

**实验一：拆炸弹实验**

山东大学网络空间安全学院计算机组成与设计 实验报告

2022 年 10 月 12 日

**1.实验任务：**

调试环境并完成linux虚拟机QEMU环境下的经典实验“拆炸弹”。程序共计5个关卡，每一关需要用户在终端上输入特定的字符或者数字才能通关，否则会引爆炸弹。每一关的答案藏在对应关卡函数的汇编代码中，所以需要将二进制可执行文件 lab1 反汇编，生成汇编代码， 通过分析汇编代码来找到每一关需要输入的正确内容

# 2.实验过程：

首先，我们分析反编译出的main函数，以第一关为例：

10176: 47f000ef jal ra,10df4 <printf>

1017a: fe843503 ld a0,-24(s0)

1017e: 0c0000ef jal ra,1023e <read\_line>

10182: fe843503 ld a0,-24(s0)

10186: 140000ef jal ra,102c6 <phase\_1>

1018a: 000247b7 lui a5,0x24

1018e: 79078513 addi a0,a5,1936 # 24790 <\_\_clzdi2+0x6a>

10192: 463000ef jal ra,10df4 <printf>

通过分析该部分代码可以得出，每一关都是先输出指令，然后读取n个输入的字符串或数字，最后将输入的字符串或数字带入相应的关卡函数中（此处为phase\_1）。下面我们将具体分析各关代码的具体作用。

**第一关**

00000000000102c6 <phase\_1>:

102c6: 7179 addi sp,sp,-48 // 开辟48的空间，sp-=48

102c8: f406 sd ra,40(sp) //将ra存于sp（40）

102ca: f022 sd s0,32(sp) //将s0存于sp（32）

102cc: 1800 addi s0,sp,48 //s0的值=栈顶+48【栈尾】

102ce: fca43c23 sd a0,-40(s0) //把a0存在s0（-40）

102d2: fe840793 addi a5,s0,-24 //a5=s0-24 存一个地址

102d6: 863e mv a2,a5 //把a5付给a2

102d8: 000257b7 lui a5,0x25 //把十六进制25付给a5

102dc: 8a878593 addi a1,a5,-1880 # 248a8 <\_\_clzdi2+0x182>

102e0: fd843503 ld a0,-40(s0) //把s0（-40）付给a0

102e4: 483000ef jal ra,10f66 <sscanf> //跳转到读取函数，此函数将读入的数字数量付给a0，此时是1（返回读入数字个数）

102e8: 87aa mv a5,a0 //a5=a0=1

102ea: fef42623 sw a5,-20(s0)

102ee: fec42783 lw a5,-20(s0)

102f2: 0007871b sext.w a4,a5 // a4=a5=1；

102f6: 4785 li a5,1 //设a5=1

102f8: 00f70463 beq a4,a5,10300 <phase\_1+0x3a> //判断a5==a4=1

102fc: f29ff0ef jal ra,10224 <explode\_bomb> //爆炸

10300: fe842783 lw a5,-24(s0) //取出是s0（-24）付给a5，这是读入数字本体；（a2）

10304: 873e mv a4,a5 //把a5付给a4

10306: 3e400793 li a5,996 //设a5=996

1030a: 00f70463 beq a4,a5,10312 <phase\_1+0x4c> //判断是否相等，相等则第一关过。

1030e: f17ff0ef jal ra,10224 <explode\_bomb>

10312: 000257b7 lui a5,0x25

10316: 8b078513 addi a0,a5,-1872 # 248b0 <\_\_clzdi2+0x18a>

1031e: 0001 nop

10322: 7402 ld s0,32(sp)

10324: 6145 addi sp,sp,48

10326: 8082 ret

**关键点：**

102ce：把a0存在s0（-40）

102e4、102e8、102f2：a4表示输入的数字数目

102f6：a5赋值为1

102f8：判断a4和a5是否相等

1030a:将输入的数字与996相比较，若相等则拆弹成功，若不相等，则炸弹爆炸

流程分析：

102c6--102e4：开辟栈空间并且读入数字

102e8--102f8:判断读入的数字个数是否等于1

102fc--10326:判断数字是否相等,若相等则拆弹成功，若不相等，则炸弹爆炸

**第二关**

0000000000010328 <phase\_2>:

