操作系统

Operating system

孔维强

大连理工大学

内容纲要

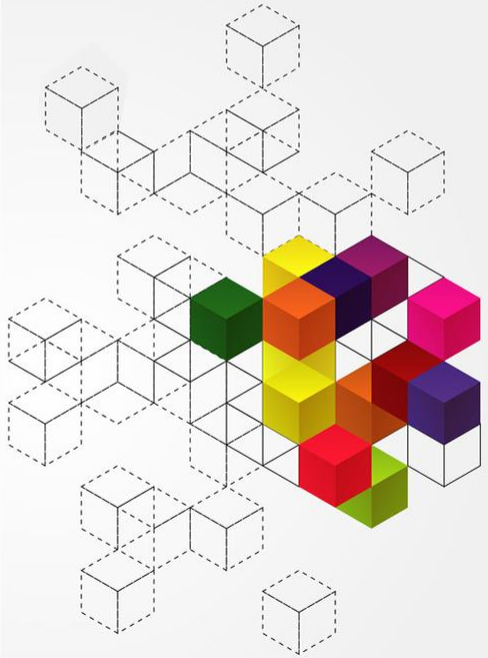
9.3 页置换算法实现

一、 页置换算法实现考虑

二、 近似实现1：附加引用位

三、 近似实现2：时钟算法

四、 近似实现3：增强型二次机会算法



一、页置换算法的实现考虑

- **基于counter的LRU实现**

  每个页表项有一个counter，每次页被访问时，将时钟值拷贝到counter

  当需要置换页时，查找具有最小counter值的页面

- **基于stack的LRU实现**

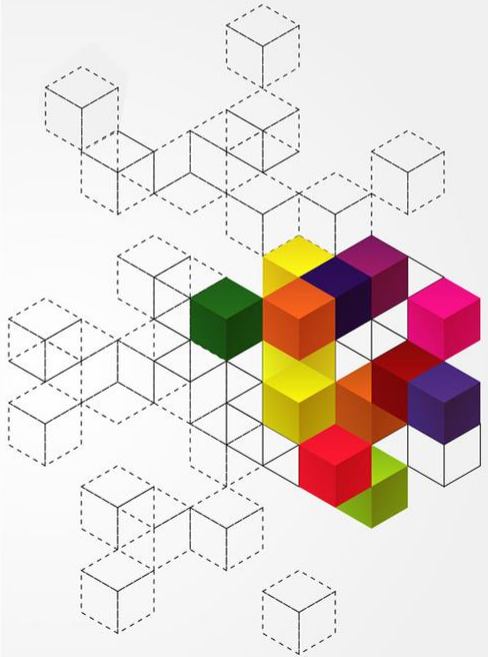
