MC514–Sistemas Operacionais: Teoria e Prática

1s2008

Gerenciamento de Memória - 1

Programas muito grandes

O que fazer se um programa for muito grande para caber na memória?

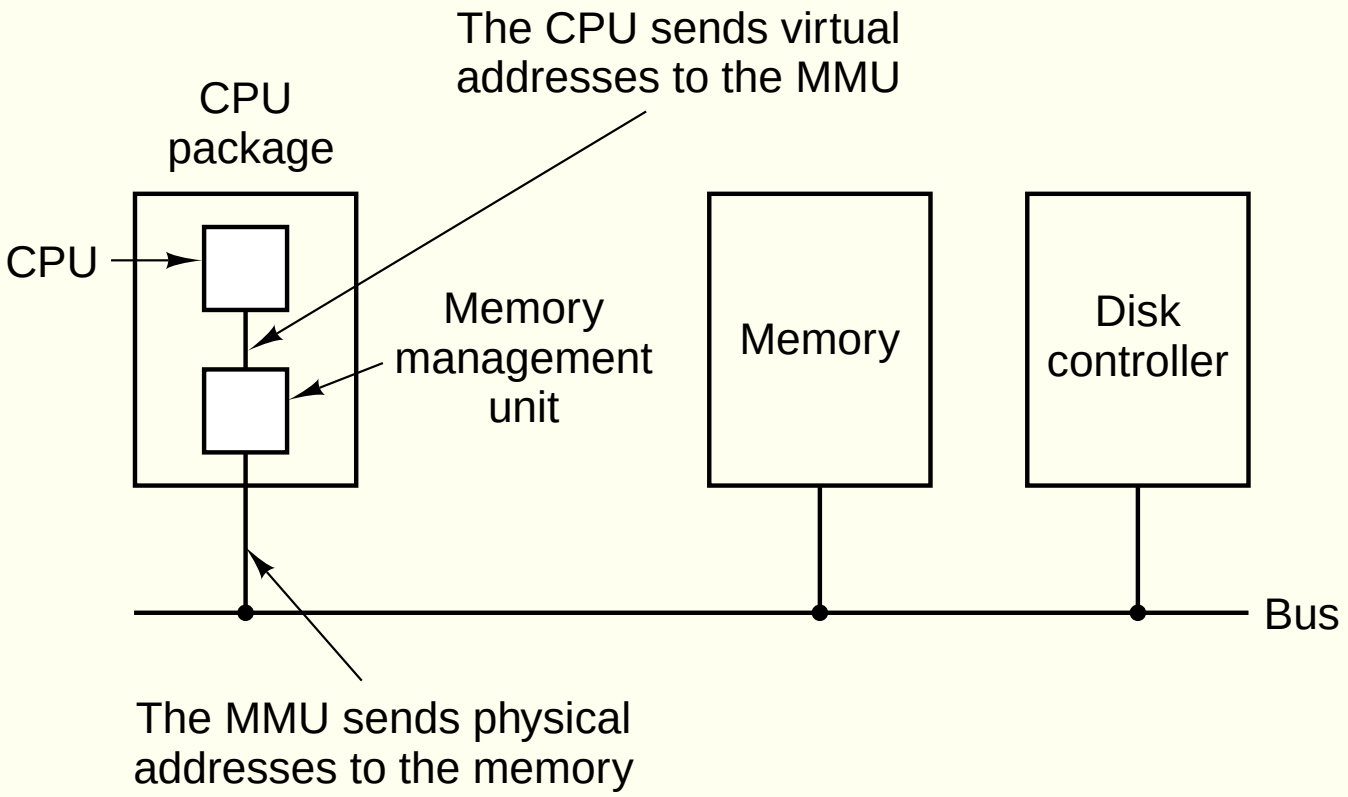
Overlays

```
dados d1, d2, d3, d4, d5;

f1();    g1();   h1();
f2();    g2();   h2();
f3();    g3();   h3();

main() {
  fase_1();  /* funçoes f e dados d1, d2, d3 */
  fase_2();  /* funçoes g e dados d1, d2, d4 */
  fase_3();  /* funçoes h e dados d1, d2, d5 */
  }
```