10328: 7179 addi sp,sp,-48 // 开辟48的空间，sp-=48

1032a: f406 sd ra,40(sp) //将ra存于sp（40）

1032c: f022 sd s0,32(sp) //将s0存于sp（32）

1032e: 1800 addi s0,sp,48 //s0的值=栈顶+48

10330: fca43c23 sd a0,-40(s0) //把a0存在s0（-40）

10334: fe440713 addi a4,s0,-28 //a4=s0-28,存一个地址

10338: fe840793 addi a5,s0,-24 //a5=s0-24,存一个地址

1033c: 86ba mv a3,a4 //把a4付给a3

1033e: 863e mv a2,a5 //把a5付给a2

10340: 000257b7 lui a5,0x25 //把十六进制25付给a5

10344: 8d878593 addi a1,a5,-1832 # 248d8 <\_\_clzdi2+0x1b2>

10348: fd843503 ld a0,-40(s0) //把a0存在s0（-40）

1034c: 41b000ef jal ra,10f66 <sscanf> //跳转到读取函数，此函数将读入的数字数量付给a0，此时是2（返回读入数字个数）

10350: 87aa mv a5,a0 //a5=a0=2

10352: fef42623 sw a5,-20(s0)

10356: fec42783 lw a5,-20(s0)

1035a: 0007871b sext.w a4,a5 //a4=a5=2

1035e: 4789 li a5,2 //设a5=2

10360: 00f70463 beq a4,a5,10368 <phase\_2+0x40>

// 与第一关同理，判断a4和a5是否相等

10364: ec1ff0ef jal ra,10224 <explode\_bomb>

10368: fe842703 lw a4,-24(s0) //与第一关同理，取出存下数字

1036c: fe442783 lw a5,-28(s0) //与第一关同理，取出存下的数字

10370: 9fb9 addw a5,a5,a4

10372: 2781 sext.w a5,a5

10374: 873e mv a4,a5

10376: 3e400793 li a5,996 //进过操作，a4为输入两数字之和

1037a: 00f70463 beq a4,a5,10382 <phase\_2+0x5a> //判断输入两数字是否和为996,若是，则第二关通过

1037e: ea7ff0ef jal ra,10224 <explode\_bomb> //爆炸

10382: 000257b7 lui a5,0x25 //把十六进制付给a5

10386: 8e078513 addi a0,a5,-1824 # 248e0 <\_\_clzdi2+0x1ba>

1038a: 317000ef jal ra,10ea0 <puts>

1038e: 0001 nop

10390: 70a2 ld ra,40(sp)

10392: 7402 ld s0,32(sp)

10394: 6145 addi sp,sp,48

10396: 8082 ret

**关键点：**

1032a：把a0存在s0（-40）

1034c、10350、1035a：a4表示输入的数字数目

1035e：a5赋值为2

102f8：判断a4和a5是否相等

10360:判断a4和a5是否相等

10376:进过操作，a4为输入两数字之和

1037a:判断输入两数字是否和为996

1037a:判断输入两数字是否和为996,若是，则第二关通过，若不相等，则炸弹爆炸

**流程分析：**

10328--1034c：开辟栈空间并且读入数字

10350--1037a:判断读入的数字个数是否等于2

1037e--10396:判断输入两数字是否和为996,若是，则第二关通过，若不相等，则炸弹爆炸

**第三关**

下面分析第三关的内容：

输入两个数，第一个数为第二个数的二倍。

0000000000010398 <phase\_3>:

10398: 7179 addi sp,sp,-48//sp=sp-48指针开辟空间

1039a: f406 sd ra,40(sp)

1039c: f022 sd s0,32(sp)

1039e: 1800 addi s0,sp,48//s0可以表示原来指针sp的位置

103a0: fca43c23 sd a0,-40(s0)

