

Sistemas Operacionais

Gerenciamento de Memória Virtual Algoritmos de Paginação

Norton Trevisan Roman
Marcelo Morandini
Jó Ueyama

Apostila baseada nos trabalhos de Kalinka Castelo Branco, Antônio Carlos Sementille, Luciana A. F. Martimiano e nas transparências fornecidas no site de compra do livro "Sistemas Operacionais Modernos"

Algoritmos de Troca de Página

- Ótimo;
- NRU;
- FIFO;
- Segunda Chance;
- Relógio;
- LRU;
- Working set;
- WSClock;
- Veremos cada um em detalhes

Algoritmos de Troca de Página

- Algoritmo ótimo
 - Cada página é marcada com o número de instruções que serão executadas antes que a página seja referenciada
 - Retira da memória a página que tem menos chance de ser referenciada (maior número de instruções faltantes)
 - Praticamente impossível de se saber;
 - Impraticável;
 - Usado em simulações para comparação com outros algoritmos;

Algoritmos de Troca de Página

- Algoritmo Not Recently Used Page Replacement (NRU)
 - Para auxiliar o S.O. a coletar estatísticas de página de uso:
 - 02 bits associados a cada página → R(eferenciada) e M(odificada)
 - Classe 0 (00) → não referenciada, não modificada;
 - Classe 1 (01) → não referenciada, modificada;
 - Classe 2 (10) → referenciada, não modificada;
 - Classe 3 (11) → referenciada, modificada;
 - Referenciada → lida ou escrita
 - Modificada → escrita

Algoritmos de Troca de Página

- Algoritmo Not Recently Used Page Replacement (NRU)
 - Para auxiliar o S.O. a coletar estatísticas de página de uso:
 - R e M são atualizados a cada referência à memória;
 - Armazenados em cada entrada da tabela de página
 - Seu valor é determinado pelo hardware
 - Quando um processo é iniciado, ambos R e M são 0 para todas suas páginas
 - Periodicamente, o bit R é limpo para diferenciar as páginas que não foram referenciadas recentemente;
 - A cada interrupção de relógio;

Algoritmos de Troca de Página

- Algoritmo Not Recently Used Page Replacement (NRU)
 - O bit M não é limpo, pois o S.O. precisa saber se deve escrever a página no disco
 - Quando ocorre uma page fault:
 - Remove uma página aleatoriamente, escolhendo dentre as classes mais inferiores → bits 00, 01, 10, 11
 - Vantagens:
 - Fácil de entender, eficiente para implementar e fornece bom desempenho;

Algoritmos de Troca de Página

- Algoritmo First-in First-out Page Replacement (FIFO)
 - SO mantém uma fila das páginas correntes na memória;
 - A página no início da fila é a mais antiga e a página no final é a mais nova;
 - Quando ocorre um page fault
 - A página do início é removida
 - A nova é inserida ao final da fila
 - Simples, mas pode ser ineficiente, pois uma página que está em uso constante pode ser retirada;
 - Pouco utilizado;

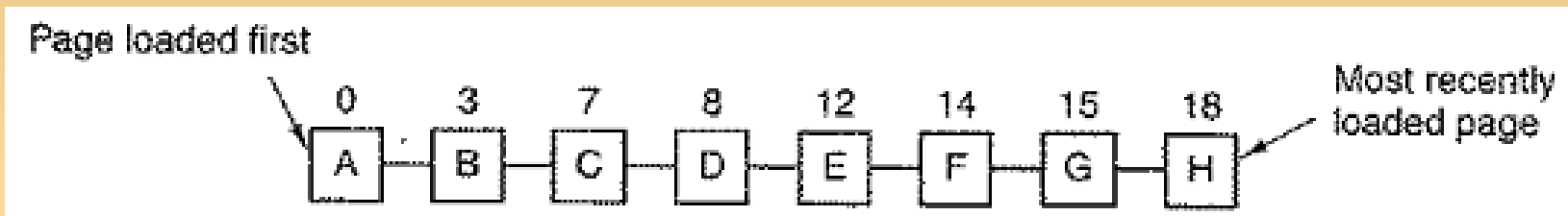
Algoritmos de Troca de Página

- Algoritmo da Segunda Chance
 - FIFO + bit R;
 - Inspecciona o bit R da página mais velha
 - Se for 0, ela é velha e não foi usada recentemente → é trocada
 - Se for 1, 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

Algoritmos de Troca de Página

- Algoritmo da Segunda Chance

- Ex:



- Ocorre page fault no tempo 20 e $R_A = 0$



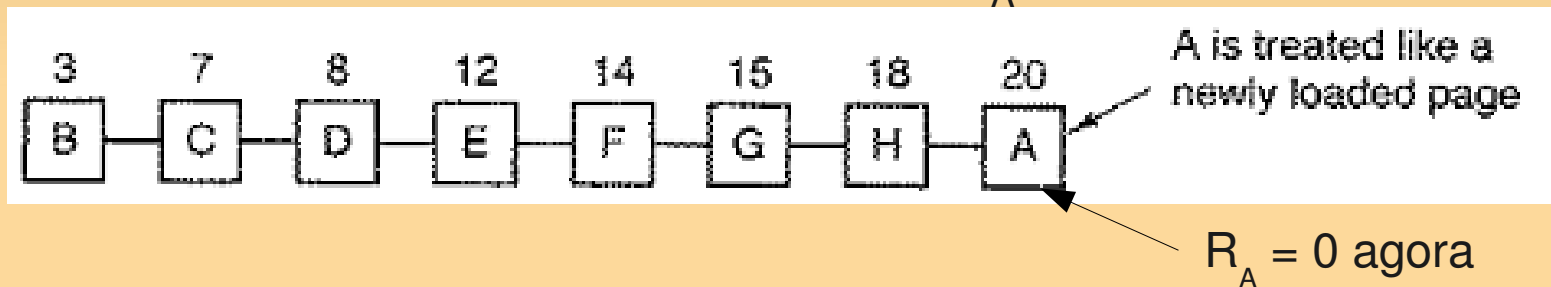
- A é removido, e o novo elemento é inserido ao final

