

## 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

```

QEMU
swap_in: load disk swap entry 4 with swap_page in vadr 0x3000
write Virt Page d in fifo_check_swap
page fault at 0x00004000: K/W [no page found].
swap_out: i 0, store page in vaddr 0x5000 to disk swap entry 6
swap_in: load disk swap entry 5 with swap_page in vadr 0x4000
write Virt Page e in fifo_check_swap
page fault at 0x00005000: K/W [no page found].
swap_out: i 0, store page in vaddr 0x1000 to disk swap entry 2
swap_in: load disk swap entry 6 with swap_page in vadr 0x5000
write Virt Page a in fifo_check_swap
page fault at 0x00001000: K/R [no page found].
swap_out: i 0, store page in vaddr 0x2000 to disk swap entry 3
swap_in: load disk swap entry 2 with swap_page in vadr 0x1000
count is 0, total is 5
check_swap() succeeded!
++ setup timer interrupts
this initproc, pid = 1, name = "init"
To U: "Hello world!!".
To U: "en.., Bye, Bye. :)"
kernel panic at kern/process/proc.c:342:
process exit!!.

Welcome to the kernel debug monitor!!
Type 'help' for a list of commands.
K>

```

## make grade

[~/moocos/ucore\_lab/labcodes/lab4]

moocos-> make grade

```

Check VMM: (1.9s)
  -check pmm: OK
  -check page table: OK
  -check vmm: OK
  -check swap page fault: OK
  -check ticks: OK
  -check initproc: OK
Total Score: 90/90

```