103a4: fe440713 addi a4,s0,-28//第一个输入存储在s0-28

103a8: fe840793 addi a5,s0,-24//第二个输入存储在s0-24

103ac: 86ba mv a3,a4//存储输入数字的地址

103ae: 863e mv a2,a5//存储输入数字的地址

103b0: 000257b7 lui a5,0x25

103b4: 8d878593 addi a1,a5,-1832 # 248d8 <\_\_clzdi2+0x1b2>

103b8: fd843503 ld a0,-40(s0)//输入数字的个数

103bc: 3ab000ef jal ra,10f66 <sscanf>//sscanf函数

103c0: 87aa mv a5,a0

103c2: fef42623 sw a5,-20(s0)

103c6: fec42783 lw a5,-20(s0)

103ca: 0007871b sext.w a4,a5//a4=a5

103ce: 4789 li a5,2//a5=2

103d0: 00f70463 beq a4,a5,103d8 <phase\_3+0x40>//判断输入是否是两个数字

103d4: e51ff0ef jal ra,10224 <explode\_bomb>

103d8: fe842783 lw a5,-24(s0)//存储第一个输入的数字

103dc: 0027979b slliw a5,a5,0x2//执行a5乘4操作

103e0: 0007871b sext.w a4,a5

103e4: fe442783 lw a5,-28(s0)//存储第二个输入的数字

103e8: 00f70463 beq a4,a5,103f0 <phase\_3+0x58>//判断第二个数字是否是第一个数字的四倍

103ec: e39ff0ef jal ra,10224 <explode\_bomb>

103f0: 000257b7 lui a5,0x25

103f4: 90878513 addi a0,a5,-1784 # 24908 <\_\_clzdi2+0x1e2>

103f8: 2a9000ef jal ra,10ea0 <puts>

103fc: 0001 nop

103fe: 70a2 ld ra,40(sp)

10400: 7402 ld s0,32(sp)

10402: 6145 addi sp,sp,48

10404: 8082 ret

**关键点：**

103e8可知：比较a4和a5是否相等

103e4可知：a5取自-28(s0)

103e0和103dc和103d8可知：a4是二倍的存储在-24(s0)的数值

103d0可知：比较a4和a5是否相等

103ce可知：a5赋值为2

103b8和103c0和103ca结合scanf可知：a4表示为输入的数字数目

**解题流程：**

10398-103bc：开辟栈空间并且读入数字

103c0-103d0：判断读入的数字个数是否等于2

103d8-103e8：执行第一个数字四倍操作并且进行判断数字是否相等

**第四关**

主要内容为输入六个数，它们是一个公比为2的等比数列，以下是结合代码分析。

0000000000010406 <phase\_4>:

10406: 715d addi sp,sp,-80//开辟一个80的空间

10408: e486 sd ra,72(sp)

1040a: e0a2 sd s0,64(sp)

1040c: 0880 addi s0,sp,80

1040e: faa43c23 sd a0,-72(s0)

10412: fc040793 addi a5,s0,-64

10416: 85be mv a1,a5//a1=s0-64,为接下来的read\_six\_numbers做准备

10418: fb843503 ld a0,-72(s0)

1041c: 0e0000ef jal ra,104fc <read\_six\_numbers>//读入六个数字，s0-64到s0-44之间

10420: fc040793 addi a5,s0,-64

10424: fef43423 sd a5,-24(s0)

10428: fc040793 addi a5,s0,-64

1042c: 07d1 addi a5,a5,20

1042e: fef43023 sd a5,-32(s0)

10432: a805 j 10462 <phase\_4+0x5c>//执行循环

10434: fe843783 ld a5,-24(s0)

10438: 439c lw a5,0(a5)//获得输入的第i个数字

1043a: 0017979b slliw a5,a5,0x1//执行乘二的操作

1043e: fcf42e23 sw a5,-36(s0)

10442: fe843783 ld a5,-24(s0)

10446: 0791 addi a5,a5,4

10448: 4398 lw a4,0(a5)//获得第i+1个数字

1044a: fdc42783 lw a5,-36(s0)

