

**实验报告**

**实 验（五）**

题 目 LinkLab

链接

专 业 计算机类

学　　 号 1190200523

班　　 级 1903002

学 生 石翔宇

指 导 教 师 郑贵滨

实 验 地 点 G709

实 验 日 期 2021.5.21

**计算机科学与技术学院**

**目 录**

[第1章 实验基本信息 - 3 -](#_Toc72677853)

[1.1 实验目的 - 3 -](#_Toc72677854)

[1.2 实验环境与工具 - 3 -](#_Toc72677855)

[1.2.1 硬件环境 - 3 -](#_Toc72677856)

[1.2.2 软件环境 - 3 -](#_Toc72677857)

[1.2.3 开发工具 - 3 -](#_Toc72677858)

[1.3 实验预习 - 3 -](#_Toc72677859)

[第2章 实验预习 - 5 -](#_Toc72677860)

[2.1 ELF文件格式解读 - 5 -](#_Toc72677861)

[2.2程序的内存映像结构 - 5 -](#_Toc72677862)

[2.3程序中符号的位置分析 - 6 -](#_Toc72677863)

[2.4程序运行过程分析 - 8 -](#_Toc72677864)

[第3章 各阶段的原理与方法 - 9 -](#_Toc72677865)

[3.1 阶段1的分析 - 9 -](#_Toc72677866)

[3.2 阶段2的分析 - 9 -](#_Toc72677867)

[3.3 阶段3的分析 - 11 -](#_Toc72677868)

[3.4 阶段4的分析 - 12 -](#_Toc72677869)

[3.5 阶段5的分析 - 14 -](#_Toc72677870)

[第4章 总结 - 15 -](#_Toc72677871)

[4.1 请总结本次实验的收获 - 15 -](#_Toc72677872)

[4.2 请给出对本次实验内容的建议 - 15 -](#_Toc72677873)

[参考文献 - 16 -](#_Toc72677874)

# 第1章 实验基本信息

## 1.1 实验目的

* 理解链接的作用与工作步骤
* 掌握ELF结构、符号解析与重定位的工作过程
* 熟练使用Linux工具完成ELF分析与修改

## 1.2 实验环境与工具

### 1.2.1 硬件环境

* Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
* 16GB RAM
* 1TB HDD + 512G SSD

### 1.2.2 软件环境

* Windows 10 21H1
* Ubuntu 20.04 LTS

### 1.2.3 开发工具

* VSCode，CodeBlocks，gcc+gdb

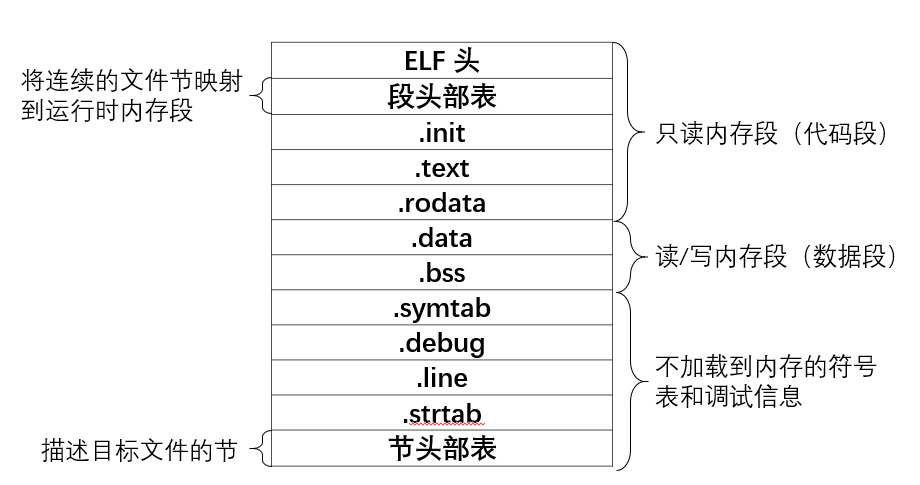
## 1.3 实验预习

* 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。
* 请按顺序写出ELF格式的可执行目标文件的各类信息。
* 请按照内存地址从低到高的顺序，写出Linux下X64内存映像。
* 请运行“LinkAddress -u 学号 姓名” 按地址顺序写出各符号的地址、空间。并按照Linux下X64内存映像结构，标出其所属各区。
  + gcc -m64 -o LinkAddress linkaddress.c
* 请按顺序写出LinkAddress从开始执行到main前/后执行的子程序的名字。(gcc与objdump/GDB/EDB)

