

**实验报告**

**实 验（三）**

题 目 Binary Bomb

二进制炸弹

专 业 计算机类

学　　 号 1190200526

班　　 级 1903002

学 生 沈城有

指 导 教 师 郑贵滨

实 验 地 点 G709

实 验 日 期 2021.04.23

**计算机科学与技术学院**

**目 录**

[第1章 实验基本信息 - 3 -](#_Toc69310788)

[1.1 实验目的 - 3 -](#_Toc69310789)

[1.2 实验环境与工具 - 3 -](#_Toc69310790)

[1.2.1 硬件环境 - 3 -](#_Toc69310791)

[1.2.2 软件环境 - 3 -](#_Toc69310792)

[1.2.3 开发工具 - 3 -](#_Toc69310793)

[1.3 实验预习 - 3 -](#_Toc69310794)

[第2章 实验环境建立 - 4 -](#_Toc69310795)

[2.1 Ubuntu下CodeBlocks反汇编（10分） - 4 -](#_Toc69310796)

[2.2 Ubuntu下EDB运行环境建立（10分） - 4 -](#_Toc69310797)

[第3章 各阶段炸弹破解与分析 - 5 -](#_Toc69310798)

[3.1 阶段1的破解与分析 - 5 -](#_Toc69310799)

[3.2 阶段2的破解与分析 - 5 -](#_Toc69310800)

[3.3 阶段3的破解与分析 - 5 -](#_Toc69310801)

[3.4 阶段4的破解与分析 - 5 -](#_Toc69310802)

[3.5 阶段5的破解与分析 - 5 -](#_Toc69310803)

[3.6 阶段6的破解与分析 - 5 -](#_Toc69310804)

[3.7 阶段7的破解与分析(隐藏阶段) - 5 -](#_Toc69310805)

[第4章 总结 - 7 -](#_Toc69310806)

[4.1 请总结本次实验的收获 - 7 -](#_Toc69310807)

[4.2 请给出对本次实验内容的建议 - 7 -](#_Toc69310808)

[参考文献 - 8 -](#_Toc69310809)

