操作系统原理实验 （Lab 1）

# 实验目的：

# 了解内核线程创建/执行的管理过程

# 了解内核线程的切换和基本调度过程

当一个程序加载到内存中运行时，首先通过ucore OS的内存管理子

系统分配合适的空间，然后就需要考虑如何分时使用CPU来“并发”执行多个程序，让每个运行的程序（这里用线程或进程表示）“感到”它们各自拥有“自己”的CPU。

本次实验将首先接触的是内核线程的管理。内核线程是一种特殊的进程，内核线程与用户进程的区别有两个：

内核线程只运行在内核态

用户进程会在在用户态和内核态交替运行

所有内核线程共用ucore内核内存空间，不需为每个内核线程维护单独的内存空间

而用户进程需要维护各自的用户内存空间

相关原理介绍可看附录B：【原理】进程/线程的属性与特征解析。

# 实验内容：

前期工作：

在虚拟机的ubuntu12.02中，执行 sudo apt-get install firefox

更新浏览器，然后在课程网站<https://easyhpc.net/course/110> 的编程作业lab2中下载lab2-sysu压缩文件，并在ubuntu12.02的文件系统里解压，以lab2-sysu压缩文件的kernel为基础，补充完整作业所需要的代码。

练习1：分配并初始化一个进程控制块（需要编码）

alloc\_proc函数（位于kern/process/proc.c中）负责分配并返回一个新的struct proc\_struct结构，用于存储新建立的内核线程的管理信息。ucore需要对这个结构进行最基本的初始化，你需要完成这个初始化过程。

【提示】在alloc\_proc函数的实现中，需要初始化的proc\_struct结构中的成员变量至少包

括：state/pid/runs/kstack/need\_resched/parent/mm/context/tf/cr3/flags/name。

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：

请说明proc\_struct中 struct context context 和 struct trapframe \*tf 成员变量含义和在

本实验中的作用是啥？（提示通过看代码和编程调试可以判断出来）

练习2：为新创建的内核线程分配资源（需要编码）

创建一个内核线程需要分配和设置好很多资源。kernel\_thread函数通过调用do\_fork函数完成具体内核线程的创建工作。do\_kernel函数会调用alloc\_proc函数来分配并初始化一个进程控制块，但alloc\_proc只是找到了一小块内存用以记录进程的必要信息，并没有实际分配这些资源。ucore一般通过do\_fork实际创建新的内核线程。do\_fork的作用是，创建当前内核线程的一个副本，它们的执行上下文、代码、数据都一样，但是存储位置不同。在这个过程中，需要给新内核线程分配资源，并且复制原进程的状态。你需要完成在kern/process/proc.c中的do\_fork函数中的处理过程。它的大致执行步骤包括：

1．调用alloc\_proc，首先获得一块用户信息块。

2．为进程分配一个内核栈。

3．复制原进程的内存管理信息到新进程（但内核线程不必做此事）

4．复制原进程上下文到新进程

5．将新进程添加到进程列表

6．唤醒新进程

7．返回新进程号

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：

请说明ucore是否做到给每个新fork的线程一个唯一的id？请说明你的分析和理由。

练习3：阅读代码，理解 proc\_run 函数和它调用的函数如何完成进程切换的。（无编码工作）

请在实验报告中简要说明你对proc\_run函数的分析。并回答如下问题：

在本实验的执行过程中，创建且运行了几个内核线程？

语句 local\_intr\_save(intr\_flag);....local\_intr\_restore(intr\_flag); 在这里有何作用?

请说明理由

完成代码编写后，编译并运行代码：make qemu

如果可以得到如 附录A所示的显示内容（仅供参考，不是标准答案输出），则基本正确。

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4月23日23:59