## 实验要求

1. 本次实验为熟悉汇编程序及其调试方法的实验；
2. 实验内容包含2个文件bomb（可执行文件）和bomb.c（c源文件）；
3. 使用gdb工具反汇编出汇编代码，结合c语言文件找到每个关卡的入口函数；
4. 分析汇编代码，找到在每个phase程序段中，引导程序跳转到“explode\_bomb”程序段的地方，并分析其成功跳转的条件，以此为突破口寻找应该在命令行输入何种字符通关。
5. 本实验一共有7个关卡，包括6个普通关卡和1个隐藏关卡。要求至少通过6个普通关卡。
6. 第9周收实验报告。

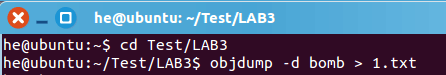
## 实验目的

熟悉汇编程序，学习gdb调试工具，熟悉并掌握函数调用过程中的栈帧结构的变化。

## 实验内容

下面以第一个关卡作为讲解，其余5个关卡请自行分析汇编代码，并通过。

1. 反汇编得出汇编代码并放入1.txt文件中以便查看。



1. 查看汇编代码，找到与关卡设置有关的几段代码并查看分析，以下截取主要的有用信息，完整代码详见txt文件。
2. **Phase\_1**

* 汇编代码：

08048f61 <phase\_1>:

8048f61: 55 push %ebp

8048f62: 89 e5 mov %esp,%ebp

8048f64: 83 ec 18 sub $0x18,%esp

8048f67: c7 44 24 04 5c a1 04 movl $0x804a15c,0x4(%esp)

8048f6e: 08

8048f6f: 8b 45 08 mov 0x8(%ebp),%eax

8048f72: 89 04 24 mov %eax,(%esp)

