

# 哈尔滨工业大学

# 实验报告

## 实 验（四）

题 目 LinkLab

链接

专 业 计算学部

学 号 1190200208

班 级 1936602

学 生 李旻翀

指 导 教 师 刘宏伟

实 验 地 点 G709

实 验 日 期 2021.5.20

计算机科学与技术学院

# 目 录

第 1 章 实验基本信息 .....	- 3 -
1.1 实验目的 .....	- 3 -
1.2 实验环境与工具 .....	- 3 -
1.2.1 硬件环境 .....	- 3 -
1.2.2 软件环境 .....	- 3 -
1.2.3 开发工具 .....	- 3 -
1.3 实验预习 .....	- 3 -
第 2 章 实验预习 .....	- 5 -
2.1 ELF 文件格式解读 .....	- 5 -
2.2 程序的内存映像结构 .....	- 5 -
2.3 程序中符号的位置分析 .....	- 6 -
2.4 程序运行过程分析 .....	- 12 -
第 3 章 各阶段的原理与方法 .....	- 14 -
3.1 阶段 1 的分析 .....	- 14 -
3.2 阶段 2 的分析 .....	- 15 -
3.3 阶段 3 的分析 .....	- 18 -
3.4 阶段 4 的分析 .....	- 19 -
3.5 阶段 5 的分析 .....	- 19 -
第 4 章 总结 .....	- 20 -
4.1 请总结本次实验的收获 .....	- 20 -
4.2 请给出对本次实验内容的建议 .....	- 20 -
参考文献 .....	- 21 -

## 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

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

Visual Studio 2010 64 位以上; GDB/OBJDUMP; DDD/EDB 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）

ELF 头
.text
.rodata
.bss
.symtab
.rel.text
.rel.data
.debug
.line
.strtab

### 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）

	0
只读代码段 (.init , .text , .rodata)	0x400000
读写段 (.data , .bss )	
运行时堆 (由 malloc 创建)	
↓	
共享库的内存映射区域	
↓	
↑	
用户栈 (运行时创建)	
内核内存	$2^{48} - 1$

## 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）

只读代码段 (.init , .text , .rodata)	show_pointer 0x557aa0fbd81a 93985175230490 useless 0x557aa0fbd84d 93985175230541 main 0x557aa0fbd858 93985175230552
读写段 (.data , .bss )	big array 0x557ae11bf040 93986251075648 huge array 0x557aa11bf040 93985177333824 local 0x7ffdea0bd4e0 140728530097376 global 0x557aa11bf02c 93985177333804
运行时堆 (由 malloc 创建)	p1 0x7fb06c397010 140395706675216 p2 0x557ae26b167093986273039984 p3 0x7fb07c976010 140395981266960 p4 0x7fb02c396010 140394632929296

	p5 0x7fafac395010 140392485441552
共享库的内存映射区域	exit 0x7fb07c3db240 140395975389760 printf 0x7fb07c3fcf70 140395975528304 malloc 0x7fb07c42f140 140395975733568 free 0x7fb07c42fa30 140395975735856
用户栈(运行时创建)	env 0x7ffdea0bd630 140728530097712 env[0] *env 0x7ffdea0be20b 140728530100747 CLUTTER_IM_MODULE=xim env[1] *env 0x7ffdea0be221 140728530100769 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:*.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:

	<pre> env[2] *env    0x7ffdea0be80d 140728530102285 LC_MEASUREMENT=zh_CN.UTF-8 env[3] *env    0x7ffdea0be828 140728530102312 LESSCLOSE=/usr/bin/lesspipe %s %s env[4] *env    0x7ffdea0be84a 140728530102346 LC_PAPER=zh_CN.UTF-8 env[5] *env    0x7ffdea0be85f 140728530102367 LC_MONETARY=zh_CN.UTF-8 env[6] *env    0x7ffdea0be877 140728530102391 XDG_MENU_PREFIX=gnome- env[7] *env    0x7ffdea0be88e 140728530102414 LANG=zh_CN.UTF-8 env[8] *env    0x7ffdea0be89f 140728530102431 MANAGERPID=1666 env[9] *env    0x7ffdea0be8af 140728530102447 DISPLAY=:0 env[10] *env    0x7ffdea0be8ba 140728530102458 INVOCATION_ID=a46741b3e6db46b1a49c26945a4194 2e env[11] *env    0x7ffdea0be8e9 140728530102505 GNOME_SHELL_SESSION_MODE=ubuntu env[12] *env    0x7ffdea0be909 140728530102537 COLORTERM=truecolor env[13] *env    0x7ffdea0be91d 140728530102557 ZEITGEIST_DATA_PATH=/home/1190200208/.local/share/zeitgeist env[14] *env    0x7ffdea0be959 140728530102617 USERNAME=1190200208 env[15] *env    0x7ffdea0be96d 140728530102637 XDG_VTNR=2 env[16] *env    0x7ffdea0be978 140728530102648 SSH_AUTH_SOCK=/run/user/1000/keyring/ssh env[17] *env    0x7ffdea0be9a1 140728530102689 LC_NAME=zh_CN.UTF-8 env[18] *env    0x7ffdea0be9b5 140728530102709 XDG_SESSION_ID=2 env[19] *env    0x7ffdea0be9c6 140728530102726 </pre>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



	<pre> USER=1190200208 env[20]*env    0x7ffdea0be9d6 140728530102742 DESKTOP_SESSION=ubuntu env[21]*env    0x7ffdea0be9ed 140728530102765 QT4_IM_MODULE=fcitx env[22]*env    0x7ffdea0bea01 140728530102785 TEXTDOMAINDIR=/usr/share/locale/ env[23]*env    0x7ffdea0bea22 140728530102818 GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/ screen/a6d0784e_3d99_4164_8a0a_e10efc8a7449 env[24]*env    0x7ffdea0bea78 140728530102904 PWD=/home/1190200208/桌面/hitics/lab5_additional env[25]*env    0x7ffdea0beaab 140728530102955 HOME=/home/1190200208 env[26]*env    0x7ffdea0beac1 140728530102977 JOURNAL_STREAM=9:49426 env[27]*env    0x7ffdea0bead8 140728530103000 TEXTDOMAIN=im-config env[28]*env    0x7ffdea0beaed 140728530103021 SSH_AGENT_PID=1788 env[29]*env    0x7ffdea0beb00 140728530103040 QT_ACCESSIBILITY=1 env[30]*env    0x7ffdea0beb13 140728530103059 XDG_SESSION_TYPE=x11 env[31]*env    0x7ffdea0beb28 140728530103080 XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share/:/u sr/share/./var/lib/snapd/desktop env[32]*env    0x7ffdea0beb7d 140728530103165 XDG_SESSION_DESKTOP=ubuntu env[33]*env    0x7ffdea0beb98 140728530103192 LC_ADDRESS=zh_CN.UTF-8 env[34]*env    0x7ffdea0beba7 140728530103215 DBUS_STARTER_ADDRESS=unix:path=/run/user/1000 /bus,guid=342e637c7f91140e34a96f3a60920cdb env[35]*env    0x7ffdea0bec07 140728530103303 LC_NUMERIC=zh_CN.UTF-8 env[36]*env    0x7ffdea0bec1e 140728530103326 </pre>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<pre> GTK_MODULES=gail:atk-bridge env[37]*env 0xffdea0bec3a 140728530103354 WINDOWPATH=2 env[38]*env 0xffdea0bec47 140728530103367 TERM=xterm-256color env[39]*env 0xffdea0bec5b 140728530103387 VTE_VERSION=5202 env[40]*env 0xffdea0bec6c 140728530103404 SHELL=/bin/bash env[41]*env 0xffdea0bec7c 140728530103420 QT_IM_MODULE=fcitx env[42]*env 0xffdea0bec8f 140728530103439 XMODIFIERS=@im=fcitx env[43]*env 0xffdea0beca4 140728530103460 IM_CONFIG_PHASE=2 env[44]*env 0xffdea0becb6 140728530103478 DBUS_STARTER_BUS_TYPE=session env[45]*env 0xffdea0becd4 140728530103508 XDG_CURRENT_DESKTOP=ubuntu:GNOME env[46]*env 0xffdea0becf5 140728530103541 GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent: 0:1 env[47]*env 0xffdea0bed29 140728530103593 GNOME_TERMINAL_SERVICE=:1.127 env[48]*env 0xffdea0bed47 140728530103623 SHLVL=1 env[49]*env 0xffdea0bed4f 140728530103631 XDG_SEAT=seat0 env[50]*env 0xffdea0bed5e 140728530103646 LANGUAGE=zh_CN:zh:en_US:en env[51]*env 0xffdea0bed79 140728530103673 LC_TELEPHONE=zh_CN.UTF-8 env[52]*env 0xffdea0bed92 140728530103698 GDMSESSION=ubuntu env[53]*env 0xffdea0beda4 140728530103716 GNOME_DESKTOP_SESSION_ID=this-is-deprecated env[54]*env 0xffdea0bedd0 140728530103760 </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<pre> LOGNAME=1190200208 env[55] *env 0x7ffdea0bede3 140728530103779 DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user /1000/bus,guid=342e637c7f91140e34a96f3a60920cdb env[56] *env 0x7ffdea0bee3f 140728530103871 XDG_RUNTIME_DIR=/run/user/1000 env[57] *env 0x7ffdea0bee5e 140728530103902 XAUTHORITY=/run/user/1000/gdm/Xauthority env[58] *env 0x7ffdea0bee87 140728530103943 XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg env[59] *env 0x7ffdea0beeb4 140728530103988 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbi n:/bin:/usr/games:/usr/local/games:/snap/bin env[60] *env 0x7ffdea0bef1c 140728530104092 LC_IDENTIFICATION=zh_CN.UTF-8 env[61] *env 0x7ffdea0bef3a 140728530104122 SESSION_MANAGER=local/MincooLee:@/tmp/.ICE-u nix/1695,unix/MincooLee:/tmp/.ICE-unix/1695 env[62] *env 0x7ffdea0bef92 140728530104210 LESSOPEN=  /usr/bin/lesspipe %s env[63] *env 0x7ffdea0befb2 140728530104242 GTK_IM_MODULE=fcitx env[64] *env 0x7ffdea0befc6 140728530104262 LC_TIME=zh_CN.UTF-8 env[65] *env 0x7ffdea0befda 140728530104282 _=./LinkAddress argc 0x7ffdea0bd4dc 140728530097372 argv 0x7ffdea0bd608 140728530097672 argv[0] 7ffdea0be1e5 argv[1] 7ffdea0be1f3 argv[2] 7ffdea0be1f6 argv[3] 7ffdea0be201 argv[0] 0x7ffdea0be1e5 140728530100709 ./LinkAddress argv[1] 0x7ffdea0be1f3 140728530100723 -u argv[2] 0x7ffdea0be1f6 140728530100726 </pre>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	1190200208 argv[3] 0x7ffdea0be201 140728530100737 李旻翀
--	-------------------------------------------------------------

