**实验四 存储管理**

# 姓 名 蒋文强 学 号 21013017 成绩

实验时间2023.11.23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

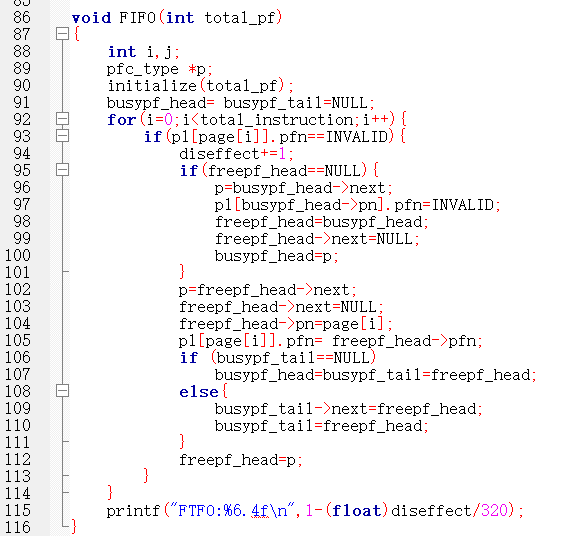
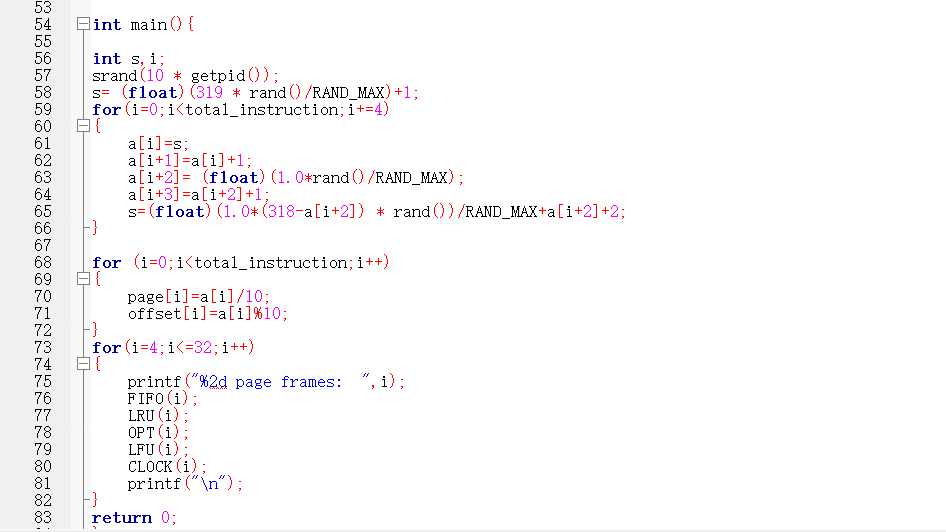
装有 Linux 操作系统的计算机。