# 第2章 实验预习

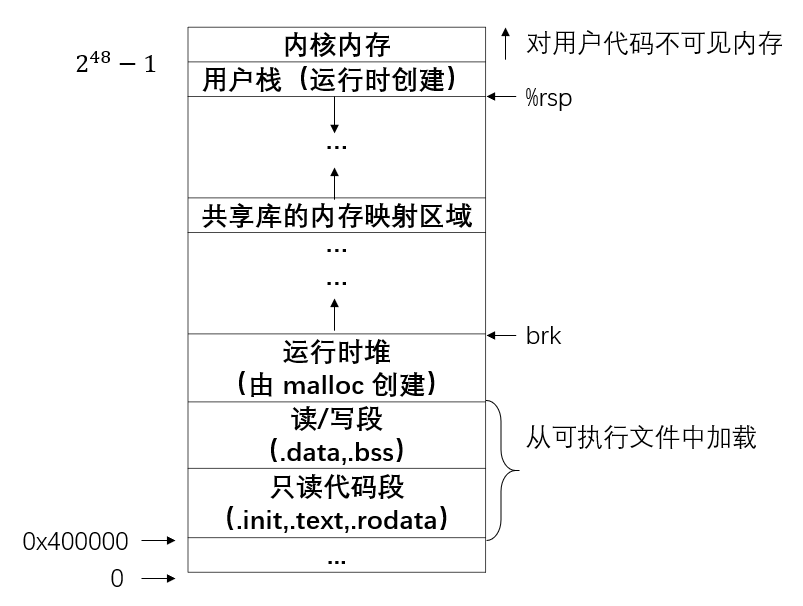
## 2.1 ELF文件格式解读

请按顺序写出ELF格式的可执行目标文件的各类信息（5分）



## 2.2程序的内存映像结构

请按照内存地址从低到高的顺序，写出Linux下X64内存映像（5分）



## 2.3程序中符号的位置分析

请运行“LinkAddress -u 学号 姓名” 按地址顺序写出各符号的地址，并按照Linux下X64内存映像标出其所属内存区段（5分）

|  |  |
| --- | --- |
| 用户栈 | env 0x7ffda06f26f0 140727295092464  env[0] \*env 0x7ffda06f43c4 140727295099844  SHELL=/bin/bash  env[1] \*env 0x7ffda06f43d4 140727295099860  SESSION\_MANAGER=local/ubuntu:@/tmp/.ICE-unix/1664,unix/ubuntu:/tmp/.ICE-unix/1664  env[2] \*env 0x7ffda06f4426 140727295099942  QT\_ACCESSIBILITY=1  env[3] \*env 0x7ffda06f4439 140727295099961  COLORTERM=truecolor  env[4] \*env 0x7ffda06f444d 140727295099981  XDG\_CONFIG\_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg  env[5] \*env 0x7ffda06f447a 140727295100026  XDG\_MENU\_PREFIX=gnome-  env[6] \*env 0x7ffda06f4491 140727295100049  GNOME\_DESKTOP\_SESSION\_ID=this-is-deprecated  env[7] \*env 0x7ffda06f44bd 140727295100093  GNOME\_SHELL\_SESSION\_MODE=ubuntu  env[8] \*env 0x7ffda06f44dd 140727295100125  SSH\_AUTH\_SOCK=/run/user/1000/keyring/ssh  env[9] \*env 0x7ffda06f4506 140727295100166  XMODIFIERS=@im=ibus  env[10] \*env 0x7ffda06f451a 140727295100186  DESKTOP\_SESSION=ubuntu  env[11] \*env 0x7ffda06f4531 140727295100209  SSH\_AGENT\_PID=1628  env[12] \*env 0x7ffda06f4544 140727295100228  GTK\_MODULES=gail:atk-bridge  env[13] \*env 0x7ffda06f4560 140727295100256  DBUS\_STARTER\_BUS\_TYPE=session  env[14] \*env 0x7ffda06f457e 140727295100286  PWD=/home/dedsec/CSAPP/Lab/Lab5  env[15] \*env 0x7ffda06f459e 140727295100318  LOGNAME=dedsec  env[16] \*env 0x7ffda06f45ad 140727295100333  XDG\_SESSION\_DESKTOP=ubuntu  env[17] \*env 0x7ffda06f45c8 140727295100360  XDG\_SESSION\_TYPE=x11  env[18] \*env 0x7ffda06f45dd 140727295100381  GPG\_AGENT\_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1  env[19] \*env 0x7ffda06f4611 140727295100433  XAUTHORITY=/run/user/1000/gdm/Xauthority  env[20] \*env 0x7ffda06f463a 140727295100474  WINDOWPATH=2  env[21] \*env 0x7ffda06f4647 140727295100487  HOME=/home/dedsec  env[22] \*env 0x7ffda06f4659 140727295100505  USERNAME=dedsec  env[23] \*env 0x7ffda06f4669 140727295100521  IM\_CONFIG\_PHASE=1  env[24] \*env 0x7ffda06f467b 140727295100539  LANG=en\_US.UTF-8  