

**实验报告**

**实 验（三）**

题 目 Binary Bomb

二进制炸弹

专 业 计算机类

学　　 号 120L022115

班　　 级 2003007

学 生 王炳轩

指 导 教 师 吴锐

实 验 地 点 线上

实 验 日 期 2022-04-01

**计算学部**

**目 录**

[第1章 实验基本信息 - 3 -](#_Toc19846)

[1.1 实验目的 - 3 -](#_Toc19505)

[1.2 实验环境与工具 - 3 -](#_Toc15894)

[1.2.1 硬件环境 **错误!未定义书签。**](#_Toc16679)

[1.2.2 软件环境 **错误!未定义书签。**](#_Toc1640)

[1.2.3 开发工具 **错误!未定义书签。**](#_Toc311)

[1.3 实验预习 - 3 -](#_Toc20293)

[第2章 实验环境建立 - 4 -](#_Toc3866)

[2.1 Ubuntu下CodeBlocks反汇编（10分） - 4 -](#_Toc24906)

[2.2 Ubuntu下EDB运行环境建立（10分） - 4 -](#_Toc18077)

[第3章 各阶段炸弹破解与分析 - 6 -](#_Toc4144)

[3.1 阶段1的破解与分析 - 6 -](#_Toc24389)

[3.2 阶段2的破解与分析 - 6 -](#_Toc25931)

[3.3 阶段3的破解与分析 - 7 -](#_Toc21845)

[3.4 阶段4的破解与分析 - 9 -](#_Toc11024)

[3.5 阶段5的破解与分析 - 10 -](#_Toc29641)

[3.6 阶段6的破解与分析 - 12 -](#_Toc18838)

[3.7 阶段7的破解与分析(隐藏阶段) - 13 -](#_Toc20276)

[第4章 总结 - 14 -](#_Toc9339)

[4.1 请总结本次实验的收获 - 14 -](#_Toc16563)

[4.2 请给出对本次实验内容的建议 - 14 -](#_Toc27793)

[参考文献 - 15 -](#_Toc10577)

