

栈区是内存中的一块特定区域，用来存放函数中的局部变量、常量、指针等。

函数是用来复用的代码块，调用一次函数，相当于把函数体插入到调用地址后执行，按照 goto 逻辑，只要记住跳转指定地址即可。但

与条件跳转不同，函数跳转在执
行函数体后，要 goto 回 call 的位置，

~ ~ ~

而条件跳转不用返回 `goto` 的位置，后续代码本就在条件跳转的代码块后。所以在即使循环中要跳转 n 次，`cpu` 中也只要在循环尾部读取跳转地址即可；而在函数迭代调用时，跳转进函数体时必须记住跳转前位置才能 `return`，多次迭代调用，即每次 `goto` 后还未 `return` 就又 `goto`，就要有

像大量 return 用地址，cpu 中的寄存
器有限，不能这样用，所以把这
些任务转到内存的栈区，用栈存
储地址和临时变量^{也是临时变量}，但迭代
调用太多或临时变量太大，栈区内
存不够就会溢出。

如果是因迭代次数太多导致
次数 \times 每次用小栈区使得栈溢出，
则必须改变代码，不用迭代；如果

少数代码块确因多次有大量临时
变量要存储而导致栈溢出,则可以
用函数块级联使编译器将函数
体直接接在调用处,不跳转也
不用栈区来解决(或是大变量写成
全局或静态的?),这就是用代
码区(静态区)将内存来替代'栈
区内存。

从CPU寄存器 → 栈区 → 静态区
—— 本地, 1-4 字节局部

都是用更大的慢的区来限制
快的区不同的问题。