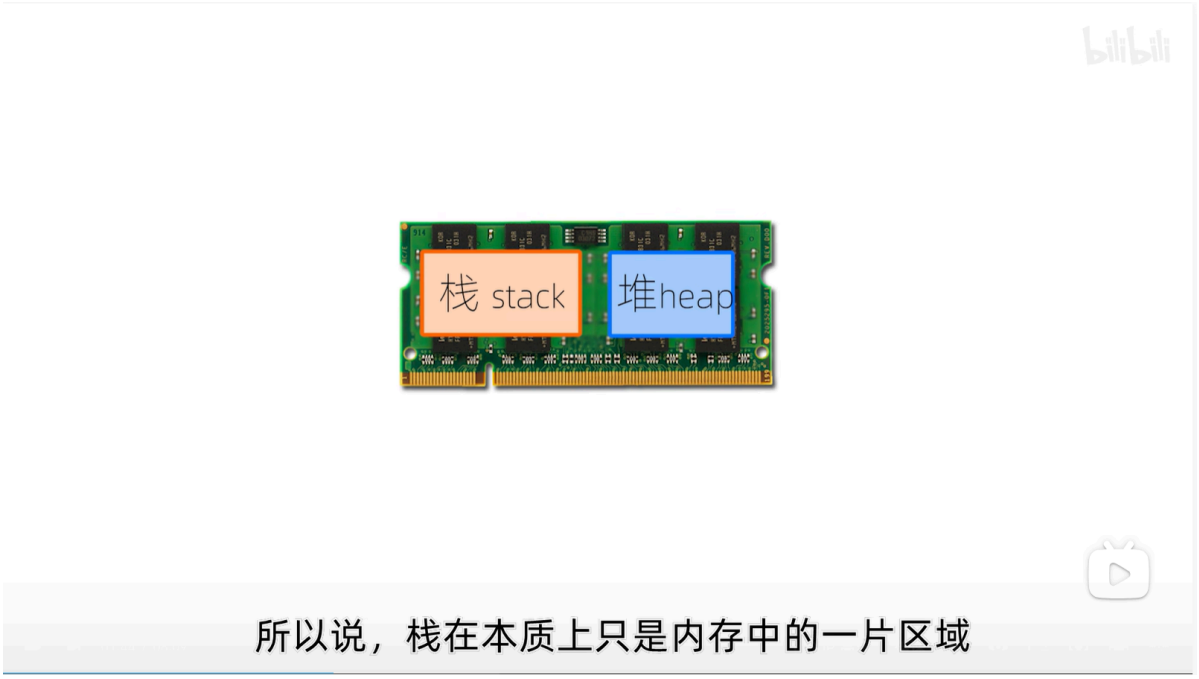
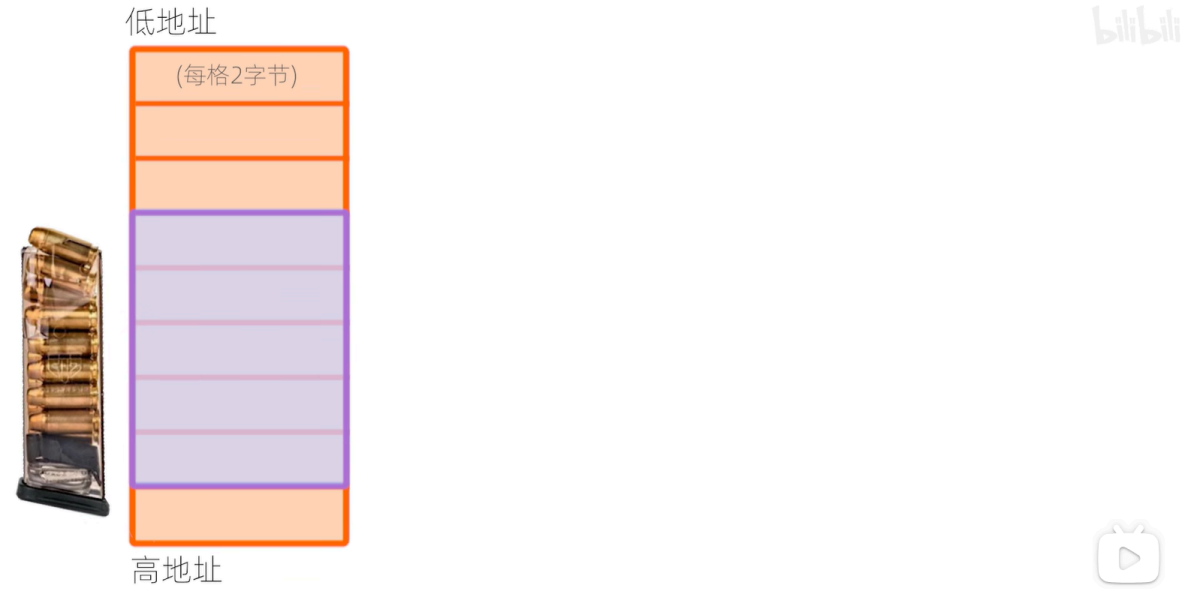
堆栈是个什么

