



物理内存管理II

SSE202/204: 操作系统原理

苏玉鑫

suyx35@mail.sysu.edu.cn

助教：龙玉丹 单诗雯 毛晨希 沈志轩 郑灿峰 胡伟峰



版权信息

- 部分内容来自：上海交通大学并行与分布式系统研究所操作系统课件
 - <https://ipads.se.sjtu.edu.cn/courses/os/>
- 其它参考资料：
 - 清华大学操作系统公开课
 - <https://open.163.com/newview/movie/courseintro?newurl=ME1NSA351>
 - 介绍标准内容，适合考研
 - 南京大学计算机软件研究所
 - <http://jyywiki.cn/OS/2025/>
 - <https://space.bilibili.com/202224425/channel/detail?sid=192498>
 - 比较有趣



大纲

- 换页机制
- 页替换策略（有私货）
- 工作集模型（有私货）



大纲

- 换页机制
- 页替换策略（有私货）
- 工作集模型（有私货）



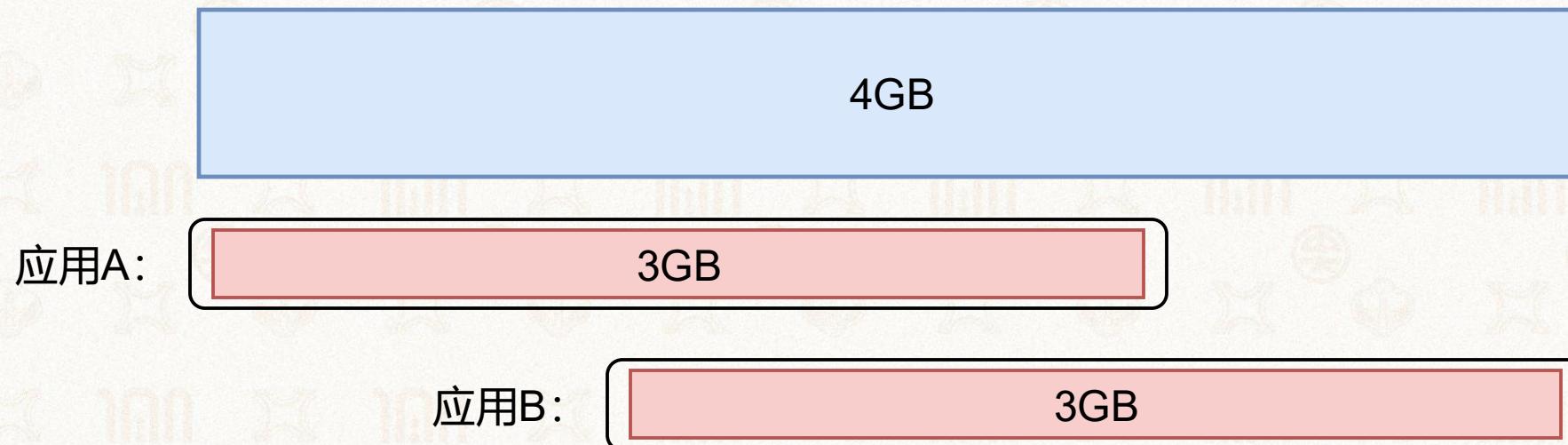
物理内存的超售(Over-commit)和按需分配

➤ 情景1:

- 两个应用程序各自需要使用 3GB 的物理内存
- 整个机器实际上总共只有 4GB 的物理内存

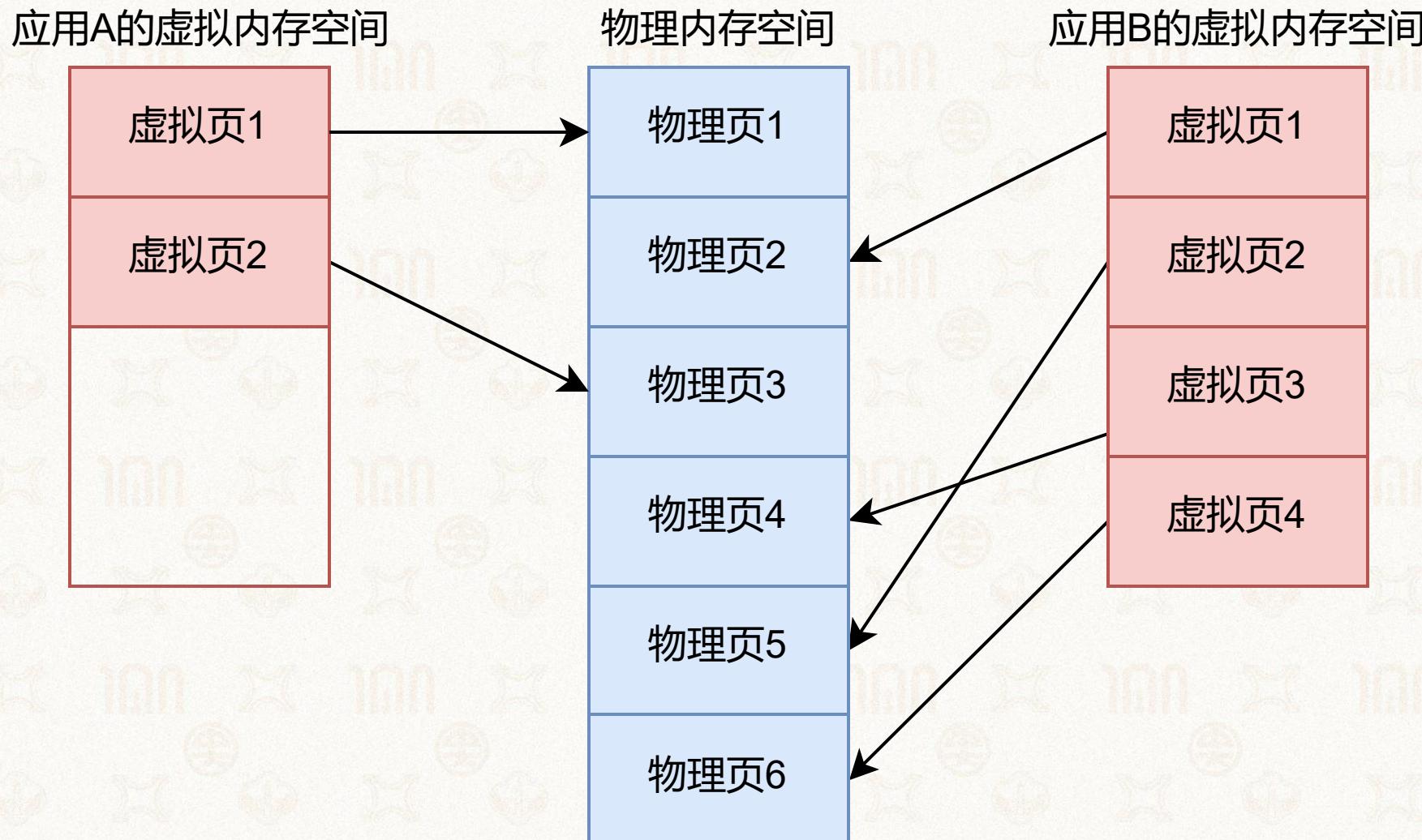
➤ 情景2:

- 一个应用程序申请预先分配足够大的（虚拟）内存
- 实际上其中大部分的虚拟页最终都不会用到



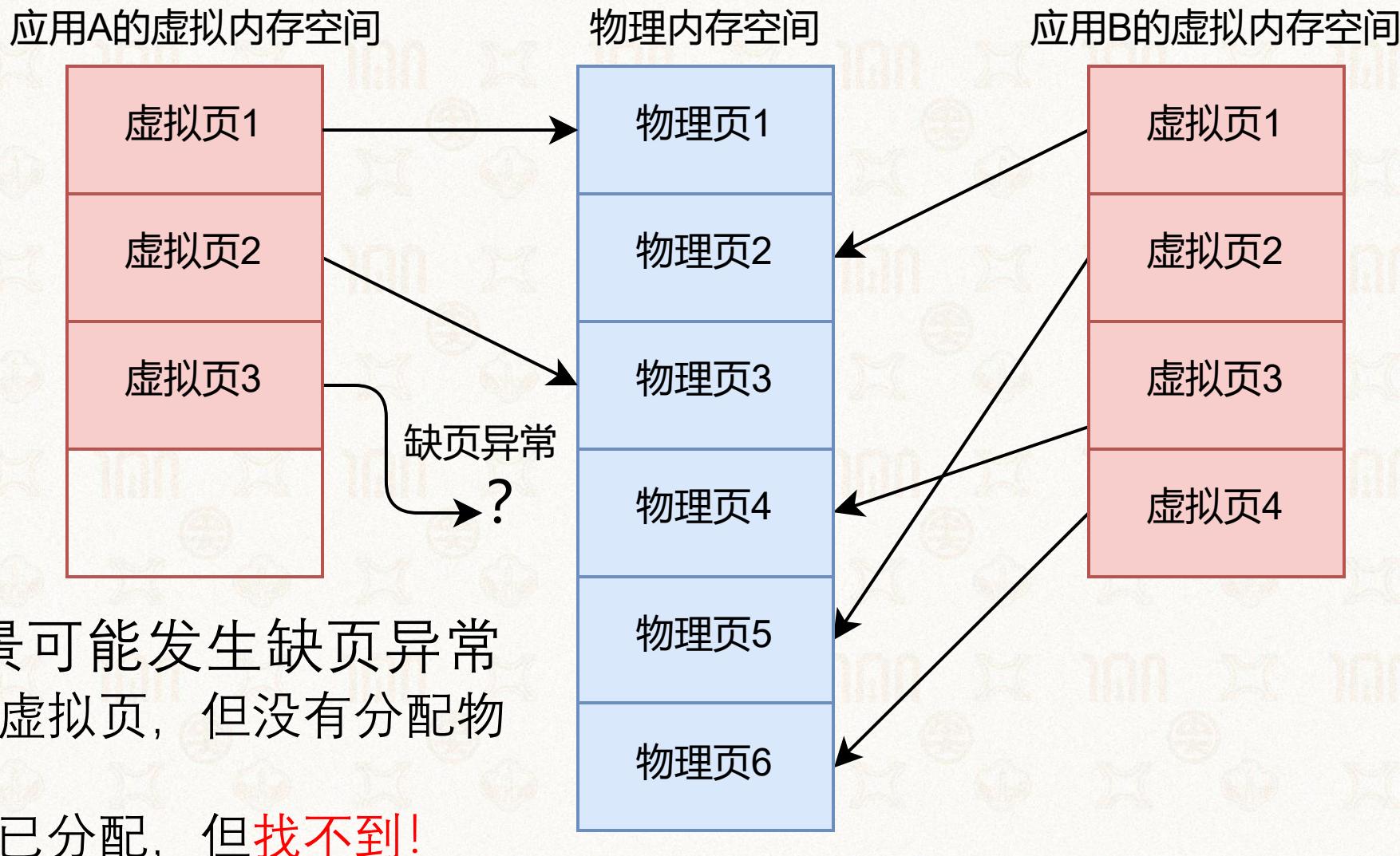


多个应用共同使用相同物理内存空间





缺页异常



- 两种情景可能发生缺页异常
 - 分配了虚拟页，但没有分配物理页
 - 物理页已分配，但找不到！



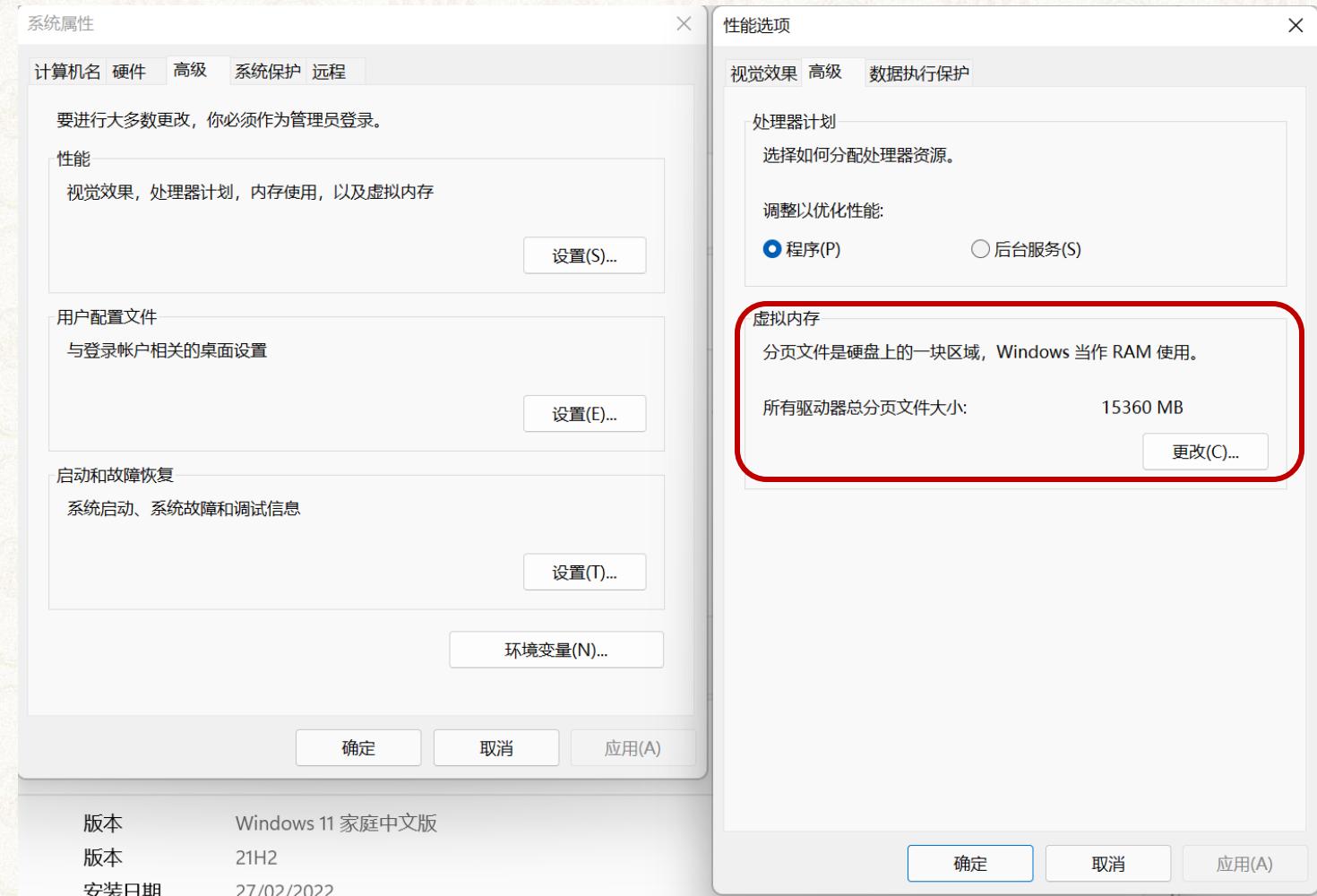
换页机制 (Swapping)