Algoritmos de Troca de Página

- Algoritmo da Segunda Chance

- Ex:

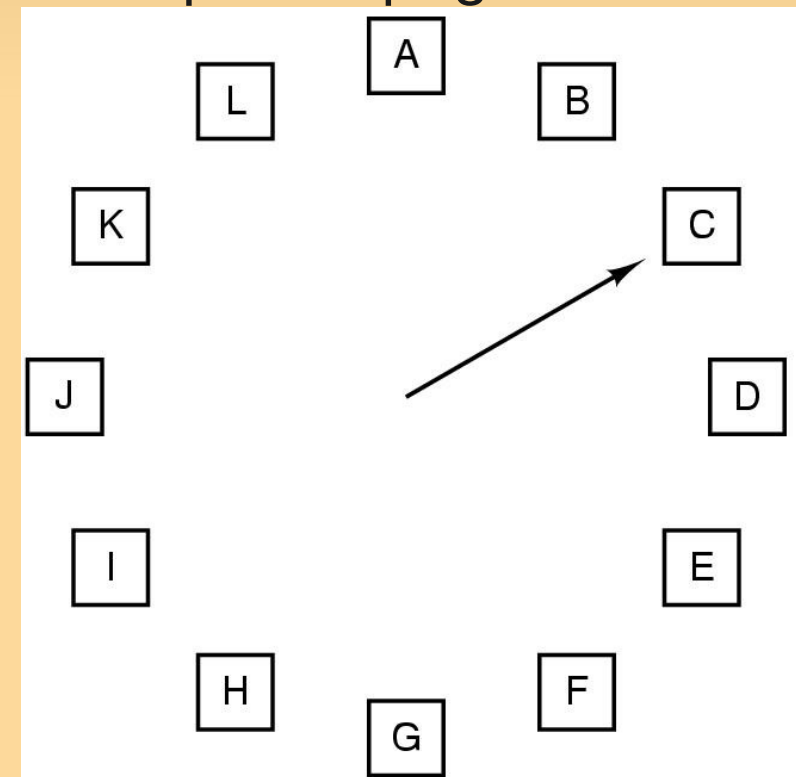
- Ocorre page fault no tempo 20 e $R_A = 1$



- Repete a operação com B
 - Se $R_B = 0$, troca
 - Senão, passa ao final da fila, com $R_B = 0$, e verifica-se C

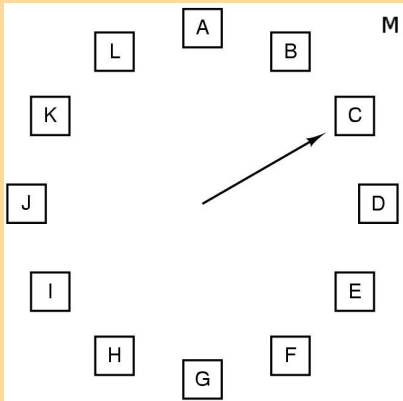
Algoritmos de Troca de Página

- Algoritmo do Relógio
 - Melhoria ao Segunda Chance:
 - Lista circular com ponteiro apontando para a página mais antiga, na forma de um relógio
 - A cabeça aponta para a página mais antiga



Algoritmos de Troca de Página

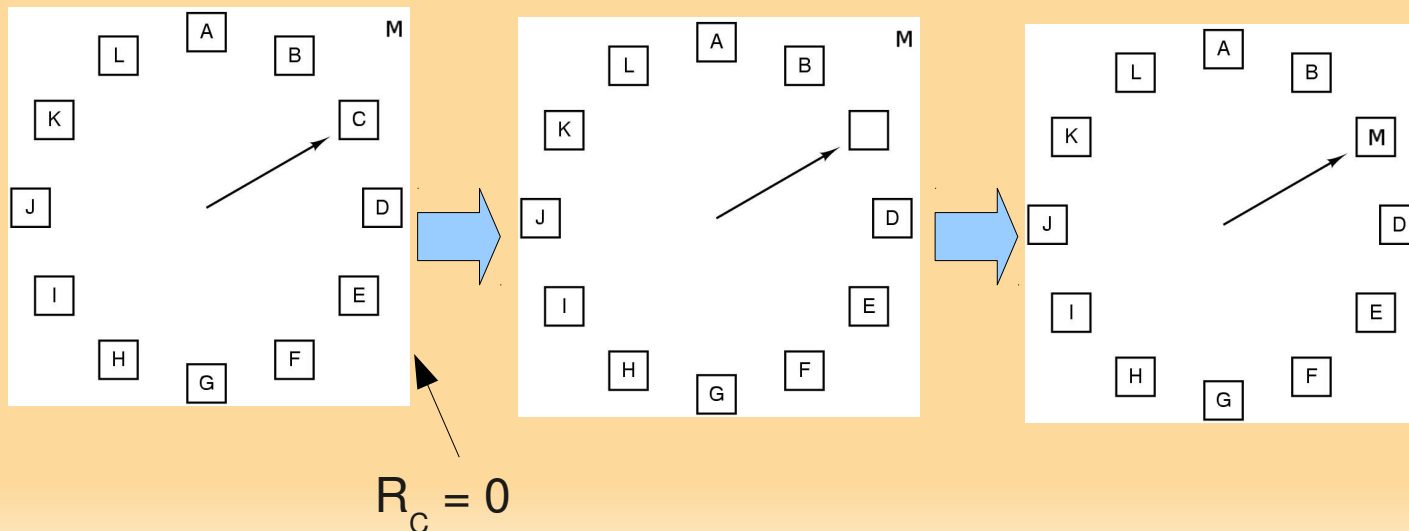
- Algoritmo do Relógio
 - Quando ocorre um page fault:
 - Inspecciona-se a cabeça da lista



