

实验报告

实验名称: 页面置换算法模拟

实验时间: 2018/5/8

实验人员: 李子强 (姓名) 115010352 (学号) 15 (年级)

实验目的: 掌握页面置换的原理, 深入理解不同策略之间的优势和劣势

实验环境: Linux

实验步骤:

1. 了解不同页面置换算法的细节与思想

2. 编写模拟程序

实验陈述:

1、基础知识回顾:

1. FIFO 置换算法的内容与复杂度分析: 置换最先调入内存的页面, 即置换在内存中驻留时间最久的页面。按照进入内存的先后次序排列成队列, 从队尾进入, 从队首删除。替换时间复杂度为 $O(1)$ 。

2. Min 置换算法的内容与复杂度分析: 置换以后不再被访问, 或者在将来最迟才回被访问的页面, 缺页中断率最低。但是, 这个是理论上在线算法的性能, 只有在离线的情况下能达到。可以以 $O(n)$ 的时间复杂度实现。

3. LRU 置换算法的内容与复杂度分析: 置换最近一段时间以来最长时间未访问过的页面。根据程序局部性原理, 刚被访问的页面, 可能马上又要被访问; 而较长时间内没有被访问的页面, 可能最近不会被访问。可以以 $O(n)$ 的时间复杂度实现。

4. Clock 置换算法的内容与复杂度分析: 为了节约 Second Chance 算法一个接着一个检查使用位的开销, 时钟轮转法又提出了改进。时钟轮转法将所有的页组成一个圆, 圆心的指针指向下一个要被置换的页面, 置换前同样检查使用位, 如果使用位为 1, 同样将其使用位置为 0, 随后将顺指针旋转, 检查下一个页面, 直到发现某页的使用位为 0, 将此页置换出内存。很容易理解此算法为什么叫“时钟”轮转法。可以以 $O(n)$ 的时间复杂度实现。

5. Second-chance 置换算法的内容与复杂度分析: 为了避免 FIFO 算法将重要的页换出内存, Second Chance 算法提供了一些改进。讲 FIFO 列表的一部分采用 LRU 模式, 踢出最近不使用。结合了 LRU 和 FIFO 的优点。可以以 $O(n)$ 的时间复杂度实现。

2、理论基础

页面置换理论最优的算法是: Min

请给出相关证明 Min 是 OPT 算法, 因为是离线算法, 运行时知道所有将来的数据。每次替换页面操作都可选出将来最晚使用的页面替换。

FIFO 算法是否有提升的空间? 如果有请提出方案, 没有请给出证明。查找, 更新时间优化为 $O(1)$, 加入哈希表和双向链表另外存储页面数据。

LRU 算法是否有提升的空间? 如果有请提出方案, 没有请给出证明。查找, 更新时间优化为 $O(1)$, 加入哈希表和双向链表另外存储页面数据。

3、遇到的问题与解决方法

问题 1: 算法逻辑错误

解决方法 排查临界值解决。

运行表格: (hit percentage) cache size: 123

Algorithms/test	1.in	2.in	3.in
FIFO	11.98%	11.85%	82.36%
MIN	42.4%	43.27%	88.58%
LRU	11.76%	11.85%	82.39%
Clock	11.93%	11.83%	82.38%
Second-chance	11.85%	11.85%	82.39%

实验总结:

学习不同页面置换算法的细节与思想。