➤ 换页的基本思想

- 将物理内存里面存不下的内容放到磁盘上
- 虚拟内存使用不受物理内存大小限制

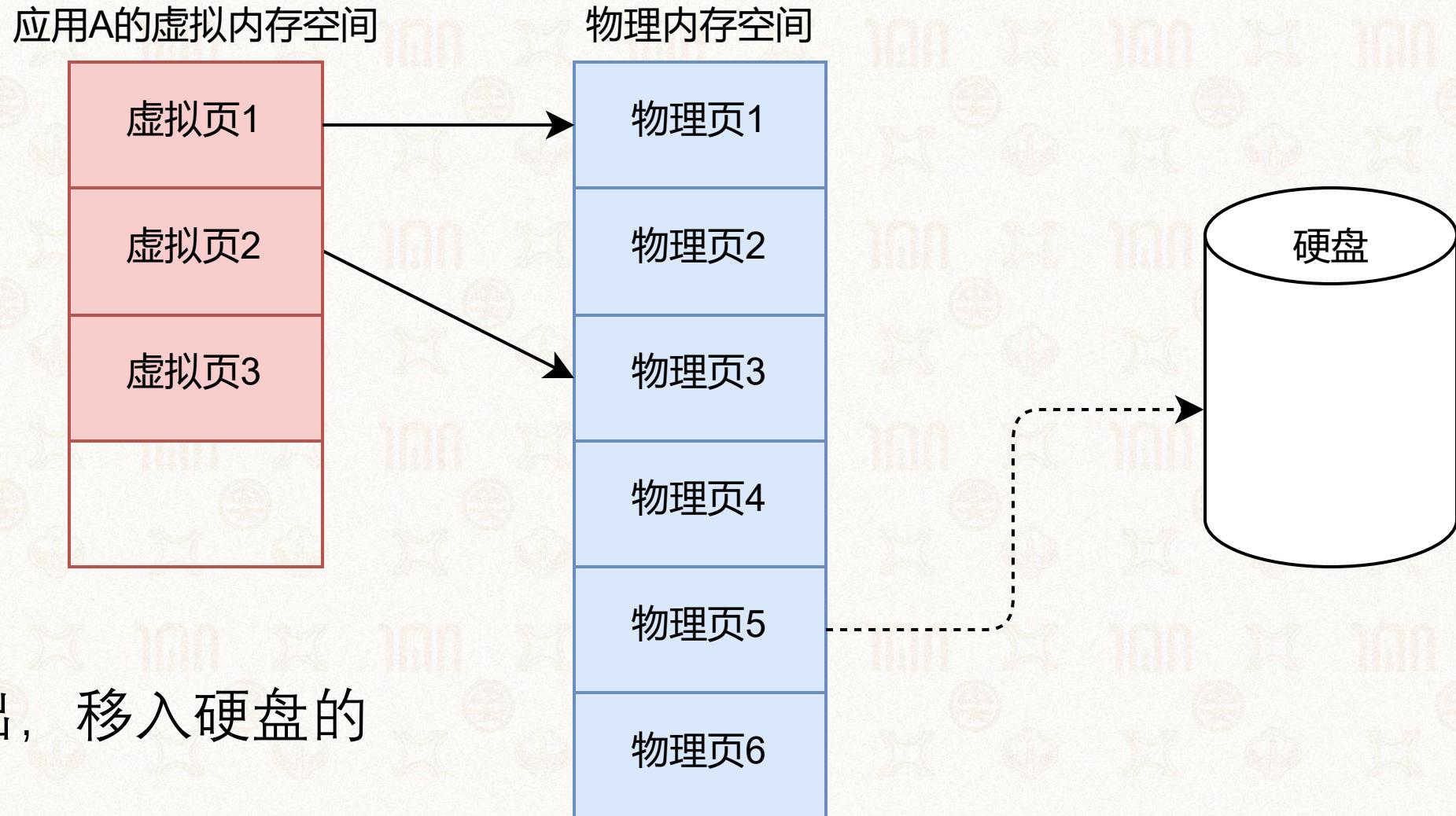
➤ 如何实现

- 磁盘上划分专门的Swap分区
- 在处理缺页异常时，触发物理内存页的换入换出





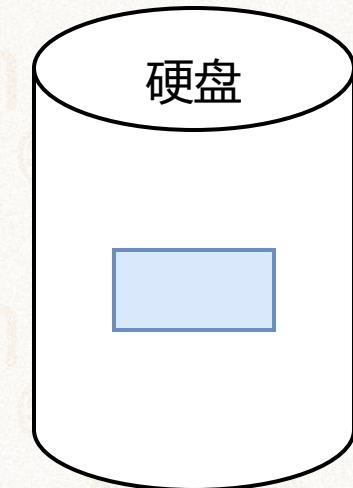
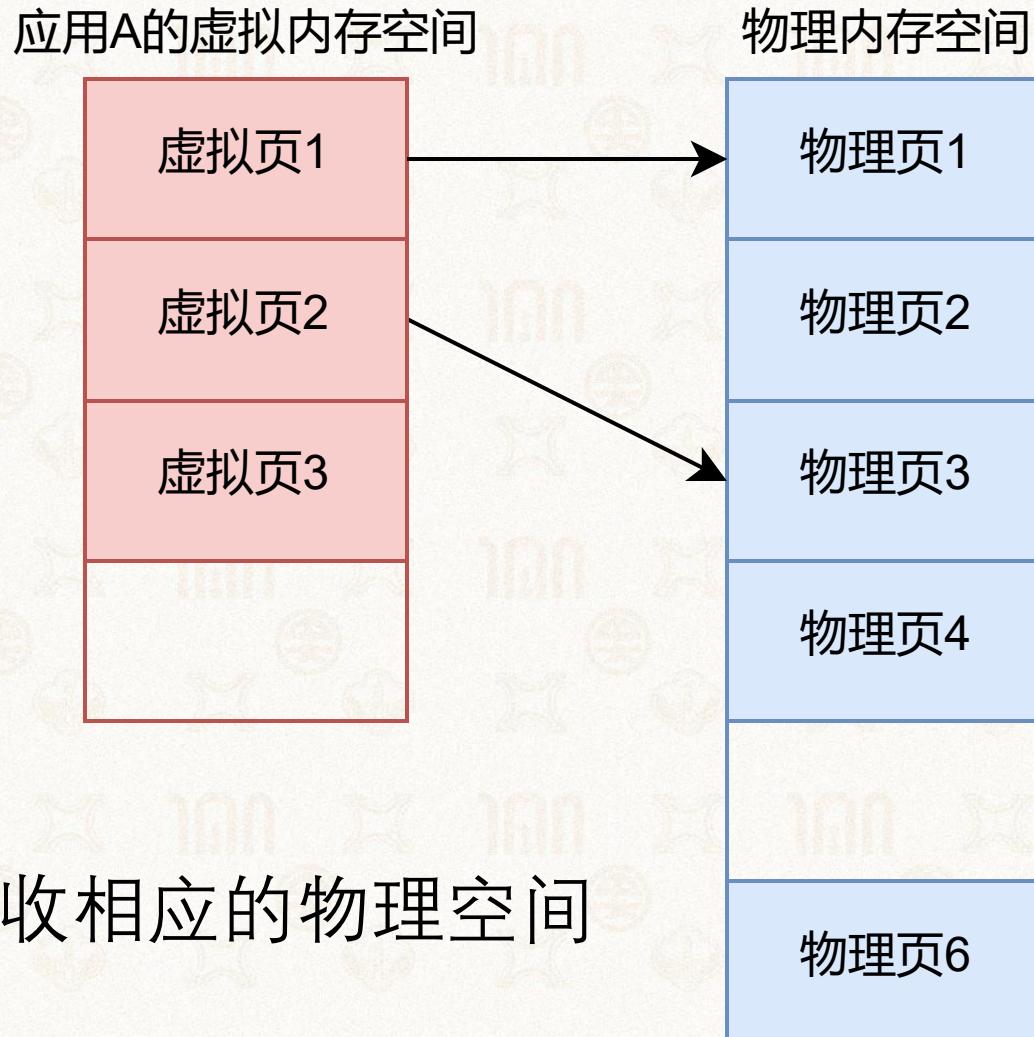
换页机制 (Swapping)



- 将“物理页5”换出，移入硬盘的 Swap 分区



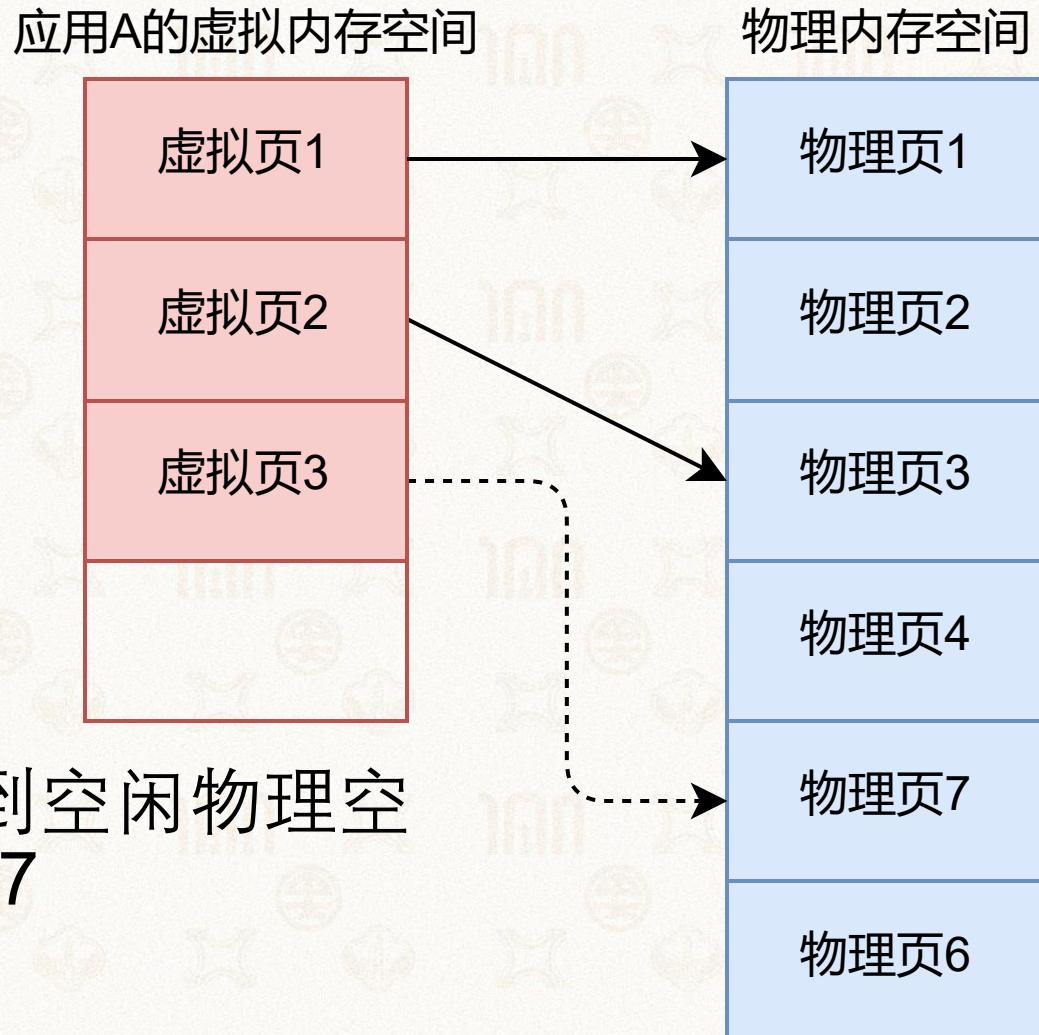
换页机制 (Swapping)



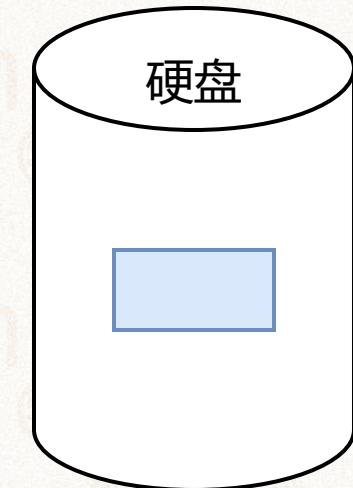
- 然后操作系统回收相应的物理空间



换页机制 (Swapping)



- 将虚拟页3映射到空闲物理空间，成为物理页7

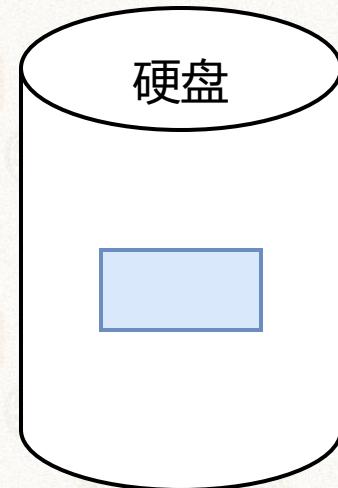
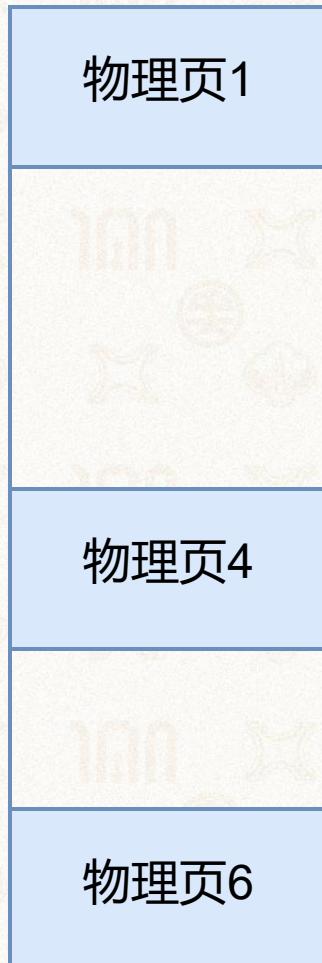




换页机制 (Swapping)

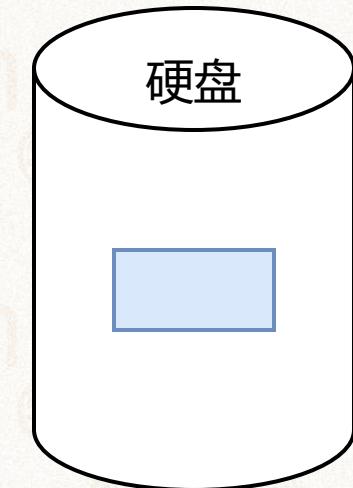
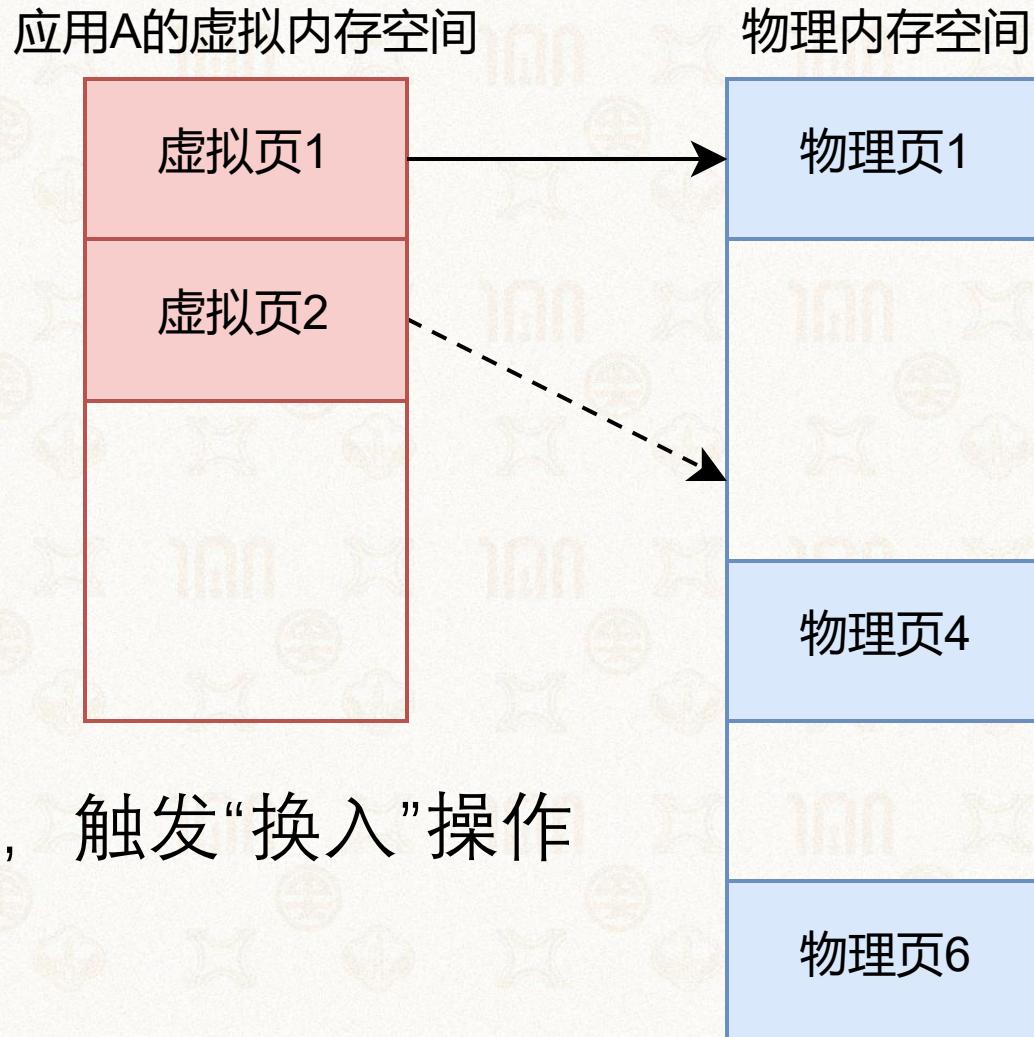
- 并不是等物理内存耗尽时才进行换入操作
- 内存快满时就开始了
- 可以腾出更多物理内存空间