8048f75: e8 31 00 00 00 call 8048fab <strings\_not\_equal>

8048f7a: 85 c0 test %eax,%eax //b2

8048f7c: 74 05 je 8048f83 <phase\_1+0x22>

8048f7e: e8 4e 01 00 00 call 80490d1 <explode\_bomb>

8048f83: c9 leave

8048f84: c3 ret

8048f85: 90 nop

8048f86: 90 nop

8048f87: 90 nop

8048f88: 90 nop

8048f89: 90 nop

8048f8a: 90 nop

8048f8b: 90 nop

8048f8c: 90 nop

8048f8d: 90 nop

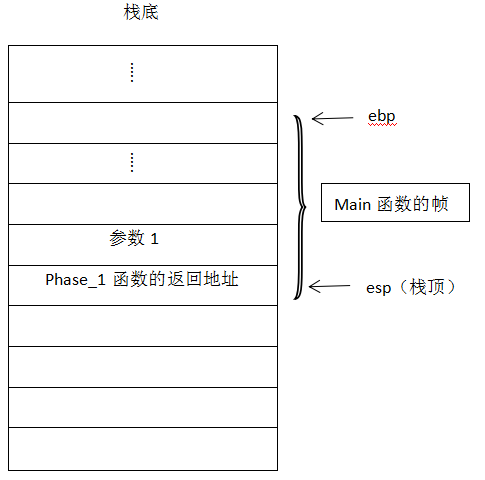
8048f8e: 90 nop

8048f8f: 90 nop

* 实验思路：仔细阅读汇编代码，逐步执行汇编代码，并绘制关键的堆栈的变化过程。

相关堆栈的知识点请阅读课本P149~150的内容。

➀ 执行phase\_1 之前的堆栈状态：



其中，每一格代表4个地址单元。

参数1是自己输入的一串字符串的首地址(详情请参考main函数部分的汇编代码)。

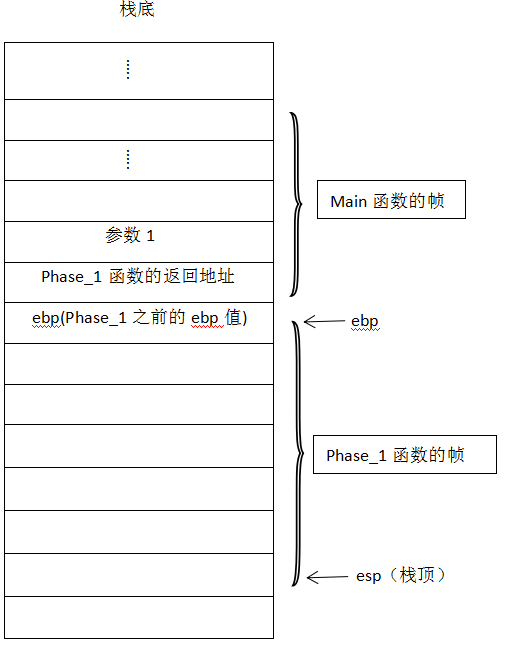
➁ 执行下述代码

**8048f61: 55 push %ebp**

**8048f62: 89 e5 mov %esp,%ebp**

**8048f64: 83 ec 18 sub $0x18,%esp**

其栈的结构改变如下：

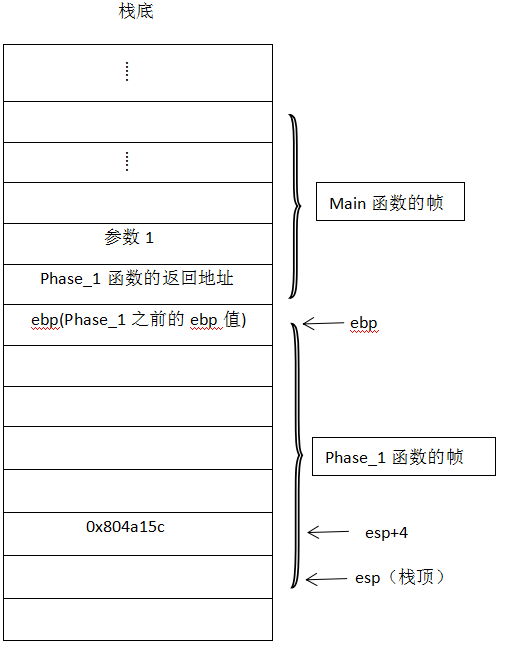


➂ 执行下述代码

8048f67: c7 44 24 04 5c a1 04 movl $0x804a15c,0x4(%esp)

8048f6e: 08

栈结构变为：

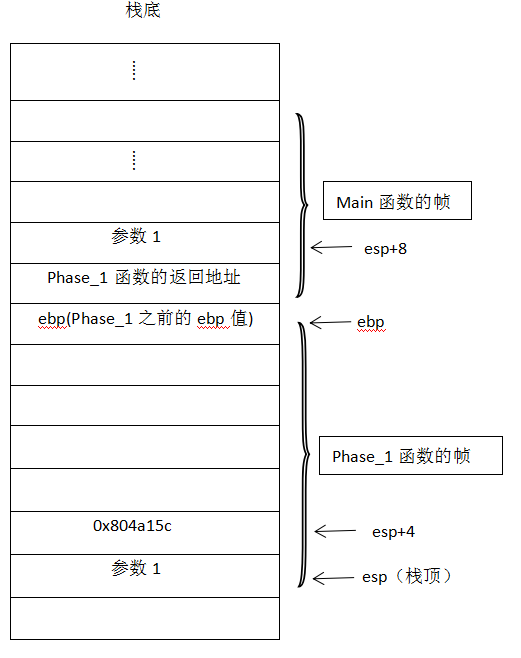


④执行下述代码

8048f6f: 8b 45 08 mov 0x8(%ebp),%eax

8048f72: 89 04 24 mov %eax,(%esp)

栈结构变为：



⑤执行下述代码

8048f75: e8 31 00 00 00 call 8048fab <strings\_not\_equal>

此行代码表示调用strings\_not\_equal函数，该函数在地址为8048fab处。（可以自行查看对应位置处的汇编代码）

该函数有两个参数，函数体是比较两个地址参数对应的字符串是否相等。

其中上一步骤中的esp与esp+4的地址，即为strings\_not\_equal函数的两个参数。

该函数有一个返回值，若两个字符串相等，则返回0，否则返回1。

执行此句的过程中会生成strings\_not\_equal函数对应的栈帧，函数调用完毕后，栈恢复到上一步④中的状态。

⑥8048f7a: 85 c0 test %eax,%eax //b2

8048f7c: 74 05 je 8048f83 <phase\_1+0x22>

8048f7e: e8 4e 01 00 00 call 80490d1 <explode\_bomb>

TEST和AND指令一样,实现“与”运算。不一样的就是,TEST不修改寄存器的值,只影响标志位。

eax中保存着函数strings\_not\_equal的返回值。

Je指令：零标志ZF=1时，则跳转。

因此，上述3条指令的作用便是：若eax的值是0，则跳过炸弹，否则调用爆炸函数。

此步骤的汇编代码不改变栈的状态。

⑦执行下述代码：

8048f83: c9 leave

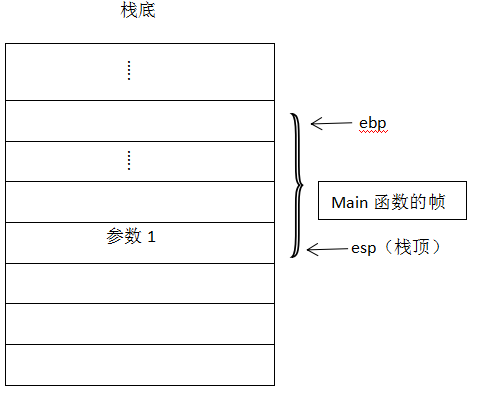
8048f84: c3 ret

leave指令等价于 movl %ebp,%esp

popl %ebp

ret指令相当于 popl %eip操作，具体内容请参考课本P150~151页。

执行上述代码后，堆栈变为：



* 破解关卡：

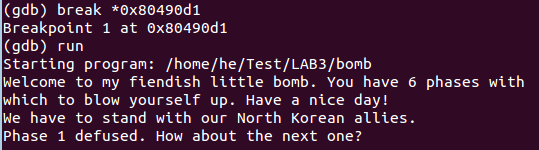
综上所述的过程可知，我们只需要查看0x804a15c地址中保存的字符串的内容，然后自己输入一串与其相同的字符串，即可安全通过phase\_1。

使用命令“gdb bomb”进入调试状态，使用命令“x/s 0x804a15c”查看0x804a15c处的内容；



因此，可得phase\_1的通关密码为字符串："We have to stand with our North Korean allies."；

* 实验结果



Phase\_1通关成功！