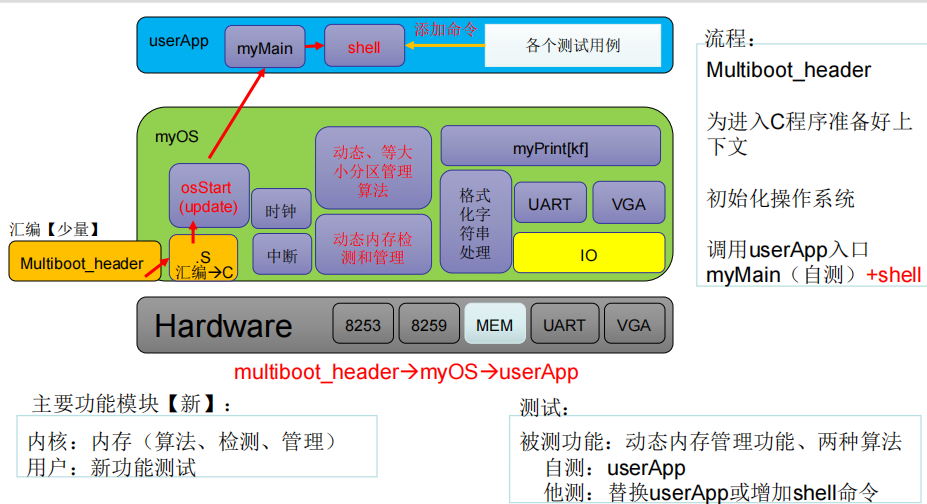
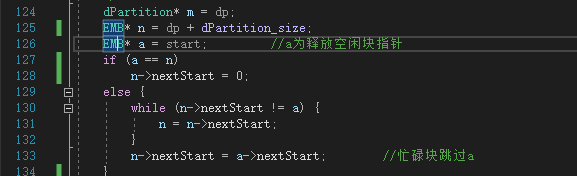
Lab4 内存管理 PB19030800 陈磊

1. 实验框架

二、基本思路：

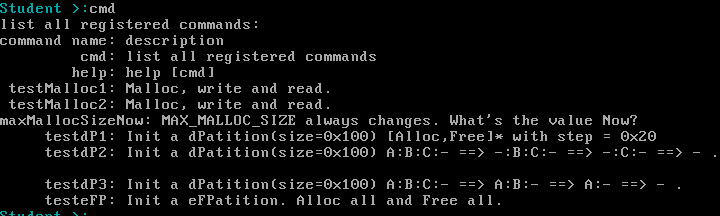
按照要求分配，只要维护空闲链表，忙碌链表不需要维护，Alloc函数会返回首地址，用首地址引用各忙碌块；



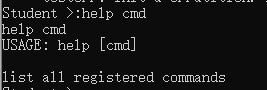
dPartition 忙碌块跳过a；后续遍历输出要用;

ePartition也类似;

