

**实验报告**

**实 验（三）**

题 目 Binary Bomb

二进制炸弹

专 业 计算机类

学　　 号 1180300303

班　　 级 1836101

学 生 宿梓航

指 导 教 师 刘宏伟

实 验 地 点 G712

实 验 日 期 2019.10.24

**计算机科学与技术学院**

**目 录**

[第1章 实验基本信息 - 3 -](#_Toc496608242)

[1.1 实验目的 - 3 -](#_Toc496608243)

[1.2 实验环境与工具 - 3 -](#_Toc496608244)

[1.2.1 硬件环境 **错误!未定义书签。**](#_Toc496608245)

[1.2.2 软件环境 **错误!未定义书签。**](#_Toc496608246)

[1.2.3 开发工具 **错误!未定义书签。**](#_Toc496608247)

[1.3 实验预习 - 3 -](#_Toc496608248)

[第2章 实验环境建立 - 4 -](#_Toc496608249)

[2.1 Ubuntu下CodeBlocks反汇编（10分） - 4 -](#_Toc496608250)

[2.2 Ubuntu下EDB运行环境建立（10分） - 4 -](#_Toc496608251)

[第3章 各阶段炸弹破解与分析 - 6 -](#_Toc496608252)

[3.1 阶段1的破解与分析 - 6 -](#_Toc496608253)

[3.2 阶段2的破解与分析 - 6 -](#_Toc496608254)

[3.3 阶段3的破解与分析 - 6 -](#_Toc496608255)

[3.4 阶段4的破解与分析 - 7 -](#_Toc496608256)

[3.5 阶段5的破解与分析 - 7 -](#_Toc496608257)

[3.6 阶段6的破解与分析 - 7 -](#_Toc496608258)

[3.7 阶段7的破解与分析(隐藏阶段) - 8 -](#_Toc496608259)

[第4章 总结 - 10 -](#_Toc496608260)

[4.1 请总结本次实验的收获 - 10 -](#_Toc496608261)

[4.2 请给出对本次实验内容的建议 - 10 -](#_Toc496608262)

[参考文献 - 11 -](#_Toc496608263)