首先要从CPU说起，CPU内部有个叫***ALU***的算术逻辑模块，在进行多次计算的时候需要临时存储计算结果，所以在CPU内部设置了寄存器用于临时存放计算结果。但是碍于设计成本和空间CPU内部不可以设置过多的寄存器，所以需要一个读写速度极快的存储介质，硬盘软盘光盘都太慢了，所以最后中标的是***内存***

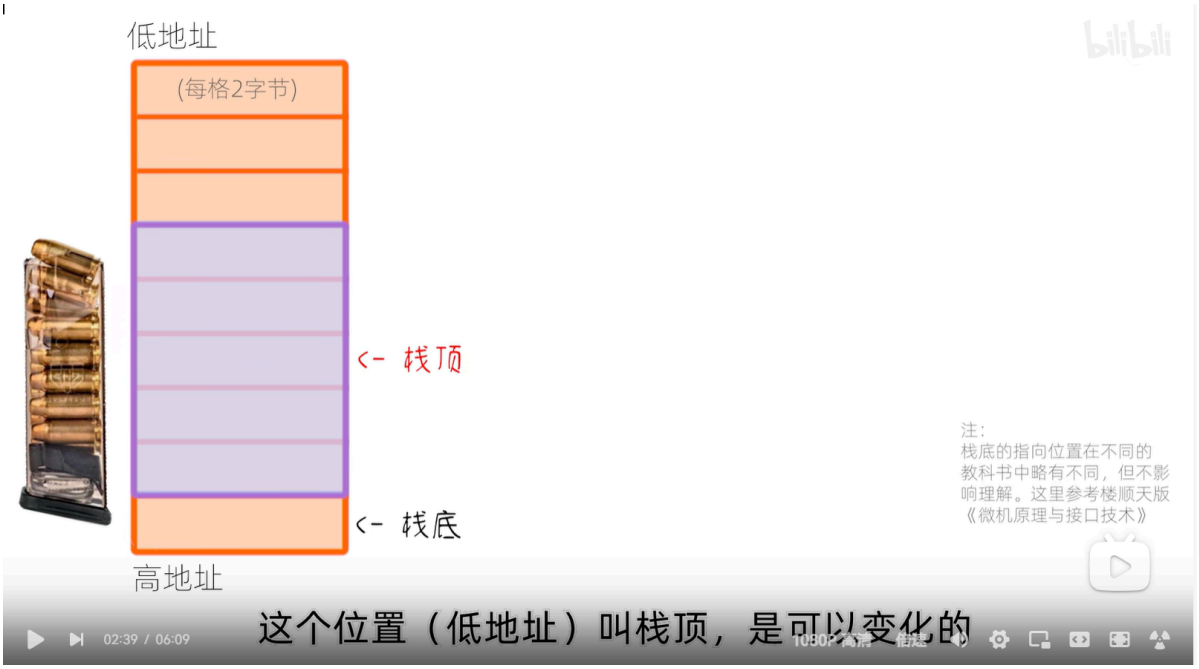


