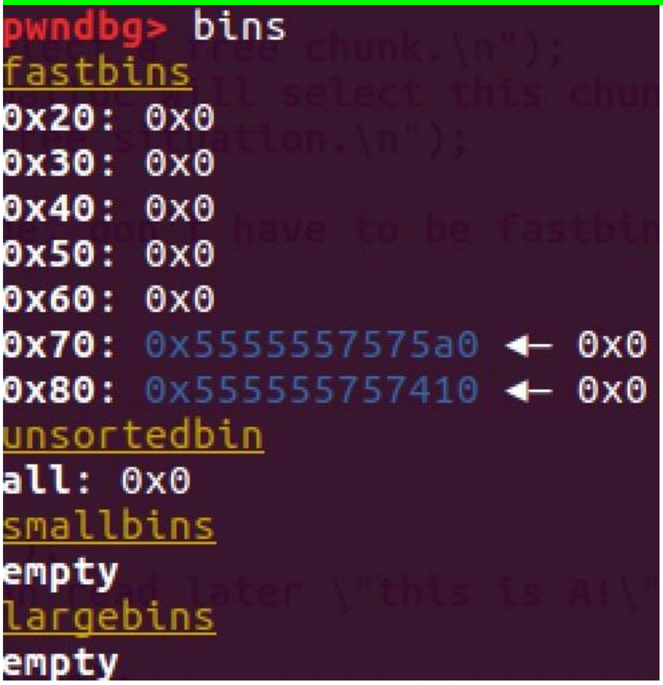
1.first fit

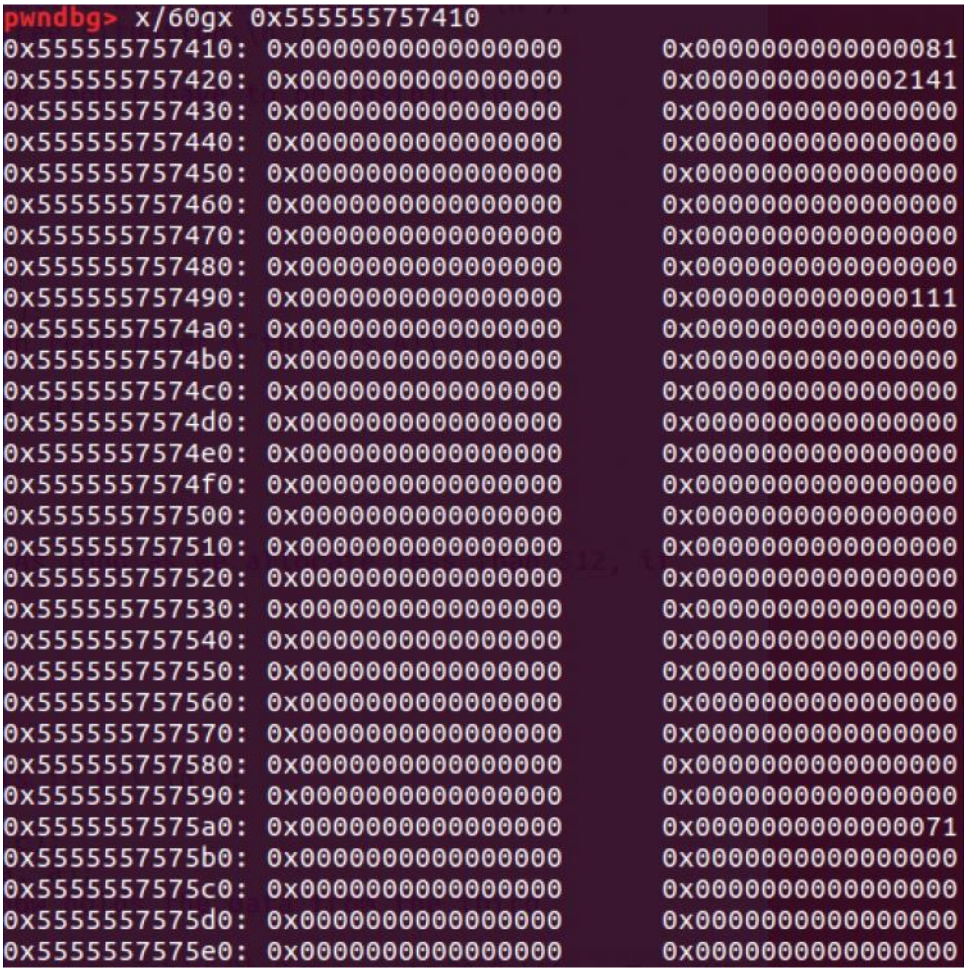
2017年10月12日　　16:53

第一个代码非常简单
a，b先各malloc 512 256字节的空间，然后往a里存“this is A”
然后把a free掉，然后申请一个比a小一点的空间c malloc（500）
这样，c肯定和a的地址是一样的
然后往c里存 this is C
然后读取c指针的字符串是this is C
读取a指针的字符串是this is C
说明这俩是重叠的。

值得注意的是这里malloc的空间都比较大，如果是fastbin这种大小，且a c的空间相差比较大，可能会不重叠？试验下吧，目前我不清楚，估计是不同的chunk
测试代码为first_fit_test1.c，验证了上述结论。将a的大小改为0x70，这样一来它的实际size=0x80,出现在fastbin中。
c的大小为0x60，实际size为0x70。然后c和a就不会重叠，c会在b的下方再开一块内存。且free c后也可以看见，c出现在了0x70的fastbin处，a在0x80的fastbin处。

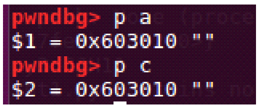


```
pwndbg> bins
fastbins
0x20: 0x0
0x30: 0x0
0x40: 0x0
0x50: 0x0
0x60: 0x0
0x70: 0x5555557575a0 ◂— 0x0
0x80: 0x555555757410 ◂— 0x0
unsortedbin
all: 0x0
smallbins
empty
largebins
empty
```



```
pwndbg> x/60gx 0x555555757410
0x555555757410: 0x0000000000000000      0x0000000000000081
0x555555757420: 0x0000000000000000      0x0000000000002141
0x555555757430: 0x0000000000000000      0x0000000000000000
0x555555757440: 0x0000000000000000      0x0000000000000000
0x555555757450: 0x0000000000000000      0x0000000000000000
0x555555757460: 0x0000000000000000      0x0000000000000000
0x555555757470: 0x0000000000000000      0x0000000000000000
0x555555757480: 0x0000000000000000      0x0000000000000000
0x555555757490: 0x0000000000000000      0x0000000000000111