$$R_c = 0$$

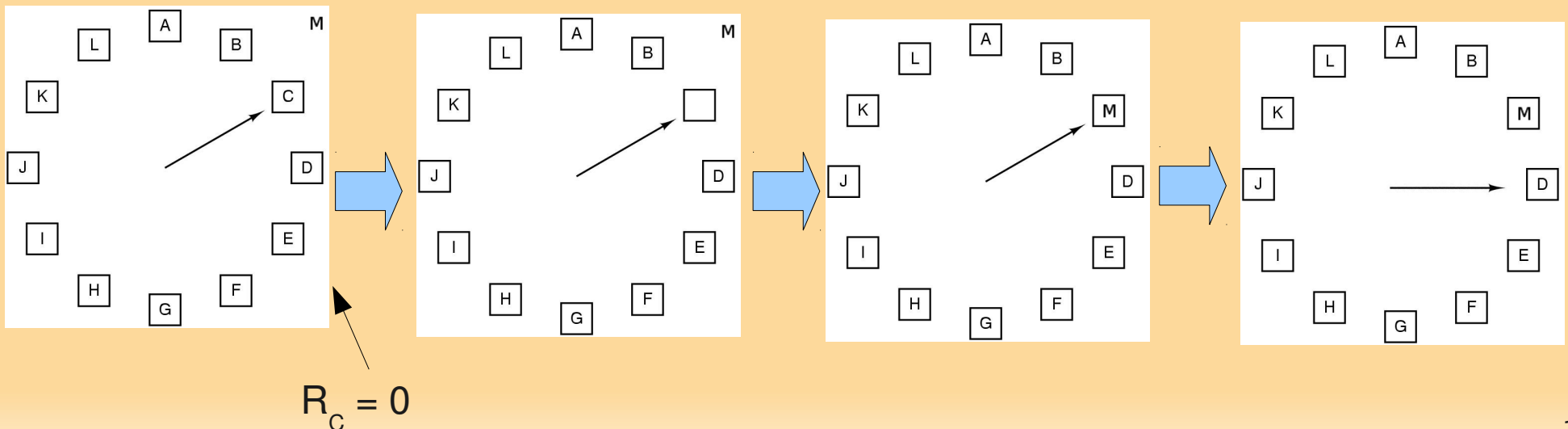
Algoritmos de Troca de Página

- Algoritmo do Relógio
 - Quando ocorre um page fault:
 - Inspecciona-se a cabeça da lista
 - Se $R = 0$:
 - Substitui-se a página da cabeça pela nova página



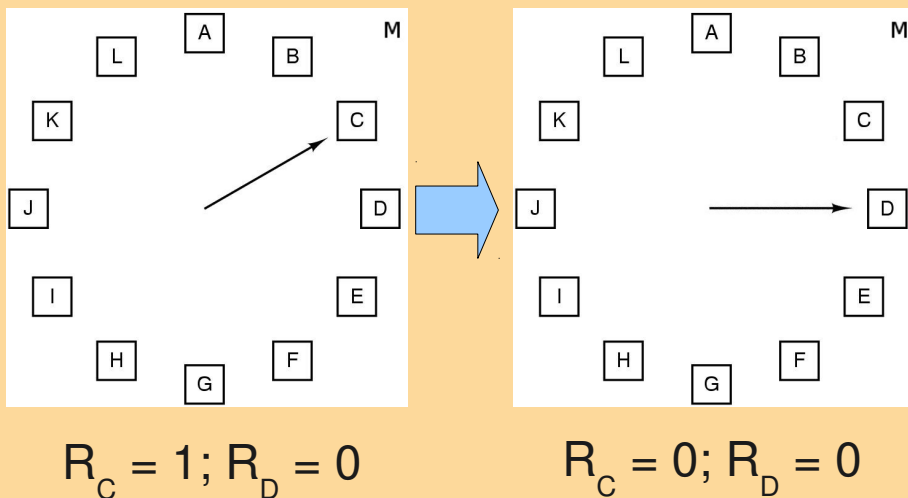
Algoritmos de Troca de Página

- Algoritmo do Relógio
 - Quando ocorre um page fault:
 - Inspecciona-se a cabeça da lista
 - Se $R = 0$:
 - Substitui-se a página da cabeça pela nova página
 - Avança-se a cabeça em uma posição



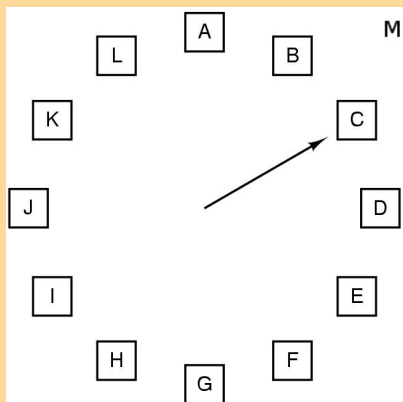
Algoritmos de Troca de Página

- Algoritmo do Relógio
 - Quando ocorre um page fault:
 - Inspecciona-se a cabeça da lista
 - Se $R = 1$:
 - Avança-se a cabeça em uma posição
 - Repete-se o processo até encontrar página com $R = 0$

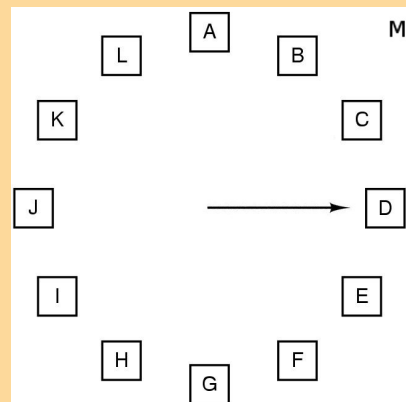


Algoritmos de Troca de Página

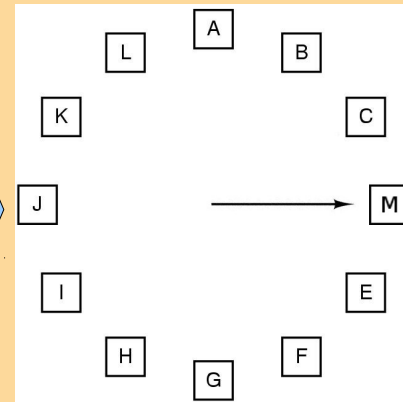
- Algoritmo do Relógio
 - Quando ocorre um page fault:
 - Inspecciona-se a cabeça da lista
 - Se $R = 1$:
 - Avança-se a cabeça em uma posição
 - Repete-se o processo até encontrar página com $R = 0$
 - Age como no caso anterior ($R=0$)



