

**实验报告**



**题目： 拆解二进制炸弹**

**班 级： 2019211315**

**学 号： 2019211608**

**姓 名： 姜彦**

**学 院： 计算机学院**

**2020年 11 月 7 日**

一、实验目的  
1.理解C语言程序的机器级表示。  
2.初步掌握GDB调试器的用法。  
3.阅读C编译器生成的x86-64机器代码，理解不同控制结构生成的基本指令模式，过程的实现。

1. 实验环境
2. Xshell（10.120.11.12）: Xshell是一个强大的安全终端模拟软件，可以在Windows界面下用来访问远端不同系统下的服务器，从而达到远程控制终端的目的
3. Linux：一套免费使用和自由传播的类UNIX操作系统，是一个基于POSIX和Unix的多用户、多任务、支持多线程和多CPU的操作系统
4. Objdump命令反汇编：objdump 有点像那个快速查看之类的工具，就是以一种可阅读的格式让你更多地了解二进制文件可能带有的附加信息。
5. GDB调试工具：UNIX及UNIX-like下的调试工具
6. 积分榜（http://10.120.11.34:19230/scoreboard）

**报告邮寄（最迟时间：2020年11月25日晚23：59）：**

**大一班（11-15班）：**[**bywork2019@bupt.edu.cn**](mailto:bywork2019@bupt.edu.cn)

三、实验内容

登录bupt1服务器，在home目录下可以找到Evil博士专门为你量身定制的一个bomb，当运行时，它会要求你输入一个字符串，如果正确，则进入下一关，继续要求你输入下一个字符串；否则，炸弹就会爆炸，输出一行提示信息并向计分服务器提交扣分信息。因此，本实验要求你必须通过反汇编和逆向工程对bomb执行文件进行分析，找到正确的字符串来解除这个的炸弹。

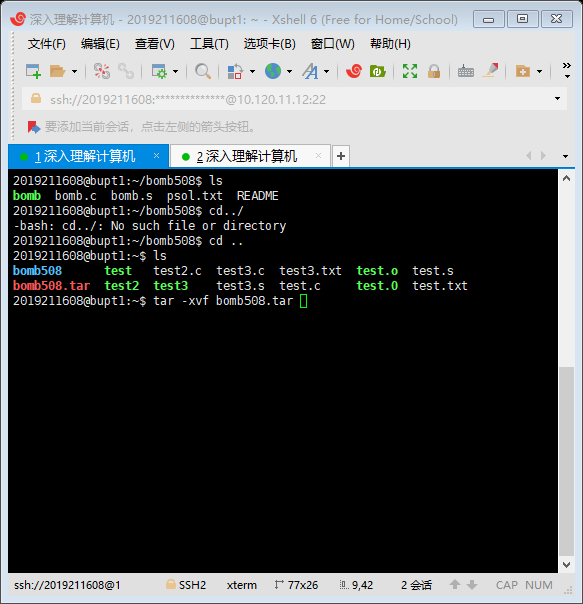
本实验通过要求使用课程所学知识拆除一个“binary bombs”来增强对程序的机器级表示、汇编语言、调试器和逆向工程等方面原理与技能的掌握。 “binary bombs”是一个Linux可执行程序，包含了5个阶段（或关卡）。炸弹运行的每个阶段要求你输入一个特定字符串，你的输入符合程序预期的输入，该阶段的炸弹就被拆除引信；否则炸弹“爆炸”，打印输出 “BOOM!!!”。炸弹的每个阶段考察了机器级程序语言的一个不同方面，难度逐级递增。

为完成二进制炸弹拆除任务，需要使用gdb调试器和objdump来反汇编bomb文件，可以单步跟踪调试每一阶段的机器代码，也可以阅读反汇编代码，从中理解每一汇编语言代码的行为或作用，进而设法推断拆除炸弹所需的目标字符串。实验2的具体内容见实验2说明。

