**实验四 存储管理**

# 姓 名 徐昊博 学 号 21013134 成绩

实验时间2023年11月23日指导教师(签名)

**（诚信声明：本实验报告内容，均由本人亲自上机完成。 签名：\_\_\_\_\_\_\_\_\_\_\_\_\_）**

一．实验目的

1.通过请求页式存储管理中页面置换算法模拟设计，了解虚拟存储技术的特点

2.掌握请求页式存储管理的页面置换算法。

二．实验工具与设备

装有 Linux 操作系统的计算机。

三．实验内容

计算并输出下面各种算法在不同内存容量下的命中率：

1. FIFO（先进先出算法）

算法代码如下：



1. LRU （最近最少使用算法）

算法代码如下：



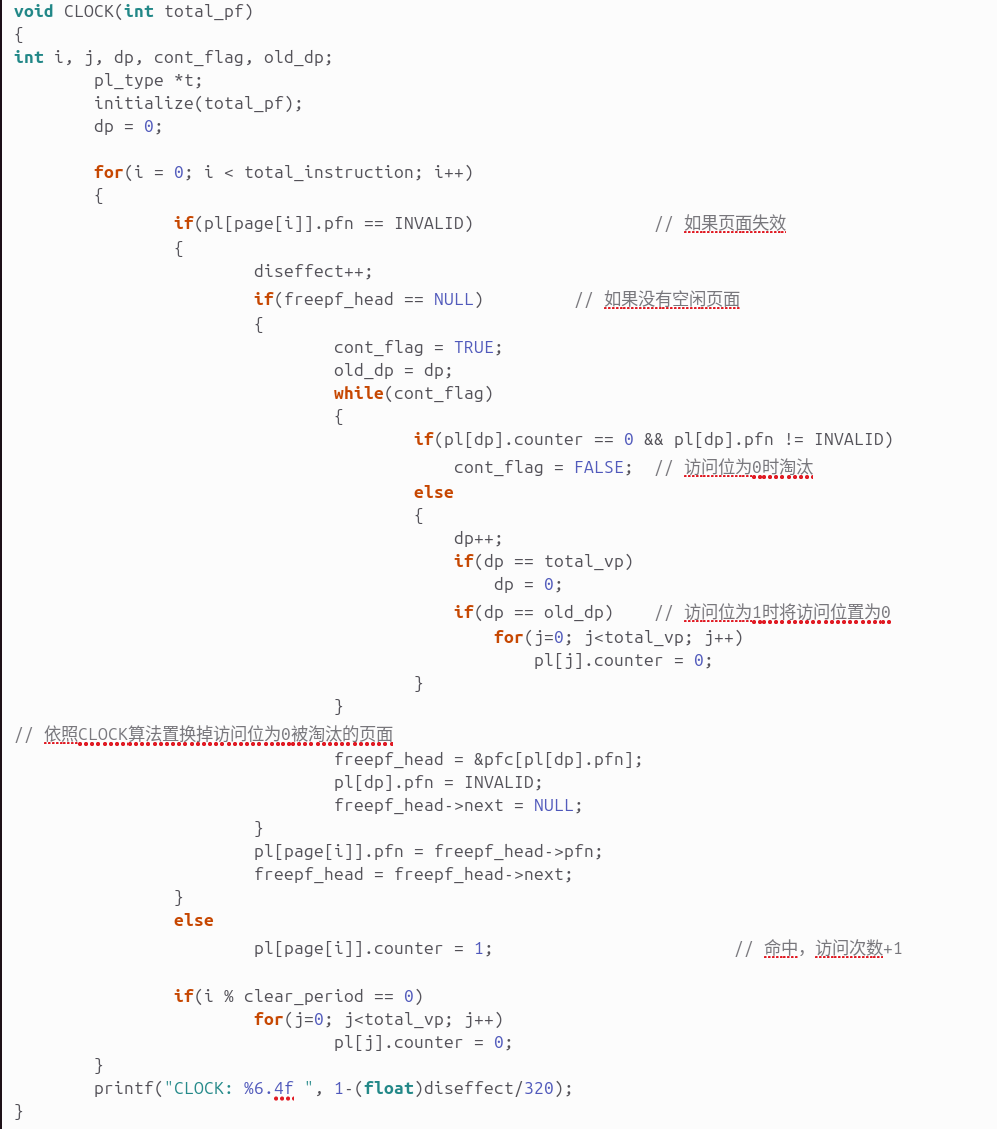
1. OPT （最优算法）



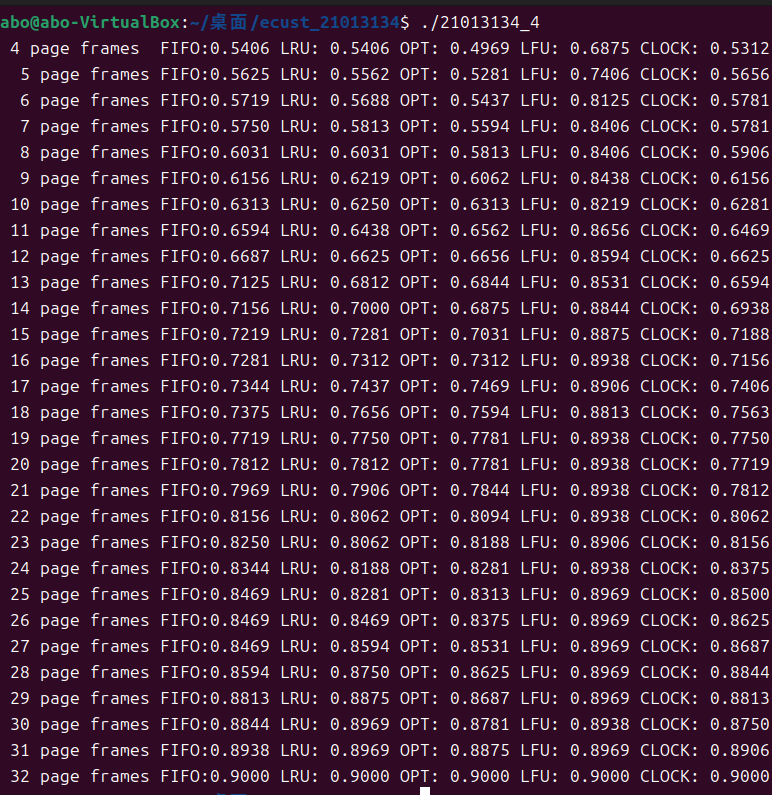
1. LFU （最少使用页面算法）



1. CLOCK （时钟算法）



运行结果如下：



可以看到随着页面数的增加，各个算法的命中率都有明显提高，最高在32页时达到了90%。

# 四．思考题

1. 为什么要进行内存管理，虚拟存储器的特点是什么？

内存管理是计算机系统中非常重要的一部分，它涉及管理计算机系统中的内存资源，以便有效地使用和分配内存。主要目的包括：

（1）资源分配：内存管理帮助操作系统有效地管理计算机的内存资源，确保每个正在运行的程序都能够获得所需的内存空间，避免因为内存不足导致程序运行出错或崩溃。

（2）内存保护：通过内存管理，操作系统可以实现内存保护机制，确保不同程序之间的内存空间相互隔离，防止程序越界访问或者恶意程序干扰其他程序的内存空间。

（3）内存共享：某些情况下，不同的程序可能需要共享内存，内存管理可以实现对共享内存区域的管理，使得多个程序能够安全地共享同一块内存区域。

虚拟存储器是一种扩展了计算机内存的概念，它的特点包括：

（1）虚拟地址空间：虚拟存储器允许每个程序拥有自己的虚拟地址空间，使得每个程序认为它拥有整个系统的内存空间，而不需要实际的全部内存支持。

（2）分页和分段：虚拟存储器通常采用分页或分段的方式将程序的地址空间划分成小块或段，当程序执行时，只有实际需要的部分会被加载到物理内存中。

（3）内存交换：虚拟存储器允许操作系统根据程序当前运行的需求将数据从磁盘交换到内存中，并在需要时将不再需要的数据移出内存，这样可以有效地利用有限的物理内存。