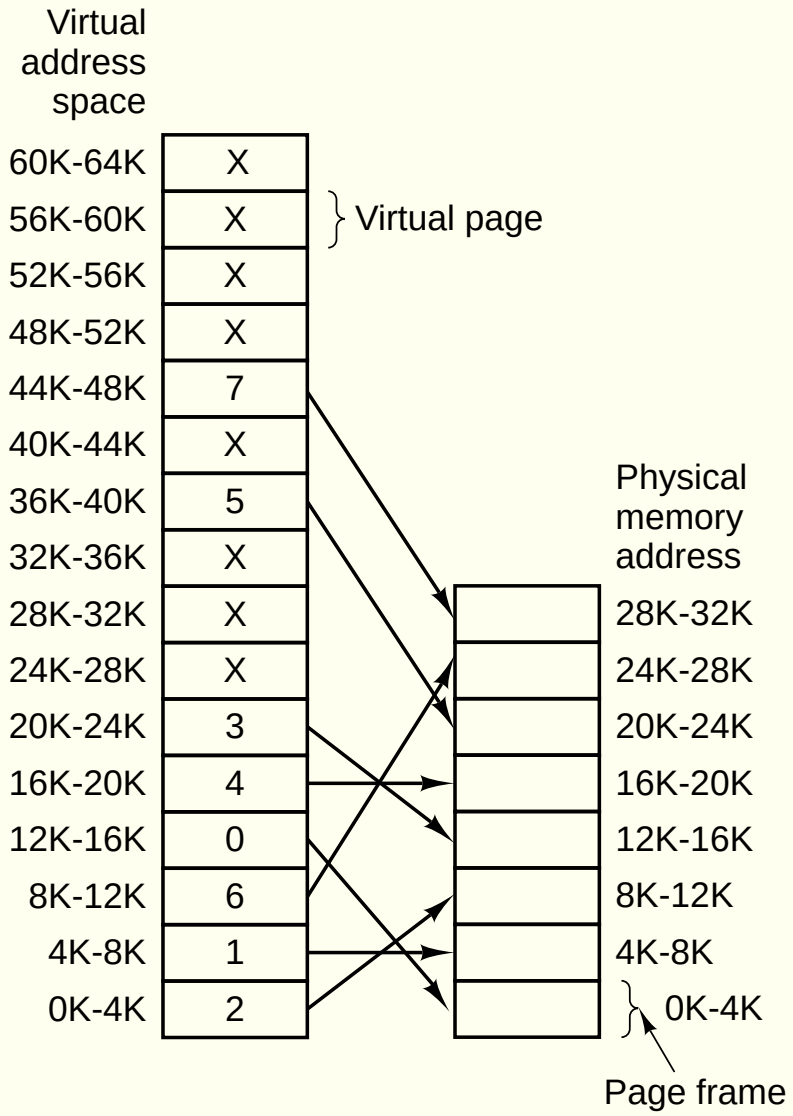
Memória Virtual



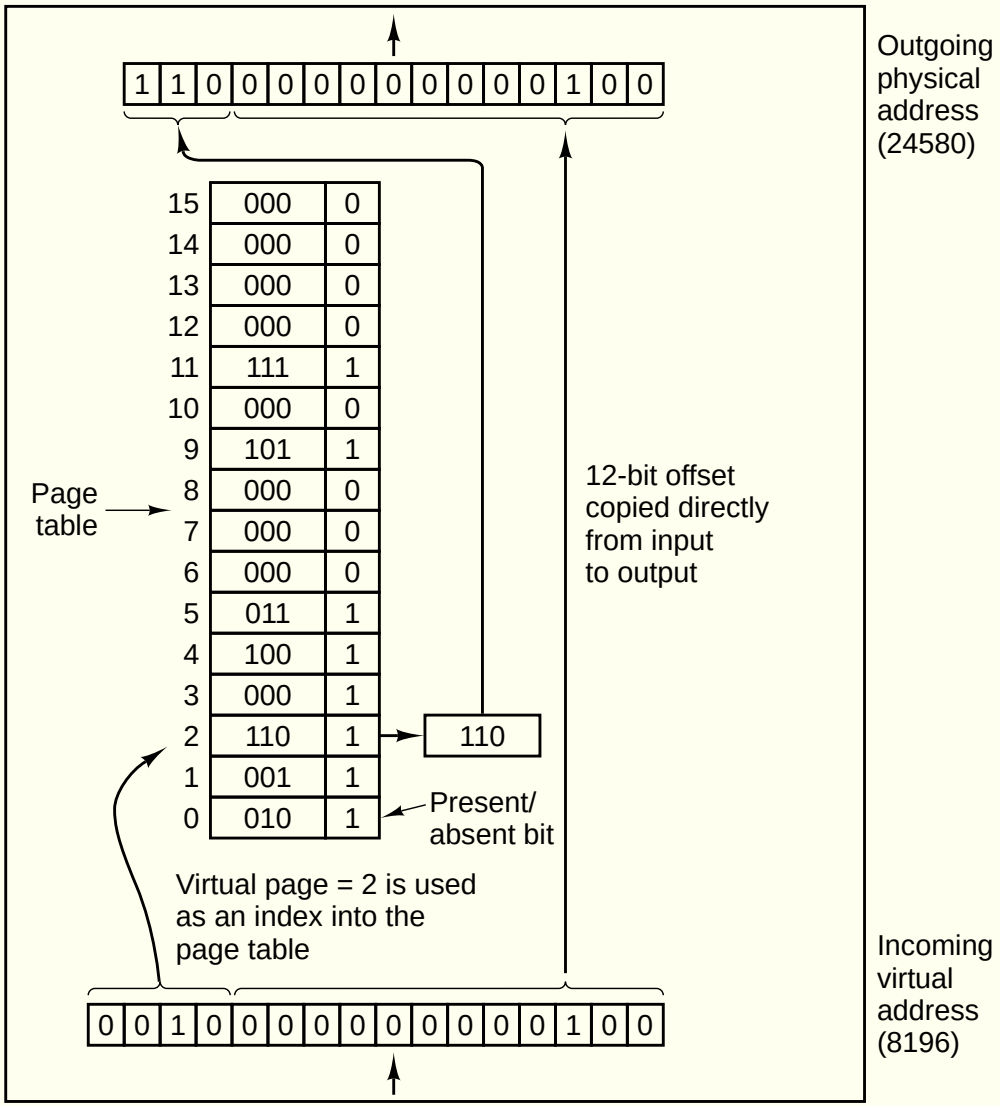
Tanenbaum: Figura 4.9

Memory Management Unit (MMU)

Paginação


Virtual address space
60K-64K
56K-60K
52K-56K
48K-52K
44K-48K
40K-44K
36K-40K
32K-36K
28K-32K
24K-28K
20K-24K
16K-20K
12K-16K
8K-12K
4K-8K
0K-4K
X
X
X
X
7
X
5
X
X
X
3
4
0
6
1
2
Virtual page
Physical memory address
28K-32K
24K-28K
20K-24K
16K-20K
12K-16K
8K-12K
4K-8K
0K-4K
Page frame

Mapeamento dos endereços



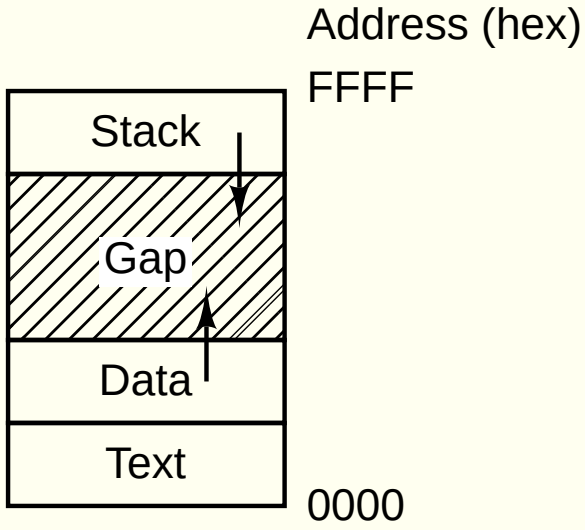