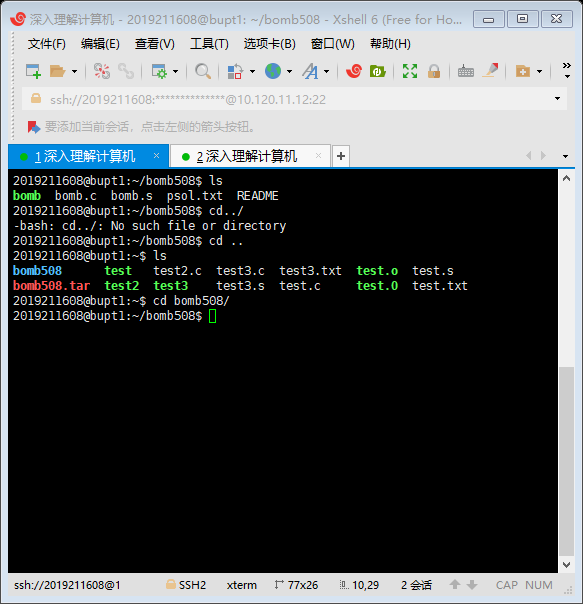
四、实验步骤及实验分析

准备工作：

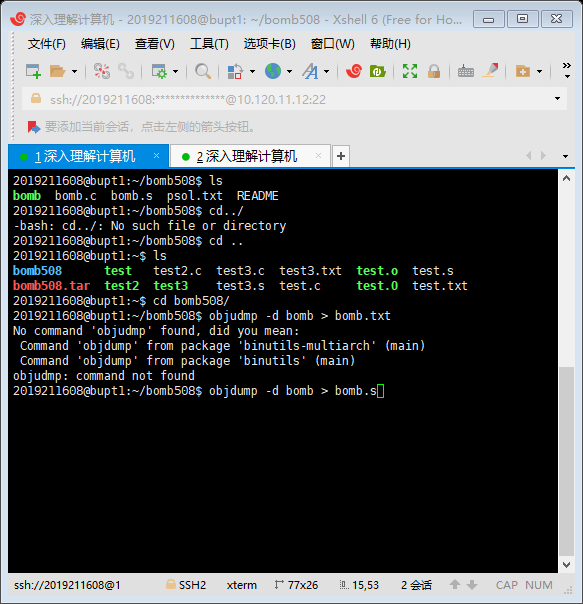
首先要将得到的炸弹解压



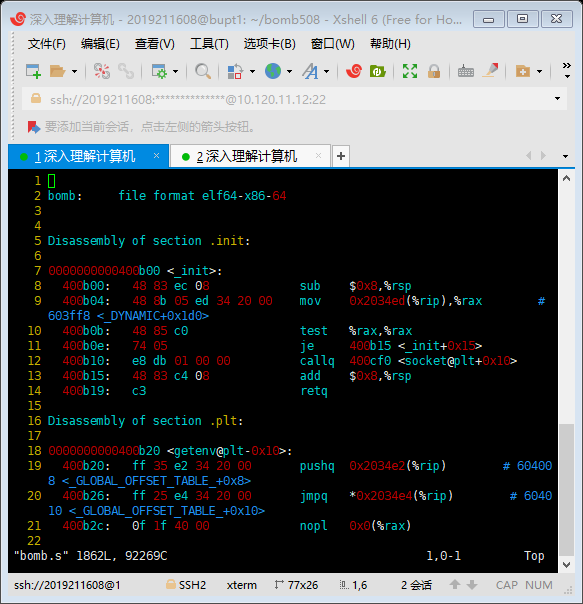
得到bomb508文件夹后进入



将可执行文件bomb反汇编并将文本保存到bomb.s

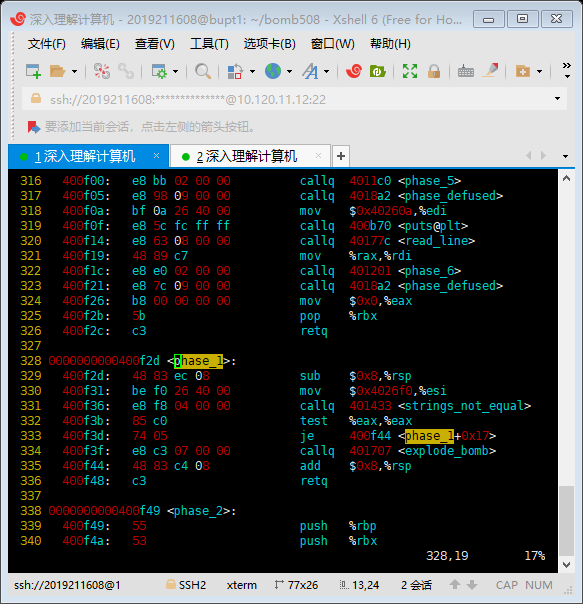


Vi打开反汇编文件bomb.s并打开另一个远程连接窗口用作调试

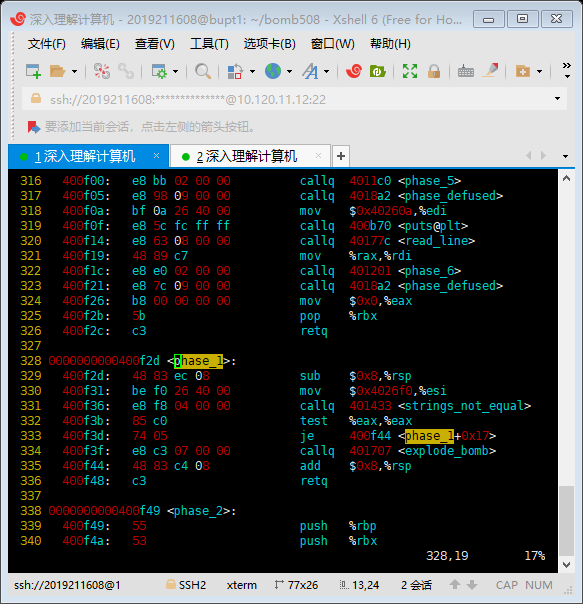


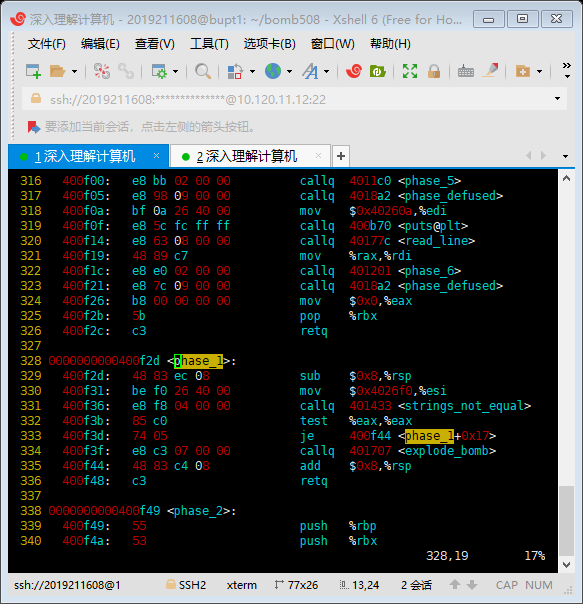
阶段1：

首先在bomb.s中找到对应phase\_1的函数位置

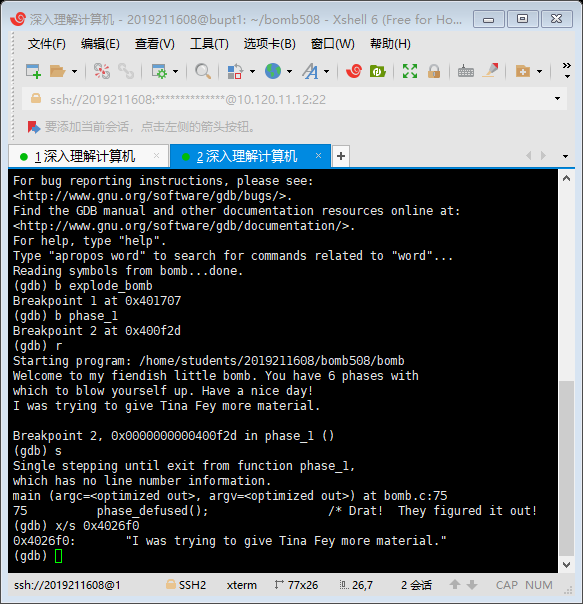


分析知道，该函数将输入与特定字符串比较，当返回值为0时跳过炸弹爆炸



