**题目：第7题：在Linux中下修改一个现有的elf可执行程序**

**摘要：**该功能是为了实现“让该程序运行后先执行一个特别的附件功能（附加功能是：创建或打开一个指定文件写入helloworld的字符串）后再继续运行该程序，否则程序结束。并且附加功能嵌入到了原来的程序中”。

实现思路是：

1. 先生成一个 elf 程序，并通过 FILE \*fp = fopen("test", "rb+") 读取该 elf 文件
2. 接着获取 elf 文件的入口地址，程序头，节区头等信息
3. 然后写出要添加的附加功能的汇编程序并编译获取机器码，通过 readelf -e elf文件名 查看 elf 文件的程序入口地址等信息，并根据 elf 文件的程序地址等信息修改生成的机器码中相应的地址信息
4. 将修改好的机器码添加进elf的text段的最后（为了方便，此处我将数据也一并添加进了text段）
5. 修改elf 的入口地址为嵌入的程序首地址，当嵌入程序执行结束再跳转到原 elf 文件的首地址，使程序继续执行原本的 elf 文件程序，再修改 elf 的程序头，节区头
6. 通过 fclose(fp) 关闭 elf 文件

**关键词：**生成机器码，读取 elf 文件，修改 elf 文件

**姓名--学号--班级：**

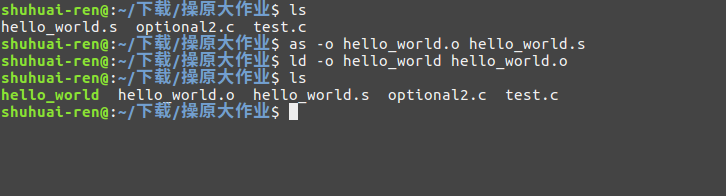
1. 刘俊傲--U201617047(软工1603)
2. 任抒怀--U201617052(软工1603)
3. 周振宇--U201617064(软工1603)