# 第1章 实验基本信息

## 1.1 实验目的

熟练掌握计算机系统的ISA指令系统与寻址方式；

熟练掌握Linux下调试器的反汇编调试跟踪分析机器语言的方法；

增强对程序机器级表示、汇编语言、调试器和逆向工程等的理解。

## 1.2 实验环境与工具

### 1.2.1 硬件环境

X64 CPU；2GHz；2G RAM；256GHD Disk 以上

### 1.2.2 软件环境

Windows7 64位以上

VirtualBox/Vmware 11以上

Ubuntu 16.04 LTS 64位/优麒麟 64位

### 1.2.3 开发工具

GDB/OBJDUMP；EDB；KDD等

## 1.3 实验预习

上实验课前，必须认真预习实验指导书（PPT或PDF）。

了解实验的目的、实验环境与软硬件工具、实验操作步骤，复习与实验有关的理论知识。

请写出C语言下包含字符串比较、循环、分支（含switch）、函数调用、递归、指针、结构、链表等的例子程序sample.c，生成执行程序sample.out。

用gcc -S或CodeBlocks或GDB或OBJDUMP等，反汇编，比较。

列出每一部分的C语言对应的汇编语言。

修改编译选项-O (缺省2)、O0、O1、O2、O3，-m32/m64。再次查看生成的汇编语言与原来的区别。

注意O1之后无栈帧，EBP做别的用途。-fno-omit-frame-pointer加上栈指针。

GDB命令详解 –tui模式 ^XA切换 layout改变等等

有目的地学习: 看VS的功能GDB命令用什么？

# 第2章 实验环境建立

## 2.1 Ubuntu下CodeBlocks反汇编（10分）

CodeBlocks运行hellolinux.c。反汇编查看printf函数的实现。