因此关键点在于找到位于处字符串的值是多少

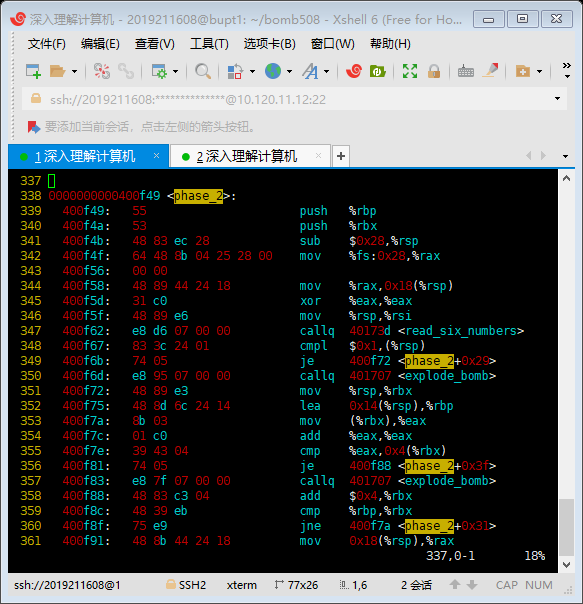
所以通过gdb调试，设置phase\_1断点，查看内存知道所要的字符串为



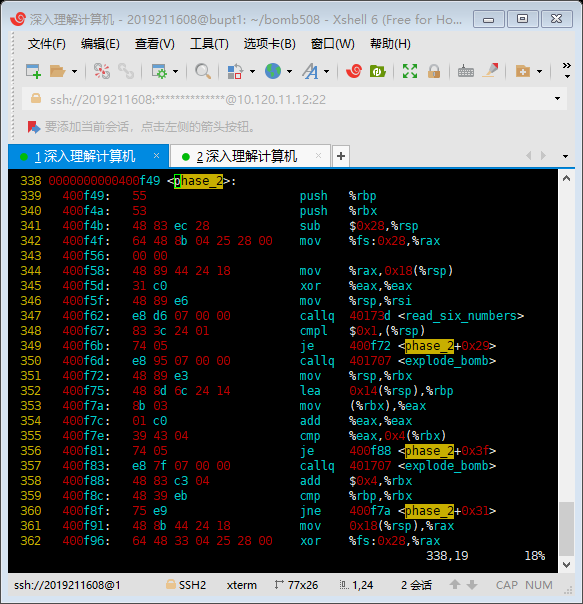
将该字符串输入后，该炸弹成功拆除

阶段2：

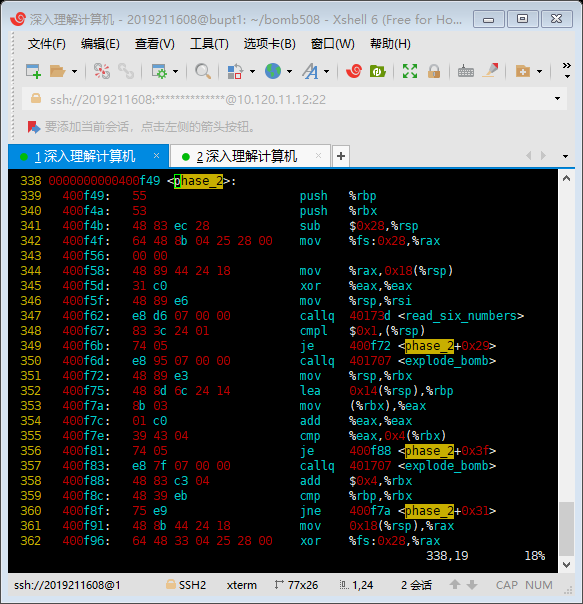
在bomb.s中找到对应phase\_2的函数位置



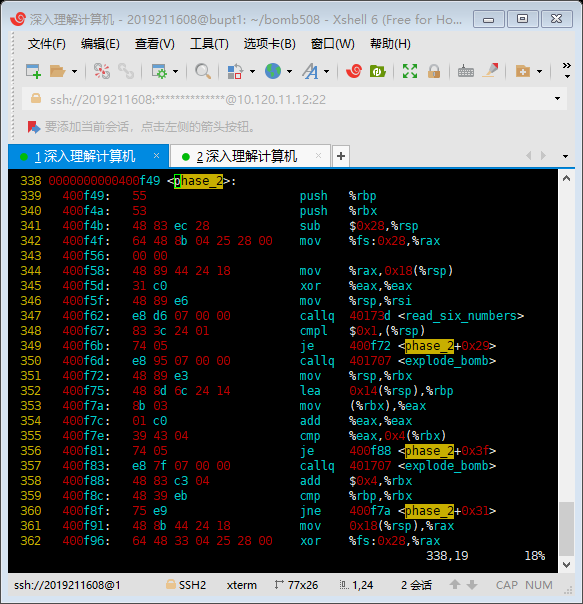
分析可知首先需要输入6个数字且第一个数字必须为1才能跳过第一个炸弹

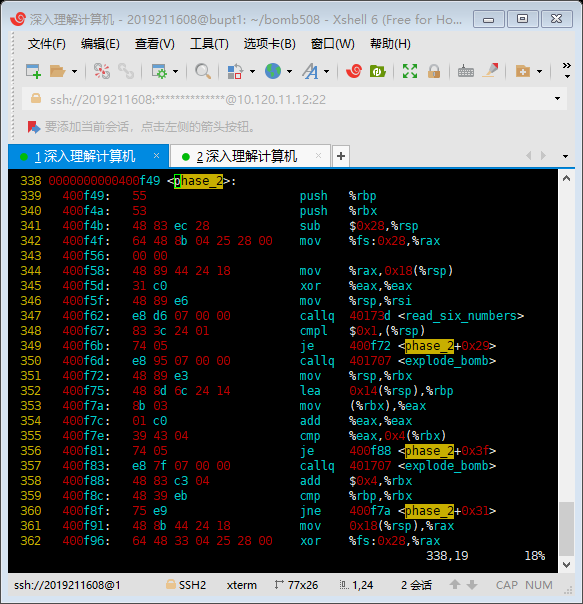


然后会将前一个数字与后一个数字比较，只有当后者为前者两倍时才跳过爆炸



之后指向下一个数字，当所有数字比较完后结束，否则跳回循环执行

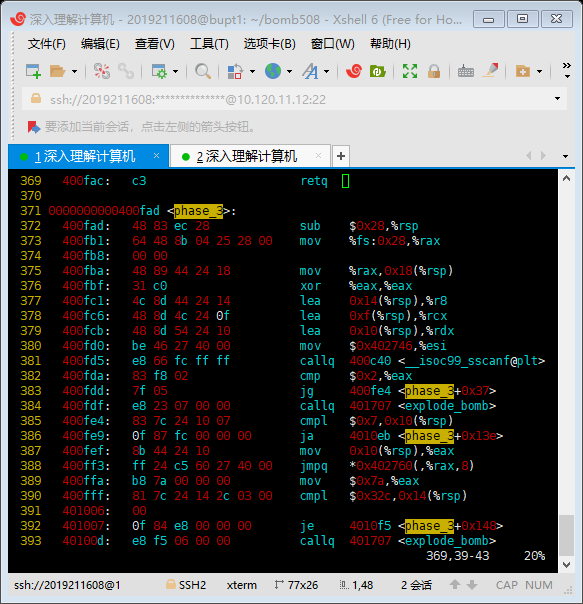