物理内存空间





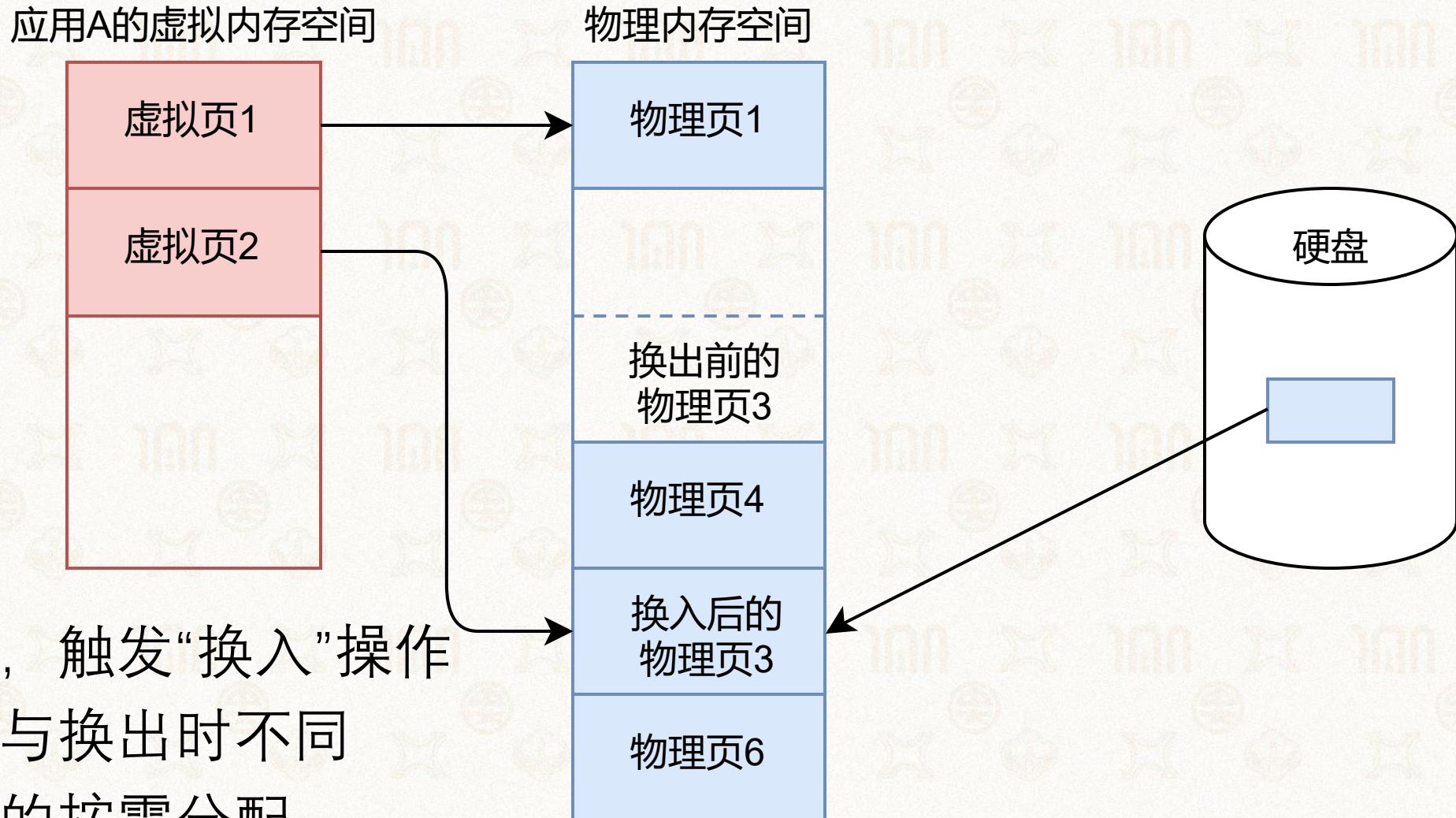
换页机制 (Swapping)



- 发生缺页异常时，触发“换入”操作



换页机制 (Swapping)



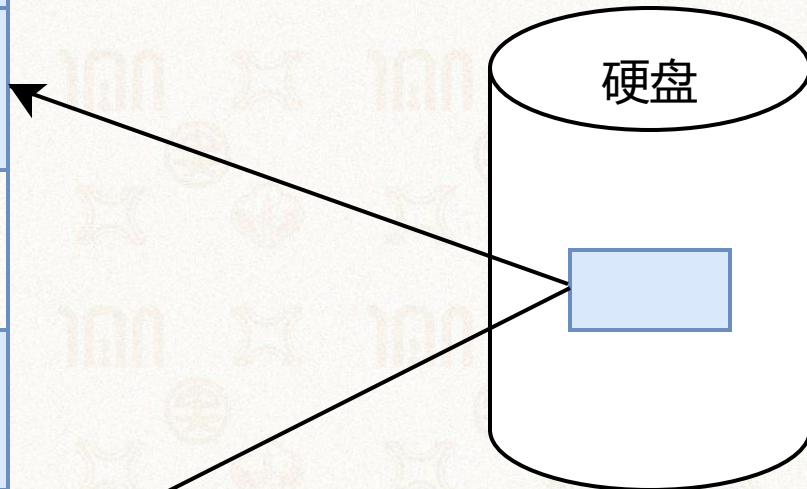


按需分配中的权衡

- 优势：节约内存资源
- 劣势：缺页异常导致访问延迟增加
- 如何取得平衡？
 - 应用程序访存具有时空局部性
 - 在缺页异常处理函数中采用预取（Prefetching）机制
 - 即节约内存又能减少缺页异常次数
 - 另外要精心设计换出策略，尽量减少缺页异常的次数

物理内存空间

物理页1
换入后的物理页2
物理页4
换入后的物理页3
物理页6





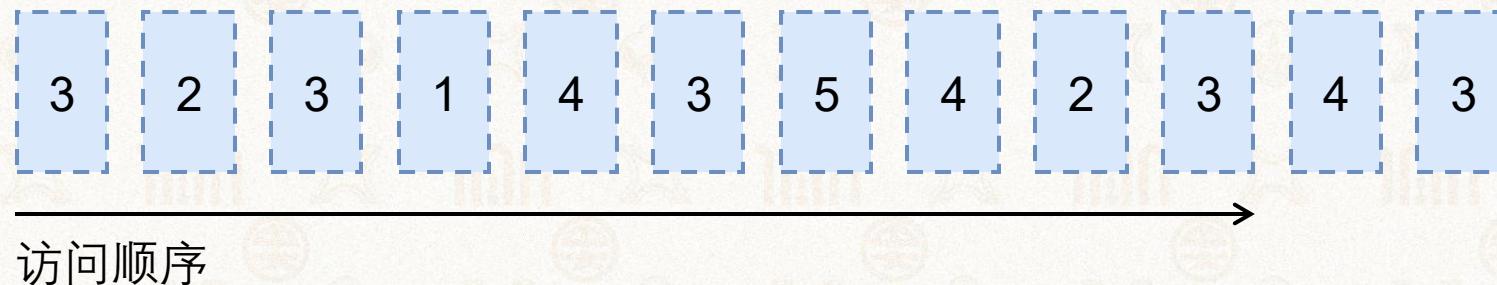
大纲

- 换页机制
- 页替换策略 (有私货)
- 工作集模型 (有私货)