# 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

Intel Core i7-8750H, 2.2GHz

16G RAM

Samsung SSD 970 PRO 1TB

### 1.2.2 软件环境

Windows 10 1903 x64

Ubuntu 19.04

Vmware Workstation 15.1

### 1.2.3 开发工具

Visual Studio 2019

GCC 8.3.0，8.1.0

GNU nano 3.2

## 1.3 实验预习

填写

# 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。

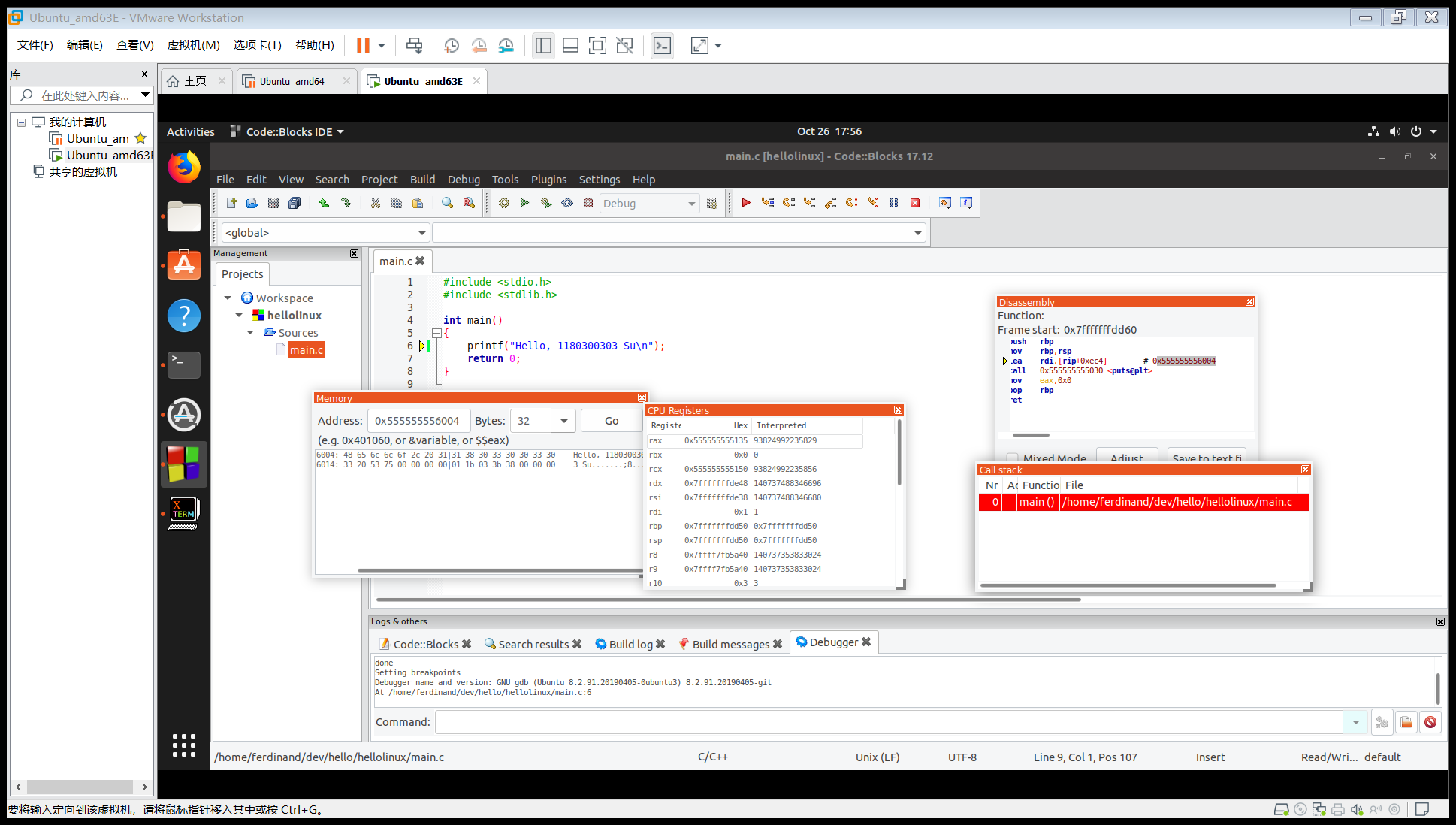
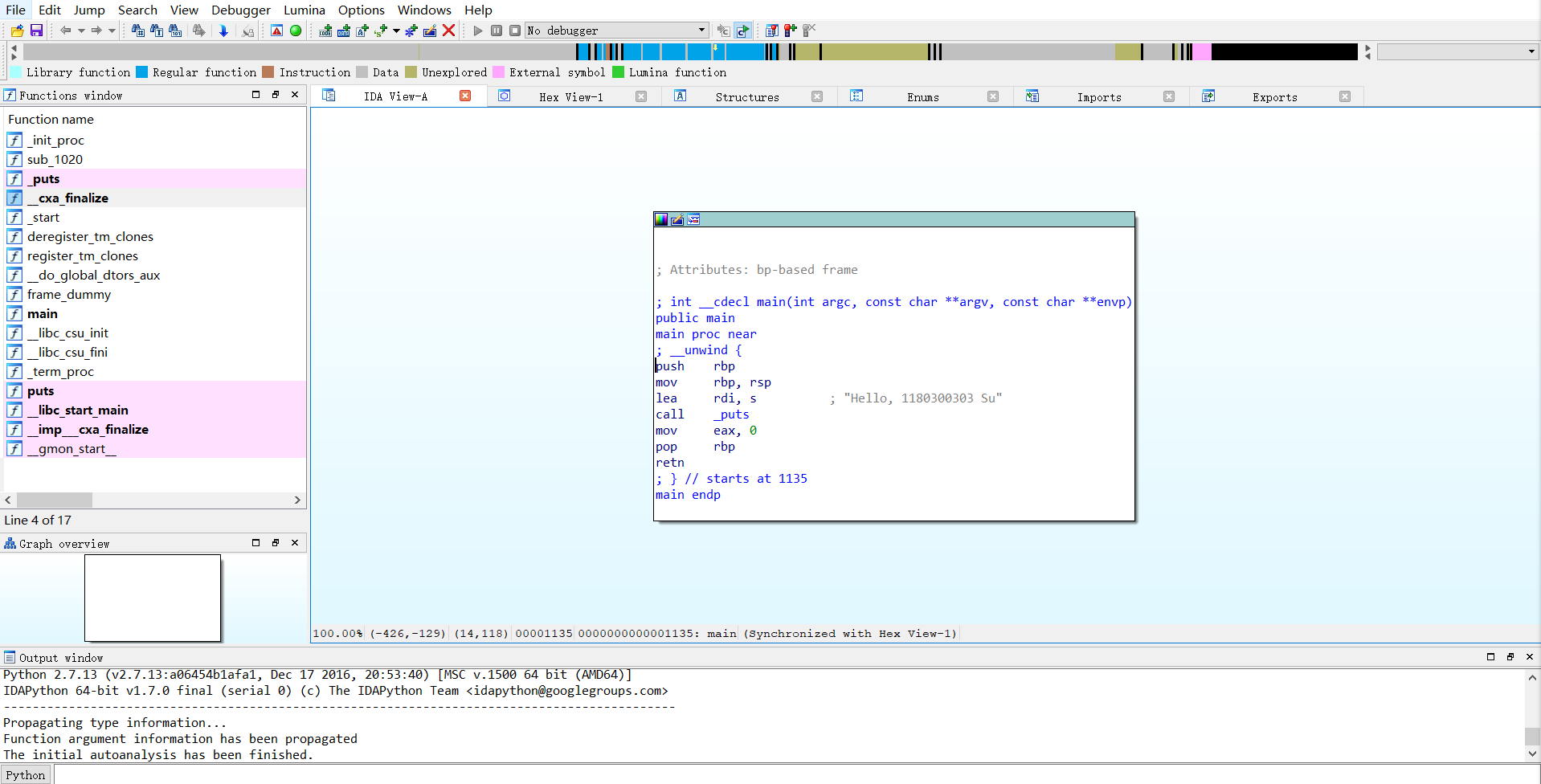


图2-1 Ubuntu下CodeBlocks反汇编截图

## 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



注：因为编译EDB所需的依赖始终安装不正常，且个人认为EDB的实用性不如IDA Pro，故此处用IDA Pro替代进行调试。

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

密码如下：For NASA, space is still a high priorit.