env[25] \*env 0x7ffda06f468c 140727295100556  LS\_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:\*.tar=01;31:\*.tgz=01;31:\*.arc=01;31:\*.arj=01;31:\*.taz=01;31:\*.lha=01;31:\*.lz4=01;31:\*.lzh=01;31:\*.lzma=01;31:\*.tlz=01;31:\*.txz=01;31:\*.tzo=01;31:\*.t7z=01;31:\*.zip=01;31:\*.z=01;31:\*.dz=01;31:\*.gz=01;31:\*.lrz=01;31:\*.lz=01;31:\*.lzo=01;31:\*.xz=01;31:\*.zst=01;31:\*.tzst=01;31:\*.bz2=01;31:\*.bz=01;31:\*.tbz=01;31:\*.tbz2=01;31:\*.tz=01;31:\*.deb=01;31:\*.rpm=01;31:\*.jar=01;31:\*.war=01;31:\*.ear=01;31:\*.sar=01;31:\*.rar=01;31:\*.alz=01;31:\*.ace=01;31:\*.zoo=01;31:\*.cpio=01;31:\*.7z=01;31:\*.rz=01;31:\*.cab=01;31:\*.wim=01;31:\*.swm=01;31:\*.dwm=01;31:\*.esd=01;31:\*.jpg=01;35:\*.jpeg=01;35:\*.mjpg=01;35:\*.mjpeg=01;35:\*.gif=01;35:\*.bmp=01;35:\*.pbm=01;35:\*.pgm=01;35:\*.ppm=01;35:\*.tga=01;35:\*.xbm=01;35:\*.xpm=01;35:\*.tif=01;35:\*.tiff=01;35:\*.png=01;35:\*.svg=01;35:\*.svgz=01;35:\*.mng=01;35:\*.pcx=01;35:\*.mov=01;35:\*.mpg=01;35:\*.mpeg=01;35:\*.m2v=01;35:\*.mkv=01;35:\*.webm=01;35:\*.ogm=01;35:\*.mp4=01;35:\*.m4v=01;35:\*.mp4v=01;35:\*.vob=01;35:\*.qt=01;35:\*.nuv=01;35:\*.wmv=01;35:\*.asf=01;35:\*.rm=01;35:\*.rmvb=01;35:\*.flc=01;35:\*.avi=01;35:\*.fli=01;35:\*.flv=01;35:\*.gl=01;35:\*.dl=01;35:\*.xcf=01;35:\*.xwd=01;35:\*.yuv=01;35:\*.cgm=01;35:\*.emf=01;35:\*.ogv=01;35:\*.ogx=01;35:\*.aac=00;36:\*.au=00;36:\*.flac=00;36:\*.m4a=00;36:\*.mid=00;36:\*.midi=00;36:\*.mka=00;36:\*.mp3=00;36:\*.mpc=00;36:\*.ogg=00;36:\*.ra=00;36:\*.wav=00;36:\*.oga=00;36:\*.opus=00;36:\*.spx=00;36:\*.xspf=00;36:  env[26] \*env 0x7ffda06f4c6e 140727295102062  XDG\_CURRENT\_DESKTOP=ubuntu:GNOME  env[27] \*env 0x7ffda06f4c8f 140727295102095  VTE\_VERSION=6003  env[28] \*env 0x7ffda06f4ca0 140727295102112  GNOME\_TERMINAL\_SCREEN=/org/gnome/Terminal/screen/2b10d8c9\_37b1\_4f96\_918e\_d0020abaeb61  env[29] \*env 0x7ffda06f4cf6 140727295102198  INVOCATION\_ID=83bcbcac7e0943bbb5049b2199ace651  env[30] \*env 0x7ffda06f4d25 140727295102245  MANAGERPID=1454  env[31] \*env 0x7ffda06f4d35 140727295102261  LESSCLOSE=/usr/bin/lesspipe %s %s  env[32] \*env 0x7ffda06f4d57 140727295102295  XDG\_SESSION\_CLASS=user  env[33] \*env 0x7ffda06f4d6e 140727295102318  TERM=xterm-256color  env[34] \*env 0x7ffda06f4d82 140727295102338  