PA 4 实验报告

- **Author:** 张昱东
- **SID:** 141120159
- **Email:** 141120159@smail.nju.edu.cn

- **Time:** 8月14日

一、实验进度

完成了部分实验内容：

- PA 4-1 异常和中断的响应 (部分完成)
  - [x] 通过自陷实现系统调用
    - [x] 添加 IDTR 结构体
    - [x] 实现 lidt、cli、sti、int、pusha、popa、iret 等指令
    - [x] 实现 raise_intr() 函数 (还存在bug)
    - [ ] hello-inline 测试样例未通过
  - [x] 响应时钟中断
    - [x] 添加 intr 并初始化
    - [x] 在 exec() 函数中添加 do_intr() 调用
    - [ ] 移除 panic (由于前面的部分存在 bug，导致这里并没有触发 panic)
- PA 4-2 外设与IO (部分完成)
  - [x] 串口模拟
    - [x] 实现 in 和 out 指令（未能测试是否实现正确）
    - [ ] 实现 serial_printc()函数
  - [ ] 硬盘加载程序
  - [ ] 键盘的模拟
  - [ ] 实现 VGA 的 MMIO

- (可选任务) PA 4-3 (未做)

  - [ ] 打字小游戏
  - [ ] 仙剑奇侠传

二、实验结果

由于在 PA 4-1 出现了为解决的 bug，hello-inline 样例未通过， 导致后面的测试也都无法顺利进行。

```
Execute ./kernel/kernel.img ./testcase/bin/sum
nemu trap output: [src/main,75,init_co] {kernel} Hello, NEMU world!
nemu trap output: [src/elf/elf.c,30,loader] {kernel} ELF loading from ram disk.
nemu: HIT GOOD TRAP at eip = 0x080480dc
NEMU2 terminated

Execute ./kernel/kernel.img ./testcase/bin/wanshu
nemu trap output: [src/main,75,init_co] {kernel} Hello, NEMU world!
nemu trap output: [src/elf/elf.c,30,loader] {kernel} ELF loading from ram disk.
nemu: HIT GOOD TRAP at eip = 0x08048125
NEMU2 terminated

Execute ./kernel/kernel.img ./testcase/bin/struct
nemu trap output: [src/main,75,init_co] {kernel} Hello, NEMU world!
nemu trap output: [src/elf/elf.c,30,loader] {kernel} ELF loading from ram disk.
nemu: HIT GOOD TRAP at eip = 0x080481a0
NEMU2 terminated

Execute ./kernel/kernel.img ./testcase/bin/string
nemu trap output: [src/main,75,init_co] {kernel} Hello, NEMU world!
nemu trap output: [src/elf/elf.c,30,loader] {kernel} ELF loading from ram disk.
nemu: HIT GOOD TRAP at eip = 0x0804820a
NEMU2 terminated

Execute ./kernel/kernel.img ./testcase/bin/hello-str
nemu trap output: [src/main,75,init_co] {kernel} Hello, NEMU world!
nemu trap output: [src/elf/elf.c,30,loader] {kernel} ELF loading from ram disk.
nemu: HIT GOOD TRAP at eip = 0x080481a5
NEMU2 terminated

Execute ./kernel/kernel.img ./testcase/bin/test-float
nemu trap output: [src/main,75,init_co] {kernel} Hello, NEMU world!
nemu trap output: [src/elf/elf.c,30,loader] {kernel} ELF loading from ram disk.
nemu: HIT BAD TRAP at eip = 0x0804815a
NEMU2 terminated

Execute ./kernel/kernel.img ./testcase/bin/hello-inline
nemu trap output: [src/main,75,init_co] {kernel} Hello, NEMU world!
nemu trap output: [src/elf/elf.c,30,loader] {kernel} ELF loading from ram disk.
nemu: src/cpu/intr.c:29: raise_intr: Assertion `idt.present == 1' failed.
Makefile:18: recipe for target 'testkernel' failed
make: *** [testkernel] Aborted
```

三、必答题

（由于实验内容没有全部完成，因此只回答已完成部分对应的简答题）

- 详细描述从测试用例中的 int $0x80 开始一直到 HIT_GOOD_TRAP 为止的详细的系统行为(完整描述控 制的转移过程，即相关函数的调用和关键参数传递过程)，可以通过文字或画图的方式来完成

    - 从 int $0x80 开始，系统调用已实现的 int_i_b 指令, 这个指令再调用 raise_sw_intr() 函数，先对 eip+2 指向下一条指令，然后调用 raise_intr() 函数。在 raise_intr() 函数中，系统先保护断点和程序状态(eflags, cs, eip)，然后判断是否关中断 (根据TYPE)，最后读取对应的 IDT，更新 cs 和 eip，使其指向异常和中断处理程序，从而下一步开始执行处理程序
    - 当处理程序执行完成后，再执行 iret 指令，回复程序断点和状态，再继续执行下面的指令直到 HIT_GOOD_TRAP.

- 在描述过程中，回答 kernel/src/irq/do_irq.S 中的 push %esp 起什么作用，画出在 call irq_handle 之前， 系统栈的内容和 esp 的位置，指出 TrapFrame 对应系统栈的哪一段内容。

    - push %esp 的作用是保存用户栈的信息，因为之后就要转入内核态而改用系统栈了，当系统调用完成后需要再恢复用户栈的状态

四、实验过程以及遇到的问题

- 在实现 PA 4-1 的过程中，我在实现 int 指令以及 raise_intr 时，遇到了 IDT 的 present 位不为 1 的问题。经过仔细检查后发现，是在读取 IDT 时根据 IDTR 给出的 base 地址加上中断号找到的内存上的内容为全 0，也就是说读到的内容是空的。

    - 因此我认为我的 lidt 的实现也可能有问题，然而 lidt 的实现几乎完全参考 lgdt 的实现，但 lgdt 是通过了 PA 3 的全部测试的。当我尝试修改 ligt 和 lgdt 的实现时，又导致了除 hello-inline 以外的样例出错的问题。导致我并不确定这个 bug 是写 PA 3 的时候未被测试出来的 (lgdt) 还是在 PA 4-1 的时候新出现的 (raise_intr)

- 因此我打算暂时放弃 debug，而是继续往后写，但是在完成响应时钟中断的代码部

分后，却发现并未出现本该出现的 panic (因为前面还有 bug 的缘故)，所以也没法测试后续代码的正确性

- 说明：在 config.h 中，我没有把后几个 HAS_DEVICE 解除注释，是因为前面的 bug 导致这里一旦解除注释，就会编译出错

- 教训：

    - 因为 PA 3 完成得较为顺利，因此对 PA 4 大意了，以为也能很快写完，结果遇到了一个麻烦的 bug，花了两天时间都没有 de 出来，导致后面的阶段无法继续下去。教训是任务还是要尽早开始，因为无法预料到后面会出现什么情况，应该预留充分的时间
    - 对代码的测试要充分，因为 PA 3 实现时没有在代码中检查一些中间状态是否正确，导致可能出现了测试样例没有检查出来的 bug，而这个 bug 在 PA 4 中出现了（当然也可能这个 bug 是 PA 4 才新引入的，和 PA 3 无关）
    - 在理论学习中我对于异常和中断的处理的理解也没有很深刻，导致出现 bug 时不能进行很有效的分析，也没法快速定位 bug