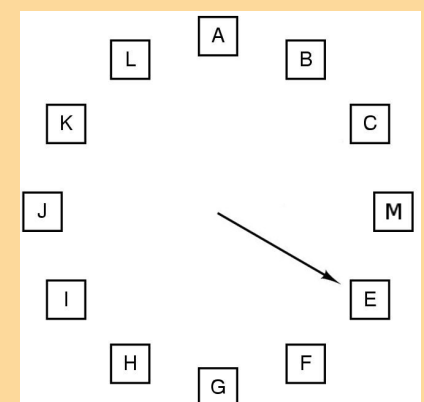
$$R_C = 1; R_D = 0$$



$$R_C = 0; R_D = 0$$



$$R_C = 0; R_D = 1$$



Algoritmos de Troca de Página

- Algoritmo Least Recently Used Page Replacement (LRU)
 - Idéia:
 - Páginas que foram muito usadas nas últimas instruções serão provavelmente usadas novamente nas próximas
 - Troca a página que permaneceu em desuso pelo maior tempo
 - Alto custo
 - Deve-se manter lista encadeada com todas as páginas que estão na memória, com as mais recentemente utilizadas no início e as menos utilizadas no final;
 - A lista deve ser atualizada a cada referência da memória;

Algoritmos de Troca de Página

- Algoritmo Least Recently Used Page Replacement (LRU)
 - Pode ser implementado tanto por hardware quanto por software:
 - Hardware: MMU deve suportar a implementação LRU;
 - Contador em hardware (64 bits), incrementado automaticamente após cada instrução
 - Tabela de páginas armazena o valor desse contador – C – em cada entrada
 - Após cada referência à memória, o valor atual de C é armazenado na entrada correspondente (página) na tabela
 - Em um page fault, o S.O. examina todas as entradas na tabela, para encontrar a com menor C

Algoritmos de Troca de Página

- Algoritmo Least Recently Used Page Replacement (LRU)
 - Implementação:
 - Hardware (Alternativo):
 - Se o computador tem n molduras, o hardware de LRU mantém uma matriz de $n \times n$ bits, inicialmente zero
 - Quando uma moldura k (ex, $k=0$) é referenciada, o hardware faz todos os bits da linha k serem 1, e os da coluna k serem 0

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

Algoritmos de Troca de Página

- Algoritmo Least Recently Used Page Replacement (LRU)
 - Implementação:
 - Hardware (Alternativo):
 - A qualquer momento, a linha com o menor valor binário é a menos recentemente usada. A linha seguinte é a segunda menos recentemente usada, e assim por diante.
 - Ex:

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

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

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

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

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

Páginas:

0

1

2

3

2

Algoritmos de Troca de Página

- Algoritmo Least Recently Used Page Replacement (LRU)

- Implementação:

- Hardware (Alternativo):

- Ex:

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

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

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

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

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

Páginas:

0 1 2 3 2

0	0	0	0	0
1	0	1	1	1
1	0	0	1	1
1	0	0	0	0

0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
0	0	0	0	0

0	1	1	0	0
0	0	1	0	0
0	0	0	0	0
1	1	1	0	0

0	1	0	0	0
0	0	0	0	0
1	1	0	1	1
1	1	0	0	0

0	1	0	0	0
0	0	0	0	0
1	1	0	0	0
1	1	1	0	0

Páginas:

1 0 3 2 3

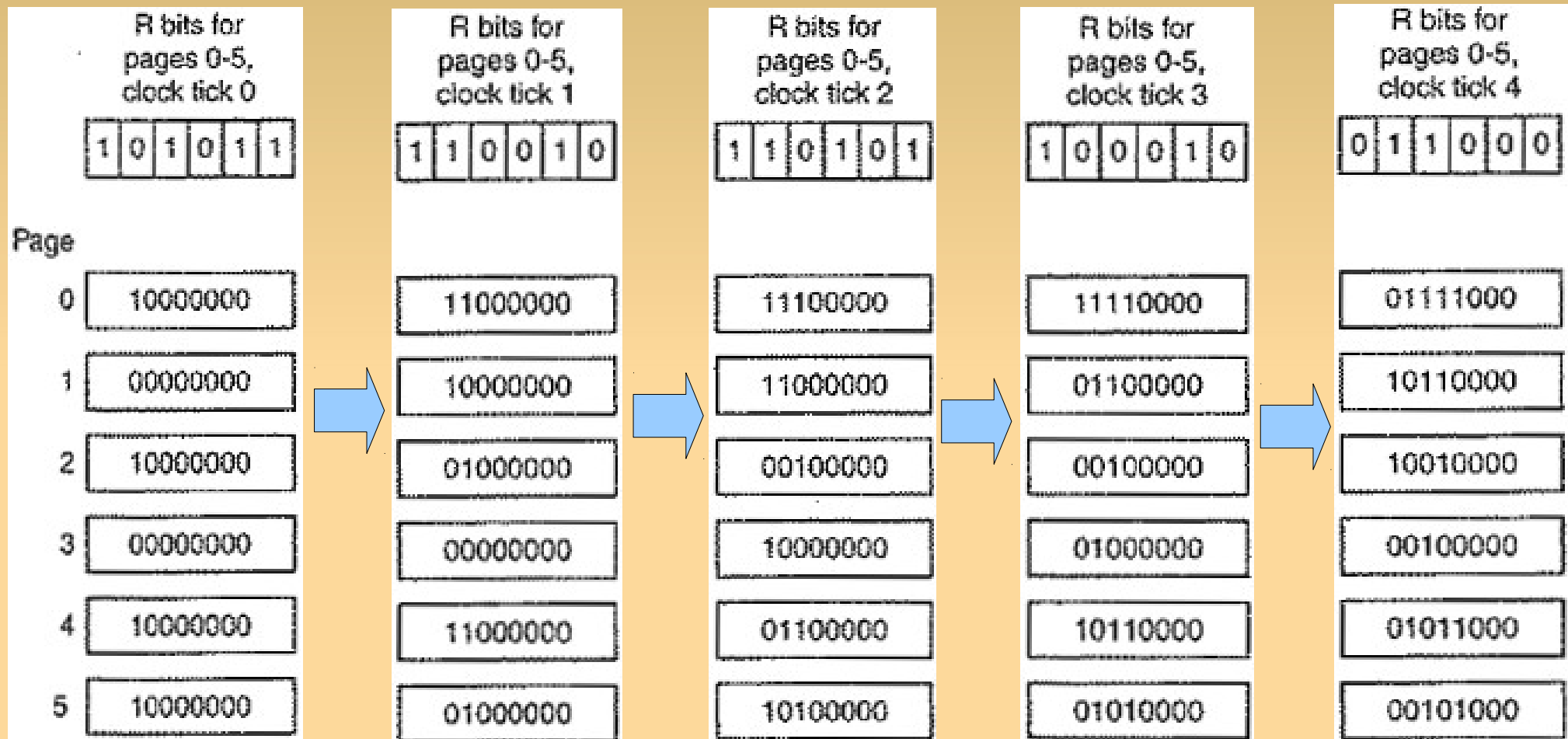
Algoritmos de Troca de Página

- Algoritmo Least Recently Used Page Replacement (LRU)
 - Pode ser implementado tanto por hardware quanto por software:
 - Software – NFU (Not frequently used):
 - Para cada página existe um contador, iniciado com zero
 - A cada interrupção do clock, o SO varre todas as páginas da memória
 - Para cada página, adiciona o bit R (residência) ao contador
 - Em um page fault, escolhe a página com o menor contador
 - Problema: Como esse algoritmo não se esquece de nada, páginas frequentemente acessadas em uma porção pequena do código, mas que não mais serão acessadas, não serão candidatas

Algoritmos de Troca de Página

- Algoritmo Least Recently Used Page Replacement (LRU)
 - Pode ser implementado tanto por hardware quanto por software:
 - Software – Aging:
 - Solução ao problema do algoritmo NFU
 - Além de saber quantas vezes a página foi referenciada, também controla quando ela foi referenciada
 - Primeiro, os contadores são deslocados à direita em um bit. Só então o bit R é adicionado, só que ao bit mais da esquerda
 - Também a cada interrupção do clock
 - Em um page fault, a página com o menor contador é removida

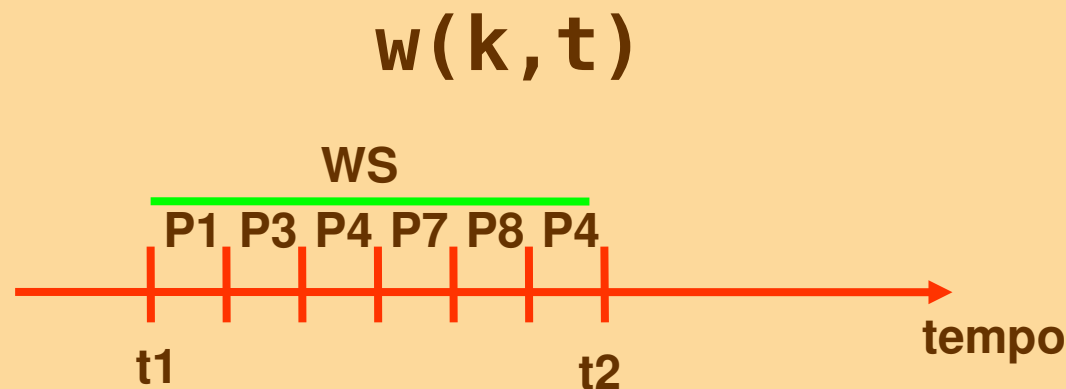
Aging



Note que após 8 clocks uma página não referenciada tem seu contador zerado. Quanto mais tempo ficar sem ser referenciada, mais zeros à sua esquerda terá, e menor será seu contador

Algoritmos de Troca de Página

- Algoritmo Working Set (WS):
 - 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 (referenciando) em um determinado tempo t ;



Algoritmos de Troca de Página

- Algoritmo Working Set (WS):
 - Objetivo principal: reduzir a falta de páginas
 - Um processo só é executado quando todas as páginas necessárias no tempo t estão carregadas na memória;
 - 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

Algoritmos de Troca de Página

- Algoritmo Working Set (WS):
 - 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.
 - Pré-paginação 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 moldura livre

Algoritmos de Troca de Página

- Algoritmo Working Set (WS):
 - Implementação:
 - Contar as k referências mais recentes é custoso
 - 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;

