

# Gerenciamento de Memória

Algoritmos de substituição de páginas



# Paginação

- Vimos que a paginação permite carregar partes de um processo na memória, em área não contínuas.
- Vimos que a MMU é responsável por traduzir endereços lógicos em endereços físicos.
- Vimos que quando uma página não está na memória principal ocorre uma page fault.
- Quando a memória está cheia e uma nova página precisa ser carregada, uma página na memória principal precisa ir para o disco
- Veremos como o S.O. escolhe a página que irá para o disco



# O algoritmo perfeito

- O melhor algoritmo de substituição de página é aquele em que:
  - A página retirada da memória para o disco seja a página que está mais longe de ser usada novamente
- Exemplo:
  - Se temos na memória a página A, B, C e D.
  - Veremos quando no futuro cada uma será usada.  
**| A = 10 | B = 5 | C = 8 | D = 20 |**
  - Retiramos a que será usada o mais distante possível no futuro, a página D.



# Algoritmo Perfeito

- Infelizmente ainda não podemos prever o futuro.
- Portanto este algoritmo como descrito, não pode ser implementado na prática.
- Precisaremos de alternativas, essas alternativas são:
  - NRU
  - FIFO
  - FIFO com segunda chance
  - Relógio
  - Menos recentemente usado
- Veremos em detalhes estes algoritmos



# Estatísticas das páginas

- Para que o S.O. saiba quanto e quando uma página foi usada, é necessário armazenar informação sobre seu uso.
- A maneira mais simples, e que muitos hardwares implementam é a o uso de 2 bits para cada página.
  - Bit **R**ead – Indica se a página foi lida por um processo.
  - Bit **M**odified – Indica que a página foi modificada por um processo.
- Uma vez setados para 1 esses bits ficam em 1 até receberem o reset do S.O.



# Não Recentemente Usado (NRU)

- A ideia central deste algoritmo é dividir as páginas em 4 classes, de acordo com os bits  $R$  e  $M$ .
- Retirar uma página qualquer que esteja na menor classe possível.

R	M	Descrição
0	0	Não referenciado, não modificado
0	1	Não referenciado, modificado
1	0	Referenciado, não modificado
1	1	Referenciado e modificado

# FIFO

- A ideia é simples:
  - Retire a página mais antiga para colocar a nova
- A primeira página a entrar na memória será a primeira a sair.
- É um algoritmo de implementação rápida
- O problema é: Este algoritmo pode substituir uma página muito lida
- Este problema impede o uso desse algoritmo nesta forma



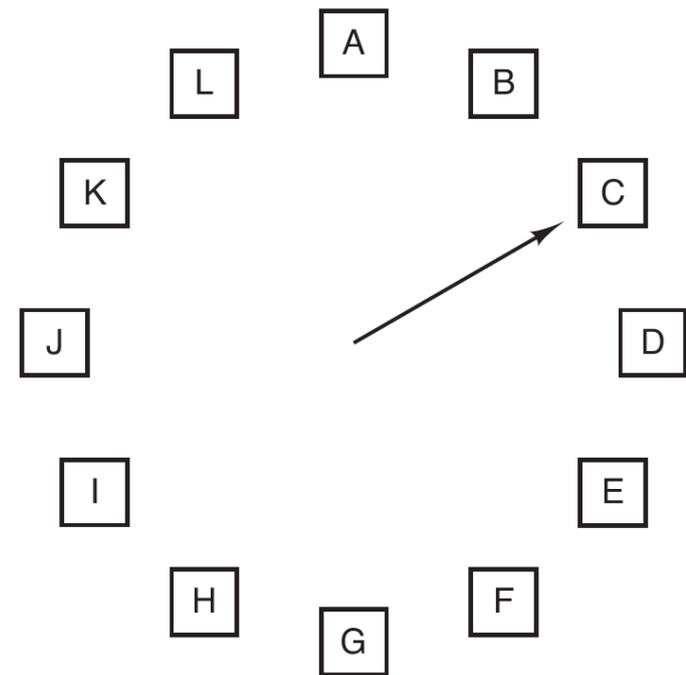
# FIFO com segunda chance

- É uma alteração simples do FIFO, mas que permite que uma página muito usada não seja retirada imediatamente.
- A diferença é:
  - Ao selecionar a página a ser retirada se o bit **R** estiver setado, significa que ela foi referenciada recentemente
  - Neste caso não retirará esta página apenas a coloca no fim da fila e reseta o bit **R**
- Essa é uma variação mais aplicável do algoritmo FIFO



# Relógio

- Uma espécie de Round-Robin para as páginas em memória.
- Se  $R = 0$  Remove a página
- Se  $R = 1$  Zerar  $R$  e avançar o ponteiro



Quando ocorre uma falta de página, a página indicada pelo ponteiro é inspecionada. A ação executada depende do bit  $R$ :

$R = 0$ : Remover a página

$R = 1$ : Zerar  $R$  e avançar o ponteiro



# Menos recentemente usado (LRU)

- É necessário manter um contador de tempo geral.
- Sempre que uma página é acessada grava-se nela a hora em que foi usada.
- Quando ocorrer uma **page fault** o S.O. percorre as páginas procurando a que foi usada a mais tempo atrás.
- Essa será substituída



# Atividade

- Escolha 2 dos algoritmos mostrados, ou qualquer outro apresentado no livro “Sistemas Operacionais Modernos”
- Implemente estes algoritmos na atividade da aula passada.

**Endereço para entrega: <https://goo.gl/BOeUmE>**