LESSOPEN=| /usr/bin/lesspipe %s  env[35] \*env 0x7ffda06f4da2 140727295102370  USER=dedsec  env[36] \*env 0x7ffda06f4dae 140727295102382  GNOME\_TERMINAL\_SERVICE=:1.183  env[37] \*env 0x7ffda06f4dcc 140727295102412  DISPLAY=:0  env[38] \*env 0x7ffda06f4dd7 140727295102423  SHLVL=1  env[39] \*env 0x7ffda06f4ddf 140727295102431  QT\_IM\_MODULE=ibus  env[40] \*env 0x7ffda06f4df1 140727295102449  DBUS\_STARTER\_ADDRESS=unix:path=/run/user/1000/bus,guid=ffd78ad7b63b39dfef2dd062607fcd38  env[41] \*env 0x7ffda06f4e49 140727295102537  XDG\_RUNTIME\_DIR=/run/user/1000  env[42] \*env 0x7ffda06f4e68 140727295102568  JOURNAL\_STREAM=8:49163  env[43] \*env 0x7ffda06f4e7f 140727295102591  XDG\_DATA\_DIRS=/usr/share/ubuntu:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop  env[44] \*env 0x7ffda06f4ed4 140727295102676  PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin  env[45] \*env 0x7ffda06f4f3c 140727295102780  GDMSESSION=ubuntu  env[46] \*env 0x7ffda06f4f4e 140727295102798  DBUS\_SESSION\_BUS\_ADDRESS=unix:path=/run/user/1000/bus,guid=ffd78ad7b63b39dfef2dd062607fcd38  env[47] \*env 0x7ffda06f4faa 140727295102890  \_=./LinkAddr  env[48] \*env 0x7ffda06f4fb7 140727295102903  OLDPWD=/home/dedsec/CSAPP/Lab/Lab5/linklab-1190200523  argc 0x7ffda06f258c 140727295092108  argv 0x7ffda06f26c8 140727295092424  argv[0] 7ffda06f43a1  argv[1] 7ffda06f43ac  argv[2] 7ffda06f43af  argv[3] 7ffda06f43ba  argv[0] 0x7ffda06f43a1 140727295099809  ./LinkAddr  argv[1] 0x7ffda06f43ac 140727295099820  -u  argv[2] 0x7ffda06f43af 140727295099823  1190200523  argv[3] 0x7ffda06f43ba 140727295099834  石翔宇  local 0x7ffda06f2590 140727295092112 |
| 共享库的内存映射区域 | exit 0x7f9624964bc0 140282835651520  printf 0x7f962497fe10 140282835762704  malloc 0x7f96249b8260 140282835993184  free 0x7f96249b8850 140282835994704 |
| 运行时堆 | p1 0x7f961491a010 140282566909968  p3 0x7f96148f9010 140282566774800  p4 0x7f95d48f8010 140281493028880  p5 0x7f95548f7010 140279345541136 |
| 读/写段 | big array 0x55f05ff94040 94490890682432  huge array 0x55f01ff94040 94489816940608  global 0x55f01ff9402c 94489816940588  p2 0x55f061b726b0 94490919904944 |
| 只读代码段 | show\_pointer 0x55f01fd9281a 94489814837274  useless 0x55f01fd9284d 94489814837325  main 0x55f01fd92858 94489814837336 |

