

The background of the slide features a close-up of a white Apple logo on the left and a green Android robot on the right, both resting on a wooden surface. A green laser beam is visible in the upper left corner. A horizontal bar with a teal segment on the left and a green segment on the right spans the top of the slide.

SISTEMAS OPERACIONAIS

Gerência de Memória
Algoritmos de Substituição de Páginas

Introdução

- Quando ocorre um Page Fault, o S.O. deve escolher que página remover para abrir espaço em memória.
- Se a página foi alterada (bit Modified setado) é preciso salvá-la em disco. Senão foi, basta sobrescrevê-la.
- É melhor não escolher para remoção uma página que é usada frequentemente, pois ela pode ter que voltar para a memória logo.

Algoritmos de substituição

- ❑ Ótimo;
- ❑ NRU;
- ❑ FIFO;
- ❑ Segunda Chance;
- ❑ Relógio;
- ❑ LRU;
 - NFU
 - Aging

Algoritmo Ótimo

- ❑ Retira da memória a página que tem menos chance de ser referenciada;
- ❑ Praticamente impossível de se saber;
 - Marcar p/ cada página, quantas instruções faltam p/ que ela seja referenciada
- ❑ Impraticável;
- ❑ Usado em simulações para comparação com outros algoritmos;
 - Solução ótima, mas inviável!

NRU – *Not Recently Used* ⁽¹⁾

- Ou seja, algoritmo de substituição da página “não usada recentemente”
- Na maioria dos computadores com memória virtual, as entradas nas tabelas de páginas têm 2 bits de status
 - Reference bit (R) ; *Modified bit* (M)
- Algoritmo
 - Quando o processo é iniciado, os bits R e M das páginas são zerados
 - Bits são sempre alterados quando a página é referenciada/modificada
 - Periodicamente o bit R é zerado (por exemplo, a cada tique de clock)
 - Quando acontece um *Page fault*, o S.O. inspeciona todas as páginas que encontram-se na memória e as separa em categorias...

NRU – *Not Recently Used* (2)

- Páginas são classificadas
 - Classe 0: Not referenced, not modified ($R=0$, $M=0$)
 - Classe 1: Not referenced, modified ($R=0$, $M=1$) ← Isso pode ocorrer???
 - Classe 2: referenced, not modified ($R=1$, $M=0$)
 - Classe 3: referenced, modified ($R=1$, $M=1$)
- O S.O. remove uma das páginas (aleatoriamente) da classe mais baixa não vazia.
- Vantagens
 - Algoritmo fácil de entender e implementar
 - Desempenho adequado