页替换策略

- 当内存不足时选择哪个物理页被换出？



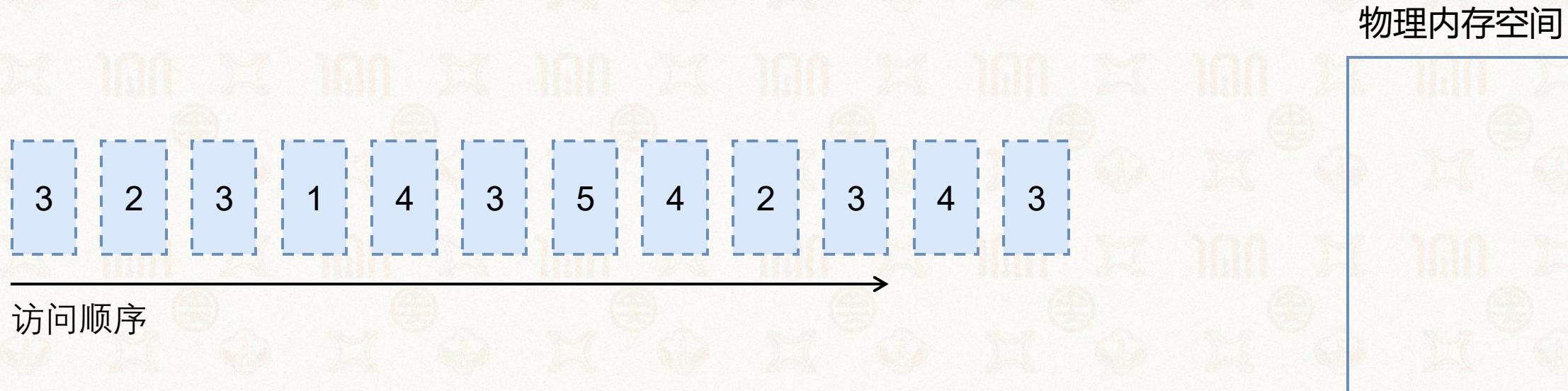
物理内存空间

物理页1
物理页2
物理页3
物理页4
物理页5
物理页6



页替换策略

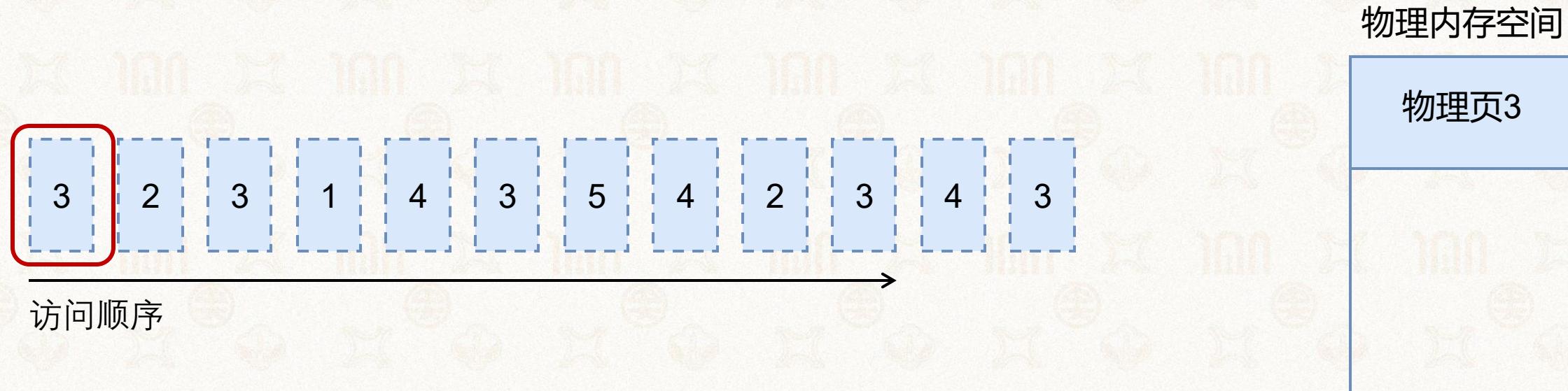
- 当内存不足时选择哪个物理页被换出？





页替换策略

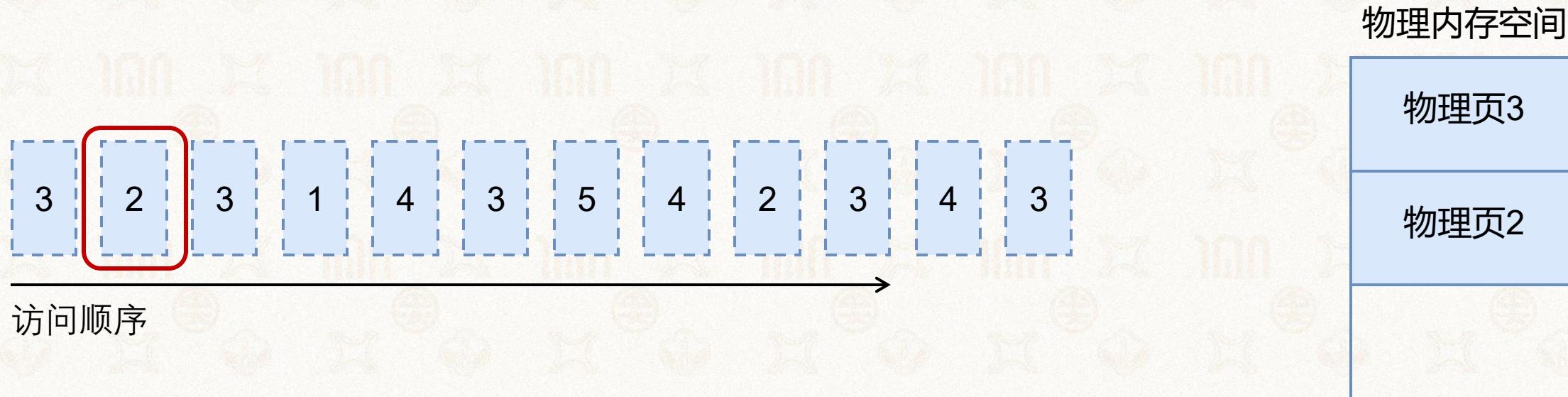
- 当内存不足时选择哪个物理页被换出?
- 发生缺页异常，因为没有分配物理内存





页替换策略

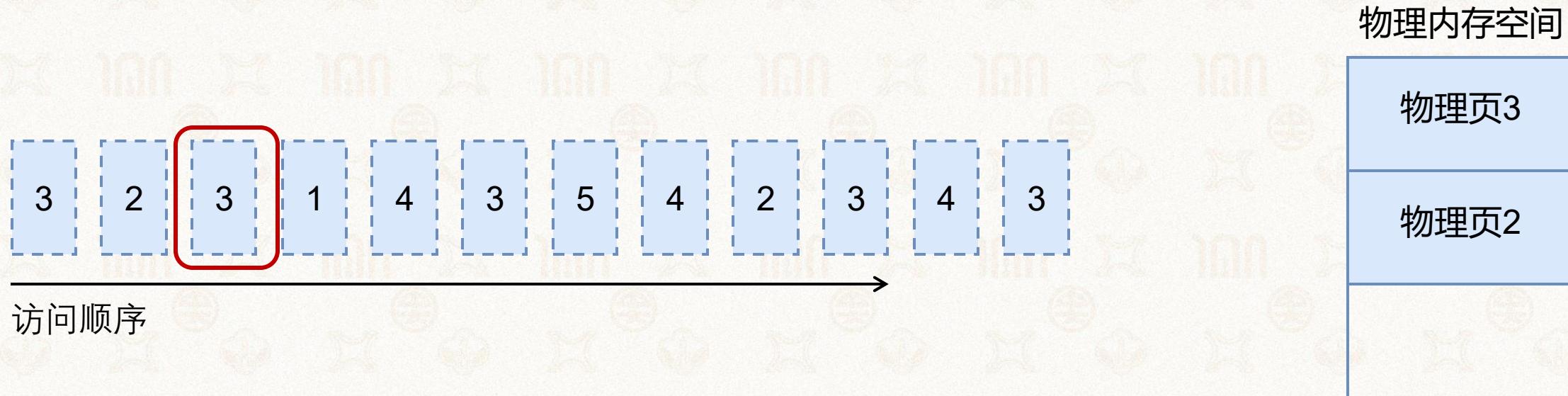
- 当内存不足时选择哪个物理页被换出?
- 发生缺页异常，因为没有分配物理内存





页替换策略

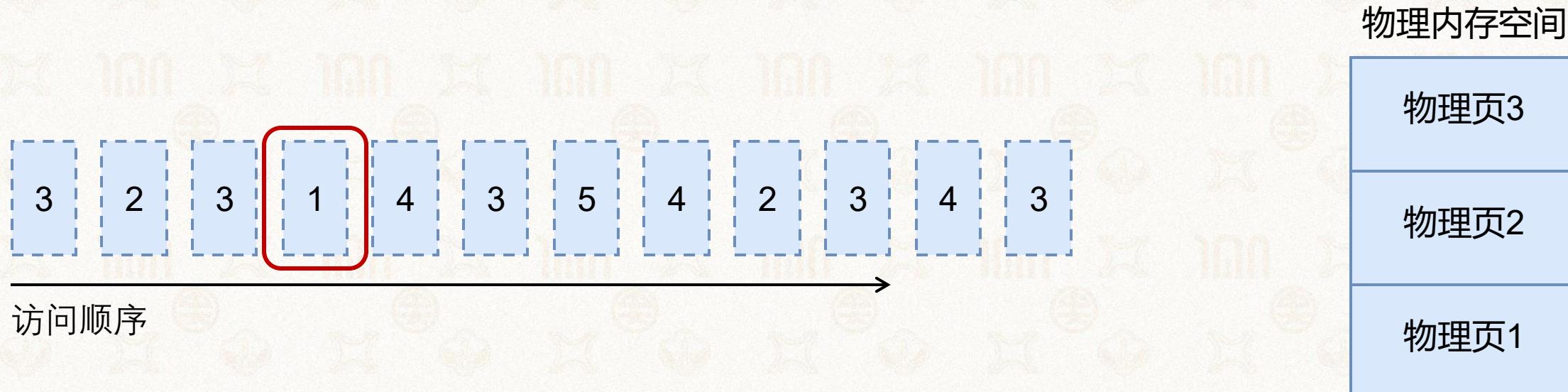
- 当内存不足时选择哪个物理页被换出?
- 没有缺页异常





页替换策略

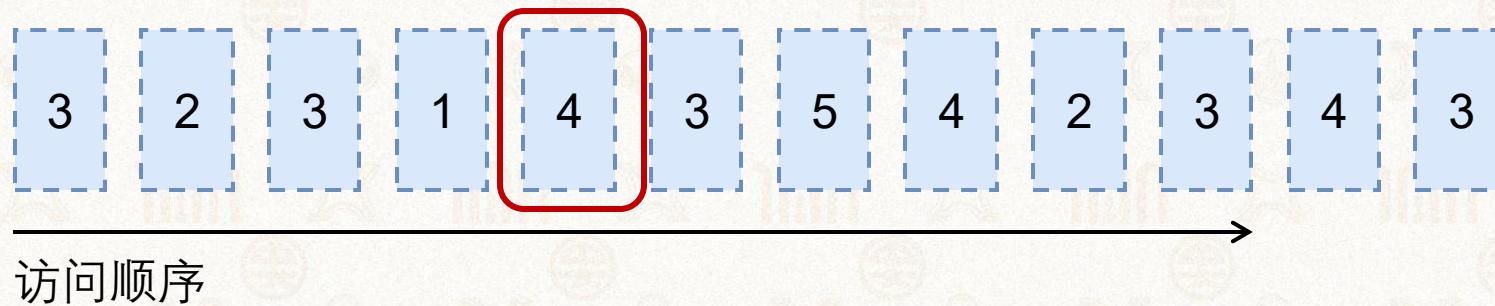
- 当内存不足时选择哪个物理页被换出?
- 发生缺页异常，因为没有分配物理内存



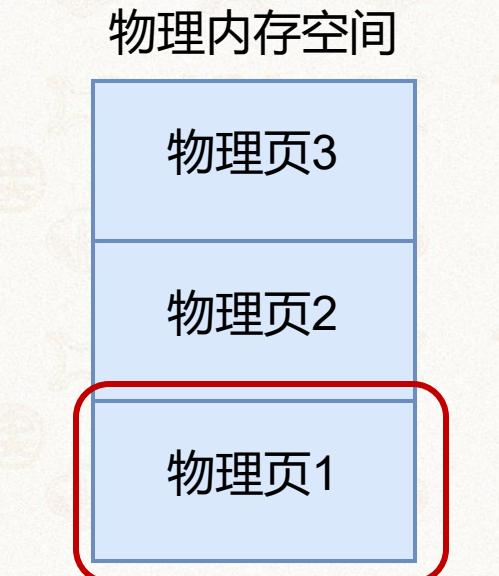


页替换策略：MIN策略

- 当内存不足时选择哪个物理页被换出？
- 发生缺页异常，因为没有分配物理内存
- 此时需要选择一个物理页换出



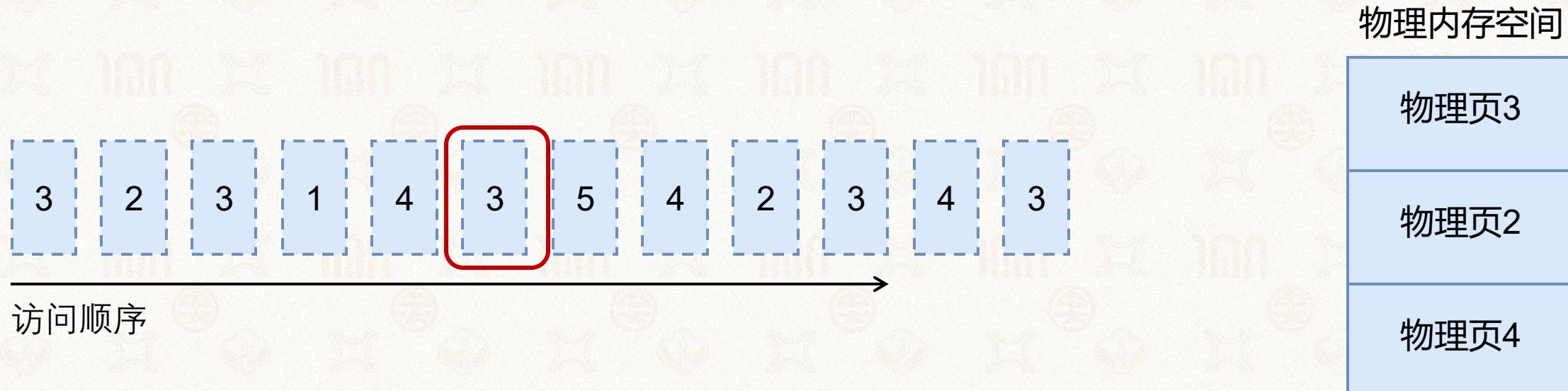
- MIN策略：
 - 选择未来不会再被访问的页
 - 或者最长时间内不再被访问





页替换策略：MIN策略

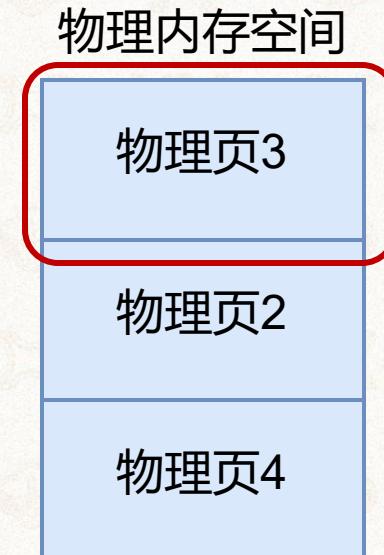
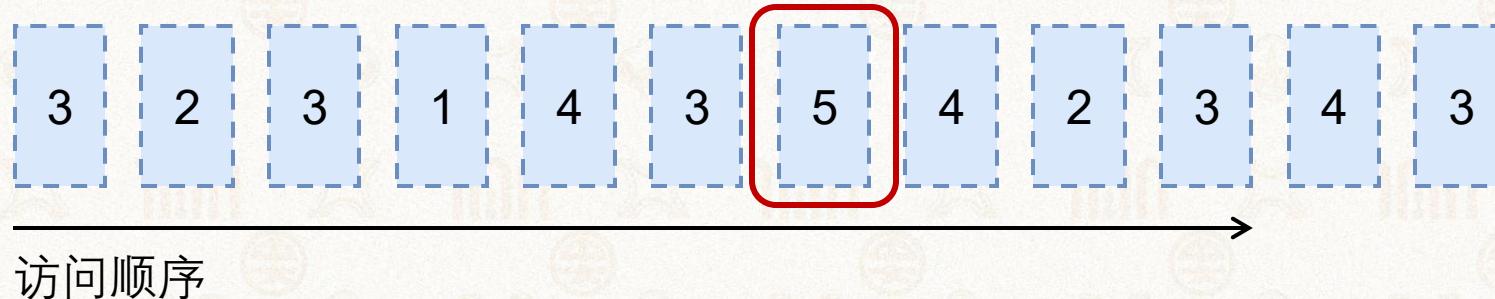
- 当内存不足时选择哪个物理页被换出？
- 没有缺页异常





