x64逆向：调用约定和作用域

调用约定

x64只有一种调用约定，前四个参数使用`rcx`、`rdx`、`r8`、`r9`寄存器传递，多余的参数从右向左压栈。

浮点用`xmm0`、`xmm1`、`xmm2`、`xmm3`来传递

任何超过8个字节或者不是1、2、4、8字节的参数必须由引用来传递。

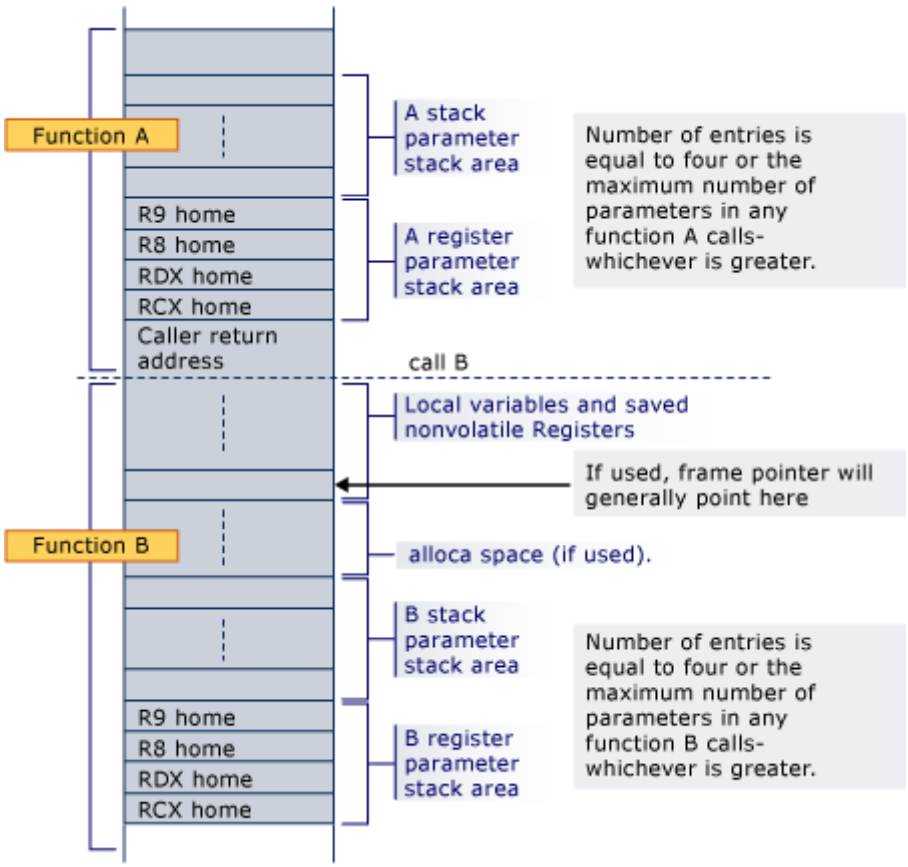
返回值存放于`rax`，浮点在`xmm0`，超过8字节的返回引用

参数	类型	浮点类型
第一个参数	RCX	XMM0
第二个参数	RDX	XMM1
第三个参数	R8	XMM2
第四个参数	R9	XMM3

注意：如果混合使用整型和浮点，则根据参数位置使用相应的寄存器，多余的入栈

栈帧

与x86下的栈帧有所区别，如图所示（其中地址从下到上增长）



作用域

1. 局部变量
    - 在预留空间之上，至于是参数还是局部变量要看情况
2. 全局变量
    - 在全局数据区
    - 以函数调用来初始化，依然会在`initerm`函数中遍历初始化函数列表来进行初始化
3. 数组
    - 初始化会利用`xmm`寄存器来进行8字节的赋值，剩余的大小在挨个赋值
    - 访问依然使用数组的寻址公式