要求：C、ASM、内存(显示hello等内容)、堆栈（call printf前）、寄存器同时在一个窗口。

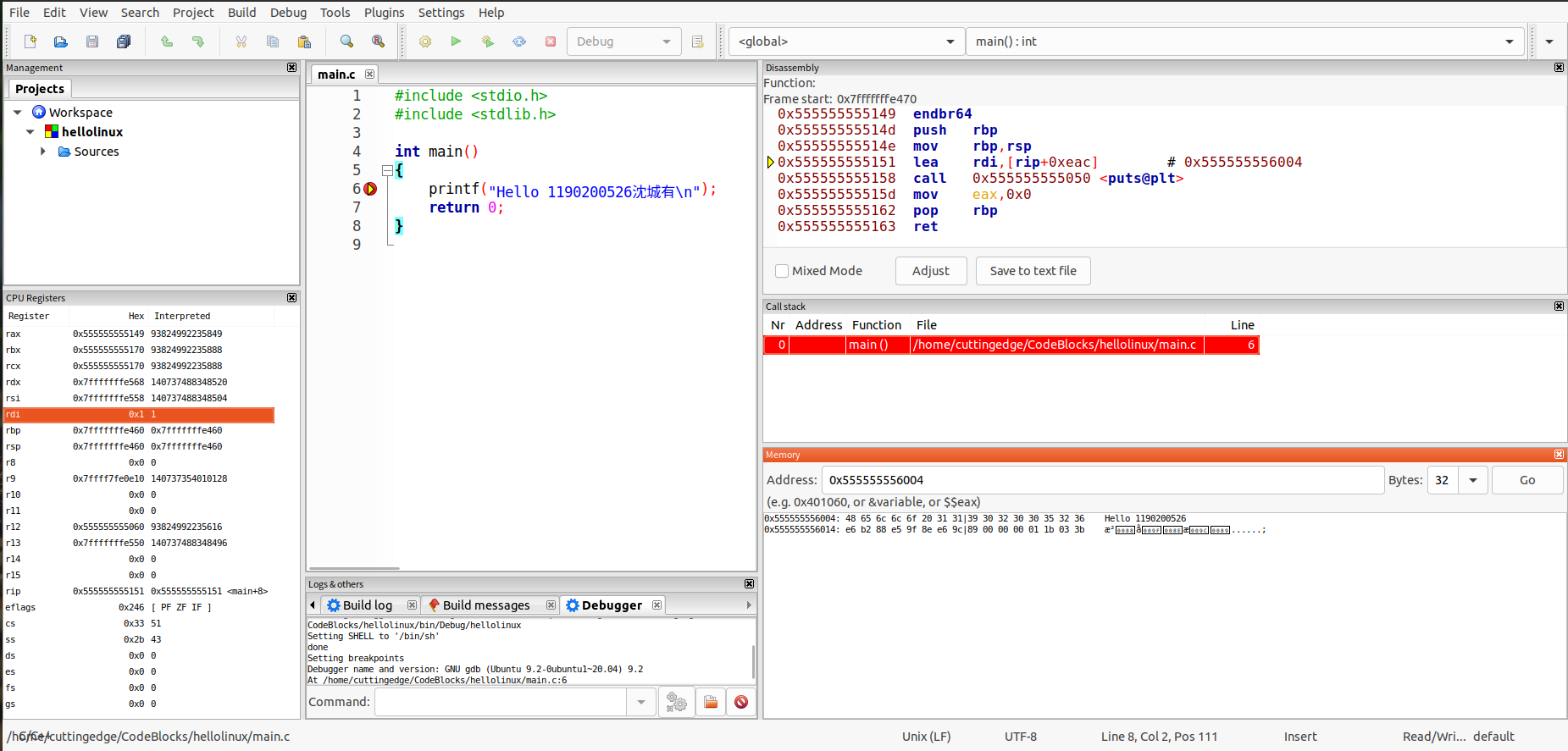


图2-1 Ubuntu下CodeBlocks反汇编截图

## 2.2 Ubuntu下EDB运行环境建立（10分）

用EDB调试hellolinux.c的执行文件，截图，要求同2.1

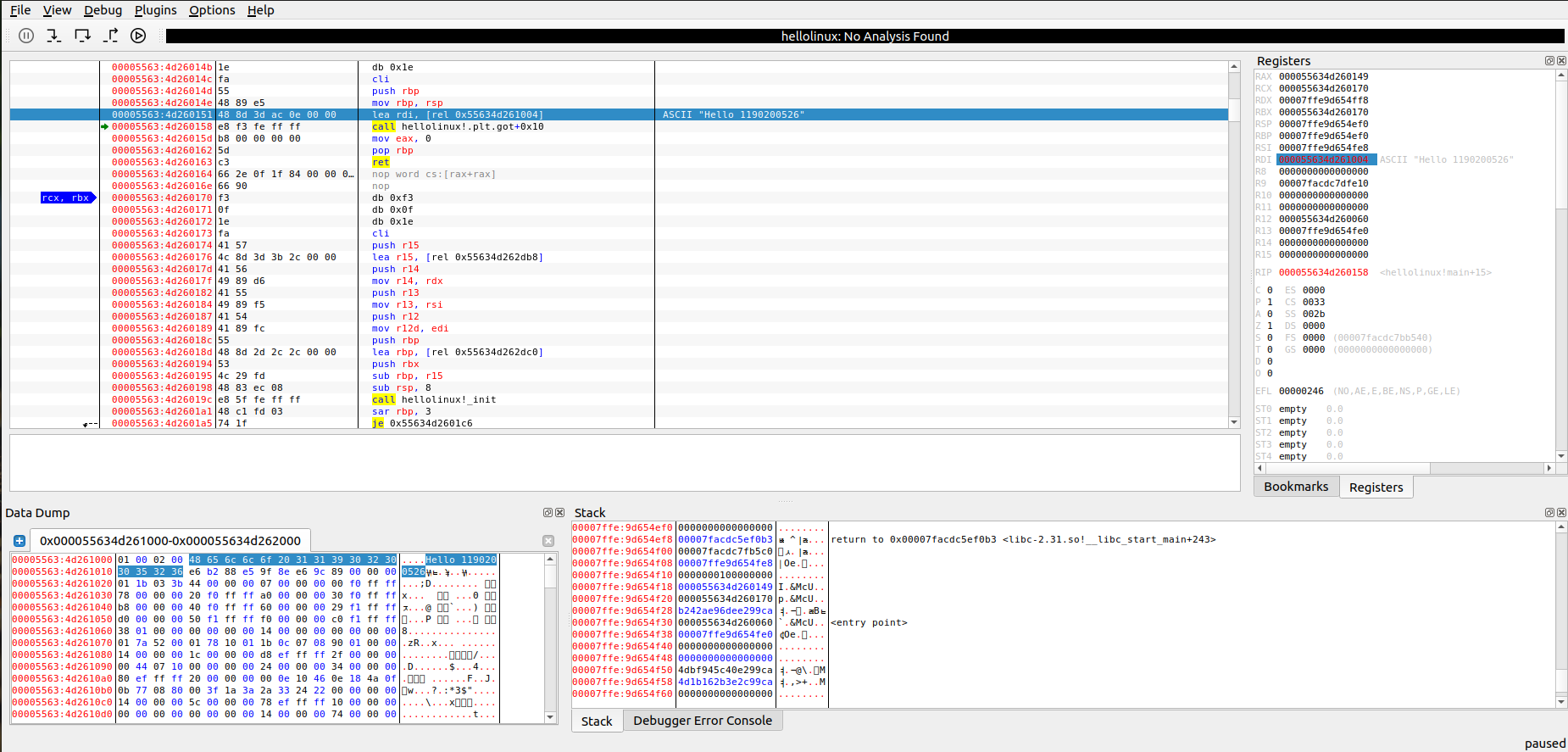


图2-2 Ubuntu下EDB截图

# 第3章 各阶段炸弹破解与分析

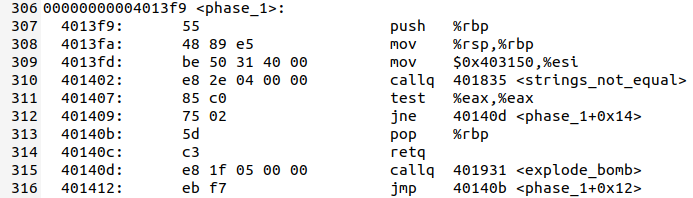
每阶段15分（密码10分，分析5分），总分不超过80分

## 3.1 阶段1的破解与分析

密码：I am for medical liability at the federal level.

