

SSC 140 - SISTEMAS OPERACIONAIS I

Turmas A e B

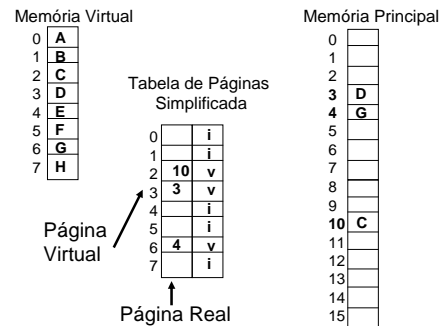
Aula 14 – Gerenciamento de Memória

Profa. Sarita Mazzini Bruschi

Slides de autoria de
Luciana A. F. Martimiano baseados no livro
Sistemas Operacionais Modernos de A. Tanenbaum

Gerenciamento de Memória

Troca de Páginas



2

Gerenciamento de Memória

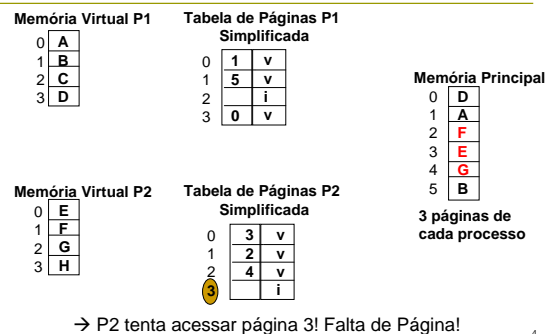
Troca de Páginas

- Se todas as páginas estiverem ocupadas, uma página deve ser retirada: página vítima;
- Exemplo:
 - Dois processos P1 e P2, cada um com 4 páginas virtuais;
 - Memória principal com 6 páginas;

3

Gerenciamento de Memória

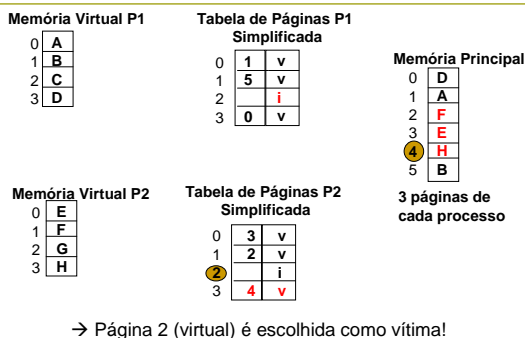
Troca de Páginas



4

Gerenciamento de Memória

Troca de Páginas



5

Gerenciamento de Memória

Troca de Páginas - Paginação

- Algoritmos:
 - Ótimo;
 - NRU;
 - FIFO;
 - Segunda Chance;
 - Relógio;
 - LRU;
 - Working set;
 - WSClock;

6

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo Ótimo:

- Retira da memória a página que tem menos chance de ser referenciada;
 - Praticamente impossível de se saber;
 - Impraticável;
 - Usado em simulações para comparação com outros algoritmos;

7

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo *Not Recently Used Page Replacement* (NRU) ou *Não Usada Recentemente* (NUR)

- Troca as páginas não utilizadas recentemente;
- 02 bits associados a cada página → R (referência) e M (modificação)
 - Classe 0 (R = 0 e M = 0) → não referenciada, não modificada;
 - Classe 1 (R = 0 e M = 1) → não referenciada, modificada;
 - Classe 2 (R = 1 e M = 0) → referenciada, não modificada;
 - Classe 3 (R = 1 e M = 1) → referenciada, modificada;
- R e M são atualizados a cada referência à memória;

8

Gerenciamento de Memória Troca de Páginas - Paginação

NRU:

- Periodicamente, o bit R é limpo para diferenciar as páginas que não foram referenciadas recentemente;
 - A cada *tick* do relógio ou interrupção de relógio;
 - Classe 3 → Classe 1;
- Vantagens: fácil de entender, eficiente para implementar e fornece bom desempenho;

