

****

《实用操作系统》实验报告

（六）

**姓　　名 ：宋泽涛**

**学　　号 ：25120222201292**

**学 院 ：信息学院**

**专 业 ：软件工程**

**2024年 11 月**

# 实验目的

* 理解内存页面调度的机理。
* 掌握几种理论页面值换算法的实现方法
* 通过实验比较各种调度算法的优劣。

# 实验环境

* PC + Linux Red Hat操作系统 + GCC
* 或 Windows + VC

# 实验内容

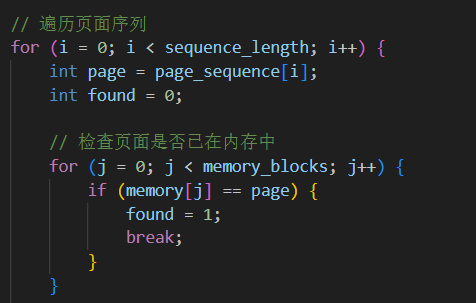
随机给出一个页面执行序列，如：1,5,3,4,2,1,3,4,5,7,9,……。要求计算以下几种置换算法的缺页数、缺页率和命中率。

* + - 最佳置换算法OPT（Optimal）
    - 先进先出算法FIFO（First In First Out）
    - 最近最少使用算法LRU（Least Recently Used）

