实验四：内核线程管理

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

实验内容：

实验2/3完成了物理和虚拟内存管理，这给创建内核线程（内核线程是一种特殊的进程）打下

了提供内存管理的基础。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

练习：

1. 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）

Context：Switch here to run process,推测应该是进程的上下文，用于进程切换。使用Switch.s汇编文件中的定义。

Tf：中断帧的指针，我的理解是：总是指向内栈核的某个位置 ，中断帧记录了进程在被中断前的某个状态。

1. 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。

在使用fork或clone系统调用时产生的进程均会由内核分配一个新的唯一的PID值。因为他的实现放在了互斥锁中。

1. 1在本实验的执行过程中，创建且运行了几个内核线程？

两个，idleproc：ucore第一个内核进程，完成内核各个子系统的初始化，之后立即调度执行其他进程；initproc：用于完成实验的进程而调度的内核进程。

3.2语句 local\_intr\_save(intr\_flag);....local\_intr\_restore(intr\_flag); 在这里有何作用?请

说明理由

保护进程切换不被中断，以免进程切换时其他进程再进行调度，相当于互斥锁。在添加进程列表的时候也需要这个操作，是因为进程进入列表的时候，可能会发生一系列的调度事件，加上这么一个保护机制可以确保进程执行不被打乱。

运行结果：