## 2.4程序运行过程分析

请按顺序写出LinkAddress从开始执行到main前/后执行的子程序的名字(使用gcc与objdump/GDB/EDB)（5分）

|  |  |
| --- | --- |
| 从开始执行到main前： | \_start  \_\_libc\_start\_main  \_\_GI\_\_\_cxa\_atexit  \_\_internal\_atexit  \_\_lll\_cas\_lock  \_\_new\_exitfn  \_\_libc\_csu\_init  \_init  frame\_dummy  register\_tm\_clones  \_\_libc\_csu\_init  \_setjmp  \_\_sigsetjmp  \_\_sigjmp\_save |
| main后执行： | \_\_GI\_exit  \_\_run\_exit\_handlers  \_\_GI\_\_\_call\_tls\_dtors  \_\_lll\_cas\_lock  \_IO\_cleanup  \_IO\_flush\_all\_lockp  IO\_validate\_vtable  \_IO\_unbuffer\_all  \_IO\_new\_file\_setbuf  \_IO\_default\_setbuf  \_IO\_new\_file\_sync  \_\_GI\_\_IO\_setb |

# 第3章 各阶段的原理与方法

每阶段40分，phasex.o 20分，分析20分，总分不超过80分

## 3.1 阶段1的分析

程序运行结果截图：

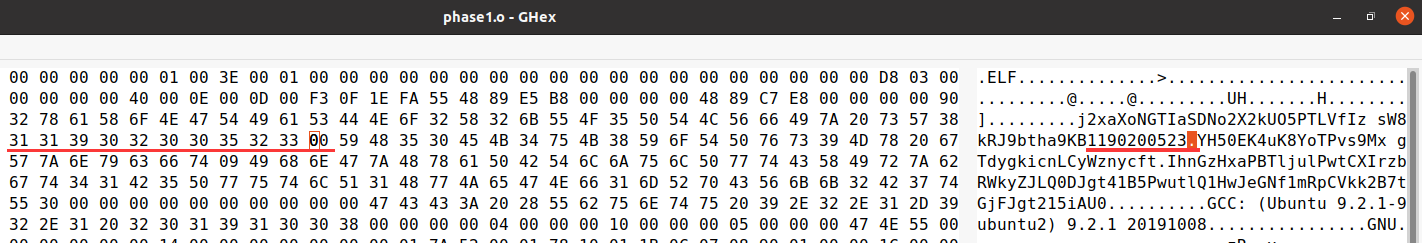


分析与设计的过程：

1. 将原文件编译运行得到结果：



1. 用Linux下的二进制文件编辑器GHex打开phase1.o文件，查找“5Zmtt……”字符串，将其修改为学号对应的ASCII的16进制，并将末尾改为00



1. 修改成功，保存并重新编译后得到预期结果。

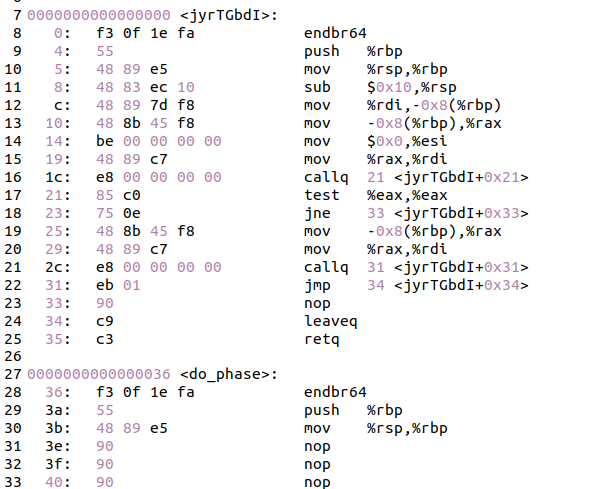
## 3.2 阶段2的分析

程序运行结果截图：

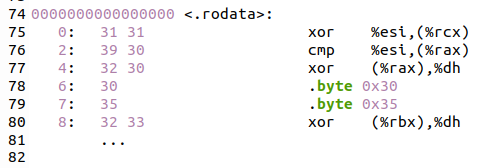


分析与设计的过程：

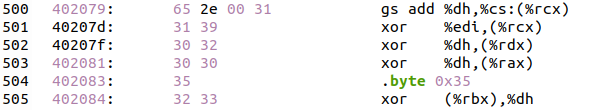
1. 将phase2.o反编译得知，我们需要在do\_phase中替换nop代码，使得能够跳转到jyrTGbdI，并且将学号作为参数传入。