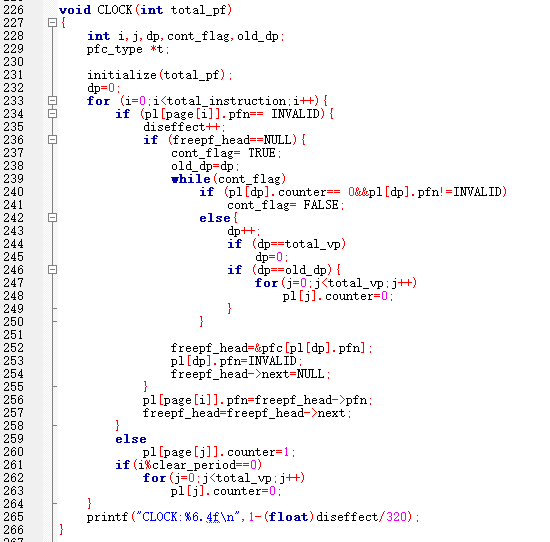
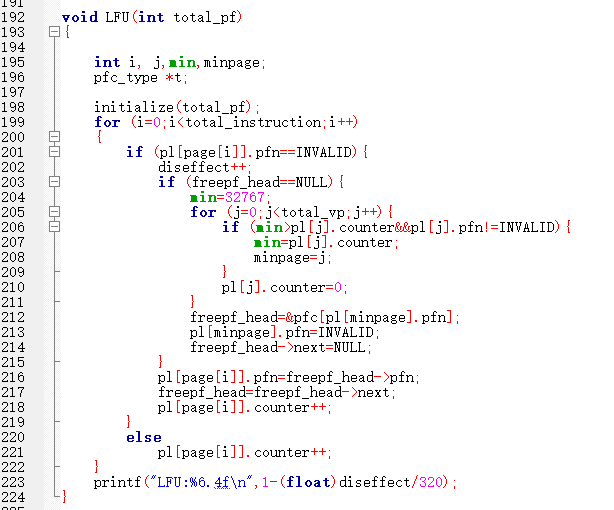
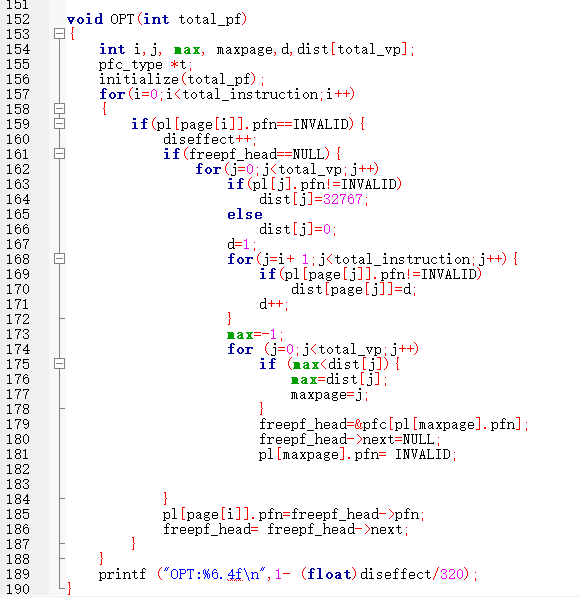
三．实验内容

