



用户申请32B, 有于此时的pool为空, 故在此节点向os malloc 32 * 20 * 2 = roundup(0 > 4) = 1280B, 切除一块给用户, 19个block给list3, 剩下640B作为内存池使用

总计申请量1280B, pool 640B

用户申请64B, 由于pool此时有640的余量, block num = 640 / 64 = 10, 第一个给用户, 其它的给list7

总计申请量1280B, pool 0B

申请96B, 由于此时的pool为空, 此时向os malloc 96 * 20 * 2 + Roundup(1280 > 4) = 3920B, 其中一块返回给用户, 19块给list11, 剩余2000B 给pool

总计申请量1280+3920B=5200, pool 2000B

申请88B, 定位在10号, 由于pool此时有余量, 进行切割 block num=2000 / 88 = 22, 最多给block num 20块, 分配给用户一块, 其他19块挂在list10上, pool=2000B-88*20=240B

总计申请量5200B, pool的大小240B

用户连续三次申请88B

总计申请量5200B, pool的大小240B

用户申请8B, 定位到list0, 由于此时pool有余量, 分配给list0的block num=240 / 8=30, 但最多给list0 20块, 其中19块给用户, 19块给list0, 花费8 * 20 B, 剩余80作为内存池

总计申请量5200B, pool的大小240B

用户申请104B, 定位到list12, pool有余量80B, 但是不足以分配, 于是将这个内存小碎片分配给list9, 然后向os申请 104 * 20 * 2 + roundup(5200 > 4), 一块分配给客户, 19块分配给list12, 剩余2408B作为pool

总计申请量9688, pool的大小2400B

用户申请112, 由于pool有2408B, 一个返回客户, 剩余19个给list3, pool剩余2408-112*20=168B

总计申请量9688, pool的大小168B

用户申请48, 由于pool有168, block=168 / 48=3, 第一块给用户, 剩余2块给list5, 剩余pool=168-3*48=24

总计申请量9688, pool的大小24B

用户申请72, 定位到list8, list8没有block可用, 向pool申请内存块, 但此时pool已经不能提供, 此时先将内存碎片24分配给list2, 向os malloc 72 * 20 * 2 + roundup(9688 > 4), heapsize 设为10000 此时malloc失败, 内存穷途末路, 但还有很多白色块未使用, 该如何处理? 从右寻找第一个block, alloc回收block pool, 在从右切出72返回给客户, pool此时的余量为8

总计申请量9688, pool的大小24B

用户申请72, list8无可使用block, pool为空, 往右查找, 找到list10, 找个一个空闲块分配72

总计申请量9688, pool的大小0

反思1 白色内存块比比皆是, 想要将它们合并为较大的内存块供用户使用, 技术难度极高, 指针指向太复杂! 反思2 此时内存剩余量312B, 在malloc失败后不可不可以将malloc的内存大小/2, 如此反复, 直到malloc的内存<=312, 分配给用户