# 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

Surface Go 3：x64、Pentium G6500Y @ 1.1GHz、16GB RAM、128GB SSD。

### 1.2.2 软件环境

Windows 11、Windows Subsystem for Linux、Ubuntu 20.04

### 1.2.3 开发工具

Code::Blocks、gcc、vim、edb、gdb

## 1.3 实验预习

上实验课前，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。

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。

# 第2章 实验环境建立

## 2.1 Ubuntu下CodeBlocks反汇编（10分）

CodeBlocks运行hello.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。

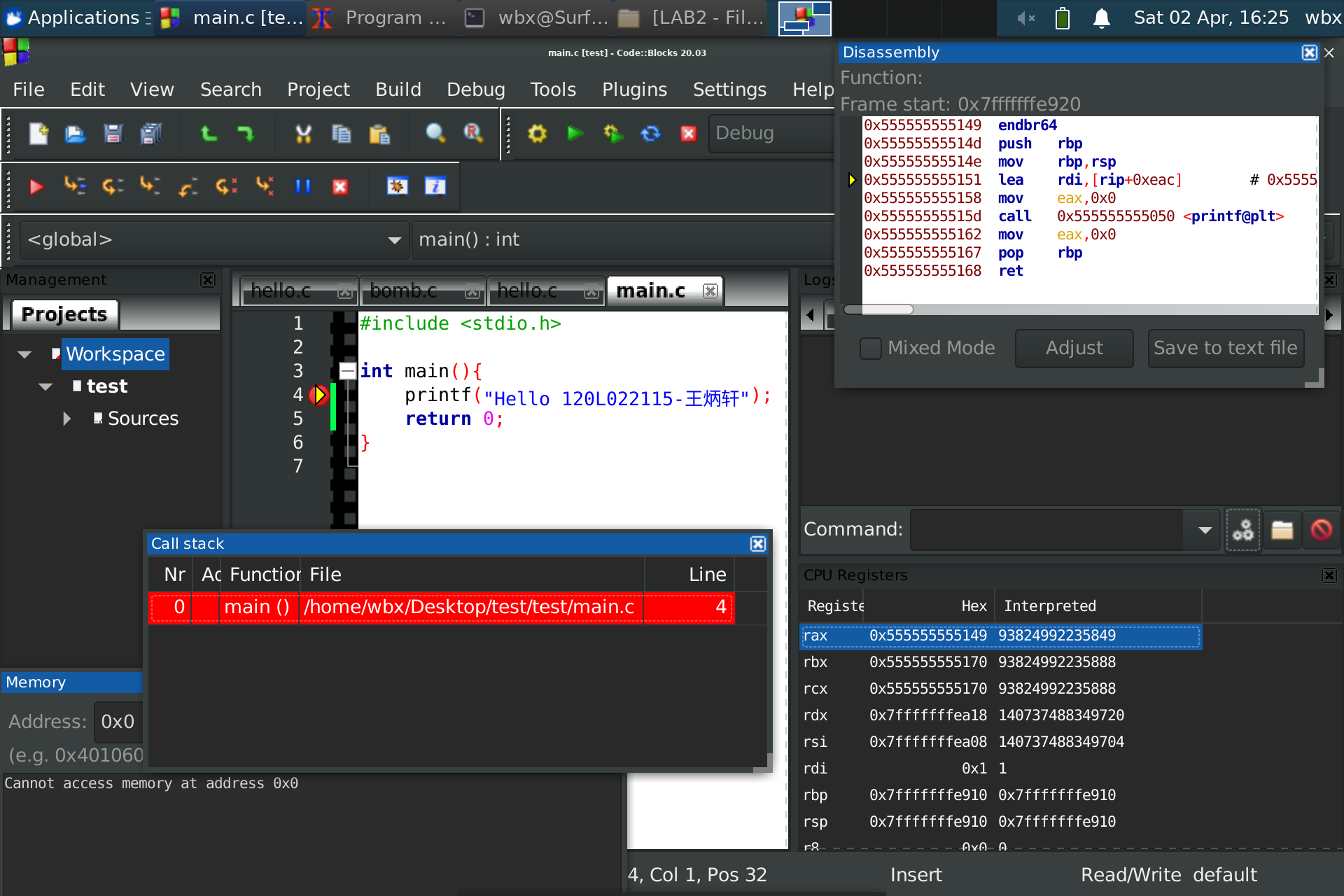


图2-1 Ubuntu下CodeBlocks反汇编截图

## 2.2 Ubuntu下EDB运行环境建立（10分）

用EDB调试hello.c的执行文件，截图，要求同2.1

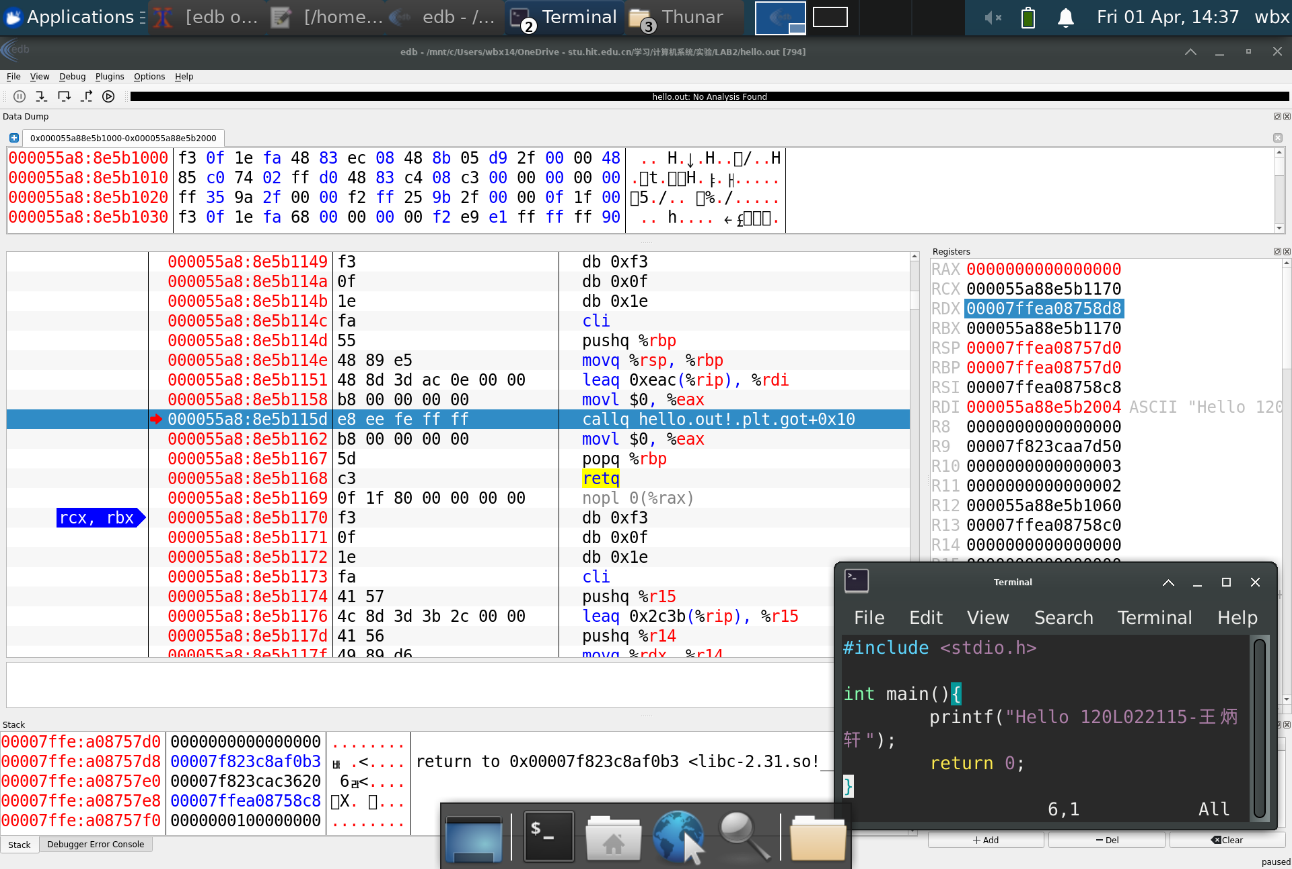


图2-2 Ubuntu下EDB截图

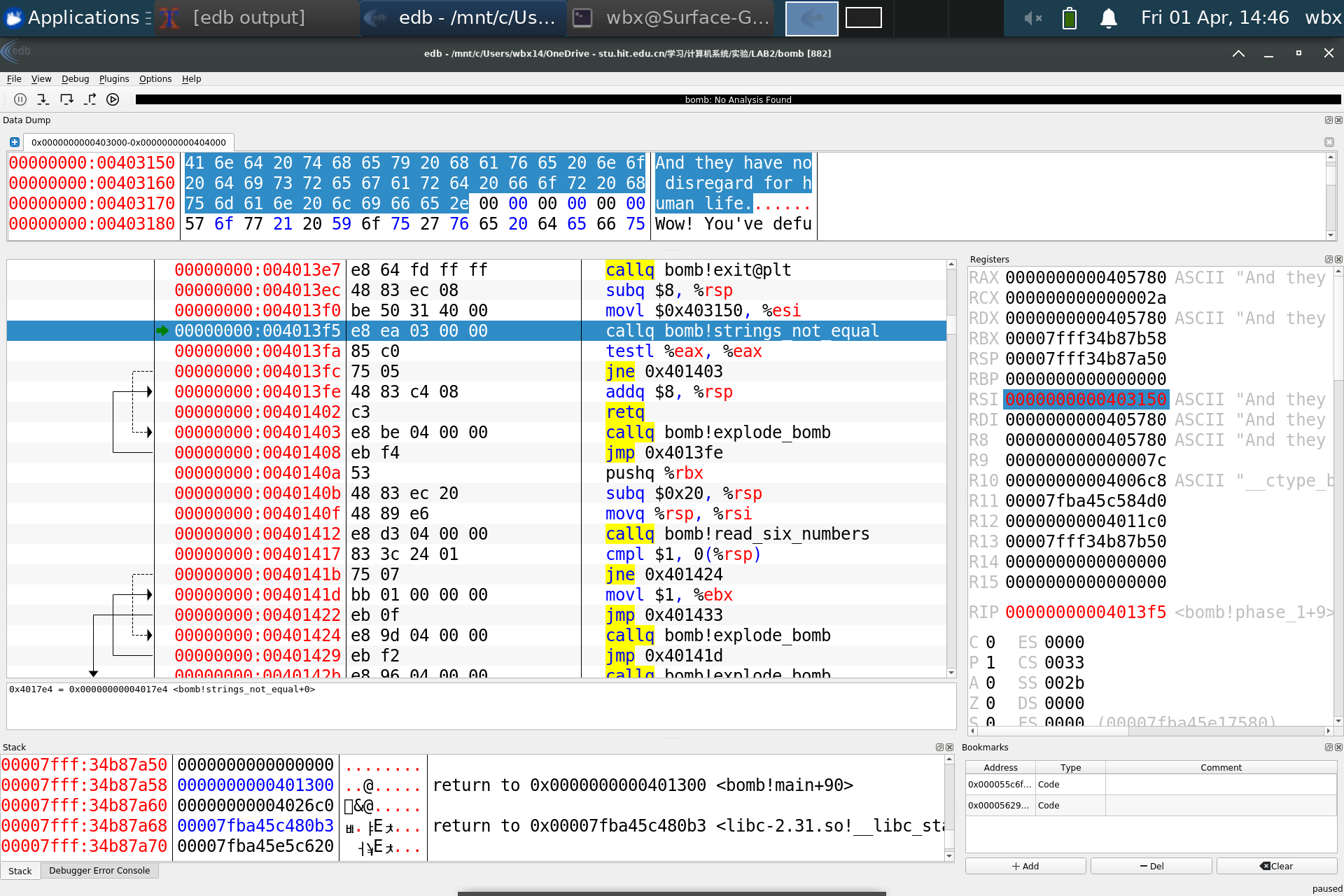
# 第3章 各阶段炸弹破解与分析

每阶段30分，密码10分，分析20分，总分不超过80分

## 3.1 阶段1的破解与分析

密码如下：And they have no disregard for human life.

破解过程：进入Phase\_1函数，发现为直接与0x403150地址的字符串进行比较，如果比较正确即破解，因此第一关的密码就是位于该地址的字符串，通过DataDump查看，得到密码。