## 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)

```

1190200208@MincooLee: ~/桌面/hitcs/lab5_additional
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
1190200208@MincooLee:~/桌面/hitcs/lab5_additional$ gdb LinkAddress
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from LinkAddress...(no debugging symbols found)...done.
(gdb) info functions
All defined functions:

Non-debugging symbols:
0x00000000000006a8 _init
0x00000000000006d0 puts@plt
0x00000000000006e0 __stack_chk_fail@plt
0x00000000000006f0 free@plt
0x00000000000006f8 printf@plt
0x0000000000000700 malloc@plt
0x0000000000000708 __cxa_finalize@plt
0x0000000000000710 _start
0x0000000000000740 deregister_tm_clones
0x0000000000000780 register_tm_clones
0x00000000000007d0 __do_global_ctors_aux
0x0000000000000810 frame_dummy
0x000000000000081a show_pointer
0x000000000000084d useless
0x0000000000000858 main
0x0000000000000c40 __libc_csu_init
0x0000000000000cb0 __libc_csu_fini
0x0000000000000cb4 _fini
(gdb) █

```

main 函数执行前	0x000000000000006a8	_init
	0x000000000000006d0	puts@plt
	0x000000000000006e0	__stack_chk_fail@plt
	0x000000000000006f0	free@plt
	0x000000000000006f8	printf@plt
	0x00000000000000700	malloc@plt
	0x00000000000000708	__cxa_finalize@plt
	0x00000000000000710	_start
	0x00000000000000740	deregister_tm_clones
	0x00000000000000780	register_tm_clones
	0x000000000000007d0	__do_global_ctors_aux
	0x00000000000000810	frame_dummy
	0x0000000000000081a	show_pointer
	0x0000000000000084d	useless
main 函数执行后	0x00000000000000c40	__libc_csu_init
	0x00000000000000cb0	__libc_csu_fini
	0x00000000000000cb4	_fini

