LAB4 田新韬 2012011268

Ex0

先diff修改后的lab3与原lab3，再patch到lab4中

Ex1 初始化proc\_struct类型

按要求初始化即可..使用了memset初始化部分成员变量。

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保存了内核态kernel进程切换时可能会被修改的寄存器，保证各进程之间相互独立，trapframe保存了中断发生时进程进入中断处理例程前的状态，用于处理完中断后恢复现场。

Ex2 为线程分配资源

按照注释的提示编码，并且：

在分配内存空间时，若无可用内存，则fork\_out

无法分配内核栈，则bad\_fork\_cleanup\_proc

数据入栈时报错，则bad\_fork\_cleanup\_kstack

先关闭中断，再获取pid，proc入链表，再恢复中断，确保pid不会冲突。

Ex3 理解proc\_run

Proc\_run函数负责将CPU资源调度给指定线程：

void proc\_run(struct proc\_struct \*proc) {

if (proc != current) { // 判断是否需要切换

bool intr\_flag;

struct proc\_struct \*prev = current, \*next = proc;

local\_intr\_save(intr\_flag); // 关闭中断，确保原子操作

{

current = proc; // 更改当前运行进程

load\_esp0(next->kstack + KSTACKSIZE); // 设置TSS

lcr3(next->cr3); // 设置cr3

switch\_to(&(prev->context), &(next->context)); // 切换进程上下文

}

local\_intr\_restore(intr\_flag); // 恢复中断

}

}

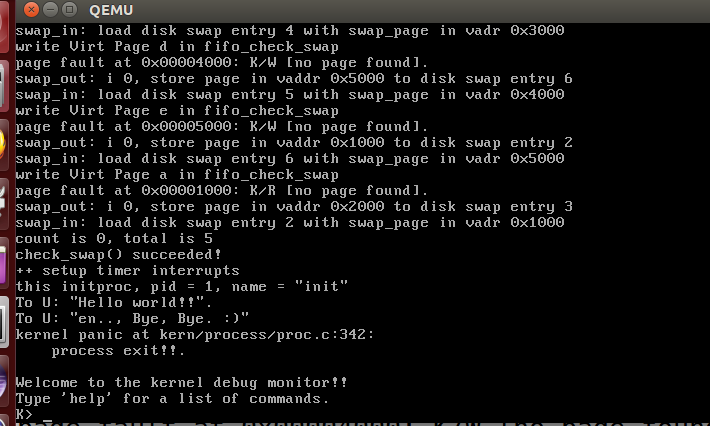
* 在本实验的执行过程中，创建且运行了几个内核线程？

两个：idleproc(id=0),initproc(id=1)

* 语句local\_intr\_save(intr\_flag);....local\_intr\_restore(intr\_flag);在这里有何作用?请说明理由

关闭中断，确保进程切换过程连续完成不被打断，保证其原子性。

实验结果 make qemu



make grade

