山东大学 软件 学院

**操作系统** 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202200300092 | 姓名：郑睿杰 | | 班级：大数据2班 |
| 实验编号：实验7 | | | |
| 实验题目：内存页面置换算法实验 | | | |
| 实验学时：4 | | 实验日期：2024.6.5 | |
| 实验目的：  加深对于存储管理的了解，掌握虚拟存储器的实现原理；观察和了解重要的页面置换算法和置换过程。练习模拟算法的编程技巧，锻炼分析试验数据的能力。 | | | |
| 硬件环境：  机型：ThinkBook-13x-ITG  CPU：11th Gen Intel(R) Core(TM) i7-1160G7 @ 1.20GHz  内存：16150988 kB | | | |
| 软件环境：  Ubuntu 9.4.0-1 ubuntu1~20.04.2 + vscode2022 | | | |
| 实验步骤与内容：  【实验模型描述】  本实验描述了几种主要的内存调度的页面置换算法。主要原理如下：  FIFO：最早进入队列的页面最先被置换  LRU：最长时间未被使用的页面先被置换  Clock：通过一个指针按照顺时针方向扫描页面，如果页面被访问过，将其位重新设置为1；否则，将其位设置为0，这样第二次扫描到的时候就会被替换掉。  EClock：在该算法中，每个页面除了引用位外，还增加了一个修改位用来表示页面是否被修改过。该算法通过优先替换引用位和修改位都为0的页面来提高性能。  LFU：选择使用次数最少的页面进行替换。  MFU：选择使用次数最多的页面进行替换。  【主要算法代码分析】  Makefile:    Myvmpr.h:    Myvmpr.cc:        【运行结果】    【实验结果的分析综合】  先进先出（FIFO）置换算法：  优点：简单易实现。  缺点：可能导致“先来后到”问题，不考虑页面的访问频率和重要性。  最近最旧未用（LRU）置换算法：  优点：考虑了页面的使用情况，通常能够保证较好的性能。  缺点：偶发性的、周期性的批量查询操作（包含冷数据）会淘汰掉大量的热点数据，导致 LRU 命中率急剧下降，缓存污染情况比较严重。  时钟（二次机会）置换算法：  优点：相对于LRU算法，实现更加简单，减少了维护开销。  缺点：不能完全准确地反映页面的访问情况，可能会有一定的误判。  增强二次机会置换算法：  优点：在时钟算法的基础上考虑了页面的访问次数，更加细致地调整页面的替换策略。  缺点：可能会增加额外的计算开销。  最不经常使用（LFU）置换算法：  优点：考虑了页面的使用频率，能够优先替换很少被使用的页面。  缺点：最近加入的数据总是易于被剔除（缓存末端抖动），因为他起始的频率很低。它无法对一个拥有最初高访问率之后长时间没有被访问的条目缓存负责。  最经常使用（MFU）置换算法：  优点：优先保留频繁被使用的页面，适用于某些特定的应用场景。  缺点：可能会导致一些长时间不再使用的页面无法被替换。  每种置换算法都有其适用的场景和局限性，选择合适的置换算法取决于具体的应用需求和系统特点。在实际应用中，通常需要根据具体情况进行权衡和选择。 | | | |
| 结论分析与体会：  【实验中遇到的问题和解决方法】  1.执行算法时遇到段错误。解决方法：init阶段初始化指针数组 | | | |