拆弹实验

目标

进一步掌握程序的机器级表示一章，理解程序控制、过程调用的汇编级实现，熟练掌握汇编语言程序的阅读。

内容

程序bomb是一个电子炸弹，当该程序运行时，需要按照一定的顺序输入口令，才能阻止炸弹的引爆。当输入错误的密码时，炸弹将会引爆。此时控制台将会产生如下输出，并结束程序。

```
BOOM!!!
The bomb has blown up.
```

在炸弹程序中，你需要输入多组口令，且每一组口令都正确才能够防止引爆。

目前已知的内容只有炸弹程序的二进制可执行文件bomb（目标平台为：x86-64）和bomb的main函数代码，见main.c。其他的细节均不会以c语言的方式呈现。

你的任务是：利用现有的资源以及相关的工具猜出炸弹的全部口令，并输入至炸弹程序中，以完成最终的拆弹工作。

一些细节

默认情况下，炸弹的口令是从从标准输入（stdin）中读入，为了简化拆弹时口令的重复输入的问题，炸弹程序支持从文本文件中输入口令。如果口令存储在password.txt文件中，通过在控制台输入以下命令可以将口令一次输入至bomb中。

```
linux> ./bomb password.txt
```

在口令文件中，每一行文本表示输入至炸弹程序的一组口令。在bomb.c函数中，read_line函数的功能是从标准输入或口令文件中加载一组口令。phase_x 和 phase_defused函数用于检测输入的口令是否正确，如果输入错误则炸弹引爆，并退出程序。如果当前口令正确，则继续从输入中读取下一组口令并进行验证，直至全部口令验证通过。

提示

1. 关于工具

以下一些工具将会为你的拆弹过程提供帮助：

- gdb

gdb 是 GNU 工具集中的调试工具，是一个基于命令行界面的调试工具。在此工具中可以为程序设置断点、查看寄存器内存的状态、返回编程序等功能。gdb是其他Linux C/C++ IDE程序调试功能的实现基础，这些IDE的调试功能都是基于gdb的图形化界面封装。gdb的具体使用方法可以从网络中查阅资料获得。

- objdump
  是一个简单的反汇编工具，可以实现基本的反汇编功能。常用指令如下：
  a. 对可执行程序execfile中符号表的提取和显示

```
linux> objdump -t execfile
```

  b. 显示exefile的反汇编程序

```
linux> objdump -d execfile
```

  由于反汇编程序比较长，上面的命令会直接把程序输出至控制台不便查看。因此在执行上述指令时，在Linux控制台中可以使用重定向命令，将输出重定向至文件中查看，例如将反汇编结果输出至execfile.s文件中可以使用下面的命令：

```
linux> objdump -d execfile > execfile.s
```

  c. strings

  Linux提供了一个strings命令，可以将文件中所有的可见字符打印出来，例如：

```
linux> strings bomb
```

2. 致学霸

本题中还包含一个隐藏关卡，找到并解决之。

作业提交要求

在平台中编辑实验报告并提交。在报告中需要逐个给出你最终找到的拆弹口令，并说明找到的每一组拆弹口令的分析过程。