

## 实验 6

实现任务数据结构实现创建和销毁原语，实现上下文切换

学号：PB15051087

姓名：姜庆彩

- 实验内容：

1. 设计任务数据结构，命名为 myTCB
2. 实现任务创建原语（接口命名为 createTsk()）和销毁原语（接口命名为 destroyTsk()）功能
3. 实现任务上下文切换（接口命名为 CTX\_SW()）功能
4. 实现 FCFS 调度算法，调度接口命名为 schedule()
5. 实现简单的就绪队列和入列/出列原语
6. 实现任务启动原语，接口命名为 tskStart()
7. 实现任务终止原语，接口命名为 tskEnd()
8. 创建 idle 任务，idle 任务的主体是一个死循环，在循环体中，调用 schedule
9. 创建 init 任务，init 任务的主体（接口命名为 initTskBody()）由测试用例提供。
10. 实现 osStart 原语
11. 测试用例[老师提供 userApp 目录]：创建一组任务，这组任务按照 FCFS 的方式先后运行并输出相关信息。
12. 必须提供的接口说明参见 QQ 群中文件“6-README-2017-task-fcfs”
13. 老师提供 userApp 目录参见 QQ 群中压缩文件“6\_userApp.rar”

- 实验过程：

这次实验主要任务是任务调度与上下文的切换，由任务队列实现。

从实模式进入保护模式后 call mymain, 首先输出 hello world 接着调用 osStart ( ) 进入操作系统。

初始化进程池，在一开始创建一个 idle 进程（等待使用），初始化任务队列，初始化完成后开始 inittskbody ( ) 进程。

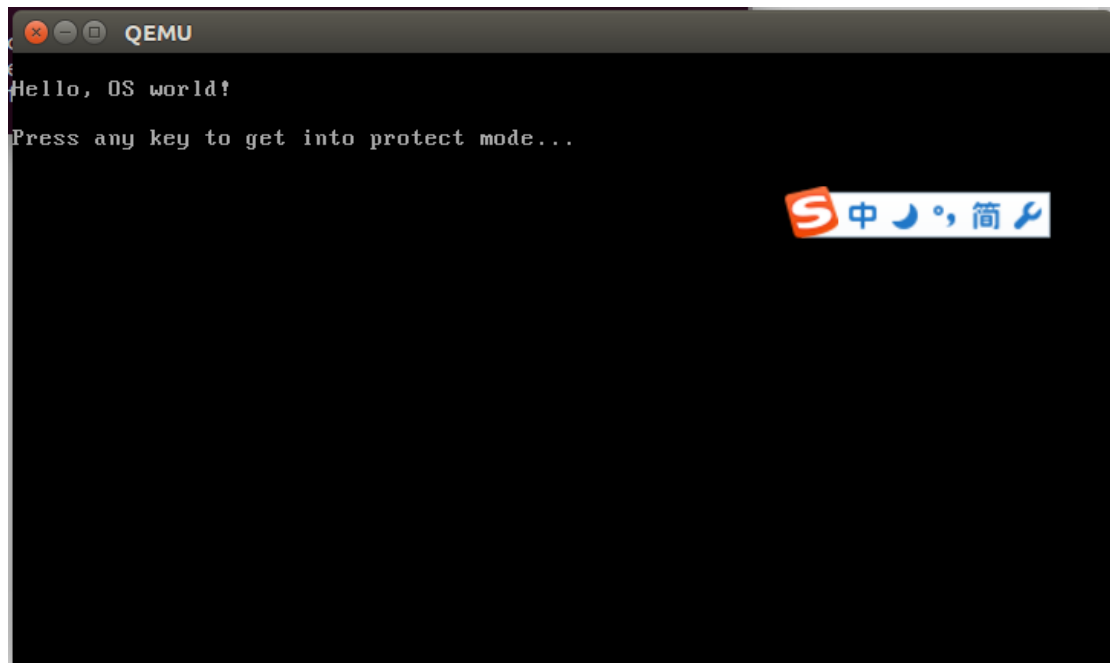
创建测试样例中三个任务进入等待队列中，FIFO。进程池中分配一块 PCB，函数地址保存至 eip 位置。测试样例中测试任务入队列后进入多任务模式。tskEnd ( ) 准备切换上下文。之后调用 CTX\_SW() 来保存上下文，调用 schedule ( )。队列中任务调度完成后调度 idle 进程。