Paginação - Exemplo

- 32 bits de endereço
- páginas de 4k
- 20 primeiros bits indicam a página
- 12 últimos bits indicam o deslocamento dentro da página

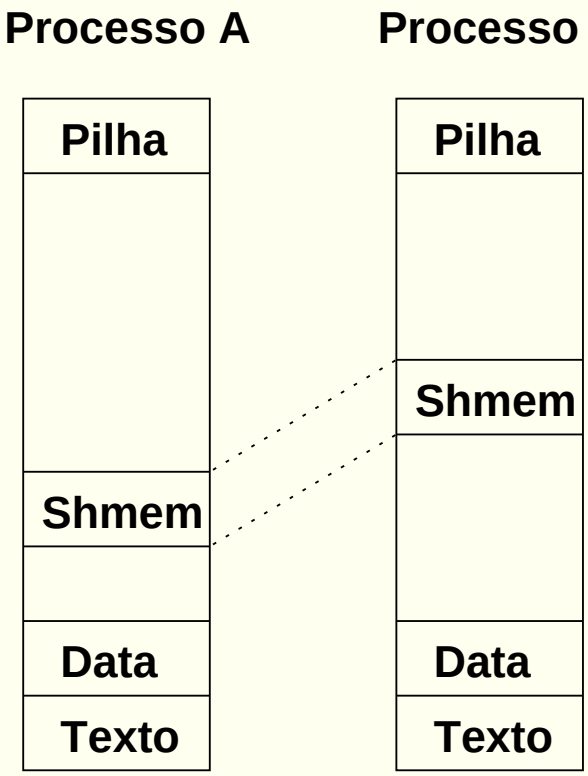
- Veja o código pagesize.c

Espaço de endereçamento

- Apenas as páginas ocupadas precisam ser mapeadas
- Veja o código sbrk.c



Memória compartilhada



Memória compartilhada

```
int shmget(key_t key, size_t size, int shmflg);
void *shmat(int shmid,
            const void *shmaddr, int shmflg);
```