  以双向链（double link）的形式用stack保存页号码

  访问页时

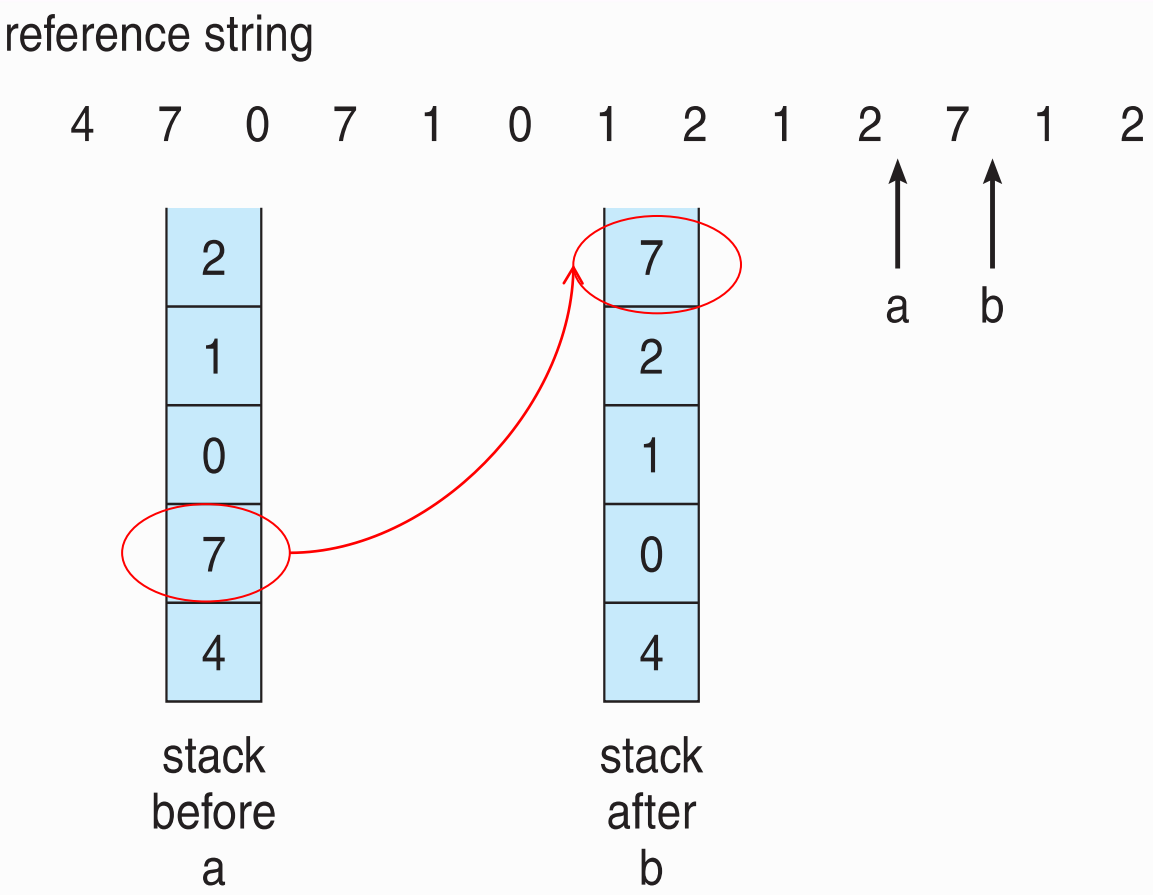
  将该页移至top，需要改变6个指针

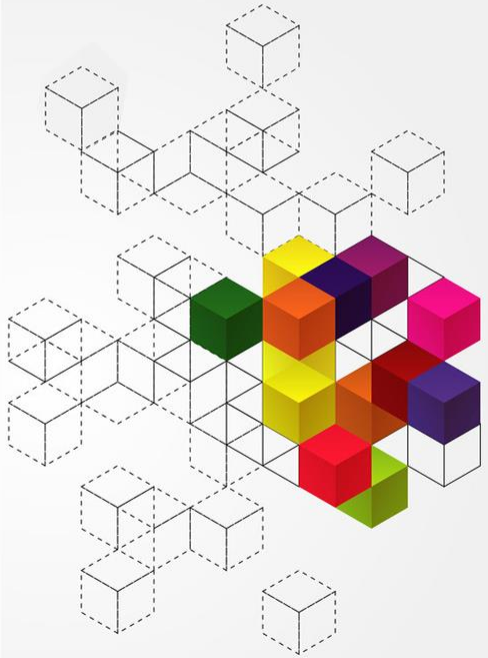
  每次更新的代价更高，但替换时无需查找

LRU和OPT是基于stack的算法，无Belady异常



一、页置换算法的实现考虑



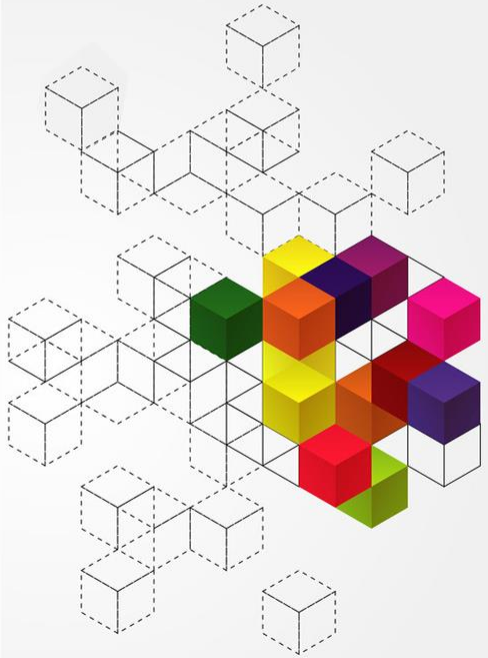


一、页置换算法的实现考虑

- **LRU算法是OPT算法的近似，不会产生Belady异常**
  - 是页置换算法的首选

问题： **使用栈来精确实现LRU算法，可能使得系统内存访问效率下降**

