

# 实验4 虚拟内存管理

——Ucore实验3

- 本次实验是在前一实验的基础上，借助于页表机制和lab1中涉及的中断异常处理机制，完成Page Fault异常处理和FIFO页替换算法的实现。
- 实验原理最大的区别是在设计了如何在磁盘上缓存内存页，从而能够支持虚存管理，提供一个比实际物理内存空间“更大”的虚拟内存空间给系统使用。

# 实验目的

- 了解虚拟内存的Page Fault异常处理实现
- 了解页替换算法在操作系统中的实现

# 实验内容

- 本次实验是在UCORE实验二的基础上，借助于页表机制和UCORE实验一中涉及的中断异常处理机制，完成Page Fault异常处理和FIFO页替换算法的实现，结合磁盘提供的缓存空间，从而能够支持虚存管理，提供一个比实际物理内存空间“更大”的虚拟内存空间给系统使用。这个实验与实际操作系统中的实现比较起来要简单，不过需要了解UCORE实验一和UCORE实验二的具体实现。实际操作系统系统中的虚拟内存管理设计与实现是相当复杂的，涉及到与进程管理系统、文件系统等交叉访问。
- 可以尝试完成扩展练习，实现clock页替换算法（选做）。

# 练习

- 为了实现实验的目标，实验提供了3个基本练习和2个扩展练习，要求完成实验报告。
- 练习0：填写已有实验
- 练习1：给未被映射的地址映射上物理页（需要编程）
- 练习2：补充完成基于**FIFO**的页面替换算法（需要编程）
- 选做
  - 扩展练习 **Challenge 1**：实现识别dirty bit的 **extended clock**页替换算法（需要编程）
  - 扩展练习 **Challenge 2**：实现不考虑实现开销和效率的**LRU**页替换算法（需要编程）

# 练习0：填写已有实验

- 本实验依赖ucore实验1/2。请把你做的ucore实验1/2的代码填入本实验中代码中有“LAB1”,“LAB2”的注释相应部分。

# 练习1：给未被映射的地址映射上物理页

(需要编程)

- 完成do\_pgfault (mm/vmm.c) 函数，给未被映射的地址映射上物理页。设置访问权限 的时候需要参考页面所在 VMA 的权限，同时需要注意映射物理页时需要操作内存控制 结构所指定的页表，而不是内核的页表。注意：在LAB3 EXERCISE 1处填写代码。执行  
make qemu  
后，如果通过check\_pgfault函数的测试后，会有“check\_pgfault() succeeded!”的输出，表示练习1基本正确。
- 请在实验报告中简要说明你的设计实现过程。请回答如下问题：
  - 请描述页目录项（Page Directory Entry）和页表项（Page Table Entry）中组成部分对ucore实现页替换算法的潜在用处。
  - 如果ucore的缺页服务例程在执行过程中访问内存，出现了页访问异常，请问硬件要做哪些事情？

# 练习2：补充完成基于FIFO的页面替换算法

(需要编程)

- 完成vmm.c中的do\_pgfault函数，并且在实现FIFO算法的swap\_fifo.c中完成map\_swappable和swap\_out\_victim函数。通过对swap的测试。注意：在LAB3 EXERCISE 2处填写代码。执行make qemu后，如果通过check\_swap函数的测试后，会有“check\_swap() succeeded!”的输出，表示练习2基本正确。
- 请在实验报告中简要说明你的设计实现过程。
- 请在实验报告中回答如下问题：
  - 如果要在ucore上实现“extended clock页替换算法”请给你的设计方案，现有的swap\_manager框架是否足以支持在ucore中实现此算法？  
如果是，请给你的设计方案。  
如果不是，请给出你的新的扩展和基此扩展的设计方案。并需要回答如下问题：
    - 需要被换出的页的特征是什么？
    - 在ucore中如何判断具有这样特征的页？
    - 何时进行换入和换出操作？



# 对实验报告的要求：

- 最终提交pdf文件和源码
- 填写各个基本练习中要求完成的报告内容
- 完成实验后，请分析ucore\_lab中提供的参考答案，请在实验报告中说明你的实现与参考答案的区别
- 列出你认为本实验中重要的知识点，以及与对应的OS原理中的知识点，并简要说明你对二者的含义，关系，差异等方面的理解（也可能出现实验中的知识点没有对应的原理知识点）
- 列出你认为OS原理中很重要，但在实验中没有对应上的知识点