# 实验具体实现

1.最佳置换算法OPT（Optimal）

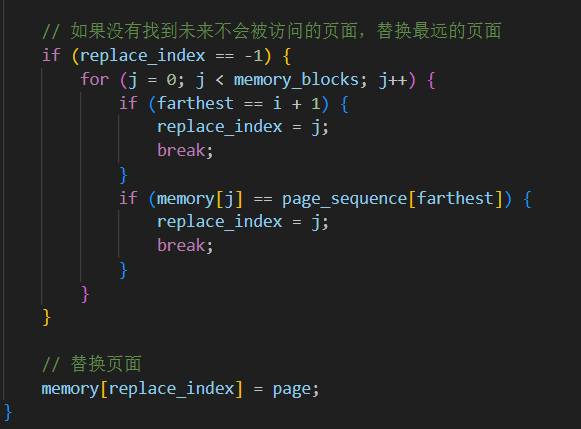
首先遍历页面序列，检查页面是否在内存中



如果不在内存中，就说明发生缺页，并进行页面替换

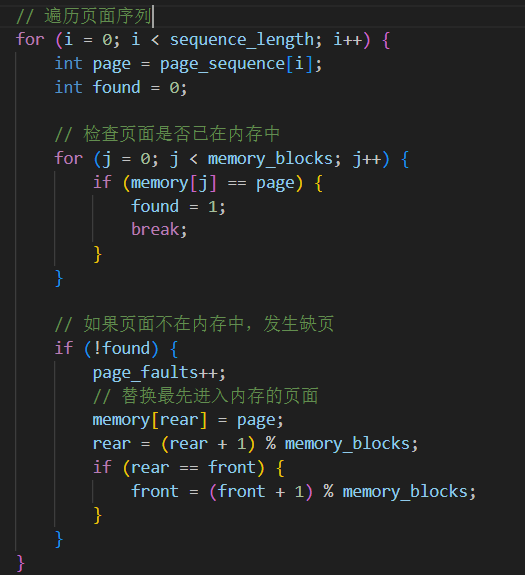


没有找到的话就替换最远的页面



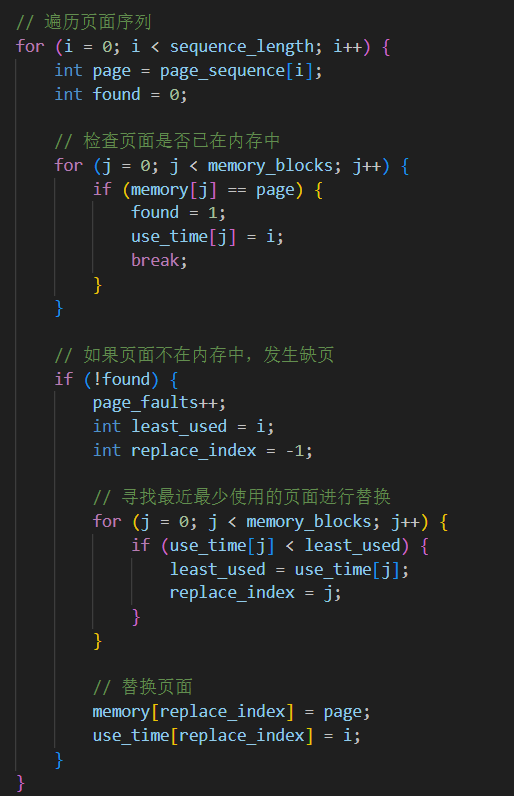
2.先进先出算法FIFO

遍历序列的时候寻找最新进入内存的页面即可

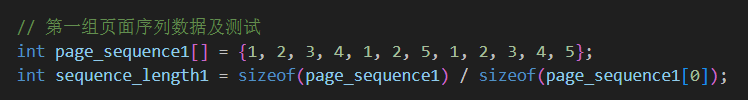


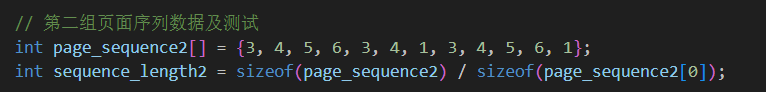
3.最近最少使用算法LRU

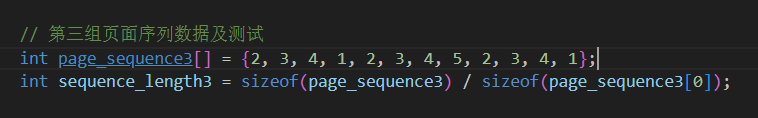
和FIFO类似，只需要将判断条件改为最近最少的页面即可

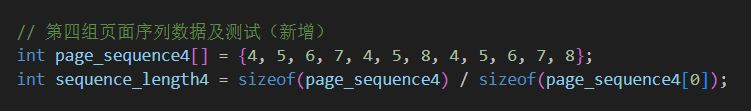


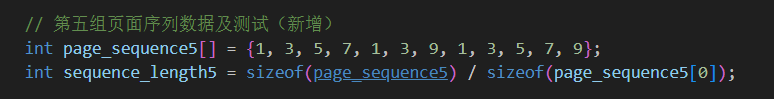
测试数据如下：











输出结果如下：





4.分析

**最佳置换算法（OPT）分析：**

原理：最佳置换算法会预测未来页面访问序列，选择在未来最长时间内不会被访问的页面进行替换（当发生缺页且内存已满时）。它是一种理想情况下的算法，因为在实际系统中很难准确预测未来的页面访问情况。

对于第一组页面序列，缺页数为 7，缺页率为 0.58，命中率为 0.42。这意味着在处理该页面序列时，按照最佳置换算法的策略，有 7 次出现了要访问的页面不在内存中的情况。由于总页面访问次数为 12（由页面序列数组长度可知），所以缺页率为 7÷12 ≈ 0.58，命中率则为 1 - 0.58 = 0.42。

同理，对于其他组页面序列，如第二组、第四组和第五组，缺页数也为 7，缺页率为 0.58，命中率为 0.42。这表明在这些页面序列的访问模式下，最佳置换算法的表现相对稳定，可能是因为这些序列存在一些相似的页面访问规律，导致按照预测未来的方式选择替换页面时，出现了类似数量的缺页情况。

而对于第三组页面序列，缺页数为 8，缺页率为 0.67，命中率为 0.33。与其他几组相比，缺页情况略有不同，可能是第三组页面序列的访问模式存在一些特殊性，使得最佳置换算法在预测未来页面访问并选择替换页面时，难以更精准地避免缺页，从而导致缺页次数相对多了一些。