不可接受！



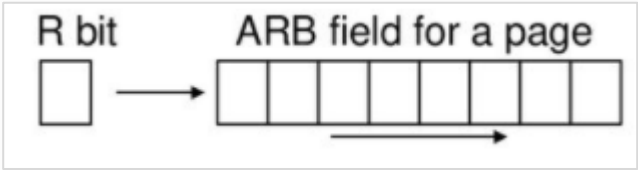
二、近似实现1：附加引用位

LRU精确实现，性能差，不实用

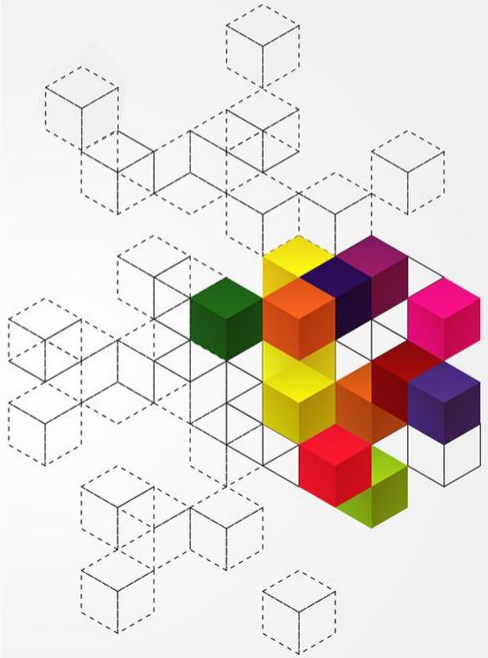
考虑对LRU进行近似实现

策略1：附加引用位

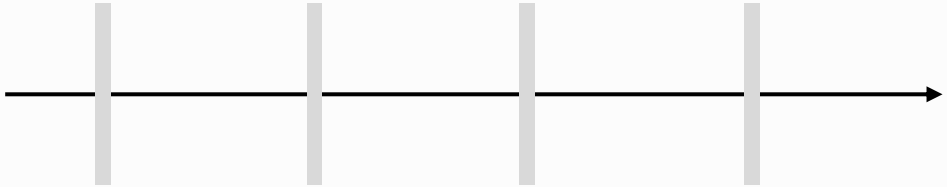
除了页面引用位R外，为每个页面增加额外的引用位
Additional Reference Bit (ARB)



At each timer interrupt, the R bit is shifted from left into the ARB, and the ARB shift accordingly

