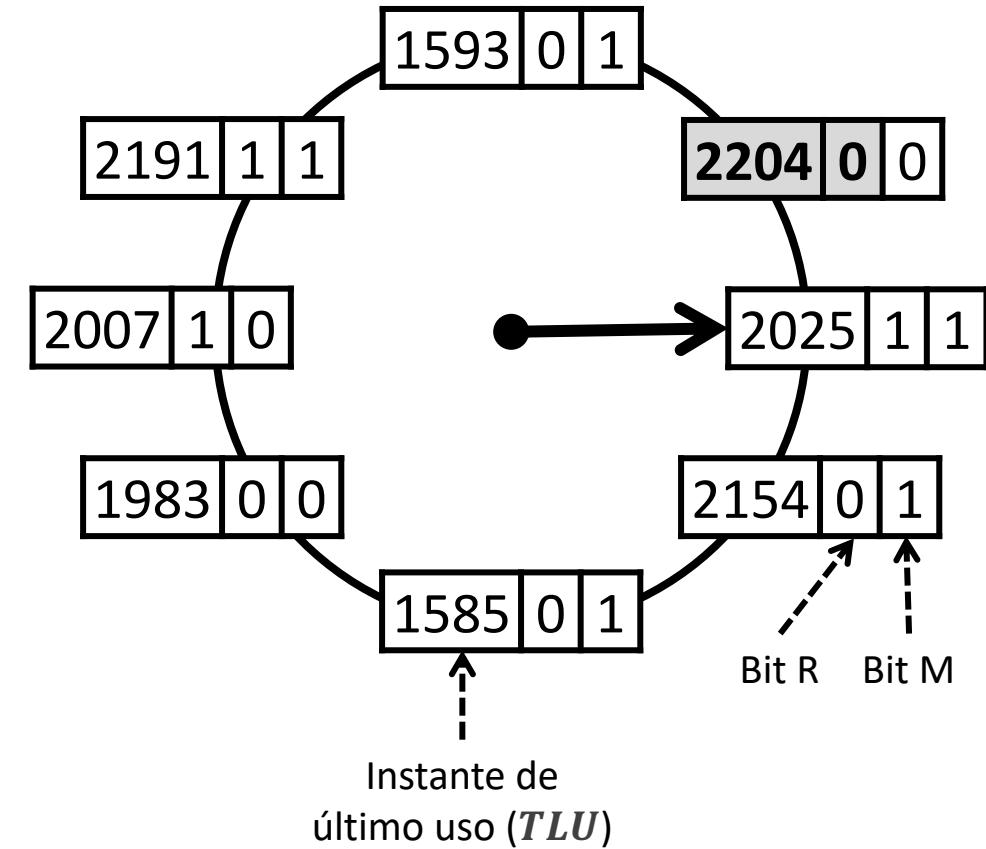


WSClock

- Se $R = 1$:
 - Coloca $R = 0$, copia o CVT para o TLU e avança o ponteiro

Tempo virtual atual (CVT) **2204**

$\mathcal{T} = 200$



WSClock

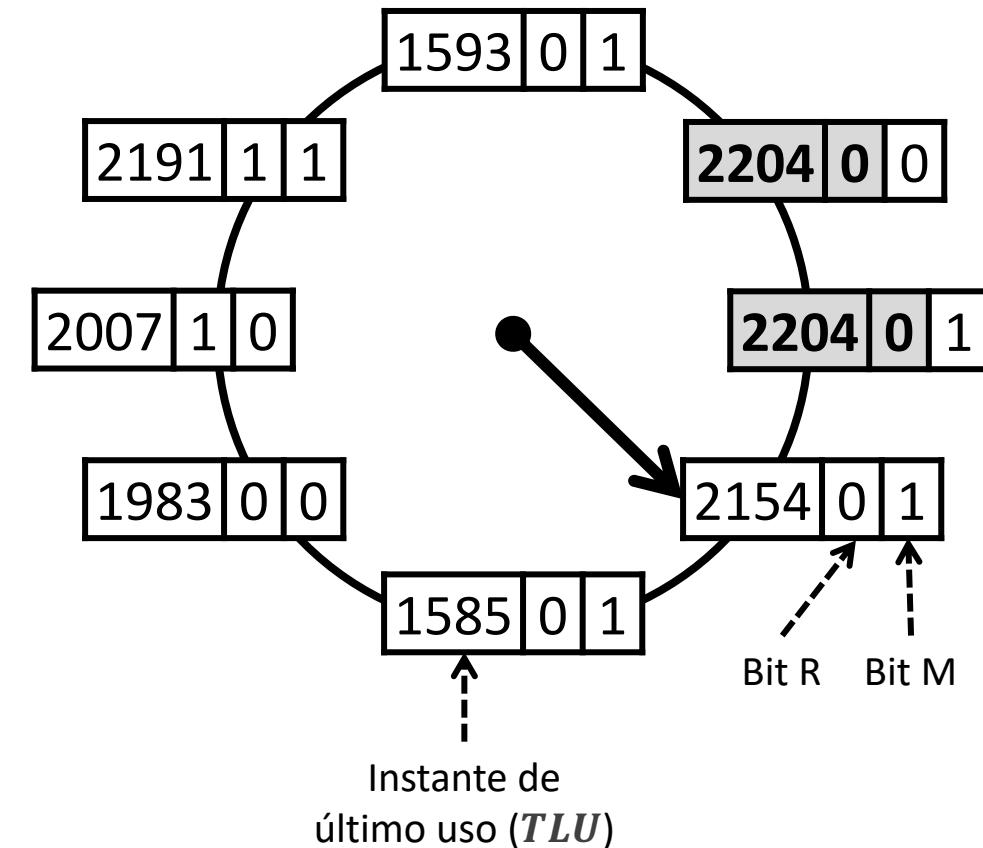
- Se $R = 1$:
 - Coloca $R = 0$, copia o CVT para o TLU e avança o ponteiro
- Se $R = 0$: calcula $idade = CVT - TLU$
 - Se $idade \leq \mathcal{T}$: guarda a maior idade e avança o ponteiro
 - Se $idade > \mathcal{T}$ e $M = 0$: substitui a página
 - Se $idade > \mathcal{T}$ e $M = 1$: escalona a escrita no disco e avança o ponteiro