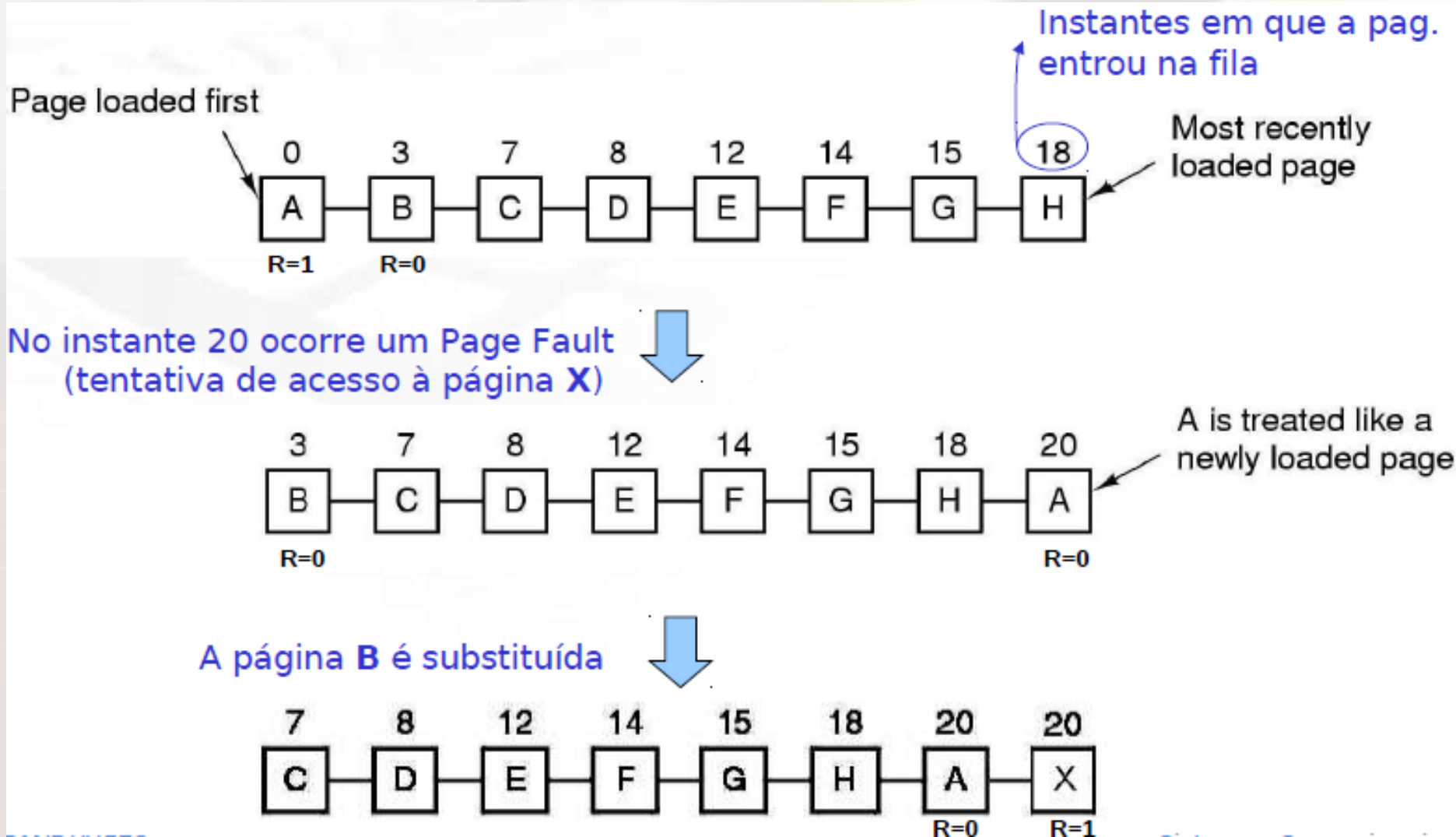
FIFO – *First In First Out*

- Página mais velha é candidata em potencial;
- Mantem-se uma lista encadeada de páginas ordenada pela chegada das páginas à memória.
- Quando ocorre um Page Fault, a página no início da lista (que é a mais antiga) é a escolhida para a troca
- Vantagem:
 - Baixo custo
- Desvantagem:
 - A página mais antiga pode ser também uma página usada muito frequentemente.
- Não empregado!

SC - Segunda Chance ⁽¹⁾

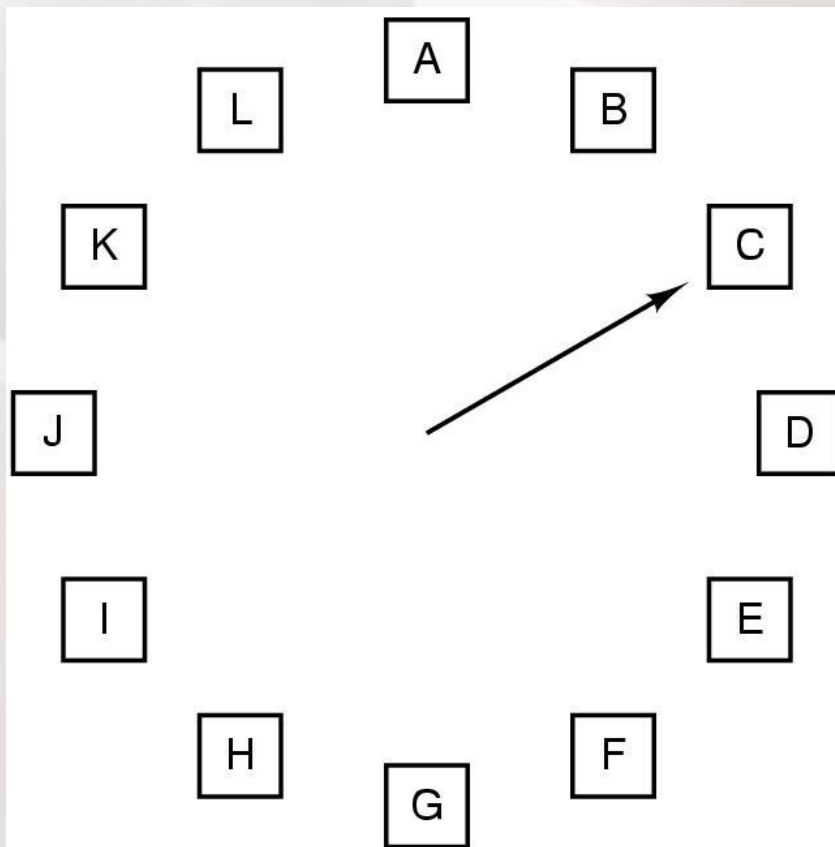
- Tenta melhorar o FIFO
- Cada página tem um bit R (referenciada)
- Antes de remover a página mais antiga (cabeça da fila), seu bit R é verificado
 - Se $R=0$, a página é substituída (a página referenciada ocupará o seu lugar na memória)
 - Se $R=1$, a página vai para fim da fila, como se houvesse sido carregada agora e seu bit é setado para 0
 - Verifica-se a página que virou “cabeça” da fila
- Se todas as páginas tiverem seu bit $R=1$, haverá uma volta completa