## 3.2 阶段2的破解与分析

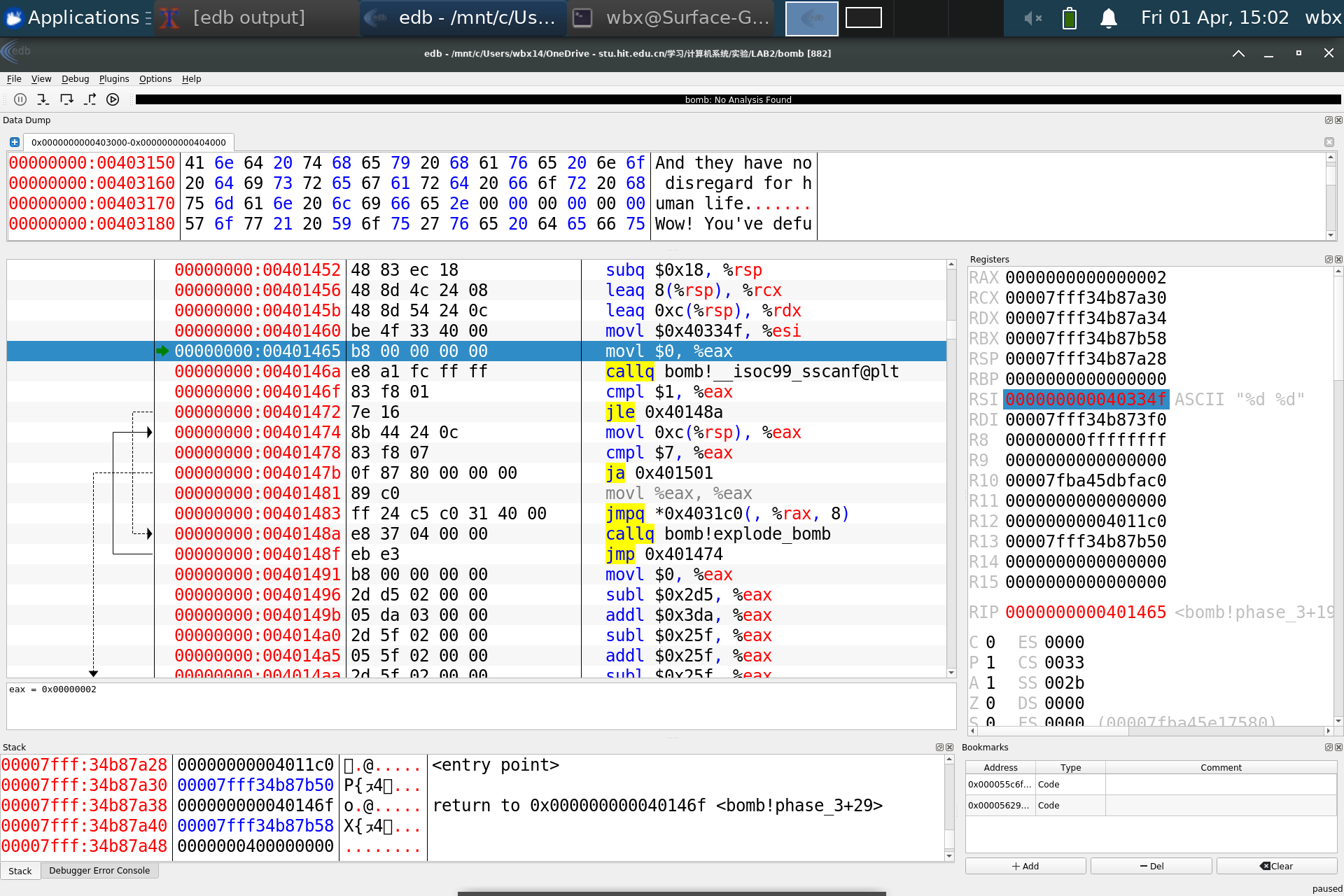
密码如下：1 2 4 8 16 32

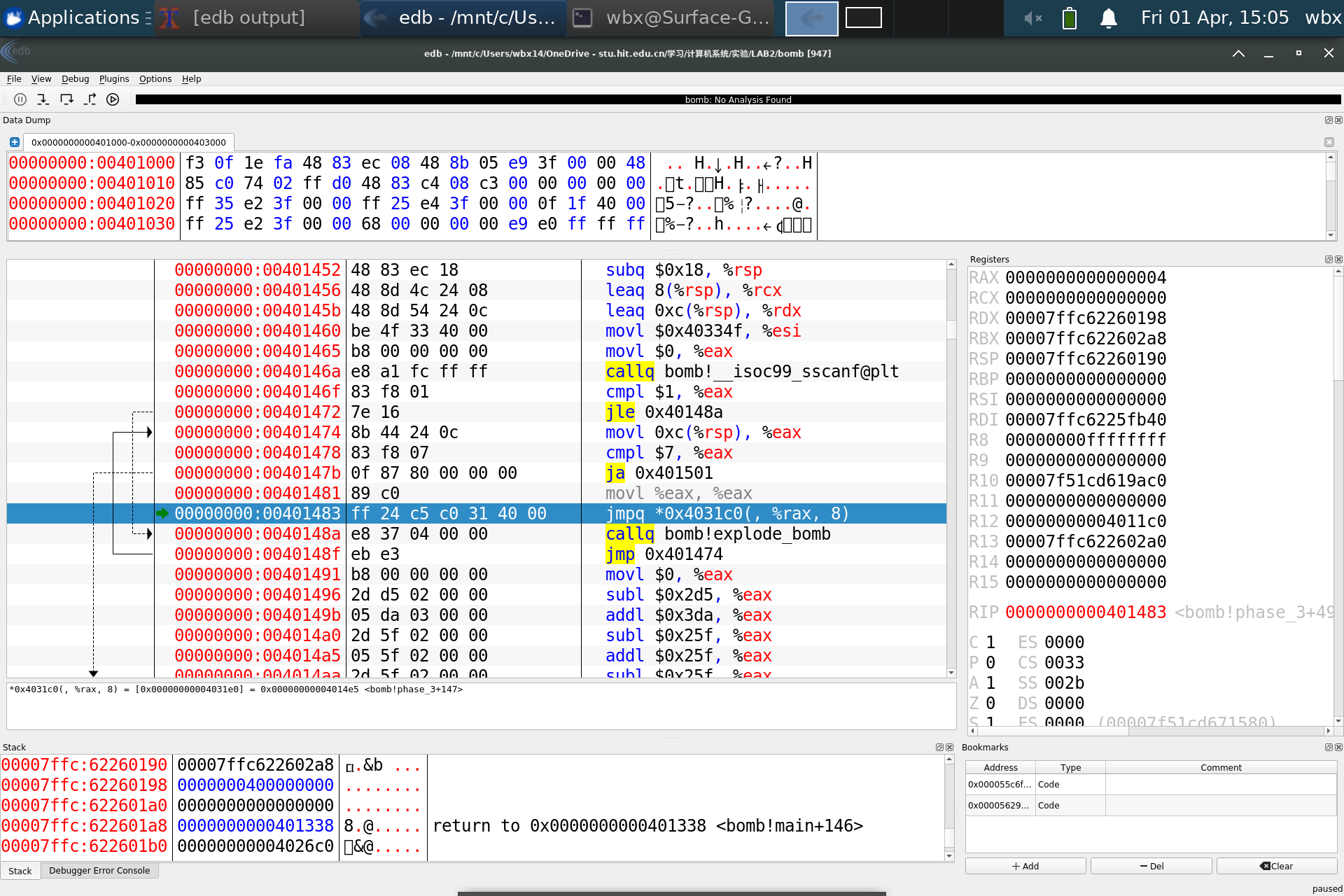
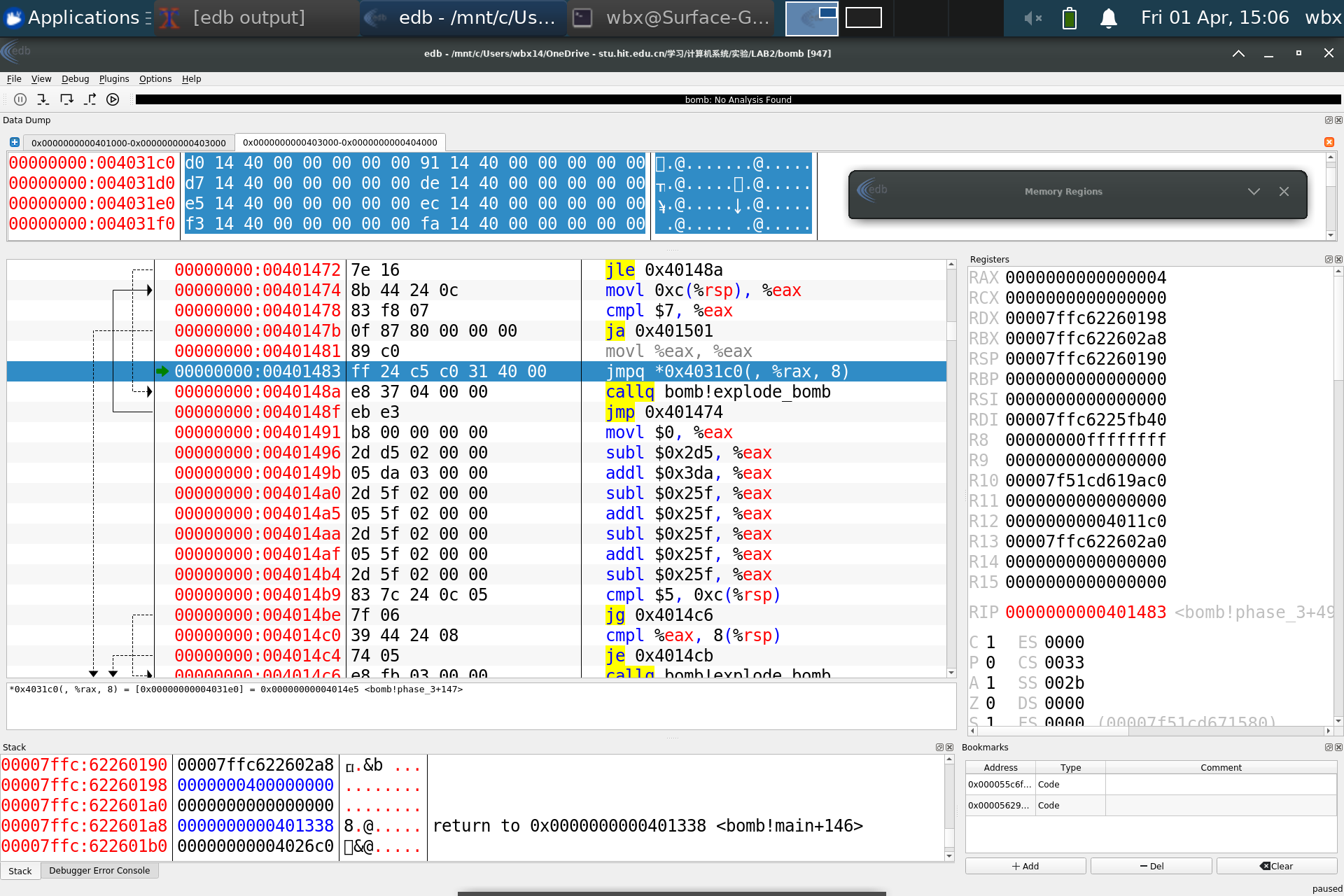
破解过程：进入Phase\_2函数，发现了一个读入6个数的函数，那么密码就一定是6个数。jne给出了第一个数为1，因为如果不等就会爆炸。之后发现有一个6次循环比较（ebx是循环次数，当为5的时候跳转到ret），6个数分别与eax比较，而eax从1开始、每次比较都翻倍，所以这6个数为1、2、4、8、16、32.