9

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo *First-in First-out Page Replacement* (FIFO)

- SO mantém uma listas das páginas correntes na memória;
 - A página no início da lista é a mais antiga e a página no final da lista é a mais nova;
- Simples, mas pode ser ineficiente, pois uma página que está em uso constante pode ser retirada;
- Pouco utilizado;

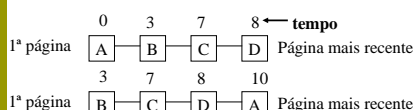
10

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo da Segunda Chance

- FIFO + *bit* R;
- Página mais velha é candidata em potencial;

Se o bit $R=0$, então página é retirada da memória, senão, $R=0$ e se dá uma nova chance à página colocando-a no final da lista;



Se página A com $R=1$; e falta de página em tempo 10; Então $R=0$ e página A vai para final da lista;

11

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo do Relógio

- Lista circular com ponteiro apontando para a página mais antiga
- Algoritmo se repete até encontrar $R=0$;

Se $R=0$

- troca de página
- desloca o ponteiro

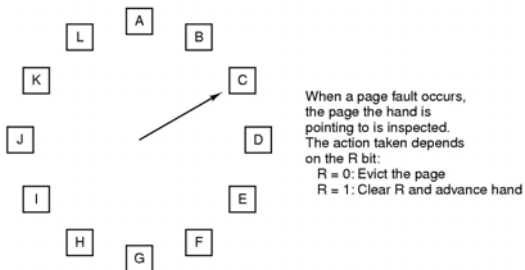
Se $R=1$

- $R=0$
- desloca o ponteiro
- continua busca

12

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo do Relógio



13

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo *Least Recently Used Page Replacement* (LRU) ou *Menos Recentemente Usada* (MRU)

- Troca a página menos referenciada/modificada recentemente;
- Alto custo
 - Lista encadeada com 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;

14

Gerenciamento de Memória Troca de Páginas - Paginação

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*);
 - Após cada referência à memória, o valor do contador é armazenado na tabela de páginas;
 - Quando ocorre falta de página, o SO examina todos os contadores e escolhe a página que tem o menor valor
 - **Software:** duas maneiras
 - NFU (*Not frequently used*) ou LFU (*least frequently used*);
 - *Aging* (Envelhecimento);

15

Gerenciamento de Memória Troca de Páginas - Paginação

Software: NFU ou LFU (*least*)

- 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;
 - Compilador com vários passos: passo 1 tem mais tempo de execução que os outros passos → páginas do passo 1 terão mais referências armazenadas;

16

Gerenciamento de Memória Troca de Páginas - Paginação

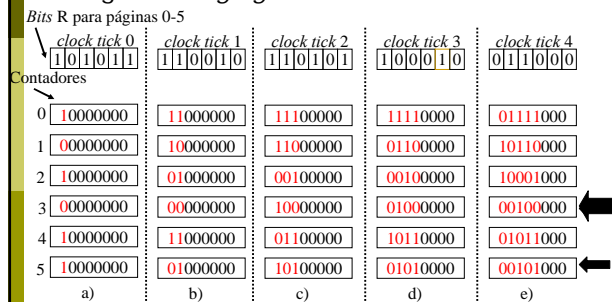
Software: Algoritmo *aging* (envelhecimento)

- 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});

17

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo *aging*

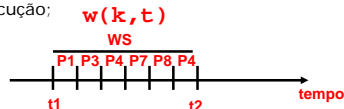


18

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo *Working Set* (WS):

- **Paginação por demanda** → páginas são carregadas na memória somente quando são necessárias;
- **Pré-paginação** → *Working set*
 - Carregar um conjunto de páginas que um processo está efetivamente utilizando (referenciando) em um determinado tempo t antes de ele ser posto em execução;



