操作系统原理实验 （Lab 4）

# 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：

练习0：填写已有实验

本实验依赖实验1和3。请把你做的实验1和3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" 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？

作业提交要求：

1. 提交至<https://easyhpc.net/course/110>
2. 提交报告和修改后的kernel源代码压缩文件 .zip或者.rar，源代码要求编译后在能在虚拟机上运行
3. 截止日期为2021年6月5日23:59