

**实验报告**

**实 验（四）**

题 目 LinkLab

链接

专 业 计算机

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 11.21.2019

**计算机科学与技术学院**

**目 录**

[第1章 实验基本信息 - 3 -](#_Toc25154534)

[1.1 实验目的 - 3 -](#_Toc25154535)

[1.2 实验环境与工具 - 3 -](#_Toc25154536)

[1.2.1 硬件环境 - 3 -](#_Toc25154537)

[1.2.2 软件环境 - 3 -](#_Toc25154538)

[1.2.3 开发工具 - 3 -](#_Toc25154539)

[1.3 实验预习 - 3 -](#_Toc25154540)

[第2章 实验预习 - 4 -](#_Toc25154541)

[2.1 ELF文件格式解读 - 4 -](#_Toc25154542)

[2.2程序的内存映像结构 - 4 -](#_Toc25154543)

[2.3程序中符号的位置分析 - 4 -](#_Toc25154544)

[2.4程序运行过程分析 - 4 -](#_Toc25154545)

[第3章 各阶段的原理与方法 - 5 -](#_Toc25154546)

[3.1 阶段1的分析 - 5 -](#_Toc25154547)

[3.2 阶段2的分析 - 5 -](#_Toc25154548)

[3.3 阶段3的分析 - 5 -](#_Toc25154549)

[3.4 阶段4的分析 - 5 -](#_Toc25154550)

[3.5 阶段5的分析 - 5 -](#_Toc25154551)

[第4章 总结 - 6 -](#_Toc25154552)

[4.1 请总结本次实验的收获 - 6 -](#_Toc25154553)

[4.2 请给出对本次实验内容的建议 - 6 -](#_Toc25154554)

[参考文献 - 7 -](#_Toc25154555)

# 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 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

段头部表

.init节 程序初始化代码

.text节 代码

.rodata节 只读数据

.data节 已初始化的全局/静态变量

.bss节 未初始化的全局/静态变量

.symtab节 符号表

.debug节 调试符号表

.line节 代码行号与机器指令的映射

.strtab节 字符串表

节头部表

## 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）

只读代码段(.init,.text,.rodata)

读/写段(.data,.bss)

运行时堆(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(对用户代码不可见)

## 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、空间。并按照Linux下X64内存映像标出其所属各区（5分）

只读代码段(.init,.text,.rodata)：

exit 0x7f264c106840 139802461628480

printf 0x7f264c11f1c0 139802461729216

malloc 0x7f264c152e10 139802461941264

free 0x7f264c153450 139802461942864

读/写段(.data,.bss)

show\_pointer 0x7f264c2db169 139802463547753

useless 0x7f264c2db19c 139802463547804

main 0x7f264c2db1a7 139802463547815

big array 0x7f268c2de060 139803537301600

huge array 0x7f264c2de060 139802463559776

global 0x7f264c2de04c 139802463559756

运行时堆(malloc创建)

p1 0x7f263c0c0010 139802192904208

p2 0x7f263c090010 139802192707600

p3 0x7f263c060010 139802192510992

p4 0x7f25fc050010 139801118703632

p5 0x7f257c040010 139798971154448

用户栈(运行时创建)