$$idade = CVT - TLU$$

$$idade = 2204 - 2154 = 50$$

Tempo virtual atual (CVT) 2204

$\mathcal{T} = 200$



WSClock

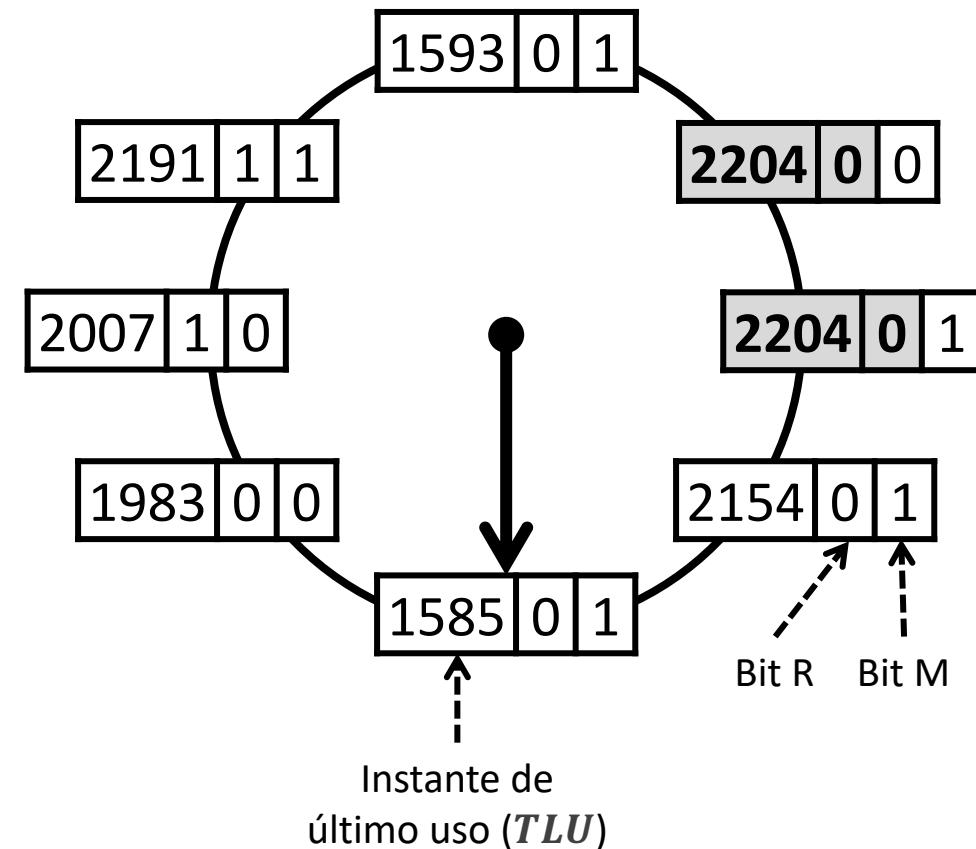
- Se $R = 1$:
 - Coloca $R = 0$, copia o CVT para o TLU e avança o ponteiro
- Se $R = 0$: calcula $idade = CVT - TLU$
 - Se $idade \leq \mathcal{T}$: guarda a maior idade e avança o ponteiro
 - Se $idade > \mathcal{T}$ e $M = 0$: substitui a página
 - Se $idade > \mathcal{T}$ e $M = 1$: escalona a escrita no disco e avança o ponteiro

