2020 PA 1

姓名:赵超懿 学号:191870271 匡亚明学院

实验进度:

我完成了所有实验内容

必答题

1.我选择的ISA是 x86 。

2.画出1+2+。。。+100的程序状态机 (PC , r1 , r2)

$$(0,x,x)->(1,0,x)->(2,0,0)->(3,0,1)->(4,1,1)->(2,1,1)$$
$$->(3,1,2)->(4,3,2)->(2,3,2)->(3,3,3)->(4,6,3)->(2,6,3)$$
$$->.....->(3,4851,99)->(4,4950,99)->(2,4950,99)$$
$$->(3,4950,100)->(4,5050,100)->(5,5050,100)->(5,5050,100)$$

3.理解基础设施

使用GDB进行调试时间:

$$t=500\times 90\%\times 30\times 20=270000(s)=75(h)$$

使用简易调试器

$$t=500\times 90\%\times 10\times 20=90000(s)=25(h)$$

节约50h

4.RTFM

x86:
EFLAGS寄存器中的CF位是什么意思?

Page 33 of 421 , 2.3.4 Flags Register 以及 Appendix C Page 419 of 421

ModR/M字节是什么?

Page 241 of 421 ,17.2.1 ModR/M and SIB Bytes 至 Page 245 of 412

mov指令的具体格式是怎么样的?

shell命令

使用 find -name "*.[h|c]" | *xargs wc -l*

结果是 6024

去掉空行 find -name "*.[h|c]" | xargs grep -v "^$" | wc -l

结果是 5017

未编写前 分别为5322 4345

RTFM:

gcc -Wall 选项显示所有警告

gcc -Werror 将所有警告视为错误,并报错

作用:将所有警告视为错误报错,可以最大可能减少隐患和将来可能出现的错误。

实验感受：

这一部分是后来返回来写的。

总体来说，刚刚开始的PA就像一个新的世界，我第一次了解到可以这样写代码，内有不断有“还有这种操作“的感慨，大一写的代码感觉就是小打小闹。

第一阶段还是比较友好的，首先，总的东西不多，主要是表达式求值，难度也不是很大（和后面相比）。

第一个成就感来自于x86的寄存器的代码，通过自己差union和struct的匿名使用方法，了解到struct和union居然可以这样用（我之前甚至都不知道union，555），通过一晚上的努力，我终于完成了这个内容。

主要困难还是在于阅读那一段监视器的框架代码，不过难度也不大（x），主要是要找到对应的API，写的时候仿照框架代码就对了。

最后是调试表达式求值的部分，这一部分首先讲义有充分的提示，然后就是基本功的问题，（可惜我的基本功并不是很好）还是花了不少时间，不过由于问题集中，总能调出来。在这里我还没学会去找好用的工具，基本就是目力debug和make GBD，然后还没改O2的优化，（充分凸显刚开始的年幼无知.jpg）

从现在的观点来看，我觉得monitor后面的作用可能是没有其作为一个让我这种菜鸡从什么都不会到学会看框架代码，学会写一个稍微长一点代码的练习的作用大的，正是PA1的这一段的适应，让以后的路没有那么艰难。

PA1最大的帮助是让我接触一个新的世界，从什么都不知道到能做一点点，并且学会了很多东西的用法。从此开始了c语言的补课之路（x）。