1044e: 2781 sext.w a5,a5

10450: 00e78463 beq a5,a4,10458 <phase\_4+0x52>

//判断第i+1个数字是不是第i个数字的两倍

10454: dd1ff0ef jal ra,10224 <explode\_bomb>

10458: fe843783 ld a5,-24(s0)

1045c: 0791 addi a5,a5,4

1045e: fef43423 sd a5,-24(s0)

10462: fe843703 ld a4,-24(s0)

10466: fe043783 ld a5,-32(s0)

1046a: fcf765e3 bltu a4,a5,10434 <phase\_4+0x2e>

//判断循环是否完成，未完成则执行下一次循环

1046e: 000257b7 lui a5,0x25

10472: 93078513 addi a0,a5,-1744 # 24930 <\_\_clzdi2+0x20a>

10476: 22b000ef jal ra,10ea0 <puts>

1047a: 0001 nop

1047c: 60a6 ld ra,72(sp)

1047e: 6406 ld s0,64(sp)

10480: 6161 addi sp,sp,80

10482: 8082 ret

下面分析第四关中的<read\_six\_numbers>函数，分析该函数可知道，该函数是读入六个字符串，每个字符串地址相差四个，读写完成后判断是否读入六个数字。

104fc: 7179 addi sp,sp,-48

104fe: f406 sd ra,40(sp)

10500: f022 sd s0,32(sp)

10502: 1800 addi s0,sp,48

10504: fca43c23 sd a0,-40(s0)

10508: fcb43823 sd a1,-48(s0)

1050c: fd043783 ld a5,-48(s0)

10510: 00478693 addi a3,a5,4

10514: fd043783 ld a5,-48(s0)

10518: 00878713 addi a4,a5,8

1051c: fd043783 ld a5,-48(s0)

10520: 00c78613 addi a2,a5,12

10524: fd043783 ld a5,-48(s0)

10528: 01078593 addi a1,a5,16

1052c: fd043783 ld a5,-48(s0)

10530: 07d1 addi a5,a5,20

10532: 88be mv a7,a5

10534: 882e mv a6,a1

10536: 87b2 mv a5,a2

10538: fd043603 ld a2,-48(s0)

1053c: 000255b7 lui a1,0x25

10540: 98058593 addi a1,a1,-1664 # 24980 <\_\_clzdi2+0x25a>

10544: fd843503 ld a0,-40(s0)

10548: 21f000ef jal ra,10f66 <sscanf>

1054c: 87aa mv a5,a0

1054e: fef42623 sw a5,-20(s0)

10552: fec42783 lw a5,-20(s0)

10556: 0007871b sext.w a4,a5

1055a: 4799 li a5,6

1055c: 00f70463 beq a4,a5,10564 <read\_six\_numbers+0x68>

10560: cc5ff0ef jal ra,10224 <explode\_bomb>

10564: 0001 nop

10566: 70a2 ld ra,40(sp)

10568: 7402 ld s0,32(sp)

1056a: 6145 addi sp,sp,48

1056c: 8082 ret

**关键点**：

10416可知：将读入的六个数字存入开辟的栈中

10438和1043a和10448和10450可知：判断第i+1个数字是否是第i个数字的两倍

10432和1046a可知：进行循环，不断读取输入的数字。

**解题流程：**

10406-1041c：开辟空间并完成读入六个数字操作

10434-10450：判断读入的前后两个数字是否是二倍关系

10458-1046a：判断循环是否结束，如果未结束进行下一次循环

**第五关**

下面我们将分析第五关的解题流程，特别的，我们还需分析递归函数func4。下面是结合代码的分析。

0000000000010484 <phase\_5>:

10484: 7179 addi sp,sp,-48

10486: f406 sd ra,40(sp)

10488: f022 sd s0,32(sp)

1048a: 1800 addi s0,sp,48

1048c: fca43c23 sd a0,-40(s0)

10490: fe440713 addi a4,s0,-28

10494: fe840793 addi a5,s0,-24

10498: 86ba mv a3,a4

1049a: 863e mv a2,a5

1049c: 000257b7 lui a5,0x25

104a0: 8d878593 addi a1,a5,-1832 # 248d8 <\_\_clzdi2+0x1b2>

104a4: fd843503 ld a0,-40(s0)