env 0x7fffe1a7ff00 140736979271424

env[0] \*env 0x7fffe1a80155 140736979272021

HOSTTYPE=x86\_64

env[1] \*env 0x7fffe1a80165 140736979272037

LANG=en\_US.UTF-8

env[2] \*env 0x7fffe1a80176 140736979272054

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/mnt/c/FerdinandScripts:/mnt/c/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0/bin:/mnt/c/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0/libnvvp:/mnt/c/Program Files (x86)/NetSarang/Xshell 6/:/mnt/c/Program Files (x86)/NetSarang/Xftp 6/:/mnt/c/ProgramData/Anaconda3:/mnt/c/ProgramData/Anaconda3/Library/mingw-w64/bin:/mnt/c/ProgramData/Anaconda3/Library/usr/bin:/mnt/c/ProgramData/Anaconda3/Library/bin:/mnt/c/ProgramData/Anaconda3/Scripts:/mnt/c/Windows/system32:/mnt/c/MinGW/bin:/mnt/c/Windows:/mnt/c/Windows/System32/Wbem:/mnt/c/Windows/System32/WindowsPowerShell/v1.0/:/mnt/c/Windows/System32/OpenSSH/:/mnt/c/Program Files/Git/cmd:/mnt/c/Program Files/dotnet/:/mnt/c/Program Files/Microsoft SQL Server/130/Tools/Binn/:/mnt/c/Program Files/Microsoft SQL Server/Client SDK/ODBC/170/Tools/Binn/:/mnt/c/Program Files/MATLAB/R2018b/runtime/win64:/mnt/c/Program Files/MATLAB/R2018b/bin:/mnt/c/Program Files (x86)/NVIDIA Corporation/PhysX/Common:/mnt/c/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0:/mnt/c/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0/lib/x64:/mnt/c/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0/bin:/%CUDA\_SDK\_PATH%/common/lib/x64:/%CUDA\_SDK\_PATH%/bin/win64:/mnt/c/Program Files/Microsoft SQL Server/Client SDK/ODBC/130/Tools/Binn/:/mnt/c/Program Files (x86)/Microsoft SQL Server/140/Tools/Binn/:/mnt/c/Program Files/Microsoft SQL Server/140/Tools/Binn/:/mnt/c/Program Files/Microsoft SQL Server/140/DTS/Binn/:/mnt/c/Program Files (x86)/Microsoft SQL Server/150/DTS/Binn/:/mnt/c/Program Files (x86)/IncrediBuild:/mnt/c/Program Files/NVIDIA Corporation/NVIDIA NvDLISR:/mnt/c/Users/Ferdinand/AppData/Local/Microsoft/WindowsApps:/mnt/c/Users/Ferdinand/AppData/Local/Programs/Microsoft VS Code/bin:/mnt/c/Users/Ferdinand/.dotnet/tools:/usr/bin/site\_perl:/usr/bin/vendor\_perl:/usr/bin/core\_perl

env[3] \*env 0x7fffe1a80906 140736979273990

TERM=xterm-256color

env[4] \*env 0x7fffe1a8091a 140736979274010

WSLENV=

env[5] \*env 0x7fffe1a80922 140736979274018

NAME=Ferdinand-IS

env[6] \*env 0x7fffe1a80934 140736979274036

HOME=/home/ferdinand

env[7] \*env 0x7fffe1a80949 140736979274057

USER=ferdinand

env[8] \*env 0x7fffe1a80958 140736979274072

LOGNAME=ferdinand

env[9] \*env 0x7fffe1a8096a 140736979274090

SHELL=/usr/bin/zsh

env[10] \*env 0x7fffe1a8097d 140736979274109

WSL\_DISTRO\_NAME=Arch

env[11] \*env 0x7fffe1a80992 140736979274130

SHLVL=1

env[12] \*env 0x7fffe1a8099a 140736979274138

PWD=/mnt/d/Documents/CSAPP/Lab5

env[13] \*env 0x7fffe1a809ba 140736979274170

OLDPWD=/home/ferdinand

env[14] \*env 0x7fffe1a809d1 140736979274193

DISPLAY=localhost:0

env[15] \*env 0x7fffe1a809e5 140736979274213

DBUS\_SESSION\_BUS\_ADDRESS=unix:abstract=/tmp/dbus-rK77II456A,guid=c2eb8d13835ba4610a8f59465dd7a282

env[16] \*env 0x7fffe1a80a47 140736979274311

DBUS\_SESSION\_BUS\_PID=12

env[17] \*env 0x7fffe1a80a5f 140736979274335

LIBGL\_ALWAYS\_INDIRECT=1

env[18] \*env 0x7fffe1a80a77 140736979274359

ZSH=/home/ferdinand/.oh-my-zsh

env[19] \*env 0x7fffe1a80a96 140736979274390

PAGER=less

env[20] \*env 0x7fffe1a80aa1 140736979274401

LESS=-R

env[21] \*env 0x7fffe1a80aa9 140736979274409

LSCOLORS=Gxfxcxdxbxegedabagacad

env[22] \*env 0x7fffe1a80ac9 140736979274441

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[23] \*env 0x7fffe1a810ab 140736979275947

