

## Cap. 4 – Gerenciamento de Memória

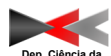
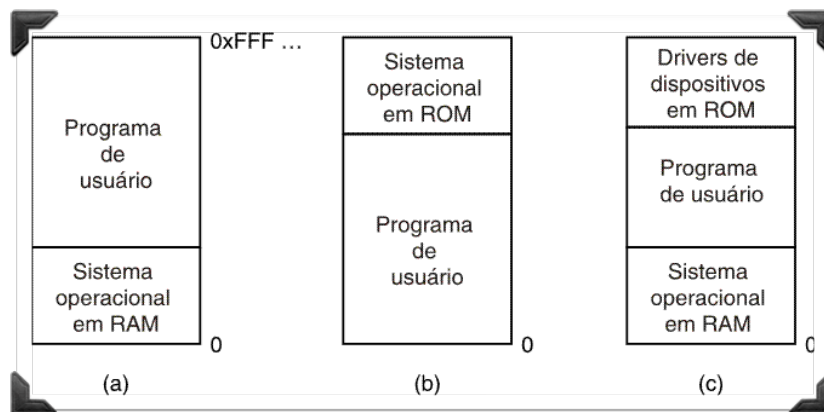
Prof. Marcelo Moreno

[moreno@ice.ufjf.br](mailto:moreno@ice.ufjf.br)



### Gerenciamento Básico de Memória

#### Monoprogramação



### Gerenciamento de Memória

- Todo programador deseja contar com memória:
  - Grande
  - Rápida
  - Não volátil
- Hierarquia de memórias
  - Pequena quantidade de memória rápida, de alto custo - cache
  - Quantidade considerável de memória de velocidade média, custo médio - memória principal
  - Enorme quantidade de armazenamento não-volátil de velocidade e custo baixos - disco
- O gerenciador de memória trata a hierarquia de memórias

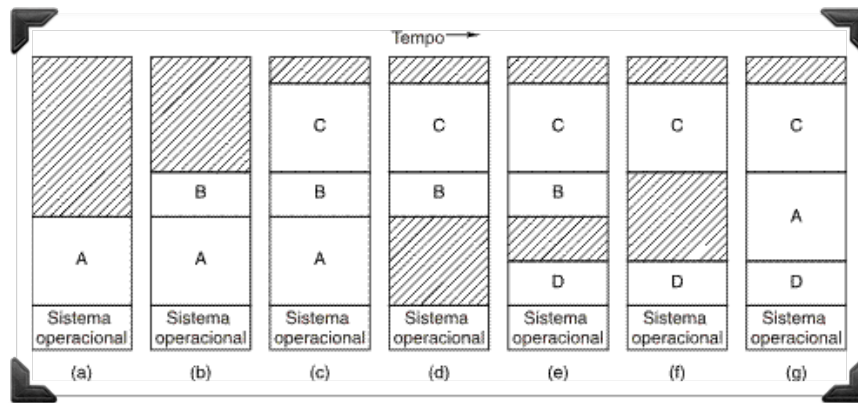


### Relocação e Proteção

- Não se sabe com certeza onde o programa será carregado na memória
  - Localizações de endereços de variáveis e de código de rotinas não podem ser absolutos
- Solução 1: instruções do programa são modificadas segundo o segmento de memória em que ele será carregado
  - Não resolve proteção!
    - Bits de proteção podem ser usados, funcionando como chaves de acesso
- Solução 2: uso de valores base e limite
  - Localizações de endereços são somadas ao valor base antes de serem mapeadas na memória física
  - Localizações de endereços maiores que o limite indicam erro



## Troca de Processos (Swap)



## Algoritmos para Seleção de Segmento

- Algoritmos usados para a escolha do segmento a ser tomado para alocação por um processo
  - First fit**
  - Next Fit**
  - Best fit**
  - Worst fit**
  - Buddy System**

## Seleção de Segmento - First fit

A20	20	108				
A15	20	15	93			
A10	20	15	10	83		
A25	20	15	10	25	58	
D20	20	15	10	25	58	
D10	20	15	10	25	58	
A8	8	12	15	10	25	58
A30	8	12	15	10	25	30 28
D15	8	37			25	30 28
A15	8	15	22		25	30 28