页替换策略：MIN策略

- 当内存不足时选择哪个物理页被换出？
- 发生缺页异常，因为没有分配物理内存
- 此时需要选择一个物理页换出

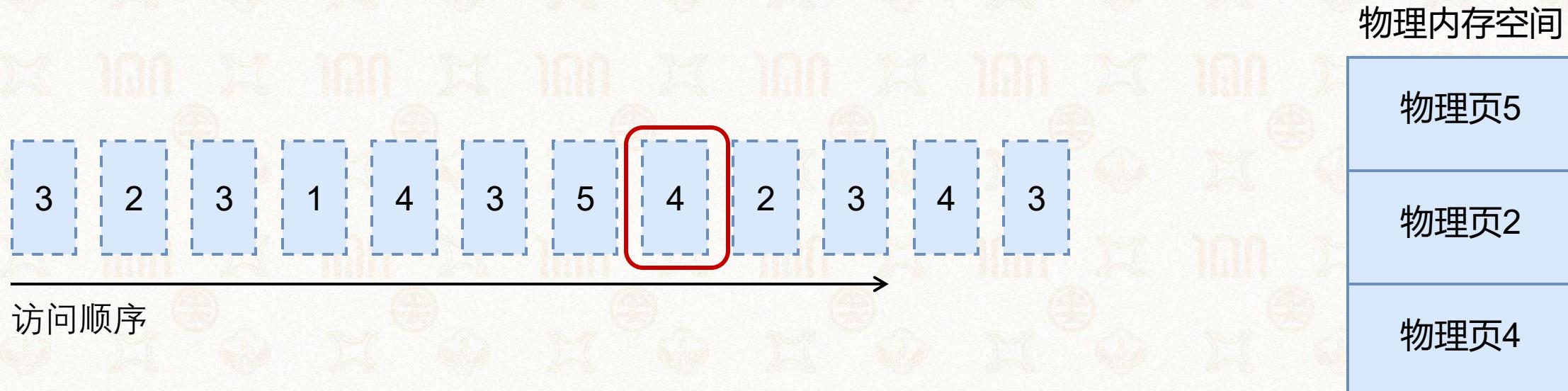


- MIN策略：
 - 选择未来不会再被访问的页
 - 或者最长时间内不再被访问
- 物理页4下次会访问
- 物理页2下下次会访问
- 只能选择物理页3



页替换策略：MIN策略

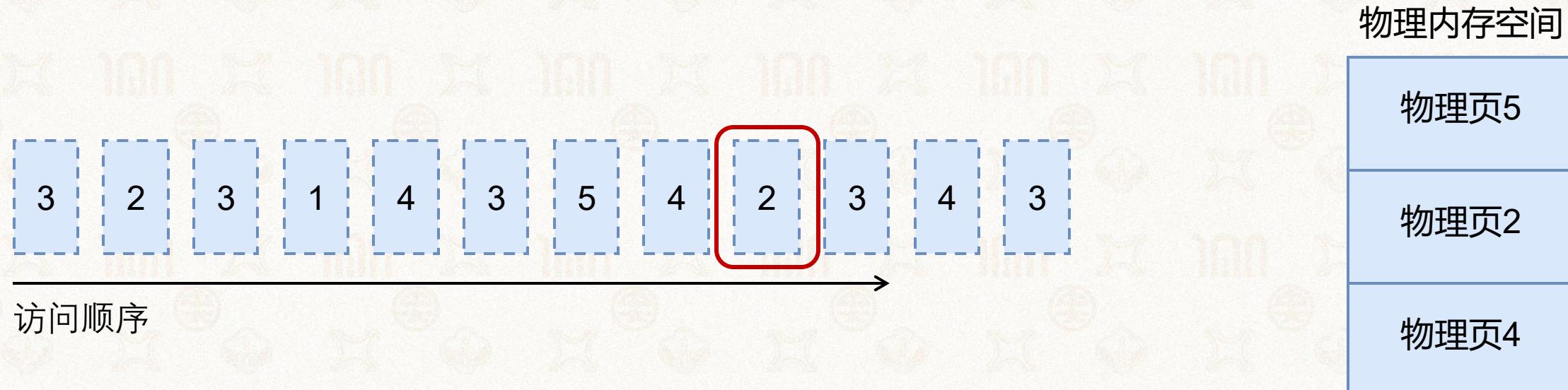
- 当内存不足时选择哪个物理页被换出？
- 没有缺页异常





页替换策略：MIN策略

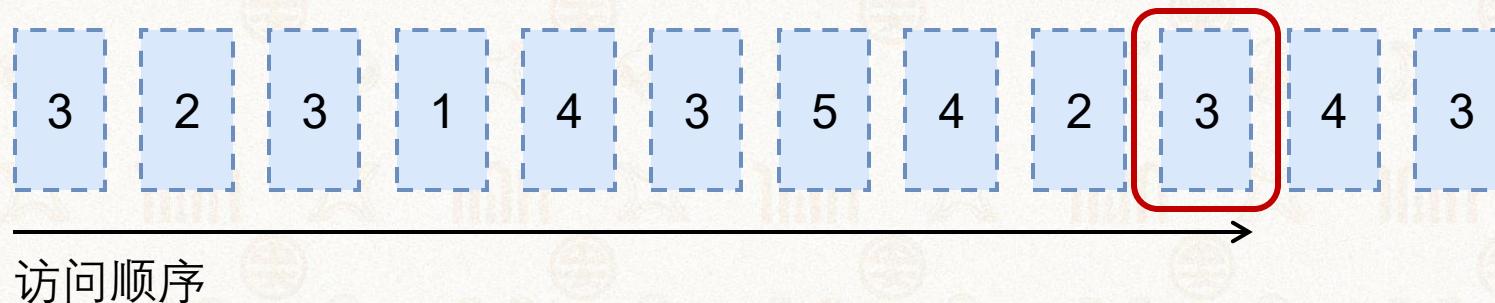
- 当内存不足时选择哪个物理页被换出？
- 没有缺页异常



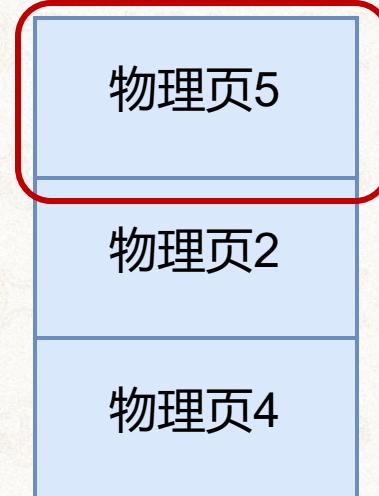


页替换策略：MIN策略

- 当内存不足时选择哪个物理页被换出？
- 发生缺页异常，因为物理页3曾被换出
- 此时需要选择一个物理页换出



物理内存空间



MIN策略：

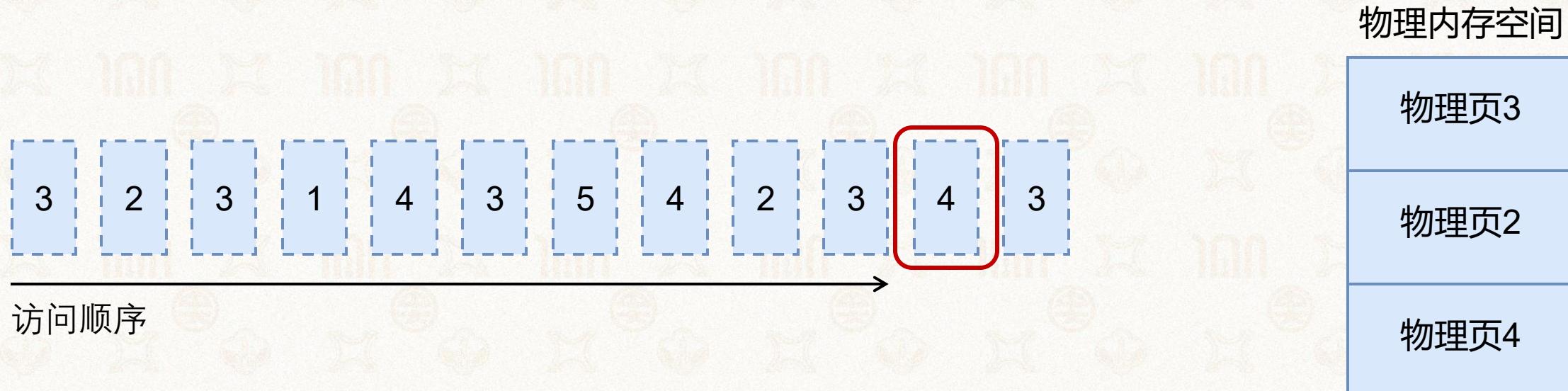
- 选择未来不会再被访问的页
- 或者最长时间内不再被访问

未来5不再被访问



页替换策略：MIN策略

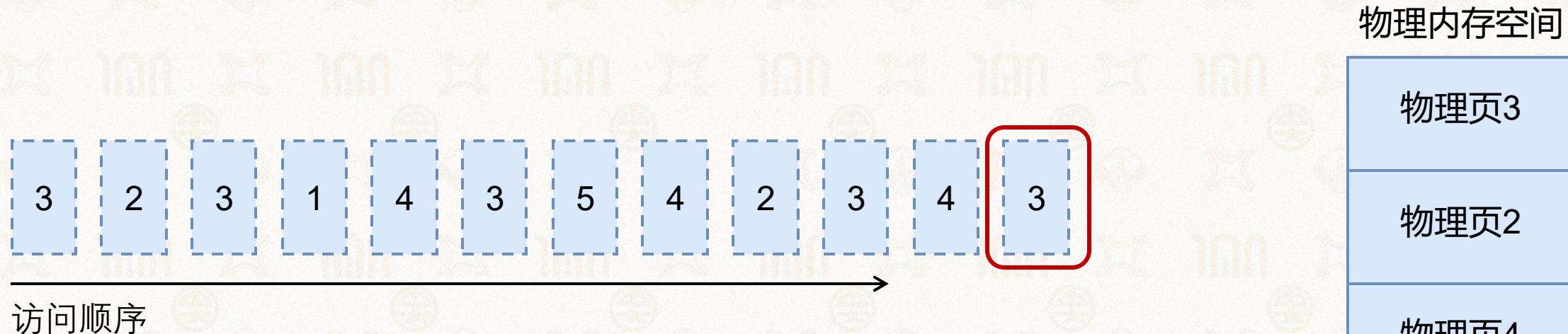
- 当内存不足时选择哪个物理页被换出？
- 没有缺页异常





页替换策略：MIN策略

- 当内存不足时选择哪个物理页被换出？
- 没有缺页异常



- MIN策略：
 - 未来如何知道？
 - 该策略只是理论上最优，实际很难实现

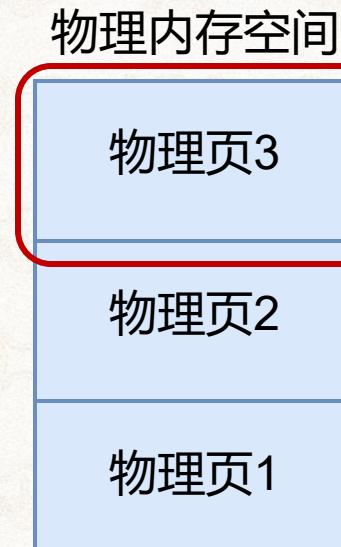
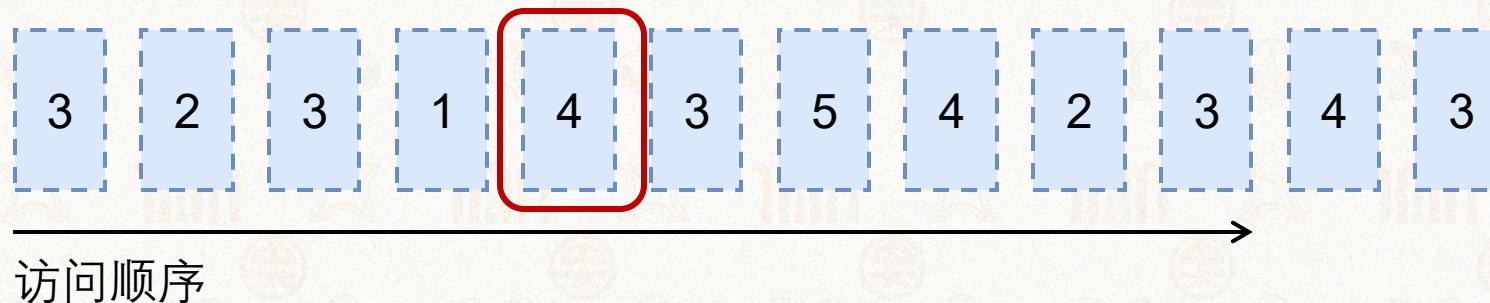


页替换策略：FIFO策略

➤ FIFO(First-In First-Out, 先进先出)

➤ 用队列记录访问过的物理页号

- 新页加入队尾
- 选择队头页换出

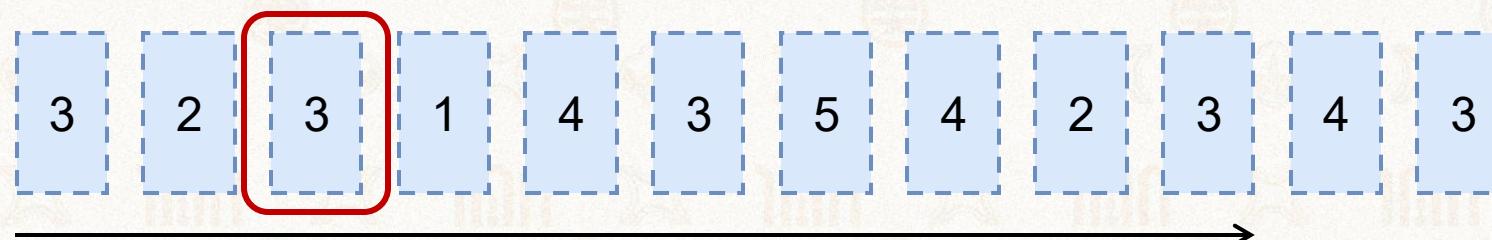


3在队头，被换出



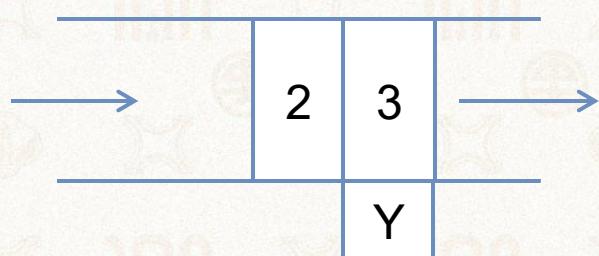
页替换策略: Second Chance策略

- FIFO的改进版
- 如果要访问的页已经队列中，加一个标记



访问顺序

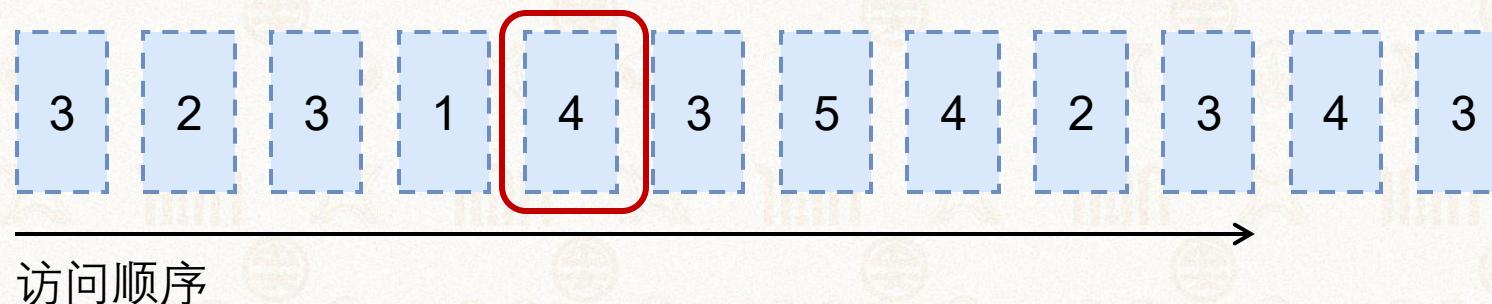
物理内存空间



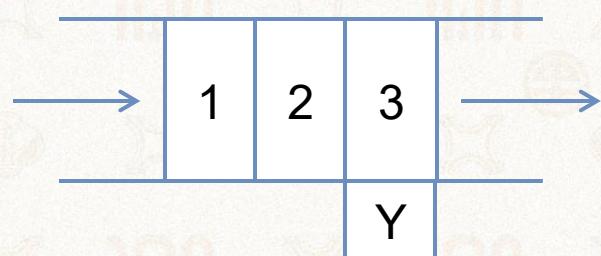
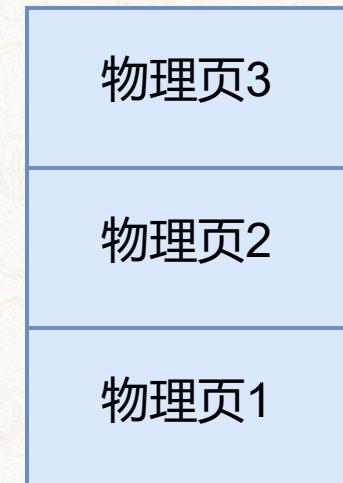


页替换策略: Second Chance策略

- FIFO的改进版
- 需要换页时:
 - 如果队头有标记, 清除标记, 移至队尾



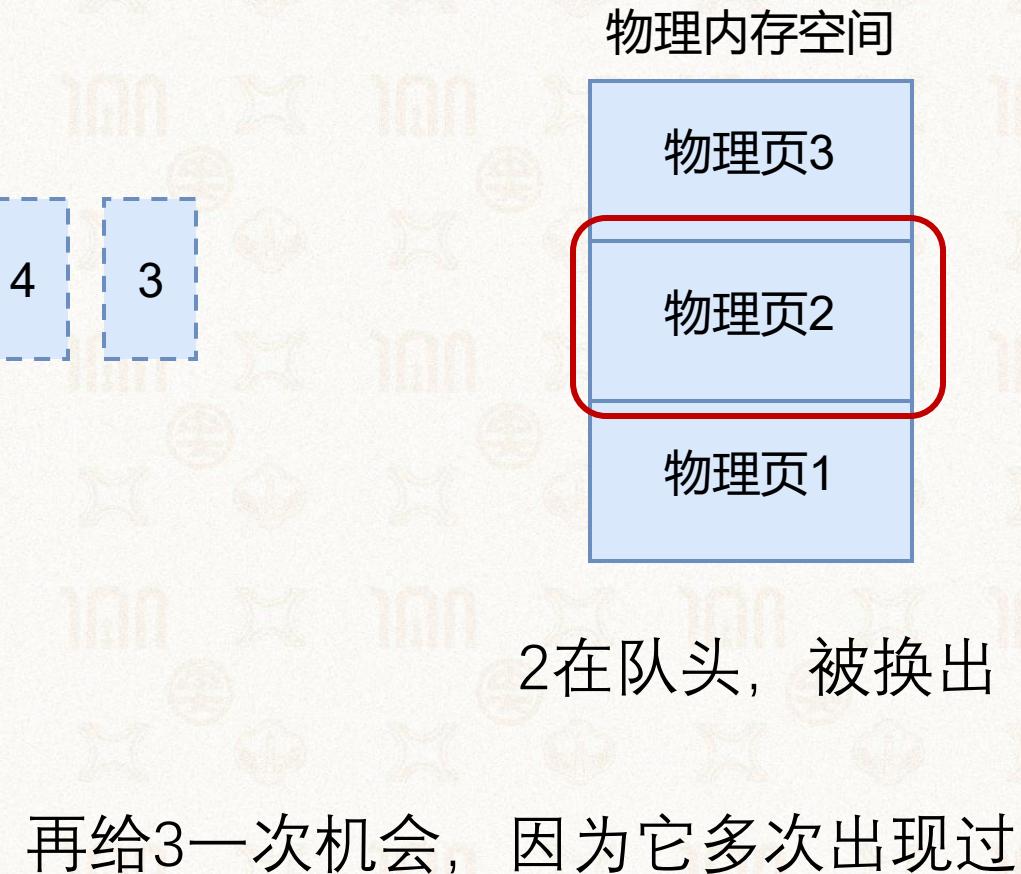
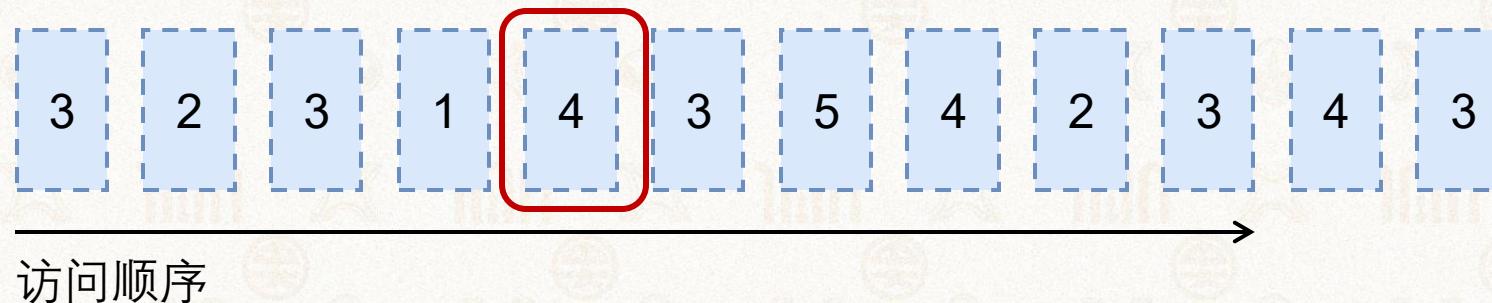
物理内存空间





