PA4实验报告

姓名：周琰轲 学号：201850187

已完成PA4所有必做内容

分时多任务的具体过程

这里主要分析hello和pal的多任务过程：

1. 创建PCB：首先会创建hello和pal的PCB结构，值得注意的是，nanos-lite代码此时运行在 `boot_pcb` 上，nanos-lite的上下文是无用的，它在完成一系列初始化后就直接跳转到进程运行，再也不会回到nanos-lite；因此 `boot_pcb`中保存的上下文实际上是无用的。
2. nanos-lite调用 `yield()`后进入trap.S，并在 `schedule()`后还原新的上下文，这个上下文已经设置好了hello的地址空间页表，程序入口，栈顶位置等；切换上下文后mret进入hello运行。
3. 在 `execute()`中，系统运行10s后INTR引脚被拉高，且在进行上下文切换的时候mstatus的MIE位设为高电平，此时触发时钟中断，调用 `schedule()`
4. 再次进入trap.S，sp将会被切换到内核栈上（由于hello是内核进程，其内核栈和用户栈共用)，在内核栈上保存上下文，切换到handler；`do_event()`识别到异常号时钟中断，切换到pal的上下文，同时将sp切换到pal的用户栈
5. 返回到trap.S，恢复上下文并切换到pal的地址空间运行（这里的地址都是虚拟地址，需要经过mmu映射)；10s后INTR引脚再次被拉高，引发中断
6. 再次进入trap.S，先将sp切换到内核栈，内核栈的地址保存在mscratch中，在内核栈保存上下文以及用户栈位置，之后再次切换到hello的上下文，一直持续下去

理解计算机系统

字符串“abc"在链接时会被链接到ELF的只读数据段，语句 `p[0]='A'`是一次访存操作；通过objdump查看汇编代码，可以发现一条：

```
c6 00 41                movb    $0x41, (%rax)
```

在进行这一条语句的时候mmu会访问 `(%rax)`这一地址，访问的时候同时检查这一地址是否有写权限，由于这一地址是只读数据段，因此不可写。

通过gdb调试可以发现，执行到这条语句后会抛出一个SIGSEGV信号表示段错误，用户进程收到这个信号后abort掉。

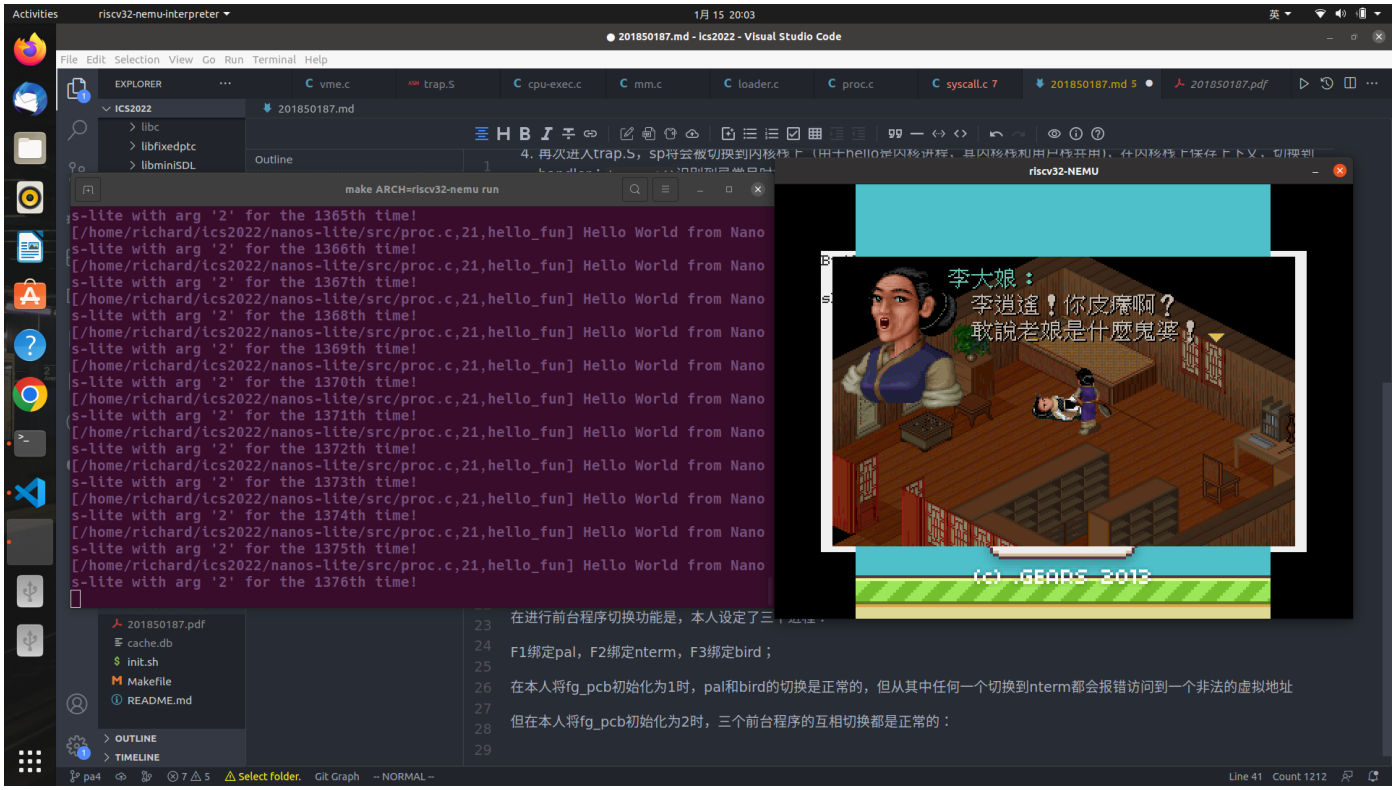
一个未解决的问题

在进行前台程序切换功能是，本人设定了三个进程：

F1绑定pal，F2绑定nterm，F3绑定bird；

在本人将fg_pcb初始化为1时，pal和bird的切换是正常的，但从其中任何一个切换到nterm都会报错访问到一个非法的虚拟地址

但在本人将fg_pcb初始化为2时，三个前台程序的互相切换都是正常的：



这一问题目前还没能解决，个人感觉是用户栈的映射上可能存在或多或少的问题。