ch3实验报告

实验

实现获取当前进程状态可以给 `task/mod.rs` 的TaskManager结构实现一个`get_current_status`的方法读取自身私有字段status,然后外面写一个函数把给全局变量`TASKMANAGER`做一个反射就行

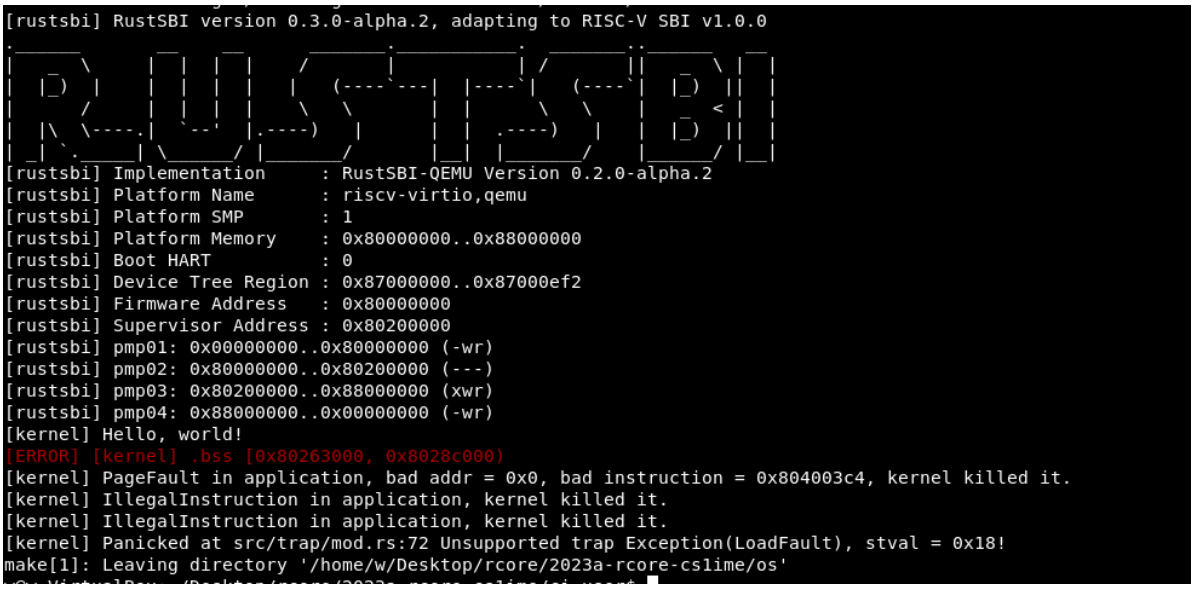
实现记录syscall调用次数可以给`TaskControlBlock`结构体加上一个`[u32; MAX_SYSCALL_NUM]`的成员负责存储syscall调用次数记录,在 `syscall/mod.rs` 里面加上记录syscall调用次数的函数,把syscall_id作为参数传进去

实现获取进程运行时间可以给`TaskControlBlock`结构体加上一个进程启动时间的参数,获取进程运行时间就调用`get_time_ms()`减去进程启动时间得到进程运行时间

简答题

1

运行截图



1. bad_address: 0地址访问
2. bad_instruction: 用户态使用sret指令(s模式返回u模式)导致异常
3. bad_register: 用户态访问sstatus寄存器导致异常

2

深入理解 trap.S 中两个函数 `__alltraps` 和 `__restore` 的作用，并回答如下问题:

1. a0是 `__switch` 函数传入的TaskContext.
   `__restore` 第一个使用情景是作为内核层返回用户层的桥梁,第二种使用情景是启动用户态程序
2. sepc,sscratch,sstatus
   sepc是sret返回用户态的地址,sscratch是用户态堆栈地址,sstatus控制用户态的中断启用关闭,中断向量表地址,特权级SPP
3. x2是sp,下面有恢复sp的代码,x4是线程局部存储指针的寄存器,暂时还未用到
4. 该指令操作过程:

   操作之后的sp是用户态的栈指针,sscratch是内核态的栈指针
5. sret,因为这是中断返回指令,sstatus的SPP位被设为0了
6. 操作之后的sp是内核态的栈指针,sscratch是用户态的栈指针
7. ecall

荣誉准则

交流对象: 群内大佬帮我解决了环境问题