SC - Segunda Chance (2)



Relógio

- O ponteiro sempre aponta para a página mais antiga



When a page fault occurs, the page the hand is pointing to is inspected. The action taken depends on the R bit:

R = 0: Evict the page

R = 1: Clear R and advance hand

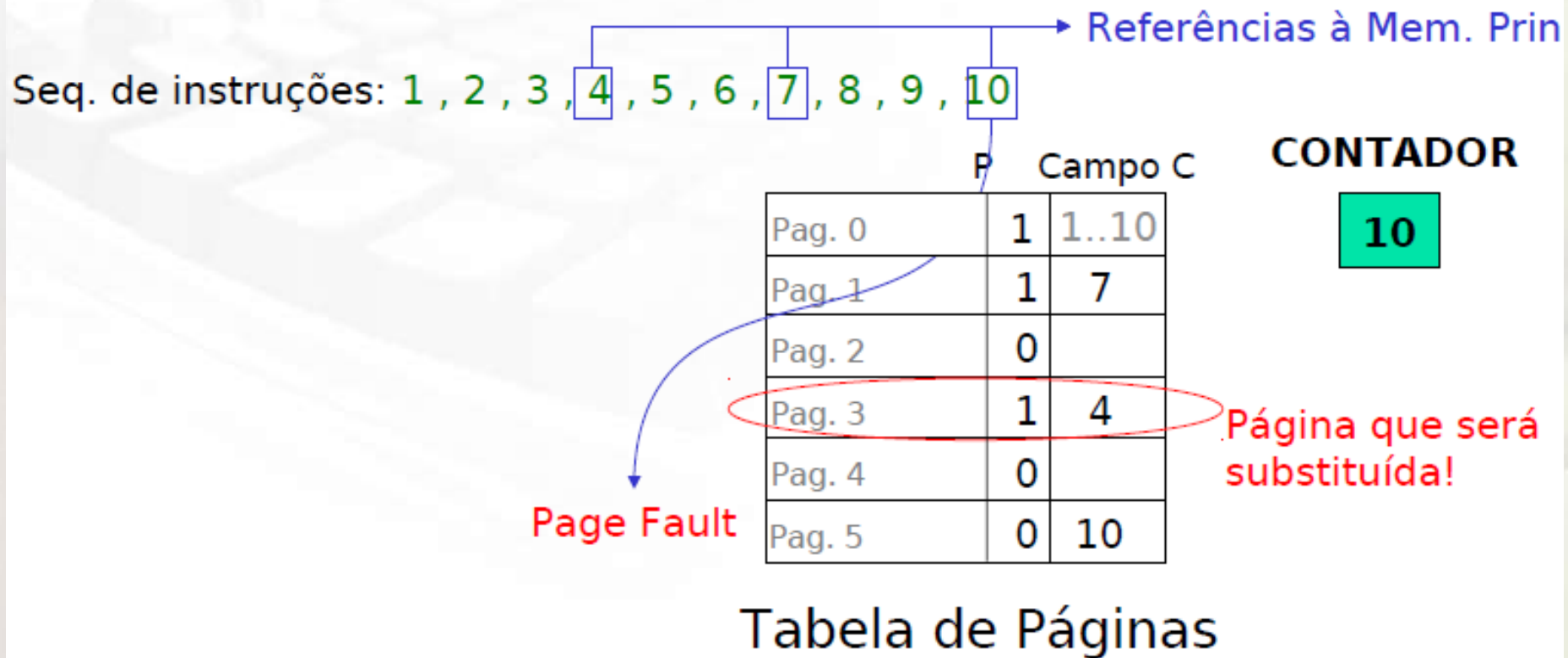
LRU – Least Recently Used ⁽¹⁾

- *Ou MRU – Menos Recentemente Usada*
- Assume que as páginas usadas recentemente voltarão a ser usadas em breve
 - Substitui páginas que estão há **mais tempo sem uso**.
- Para implementá-lo completamente, deve-se manter lista encadeada de todas as páginas que estão na memória (muito custoso!)
 - página usada mais recentemente vai para o início da lista;
 - lista é reordenada a cada referência a memória
 - quando há Page Fault, escolhe-se a última página da fila

LRU – Least Recently Used (2)