## Seleção de Segmento - Next fit

A8	20	15	10	25	8	50	
A30	20	15	10	25	8	30	20
D15	45			25	8	30	20
A15	45			25	8	30	15 5

## Seleção de Segmento - Best fit

A8	20	15	8	2	25	58	
A30	20	15	8	2	25	30	28
D15	35		8	2	25	30	28
A15	35		8	2	25	30	15 13

## Seleção de Segmento - Worst Fit

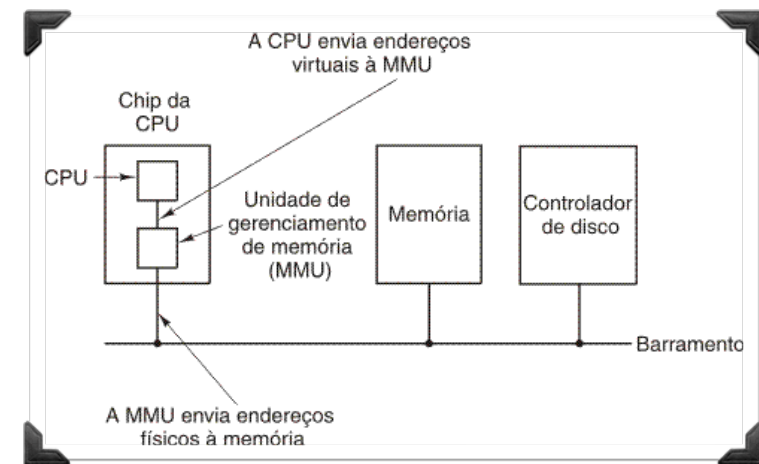
A8	20	15	10	25	8	50	
A30	20	15	10	25	8	30	20
D15	45			25	8	30	20
A15	15	30		25	8	30	20



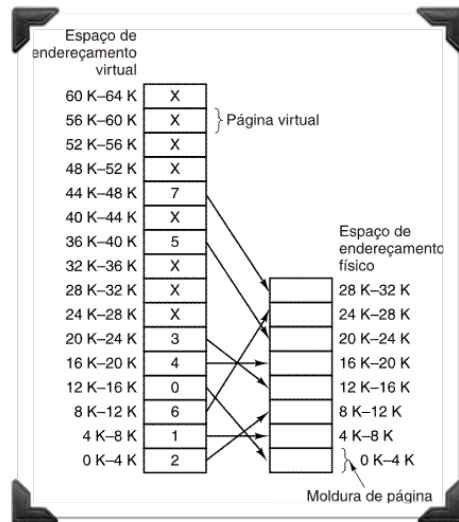
## Alocação Buddy System

	128					
A20	32:20	32		64		
A15	32:20	16:15	16	64		
A10	32:20	16:15	16:10	64		
A25	32:20	16:15	16:10	32:25	32	
D20	32	16:15	16:10	32:25	32	
D10	32	16:15	16	32:25	32	
A8	32	16:15	8:8	8	32:25	32
A30	32:30	16:15	8:8	8	32:25	32
D15	32:30	16	8:8	8	32:25	32
A15	32:30	16:15	8:8	8	32:25	32
D8	32:30	16:15	16	32:25	32	
D30	32	16:15	16	32:25	32	
D15	64			32:25	32	