所以我们在内存中特意划出一片区域用于临时存储数据，名字叫做栈。特点在于CPU从中存取数据的方式（先进后出）类比弹夹

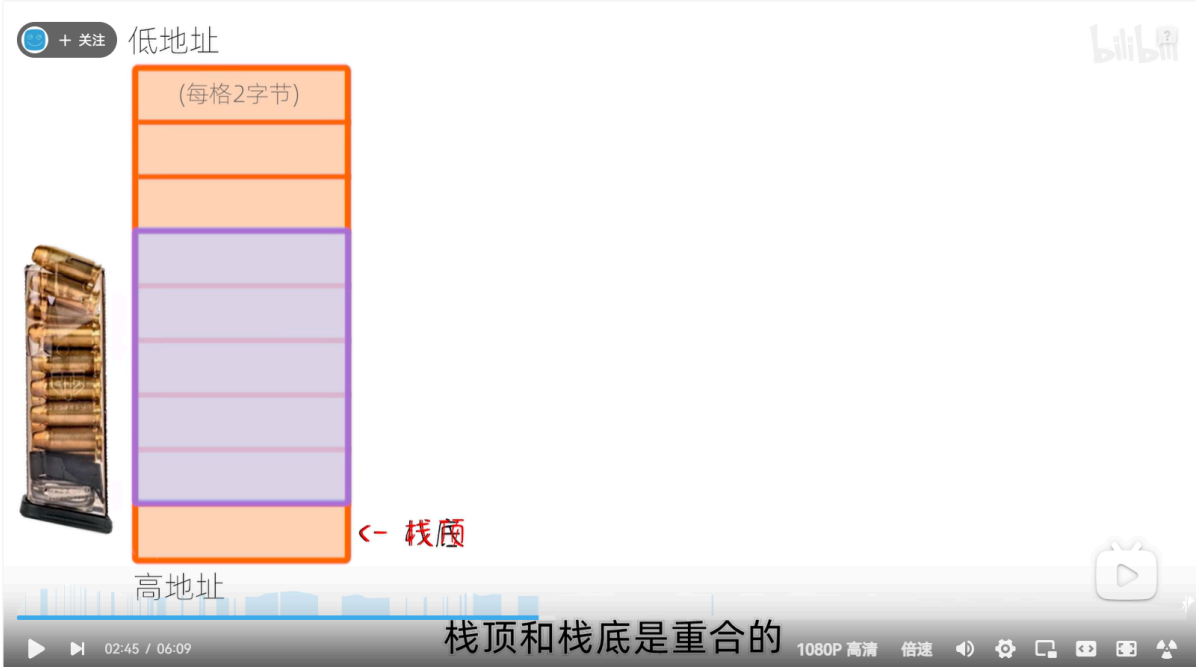
在内存里划出一段连续区域，紫色的就是堆栈



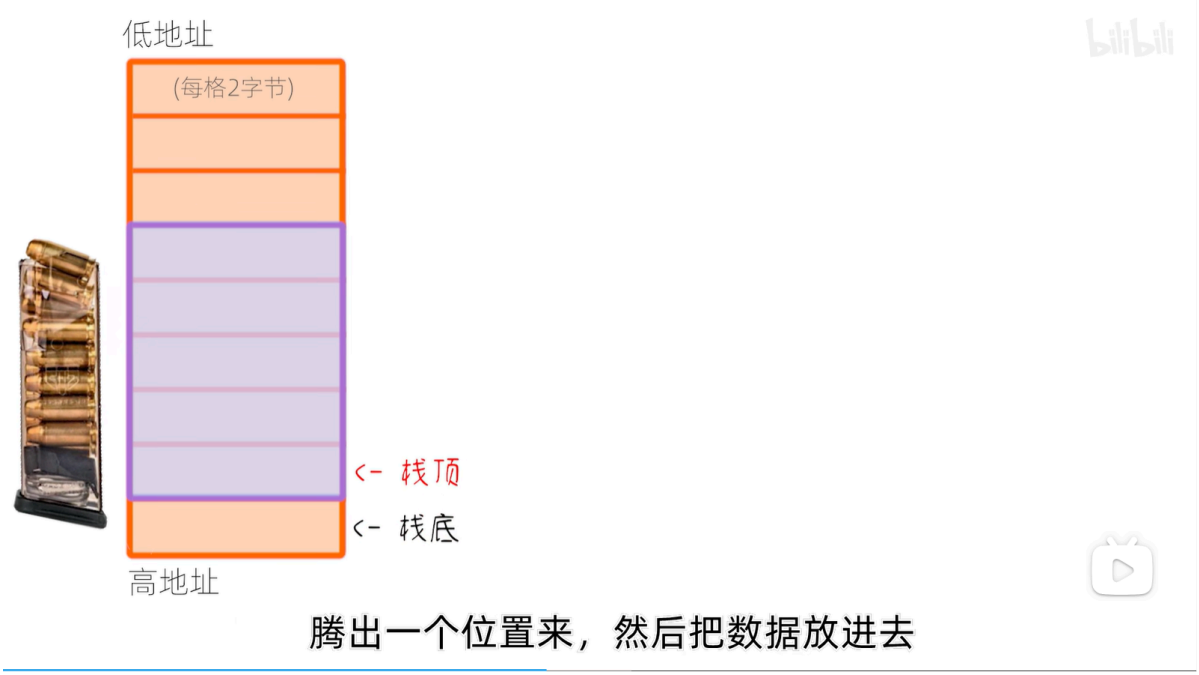
栈顶可以随意移动，栈底不动



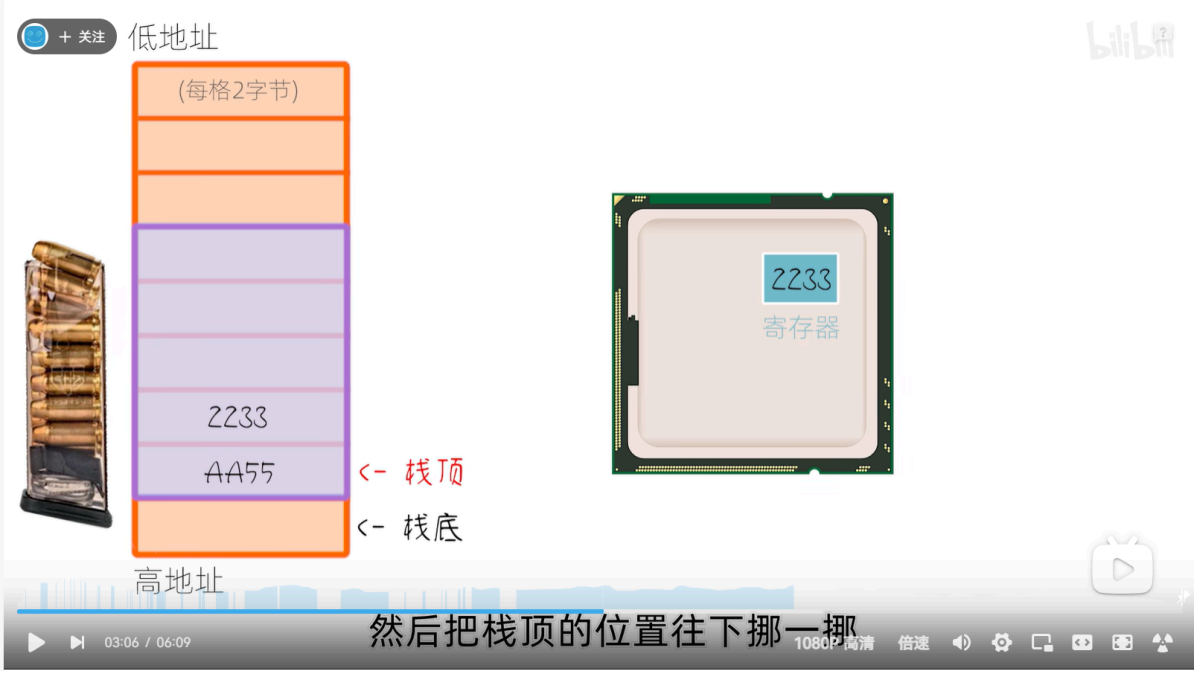