$$idade = CVT - TLU$$

$$idade = 2204 - 1585 = 619$$

Tempo virtual atual (CVT) 2204

$\mathcal{T} = 200$



WSClock

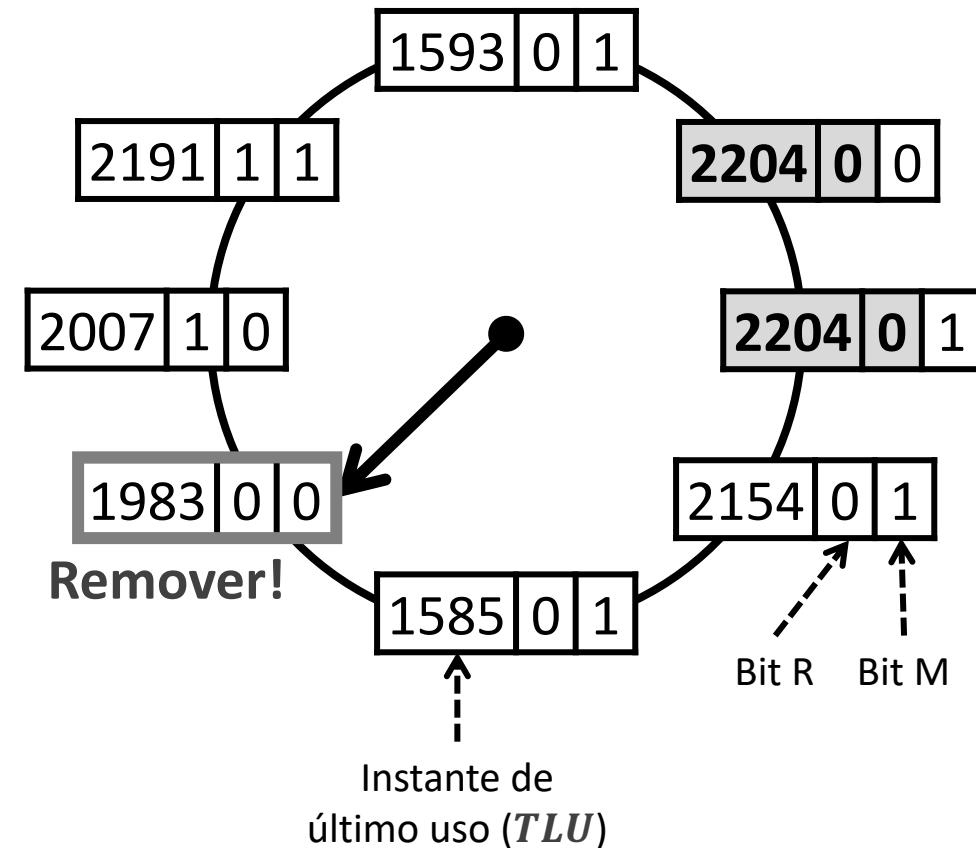
- Se $R = 1$:
 - Coloca $R = 0$, copia o CVT para o TLU e avança o ponteiro
- Se $R = 0$: calcula $idade = CVT - TLU$
 - Se $idade \leq \mathcal{T}$: guarda a maior idade e avança o ponteiro
 - Se $idade > \mathcal{T}$ e $M = 0$: substitui a página
 - Se $idade > \mathcal{T}$ e $M = 1$: escalona a escrita no disco e avança o ponteiro