流程图：

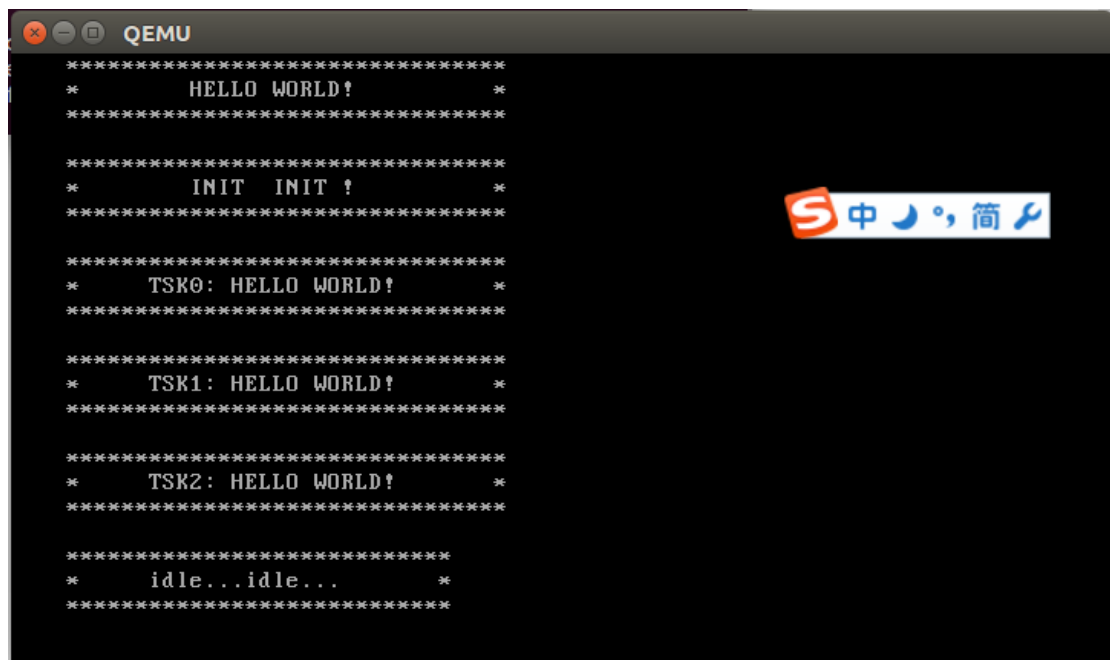


●结果截图：

进入保护模式前：



任务切换：



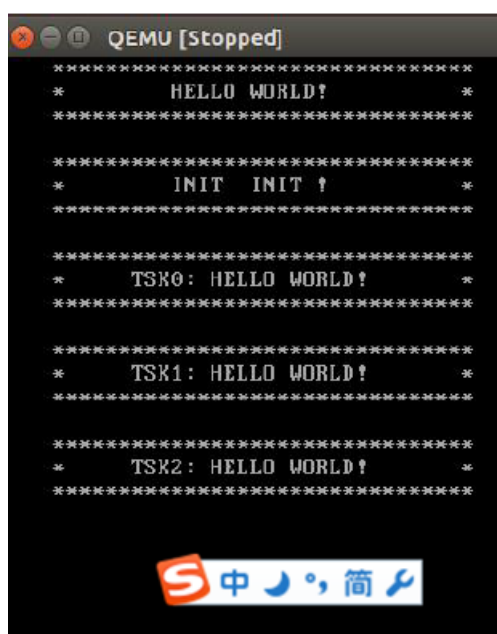
断点设置为 tskEnd ( ) 时  
第一个：



第二个:



第三个:



Idle:

```
QEMU
*****
*      HELLO WORLD!      *
*****

*****
*      INIT  INIT  !      *
*****

*****
*      TSK0: HELLO WORLD!  *
*****

*****
*      TSK1: HELLO WORLD!  *
*****

*****
*      TSK2: HELLO WORLD!  *
*****

*****
*      idle...idle...      *
*****
```