Project 3 设计文档

马月骁 2019K8009915025

基于 design review

1. Shell 支持的命令

   详见 README

2. Spawn、kill、wait、exit 内核实现

（1）spawn

在具体操作上与 Project 2 中的 init_pcb 中进行的操作基本保持一致。

新增的功能：

a）在初始化栈时，添加参数支持初始化 a0 等寄存器

b）保持 spawn 的子进程 mask 与父进程保持一致

（2）kill & exit

两者进行的操作基本一致，kill 需要寻找 pid 对应的进程，exit 的对象则为当前进程。

Kill & exit 需要释放操作对象的 mode 执行对应操作：

a） AUTO_CLEANUP_ON_EXIT：释放等待队列中进程；
释放在内核中所占用资源

b） ENTER_ZOMBIE_ON_EXIT：释放等待队列中进程；
释放锁等内核资源；
保存在内核中所占用空间；

（3）wait

将当前进程加入指定进程的等待队列。直至指定进程完成后继续运行。

3. Kill 进程时锁的处理

不同于讲义中的描述，此次实验中我们实现的锁为内核态的资源，故 kill 进程时，也应将其所占有的锁释放。

4. 信号量与屏障

与锁一样，将这两种同步原语也设计为内核资源。具体的实现方法与 project 2 中的锁实现方式基本一致，不加赘述。

由于为内核资源，故在处理信号量与屏障系统处于内核态，屏蔽时钟中断。

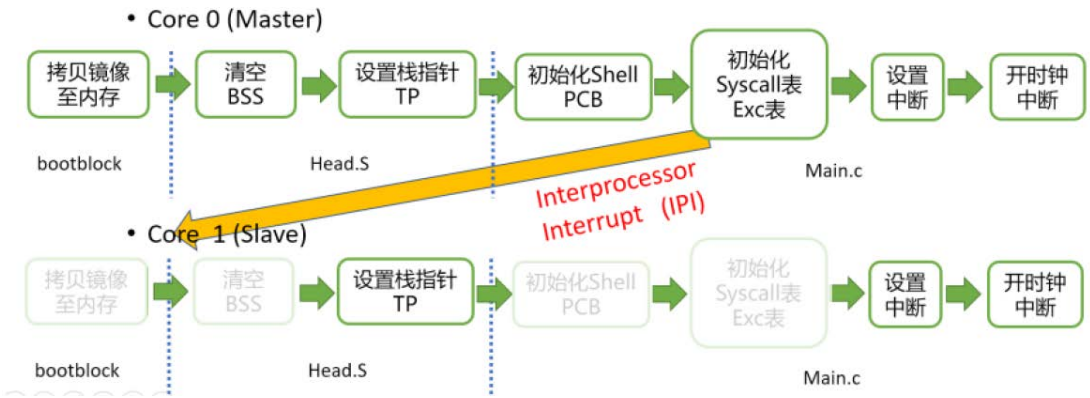
5. 邮箱

Mailbox struct:

```
typedef enum {
    MBOX_CLOSE,
    MBOX_OPEN,
} mailbox_status_t;
typedef struct mailbox_k
{
    char name[25];                  //name of mailbox
    char msg[MAX_MBOX_LENGTH];      //content of message
    int index;                      //current length of msg
    mailbox_status_t status;
    list_head empty_queue;
    list_head full_queue;
} mailbox_k_t;
```

6. 双核启动

具体流程如下:



~~(图是嫖的)~~

Debug 记录

TASK 1:

未出现严重问题。

修复了 Project 2 中的部分不合理之处：唤醒 sleep 队列中 PCB 时，没有再次调度的必要。

```
void do_unblock_timer(pcb_t *pcb_node){
    pcb_node->status = TASK_READY;
    add_queue(&ready_queue, pcb_node);
}
```

TASK 2:

出现问题：mailbox 初始化时忘记初始化两个阻塞队列，导致指针跑飞

解决方法：在第一次 open 时初始化

```
init_list_head(&(MBOX_LIST[mbox_count].full_queue));
init_list_head(&(MBOX_LIST[mbox_count].empty_queue));
```

出现问题：mailbox 测试文件中 sys_move_cursor(0,0)为非法地址，导致指针跑飞

解决方法：改个地址

```
sys_move_cursor(1, 4);
printf("[Server]: recved msg from %d (blocked: %d, correctBytes: %d, errorBytes: %d)",
       header.sender, blockedCount, correctRecvBytes, errorRecvBytes);
```

（~~想必写测试文件的助教没有实际跑过，裂开~~）

TASK 3:

出现问题：例外返回时，由于先进入了 unlock_kernel 函数，导致 restore 的 ra/sepc 错误，陷入死循环

解决方法：先处理内核锁，再恢复现场即可

```
ENTRY(ret_from_exception)
  /* TODO: */
  //csrw CSR_SSCRATCH, tp
  call unlock_kernel
  RESTORE_CONTEXT
  sret
ENDPROC(ret_from_exception)
```

出现问题：双核启动时，主核误接收核间中断

解决方法：主核只打开时钟中断

```
ENTRY(setup_exception)
  /* TODO:
   * save exception_handler_entry into STVEC
   * enable global exceptions */
  la t0, exception_handler_entry
  csrw stvec, t0
  /*
  li s0, SR_SIE
  csrw sstatus, s0
  */
  li t0, SIE_STIE//SIE_SSIE | SIE_STIE | SIE_SEIE
  csrw sie, t0
  jr ra
ENDPROC(setup_exception)
```

出现问题：qemu 能跑，上板 core-dump
解决方法：根据 sepc，注释了 wfi
不是很能理解为什么，十分玄学。。。

TASK 4:

没什么问题，只是这和双核有什么必然的关系？
（~~用了 goto，感觉不错~~）

TASK 5:

没什么问题
~~C core 不应该和 A core 换个位置？~~

感谢贾文庆助教帮助我定位 Task 3 的 bug