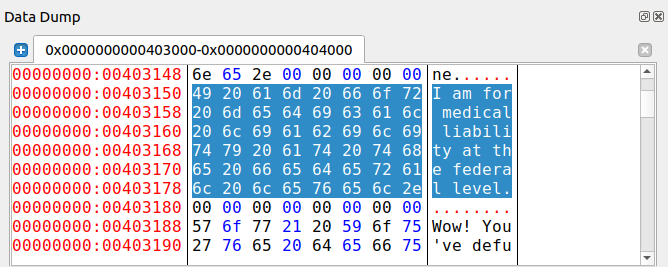
破解过程：

下图为phase\_1函数的汇编指令：



可知此函数将输入字符串与寄存器ESI指向的字符串进行比较，不相同则调用explode\_bomb函数，导致炸弹爆炸。

通过EDB查看对应内存位置字符串如下：

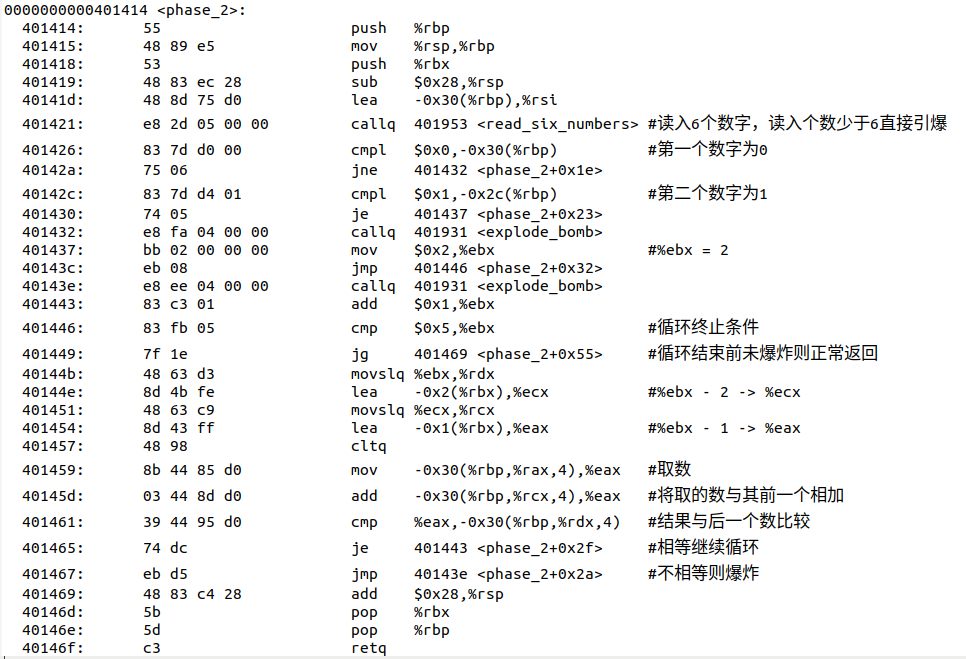


## 3.2 阶段2的破解与分析

密码：0 1 1 2 3 5

破解过程：

下图为phase\_2函数的汇编指令及分析：



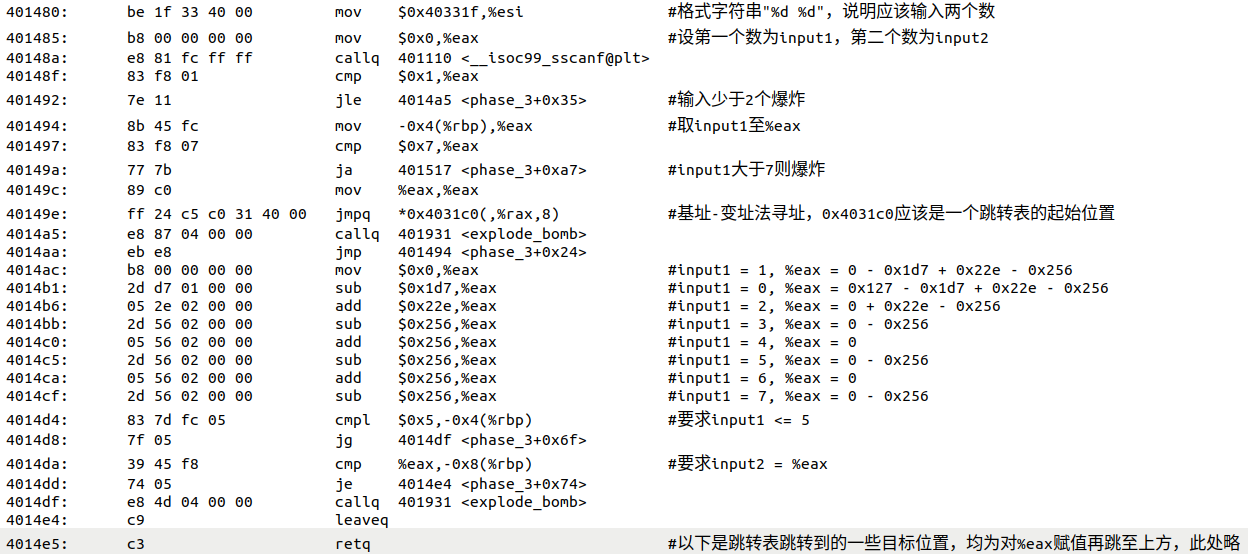
程序中有一个类似于检测斐波那契数列的循环体，且可知前两个数字分别为0、1，之后四个数字即可推出，分别为0+1=1，1+1=2，1+2=3，2+3=5。