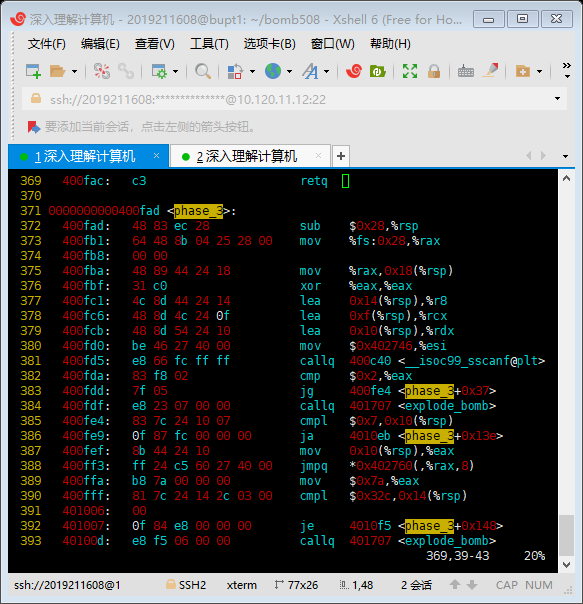
由此可以知道本关答案为1 2 4 8 16 32

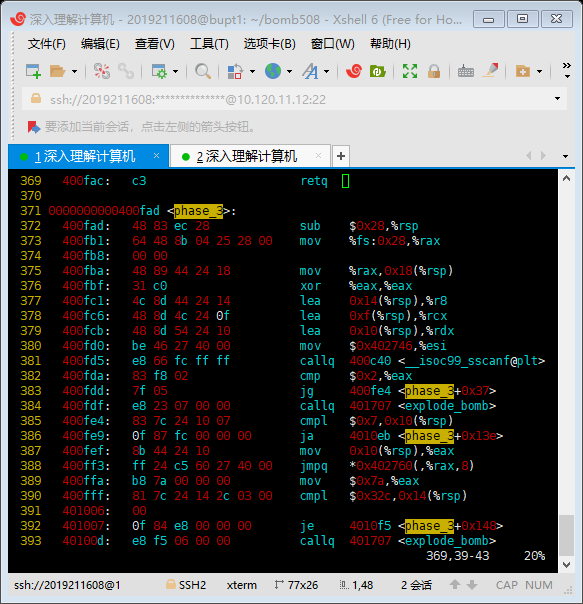
阶段3：

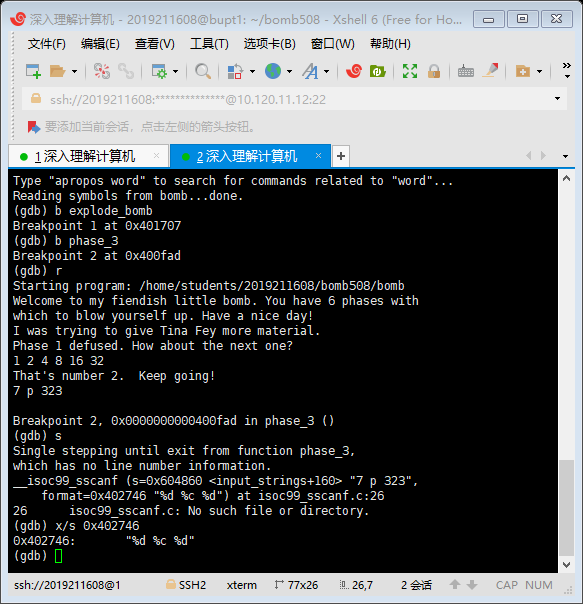
在bomb.s中找到对应phase\_3的函数位置

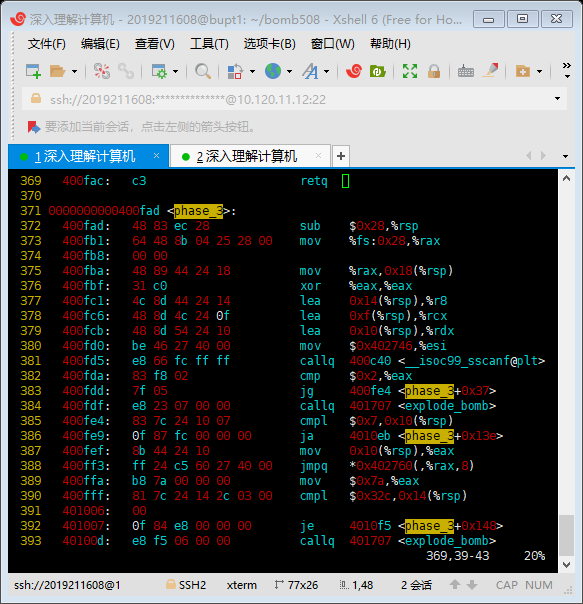


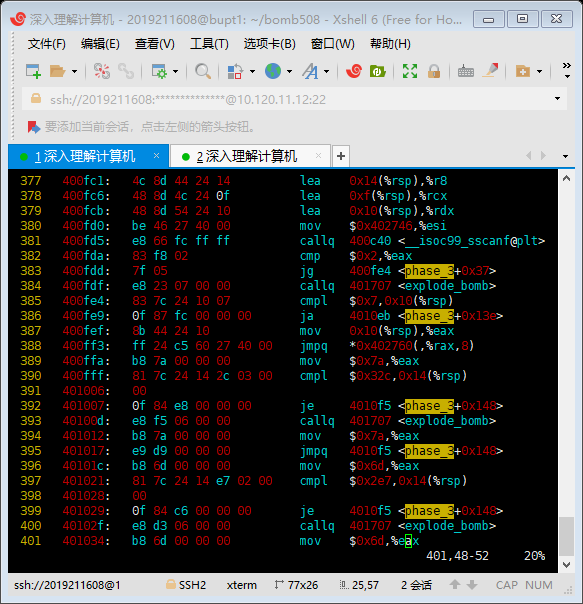
由下图可知本关要输入3个数



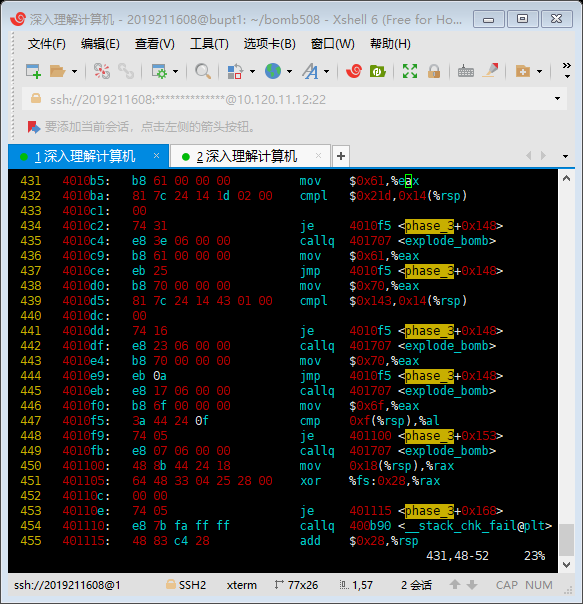


通过查看内存知道输入的三个数据类型分别为整型，字符型，整形

并由该句知道输入的第一个数要小于等于7

之后程序会通过输入的第一个数进行跳转

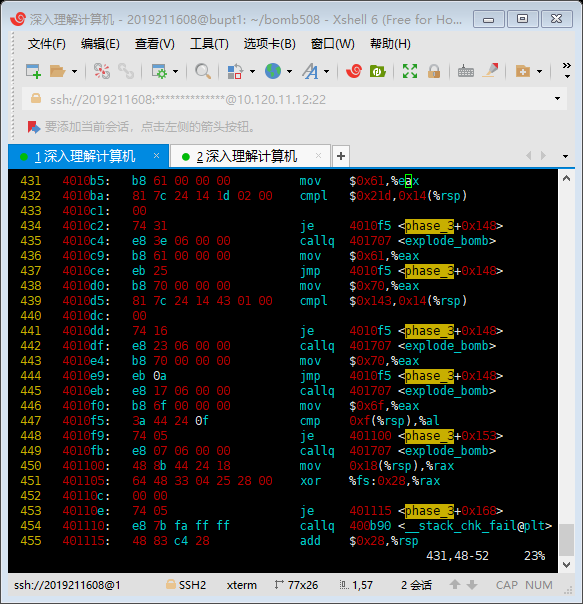
经观察，其后分别对应有8小段相似代码，可知该处为一个switch语句，不妨选择最后一段分析