Algoritmos de Troca de Página

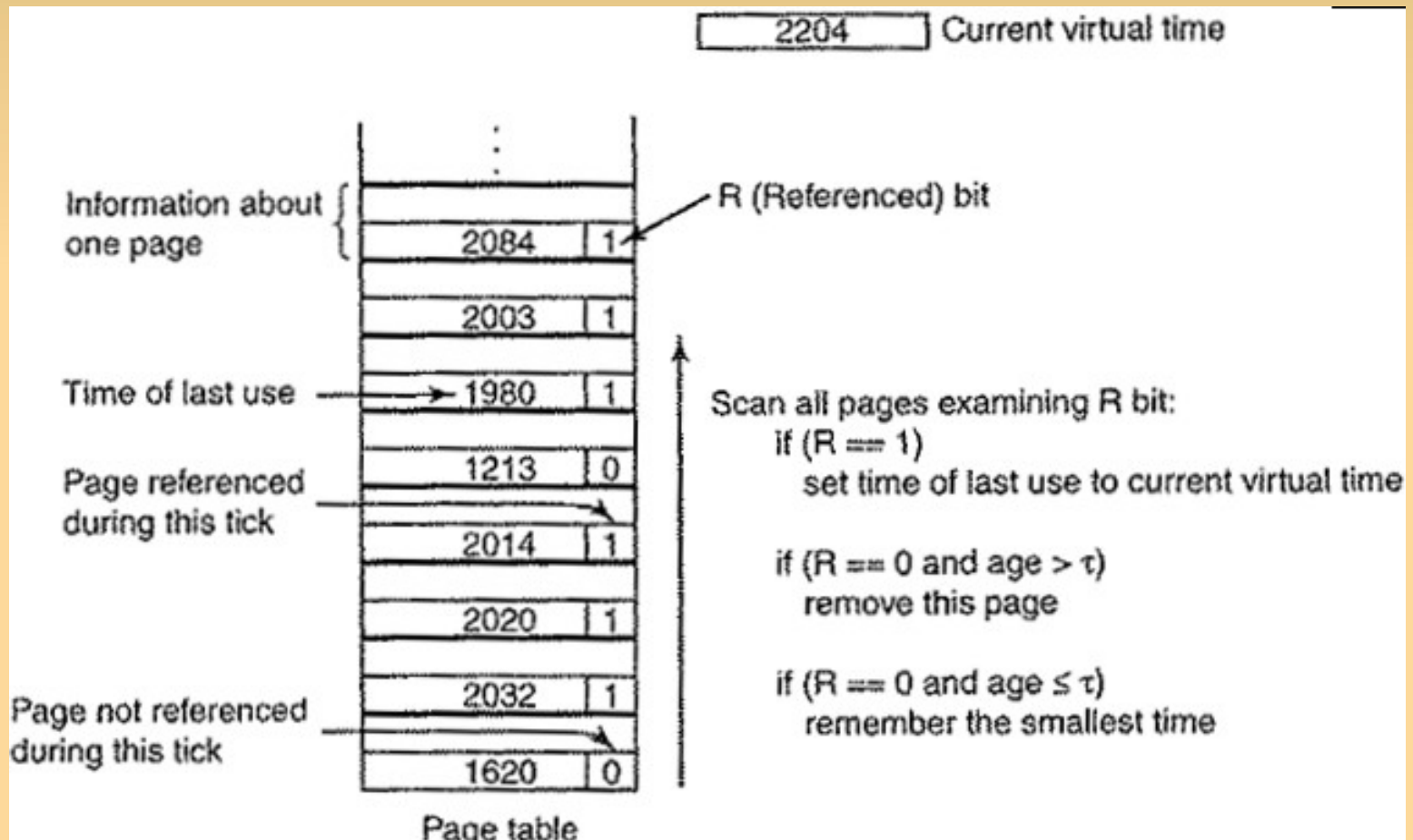
- Algoritmo Working Set (WS):
 - Algoritmo:
 - Pressupostos:
 - O hardware define os bits R e M
 - Em cada ciclo do clock, o bit de referência é limpo
 - O tempo do working set se estende por vários ciclos do clock
 - Em cada page fault, a tabela de páginas inteira é buscada
 - À medida que cada entrada 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

Algoritmos de Troca de Página

- Algoritmo Working Set (WS):
 - 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
 - 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 $R=0$

Algoritmos de Troca de Página

- Algoritmo Working Set (WS):



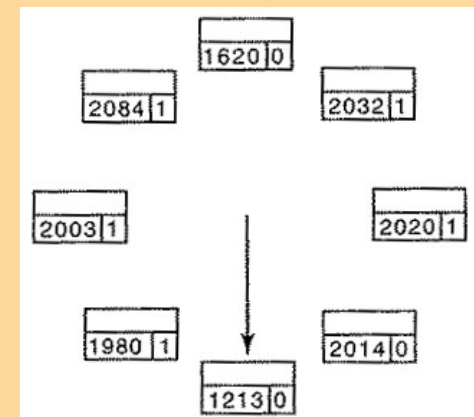
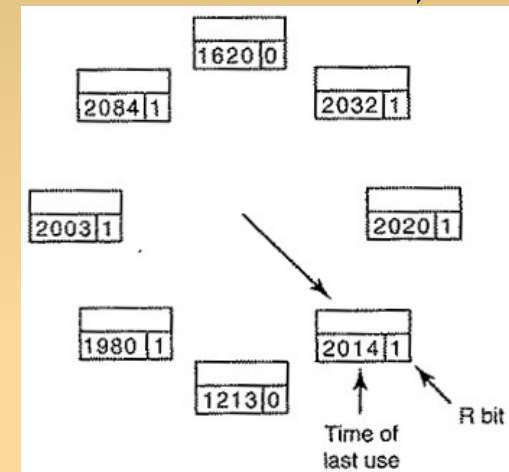
Algoritmos de Troca de Página

- Algoritmo WSClock:
 - Clock + 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