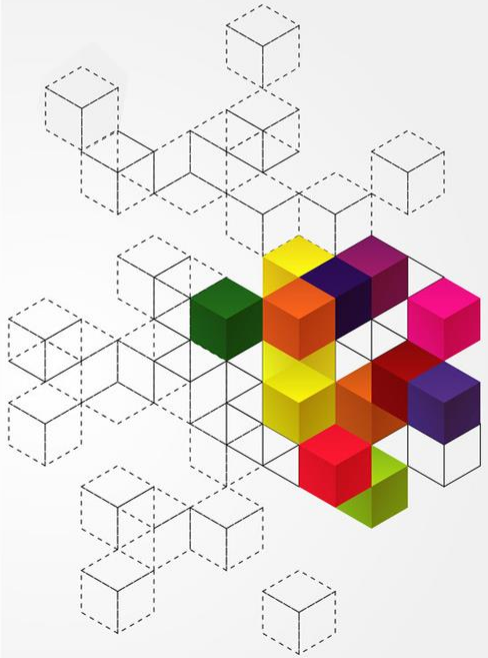


二、近似实现1：附加引用位



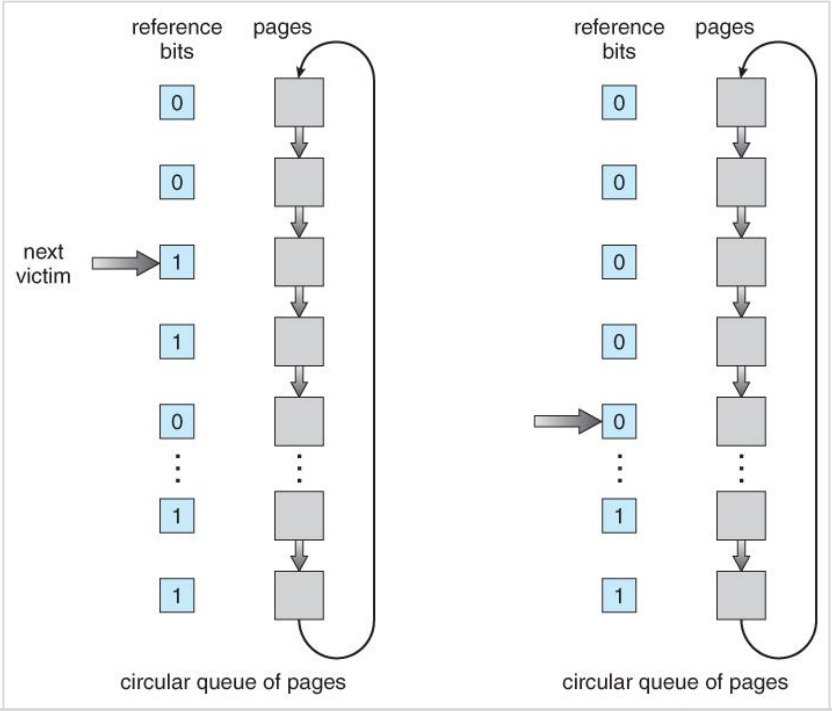
R:		0	1	0	1
ARB:	0000000	0000000	1000000	0100000	1010000

When a page is replaced, select the page with least ARB



三、近似实现2：时钟算法

Second-Chance Page Replacement Algorithm

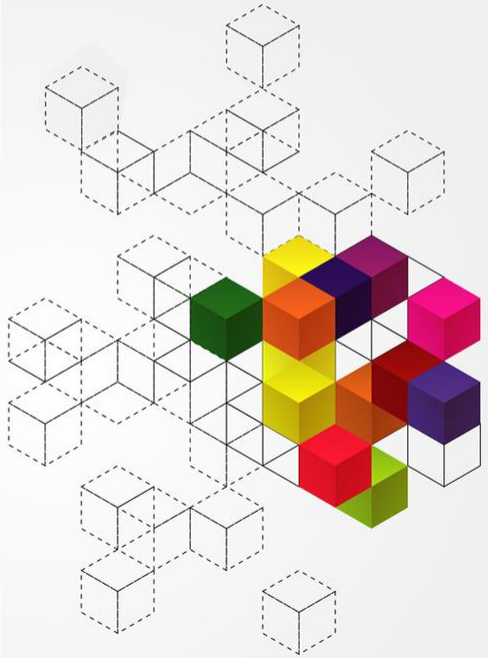


① 页面组织成环

② 每次页置换后，将next victim指针指向被置换的后继页面

③ 如需要进行置换，从当前next victim指向的位置开始，寻找第一个引用为为0的页，搜索途中遇到的引用位为1的改为0

引用位为1的页面，会赢得驻留内存的第2次机会

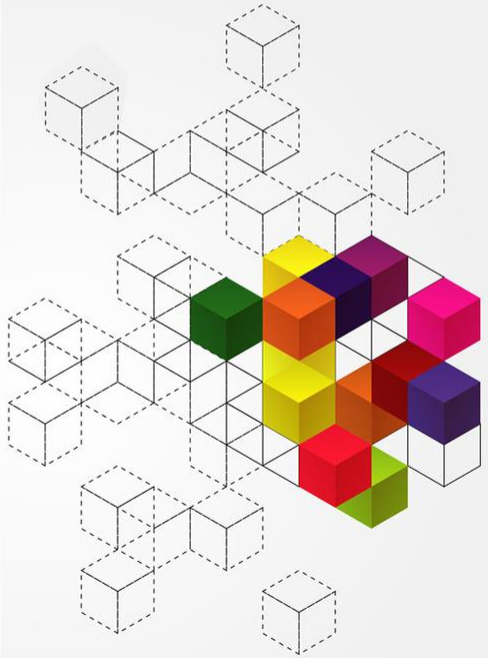


四、近似实现3：增强型二次机会算法

- **原理**：引入引用位(r)和修改位(c)作为有序对

(r,c)	页面描述
(0,0)	最近未使用，也未修改过（置换最佳候选）
(0,1)	最近未使用，但被修改过（需写出）
(1,0)	最近使用过，但未被修改（可能即将使用）
(1,1)	最近使用过，也被修改过

置换时，优先选择rc位为00的页面置换，01，10，11次之



本讲小结

- 页置换算法近似实现方法讨论