可以看出输入的第二个数必须与0x143相等即十进制数323

此处也将寄存器eax的值赋值为了0x70即112

并由该段可知输入的字符必须与eax的值对应的ASCII码字符相等

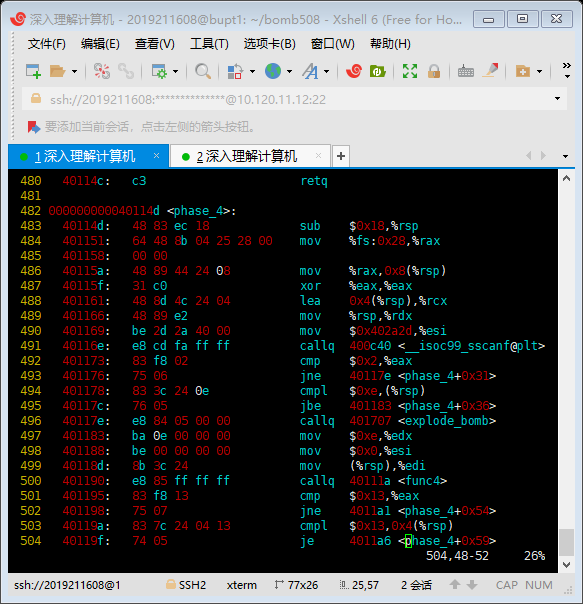


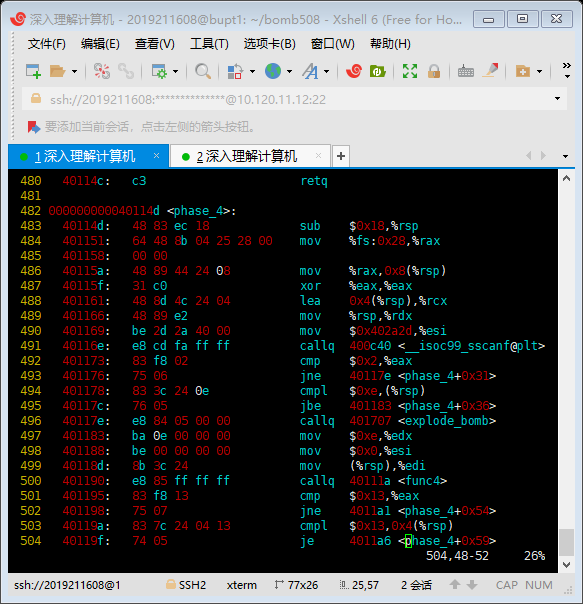
查表可得该字符为p

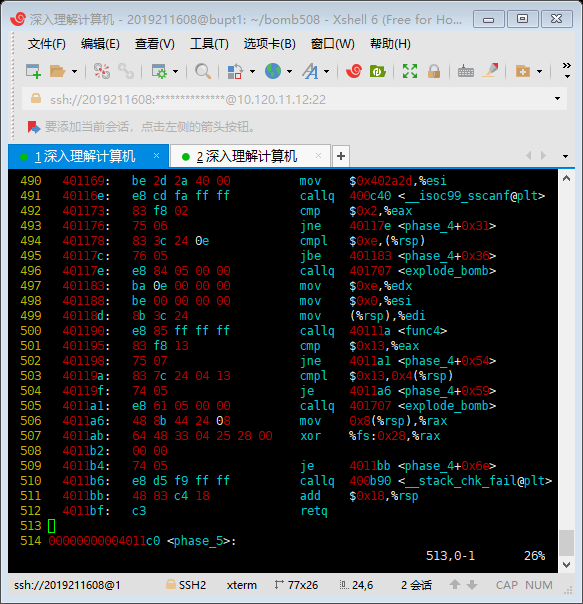
所以可知本对应的一组答案为7 p 323

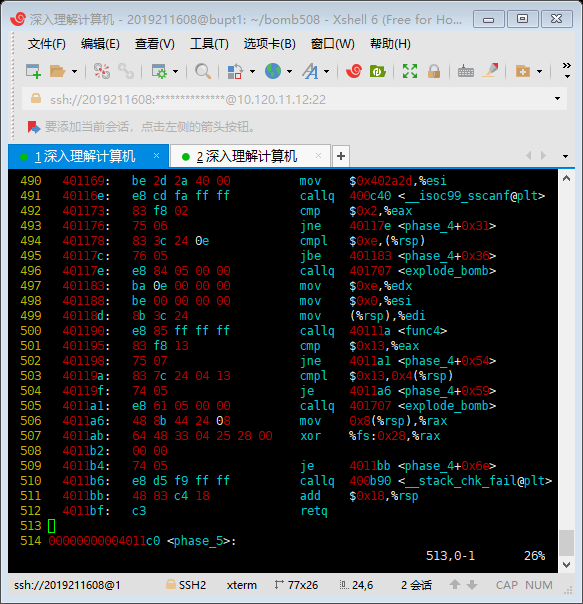
阶段4：

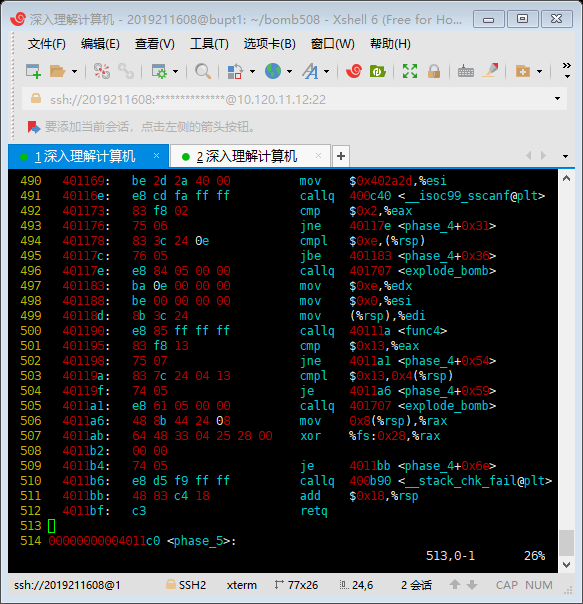
在bomb.s中找到对应phase\_4的函数位置



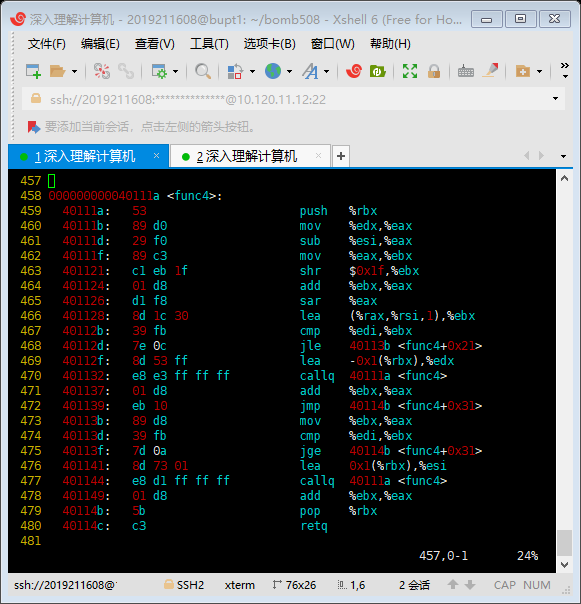
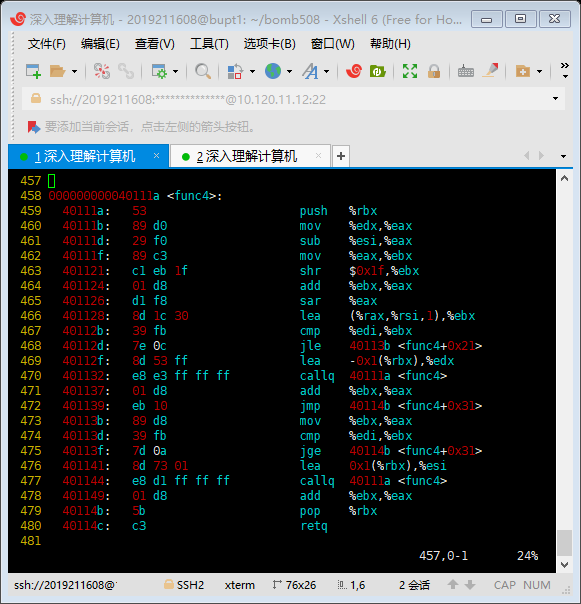
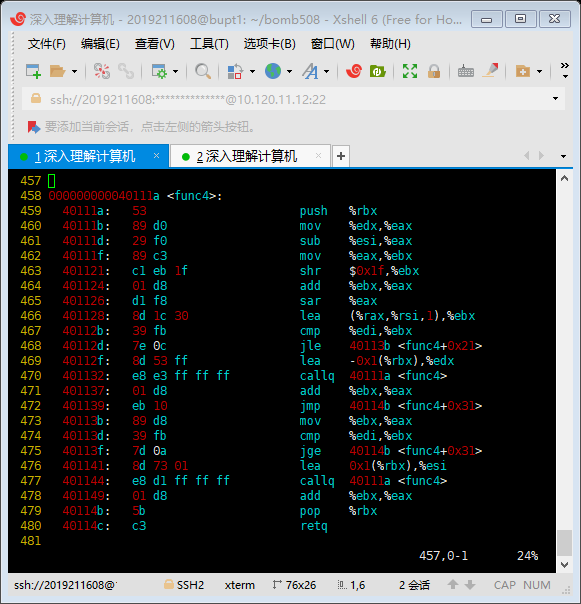
由该处可知本关要输入2个数据

