

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？

对应于模拟内存从 `0x30000` 开始的连续区域。

形式是二进制位串。

2.

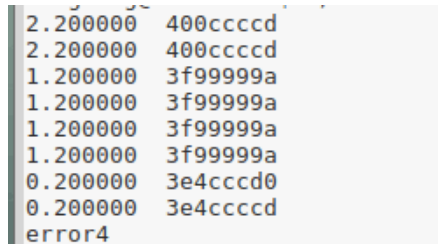
如果去掉 `instr_execute_2op()` 函数前面的 `static` 关键字会发生什么情况？为什么？

会发生 `instr_execute_2op` 的重复定义。

`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。而去掉这个 `static` 关键字使得所有定义的函数不受限制，从而会发生重复定义。

3.

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



```
2.200000  400ccccd
2.200000  400ccccd
1.200000  3f99999a
1.200000  3f99999a
1.200000  3f99999a
1.200000  3f99999a
0.200000  3e4ccccd0
0.200000  3e4ccccd
error4
```

可以看出是在执行减法的时候出现的错误，在执行减法之后虽然得出的结果是 `0.2`，但是在机器里的表示是 `0x3e4ccccd0`，而直接的 `0.2` 在机器里的表示是 `0x3e4ccccd`，这就产生了错误。根本原因是有很多数不能用浮点数精确表示，在计算的时候由于这种不精确表示，会产生误差。

所以以后编程的时候要特别注意浮点数的计算，尽量少使用判断是否相等的操作，往往结果不符合预期。

4.

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？

因为这一段内容对应于 ELF 文件中的 `.bss` 段，`.bss` 段用来存放程序中未初始化的全局变量的一块内存区域，一般在初始化时 `.bss` 段部分将会清零。`.bss` 段属于静态内存分配，因此程序一开始就将其清零了。

5.

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)？

宏定义在预编译的阶段就被解析了，纯粹是一种文本的替换，因为不会存到符号表中。局部变量和形参都是在程序运行的过程中在栈区生成的，因此不会事先存到符号表中。存在符号表中的符号一般是已经初始化的全局变量和函数名。