破解过程：

1. 根据’ call phase\_1’找到phase\_1函数，进而读取其中的内容，发现其为一个字符串比较函数，可知密码。

## 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：

1. ’ call phase\_2’找到phase\_2函数，

2. 发现其申请了一段内存（栈），然后调用read\_six\_numbers函数；通过分析调用前的汇编知，其传递了input和刚刚申请的的内存地址两个参数

3. 分析read\_six\_numbers函数，知其使用scanf读取6个数字到刚才的栈中；

4.分析phase\_2，知其要求的数字序列为1 2 4 8 16 32

## 3.3 阶段3的破解与分析

密码如下：5 -656

破解过程：

1. ’ call phase\_3’找到phase\_3函数，

2. 发现其声明2个变量，然后调用scanf函数；通过分析调用前的汇编知，其传递了刚刚申请的的内存地址（两个参数）

3. 分析知，此时程序读入了两个数字(index，sum)。之后是一个跳转表结构；根据最后的判断知，要求0<=index<=5且sum=运算过后的%rax

4.计算知5 -656符合条件。

## 3.4 阶段4的破解与分析

密码如下：3 10

破解过程：

1. ’ call phase\_4’找到phase\_4函数，

2. 发现其申请了两段栈，然后调用scanf函数；通过分析调用前的汇编知，其传递了刚刚申请的的内存地址（两个参数）

3. 分析知，此时程序读入了两个整数(user\_val，user\_sum)。根据条件有user\_sum=10, 0<=user\_val<=14

4.user\_val被带入func4进行递归计算，通过穷举0-14，知3符合条件。

## 3.5 阶段5的破解与分析

密码如下：)))))"

破解过程：

1. ’ call phase\_5’找到phase\_5函数，

2. 发现其利用读入的ASCII码%16作为下标来寻址

3. 分析被寻址的数组知，为了满足加和得41的需求，只需要5个9+16k和1个2+16k，取k=2得)))))"

## 3.6 阶段6的破解与分析

密码如下：5 3 1 4 2 6

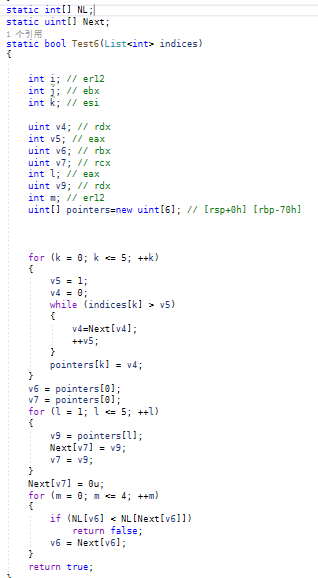
破解过程：

1. ’ call phase\_6’找到phase\_6函数，

2. 发现其和阶段2一样调用了read\_six\_numbers函数。

3. 分析知，输入的要求是整数1-6一个全排列，然后根据这个全排列对一个一维六元链表进行操作

4. 按照程序逻辑编写程序(C#)，穷举得到5 3 1 4 2 6

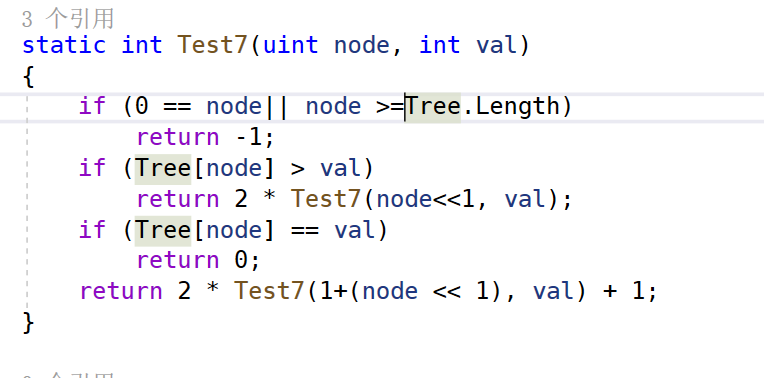


## 3.7 阶段7的破解与分析(隐藏阶段)

密码如下：99(猜测)

破解过程：

1. 搜索反汇编文件中搜索phase找到’secret\_phase’，猜测为隐藏阶段
2. 读反汇编，大概获取程序意图，编写模拟程序获得结果。



# 第4章 总结

## 4.1 请总结本次实验的收获

学会了通过反汇编来调试和解读程序。

## 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.