1. 测试结果
2. Cmd



1. Help

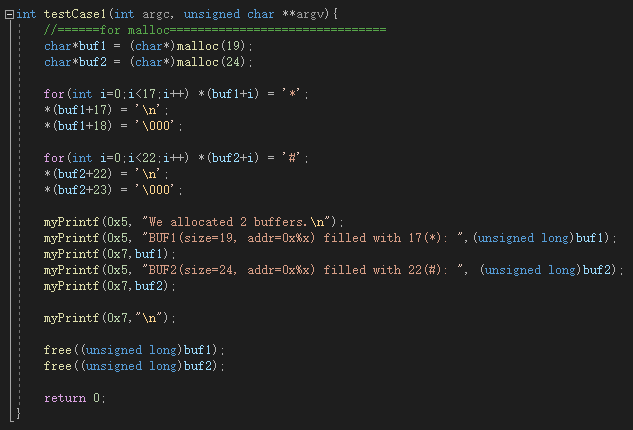


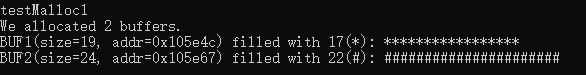
1. maxMallocSizeNow

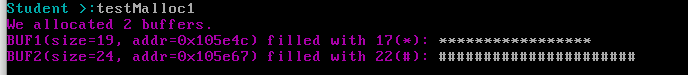


初始地址为0x1005000，与maxMalloc\_Size相加恰为0x8000000=总可用内存

1. testMalloc1





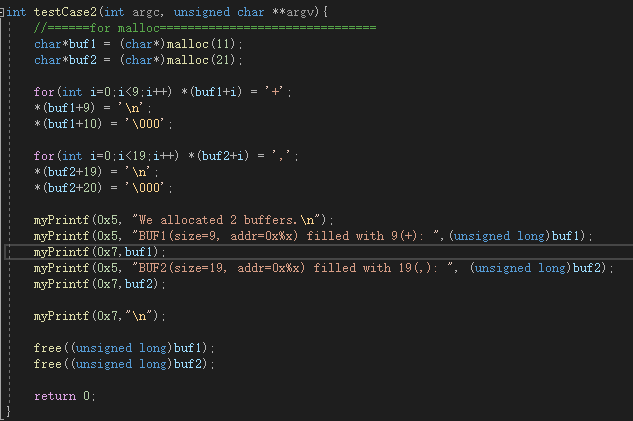


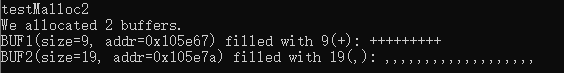
可见，分配给buf1的EMB总大小=19+EMB\_size=27B，

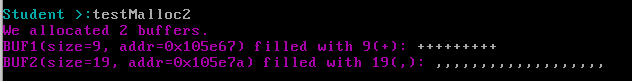
故addr(BUF2)-addr(BUF1)=0x105e67-0x105e4c=0x1b=27