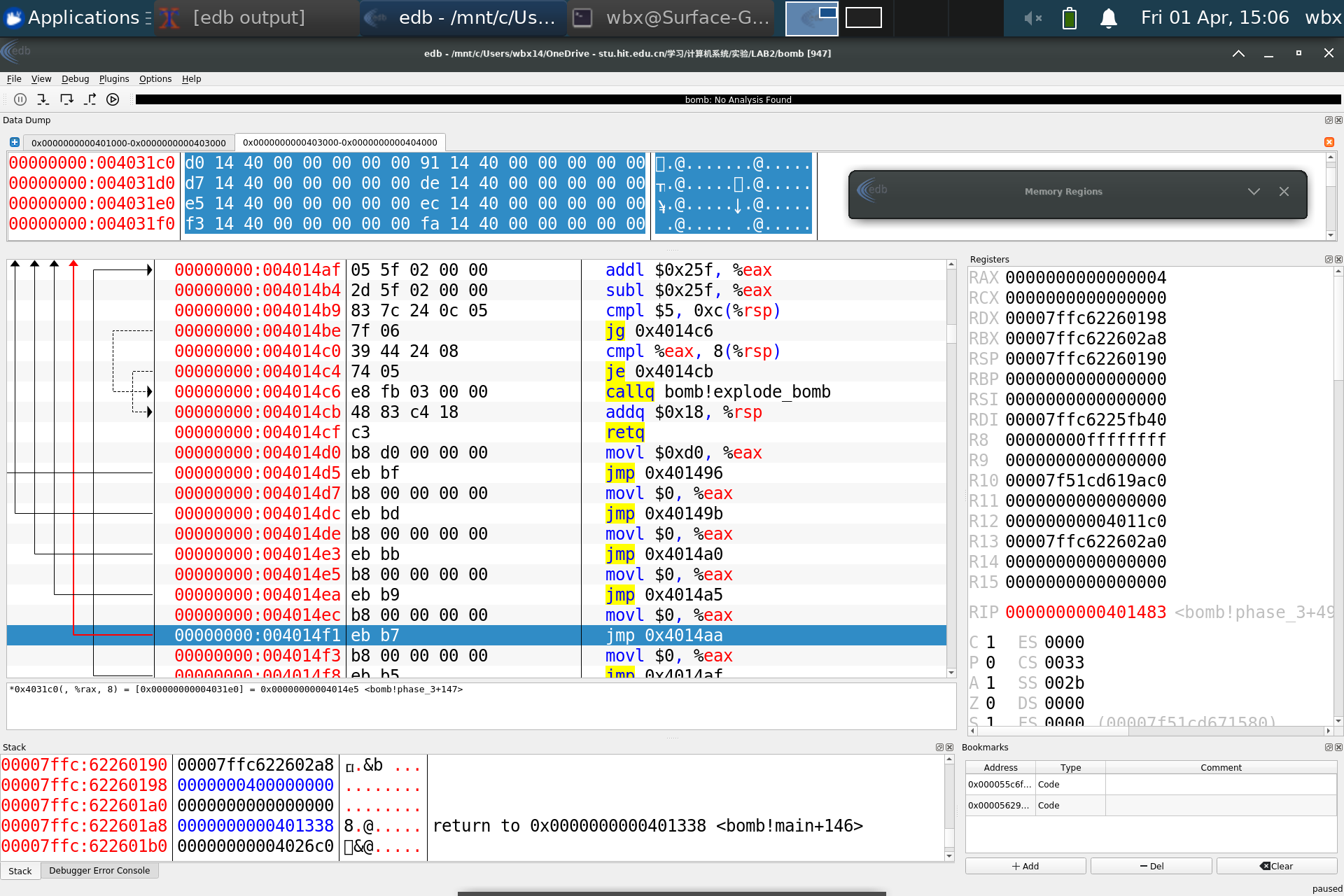
## 3.3 阶段3的破解与分析

密码如下：4 0（密码不唯一）

破解过程：进入Phase\_3函数，发现”%d %d”传入了scanf函数，那么密码就一定是2个数。j给出了第一个数必须小于7，之后是一个含有第二个数的地址寻址模式，经过Memory查看，发现了跳转列表，显然是一个switch语句。经过计算得到了第2个数为4才能跳转到正确的计算结果，使eax不超出范围（大于5），最后eax与第一个数进行比较，相等即可拆除，得到第一个数为0的结果。实际上，通过其他switch分支也有可能破解，但本次实验未要求（造成密码不唯一）。