$$idade = CVT - TLU$$

$$idade = 2204 - 1983 = 221$$

Tempo virtual atual (CVT) 2204

$\mathcal{T} = 200$

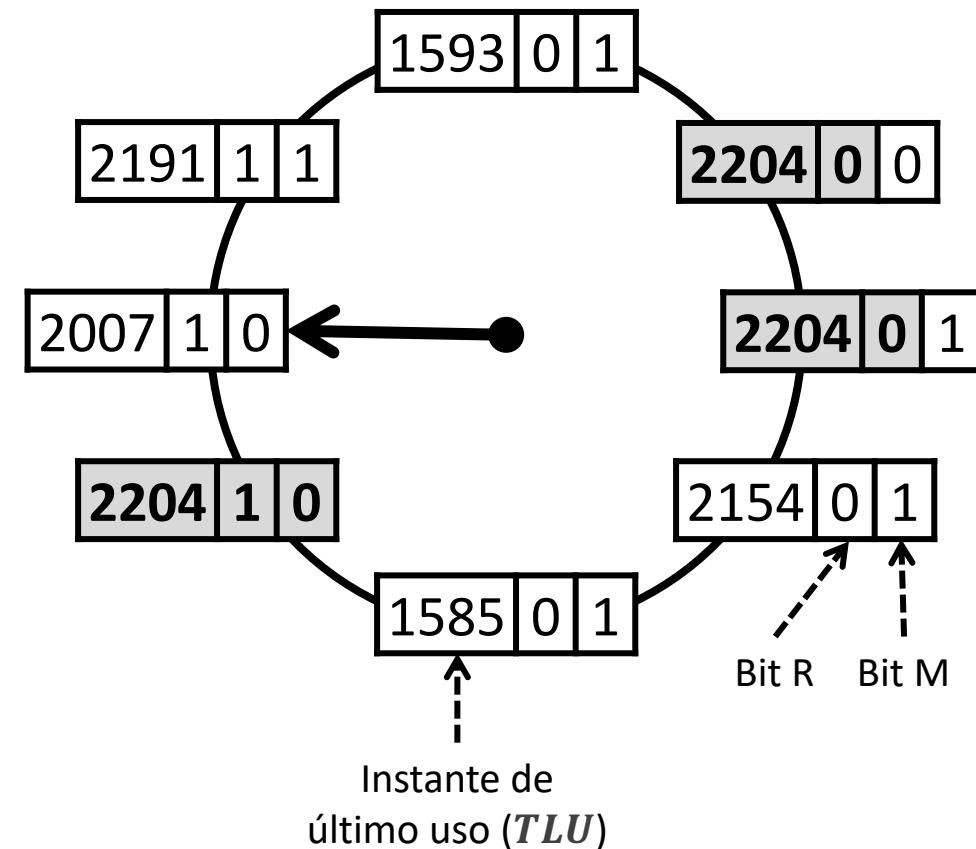


WSClock

- Se $R = 1$:
 - Coloca $R = 0$, copia o CVT para o TLU e avança o ponteiro
- Se $R = 0$: calcula $idade = CVT - TLU$
 - Se $idade \leq \mathcal{T}$: guarda a maior idade e avança o ponteiro
 - Se $idade > \mathcal{T}$ e $M = 0$: substitui a página
 - Se $idade > \mathcal{T}$ e $M = 1$: escalona a escrita no disco e avança o ponteiro