104a8: 2bf000ef jal ra,10f66 <sscanf>

104ac: 87aa mv a5,a0

104ae: fef42623 sw a5,-20(s0)

104b2: fec42783 lw a5,-20(s0)

104b6: 0007871b sext.w a4,a5

104ba: 4789 li a5,2 //与上面前三关相同，判断读取两个数字

104bc: 00f70463 beq a4,a5,104c4 <phase\_5+0x40>

104c0: d65ff0ef jal ra,10224 <explode\_bomb>

104c4: fe842783 lw a5,-24(s0) //a5取出s0（-24）

104c8: 853e mv a0,a5 //a0=a5

104ca: 0a4000ef jal ra,1056e <func4> //运行函数4，获得阶乘

104ce: 87aa mv a5,a0

104d0: fef42623 sw a5,-20(s0) //把a5存在s0（-20）

104d4: fe442703 lw a4,-28(s0) //把a4取出s0（-28）

104d8: fec42783 lw a5,-20(s0) //把a5取出s0（-20）

104dc: 2781 sext.w a5,a5

104de: 00e78463 beq a5,a4,104e6 <phase\_5+0x62> //判断函数4得到的阶乘，与自行输入的阶乘比较

104e2: d43ff0ef jal ra,10224 <explode\_bomb>

104e6: 000257b7 lui a5,0x25

104ea: 95078513 addi a0,a5,-1712 # 24950 <\_\_clzdi2+0x22a>

104ee: 1b3000ef jal ra,10ea0 <puts>

104f2: 0001 nop

104f4: 70a2 ld ra,40(sp)

104f6: 7402 ld s0,32(sp)

104f8: 6145 addi sp,sp,48

104fa: 8082 ret

**关键点：**

104ba：判断是否读入了两个数字

104ca：进入递归函数，获得阶乘

104de：判断第二个数是否为第一个数的阶乘

**解题过程**：

结合对func4的分析（具体分析在第五关分析之后），第六关的关卡函数分为判断输入和判断阶乘两步：

首先，验证输入的内容是否为六个数字，若是则进入第二步，若不是则炸弹爆炸。

其次，将输入的第一个数字的阶乘与输入的第二个数字相比较，若相等则拆弹成功，若出现不相等则炸弹爆炸

所以推得第5关的密码为：输入整数n和n！。

下面单独分析递归函数func4的过程：

000000000001056e <func4>:

1056e: 1101 addi sp,sp,-32

10570: ec06 sd ra,24(sp)

10572: e822 sd s0,16(sp)

10574: 1000 addi s0,sp,32

10576: 87aa mv a5,a0 //把n付给a5

10578: fef42623 sw a5,-20(s0) //把n存储在-20这个位置

1057c: fec42783 lw a5,-20(s0)

10580: 2781 sext.w a5,a5

10582: e399 bnez a5,10588 <func4+0x1a> //如果a5>0，调转到10588 ，否则向下

10584: 4785 li a5,1 //当a5=0，把它附1.

10586: a839 j 105a4 <func4+0x36> //上一句完成后转到105a4

10588: fec42783 lw a5,-20(s0)

1058c: 37fd addiw a5,a5,-1 //a5--并存储

1058e: 2781 sext.w a5,a5

10590: 853e mv a0,a5

10592: fddff0ef jal ra,1056e <func4> //a5--后继续调用自身（继续--）

10596: 87aa mv a5,a0 //这里是ret回溯的开始点

10598: 873e mv a4,a5

1059a: fec42783 lw a5,-20(s0)

1059e: 02e787bb mulw a5,a5,a4 //累乘

105a2: 2781 sext.w a5,a5

105a4: 853e mv a0,a5 //以下是存储操作

105a6: 60e2 ld ra,24(sp)

105a8: 6442 ld s0,16(sp)