在没有数据的时候，栈底和栈顶是重合的



需要存数据的时候就让栈顶往上挪挪腾出位置（压栈/入栈），再将数据存入进去。



出栈的时候，先将数据复制到CPU中的寄存器里，然后再将栈顶的位置往下挪一挪

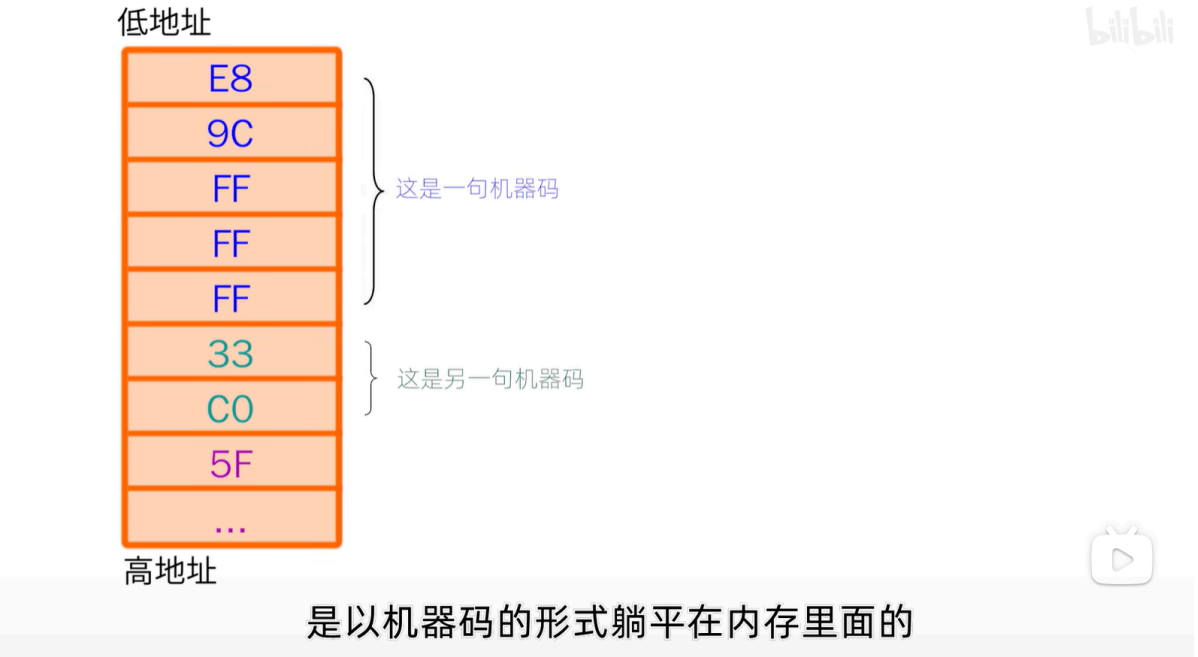


注意的是，出栈以后数据依然还是在栈中但是已经被当做垃圾了。

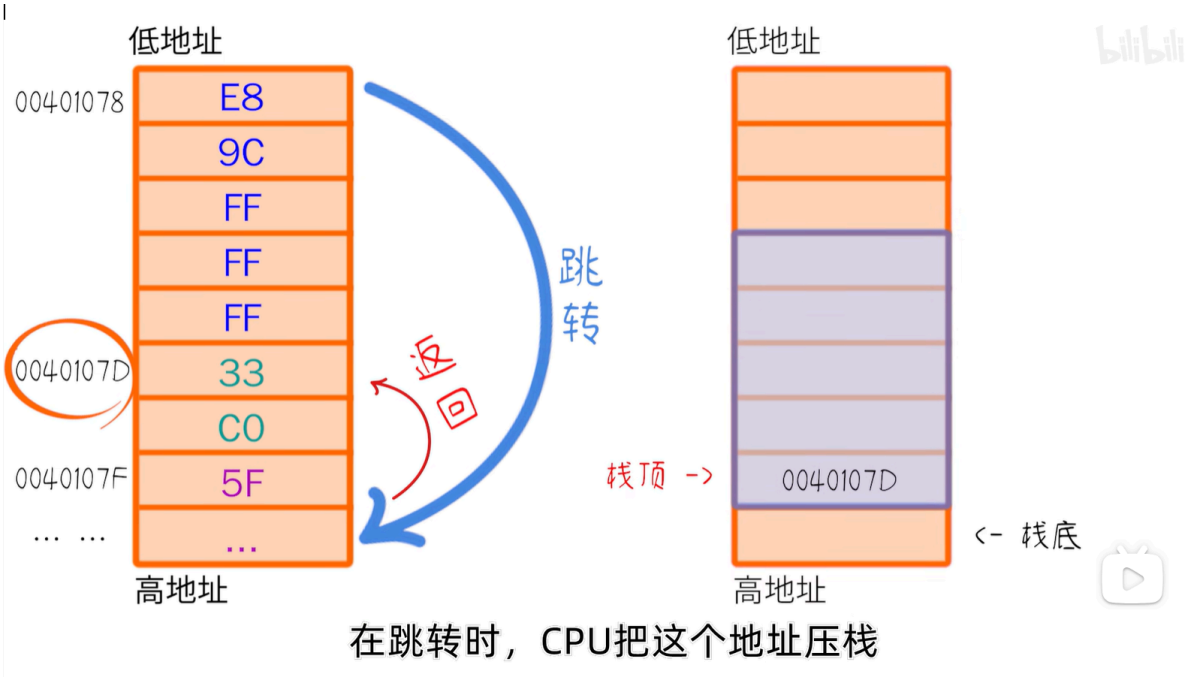
