二进制炸弹实验

注意：由于慕课学期较短，本学期只要求完成本实验的前 4 阶段（即阶段 0-3）。实验数据中包含的剩余阶段可供进一步练习，以加深对相关理论课程知识的理解和掌握。

一、实验介绍

本实验要求运用课程所学程序与数据的机器表示方面知识，拆除一个二进制炸弹（“binary bombs”，下文简称为炸弹）程序中设置的多个关卡，在该过程中增强对程序与数据的机器级表示、汇编语言、调试器和逆向工程等方面知识与技能的掌握。

一个二进制炸弹是一个 Linux 可执行程序，包含了多个阶段（又称为层次、关卡）。在炸弹程序运行的每个阶段要求输入一个特定字符串，如果该输入字符串符合程序的要求，该阶段的炸弹就被拆除了，否则炸弹“爆炸——即打印输出“BOOM!!!”的提示。

每个炸弹阶段考察了程序与数据的机器级表示的不同方面，难度逐级递增：

- 阶段 0：字符串比较
- 阶段 1：浮点数表示
- 阶段 2：循环
- 阶段 3：条件/分支
- 阶段 4：递归调用和栈
- 阶段 5：指针
- 阶段 6：链表/指针/结构
- 另外还有一个隐藏阶段作为阶段 7，只有在阶段 4 的拆解字符串后附加一特定字符串后，才能在实验最后进入隐藏阶段。

实验的目标是拆除尽可能多的炸弹关卡——分析获得尽可能多的正确拆解字符串。

- 实验环境：Linux i386
- 实验语言：汇编

二、实验数据

在本实验中，每位学生应从下列链接下载包含本实验相关文件的一个 tar 文件：

http://cs.nju.edu.cn/sufeng/course/mooc/0809NJU064_bomblab.tar

可在 Linux 实验环境中使用命令“**tar xvf 0809NJU064_bomblab.tar**”将其中包含的文件提取到当前目录中。该 tar 文件中包含如下实验所需文件：

- bomb：二进制炸弹可执行程序
- bomb.c：包含 bomb 程序中 main 函数的 C 语言程序框架

运行二进制炸弹可执行程序 bomb 需要指定 0 或 1 个命令行参数（详见 bomb.c 源文件中的 main()函数）：

- 如果运行时不指定参数，则程序打印出欢迎信息后，期望你按行输入每一阶段用来拆除炸弹的字符串，程序根据输入字符串决定是通过相应阶段还是引爆炸弹导致该阶段任务失败。
- 也可将拆除每一炸弹阶段的字符串按行（一行一个字符串）记录在一个文本文件（必需采用 Unix/Linux 换行格式）中（即实验结果提交文件的形式），然后将该文件作为

运行二进制炸弹程序时的唯一命令行参数，程序将依次检查对应每一阶段的字符串来决定炸弹拆除成败。

注意：如果拆解字符串（来自命令行输入或文本文件）不正确导致相应炸弹阶段被引爆，程序在输出炸弹爆炸的提示文字“BOOM!!!”后，将进入下一阶段的字符串检查（等待命令行输入或读取文件下一行）而不会终止程序的运行。因此，如果暂时未能正确获得某阶段的拆解字符串，可用任意非空字符串（即不同于空格、制表、换行的一个以上字符）临时作为拆解字符串，从而在引爆相应炸弹阶段后，跳到以后阶段继续开展实验。

三、实验结果提交

注意 1：因慕课平台对文件提交和处理的功能限制，**本实验无法采取课程视频最后说明的上传实验结果文件的提交与评分形式，因而改为在相应单元测验的相应填空题中输入所指定实验阶段的拆解字符串**。仅当所输入作为填空题答案的字符串与正确拆解字符串在**字符个数和字符内容（包括大小写）上完全一致、精确匹配**时，相应填空题才得分。

注意 2：有些实验阶段要求输入由 1 个或多个数字组成的拆解字符串，由于同一取值的数字可能存在多种表示形式，考虑到**填空题只依据所填入字符串与答案字符串是否完全一致来给分**，为避免因表示形式不一致而失分，现对此类**拆解字符串的正确组织形式**说明如下：

1）第一个数字应从拆解字符串的首字符位置开始，最后一个数字应作为拆解字符串的结尾——拆解字符串前、后不能再有其它字符；
2）如果要求输入多个数字，则在拆解字符串中，相邻两个数字以**单个空格**加以分隔；
3）每个数字应以**十进制整数形式**输入，除**负数前加负号**（**正数前无需且不允许加正号**）和上述用以分隔的空格外，数字前后不能出现任何其它字符，且数字本身只由 0-9 字符组成。
4）示例 1：包含一个负数(-1234567890)和一个正数(9876543210)的拆解字符串形式如下（注意两个数字间有**单个空格**用以分隔）：

```
-1234567890 9876543210
```

示例 2：包含多个数字的拆解字符串形式类似如下（注意每两个相邻数字间均由**单个空格**加以分隔）：

```
153 77 39 20 11 6 4 3
```

注意 3：由于本慕课所有测验对每道填空题统一限定为 1 分，为适应该约定，**假设某一实验阶段占实验总分 10 分中的 3 分，则有可能(但不一定——具体见测验内容)测验中会包含 3 道完全一样的填空题，每题均要求输入该实验阶段的同一拆解字符串**，每个正确输入的拆解字符串可得 1 分。

四、实验工具

为完成二进制炸弹拆除任务，可使用 objdump 工具程序反汇编可执行炸弹程序，并使用 gdb 工具程序单步跟踪每一阶段的机器指令，从中理解每一指令的行为和作用，进而推断拆除炸弹所需的目标字符串的内容组成。例如，可在每一阶段的起始指令前和引爆炸弹的函数调用指令前设置断点。

下面简要说明完成本实验所需要的一些实验工具：

GDB

为从二进制炸弹可执行程序” bomb “中找出触发炸弹爆炸的条件，可使用 GDB 程序帮助对程序的分析。GDB 是 GNU 开源组织发布的一个强大的交互式程序调试工具。一般来说，

GDB 可帮助完成以下几方面的调试工作（更详细描述可参看 GDB 文档和相关资料）：

- 装载、启动被调试的程序
- 使被调试程序在指定的调试断点处中断执行，以方便查看程序变量、寄存器、栈内容等程序运行的现场数据
- 动态改变程序的执行环境，如修改变量的值

objdump

- –t 选项：打印指定二进制程序的符号表，其中包含了程序中的函数、全局变量的名称和存储地址
- –d 选项：对二进制程序中的机器指令代码进行反汇编。通过分析汇编源代码可以发现 bomb 程序是如何运行的

strings

该命令显示二进制程序中的所有可打印字符串