Tempo virtual atual (CVT) **2204**

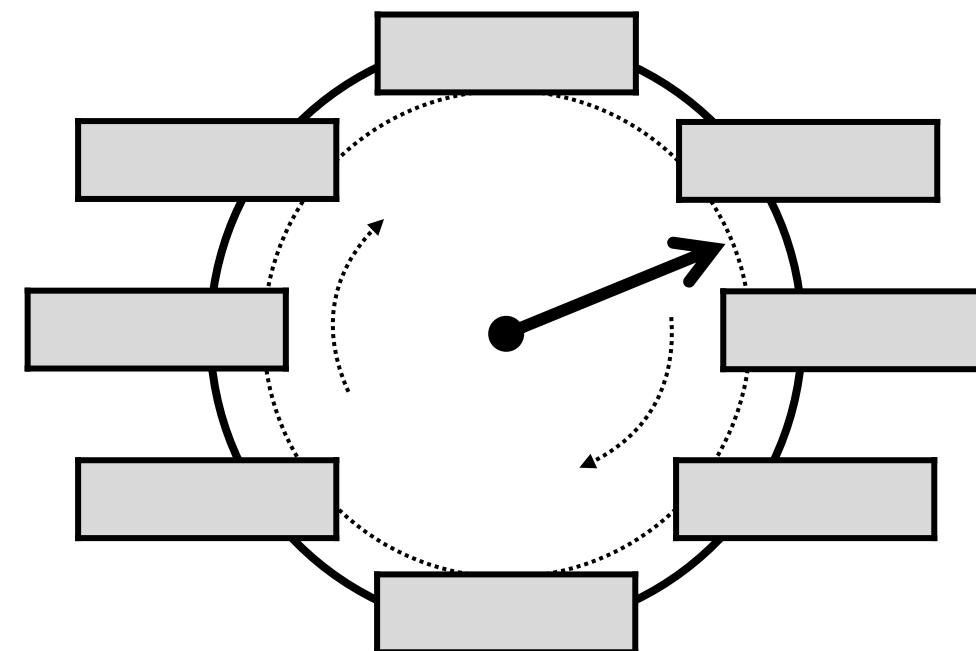
$\mathcal{T} = 200$



O que acontece se o ponteiro der uma volta completa?

Dois casos possíveis:

- **Se alguma escrita escalonada**
 - É bem provável que alguma escrita tenha sido **finalizada**
 - Substitui página **velha** com $M = 0$
- **Se nenhuma escrita escalonada**
 - Todas as páginas fazem parte do conjunto de trabalho!
 - Substitui a **próxima** página com $M = 0$
 - Se não há página com $M = 0$, substitui a **página com maior idade**



Conclusões

Conclusões

| Algoritmo | Característica |
|----------------|---|
| Ótimo | Não implementável na prática, mas útil para <i>benchmarking</i> |
| FIFO | Pode substituir páginas importantes |
| Clock | Desempenho bom, mas não diferencia páginas modificadas |
| LRU | Excelente, mas muito custoso |
| NRU | Aproximação extremamente rudimentar do LRU |
| NFU | Aproximação rudimentar do LRU |
| Aging | Algoritmo eficiente que aproxima muito bem o LRU |
| WSClock | Algoritmo bom e eficiente |

!

Obrigado pela atenção!



Dúvidas? Entre em contato:

- marcio.castro@ufsc.br
- www.marciocastro.com



Distributed Systems Research Lab
www.lapesd.inf.ufsc.br