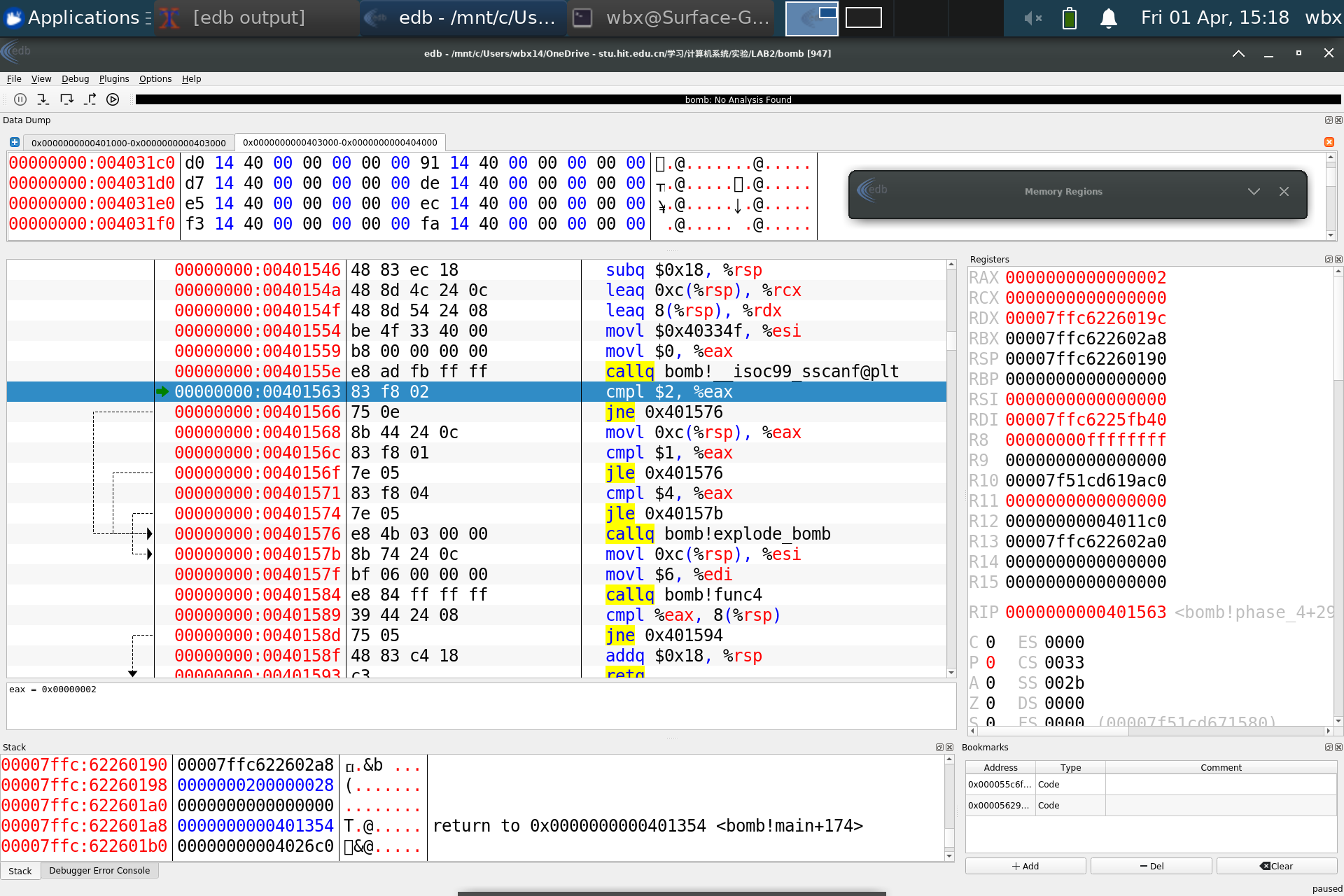


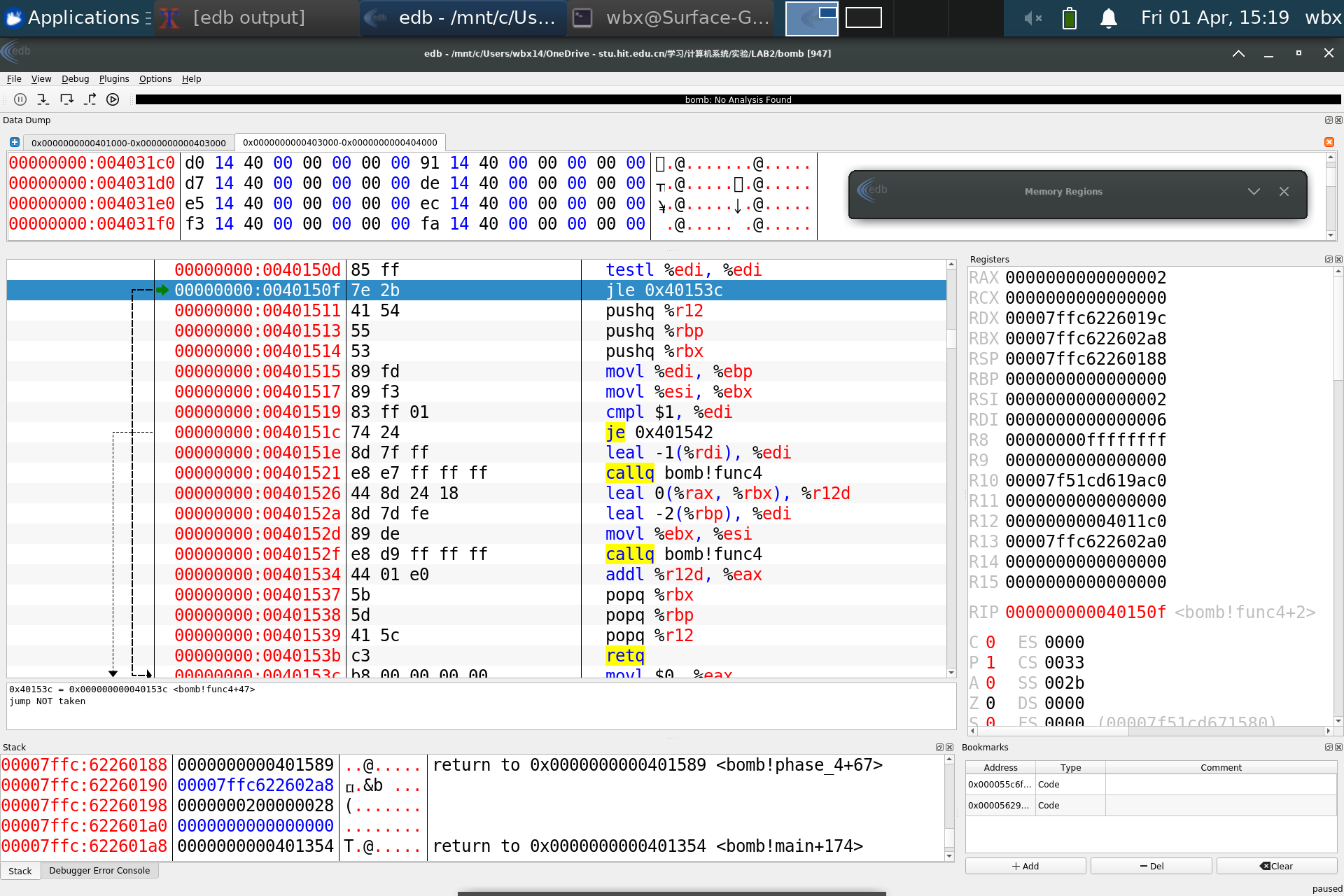


## 3.4 阶段4的破解与分析

密码如下：40 2（密码不唯一）

破解过程：进入Phase\_4函数，发现”%d %d”传入了scanf函数，即密码是两个数。并且发现第二个数介于2~4之间（造成密码不唯一），否则就会爆炸。紧接着发现func4的递归函数，经过对汇编代码的抽象处理得到了f(a,b)=f(a-1,b)+f(a-2,b)+b，f(1,b)=b，f(0,b)=b的数学表达式模型，其中a为固定值6，通过给出数2（即b）作为第2个参数，得到f(a,b)的结果返回值eax为0x28，此时有je判断数1等于eax，给数1为0x28，即可破解。