**先进先出算法（FIFO）分析**

原理：最近最少使用算法记录每个页面在内存中的最近使用时间，当发生缺页且内存已满时，选择最近最少使用的页面进行替换。

对于五组页面序列的测试结果，最近最少使用算法的缺页数大多为 10 或 12，缺页率相应地为 0.83 或 1.00，命中率为 0.17 或 0.00。比如，对于第一组页面序列，缺页数为 10，缺页率为 0.83，命中率为 0.17。这是因为该算法虽然关注了页面的最近使用情况，但在某些页面序列的访问模式下，可能会出现一些页面虽然最近使用过，但在后续一段时间内又长时间未被使用，而另一些页面则频繁被使用，导致按照 LRU 算法选择替换页面时，仍然会频繁出现缺页情况。

特别是对于第三组页面序列，缺页数达到了 12，缺页率为 1.00，命中率为 0.00。这说明在该页面序列的访问模式下，先进先出算法几乎每次访问页面都需要进行缺页处理，可能是因为页面的访问顺序与 FIFO 的替换策略严重不匹配，导致频繁地替换掉了后续还会马上用到的页面。

最近最少使用算法（LRU）分析

原理：最近最少使用算法记录每个页面在内存中的最近使用时间，当发生缺页且内存已满时，选择最近最少使用的页面进行替换。

对于五组页面序列的测试结果，最近最少使用算法的缺页数大多为 10 或 12，缺页率相应地为 0.83 或 1.00，命中率为 0.17 或 0.00。比如，对于第一组页面序列，缺页数为 10，缺页率为 0.83，命中率为 0.17。这是因为该算法虽然关注了页面的最近使用情况，但在某些页面序列的访问模式下，可能会出现一些页面虽然最近使用过，但在后续一段时间内又长时间未被使用，而另一些页面则频繁被使用，导致按照 LRU 算法选择替换页面时，仍然会频繁出现缺页情况。

同样，对于第三组页面序列，缺页数为 12，缺页率为 1.00，命中率为 0.00。这表明在该页面序列的访问模式下，最近最少使用算法也无法很好地适应，可能是因为页面的访问规律使得算法难以准确判断哪些页面是真正近期不会再被使用的，从而导致几乎每次访问页面都需要进行缺页处理。

# 实验分析与总结

1.最佳置换算法

OPT 算法由于能够提前知晓未来页面访问情况，所以在理论上可以做出最优的页面替换决策。然而，在实际系统中这种预测几乎是不可能实现的，它更多地作为一种性能评估的理想基准。从实验结果来看，尽管它在多数页面序列中表现相对较好，但仍然存在缺页现象。这是因为即使能够预测未来，页面序列的复杂性和不确定性仍然可能导致无法完全避免缺页。例如，当页面访问序列存在频繁的局部性变化或者较长的周期性重复但周期内页面分布不均衡时，OPT 算法也难以做到完美的页面替换规划。

2.先进先出算法

FIFO 算法仅依据页面进入内存的先后顺序进行替换，完全忽略了页面后续的使用情况。这使得它在面对页面访问序列中存在较多循环或者近期频繁再次使用的页面时，容易出现将后续仍需频繁访问的页面过早替换出去的情况，从而导致大量的缺页。例如，在一些测试序列中，页面可能按照特定的循环模式访问，而 FIFO 算法由于不考虑页面的 “热度”，按照固定的先进先出规则替换页面，使得内存中始终难以保留当前活跃的页面，缺页率居高不下。

3.最近最少使用算法

LRU 算法通过记录页面的最近使用时间来选择替换页面，相比 FIFO 算法更具智能性。它能够较好地适应具有局部性原理的页面访问模式，即近期使用过的页面在未来较短时间内再次被使用的概率较高。但是，当页面访问序列出现较长时间跨度的重复访问或者不规则的访问模式时，LRU 算法可能会误判页面的 “热度”。例如，某个页面在一段时间内被频繁访问后，长时间未被再次使用，但由于其最近使用时间相对其他页面仍然较新，在内存紧张需要替换页面时未被选中，而后续又突然再次被访问，导致缺页发生。