冯吕. 张旭

仁夕

rate at

问题及解决方 法

非抢占式内核

任务 1: 多 tasks 启动与 context switch

冯吕,张旭

University of Chinese Academy of Sciences

2017年10月11日



实现 问题及解决方

任务

1 任务

2 实现

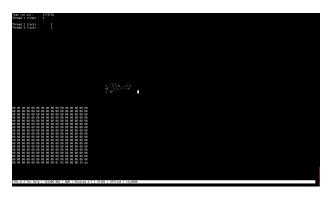
3 问题及解决方法

冯吕,张旭

任务

问题及解决方

整个 task 的实现。



函数以及函数调用关系

- bootblock读盘⇒ stat() _stat() 函数完成 PCB 的初始化和准备队列的初始化, 之后调用 scheduler_entry() 启动进程 (线程 1);
- scheduler_entrry() ⇒ scheduler() ⇒ 启动第一个进程
- 线程切换: do_yield() ⇒ $save_PCB()\&\&queue_pushscheduler_entry()$
- 进程切换: yield() ⇒ 系统调用 ⇒ do_yield()

任务实现

问题及解决方法

需要实现下面的函数:

- 修改 createimage, 需要将 process1,2,3 也写入 image.
- PCB 的设置
- 栈的设置
- _stat(): 初始化 PCB 和准备队列。
- save_pcb(): 保存上下文
- scheduler_entry(): 选取下一个进程来运行
- ...

任务

实现

问题及解决方 法

需要保存的信息:

• s0 ~ s7: 寄存器变量, 切换时需要保存。

• sp: 栈指针

• ra: 返回地址

进程状态(在任务一中暂时没什么用)。

任务 实现

问题及解决方

不能过大,否则 ⇒ TLB miss. 不能太小,否则与进程地址冲突。 冯吕,张旭

任务 实现

问题及解决方

看代码。

非抢占式内核

冯吕. 张旭

任务 实现

问题及解决方

入口地址变化

- 在实验一中, 读完盘后调到 0xa080026c 处执行。
- 在实验二中, _stat() 函数的入口地址变为 0xa08002bc.

PCB 信息存储

- 队列结构中只有一个 PCB 指针的指针, kernel.c 中有一 个 PCB 指针数组, 所以还需要自己声明一个 PCB 数组 存储信息。
- queue— > pcbs ⇒ ready_array[] ⇒ 真正存储 PCB 信息 的数组

栈的设置

- 栈的设置过大、导致跳到线程1压栈时出现 TLB miss.
- 减小栈的地址。
- 貌似一开始没有什么好的方法来确定,只能通过出错来 摸索。

保存 PCB 出错

保存 PCB 的时候 sp 和 ra 会发生变化。

```
a0800660 <do vield>:
a0800660:
           27bdffe8
                      addiu sp,sp,-24
           afbf0010
                      sw ra, 16(sp)
a0800664:
a0800668:
           0c200159
                      jal a0800564 <save pcb>
a080066c:
           00000000
                      nop
a0800670: 3c05a080
                      lui a1,0xa080
a0800674: 8ca32098
                      lw v1.8344(a1)
a0800678: 24020001
                      li
                          v0,1
```

非抢占式内核

冯吕,张旭

任务

实现 问题及解决方

kernel 入口地址

 刚开始忘记修改 kernel 入口地址, 导致进入 process1 后 无法切换到内核态。