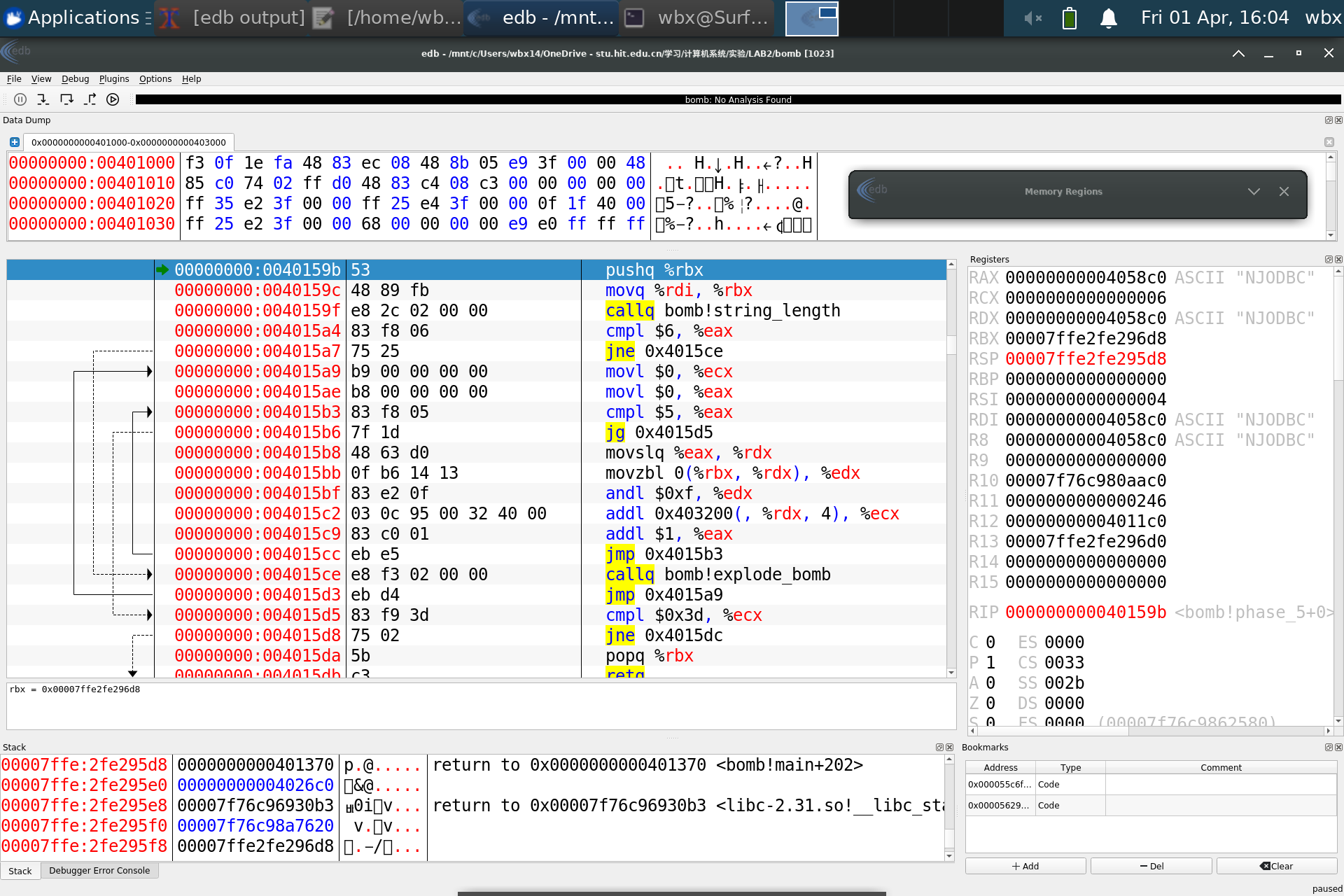


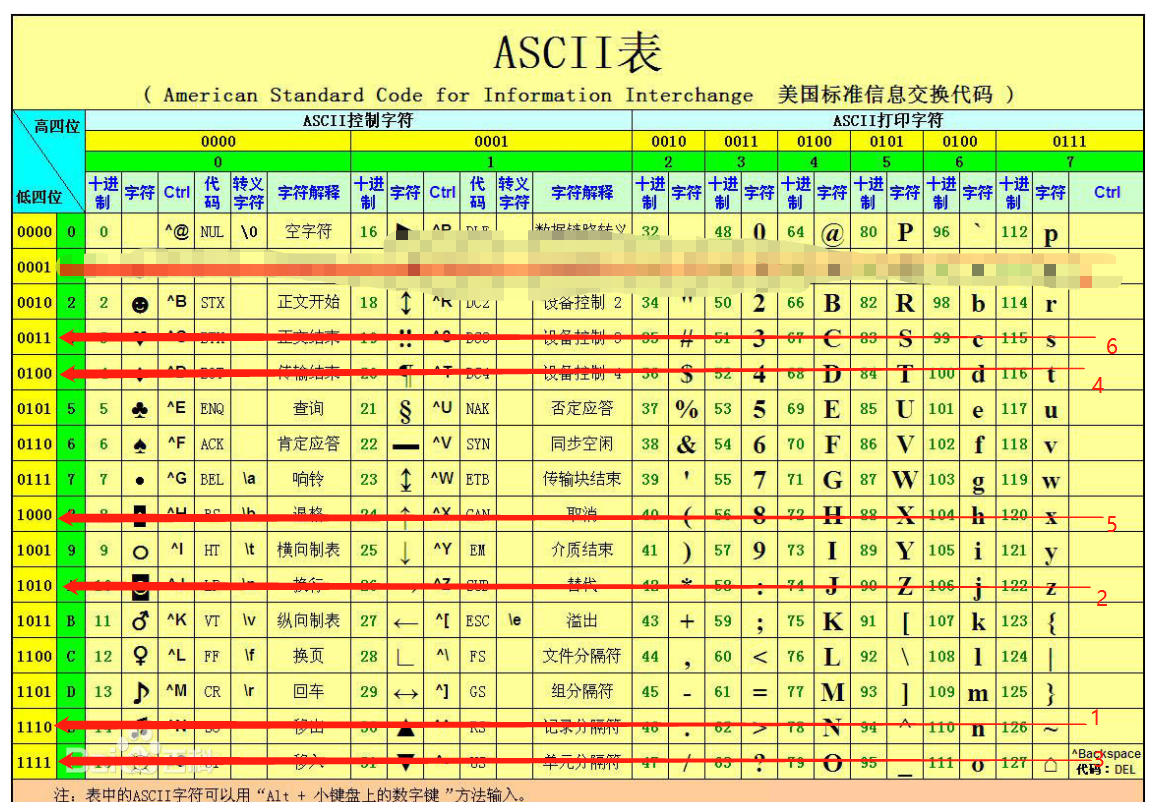


## 3.5 阶段5的破解与分析

密码如下：NJODBC（密码不唯一）

破解过程：进入Pharse\_5函数，发现了一个判断读入字符串长度为6的指令，那么密码就一定是6个字符。经过对后续代码的分析，得出最终要让ecx=0x3d的结果，而eax作为循环计数，总共循环6次，初始化ecx=0，读入的字符串放在rbx指定的地址中，rdx为变址计数，指向字符串索引为0-5的字符，edx作为结果，再与0xf相与，保留低4位，再寻址0x403200+rdx\*4，得到值加到ecx中。此时发现，0x403200所在空间为16个32位整数，从地址低位开始依次是：2、A、6、1、C、10、9、3、4、7、E、5、B、8、F、D。总结得到，读入的6个字符的低四位进行这样的一个函数映射（f(0)=2，f(1)=A，f(2)=6，……），得到结果相加最终得到0x3d即可，因此将0x3d分解得到F+E+D+C+6+1得到其对应的低四位为14、10、15、4、2、3。再经过查阅ASCII码表得到低四位对应的可打印字符（造成密码不唯一），得到密码。





## 3.6 阶段6的破解与分析

密码如下：1 5 4 2 3 6



破解过程：经过对汇编代码的逐步分析，抽象成以下等价的C语言代码。实际上就是对一个链表的重新排序，使其值按从大到小排列（给出6个索引值）。

|  |
| --- |
| #include <stdio.h>  typedef struct Table TB;  struct Table {  int val;  int id;  TB \*next;  };  int explode(){  printf("bomb\n");  exit(8);  return 0;  }  int main(){  TB T[6];  int num[] = {0x3cb,0x25a,0x1c6,0x2ec,0x3b0,0x154};  int i=0,j=0;  for(i=0;i<6;i++){  T[i].val = num[i];  T[i].id = i;  if(i<5) T[i].next = &T[i+1];  else T[i].next = NULL;  }  int a[6]={0};  for(i=0;i<6;i++){  scanf("%d",&a[i]);  }  for(i=0;i<6;i++){  if(a[i]>6) explode();  for(j=i+1;j<6;j++){  if(a[i]==a[j]) explode();  }  }  TB \*t,\*nt[6]={0},\*t1;  for(i=0;i<6;i++){  t = &T[0];  for(j=0;j<a[i]-1;j++) t=t->next;  nt[i] = t;  }  for(i=0;i<5;i++){  t = nt[i];  t1 = t->next;  t->next = nt[i+1];  }  t = nt[5];  t->next = NULL;  t = nt[0];  for(i=0;i<5;i++){  if((t->val)<(t->next->val)) explode();  t = t->next;  }  return 0;  } |

## 3.7 阶段7的破解与分析(隐藏阶段)

密码如下：未分析

破解过程：未分析

# 第4章 总结

## 4.1 请总结本次实验的收获

学会并了解了许多C语言语句在汇编语言中的形式。

学会使用EDB\GDB等调试工具对可执行文件进行调试或分析。

了解一些逆向工程的概念与知识。

## 4.2 请给出对本次实验内容的建议

无

注：本章为酌情加分项。

# 参考文献

[1] [ASCII\_百度百科 (baidu.com)](https://baike.baidu.com/item/ASCII/309296?fr=aladdin)