BUF1/2后面的\*和#数目符合指派

1. testMalloc2





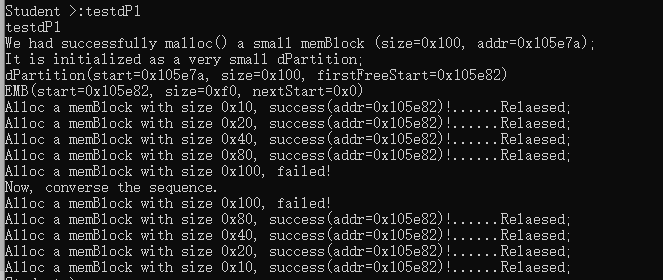


配给buf1的EMB总大小=9+2+EMB\_size=19B

addr(buf2)-addr(buf1)=0x105e7a-0x105e67=0x13=19

Buf1/2后面的\*和#数目符合指派

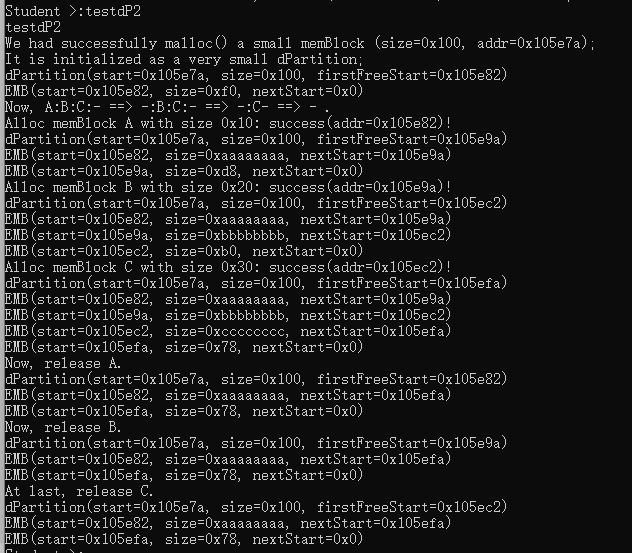
1. testdP1

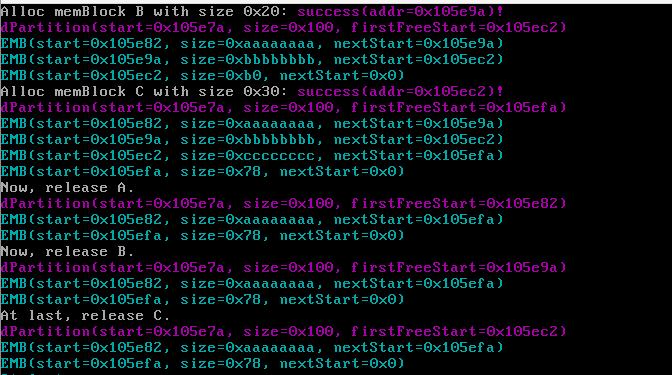


开始时分配dPartition大小为100B，

故前四次分配均满足要求，最后一次分配过大，（其实刚好可以，只是比较大小时用了<而不是<=）

1. testdP2





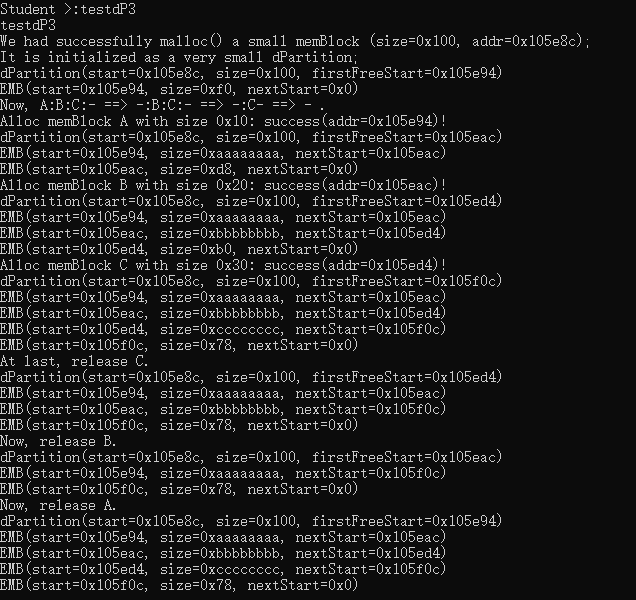
依次:sizeA=0x10=16B,实际大小=nxetstart-start=0x105e9a-0x105e82=0x18=24B

SizeB=0x20,实际大小=0xc2-0x9a=0x28(B)

sizeC=0x30,实际大小=0xfa-0xc2=0x38(B)

release有问题；

1. testdP3

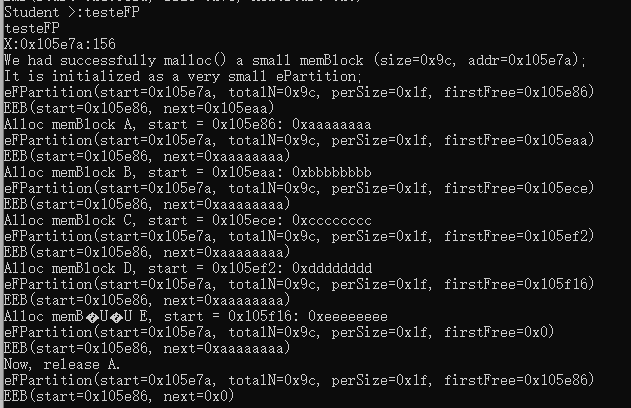


类testdP2，分配无问题;

Release B&C无误；

releaseA反而将B,C都再次分配，不解。。

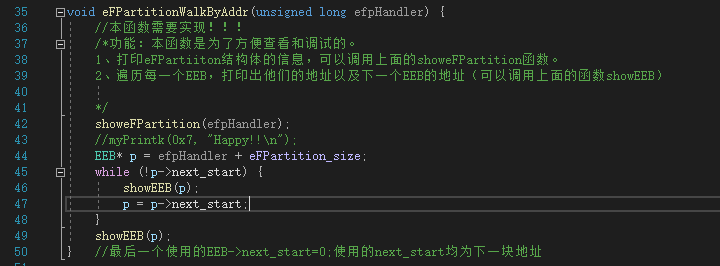
1. testeFP



walkby函数有问题,只能输出第一个数；

但可以看出随着块的分配，memBlockA/B/C/D,start:0xxxxxxxx均符合要求，

每次分配完walkby输出ePartition的各项数据包括firstFree（为下一次分配的空闲块首地址）均符合要求



实际上while语句的判断应为p->next\_start，但不知为什么那样只能初始化后walkby即停顿

解决不了。。