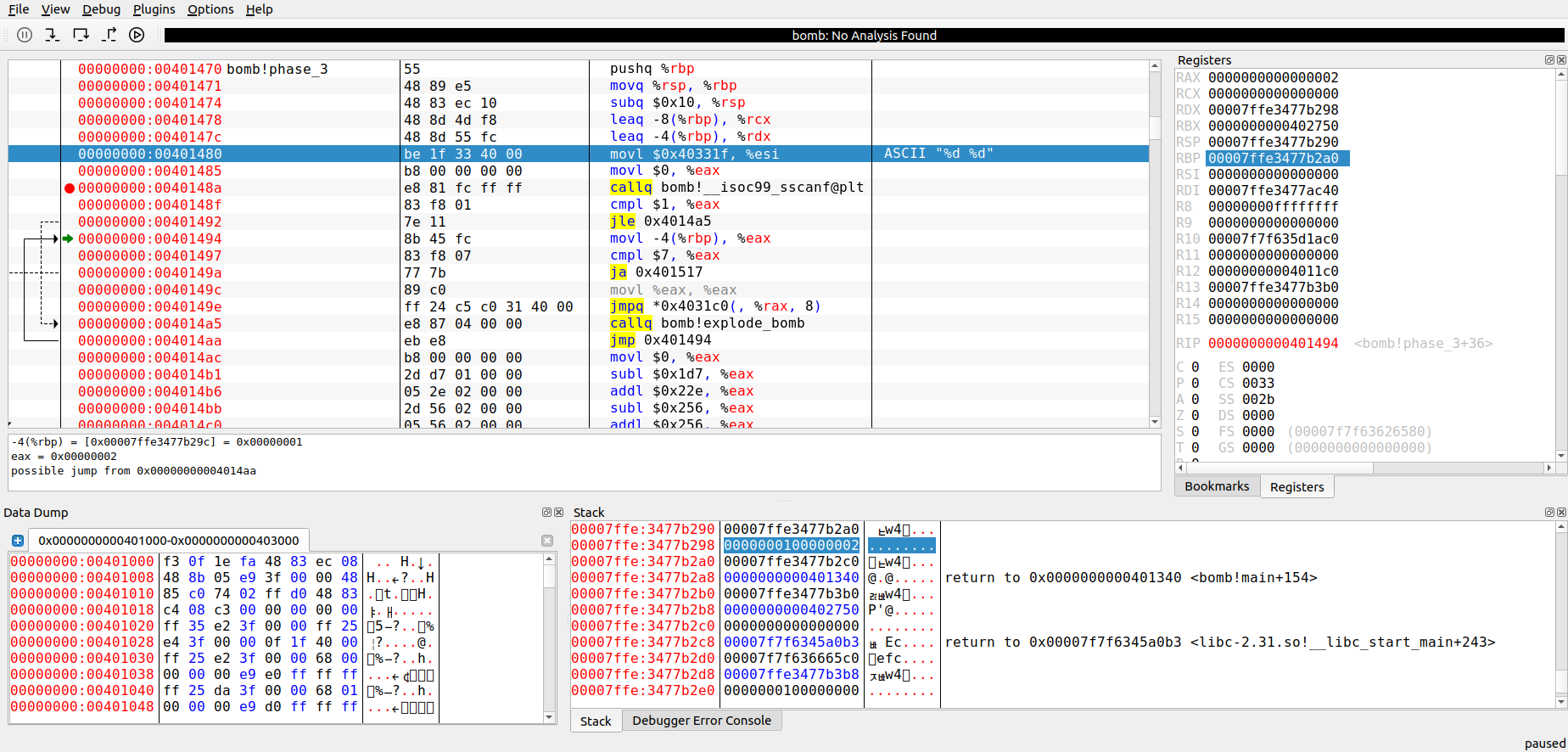
## 3.3 阶段3的破解与分析

密码：4 0 （注：其中一种可行数字组合）

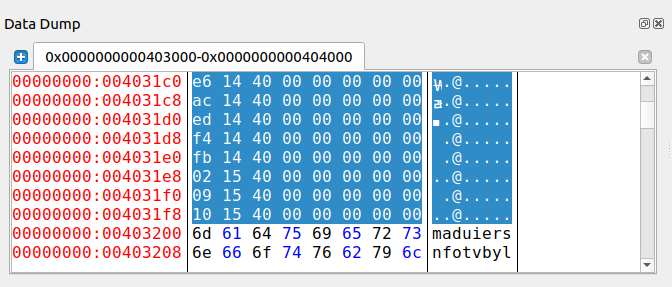
破解过程：

下图为phase\_3函数部分汇编代码及分析：





由上图，输入1 2进行测试，可知阶段3读入两个数，第一个数保存于(%rbp-4)指向的栈位置，第二个数保存于(%rbp-8)指向的栈位置。



上图为汇编代码分析中所提到的跳转表。结合汇编代码，能够确定两个输入数字之间的对应关系，且要求第二个数不大于5，故正确的输入数字组合共有6组，分别为：

[0 -216] [1 -511] [2 -40] [3 -598] [4 0] [5 -598]。

## 3.4 阶段4的破解与分析

密码：8 35

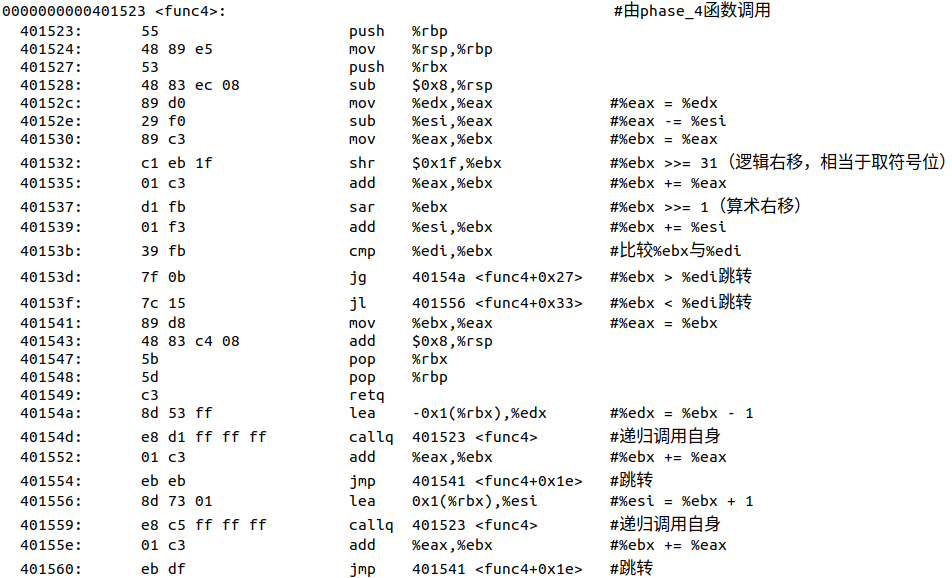
破解过程：

读入过程同阶段三，保存位置也同理。

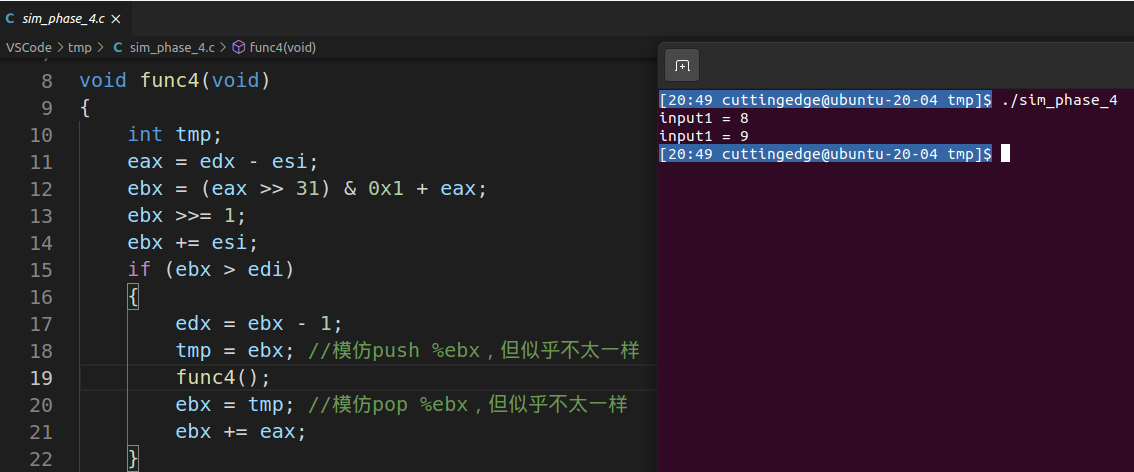
下图为phase\_4函数部分汇编代码及分析：



下图为递归调用的func4函数汇编代码及分析：



根据分析，我编写了C语言程序模拟这一过程并最终得到密码。



注：C程序中模拟的func4函数调用可能存在问题，但得到了正确答案8。

## 3.5 阶段5的破解与分析

密码：

破解过程：

## 3.6 阶段6的破解与分析

密码如下：

破解过程：

## 3.7 阶段7的破解与分析(隐藏阶段)

密码如下：

破解过程：

# 第4章 总结

## 4.1 请总结本次实验的收获

## 4.2 请给出对本次实验内容的建议

注：本章为酌情加分项。

# 参考文献

**为完成本次实验你翻阅的书籍与网站等**

[1] 林来兴. 空间控制技术[M]. 北京：中国宇航出版社，1992：25-42.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集：A集[C]. 北京：中国科学出版社，1999.

[3] 赵耀东. 新时代的工业工程师[M/OL]. 台北：天下文化出版社，1998 [1998-09-26]. http://www.ie.nthu.edu.tw/info/ie.newie.htm（Big5）.

[4] 谌颖. 空间交会控制理论与方法研究[D]. 哈尔滨：哈尔滨工业大学，1992：8-13.

[5] KANAMORI H. Shaking Without Quaking[J]. Science，1998，279（5359）：2063-2064.

[6] CHRISTINE M. Plant Physiology: Plant Biology in the Genome Era[J/OL]. Science，1998，281：331-332[1998-09-23]. http://www.sciencemag.org/cgi/ collection/anatmorp.