1. 在.rodata段找到了学号，则只需将对应地址的内容当作参数传入函数即可



1. 将linkbomb2反编译得到学号的地址，为0x40207c



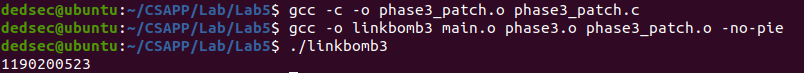
1. 考虑代码mov $0x40207c,%rdi，将其编译并反汇编得到对应的汇编指令48 c7 c7 7c 20 40 00
2. 查找callq PC相对寻址的汇编指令代码为e8，计算0x4011db到0x401196的PC相对地址差，为ff ff ff b6，将其拼装为e8 b6 ff ff ff。
3. 利用GHex将nop代码修改为上述代码，并保存。



1. 修改成功，保存并重新编译后得到预期结果。

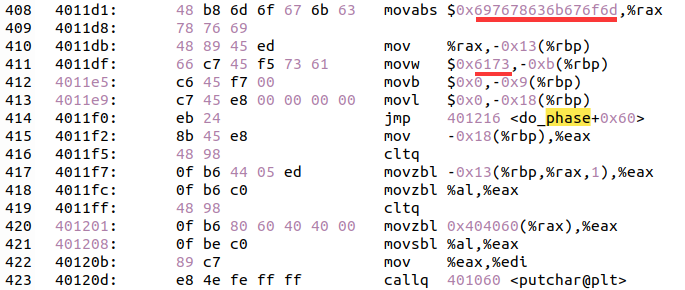
## 3.3 阶段3的分析

程序运行结果截图：

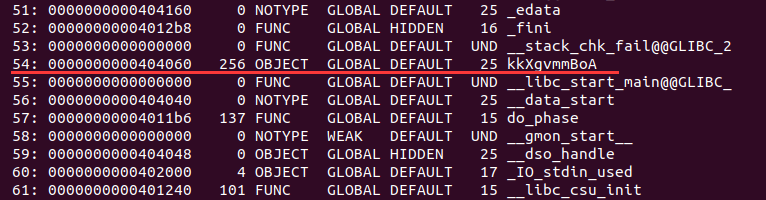


分析与设计的过程：

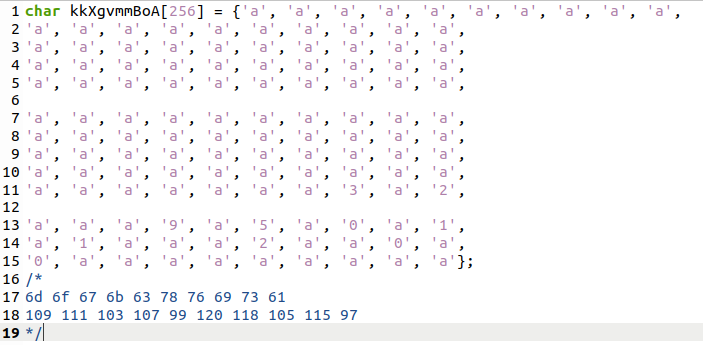
1. 将phase3.o与main.o链接编译并反汇编可得linkbomb3的反汇编代码。
2. 分析do\_phase的反汇编代码可知：do\_phase会输出10个字符。这些字符的起始位置是0x404060，10个字符的位置偏移分别是：6d 6f 67 6b 63 78 76 69 73 61