并且第一个数必须小于等于14

之后传入func4函数并且要求返回值必须为19

然后输入的第二个数必须等于19

然后观察func4函数，可以知道其为一个递归函数，根据大于等于小于有3中不同情况

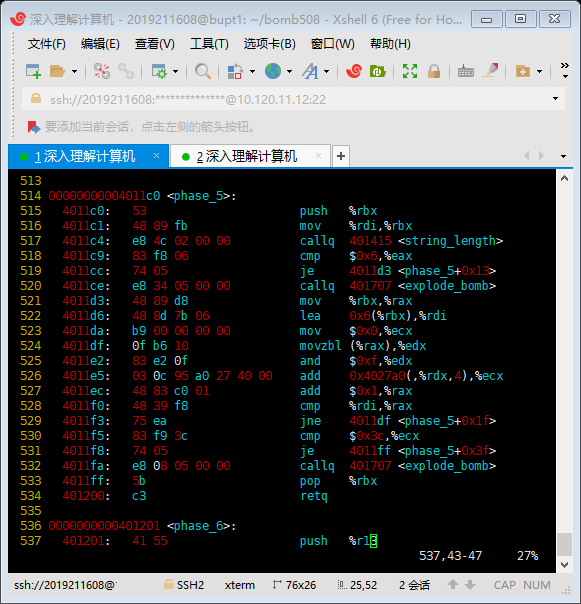


经运算知道应该输入4

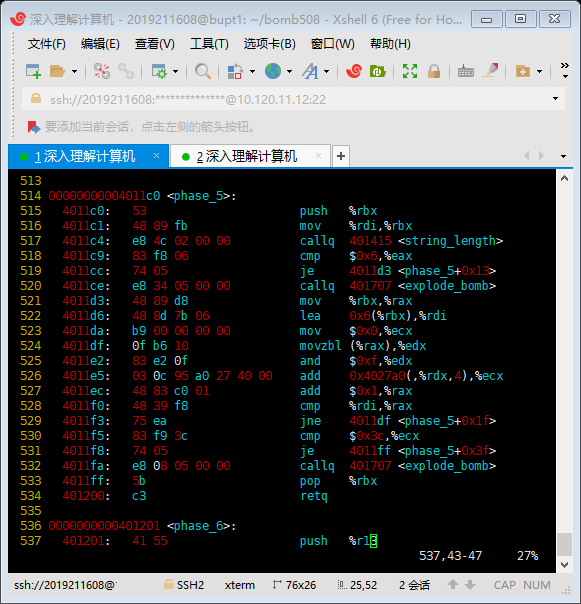
故本关答案为4 19

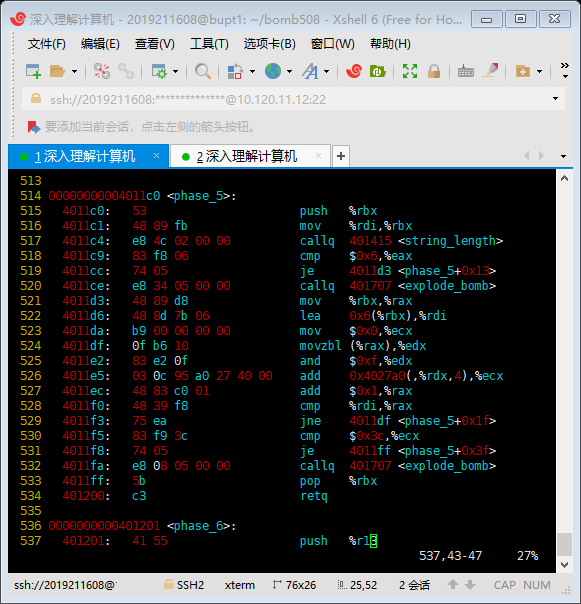
阶段5：

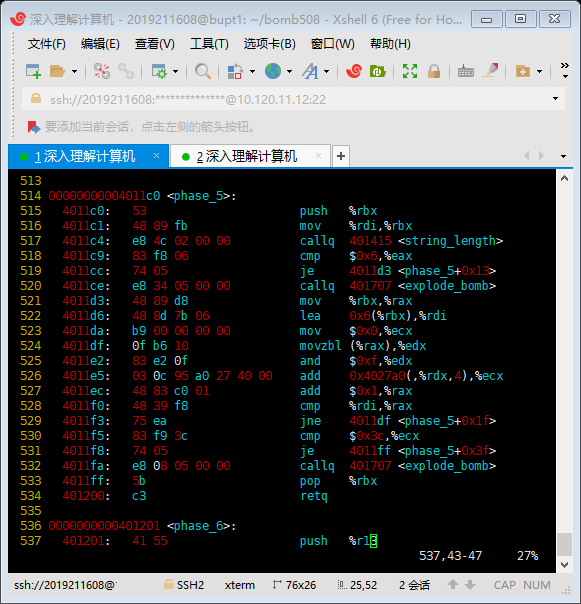
在bomb.s中找到对应phase\_5的函数位置

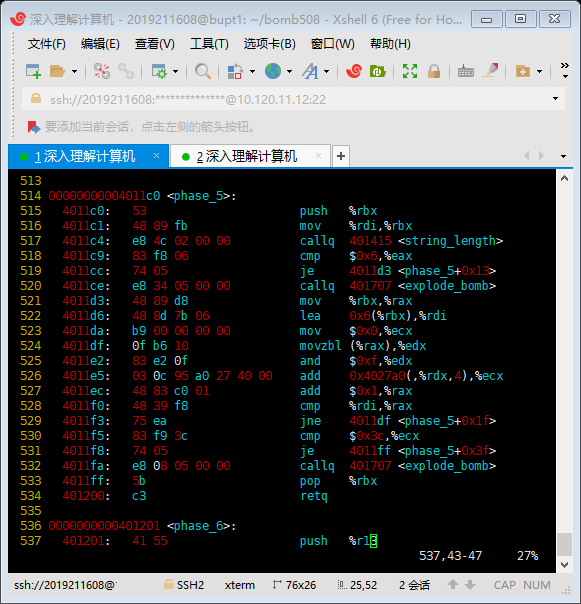


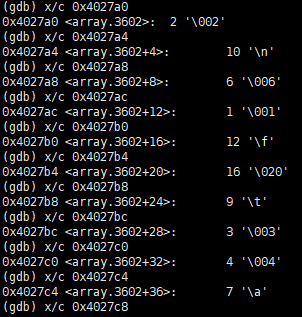
首先可以看到 本关要求输入一个长度为6的字符串

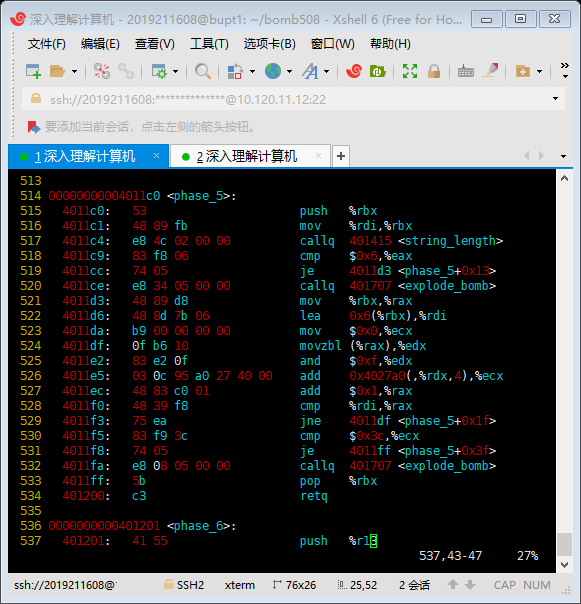


又该段代码可知，该程序有一个需要进行6次的循环

每次会对一个字符对应值取最后8位所对应的值

然后根据该值做偏移量取内存上的值进行累加

查看内存可知对应的值为

最后，所得的累加值必须为60才能避免爆炸

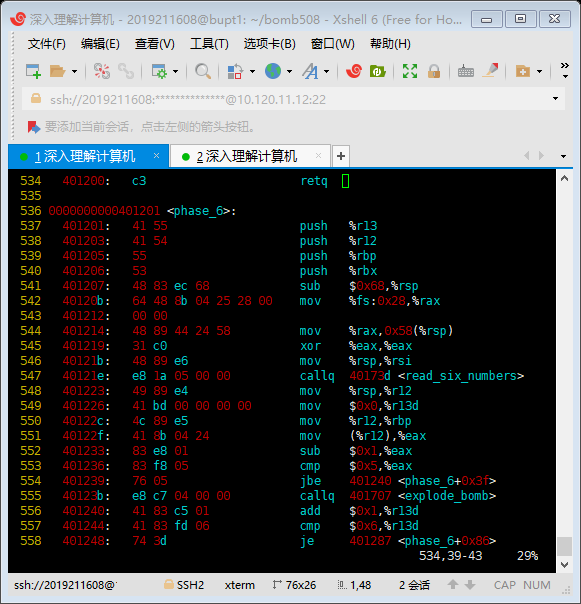
偏移量为1时值为10，所以只需要六个字符偏移量均为1所得值得求和即为60

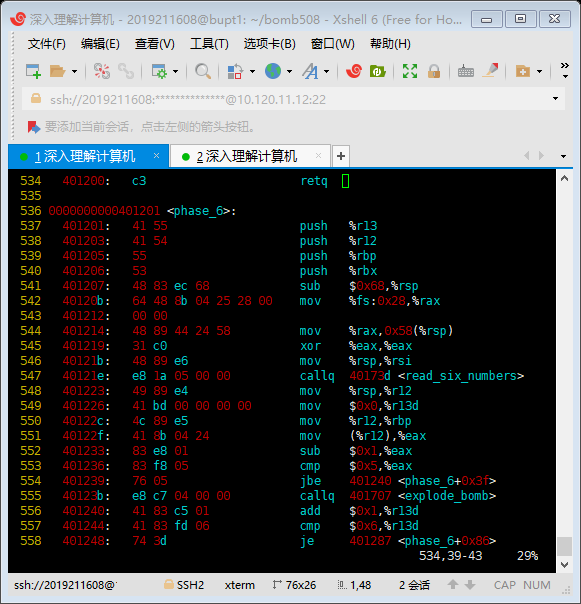
查看ASCII表可知字符a而97=0x61,取偏移量即为1

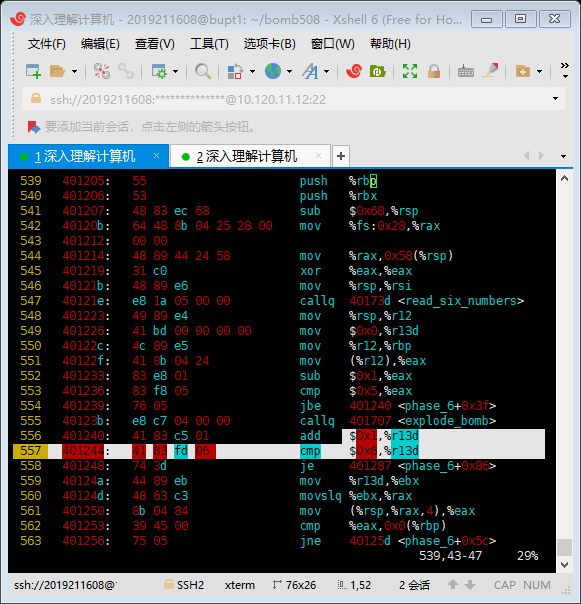
所以本关答案可为aaaaaa

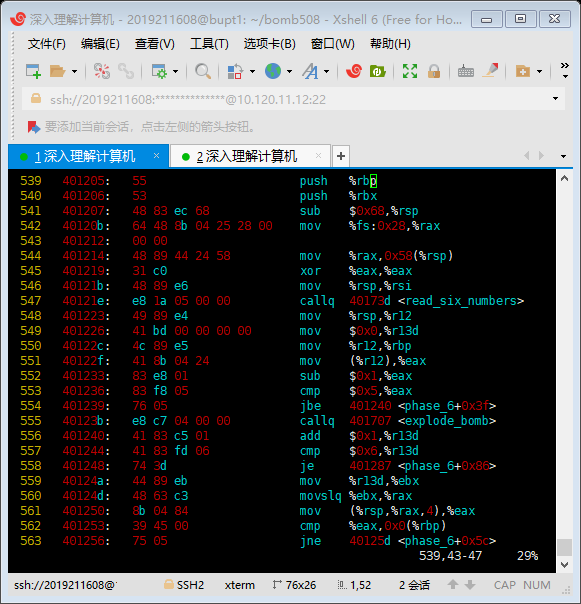
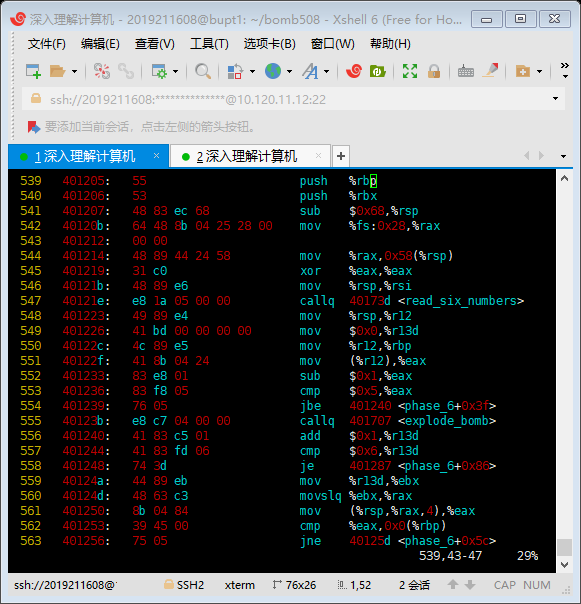
阶段6：