**作业概述：**该功能是为了实现“让该程序运行后先执行一个特别的附件功能（附加功能是：创建或打开一个指定文件写入helloworld的字符串）后再继续运行该程序，否则程序结束。并且附加功能嵌入到了原来的程序中”。

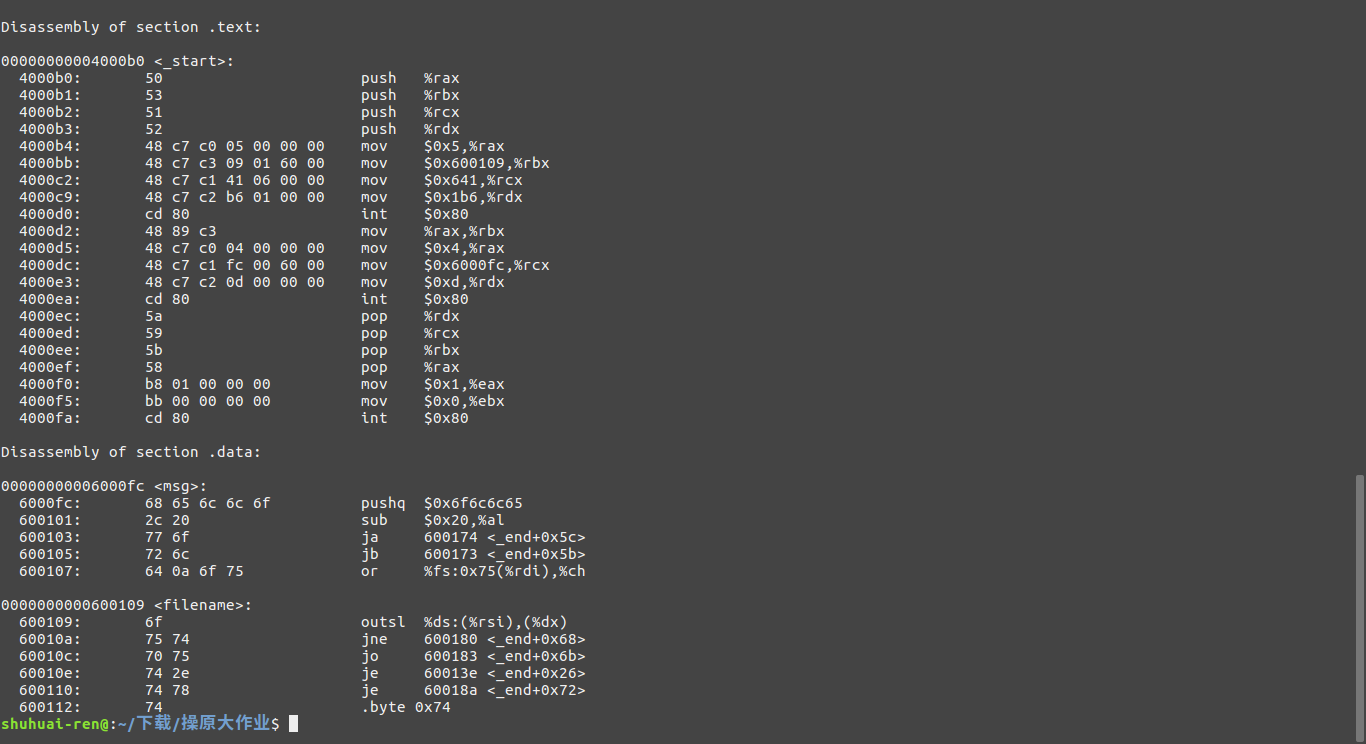
实现思路是：

1. 先生成一个 elf 程序，并通过 FILE \*fp = fopen("test", "rb+") 读取该 elf 文件
2. 接着获取 elf 文件的入口地址，程序头，节区头等信息
3. 然后写出要添加的附加功能的汇编程序并编译获取机器码，通过 readelf -e elf文件名 查看 elf 文件的程序入口地址等信息，并根据 elf 文件的程序地址等信息修改生成的机器码中相应的地址信息
4. 将修改好的机器码添加进elf的text段的最后（为了方便，此处我将数据也一并添加进了text段）
5. 修改elf 的入口地址为嵌入的程序首地址，当嵌入程序执行结束再跳转到原 elf 文件的首地址，使程序继续执行原本的 elf 文件程序，再修改 elf 的程序头，节区头
6. 通过 fclose(fp) 关闭 elf 文件
7. **功能需求：**
8. **设计思路：**
9. **开发环境下载，安装和配置：**
10. **程序的难点和核心技术分析：**
11. **运行和测试过程：**
12. 首先通过下面两条指令编译，链接 hello\_world.s 汇编代码

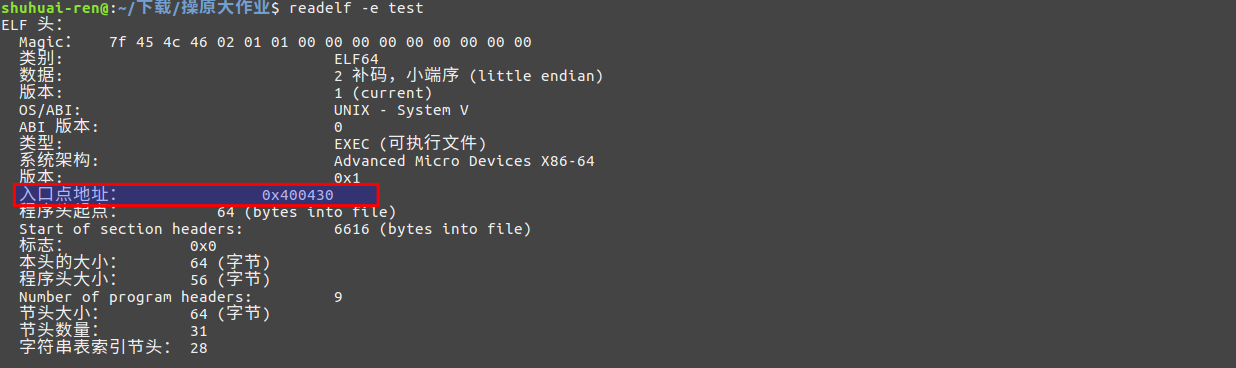
|  |
| --- |
| as -o hello\_world.o hello\_world.s |
| ld -o hello\_world hello\_world.o |