（4）更大的地址空间：通过虚拟存储器，系统可以提供比实际物理内存更大的地址空间，允许程序处理比实际物理内存更大的数据集。

1. 几种内存管理算法有何区别与联系？

这些算法是用于操作系统中的页面置换算法，用于管理虚拟内存系统中的页面置换，确保在物理内存不足时，合理地选择哪些页面被淘汰，哪些页面被保留。它们之间的区别在于选择淘汰页面的依据和策略。

FIFO：按照页面进入内存的先后顺序进行淘汰，最先进入的页面最先被淘汰。算法简单易实现，但可能会出现Belady异常（即增加页面数时缺页率反而增加）。

LRU：选择最近最久未使用的页面进行淘汰，即最近一段时间内没有被访问过的页面被淘汰。LRU通常能够较好地模拟人类访问内存的行为，但实现起来可能较为复杂，需要记录页面的访问时间戳或使用栈等数据结构。

OPT：理论上是最佳的页面置换算法，它淘汰未来最长时间内不会被访问的页面。实际不可行\*\*：但是实现起来不现实，因为需要未来的访问情况预测，通常用于评估其他算法的性能表现。

LFU：根据页面被访问的频率来进行淘汰，选择最不经常被访问的页面进行淘汰。需要记录每个页面的访问频率，实现相对复杂，并且可能因为一些页面在初始阶段频繁访问而长期得不到淘汰。

CLOCK：也称为“时钟”算法，通过设置一个指针，模拟一个环形链表，当页面被访问时，将其对应的位设置为1；当需要淘汰页面时，检查指针位置的页面位，如果为0则选择淘汰，如果为1则将位清零并将指针向前移动。相比于LRU，CLOCK算法的实现相对简单，并且在性能上与LRU有一定的接近。

这几个内存管理算法（FIFO、LRU、OPT、LFU、CLOCK）之间存在联系和相似之处，主要体现在它们对于页面置换的处理方式和对内存中页面的管理。

1. 页面置换策略：这些算法都是为了解决当内存不足时，需要选择哪些页面淘汰出内存的问题而设计的。它们从不同的角度出发，选择要淘汰的页面，比如按时间顺序（FIFO、LRU、CLOCK）、按访问频率（LFU）、或理想情况下最佳化地选择（OPT）。

2. 管理内存页面：这些算法都涉及对内存中页面的管理，需要维护页面的访问时间、频率或者其他信息。无论是记录页面进入内存的时间顺序、记录最近访问时间（LRU）、统计访问频率（LFU）还是维护页面的访问位（CLOCK），它们都需要对页面状态进行跟踪和更新。

3. 解决内存不足问题：所有这些算法都旨在解决内存不足时的页面置换问题。它们的目标都是在保证程序运行正确性的前提下，尽可能减少页面淘汰对程序性能的影响。