山东大学 软件 学院

**操作系统** 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100300063 | 姓名：李彦浩 | | 班级：软件菁英班 |
| 实验编号：实验*2* | | | |
| 实验题目：进程调度算法实验 | | | |
| 实验学时：2 | | 实验日期：2023.05.1 | |
| 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：  ThinkPad 8核 Intel Core i5-8265U 1.60GHz 8G内存  山大云算力平台 devlinux 1核 1G内存 | | | |
| 软件环境：  Vmware workstation;Ubuntu 20.04 | | | |
| 实验步骤与内容：  独立实验要求在以上示例实验程序中补充“增强二次机会”等置换算法的模拟程序。输入不同 的内存页面引用串和实存帧数，观察并分析其页面置换效果和性能，并将其与 LRU 和 FIFO 算法进行比较。  仿照示例程序给出的代码，首先完成Clock函数，也即二次机会算法。根据这个算法的定义，我需要先在实存中已有的所有页号，如果有匹配的，那么不需要置换，同时把reference\_bit置为1。如果没有匹配的，那么需要从头重新查找第一个reference\_bit为0的元素，并把沿途所有reference\_bit为1的页面的r\_bit全部置为0。这个过程持续到找到这个元素为止。然后替换这个页面，并把reference\_bit置为1。  首先是暴力查找是否有匹配的页面，如果有匹配可以复用。这一块儿的代码在扩展的四个函数里（Clock,Eclock,Lfu,Mfu）都是适用的，下面几个函数的说明不再赘述。    如果没有匹配到已有的页面，需要替换。注意“替换”可能也是添加，即分配的帧数如果没有完全使用完，（有空闲的）需要直接占用这些帧直到占满需要替换为止。    当需要替换的时候，按照我们的算法，建立标记位，这样做的目的是，可能出现所有页面的引用位都为1，一轮是找不到第一个引用位为0的元素的，需要重复这个过程直到找到为止。    随后记录淘汰页面，替换页面，置引用位。    随后是分析报告算法的性能，这部分和示例里的代码一致，不赘述了。  第二个算法是增强二次机会，实际上就是多加了一位modify\_bit修改位，这里(0,0)的被替换优先级要高于(0,1)，因为替换(0,1)的页面还需要把数据写回外存，效率上更低。所以唯一不同的地方就是要做这个优先级的判断。  在判断是否有(0,0)时，我们需要暴力遍历，直到发现第一个。但在这途中可能有(0,1)的页面，但这时不一定置换，(不清楚后面是否有(0,0))因此需要一些trick来分析是否要置换(0,1)。如果我们遍历了所有页面，都没有发现(0,0)，那么所有页面里就只有(0,1)，我们可以直接找第一个(0,1)去置换。    这里round\_already就是记录是否已经走过一轮的标记位，如果走了一轮，再找到第一个(0,1)准备置换。如果直接找到了(0,0)，那么直接置换。  需要注意的是，由于没有用户交互的部分，我只能暂定修改位变化的方式是在页面匹配复用时，默认修改了数据。    第三个算法是Lfu，最少频次替换算法。实际上Mfu(最大频次替换算法)和Lfu在本质上是一模一样的，只是判定时的小于号变成了大于号而已，因此这里只介绍Lfu。Lfu在需要置换时，先查找被引用次数最少的页面，这里我选择了线性查找。    其余部分都很常规，不介绍了。  测试时，在ubuntu下terminal中gmake写好的Makefile。    随后运行./vmrp，测试用例为示例中的belady串。    之后是四个需要自己实现的算法的输出结果与性能分析。(这里输出太长了，就放一部分)  Clock:    Eclock:    Lfu:    Mfu:    实验中我采用数组模拟了内存页面置换算法中的Clock,Eclock,Lfu,Mfu四个算法，分别对其进行性能分析，和Lru还有Fifo进行了比对。 | | | |
| 结论分析与体会：  实验中我遇到一个bug，即一开始Clock进入了死循环。这是因为我一开始在帧空闲时没有直接占用，二是走了二次机会算法，这样程序就会一直寻找第一个引用位为0的页面，但这个时候帧数组中又没有页面，就陷入了死循环。之后我改为，如果有空闲，则直接占用；没有空闲，才走置换算法。完成了debug。  实验结果在上一个部分中已经列出，与预测的结果一致，复现了设计目标。  实验中我采用数组模拟了内存页面置换算法中的Clock,Eclock,Lfu,Mfu四个算法，分别对其进行性能分析，和Lru还有Fifo进行了比对。加深了对于存储管理的了解，掌握了虚拟存储器的实现原理 | | | |