- Veja os exemplos: sh1.c sh2.c sh_fork.c sh_server.c e sh_client.c

Substituição de páginas

- Veja o código: pag1.c

Algoritmo ótimo:

- Baseado no uso futuro de uma página
- Impossível de ser implementado
- Pode ser simulado (segunda execução do mesmo processo com a mesma entrada)
- Útil para medidas de desempenho

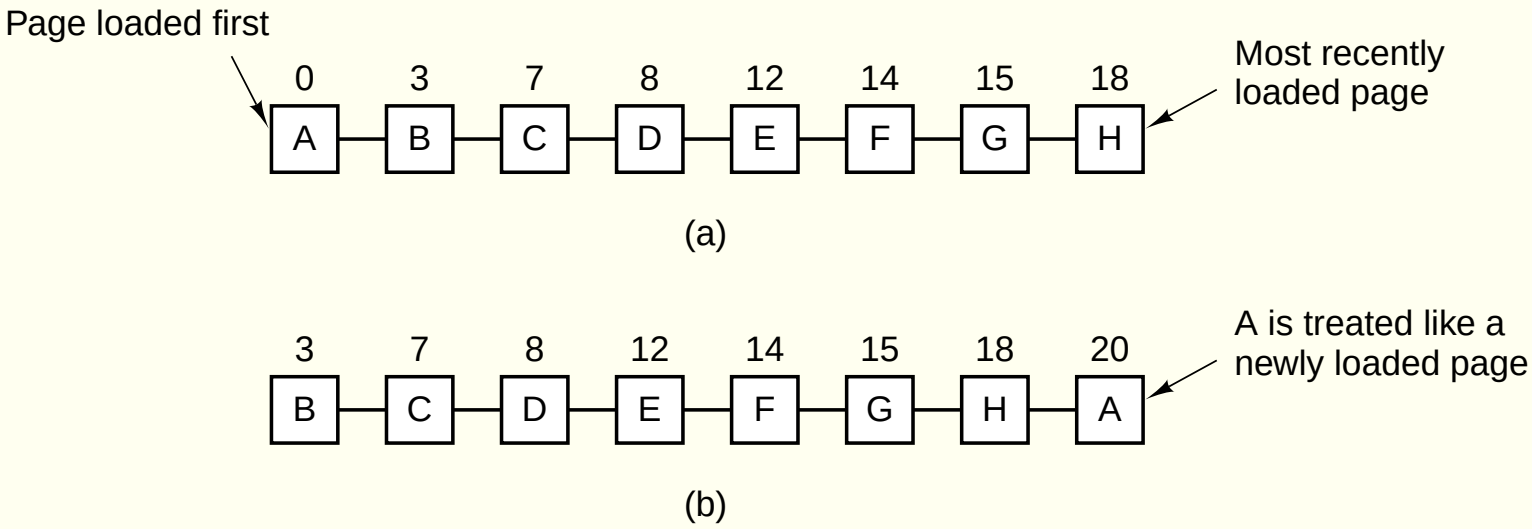
Não usada recentemente

- Classe 0: não referenciada, não modificada
- Classe 1: não referenciada, mas modificada
- Classe 2: referenciada, mas não modificada
- Classe 3: referenciada e modificada

First In, First Out

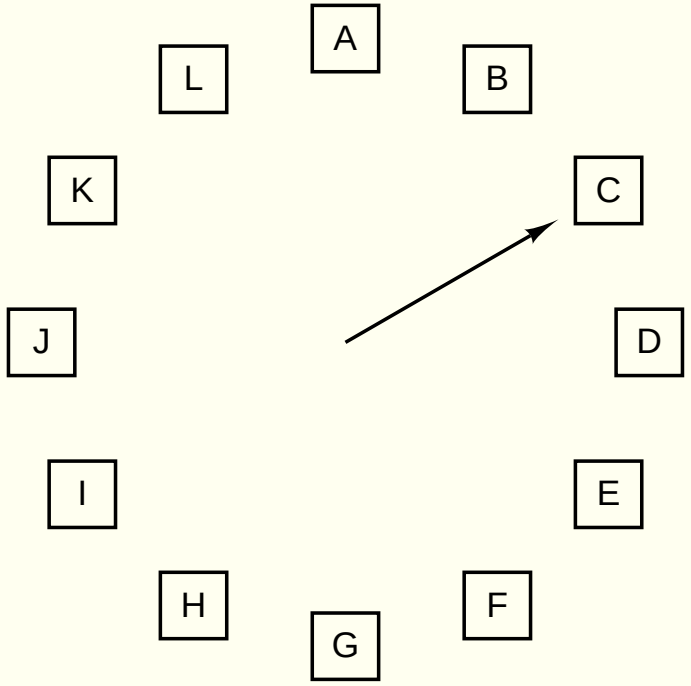
- Simplementes coloca as páginas em uma fila
- Pode remover páginas importantes

