



假设申请20Byte,系统调整为24Byte,初始状态内存池为空,free_list[24]存储的节点指针为NULL,配置器默认申请20个24Byte,1个返回给用户,19个存储在free_list[24].向heap申请20*24Byte空间,实际请求的是2*20*24Byte,成功后一半480Byte给free_list[24],一半480Byte继续留在内存池中.

假设用户后续请求125Byte,实际被调整为128字节,free_list[128]链表为NULL,触发请求20*128Byte行为,内存池剩余480Byte,最多能分配3*128=384字节,剩余96Byte,1个返回给用户,2个留在free_list[128].

假设用户又请求了2次125Byte,free_list[128]将剩余的两个节点分配出去,然后置空.

假设用户申请100Byte,系统调整为104字节,free_list[104]为NULL,内存池中剩余空间为96字节,剩余空间不够分配1个104Byte,将剩余空间挂载到free_list[96](由于每一块内存按照8Byte对齐且每次申请的是8的整数倍,因此最终剩余的内存一定是8的整数倍,可以挂载到free_list)
内存池空间不够需要扩展内存池到 $2*104*20+_S_round_up(2*20*24>4)=4160+64=4224Byte$
 $20*104=2080Byte$ 提供给用户和free_list[104],剩余2144Byte留在内存池.