## Memória Virtual



## Paginação

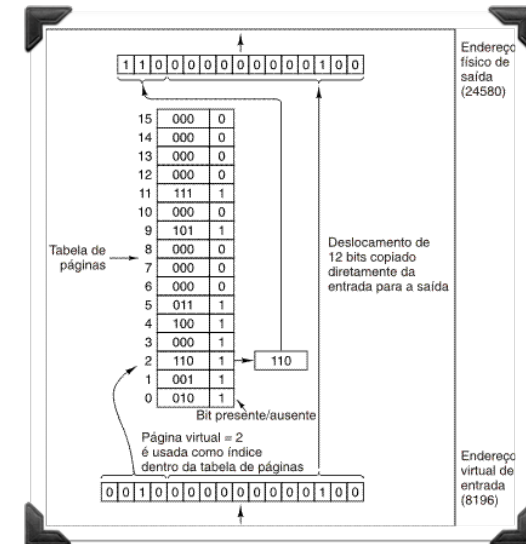


## Envolvimento S.O e Paginação

- Quatro circunstâncias de envolvimento:
  - Criação de processo**
    - determina tamanho do programa e cria tabela de páginas
  - Execução de processo**
    - MMU é reinicializada com tabela para novo processo
  - Ocorrência de falta de página**
    - determina endereço virtual que causou a falta
    - descarta, se necessário, uma página antiga
    - carrega página requisitada para a memória
  - Terminação de processo**
    - Libera tabela de páginas, frames de páginas, e espaço em disco usado



## Paginação



## Algoritmos para Substituição de Páginas

- Algoritmos usados para a escolha de qual página será removida da memória principal para dar lugar a outra
  - Ótimo**
  - Não usada recentemente (NRU)**
  - Primeira a Entrar, Primeira a Sair (FIFO)**
  - Segunda Chance (SC)**
  - Relógio (Clock)**
  - Menos recentemente usada (LRU)**
  - Conjunto de Trabalho (Working set)**
  - WSClock**



## Substituição de Páginas: Ótimo

- **Algoritmo Ótimo:**
  - Rotular cada página com quantas instruções faltam para que ela seja realmente usada
  - Substituir a página que será necessária o mais a frente possível, ou seja, a que possui maior rótulo
  - Adia a ocorrência da falta para o mais tarde possível
  - Ótimo, mas não realizável
- **Próximo uso pode ser estimado**
  - Registro do uso da página em execuções anteriores do processo
- **Muito bom para comparações. Inviável na prática.**



## Substituição de Páginas: NRU

- **Não Usada Recentemente (NRU)**
  - Cada página tem os bits Referenciada (R) e Modificada (M)
  - Bits são colocados em 1 sempre que a página é referenciada (R) e modificada (M)
- **As páginas são classificadas**
  - Classe 0: não referenciada, não modificada
  - Classe 1: não referenciada, modificada
  - Classe 2: referenciada, não modificada
  - Classe 3: referenciada, modificada
- **NRU remove página aleatoriamente da classe de ordem mais baixa que não esteja vazia**



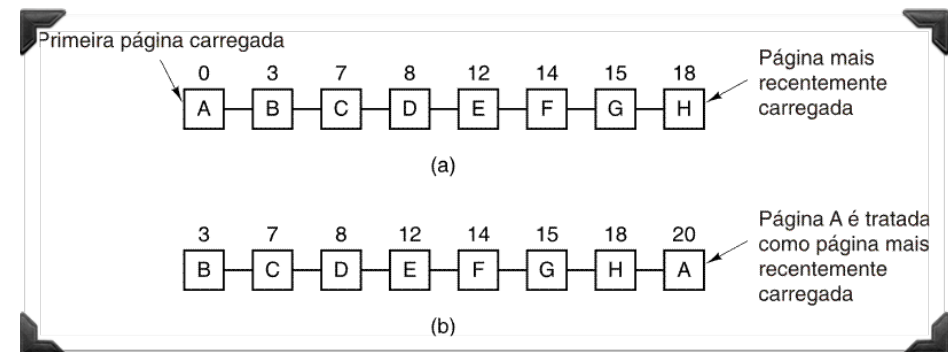
## Substituição de Páginas: FIFO

- **Primeira a Entrar, Primeira a Sair (FIFO)**
  - **Mantém uma lista encadeada de todas as páginas**
    - Página mais antiga na cabeça da lista
    - Página que chegou por último na memória no final da lista
  - **Na ocorrência de falta de página**
    - Página na cabeça da lista é removida
    - Nova página adicionada no final da lista
  - **Desvantagem**
    - Página há mais tempo na memória pode ser usada com muita frequência