页替换策略: Second Chance策略

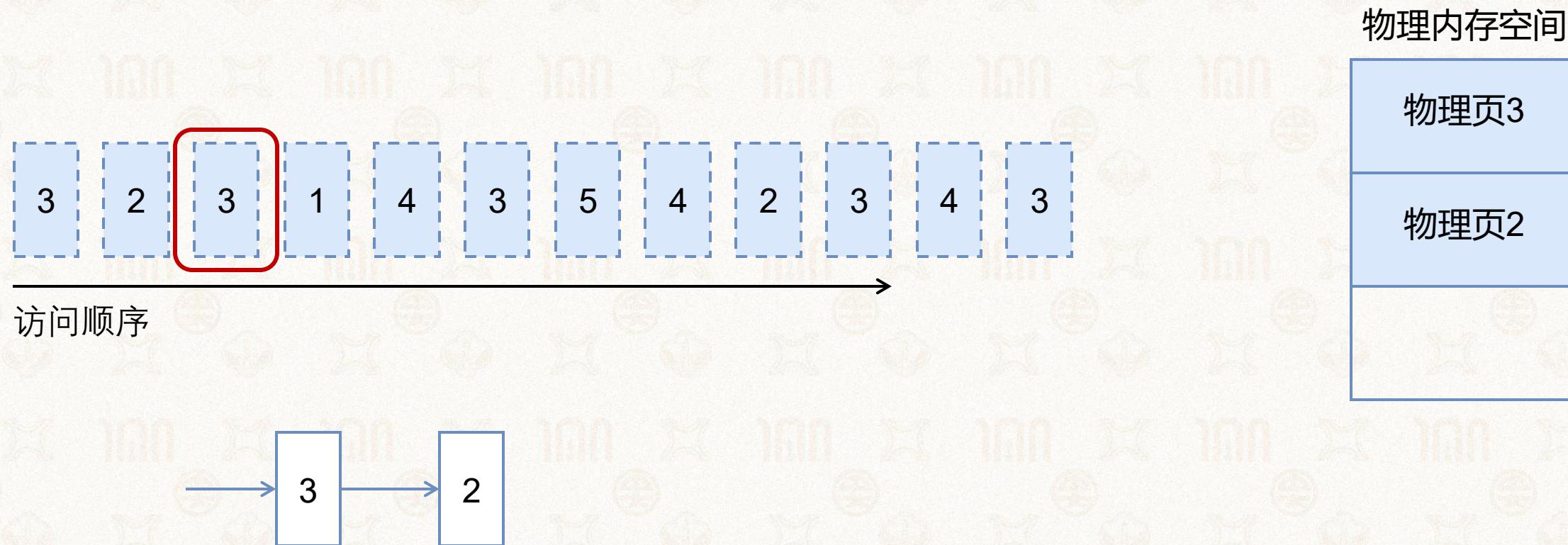
- FIFO的改进版
- 需要换页时:
 - 如果队头没有标记, 换出队头





页替换策略：LRU策略（高频面试知识点）

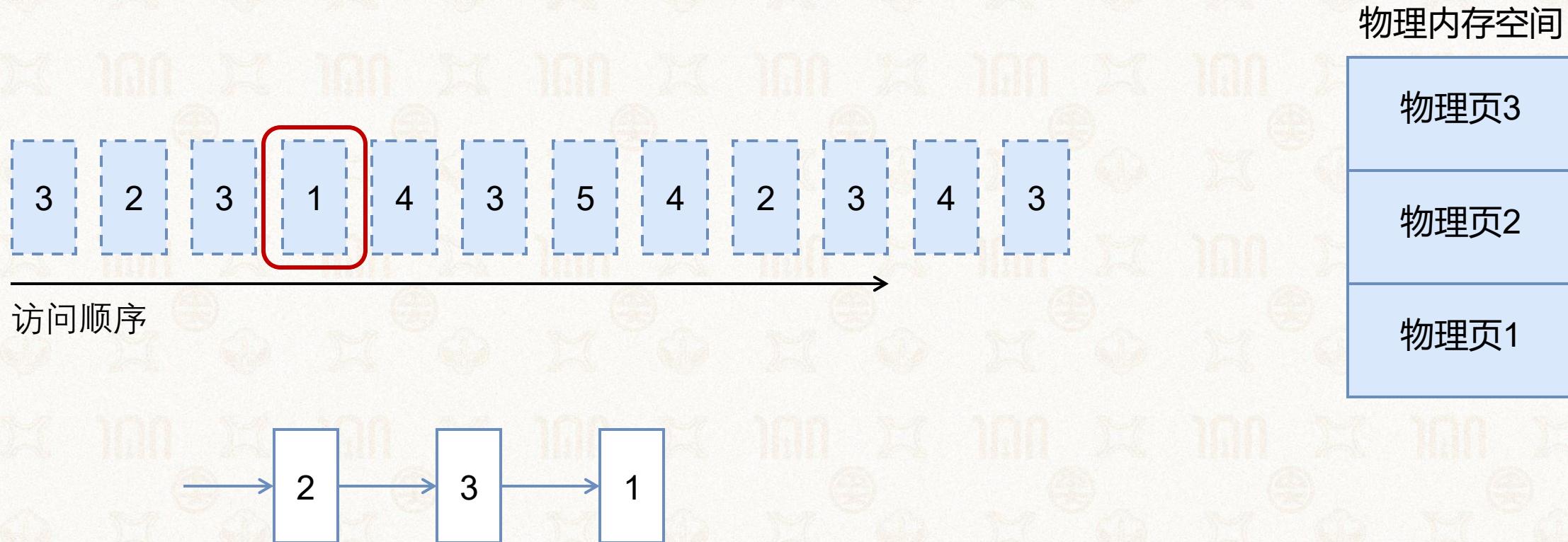
- LRU (Least Recently Used): 选择最久未被访问的页
- 用链表记录访问过程，新访问的移到尾端，换出首端页





页替换策略：LRU策略（高频面试知识点）

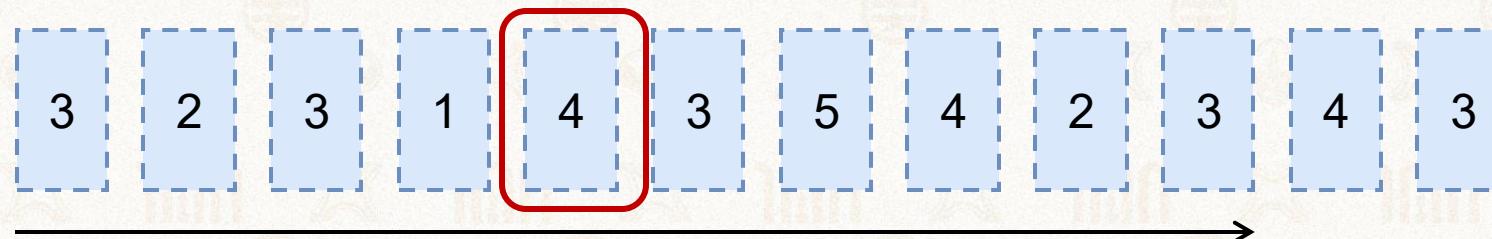
- LRU (Least Recently Used): 选择最久未被访问的页
- 用链表记录访问过程，新访问的移到尾端，换出首端页



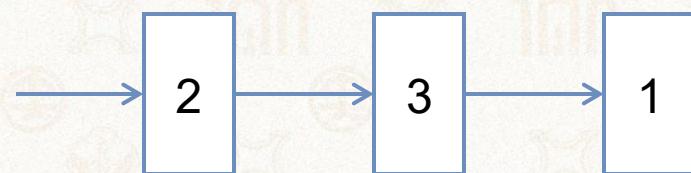


页替换策略：LRU策略（高频面试知识点）

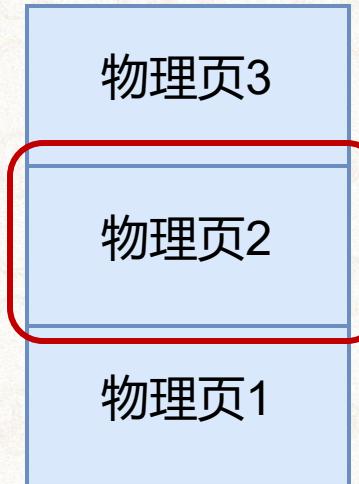
- LRU (Least Recently Used): 选择最久未被访问的页
- 用链表记录访问过程，新访问的移到尾端，换出首端页



访问顺序



物理内存空间

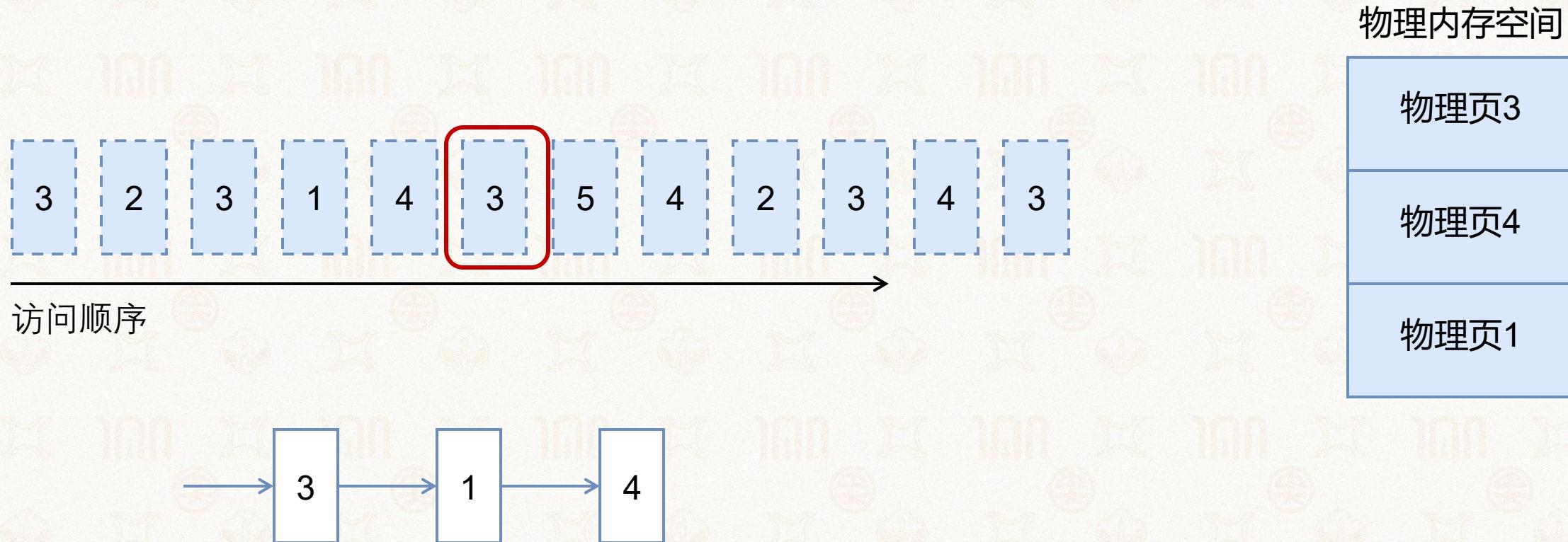


2在首端，最久未
被访问，被换出



页替换策略：LRU策略（高频面试知识点）

- LRU (Least Recently Used): 选择最久未被访问的页
- 用链表记录访问过程，新访问的移到尾端，换出首端页



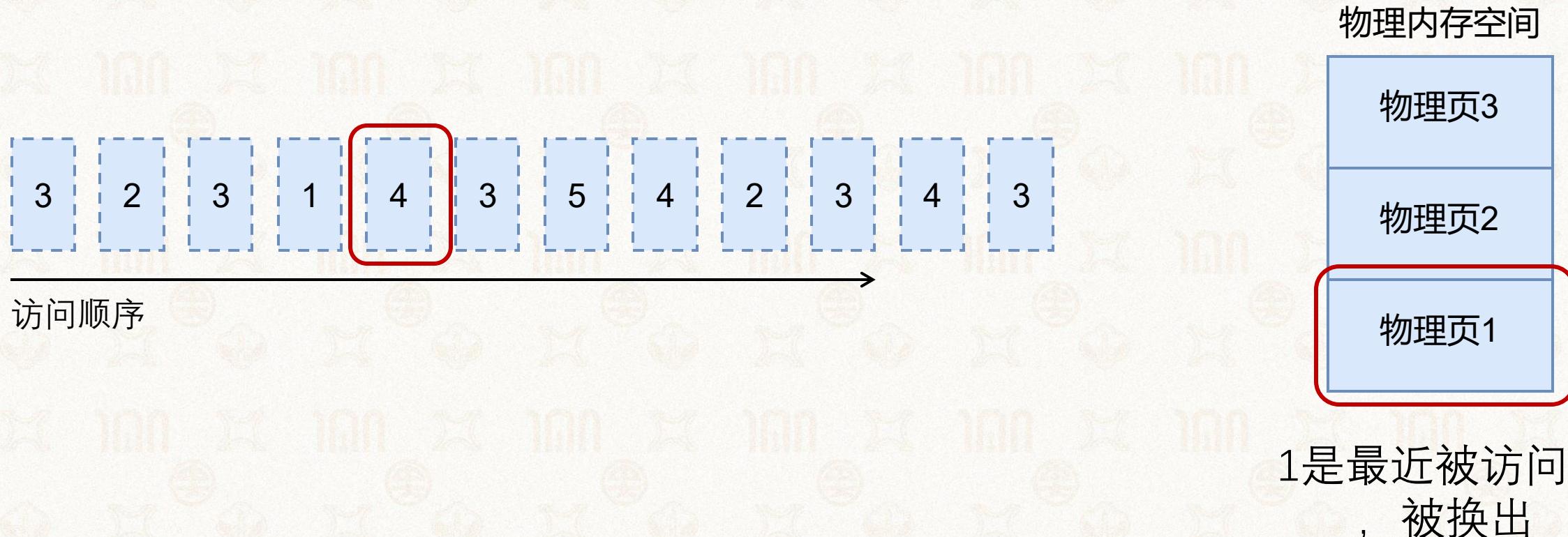
高频面试题：手写LRU。说一下思路，然后写一下简单实现。要求时间复杂度为 $O(1)$
(本题只回答思路就好)

作答



页替换策略：MRU策略

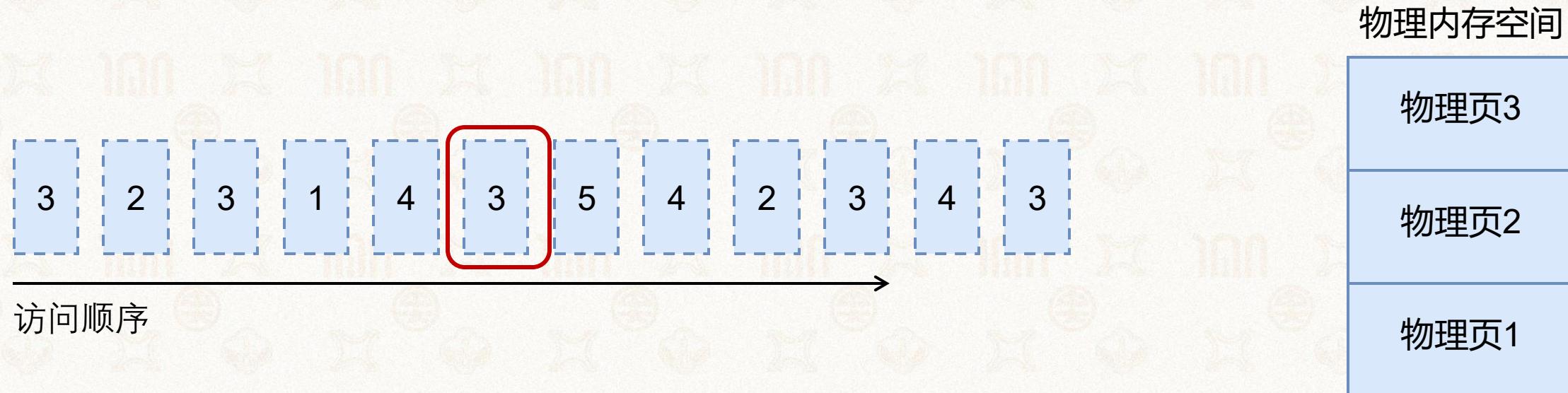
- MRU (Most Recently Used): 选择最近被访问的页
- 假设程序不会访问相同的地址