计算并输出下面各种算法在不同内存容量下的命中率：

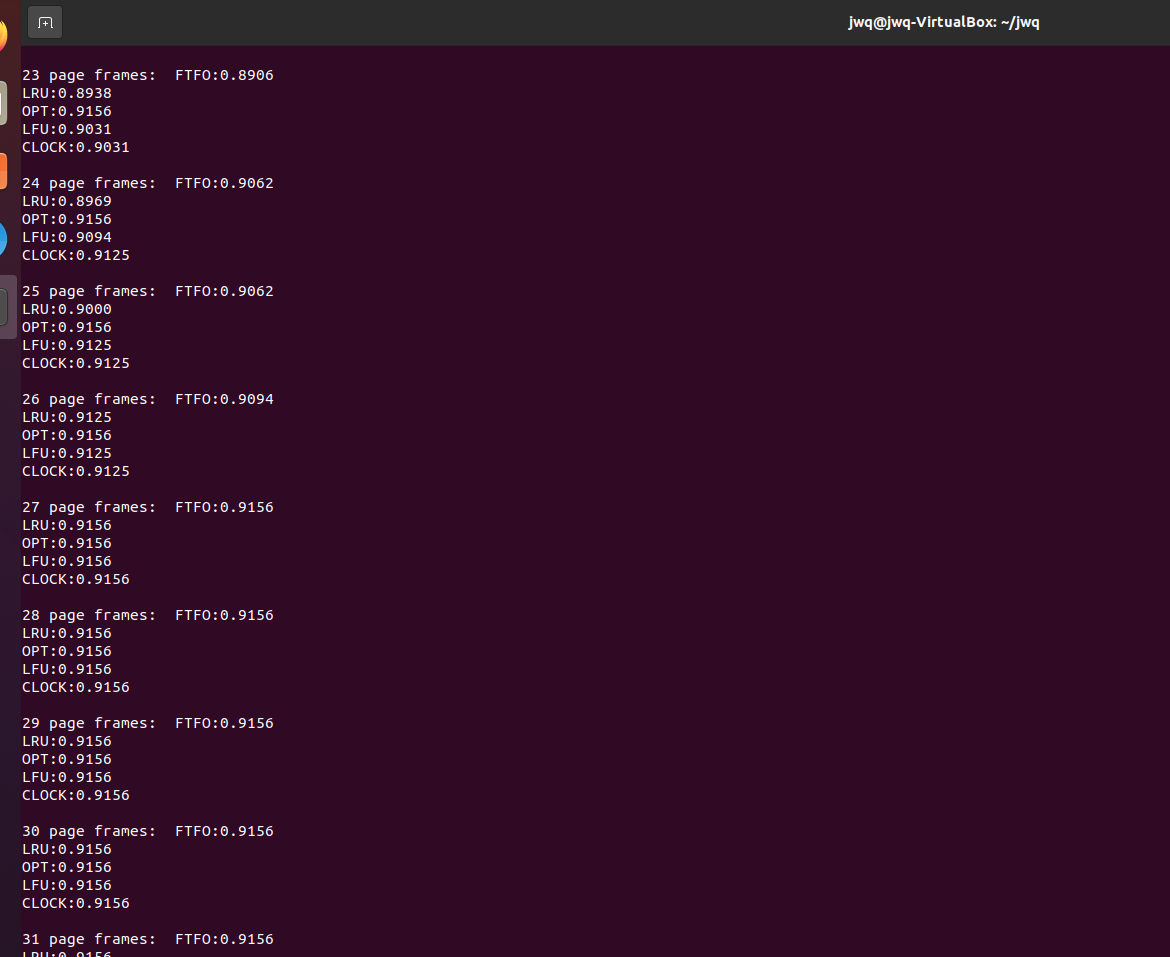
1. FIFO（先进先出算法）
2. LRU （最近最少使用算法）
3. OPT （最优算法）
4. LFU （最少使用页面算法）
5. CLOCK （时钟算法）

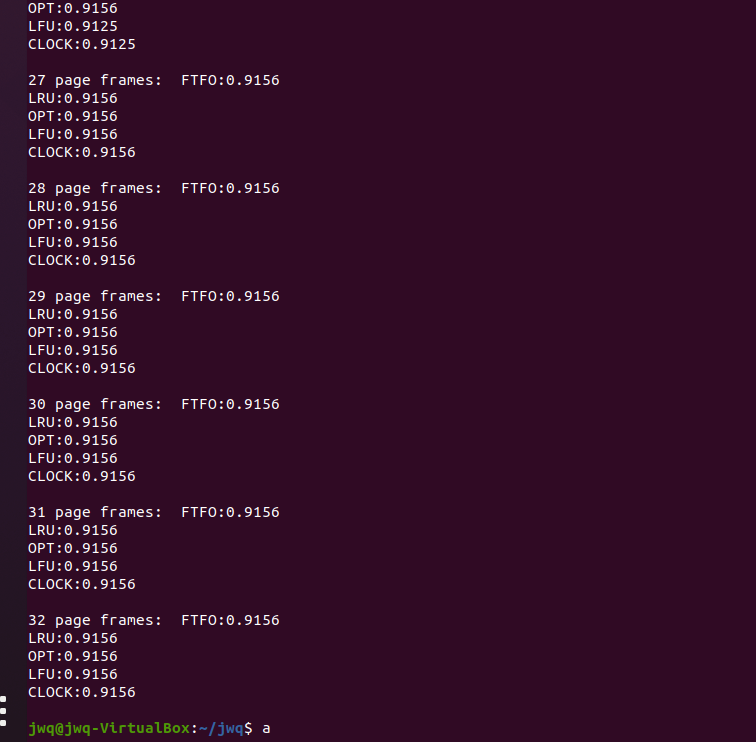
**具体代码如下：**





**运行结果如图**





# 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？

资源分配和释放： 内存管理允许操作系统有效地为正在运行的程序分配内存空间，并在程序执行完毕后释放这些空间。这确保了系统资源的有效利用。

资源分配和释放：内存管理允许操作系统有效地为运行的程序分配所需的内存空间，并在程序执行完毕后将这些空间释放。这确保了系统资源的有效利用，防止程序在执行过程中因为内存不足而崩溃。