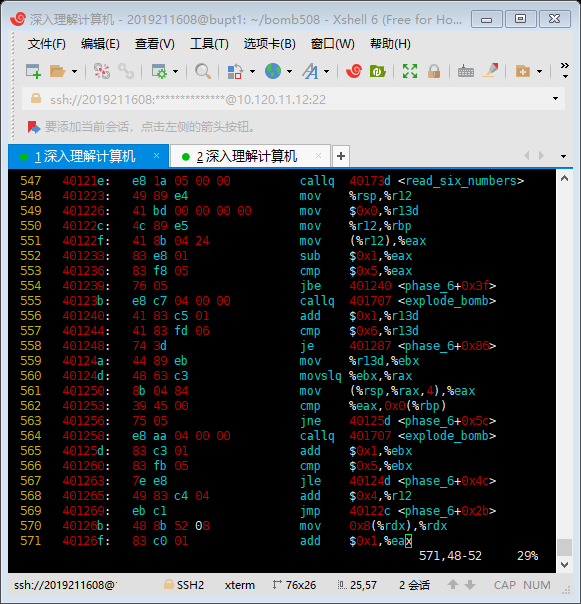
在bomb.s中找到对应phase\_6的函数位置



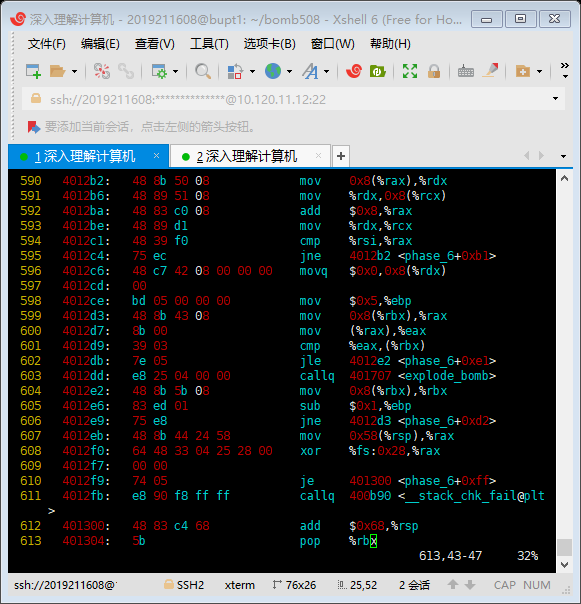
首先要读入6个数字

读入的数字必须小于等于6

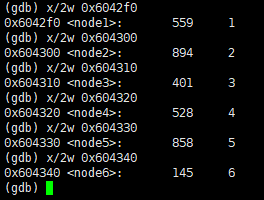
每个数字都执行该检查

并且每个数字互不相等

最后由下图的代码可以知道，依据输入对应内存的值需按升序排列



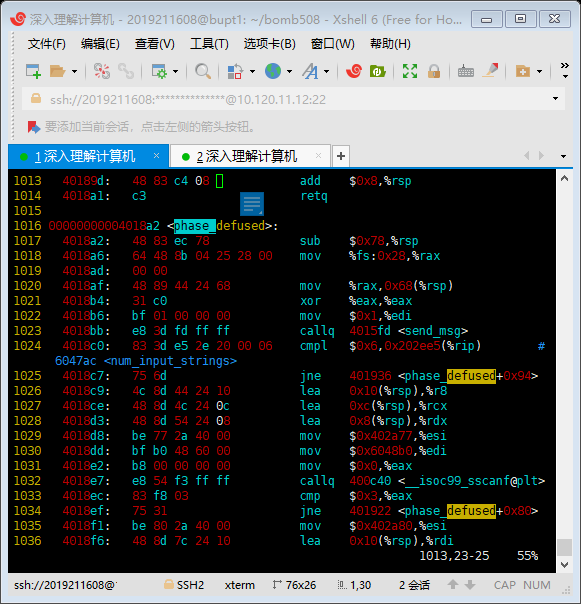
通过查看内存可以知道对应值为，由node标签也可以知道这是一个链表

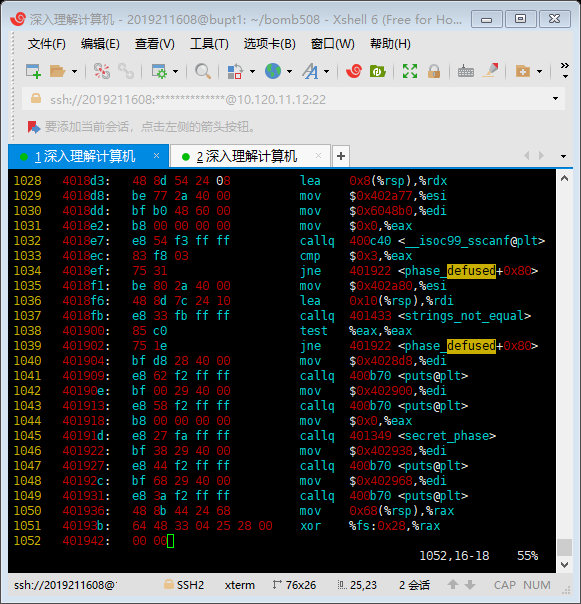


依次排出答案序列6 3 4 1 5 2

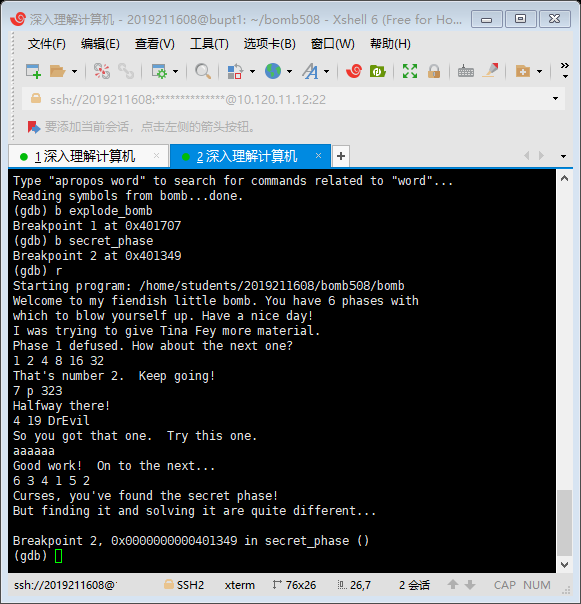
阶段7：

通过查看每关都会调用的phase\_defused函数



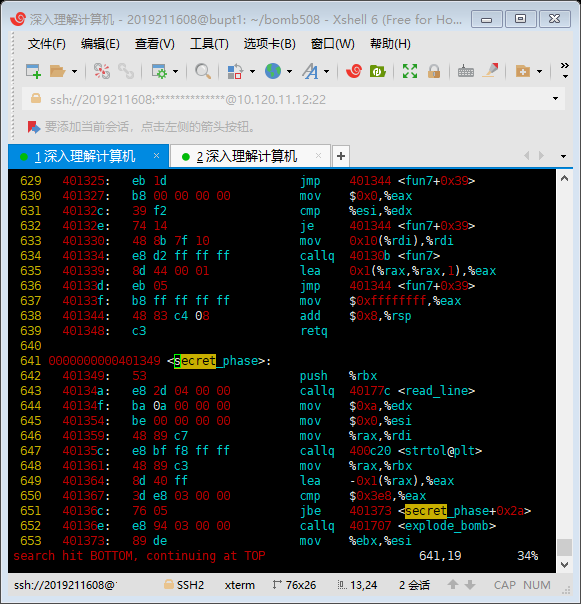
可以找到其中包含了一个secret\_phase函数的调用

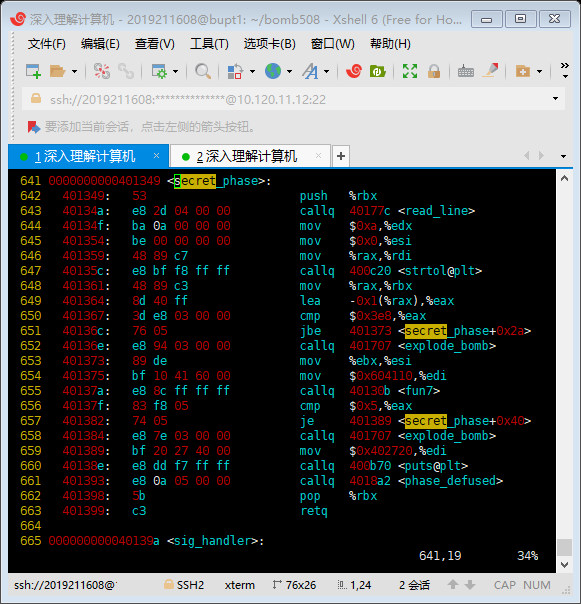
往上查找看到有一个字符串匹配函数，只有当字符串匹配时才会调用secret\_phase函数通过查看内存知道对应字符串为并且该处输入为两个整形和这个字符串，分析每个关卡的答案可以知道，需要在4关多输入一个DrEvil