1. 由readelf linkbomb3 -a可知，一个名为kkXgvmmBoA的长度为256个字节的数组起始地址就是0x404060



1. 那么我们可以编写C代码，对kkXgvmmBoA数组赋初值。由于我们编写的代码中kkXgvmmBoA数组是强符号，链接后原代码中访问到的0x404060地址中的内容就是我们赋值的了。
2. 我们构造kkXgvmmBoA数组的内容，其中第109、111、103、107、99、120、118、105、115、97个字符（下标从0开始）分别为’1’、’1’、’9’、’0’、’2’、’0’、’0’、’5’、’2’、’3’。
3. 得到phase3\_patch.c的代码：



1. 将phase3\_patch.c编译得到phase3\_patch.o，再与phase3.o、main.o编译链接得到新的linkbomb3，运行得到预期结果。

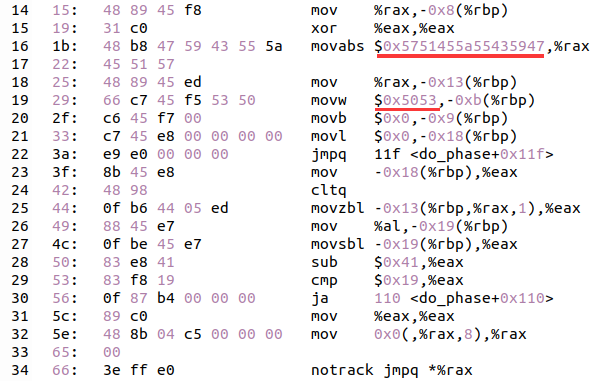
## 3.4 阶段4的分析

程序运行结果截图：

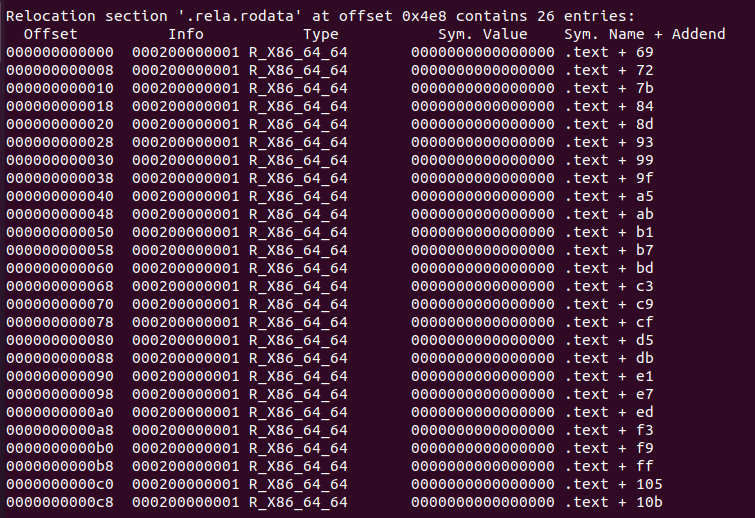


分析与设计的过程：

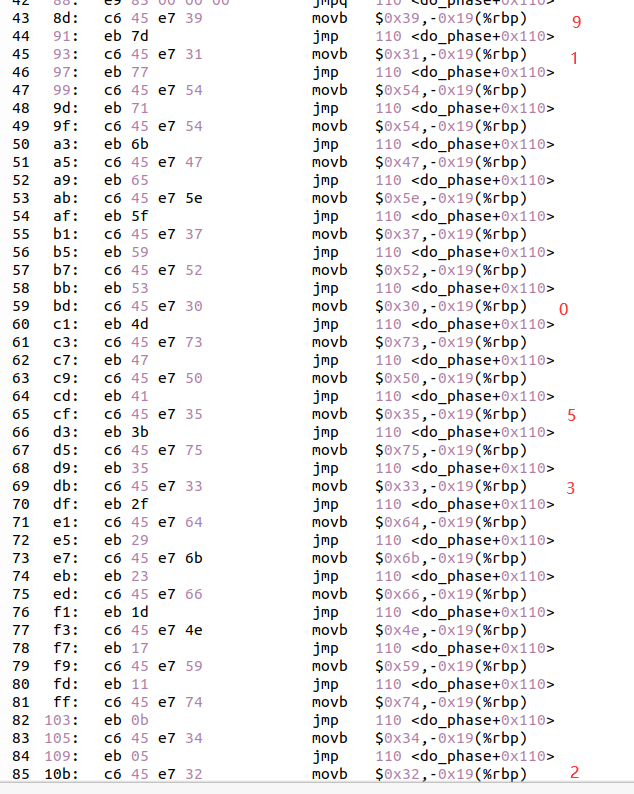
1. 将phase3.o反汇编。
2. 分析do\_phase的反汇编代码可知：do\_phase首先会根据地址表来跳转到相应的代码位置。地址表共有10个地址，这些地址的位置偏移分别是：0x6、0x18、0x2、0x14、0x19、0x4、0x10、0x16、0x12、0xf（原值减去0x41）。转成十进制则为6、24、2、20、25、4、16、22、18、15。