多道程序设计：内存管理使得多个程序可以同时存在于内存中，实现了多道程序设计。通过合理地切分和分配内存，操作系统可以在同一时间内运行多个程序，提高了计算机系统的吞吐量和效率。

进程间通信：在多任务和多进程的系统中，进程之间可能需要进行通信。共享内存是一种实现进程间通信的方式，而内存管理就是负责管理这些共享内存区域的分配和访问。

虚拟存储器：虚拟存储器是一种扩展计算机内存的技术，它允许程序使用比物理内存更大的地址空间。虚拟存储器通过将部分程序或数据存储在辅助存储器（如硬盘）上，从而释放物理内存，使得系统可以同时运行更多的程序。这样，即使程序的大小超过了物理内存的限制，也可以在需要时将其加载到内存中执行。

关于虚拟存储器的特点：

地址空间扩展：虚拟存储器允许每个程序拥有更大的地址空间，这样程序可以被设计成比实际可用的物理内存更大。

分页和分段：虚拟存储器通常通过分页或分段的方式来实现。分页将程序和数据划分成固定大小的页，而分段则将其划分成逻辑上相关的段。这些技术使得程序可以被部分加载到内存中，而不必全部加载。

页面置换：当物理内存不足时，虚拟存储器系统可以通过页面置换将不常用的页面移到辅助存储器中，从而腾出空间给新的页面。

透明性：对于程序而言，虚拟存储器的使用是透明的，程序员无需过多关注实际的物理内存情况。这提供了更高的抽象级别，简化了程序设计。

1. 几种内存管理算法有何区别与联系？

FIFO（先进先出算法）：

策略： 将最早进入内存的页面置换出去。

实现： 使用一个队列来维护页面的顺序，当需要进行页面置换时，选择队列头部的页面进行置换。

特点： 简单易实现，但可能会导致“先进入”的页面不一定是最近被使用的，因此可能不够智能。

LRU（最近最少使用算法）：

策略： 选择最近最久未使用的页面进行置换。

实现： 通过记录每个页面的访问时间戳或使用计数来判断哪个页面最久未使用。

特点： 相对较智能，但实现相对复杂，需要维护额外的数据结构。

OPT（最优算法）：

策略： 置换将来最长时间内不被访问的页面。

实现： 需要预知未来的页面访问情况，实际上是一个理论上的算法，因为在实际中难以准确预测未来访问情况。

特点： 置换效果最好，但无法在实践中完全实现。

LFU（最少使用页面算法）：

策略： 选择使用频次最低的页面进行置换。

实现： 维护每个页面的使用计数，选择使用频次最低的页面进行置换。

特点： 对于具有明显的使用频次模式的系统比较有效，但可能对于突发性的使用模式表现不佳。

CLOCK（时钟算法）：

策略： 类似于FIFO，但是通过使用一个类似时钟指针的数据结构，检查每个页面的访问位，以确定置换的页面。

实现： 维护一个环形链表，指针按照顺时针方向移动，当需要进行页面置换时，选择指针指向的页面。

特点： 比FIFO稍智能，但相对简单，适用于实现相对较为简单的时钟置换算法。

**联系：**

这些算法都是为了解决在有限的物理内存空间中，如何合理地选择要保留的页面和置换的页面的问题。

它们都属于页面置换算法的范畴，目标是优化程序的页面命中率，提高内存的利用率。