2020/04/08_第7课_函数的退出、递归

笔记本: C
创建时间: 2020/4/8 星期三 15:37
作者: ileemi
标签: 递归的使用, 函数的退出

函数的调用机制2（函数的退出）

10、恢复寄存器
11、释放局部变量
12、恢复调用放栈底
13、（1）如果是__fastcall 或者__stdcall，取出返回地址，并按此更新流程，释放参数空间。（2）如果是__cdecl（C约定）取出返回地址，并按此更新流程，这时流程回到调用方（caller）。
14、如果是__cdecl（C约定），在这个时候释放参数空间

函数开始及退出的执行过程如下：
2、按照调用约定传参，栈顶传入到被调用方，传入到参数部分
3、保存返回地址 top从参数部分到达函数的返回地址处
4、保存caller的栈底位置 被调用方保存调用方的栈底地址 top到达调用方栈底处
5、更新当前栈底到被调用方callee的位置 被调用方的栈底更新为调用方的栈底地址
6、为局部变量申请空间 栈顶到达局部变量区
7、初始化局部变量区CC (烫烫烫烫)
8、保存其它受影响的寄存器 栈顶上移
9、执行函数体
10、恢复寄存器 栈顶回到局部变量区

11、释放局部变量
12、恢复调用放栈底（从被调用方讲调用方的栈底进行恢复）
13、（1）如果是__fastcall 或者__stdcall，取出返回地址，并按此更新流程，释放参数空间。（栈顶到达调用方的保存寄存器位置）
（2）如果是__cdecl（C约定）取出返回地址，并按此更新流程，这时流程回到调用方（caller）。（栈顶到达被调用方的参数区）
14、如果是__cdecl（C约定），在这个时候释放参数空间，流程返回到caller，由caller释放参数空间
释放被调用方的栈空间，在没有新值进入时，之前栈内的数据还再（残留数据），新函数会覆盖所用到的内存数据

所谓出栈，知识修改栈顶的位置，而不会将栈内容清空

递归

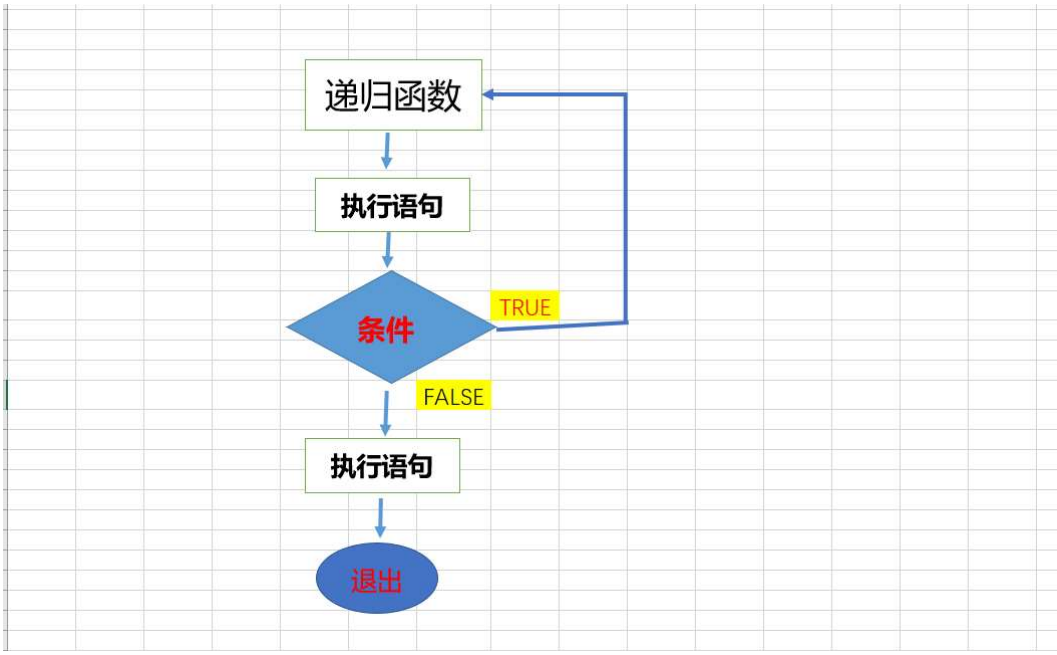
递归：可以自己调用自己的函数称为递归，使用递归需要为其做一个结束条件，如果没有，该函数就会无限调用自己，无限递归。

递归和函数间设计思路上的区别

递归：是个解决非线性问题，不适合解决线性问题
循环：适合解决线性问题，不适合解决非线性问题

线性问题：有且仅有一个前驱及后继。

线性问题使用循环远比使用递归的效率高



C 语言支持递归，即一个函数可以调用其自身。但在使用递归时，程序员需要注意定义一个从函数退出的条件，否则会进入死循环。递归函数在解决许多数学问题上起了至关重要的作用，比如计算一个数的阶乘、生成斐波那契数列，等等。