Algoritmos de Troca de Página

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

M não é mostrado na figura



Algoritmos de Troca de Página

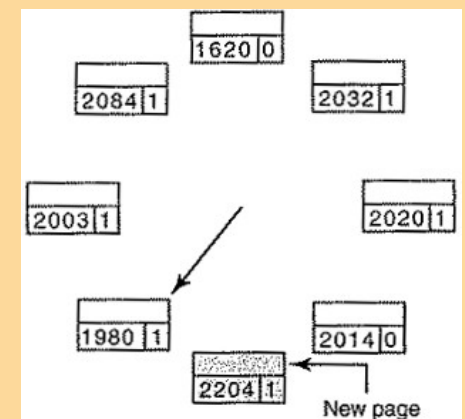
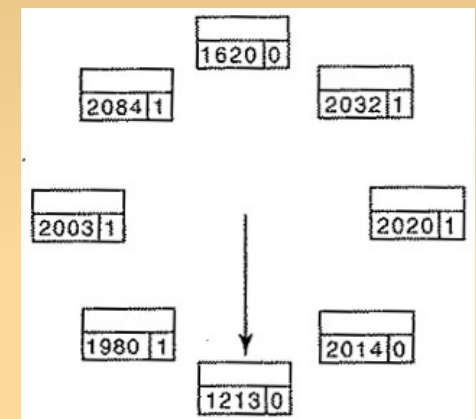
- Algoritmo WSClock:

- Funcionamento:

- 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
- A cabeça da lista avança



Algoritmos de Troca de Página

- Algoritmo WSClock:

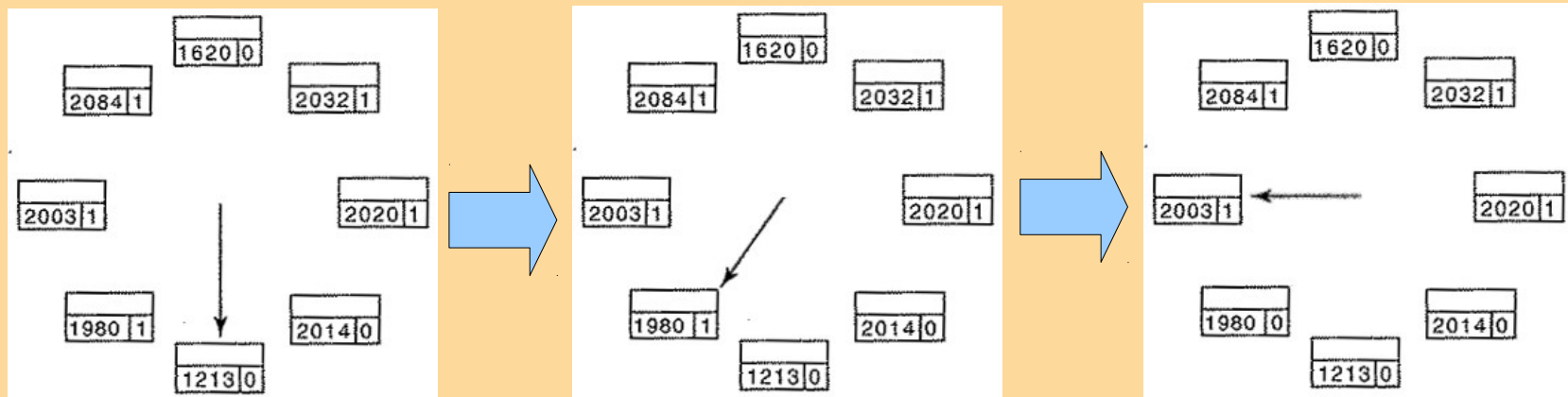
- Funcionamento:

- Se $R=0$

- Se, contudo, a página estiver suja \rightarrow não possui cópia válida no disco

- Agenda uma escrita ao disco, evitando troca de processo

- Avança a cabeça da lista, prosseguindo da página seguinte



Algoritmos de Troca de Página

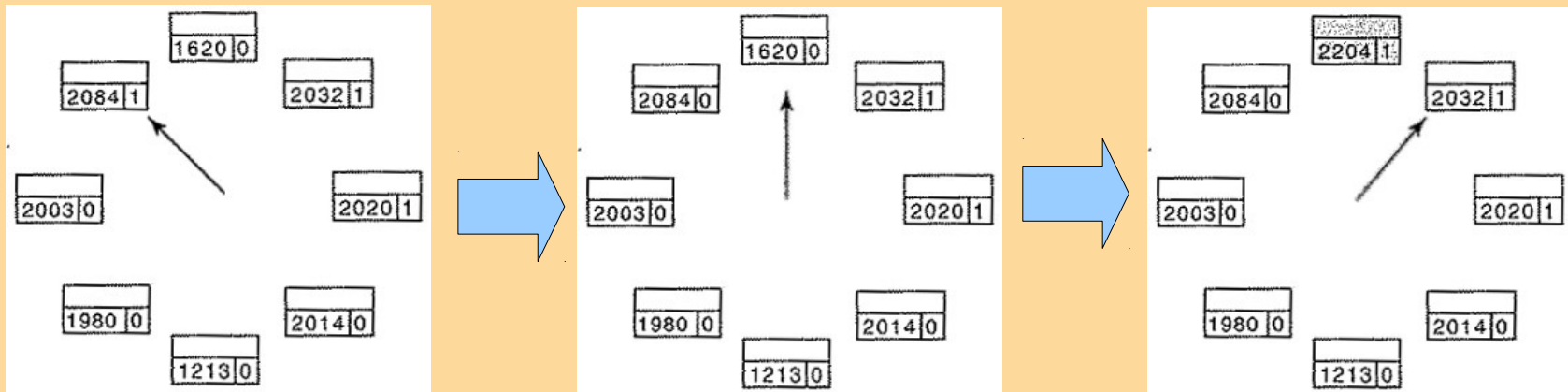
- Algoritmo WSClock:

- Funcionamento:

- Se $R=0$

- Se, contudo, a página estiver suja \rightarrow não possui cópia válida no disco

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



Algoritmos de Troca de Página

- Algoritmo WSClock:
 - 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

Algoritmos de Troca de Página

- Algoritmos de substituição local:
 - Working Set;
 - WSClock;
 - O conceito de working set se aplica somente a um único processo → não há working set para a máquina como um todo
- Algoritmos de substituição local/global:
 - Ótimo;
 - NRU;
 - FIFO;
 - Segunda Chance;
 - LRU;
 - Relógio;