- Uma solução simples: manter uma idade para cada página.
 - Usar um contador C de 64 bits incrementado a cada instrução (em hardware)
 - Cada entrada da tabela de páginas deve ter um campo extra para armazenar o valor do contador
 - A cada referência à memória o valor corrente de C é armazenado na entrada da tabela de páginas na posição correspondente à página referenciada
 - Quando ocorre um Page Fault, a tabela de páginas é examinada, a entrada cujo campo C é de menor valor é a escolhida
 - Substitui página com o menor valor no campo do contador (maior idade)

LRU – Least Recently Used (3)



LRU – Least Recently Used (4)

- LRU usando matrizes
 - **Hardware** especial que mantém uma matriz $n \times n$, onde n é o número de molduras
 - Inicialmente todos os bits da matriz são 0
 - Sempre que a moldura k é referenciada, o hardware seta todos os **bits da linha k para 1**, e depois zera todos os **bits da coluna k para 0**
 - Deste modo, a qualquer instante a **linha** com o **menor valor binário** é a menos recentemente usada

LRU – Least Recently Used (5)

Página na
moldura 0

	0	1	2	3
0	0	1	1	1
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

Página na
moldura 1

	0	1	2	3
0	0	0	1	1
1	1	0	1	1
2	0	0	0	0
3	0	0	0	0

Página na
moldura 2

	0	1	2	3
0	0	0	0	1
1	1	0	0	1
2	1	1	0	1
3	0	0	0	0

Página na
moldura 3

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

Página na
moldura 2

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	1
3	1	1	0	0

Página na
moldura 1

	0	0	0	0
0	0	0	0	0
1	1	0	1	1
2	1	0	0	1
3	1	0	0	0

Página na
moldura 0

	0	1	1	1
0	0	1	1	1
1	0	0	1	1
2	0	0	0	1
3	0	0	0	0

Página na
moldura 3

	0	1	1	0
0	0	1	1	0
1	0	0	1	0
2	0	0	0	0
3	1	1	1	0

Página na
moldura 2

	0	1	0	0
0	0	1	0	0
1	0	0	0	0
2	1	1	0	1
3	1	1	0	0

Página na
moldura 3

	0	1	0	0
0	0	1	0	0
1	0	0	0	0
2	1	1	0	0
3	1	1	1	0

Aproximando LRU em Software

- Problema das abordagens LRU em HW
 - Dependem de um HW especial
 - Procurar uma solução em SW
- Aproximando LRU em Software
 - Algoritmo NFU - Not Frequently Used
 - Algoritmo Aging

NFU - *Not Frequently Used*

- Para cada página existe um contador
 - Iniciado com zero e incrementado a cada referência à página;
 - Página com menor valor do contador é candidata a troca;
- Esse algoritmo não se esquece de nada
 - Problema: pode retirar páginas que estão sendo referenciadas com frequência;
- Processo com várias funções:
 - F1 tem mais tempo de execução que os outras funções
 - páginas da F1 terão mais referências armazenadas;
 - Depois de um tempo F1 não será mais utilizado
 - Mas não sairá da memória pois contador esta alto!

Aging

- Modificação do NFU, resolvendo o problema descrito anteriormente;
 - Além de saber **quantas vezes** a página foi referenciada, também controla **quando** ela foi referenciada;
 - Geralmente, 8 bits são suficientes para o controle se as interrupções de relógio (clock ticks) ocorrem a cada 20ms (10^{-3});

Aging

Bits R para páginas 0-5

clock tick 0

1 0 1 0 1 1

Contadores



1	0	1	0	1	1
---	---	---	---	---	---

1	1	0	0	1	0
---	---	---	---	---	---

1	1	0	1	0	1
---	---	---	---	---	---

1	0	0	0	1	0
---	---	---	---	---	---

0	1	1	0	0	0
---	---	---	---	---	---

0	100000000
---	-----------

11000000

11100000

11110000

01111000

1 00000000

10000000

11000000

01100000

10110000

2 100000000

01000000

00100000

00100000

10001000

3 00000000

00000000

1000000

01000000

00100000

4 10000000

11 000000

01100000

10110000

01011000

5 10000000

01000000

10100000

01010000

00101000

a)

b)

c)

d)

e)



Referências

- Slides adaptados de Roberta Lima Gomes (UFES)
- Bibliografia
 - A. S. Tanenbaum, "Sistemas Operacionais Modernos", 3a. Edição, Editora Prentice-Hall, 2010.
 - Seção 3.4
 - Silberschatz A. G.; Galvin P. B.; Gagne G.; "Fundamentos de Sistemas Operacionais", 8a. Edição, Editora LTC, 2010.
 - Capítulo 9