## 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：

```
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ gcc -m32 -o linkbomb1 main.o phase1.o
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ ./linkbomb1
1190200208
```

分析与设计的过程：

首先生成 main.o 的可执行文件，发现打印欢迎字符：

```
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ gcc -m32 -o linkbomb main.o
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ ./linkbomb
Welcome to this small lab of linking. To begin lab, please link the relevant object module(s) with the main module.
```

再链接 main.o 与 phase1.o，发现得到的可执行文件为一串乱码，我们需要将这段乱码改成自己的学号。

```
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ gcc -m32 -o linkbomb main.o phase1.o
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ ./linkbomb
7lcFG9U0TkBS MMSfrEYBUpS1TNErWzt9qzhVhmHiYs9Nbvy8FmM0bf0pDD8Qw vI
```

通过 HEX Editor 打开 phase1.o，查看内容，发现里面有与上面乱码对应的字符串。便考虑通过修改这一字段来使之打印学号。

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	012	456789ABCDEF
0x000	7F45	4C46	0101	0100	0000	0000	0000	0000	01	ELF.....
0x010	0100	0300	0100	0000	0000	0000	0000	0000	02	.....
0x020	B402	0000	0000	0000	3400	0000	0000	2800	03	?.....4.....(.
0x030	0E00	0D00	F30F	1EFB	5589	E583	EC08	B803	04	.....?U?????.?
0x040	0000	0083	EC0C	50E8	FCFF	FFFF	83C4	1090	05	...??P?? ??.
0x050	C9C3	0000	0000	0000	0000	0000	0000	0000	06	??.....
0x060	7469	4937	6C63	4647	3955	4F54	6B42	5309	07	tiI7lcFG9U0TkBS.
0x070	4D4D	5366	7245	5942	5570	5331	544E	4572	08	MMSfrEYBUpS1TNEr
0x080	577A	7439	717A	6856	686D	4869	5973	394E	09	Wzt9qzhVhmHiYs9N
0x090	6276	7938	466D	4D30	6266	3070	4444	3851	0A	bvy8FmM0bf0pDD8Q
0x0A0	7720	7649	0000	0000	0000	0000	0047	4343	0B	w vI.....GCC
0x0B0	3A20	2855	6275	6E74	7520	392E	322E	312D	0C	:(Ubuntu 9.2.1-
0x0C0	3975	6275	6E74	7532	2920	392E	322E	3120	0D	9ubuntu2) 9.2.1
0x0D0	3230	3139	3130	3038	0000	0000	0400	0000	0E	20191008.....
0x0E0	0C00	0000	0500	0000	474E	5500	0200	00C0	0F	.....GNU....?

相同的乱码字段！

将 7lcFG9 开头的字段修改为 110200208\0，便实现了对 phase1.o 的修改。

0x050	C9C3	0000	0000	0000	0000	0000	0000	0000	??	.....
0x060	7469	4931	3139	3032	3030	3230	3800	5309	ti1	1190200208.S.
0x070	4D4D	5366	7245	5942	5570	5331	544E	4572	MMSfr	EYBUpSlTNEr
0x080	577A	7439	717A	6856	686D	4869	5973	394E	Wzt9qzh	VhmHiYs9N
0x090	6276	7938	466D	4D30	6266	3070	4444	3851	bvy8FmM0bf0p	DD8Q

完成上述工作后，重新链接，成功显示自己学号。

## 3.2 阶段 2 的分析

程序运行结果截图：

```

1190200208@MincooLee: ~/桌面/hitcs/linklab-1190200208
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ gcc -m32 -o linkbomb2 mai
n.o phase2.o -no-pie
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ ./linkbomb2
1190200208

```

分析与设计的过程：

首先通过如下命令，将 main.o 与 phase2.o 链接，生成 linkbomb2 文件，通过 objdump 将其汇编代码保存在 linkbomb2.txt 中。

```

1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ gcc -m32 -o linkbomb2 main.o phase2.o -no-pie
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ objdump -d -s linkbomb2 > linkbomb2.txt