105aa: 6105 addi sp,sp,32

105ac: 8082 ret //递归回溯

**关键点：**

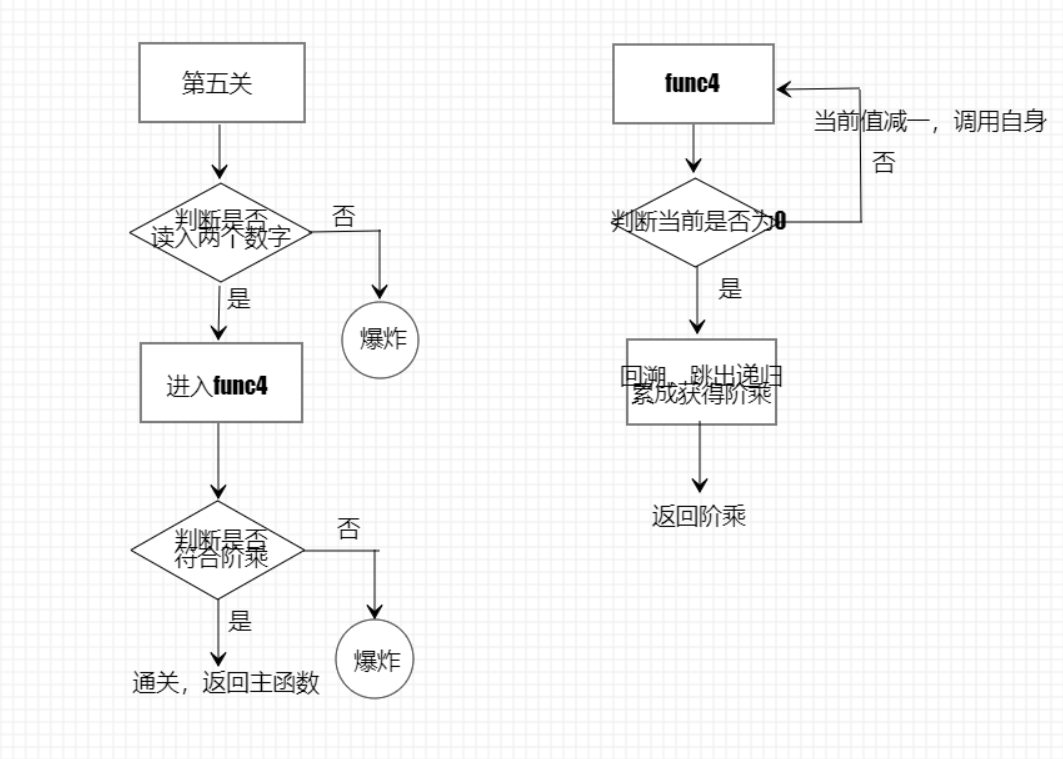
10592：调用func4自身实现递归

10582：判断回溯条件（a5=0）

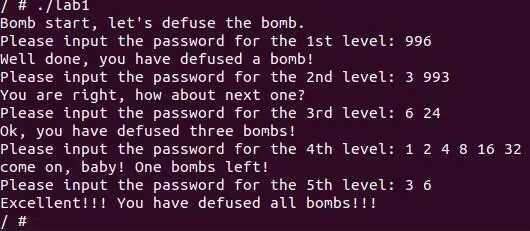
**解题过程：**

通过分析可知，func4函数通过递归实现了对输入的值求阶乘的功能。

其中 10582处对a5的判0是循环的回溯条件，10592处调用自身获得f（n-1）的递归。通过构造的递归条件，func4计算了输入值的阶乘。具体流程为：



最后将各关答案带入实验中，成功拆除炸弹：



# 3.实验总结：

通过本次实验，我们了解和熟悉了Linux系统，学会了基础的命令行操作，并学会了如何生成反编汇代码，练习了risc-v语言的代码阅读。

同时，通过本次实验，我们学习了许多课上未涉及的risc-v指令，通过翻阅参考书和网络检索了解了它们的具体功能，从而加深了对rics-v的理解，也锻炼了小组合作的能力。

# 4.小组分工：

唐诗：代码阅读分析；第3，4关代码解析和总结。

于洁：代码阅读分析；main函数与第1，2关代码解析和总结。

李昕：代码阅读分析；第5关及相关函数解析与总结；实验报告撰写。