Implementação da Paginação

- Onde colocar as páginas no disco, quando retiradas da memória?
 - A solução mais simples é ter uma partição especial de swap
 - Solução do Unix e Linux
 - Não possui um sistema de arquivos normal
 - Quando o sistema inicia, a partição está vazia
 - Representada na memória como uma única entrada contendo sua origem e tamanho
 - À medida em que processos são iniciados, o SO reserva um pedaço da área de swap do tamanho do processo
 - Quando terminam, o espaço é liberado
 - A área de troca é gerenciada como uma lista de espaços disponíveis;

Implementação da Paginação

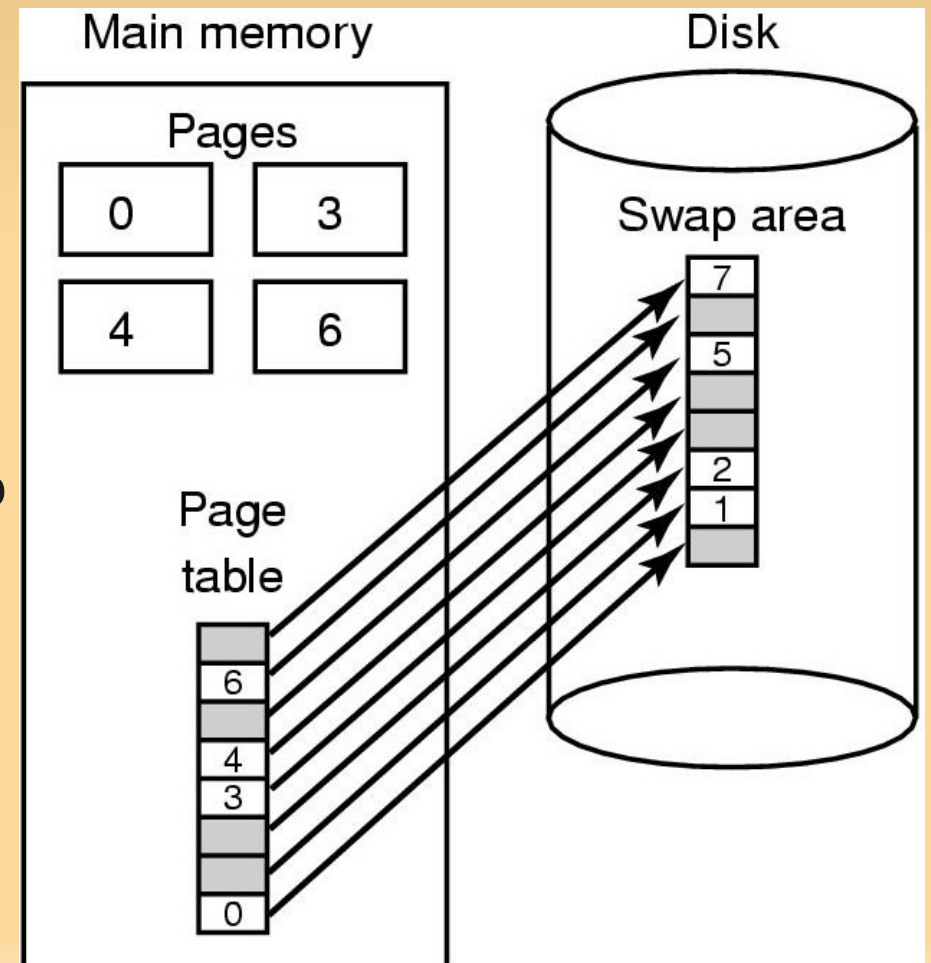
- Onde colocar as páginas no disco, quando retiradas da memória?
 - A solução mais simples é ter uma partição especial de swap
 - Há algoritmos melhores, mas que não serão discutidos
 - Associado a cada processo está o endereço no disco de sua área de swap
 - Mantido na tabela de processos
 - Cálculo do endereço para escrever uma página:
 - Adicione o endereço do início da página (seu valor no endereço virtual) ao início da área de swap associada ao processo

Implementação da Paginação

- Onde colocar as páginas no disco, quando retiradas da memória?
 - Problema: antes de um processo iniciar, a área de swap deve ser inicializada
 - Possibilidade A – Assim que o processo é criado, ele é copiado todo para sua área de troca no disco, sendo carregado para memória quando necessário;
 - Alternativamente, podemos copiá-lo todo para a memória principal (espelhamento)
 - Problema: processos podem aumentar de tamanho após iniciarem (pilha e dados)
 - Solução: reservar áreas de troca diferentes para texto do programa, dados e pilha, permitindo que elas consistam de mais de um bloco no disco

Implementação da Paginação

- Onde colocar as páginas no disco, quando retiradas da memória?
 - Problema: antes de um processo iniciar, a área de swap deve ser inicializada
 - Possibilidade A
 - Basta saber o endereço do início da área de swap do processo
 - As páginas são espelhadas no disco
 - Área de troca (swap) estática



Implementação da Paginação

- Onde colocar as páginas no disco, quando retiradas da memória?
 - Problema: antes de um processo iniciar, a área de swap deve ser inicializada
 - Possibilidade B – Nada é alocado antecipadamente.
 - Espaço é alocado em disco quando a página for enviada para lá e desalocado quando volta para a memória
 - Assim, processo na memória RAM não fica “amarrado” a uma área específica;
 - Desvantagem: precisamos, na memória, de um endereço de disco para cada página.
 - Deve haver uma tabela em cada processo dizendo onde cada página está no disco (se estiver lá)
 - Antes, bastava saber onde o processo estava no disco

Implementação da Paginação

- Onde colocar as páginas no disco, quando retiradas da memória?
 - Problema: antes de um processo iniciar, a área de swap deve ser inicializada
 - Possibilidade B
 - Além do endereço do início da área de swap do processo, temos que saber onde está a página dentro desse endereço (seu deslocamento)
 - Área de troca dinâmica