```

查看生成的汇编代码文件，发现在函数 hONfoDPL 中调用了 strcmp 函数，且在调用前，向栈中压入了两个值，结合 PPT 上的提示，考虑这两个值分别为 id 和 MYID。



```

0804849a <hONfoDPL>:
0804849a:    f3 0f 1e fb          endbr32
0804849e:    55                   push    %ebp
0804849f:    89 e5                mov     %esp,%ebp
080484a1:    83 ec 08             sub     $0x8,%esp
080484a4:    83 ec 08             sub     $0x8,%esp
080484a7:    68 f4 85 04 08       push    $0x80485f4
080484ac:    ff 75 08             pushl   0x8(%ebp)
080484af:    e8 4c fe ff ff       call    8048300 <strcmp@plt>
080484b4:    83 c4 10             add     $0x10,%esp
080484b7:    85 c0                test    %eax,%eax
080484b9:    75 10                jne     80484cb <hONfoDPL+0x31>
080484bb:    83 ec 0c             sub     $0xc,%esp
080484be:    ff 75 08             pushl   0x8(%ebp)
080484c1:    e8 4a fe ff ff       call    8048310 <puts@plt>
080484c6:    83 c4 10             add     $0x10,%esp
080484c9:    eb 01                jmp     80484cc <hONfoDPL+0x32>
080484cb:    90                   nop
080484cc:    c9                   leave
080484cd:    c3                   ret

```

查看 0x80485f4 地址对应的值,发现正好为学号 1190200208 对应的 10 位 ASCII 码值。

```

(gdb) x /10xb 0x80485f4
0x80485f4:    0x31    0x31    0x39    0x30    0x32    0x30    0x30    0x32
0x80485fc:    0x30    0x38
(gdb)

```

30为0的ASCII码,这一串数字对应学号1190200208

由 PPT 提示,我们的目标是修改 phase2.o 对应的 do\_phase() 函数,使其能够把我们的学号压栈并跳转至函数 hONfoDPL()。

```

static void OUTPUT_FUNC_NAME( const char *id ) {
// 该函数名对每名学生均不同
    if( strcmp(id,MYID) != 0 ) return;
    printf("%s\n", id);
}

```

**注释:** 各阶段phase[n].c中的全局函数指针变量phase是经初始化的“强”符号,在将phase[n].o模块与main.o链接后,前者中的phase变量定义将取代后者中的同名“弱”符号(变量),因此相应阶段中完成具体功能的do\_phase函数将被调用执行。

我们可以采取这样的思路:

修改 phase2.o 中的 do\_phase() 函数,使其将学号压栈,再跳转到 hONfoDPL()。

跳转到 hONfoDPL() 需要确定 call 指令码后的操作数。

由此,我们可以构建如下汇编代码,其中的 call 0xe 仅为占满位置,具体的地



址通过稍后步骤分析得出：

```
push $0x80485f4  → 将学号的地址压
call 0xe        → 填充位置
mov %ebp,%esp
                → 恢复栈帧
```

经过汇编与反汇编，我们得到对应的机器代码：

```
1190200208@MincoolLee:~/桌面/hitics$ gcc -m32 -c AssC
1190200208@MincoolLee:~/桌面/hitics$ objdump -d AssC

AssCode.o:          文件格式 elf32-i386

Disassembly of section .text:

00000000 <.text>:
   0:  68 f4 85 04 08          push    $0x80485f4
```

其中，需要将 call 指令 e8 后的 4 字节用相对寻址改为 hONfoDPL ( ) 的首地址。

操作数=跳转到的目标地址 - call 指令后一条指令长度的起始地址

目标地址：0804849a <hONfoDPL>:

call 指令下条指令地址：

```
080484ce <do_phase>:
  080484ce:  f3 0f 1e fb          endbr32
  080484d2:  55                   push    %ebp
  080484d3:  89 e5                mov     %esp,%ebp
  080484d5:  90                   nop
  080484d6:  90                   nop
  080484d7:  90                   nop
  080484d8:  90                   nop
  080484d9:  90                   nop
  080484da:  90                   nop
  080484db:  90                   nop
```