Segunda chance



- Se o bit R == 0, a página é substituída, senão
- bit R é limpo e a página é colocada no final da fila

Relógio



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

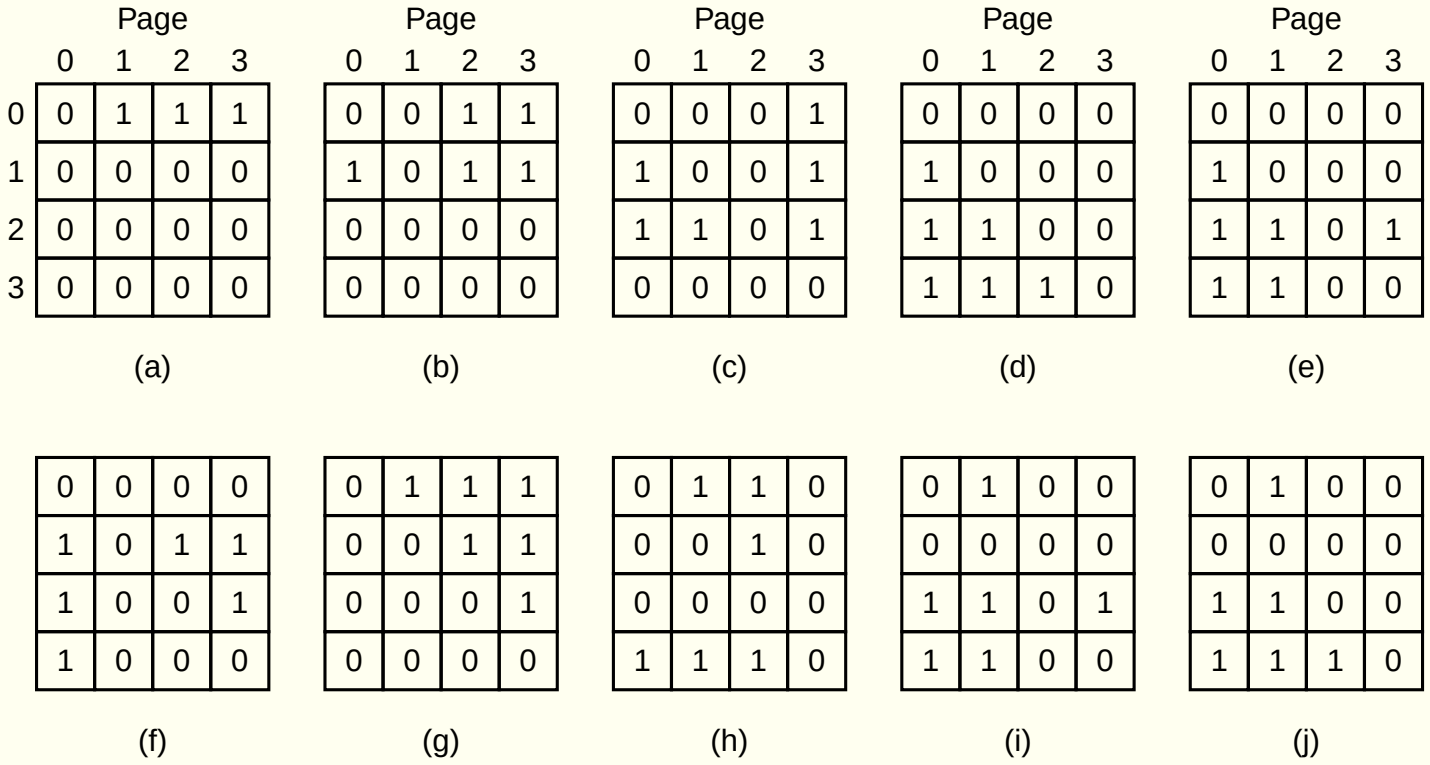
R = 1: Clear R and advance hand

- Implementação circular da segunda chance

Uso recente

- LRU (Least Recently Used)
- Implementação utilizando lista ligada
- Implementação em hardware com contador
  - incrementado a cada instrução
  - entrada na tabela deve armazenar o contador
- Implementação com matriz n x n