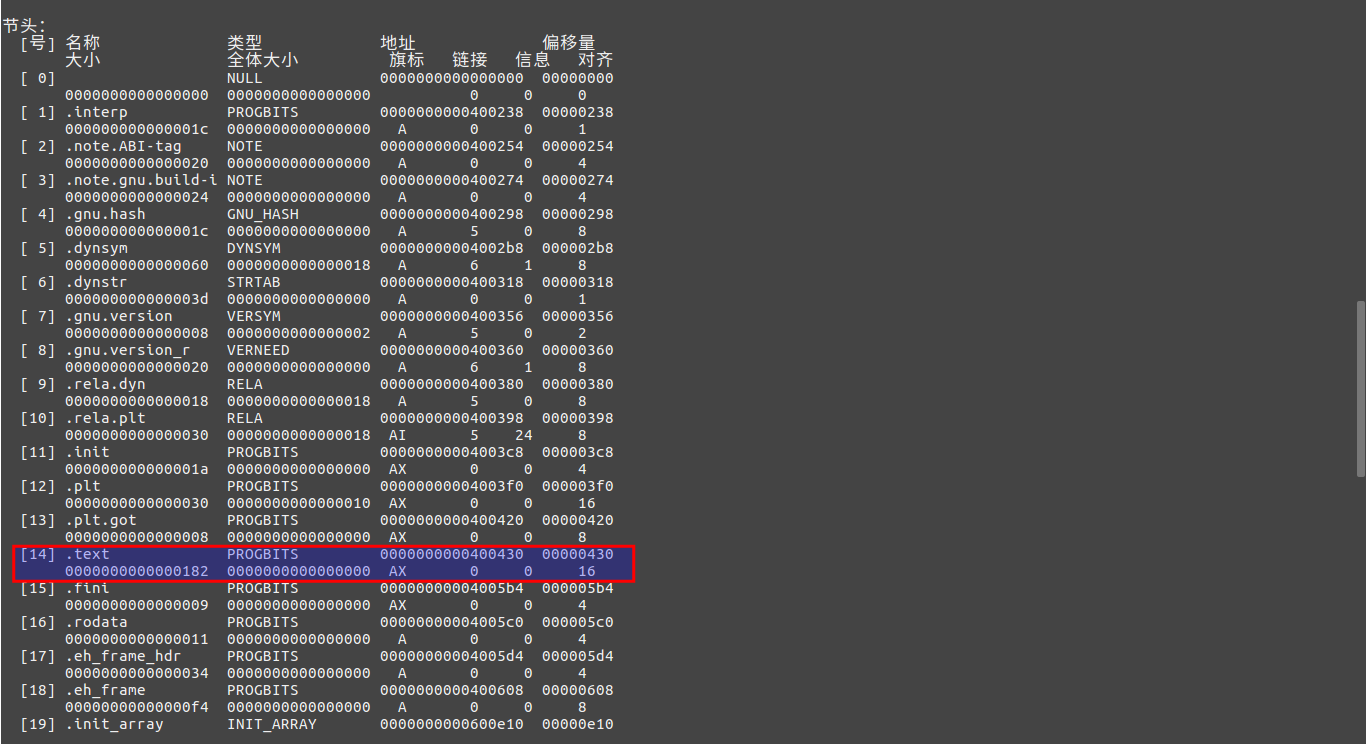


1. 使用 objdump -D hello\_world 查看生成的 hello\_world 的机器码等详细信息

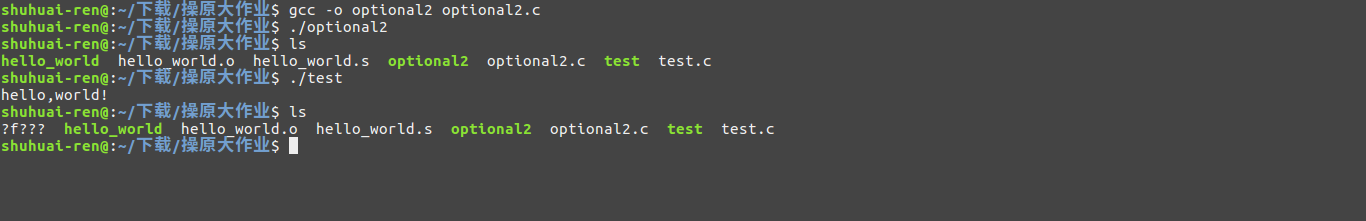


1. 使用 gcc -o test test.c 生成用来测试的 elf 可执行文件，并通过 readelf -e test 来查看程序入口地址等详细信息

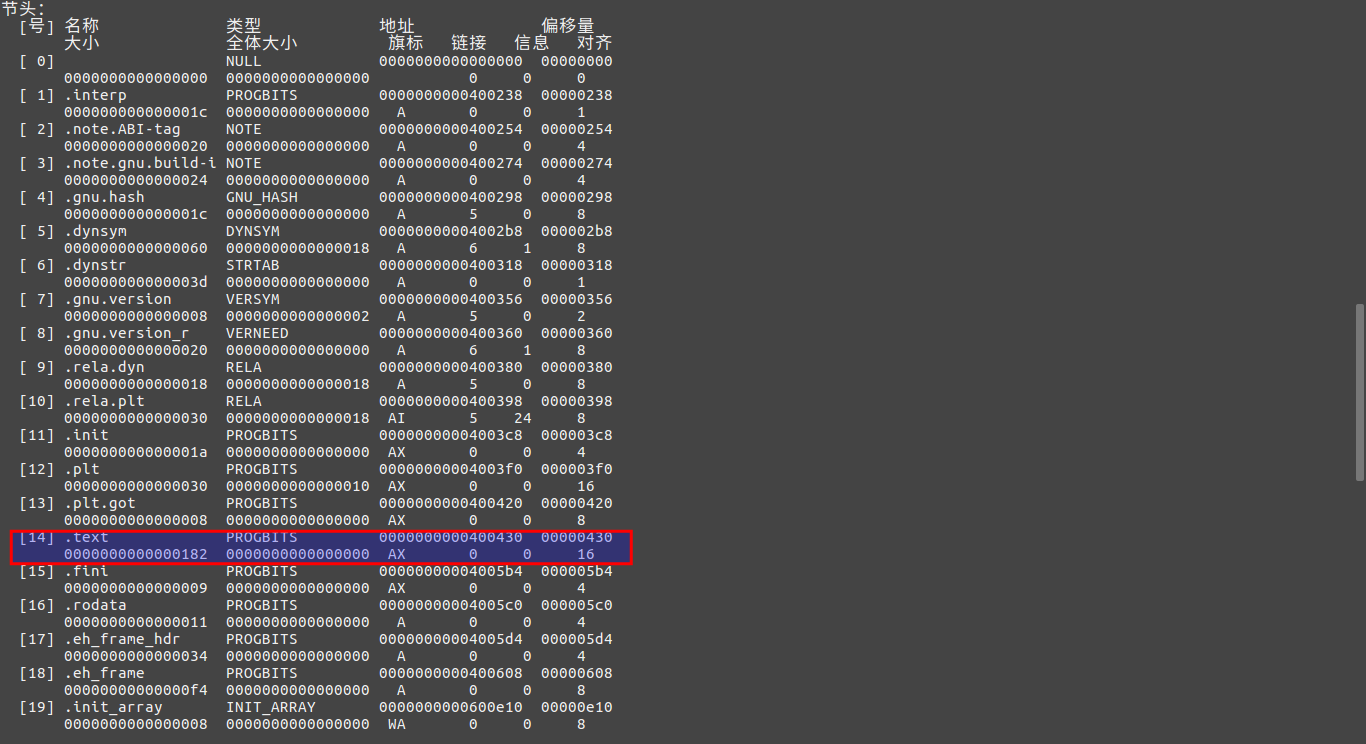


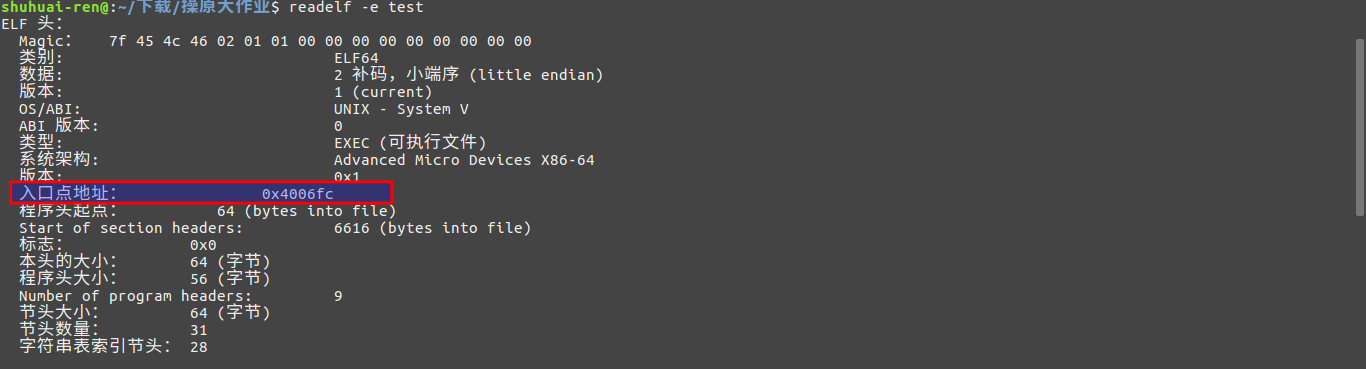


1. 使用 gcc -o optional2 optional2.c 生成修改 elf 文件的程序，然后先运行 ./optional2，再运行 ./test ，就会发现同级目录下生成了一个文件



1. 接着，再使用 readelf -e test 查看此时的 test 可执行文件，会发现此时的程序入口地址已经被修改，但 text 段的入口地址并未被修改



****

1. **软件使用说明：** 运行该程序必须要在原生的 ubuntu 系统上进行，因为系统的改变，可能导致指令的机器码也会相应变化，导致出现段错误，就比如跳转指令，在 ubuntu/kali 就会报错
2. **编程参考网址：**
3. [Linux 汇编语言开发指南](https://www.ibm.com/developerworks/cn/linux/l-assembly/)
4. [linux下汇编语言开发总结](https://blog.csdn.net/q_l_s/article/details/52722667)
5. [Linux 汇编器：对比 GAS 和 NASM](https://www.ibm.com/developerworks/cn/linux/l-gas-nasm.html)
6. [自己动手写病毒—ELF文件病毒](https://blog.csdn.net/luojiafei/article/details/7225818)
7. [Linux及安全实践四——ELF文件格式分析](https://www.cnblogs.com/lxq20135309/p/5551658.html)
8. [ELF文件病毒的分析和编写](https://blog.csdn.net/luojiafei/article/details/7206063)
9. [ELF格式可执行程序的代码嵌入](ELF格式可执行程序的代码嵌入.pdf)