思考题 2

0、昨天我讲的 `sched_init(void)` 中，有这样的代码，`for(i=1;i<NR_TASKS;i++) {task[i] = NULL;}`。没有涉及到 `task[0]`，我提到 `task[0]` 已经给进程 0 了，请大家找到代码证据。

1、进程0的task_struct、内核栈、用户栈在哪？证明进程0的用户栈就是未激活进程0时的0特权栈，即user_stack，而进程0的内核栈并不是user_stack，给出代码证据。

2、在system

```
#define _set_gate(gate_addr,type,dpl,addr) \
__asm__ (   "movw %%dx,%%ax\n\t"    \
            "movw %0,%%dx\n\t"      \
            "movl %%eax,%1\n\t"     \
            "movl %%edx,%2"         \
            : \
            : "i" ((short) (0x8000+(dpl<<13)+(type<<8))), \
            "o" (*((char *) (gate_addr))), \
            "o" (*(4+(char *) (gate_addr))), \
            "d" ((char *) (addr)),"a" (0x00080000))

#define set_intr_gate(n,addr) \
    _set_gate(&idt[n],14,0,addr)

#define set_trap_gate(n,addr) \
    _set_gate(&idt[n],15,0,addr)

#define set_system_gate(n,addr) \
    _set_gate(&idt[n],15,3,addr)
```

3、这里中断门、陷阱门、系统调用都是通过 `_set_gate` 设置的，用的是同一个嵌入汇编代码，比较明显的差别是 `dpl` 一个是3，另外两个是0，这是为什么？说明理由。

4、进程 0 fork 进程 1 之前，为什么先要调用 `move_to_user_mode()`？用的是什么方法？解释其中的道理。

5、在 IA-32 中，有大约 20 多个指令是只能在 0 特权级下使用，其他的指令，比如 cli，并没有这个约定。奇怪的是，在 Linux0.11 中，在 3 特权级的进程代码并不能使用 cli 指令，会报特权级错误，这是为什么？请解释并给出代码证据。

6、用户进程自己设计一套 LDT 表，并与 GDT 挂接，是否可行，为什么？

7、分析初始化 IDT、GDT、LDT 的代码。

8、在 `sched_init(void)` 函数中有这样的代码：

```
for(i = 1;i < NR_TASKS; i++) {
    task[i] = NULL;
```

```
    ......
```

但并未涉及 `task[0]`，从后续代码能感觉到已经给了进程 0，请给出代码证据。