19

Gerenciamento de Memória Troca de Páginas - Paginação

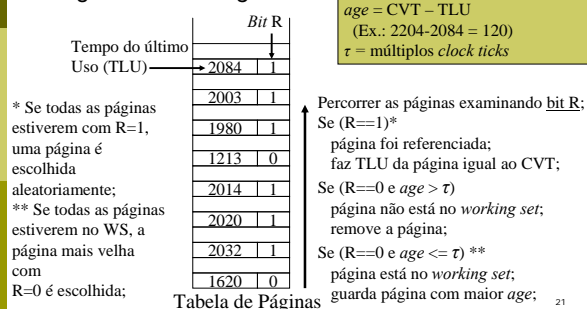
Algoritmo *Working Set* (WS):

- Objetivo principal: reduzir a falta de páginas
 - Um processo só é executado quando todas as páginas necessárias no tempo t estão carregadas na memória;
 - SO gerencia quais páginas estão no *Working Set*;
- Para simplificar → o *working set* pode ser visto como o conjunto de páginas que o processo referenciou durante os últimos t segundos de tempo;
- Utiliza *bit R* e o tempo de relógio (tempo virtual) da última vez que a página foi referenciada;

20

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo *Working Set*:



21

Gerenciamento de Memória Troca de Páginas - Paginação

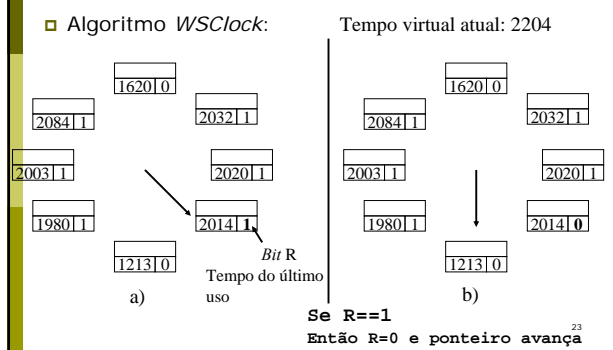
Algoritmo *WSClock*:

- *Clock + Working Set*;
- Lista circular de molduras de páginas formando um anel a cada página carregada na memória;
- Utiliza *bit R* e o tempo da última vez que a página foi referenciada;
- *Bit M* utilizado para agendar escrita em disco;

22

Gerenciamento de Memória Troca de Páginas - Paginação

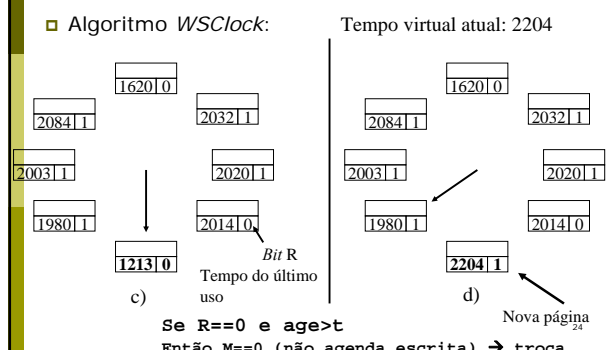
Algoritmo *WSClock*:



23

Gerenciamento de Memória Troca de Páginas - Paginação

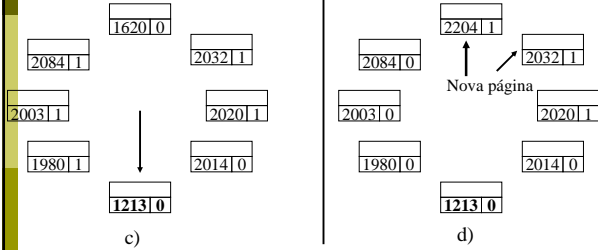
Algoritmo *WSClock*:



24

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo *WSclock*:



$R==0$ e $age>t$
 $M==1$ (agenda escrita e continua procura) ²⁵

Gerenciamento de Memória Troca de Páginas - Paginação

Algoritmo *WSclock*:

- Se todas estiverem com $M==1$; então escreve página atual no disco, e troca a página;
- Melhor desempenho → menos acessos ao disco;

26