

# Lab2挑战性任务

---

冯文韬 2021.6.7

## 任务1：页面锁机制

难度：★ ~ ★★★

Linux系统中有着mlock机制，即对特定的物理页面加锁，使其不会被操作系统页面置换到交换空间（swap space）。被加锁的页面会一直驻留在内存中。在这部分中，我们需要在MOS操作系统中实现页面锁机制。

### 具体任务

- 实现四个页面锁函数
  - `int mlock(u_long addr, size_t len);`
  - `int munlock(u_long addr, size_t len);`
  - `int mlockall(int flags);`
  - `int munlockall(void);`
  - 以上四个函数具体的原理和功能请各位同学自行查阅资料，同时实现的方法也不作限制，只要完成功能即可。
- 自行构造测试样例以展示完成效果

### 基本功能要求

- 一个页面只能被加一次锁。
- 已经被多个虚拟页面映射的物理页面不能被加锁。
- 被加锁的页面不能被多个虚拟页面映射。
- 被加锁的页面不能被移除。
- 在MOS系统物理内存初始化结束后，需要对此时已经分配的所有物理页面加锁。

## 任务2：页面置换机制

难度：★★ ~ ★★★

页面置换机制是操作系统理论课中的一个重要部分，但是只学会了理论不一定真正理解了页面置换机制的实现细节。因此，在这部分就让我们一起探索实现一个在MOS系统中真正能运行的页面置换机制！

### 具体任务

- 自行设计实现细节以完成页面置换机制。
- 自行构造测试样例以展示完成效果。

### 基本功能要求

- 被加锁的页面不允许置换出去。
- 被换出的页面在被访问的时候能够换回内存。
- 被换回的页面信息不能缺失（如映射关系、页面内容等等）。
- 初始化时被分配的页面永远不会被换出。

## 特别说明

页面置换算法可以自行选择，完成功能要求即可。为了降低难度，可以直接从内存中划出一块存储空间作为交换空间（swap space）。

## 任务3：反置页表

难度：★★★★ ~ ★★★★★★

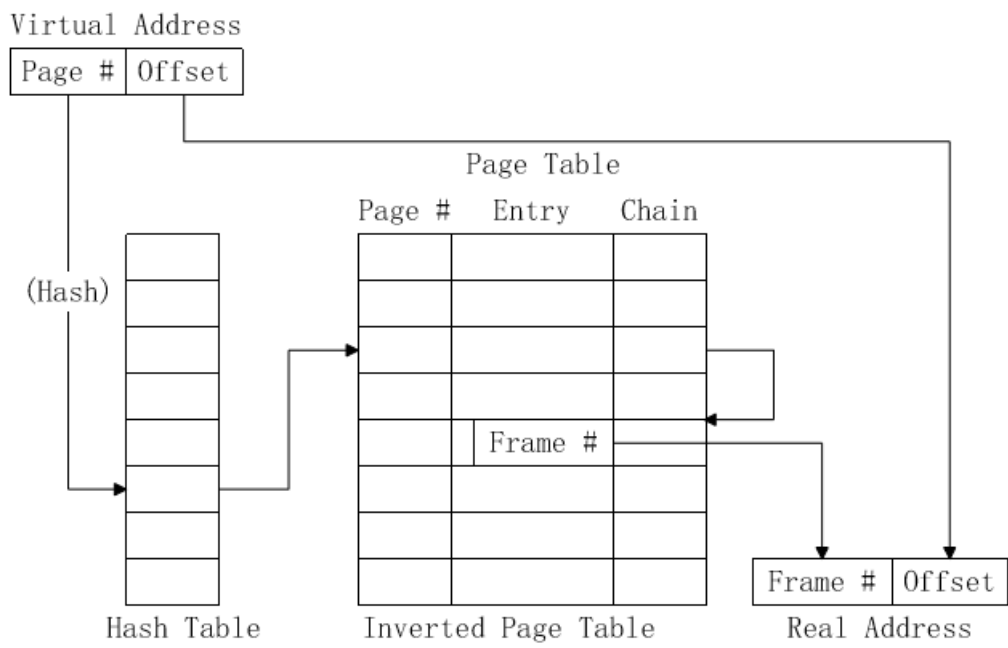
反置页表也是理论课所学的知识，现在我们在MOS系统中实现一个反置页表机制。

### 具体任务

- 根据理论课所学的反置页表知识，自行设计实现反置页表。
- 自行构造测试样例以展示完成效果。

### 基本功能要求

- 具有完整的内存管理功能，与原先MOS系统的内存管理功能和结果一致
- 需要按照理论课PPT中（下图所示）的架构来实现，实现细节可以自行设计，但是hash机制和Chain机制不能修改。



在Lab2的挑战性任务中，三个任务都为必做。在书写实验报告和准备申优答辩时，请加入以下内容：

- 清楚地叙述**设计思路和实现方法**
- 详细介绍构造的测试样例，并清楚地说明**每个样例对应测试的功能**
- 展示运行结果，根据结果分析说明功能的正确性