页替换策略：颠簸(thrashing)现象

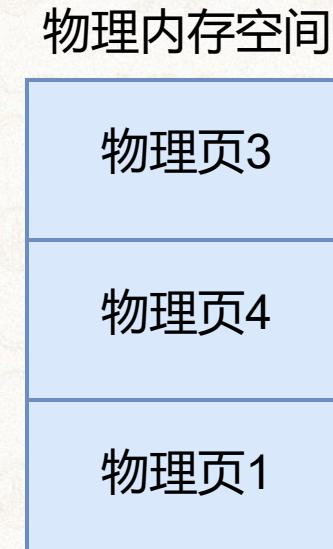
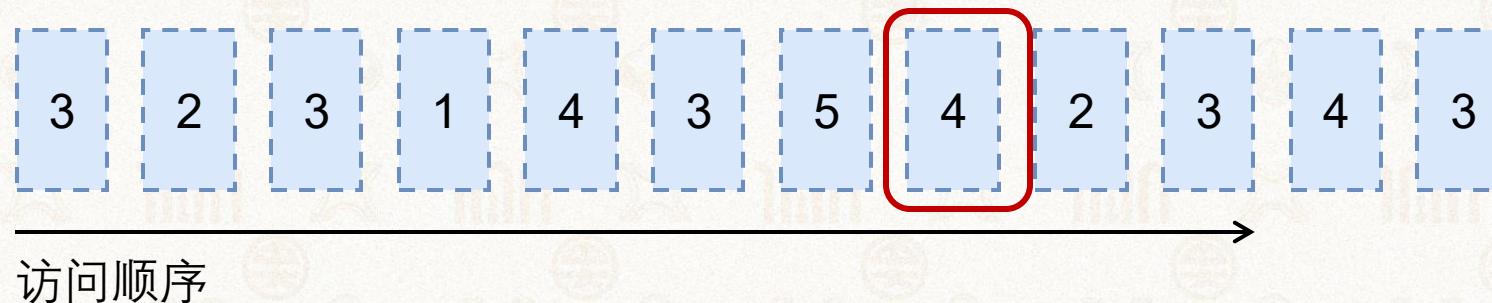
- 若替换策略选择不好，物理页刚被使用就被换出，很快再被换入
- 4被换出





页替换策略：颠簸(thrashing)现象

- 若替换策略选择不好，物理页刚被使用就被换出，很快再被换入
- 4被换入
- 频繁换出、换入相同物理页严重降低性能





页替换策略：颠簸(thrashing)现象

➤ 直接原因

- 过于频繁的缺页异常（物理内存总需求过大）

➤ 大部分 CPU 时间都被用来处理缺页异常

- 等待缓慢的磁盘 I/O 操作
- 仅剩小部分的时间用于执行真正有意义的工作

➤ 调度器造成问题加剧

- 等待磁盘 I/O 导致 CPU 利用率下降
- 调度器载入更多的进程以期提高 CPU 利用率
- 触发更多的缺页异常、进一步降低 CPU 利用率、导致连锁反应

物理内存空间

物理页3

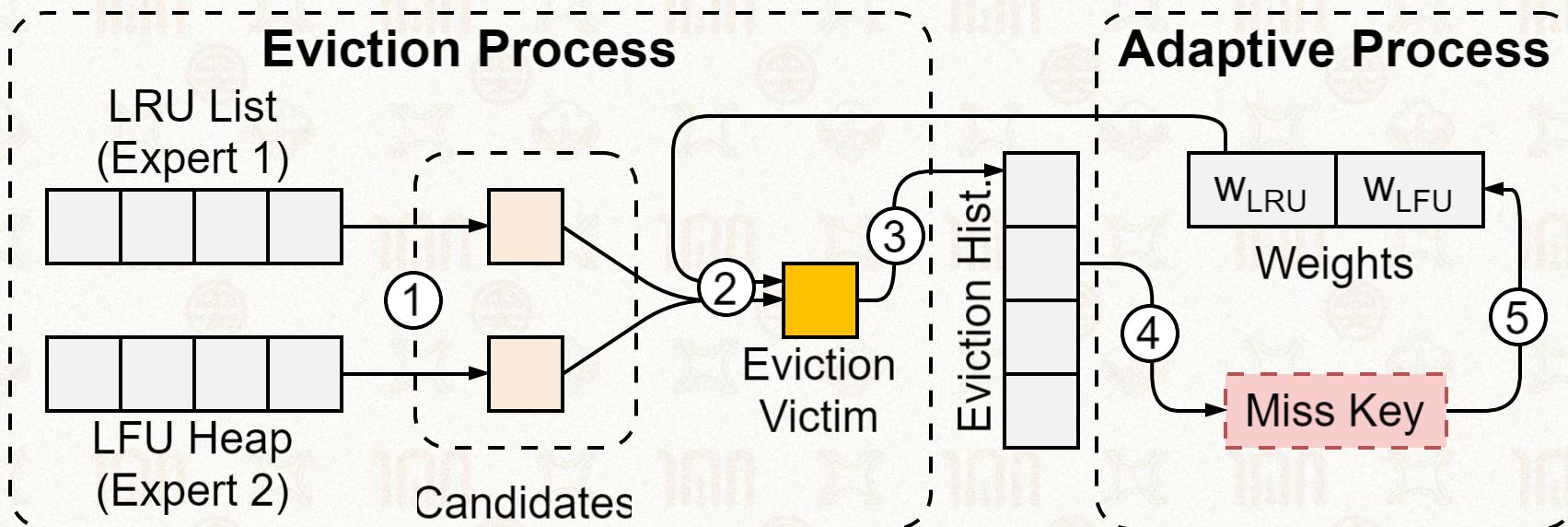
物理页4

物理页1



私货：机器学习算法实现动态缓存替换

- 真实的程序更复杂，教科书级算法不能有效应对
- 利用多臂赌博机理论预测某一段时刻更适合使用哪一种替换策略





大纲

- 换页机制
- 页替换策略（有私货）
- 工作集模型（有私货）



工作集模型(working set method)

- 现代操作系统中常用的方法，避免颠簸
- 上一时段使用的所有物理页，在下一时间段大概率也会用到
- 应用在t时刻的工作集W是[t-x, t]时间内使用的所有物理内存页集合

物理内存空间	上次使用时间	访问位
物理页1		0
物理页2		0
物理页3		0
物理页4		0
物理页5		0
物理页6		0



工作集模型(working set method)

- 现代操作系统中常用的方法，避免颠簸
- 上一时段使用的所有物理页，在下一时间段大概率也会用到
- 应用在t时刻的工作集W是[t-x, t]时间内使用的所有物理内存页集合

- 周期性地扫描所有物理页：
 - 访问位为1，表示曾被访问过
 - 在时刻 t_1 ：
 - 如果访问过，设置时间戳，然后重置状态（设回0）

物理内存空间	上次使用时间	访问位
物理页1		0
物理页2	t_1	1
物理页3		0
物理页4	t_1	1
物理页5	t_1	1
物理页6		0



工作集模型(working set method)

- 现代操作系统中常用的方法，避免颠簸
- 上一时段使用的所有物理页，在下一时间段大概率也会用到
- 应用在t时刻的工作集W是[t-x, t]时间内使用的所有物理内存页集合

- 周期性地扫描所有物理页：
 - 访问位为1，表示曾被访问过
 - 在时刻 t_1 ：
 - 如果访问过，设置时间戳，然后重置状态

物理内存空间	上次使用时间	访问位
物理页1		0
物理页2	t_1	0
物理页3		0
物理页4	t_1	0
物理页5	t_1	0
物理页6		0



工作集模型(working set method)

- 现代操作系统中常用的方法，避免颠簸
- 上一时间段使用的所有物理页，在下一时间段大概率也会用到
- 应用在t时刻的工作集W是[t-x, t]时间内使用的所有物理内存页集合

- 周期性地扫描所有物理页：
 - 访问位为1，表示曾被访问过
 - 在时刻 t_2 ：
 - 如果没被访问过，计算有多久没被访问过: $t_2 - t_1$
 - 如果超出预定时间范围x，该页不再属于工作集：
 - $t_2 - t_1 > x$
 - 可以被换出

物理内存空间	上次使用时间	访问位
物理页1	t_2	1
物理页2	t_1	0
物理页3	t_2	1
物理页4	t_1	0
物理页5	t_1	0
物理页6	t_2	1



工作集模型(working set method)

- 现代操作系统中常用的方法，避免颠簸
- 上一时间段使用的所有物理页，在下一时间段大概率也会用到
- 应用在t时刻的工作集W是[t-x, t]时间内使用的所有物理内存页集合

- 周期性地扫描所有物理页：
 - 访问位为1，表示曾被访问过
 - 在时刻 t_2 ：
 - 如果没被访问过，计算有多久没被访问过: $t_2 - t_1$
 - 如果超出预定时间范围x，该页不再属于工作集：
 - $t_2 - t_1 > x$
 - 可以被换出

物理内存空间	上次使用时间	访问位
物理页1	t_2	1
物理页2	t_1	0
物理页3	t_2	1
物理页4	t_1	0
物理页5	t_1	0
物理页6	t_2	1

物理页2、4、5有一阵子没有被访问过了，
要检查是不是换出它们



工作集模型(working set method)

- 现代操作系统中常用的方法，避免颠簸
- 上一时段使用的所有物理页，在下一时间段大概率也会用到
- 应用在t时刻的工作集W是[t-x, t]时间内使用的所有物理内存页集合
- 周期性地扫描所有物理页：
 - 访问位为1，表示曾被访问过
 - 在时刻 t_2 ：
 - 如果访问过，设置时间戳 t_2 ，然后重置状态（就是重新设成0），重新观察未来是否还被访问

物理页1、3、6在 t_2 之前被访问过，现在重置，接下来要考察它们是不是要在未来 t_3 时刻被换出了

物理内存空间	上次使用时间	访问位
物理页1	t_2	0
物理页2	t_1	0
物理页3	t_2	0
物理页4	t_1	0
物理页5	t_1	0
物理页6	t_2	0



工作集模型(working set method)

- 需要硬件支持：访问页表时可以自动置“访问位”为1

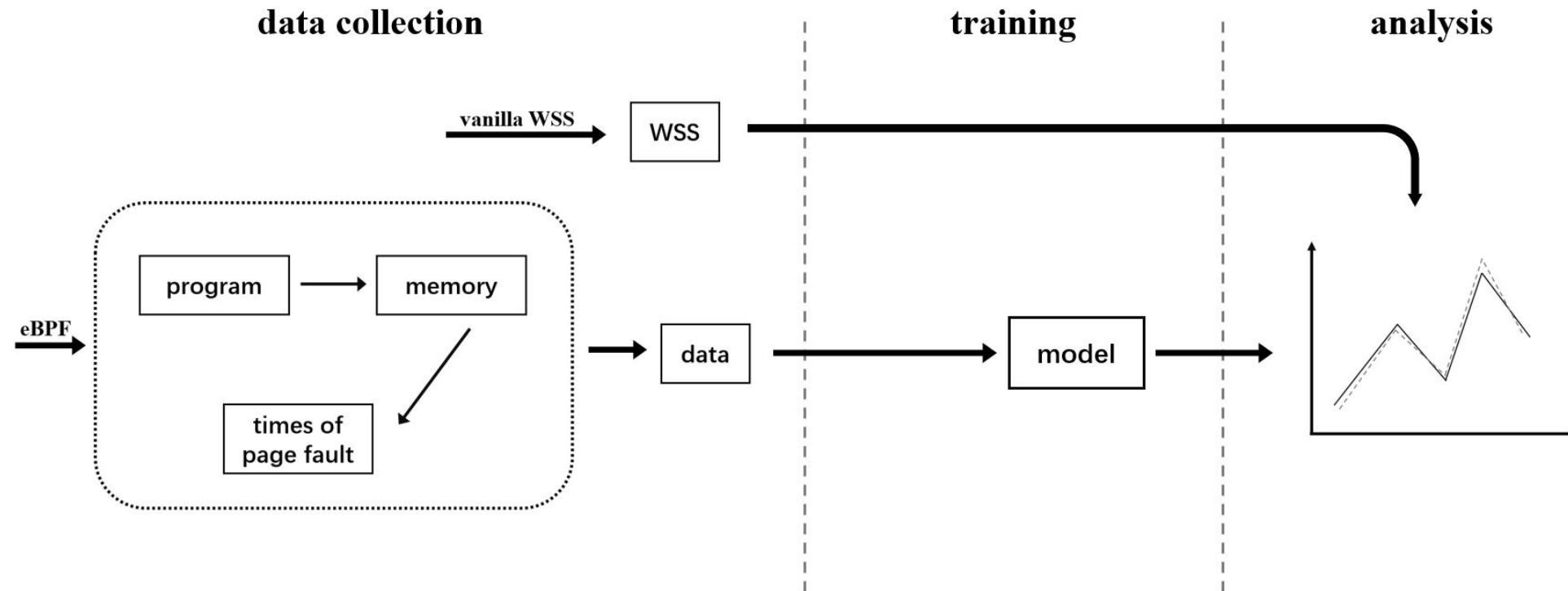
进程						
	PID	硬中断/秒	提交(KB)	工作集(KB)	可共享(KB)	专用(KB)
<input type="checkbox"/> 名称						
<input type="checkbox"/> OneDrive.exe	10788	0	534,104	561,792	61,316	500,476
<input type="checkbox"/> POWERPNT.EXE	17968	0	522,612	496,652	160,712	335,940
<input type="checkbox"/> explorer.exe	8372	0	550,536	489,368	228,688	260,680
<input type="checkbox"/> msedge.exe	15960	0	199,764	346,764	166,532	180,232
<input type="checkbox"/> WeChat.exe	18560	0	236,560	314,564	120,940	193,624
<input type="checkbox"/> msedge.exe	16160	0	888,644	310,412	66,976	243,436
<input type="checkbox"/> DingTalk.exe	21148	0	393,844	307,984	109,628	198,356
<input type="checkbox"/> MsMpEng.exe	4952	0	302,792	297,004	70,916	226,088





私货：利用机器学习方法预测工作集大小

- 使用eBPF高效、准确地收集缺页异常次数
- 借助LightGBM训练回归模型，拟合缺页异常次数与工作集大小的关系

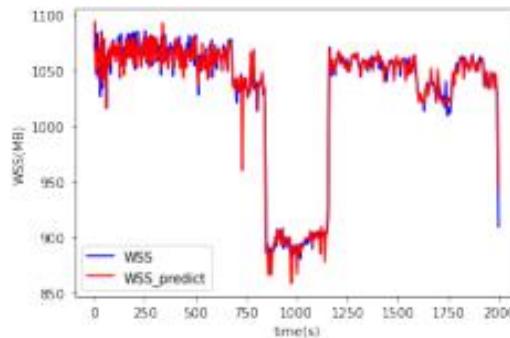
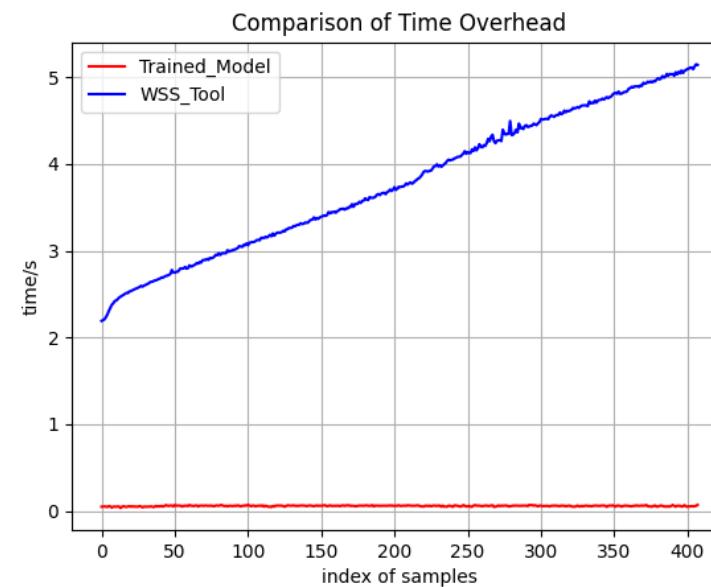


五位20级同学大二时的大创项目：Zhilu Lian (练芷璐), Yangzi Li, Zhixiang Chen, Shiwen Shan (单诗雯), Baoxin Han and Yuxin Su. “eBPF-based Working Set Size Estimation in Memory Management”. International Conference on Service Science (ICSS), 2022