## Substituição de Páginas: SC

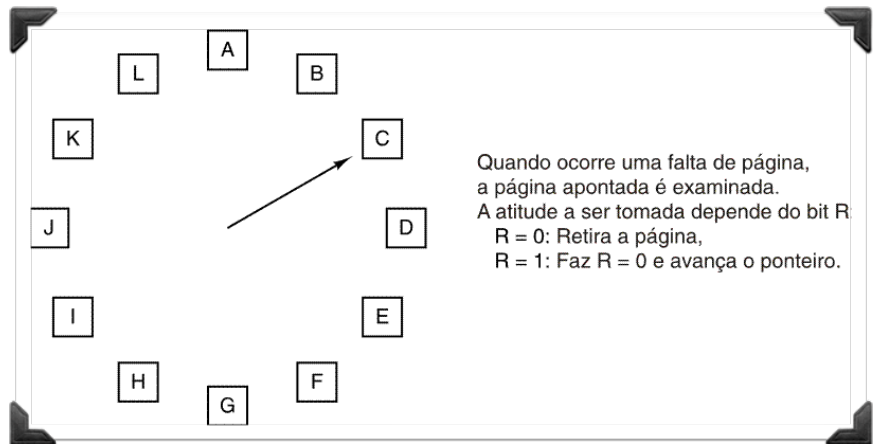
- **Segunda Chance**
  - **Levar em conta o bit R da página mais antiga (cabeça)**



## Substituição de Páginas: Relógio

### Relógio (Clock)

- Manter as páginas em uma lista circular



## Substituição de Páginas: LRU

### Menos Recentemente Usada (LRU)

- Assume que páginas usadas recentemente logo serão usadas novamente
- Retira da memória página que há mais tempo não é usada
- Uma lista encadeada de páginas deve ser mantida
  - A página mais recentemente usada no início da lista, a menos usada no final da lista
  - Atualização da lista à cada referência à memória
- Alternativamente, manter contador em cada entrada da tabela de página, com auxílio de hardware
  - Escolhe página com contador de menor valor
  - Zera o contador periodicamente

## Substituição de Páginas: LRU

### Simulação em Software

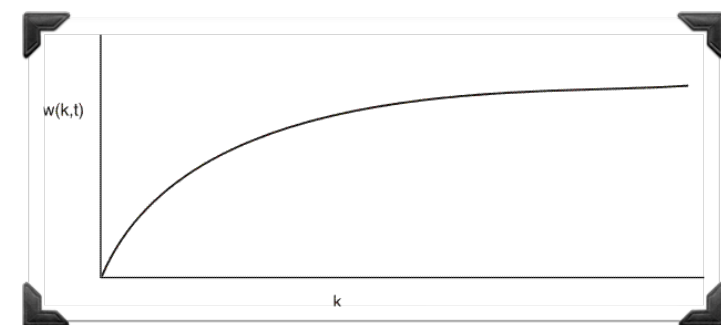
- Algoritmo Não Usada Frequentemente (NUF)
- Com Aging

Bits R das páginas 0-5, interrupção de relógio 0	Bits R das páginas 0-5, interrupção de relógio 1	Bits R das páginas 0-5, interrupção de relógio 2	Bits R das páginas 0-5, interrupção de relógio 3	Bits R das páginas 0-5, interrupção de relógio 4
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
Página				
0	10000000	11000000	11100000	01110000
1	00000000	10000000	01100000	10110000
2	10000000	01000000	00010000	10001000
3	00000000	00000000	10000000	00100000
4	10000000	11000000	01100000	01011000
5	10000000	01000000	10100000	00101000
(a)	(b)	(c)	(d)	(e)

## Substituição de Páginas: Conjunto de Trabalho

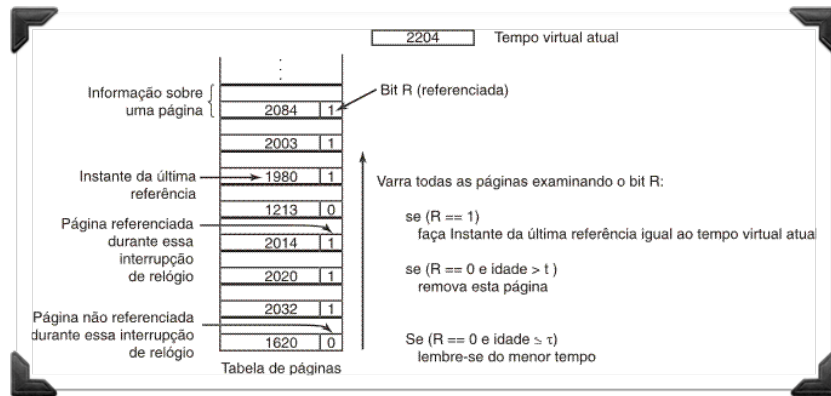
### Conjunto de Trabalho (Working Set)