0x5555557574a0: 0x0000000000000000      0x0000000000000000
0x5555557574b0: 0x0000000000000000      0x0000000000000000
0x5555557574c0: 0x0000000000000000      0x0000000000000000
0x5555557574d0: 0x0000000000000000      0x0000000000000000
0x5555557574e0: 0x0000000000000000      0x0000000000000000
0x5555557574f0: 0x0000000000000000      0x0000000000000000
0x555555757500: 0x0000000000000000      0x0000000000000000
0x555555757510: 0x0000000000000000      0x0000000000000000
0x555555757520: 0x0000000000000000      0x0000000000000000
0x555555757530: 0x0000000000000000      0x0000000000000000
0x555555757540: 0x0000000000000000      0x0000000000000000
0x555555757550: 0x0000000000000000      0x0000000000000000
0x555555757560: 0x0000000000000000      0x0000000000000000
0x555555757570: 0x0000000000000000      0x0000000000000000
0x555555757580: 0x0000000000000000      0x0000000000000000
0x555555757590: 0x0000000000000000      0x0000000000000000
0x5555557575a0: 0x0000000000000000      0x0000000000000071
0x5555557575b0: 0x0000000000000000      0x0000000000000000
0x5555557575c0: 0x0000000000000000      0x0000000000000000
0x5555557575d0: 0x0000000000000000      0x0000000000000000
0x5555557575e0: 0x0000000000000000      0x0000000000000000
```

测试代码first_fit_test2.c中，a和c都是malloc(0x70)，这时依然重叠。



```
pwndbg> p a
$1 = 0x603010 ""
pwndbg> p c
$2 = 0x603010 ""
```