\_=/mnt/d/Documents/CSAPP/Lab5/./linkaddress

local 0x7fffe1a7fdb0 140736979271088

argc 0x7fffe1a7fdac 140736979271084

argv 0x7fffe1a7fed8 140736979271384

argv[0] 7fffe1a8012f

argv[1] 7fffe1a8013d

argv[2] 7fffe1a80140

argv[3] 7fffe1a8014b

argv[0] 0x7fffe1a8012f 140736979271983

./linkaddress

argv[1] 0x7fffe1a8013d 140736979271997

-u

argv[2] 0x7fffe1a80140 140736979272000

1180300303

argv[3] 0x7fffe1a8014b 140736979272011

宿梓航

## 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）

<main运行前>

malloc ()

free ()

\_init ()

\_start ()

\_\_libc\_start\_main ()

\_\_libc\_csu\_init ()

frame\_dummy ()

register\_tm\_clones ()

main ()

useless ()

show\_pointer ()

printf ()

malloc ()

puts ()

free ()

exit ()

\_\_do\_global\_dtors\_aux ()

\_\_cxa\_finalize ()

deregister\_tm\_clones ()

\_dl\_fini ()

\_fini ()

\_\_run\_exit\_handlers ()

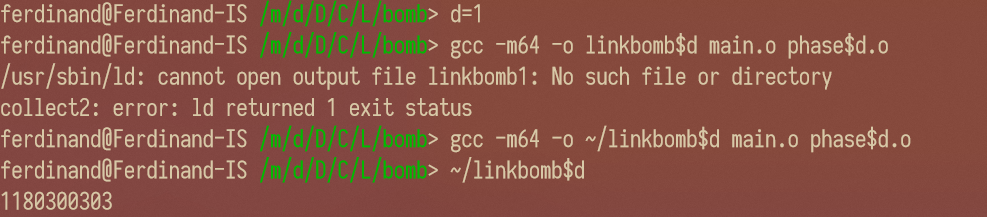
<main运行后>

# 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：

运行未修改版本编译成的linkbomb1，得到乱码字符串；

然后再用hexedit打开phase1.o，找到字符串相应位置，将其内容改为

“1180300303\0”

即

31 31 38 30 33 30 30 33 30 33 00

即可

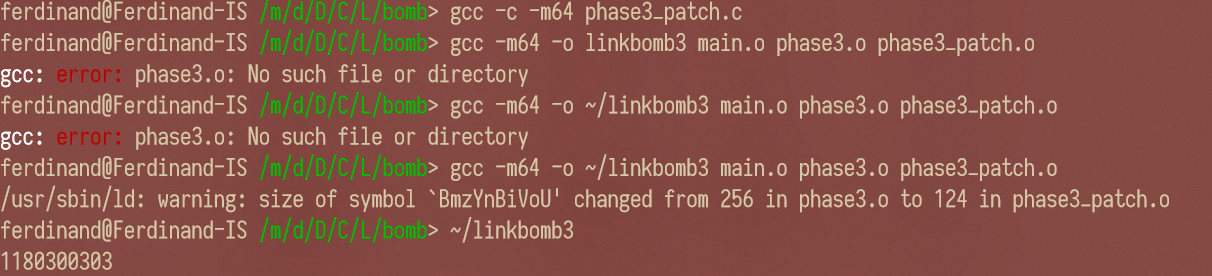
## 3.2 阶段2的分析

程序运行结果截图：

分析与设计的过程：

## 3.3 阶段3的分析

程序运行结果截图：



分析与设计的过程：

依题意，我们只需要再phase3\_patch.c中定义一个附了合适初始值的BmzYnBiVoU变量，即可覆盖原phase3.c中的对应变量，完成期待效果。

该初始值使用C#代码生成：

var dictionary = "ycnzptedxq";

var number = "1180300303";

var sb = new StringBuilder();

for (var i = (char)0; i < 'z' + 1; i++)

{

sb.Append(dictionary.Contains(i) ? number[dictionary.IndexOf(i)] : '-');

}

System.IO.File.WriteAllText(@"D:\p3p.txt", sb.ToString());

结果为

---------------------------------------------------------------------------------------------------130--------8-33--0---010

写入phase3\_patch.c

然后编译即可。

## 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

## 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.