LRU em hardware



Acessos: 0 1 2 3 2 1 0 3 2 3

Simulando LRU em software

Página não usada frequentemente

- Contador de uso para cada página (soma o bit R a cada clock tick)
- Não esquece nada...
- Considere um compilador baseado em passos

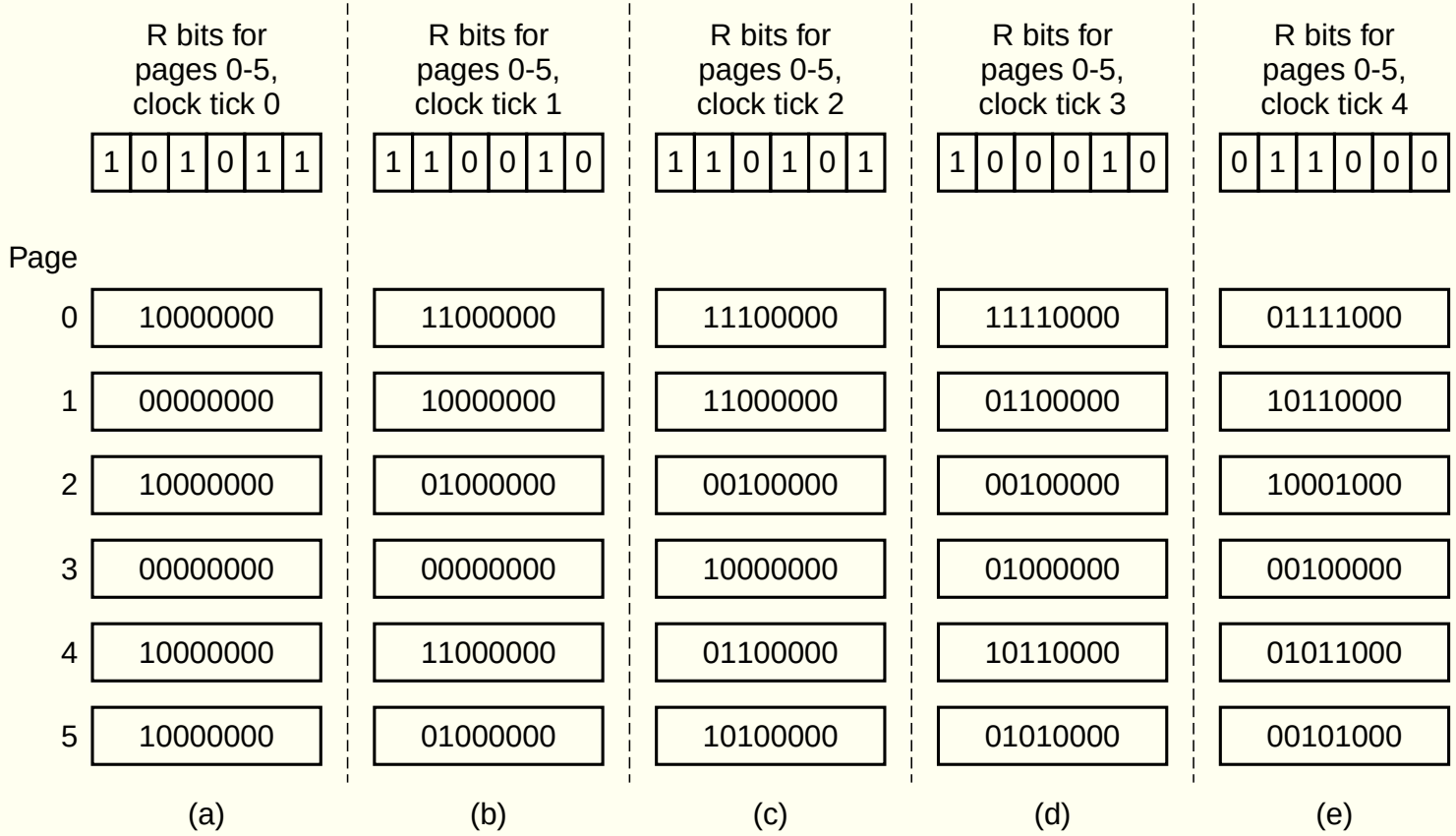
Não usada frequentemente

Aging

- O contador é deslocado à direita
- Bit R é adicionado à esquerda

```
          1 0 1 0 1 1                 1 0 1 1 0 1
>> 1        1 0 1 0 1      >> 1        1 0 1 1 0
+  1     1 1 0 1 0 1       +  0     0 1 0 1 1 0
```