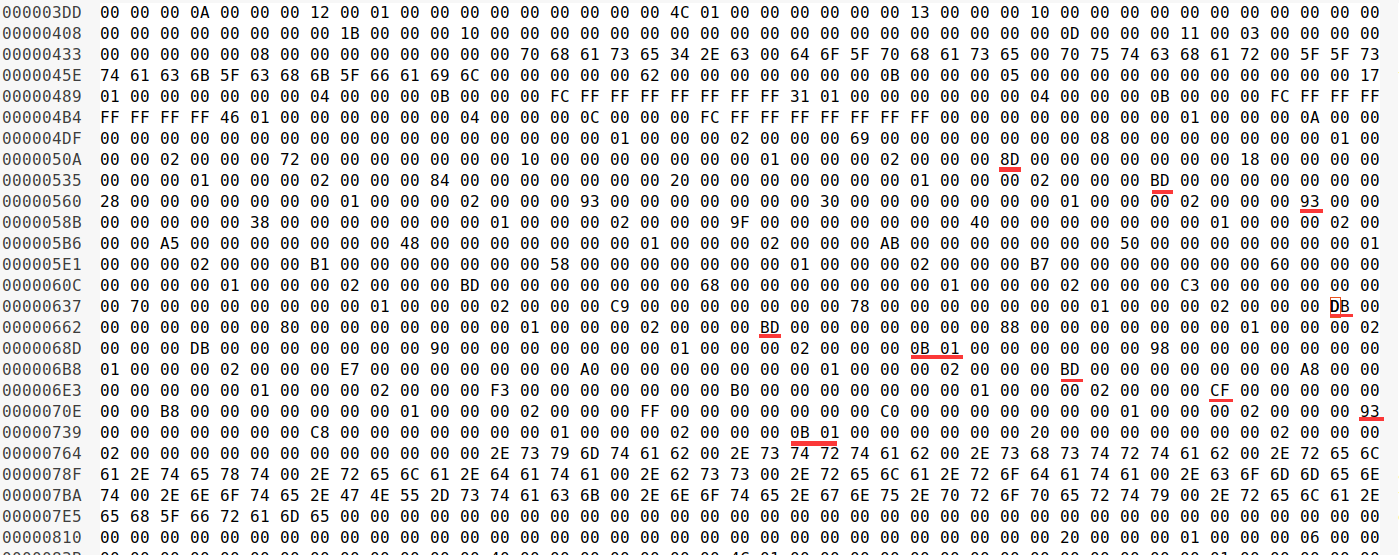
1. 由phase3.o的ELF表可知每个偏移量对应的跳转代码位置，也就知道了实际上要输出的内容。



1. 我们在do\_phase中找到拼成学号所需代码的位置，分别为0x93、0x93、0x8d、0xbd、0x10b、0xbd、0xbd、0xcf、0x10b、0xdb。



1. 即将ELF表中第6、24、2、20、25、4、16、22、18、15个地址（注意下标从0开始）的值分别修改为0x93、0x93、0x8d、0xbd、0x10b、0xbd、0xbd、0xcf、0x10b、0xdb。
2. 利用二进制编辑器GHex，将地址表中的内容修改为上述内容。



1. 修改成功，保存并重新编译后得到预期结果。

## 3.5 阶段5的分析

程序运行结果截图：

分析与设计的过程：

# 第4章 总结

## 4.1 请总结本次实验的收获

1. 更加熟悉了ELF结构；
2. 对程序的运行周期有了更深的理解；
3. 对readelf、objdump、gdb等工具的运用更加熟练。

## 4.2 请给出对本次实验内容的建议

1. 希望实验指引能够更加具体。

# 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.