- O conjunto de trabalho é o conjunto de páginas usadas pelas  $k$  referências mais recentes à memória
- $w(k, t)$  é o tamanho do conjunto de trabalho no instante  $t$

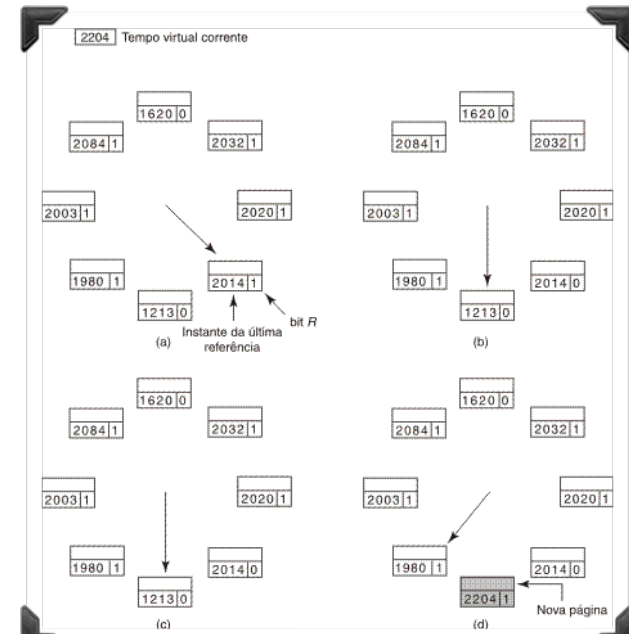


## Substituição de Páginas: Conjunto de Trabalho

### Conjunto de Trabalho (Working Set)



## Substituição de Páginas: WSClock



## Substituição de Páginas: Comparação

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NUR (não usada recentemente)	Muito rudimentar
FIFO (primeira a entrar, primeira a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
MRU (menos recentemente usada)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não frequentemente usada)	Aproximação bastante rudimentar do MRU
Envelhecimento ( <i>aging</i> )	Algoritmo bastante eficiente que se aproxima bem do MRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

## Tratamento de Faltas de Página

1. Hardware desvia a execução para o núcleo
2. Salva conteúdo de registradores e outras informações voláteis
3. SO determina a página virtual necessária
4. SO checa validade de endereço, busca moldura de página
5. Se moldura de página selecionada foi modificada (suja), salvá-la em disco
5. SO busca em disco página virtual referenciada
7. Tabela de páginas é atualizada
3. Estado da instrução que causou falta de página é recuperado
9. Processo que causou falta de página é escalonado para executar
10. Programa continua



## Exercícios

- Dado um sistema de memória virtual com páginas de 4K palavras, e como tabela de páginas:

Pág. Virtual (12 bits)	Frame página (8 bits)
0x000	0x04
0x001	0x01
0x002	0x43
0x003	0x14
...	...

Pede-se:

- O endereço físico do endereço virtual 0x1ABC.
- O endereço virtual do endereço físico 0x4310.
- O maior tamanho de processo possível.
- O tamanho máximo da memória física.

Obs.: 0x é uma notação para números hexadecimais



## Exercícios

- Dado um sistema com memória virtual que utiliza o algoritmo de substituição de páginas “Menos Recentemente Usada” e que possua 4 frames de página. Mostre passo-a-passo a situação da memória física frente à seguinte sequência de acesso a páginas virtuais: 0, 1, 2, 3, 0, 4, 5, 0, 1, 2, 1, 3, 1, 0. Marque para cada passo se houve falha (MISS) ou acerto (HIT) no acesso à página. Responda quantas falhas e acertos ocorrem no total.