有人说我想绕过2233这个数据直接获得AA55行不行，答案是不可以

堆栈和函数的调用

程序在运行的时候是以机器码的形式躺平在内存中的

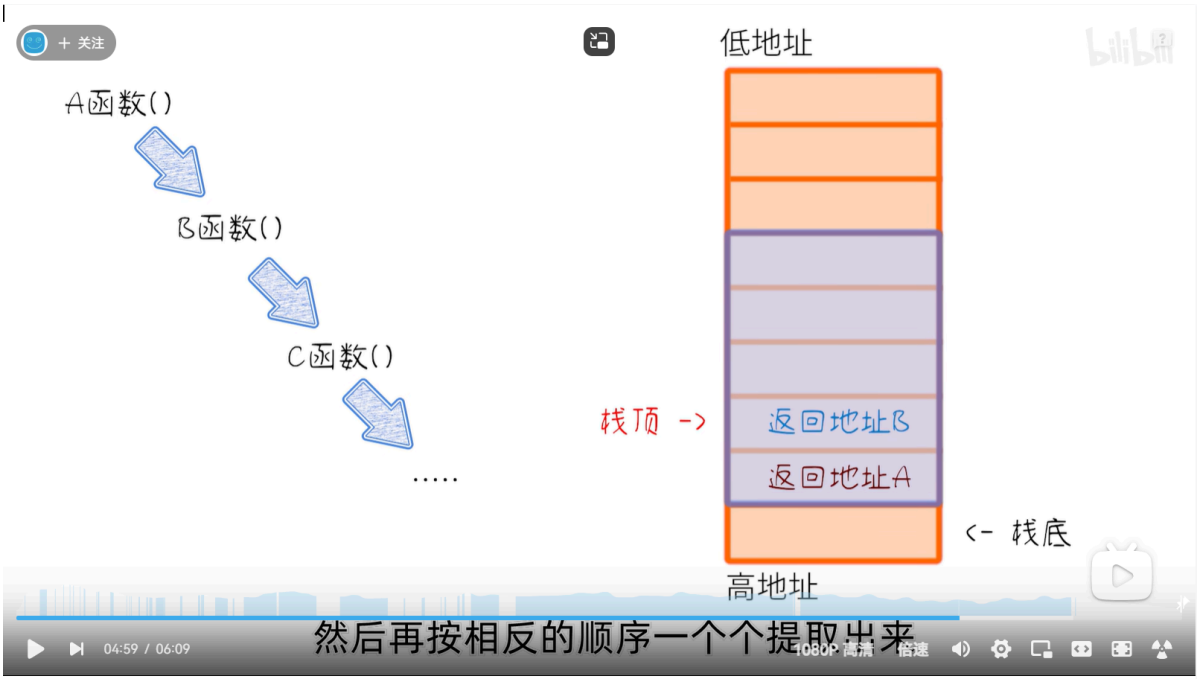


一个函数从跳转到另一个函数的时候，CPU需要记录下来返回时所需执行的代码地址

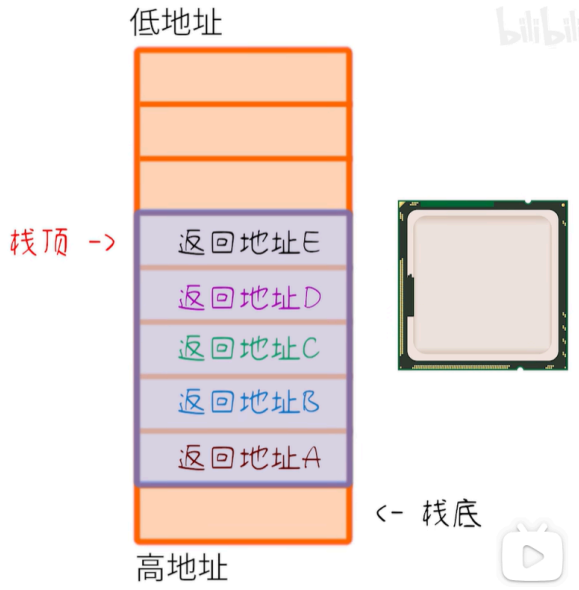


返回的时候，就会将这个地址出栈然后复制到CPU中，CPU就会根据这个地址跑到这来继续执行代码

之所以使用堆栈来存储函数调用时所返回的地址，是因为函数调用实际上是层层嵌套的，这时候就需要将返回地址按照顺序保存，再按照相反顺序一个个提取出来（假如我们只嵌套循环到c函数，那么c函数代码执行完以后我们是需要先返回B函数然后再返回A函数的）



堆栈溢出错误



这个时候再试图把数据压栈就会导致错误

如果函数无限调用而不返回，那么由于堆栈的空间有限，返回地址过多会导致栈溢出而报错。