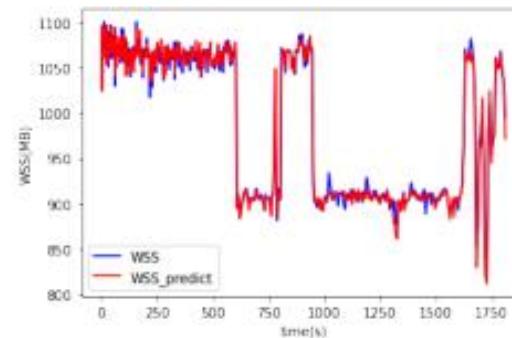


私货：利用机器学习方法预测工作集大小

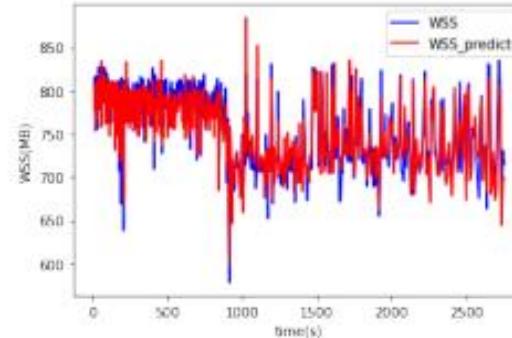
- 运行速度快，可实时预测
- 准确度高，预测误差小



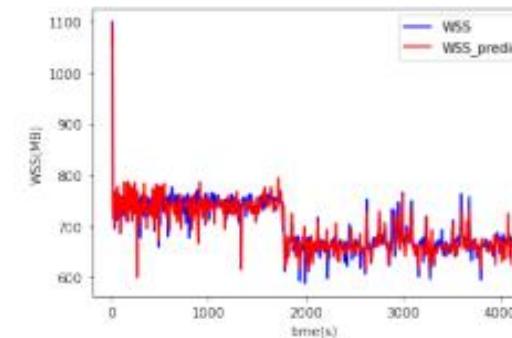
(a) Output of Test 1



(b) Output of Test 2



(c) Output of Test 3



(d) Output of Test 4

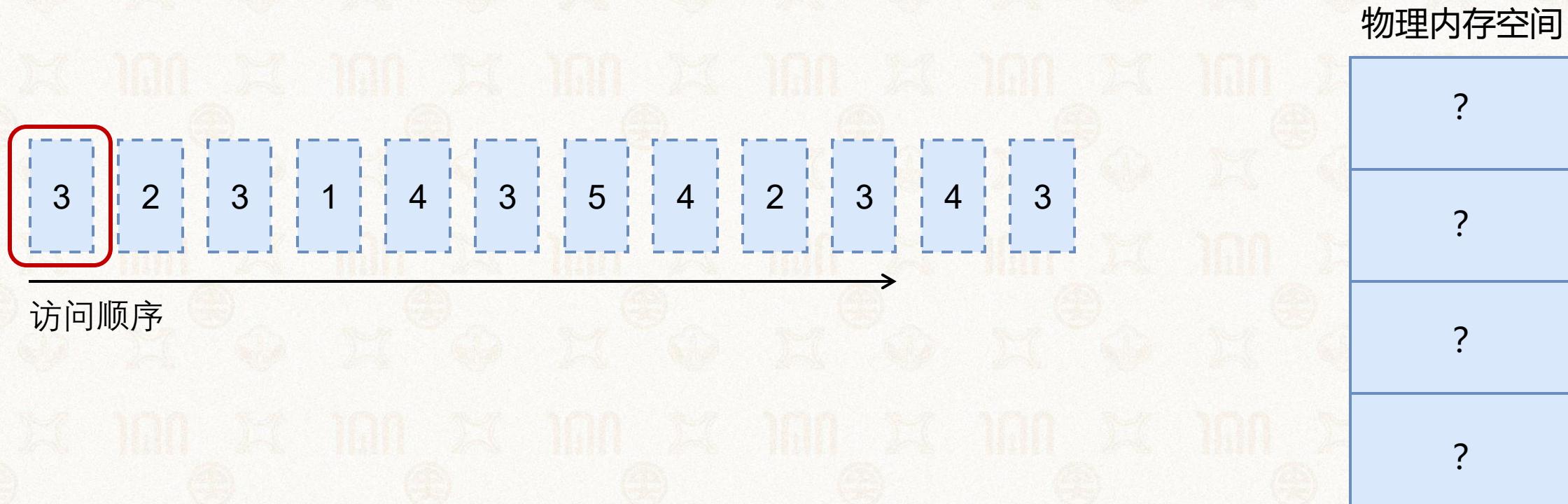


大纲

- 换页机制
- 页替换策略（有私货）
- 工作集模型（有私货）

页替换策略：MIN策略

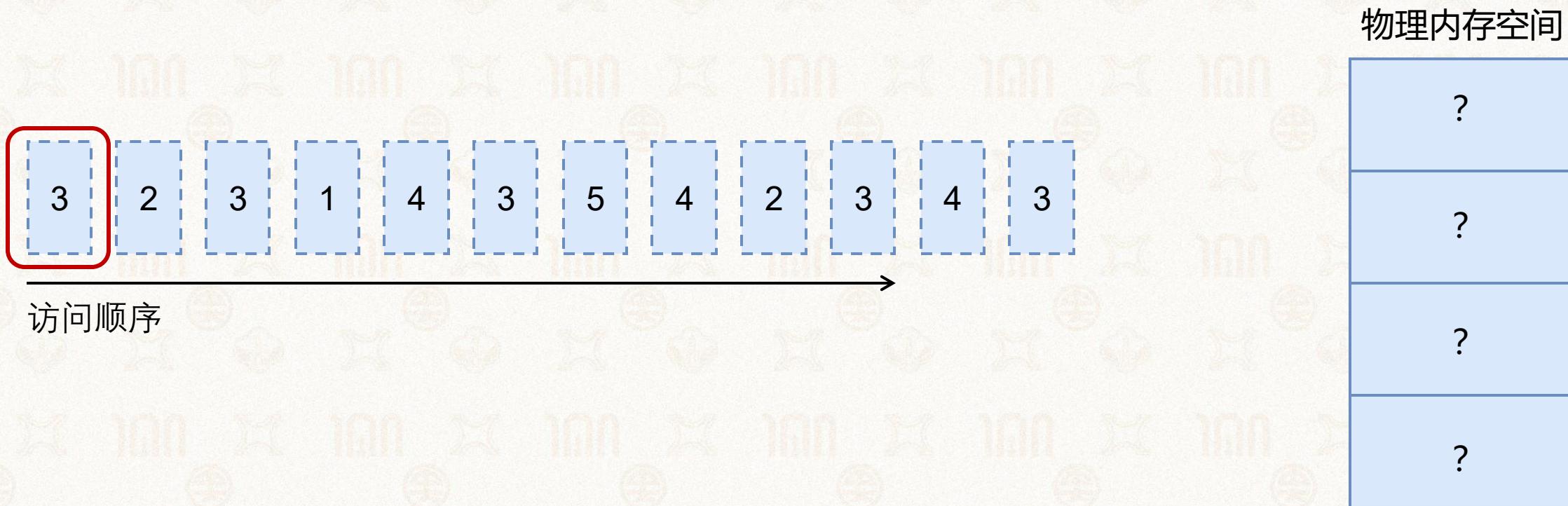
如果物理内存空间可以容纳4个物理页，那么执行完该策略之后，物理内存空间里的物理页号为多少？



作答

页替换策略：FIFO策略

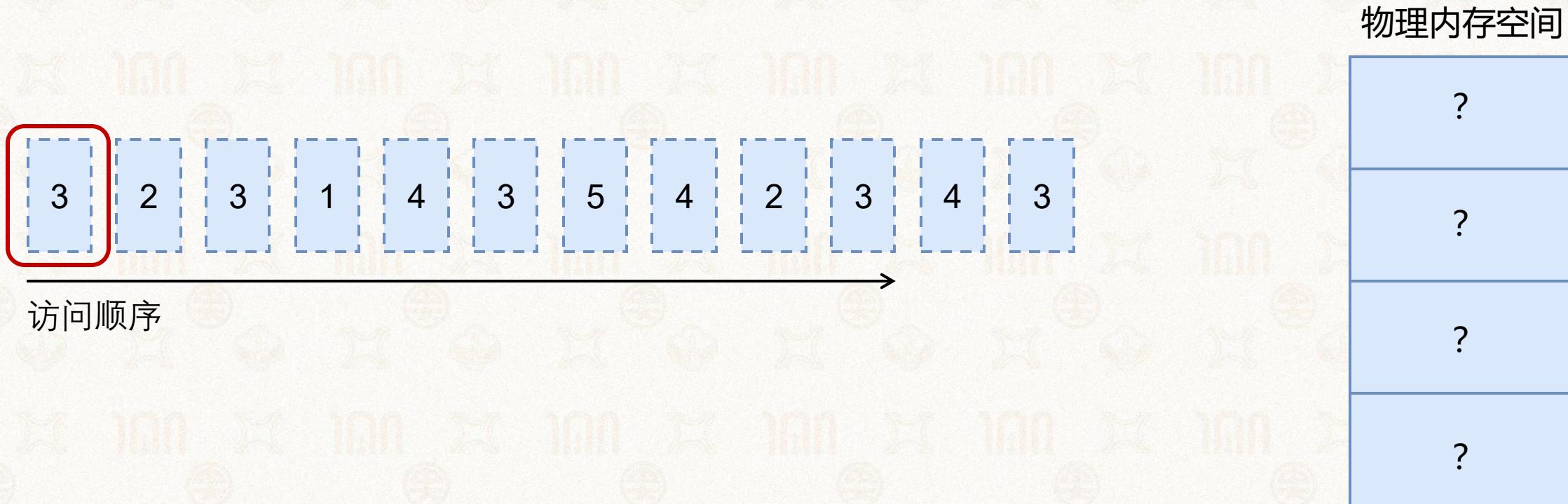
如果物理内存空间可以容纳4个物理页，那么执行完该策略之后，物理内存空间里的物理页号为多少？



作答

页替换策略：Second Chance策略

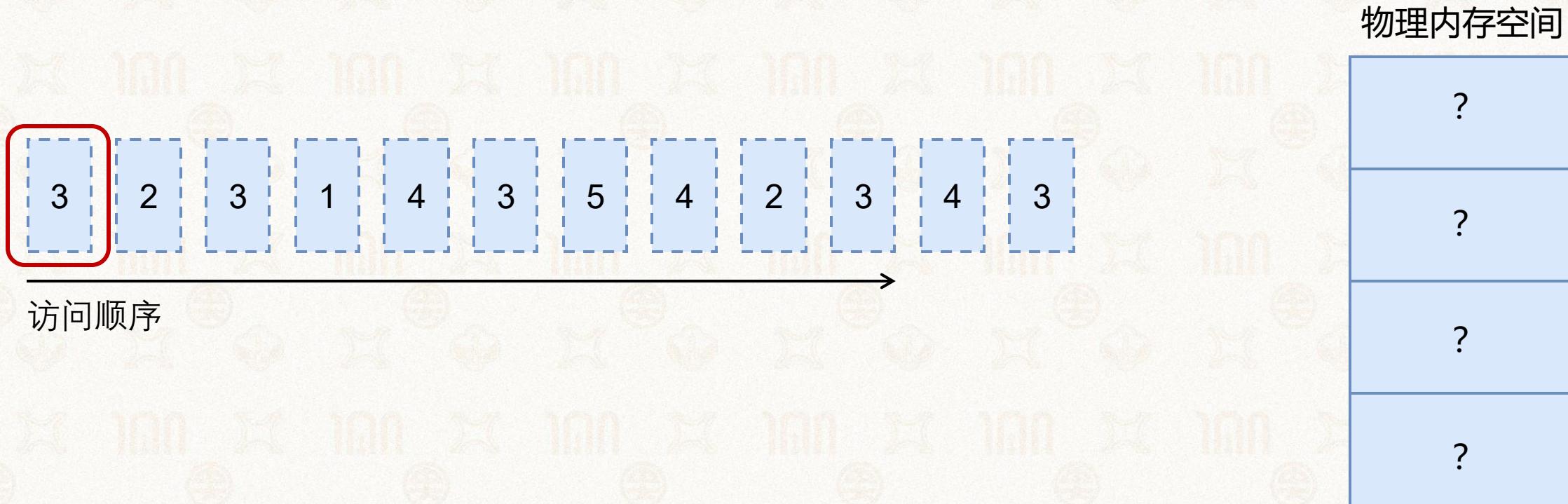
如果物理内存空间可以容纳4个物理页，那么执行完该策略之后，物理内存空间里的物理页号为多少？



作答

页替换策略：LRU策略

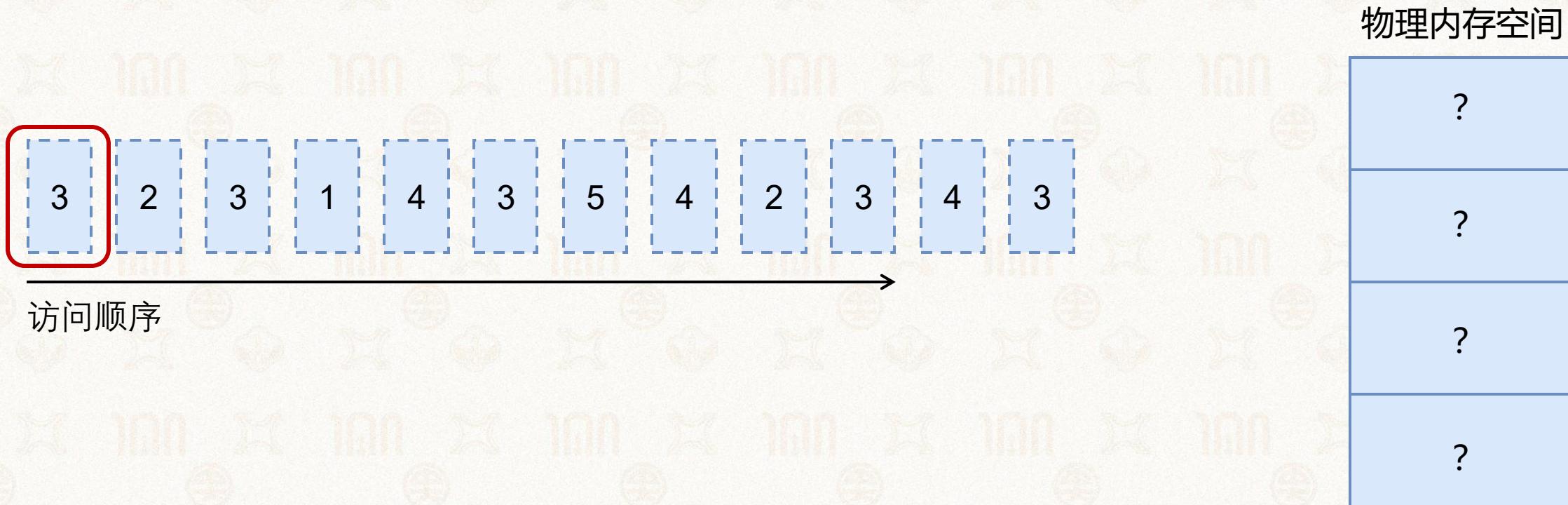
如果物理内存空间可以容纳4个物理页，那么执行完该策略之后，物理内存空间里的物理页号为多少？



作答

页替换策略：MRU策略

如果物理内存空间可以容纳4个物理页，那么执行完该策略之后，物理内存空间里的物理页号为多少？



作答

关于这节课，有什么疑问？

作答



1924-2024
中山大學 世纪华诞
100th ANNIVERSARY
SUN YAT-SEN UNIVERSITY

1924-2024

谢谢

微信: suyuxin

钉钉: 苏玉鑫

B站: <https://space.bilibili.com/502854403>

软工集市课程专区: <https://ssemarket.cn/new/course>

匿名提问箱: <https://suask.me/ask-teacher/106/苏玉鑫>