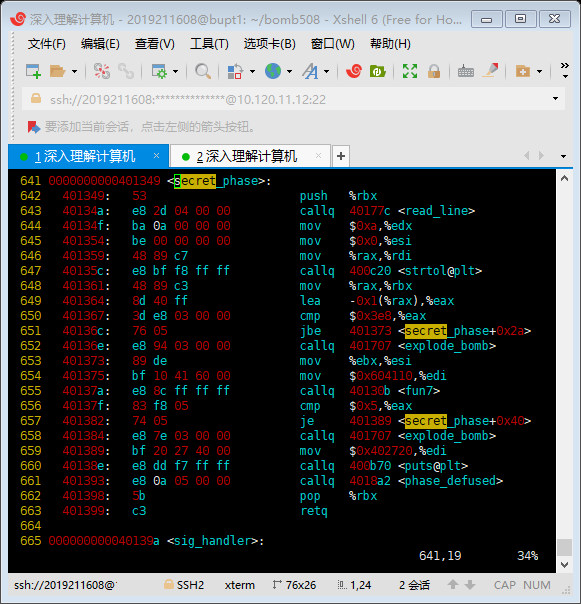


成功进入隐藏关卡

然后在bomb.s中找到对应secret\_phase的函数位置

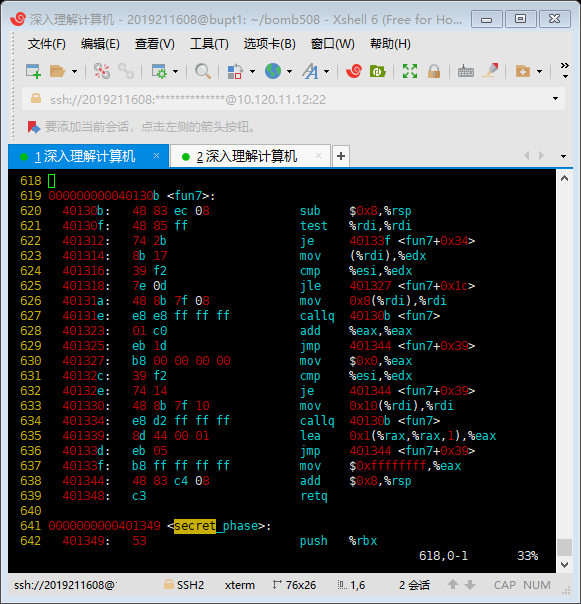
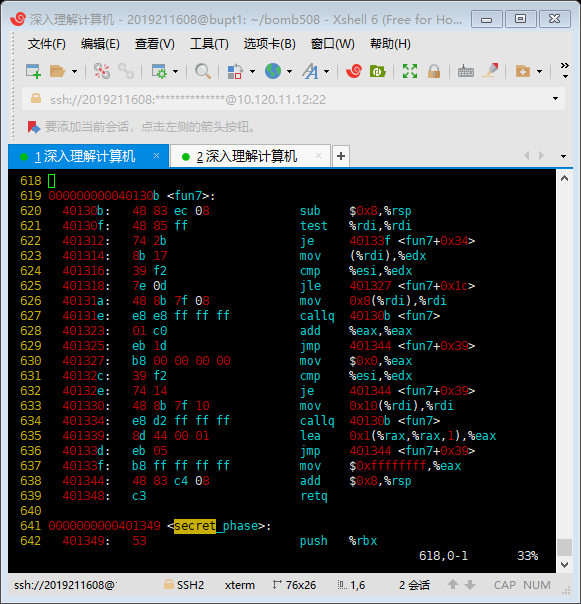


首先看到输入的数必须小于等于0x3e8

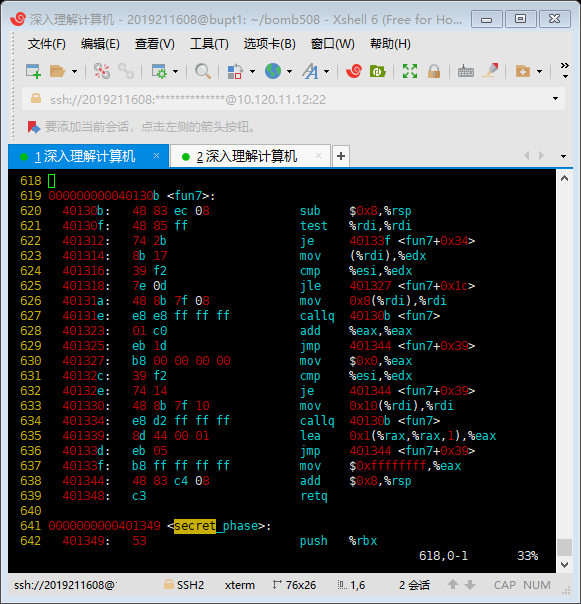
然后调用fun7函数，其返回值必须为5

然后分析fun7函数

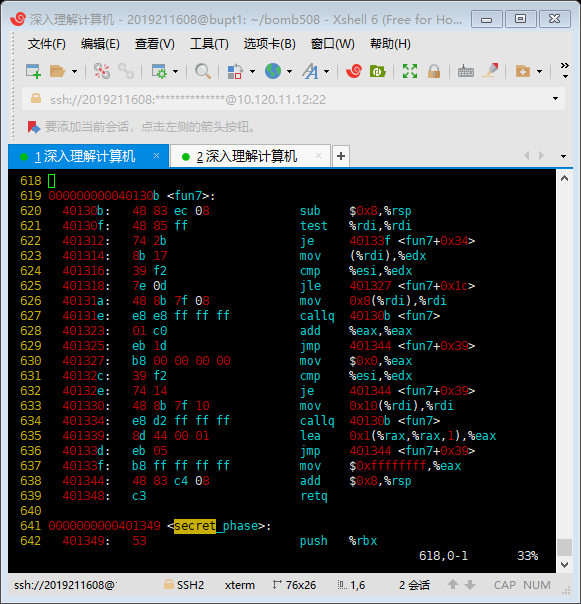
当值为0时返回-1



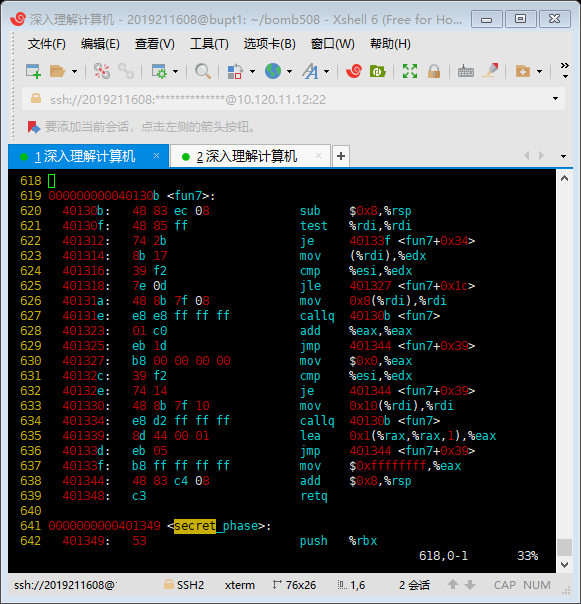
当相等时返回0



当大于参数时返回2\*fun7



当小于等于参数时返回2\*fun7+1

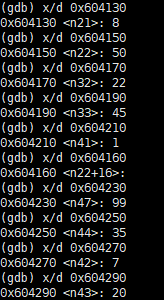


分析可知该函数为一个二叉树搜索的递归函数

5={2\*[2\*（1+2\*0）]+1}即右->左->右

查看内存可知





构成的二叉树如下形状

36

8 50

6 22 45 107

1 7 20 35 40 47 99 1001

故本关的答案为47。

到此bomb完全拆除。

五、总结体会

要拆除一个炸弹，需对每一关的反汇编代码进行认真的分析，注意每个变量的变化，寄存器值的变化，并且许多关卡的关键点在一些内存地址的初始化，查看该处的值往往有较大的帮助。

实验过程中使用gdb调试时为了避免炸弹爆炸，可以在explode\_bomb函数前设置断点。

在分析得到的答案不正确时可以尝试用类c语言将函数大体框架还原，这样对此处关卡的思路开拓有较大的帮助

六、诚信声明（不签扣10分）

需要填写如下声明，并在底部给出手写签名的电子版。

在完成本次实验过程中，我曾分别与以下各位同学就以下方面做过交流：

1、简单描述交流内容，例如：来自\*\*\*的建议，采用\*\*\*方式\*\*\*

2、

此外，我还参考了以下资料：

1. 网址等

在我提交的程序中，还在对应的位置以注释形式记录了具体的参考内容。

我独立完成了本次实验除以上方面之外的所有工作，包括分析、设计、编码、调试与测试。

我清楚地知道，从以上方面获得的信息在一定程度上降低了实验的难度，可能影响起评分。

我从未使用他人代码，不管是原封不动地复制，还是经过某些等价转换。

我未曾也不会向同一课程（包括此后各届）的同学复制或公开我这份程序的代码，我有义务妥善保管好它们。

我编写这个程序无意于破坏或妨碍任何计算机系统的正常运行。

我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按照0分计。

（签名）