故操作数=0x804849a-(0xa+0x80484d5)=0x804849a-0x80484df=-0x45

用补码表示为 0xFFFFFBB

所以整体汇编代码为：

68 f4 85 04 08 e8 bb ff ff ff 89 ec

打开 hexedit，找到 ASCII 码为 90 的代码段：

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF
0x000	7F45	4C46	0101	0100	0000	0000	0000	0000	ELF.....
0x010	0100	0300	0100	0000	0000	0000	0000	0000	.....
0x020	0403	0000	0000	0000	3400	0000	0000	2800	.....4.....(.
0x030	0F00	0E00	F30F	1EFB	5589	E583	EC08	83EC	....?..?U??????
0x040	0868	0000	0000	FF75	08E8	FCFF	FFFF	83C4	.h.... u.?? ??
0x050	1085	C075	1083	EC0C	FF75	08E8	FCFF	FFFF	.??u.?? u.??
0x060	83C4	10EB	0190	C9C3	F30F	1EFB	5589	E590	???.?????..?U???
0x070	9090	9090	9090	9090	9090	9090	9090	9090	????????????????
0x080	9090	9090	9090	9090	9090	9090	9090	9090	????????????????
0x090	5DC3	0000	0000	0000	3131	3930	3230	3032	]?.....11902002
0x0A0	3038	0000	4743	433A	2028	5562	756E	7475	08..GCC: (Ubuntu
0x0B0	2039	2E32	2E31	2D39	7562	756E	7475	3229	9.2.1-9ubuntu2)
0x0C0	2039	2E32	2E31	2032	3031	3931	3030	3800	9.2.1 20191008.
0x0D0	0400	0000	0C00	0000	0500	0000	474E	5500	.....GNU.
0x0E0	0200	00C0	0400	0000	0300	0000	1400	0000	...?.....

从第一个 ASCII 码 90 开始修改，替换为我们的汇编代码。

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF
0x000	7F45	4C46	0101	0100	0000	0000	0000	0000	ELF.....
0x010	0100	0300	0100	0000	0000	0000	0000	0000	.....
0x020	0403	0000	0000	0000	3400	0000	0000	2800	.....4.....(.
0x030	0F00	0E00	F30F	1EFB	5589	E583	EC08	83EC	....?..?U??????
0x040	0868	0000	0000	FF75	08E8	FCFF	FFFF	83C4	.h.... u.?? ??
0x050	1085	C075	1083	EC0C	FF75	08E8	FCFF	FFFF	.??u.?? u.??
0x060	83C4	10EB	0190	C9C3	F30F	1EFB	5589	E568	???.?????..?U???
0x070	F485	0408	E8BB	FFFF	FF89	EC90	9090	9090	???.?? ????h
0x080	9090	9090	9090	9090	9090	9090	9090	9090	????????????????
0x090	5DC3	0000	0000	0000	3131	3930	3230	3032	]?.....11902002
0x0A0	3038	0000	4743	433A	2028	5562	756E	7475	08..GCC: (Ubuntu
0x0B0	2039	2E32	2E31	2D39	7562	756E	7475	3229	9.2.1-9ubuntu2)
0x0C0	2039	2E32	2E31	2032	3031	3931	3030	3800	9.2.1 20191008.
0x0D0	0400	0000	0C00	0000	0500	0000	474E	5500	.....GNU.

替换后，重新链接执行，成功显示学号。

```

1190200208@MincooLee: ~/桌面/hitcs/linklab-1190200208
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ gcc -m32 -o linkbomb2 mai
n.o phase2.o -no-pie
1190200208@MincooLee:~/桌面/hitcs/linklab-1190200208$ ./linkbomb2
1190200208

```

### 3.3 阶段 3 的分析

程序运行结果截图：

分析与设计的过程：

### 3.4 阶段 4 的分析

程序运行结果截图：

分析与设计的过程：

### 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

对 ELF 格式文件结构以及文件链接的理解进一步加深  
学习了 HEX Editor 的使用，能够通过这一工具对.o 文件进行修改  
对 Linux 下汇编与反汇编工具的使用更加熟悉

### 4.2 请给出对本次实验内容的建议

难度梯度较大，而且在教学时对于这一部分内容并未涉及过多。建议给予更为详细的指导，否则做起来有些力不从心。

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>.