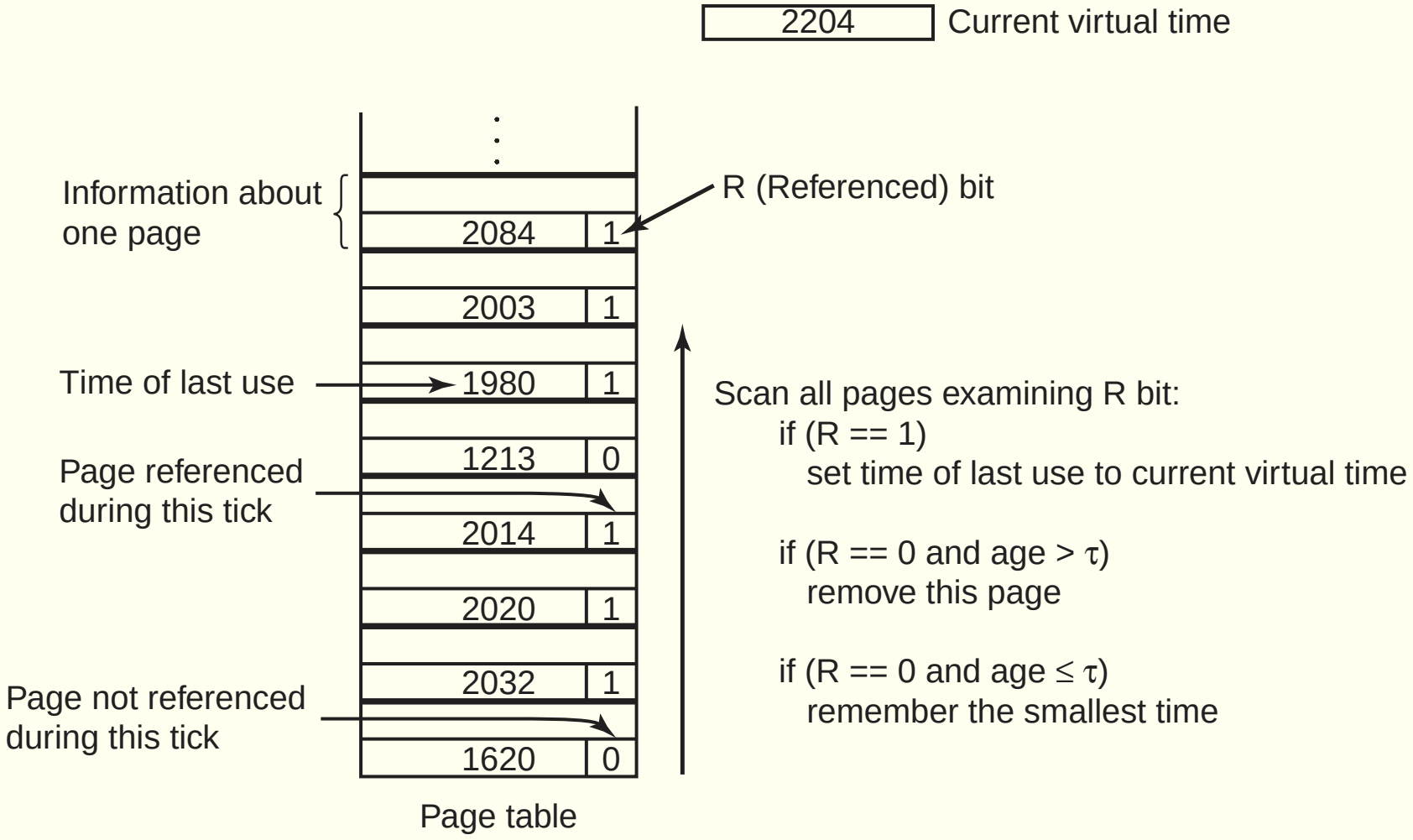
Não usada frequentemente



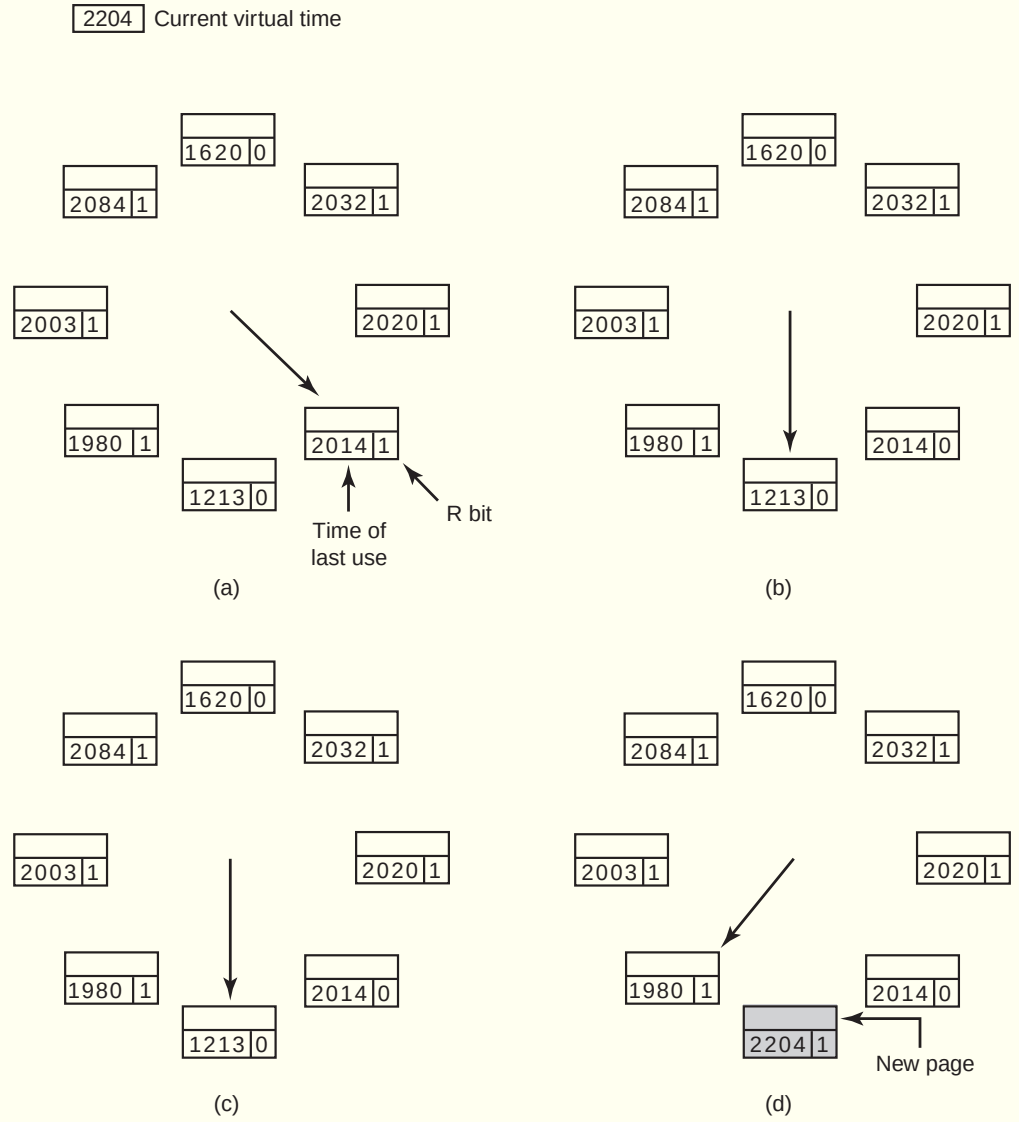
Working Set $w(k, t)$

- Conjunto de páginas utilizadas nas últimas k referências em relação ao instante t.
- Paginação sob demanda
- Prepaging
- Implementação exata é muito cara

Working Set


2204
Current virtual time
Information about one page
R (Referenced) bit
2084
1
2003
1
Time of last use
1980
1
1213
0
Page referenced during this tick
2014
1
2020
1
2032
1
Page not referenced during this tick
1620
0
Page table
Scan all pages examining R bit:
if (R == 1)
set time of last use to current virtual time
if (R == 0 and age > τ)
remove this page
if (R == 0 and age ≤ τ)
remember the smallest time

WSClock


2204 Current virtual time
2084 1
1620 0
2032 1
2003 1
2020 1
1980 1
2014 1
1213 0
Time of last use
R bit
(a)
(b)
(c)
(d)
2014 0
2204 1
New page

Resumo

Algorithm	Comment
Optimal	Not implementable, but useful as a benchmark
NRU (Not Recently Used)	Very crude
FIFO (First-In, First-Out)	Might throw out important pages
Second chance	Big improvement over FIFO
Clock	Realistic
LRU (Least Recently Used)	Excellent, but difficult to implement exactly
NFU (Not Frequently Used)	Fairly crude approximation to LRU
Aging	Efficient algorithm that approximates LRU well
Working set	Somewhat expensive to implement
WSClock	Good efficient algorithm