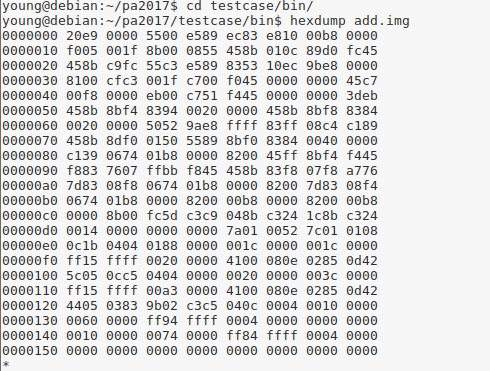
# PA2实验报告

161220096 欧阳鸿荣

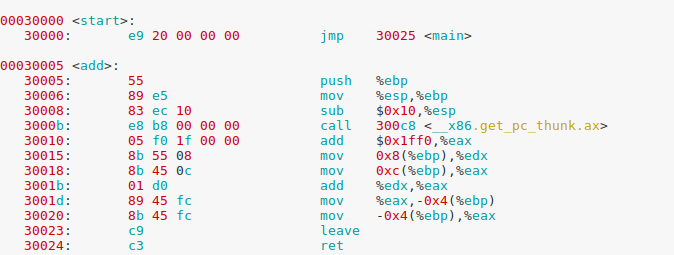
## PA 2.1

**1. 使用 hexdump 命令查看测试用例的.img 文件，所显示的.img 文件的内容对应模拟内存的哪一个部分？指令在机器中表示的形式是什么？**

在pa2017/ testcase / bin 目录下，以add为例，用hexdump命令查看测试用例的add.img



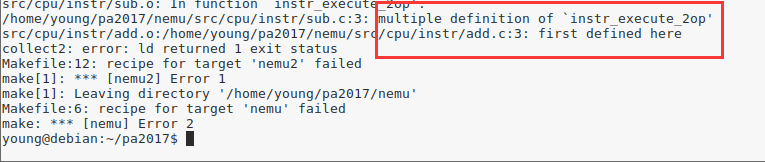
可以看到，这是用十六进制表示的信息，用objdump输出 add反汇编的结果add.s，打开



可以看出，实际上.img文件中存储的便是模拟内存中从<start>处开始的机器指令，并且是按照16位二进制数为单位，大端模式存储的。

**2. 如果去掉 instr\_execute\_2op()函数前面的 static 关键字会发生什么情况？为什么？**

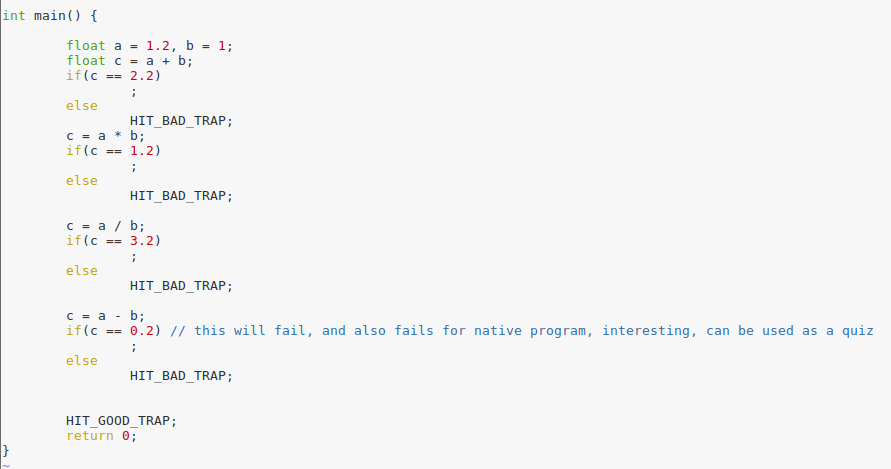
把pa2017/neum/src/cpu/instr/中的add.c和sub.c中instr\_execute\_2op()函数前面的 static 关键字去掉，在make的时候，可以看到如下报错信息：



很显然，这是因为去掉了static关键字导致的。static 关键字是为了将 instr\_execute\_2op()函数的作用域限制在该.c 文件中，以便我们能够在别的指令对应的.c 文件中复用该函数名,这有点类似与C++中类的private。去掉static关键字后，对于instr\_execute\_2op之类的同名函数，在编译链接的过程中便会产生重定义，编译器不清楚应该引用谁，那就会对程序的执行产生冲突。

**3. 为什么 test-float 会 fail？以后在写程序的时候要注意什么？**

在pa2017/testcase/src中打开test-float.c，观察源代码：



可以看到，test-float中进行了浮点数的运算。

但我们其实知道，并不是所有的浮点数都可以在计算机中精确表示，

而由于0.1的32位浮点表示为0011 1101 1100 1100 1100 1100 1100 1100...

因此0.2的32为浮点表示为0111 1011 1001 1001 1001 1001 1001 1001…

都是无法精确表示的浮点数，然而在源代码中用了“==”号，因此无法得到算术上c严格等于0.2的结果，因此hit bad trap

因此以后写程序时，对于涉及浮点数的运算，应该注意浮点数并不能完全精确表示，如果要判断两个数是否相等，应该用类似 abs(a-b)<1e-5，差绝对值小于一个小数即可认为相等。

## PA 2.2

**1. 为什么在装载时要把内存中剩余的 p\_memsz - p\_filesz 字节的内容清零?**

防止在kenerl在取地址时误操作或误读，把.bss节清空

## PA 2.3

**1.我们在 add.c 中定义了宏 NR\_DATA , 同时也在 add() 函数中定义了局部变量 c 和形参 a , b , 但你会发现在符号表中找不到和它们对应的表项, 为什么会这样? 思考一下, 什么才算是一个符号(symbol)?**

每个可重定位目标模块m都有一个符号表，它包含了在m中定义的符号。

有三种链接器符号：

1.Global symbols（模块内部定义的全局符号）由模块m定义并能被其他模块引用的符号。例如，非static 函数和非static的全局变量（指不带static的全局变量）

2.External symbols（外部定义的全局符号）由其他模块定义并被模块m引用的全局符号

3.Local symbols（本模块的局部符号）仅由模块m定义和引用的本地符号。

只有满足上列条件之一的变量或者函数才能算是一个符号(symbol)，而add() 函数中定义的局部变量 c 和形参 a , b ,都只是局部变量，并不在符号表中，不能算是一个符号。而宏 NR\_DATA，在预处理时便把宏名替换成相应的字符串代码，因此其实本质上他也不算一个符号。