实验 1

1.实验现象

```
yaaaome@yaaaome-virtual-machine:~/workspace/lab05$ ./overflow1
why u r here?!
```

2.

(1)buf 的开始地址是 0xbfffef20

(2)buf[2]开始地址是 0xbfffef28

3.原代码中并没有调用 where_here 函数，但是结果却显示了该函数的输出，原因是数组 buf 数组越界，导致函数返回地址被改变，结果调用了 where_here 函数

实验 2.

1. 连续输入 16 个字符出现 segmentation fault
2. 内存范围内全部覆盖为输入的字符串
3. 0xbfffef30-0xbfffef37 无，0xbfffef38-0xbfffef3b 是%ebx 的值，0xbfffef3c-0xbfffef3f 是%eax 的值，0xbfffef40-0xbfffef43 是返回地址，剩下的是 caller 中的状态
4. Fgets 不会发生栈溢出，因为 fgets 有一个参数限制了最大读入字节数
5. 栈随机化的思想是栈的位置在每次程序运行的时候都有变化
   栈破坏检测的思想是插入金丝雀值，在函数返回前检查该值有无变化

实验 3

1. 现象

```
yaaaome@yaaaome-virtual-machine:~/workspace/lab05$ ./overflow3 20
malloc 80 bytes
loop time: 20[0x14]
yaaaome@yaaaome-virtual-machine:~/workspace/lab05$ ./overflow3 1073741824
malloc 0 bytes
loop time: 1073741824[0x40000000]
段错误 (核心已转储)
```

2. 范围是-2^31~2^31-1

   第一次 20*4=80，所有分配了 80bytes

   第二次是 1073741824*4=2^32=0,高位溢出，所有分配 0bytes

3. 实验现象：第一次分配 80 个 bytes，，所以循环 20 次
   第二次分配 0bytes，所以发生段错误