作者：代兴　邮箱：503268771@qq.com　Vx 公众号：东方架构师

## 函数的组成部分

函数的含义为实现一个独立功能的可以重复调用的代码块，代码块包含很多汇编指令。函数是功能模块的最小载体，非常重要。

函数的组成部分：函数名，函数体，入参，返回值。函数体，入参，返回值，可以没有。
高级编程语言的函数格式：

```
返回值 函数名 ( 入参 ) {
    函数体
}
```

## 用 C 和汇编分析函数结构

编写代码： structure.c

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

// 函数的 4 个部分：函数名、入参、返回值、函数体
float func_sum_num(int age, int *age2)
{
    int sum = age + *age2 + 1;
    float ret = (float)sum;
    return ret;
}

// main 函数
int main()
{
    // 局部变量
    int age = 3;
    int age2 = 5;

    // 调用自定义函数 func_sum_num
    float ret = func_sum_num(age, &age2);

    // 调用库函数 printf
    printf("函数的返回值 = %6.2f \n", ret);

    return 0;
}
```

编译代码：

```
gcc structure.c  -o structure
gcc structure.c -S -o structure.s
```

运行代码：

```
[root@localhost func]# ./structure
函数的返回值 =   9.00
```

分析结果：

对比源文件 structure.c 和汇编文件 structure.s。

| | C 语言代码 | 汇编代码 | 分析 |
|---|---|---|---|
| 函数名 | func_sum_num | func_sum_num: | C 语言，函数名等于符号名，两个名称相同。 |
| 入参 | int age, int *age2 | movl %edi, -20(%rbp)<br>movq %rsi, -32(%rbp) | 入参 int age，是个整数，使用寄存器 edi 传递。<br>入参 int *age2，是个指针，使用寄存器 rsi 传递。<br>把 2 个入参保存到函数栈上面。 |
| 函数体 | int sum = age + *age2 + 1;<br>float ret = (float)sum;<br>return ret; | movq -32(%rbp), %rax<br>movl (%rax), %edx<br>movl -20(%rbp), %eax<br>addl %edx, %eax<br>addl $1, %eax<br>movl %eax, -4(%rbp)<br>cvtsi2ss -4(%rbp), %xmm0<br>movss %xmm0, -8(%rbp)<br>movl -8(%rbp), %eax<br>movl %eax, -24(%rbp) | 函数体包含加法，指针取值，int 转 float。<br>汇编代码，一行使用一个指令。<br>3 行 C 语言代码，变成很多行汇编代码。 |
| 返回值 | float | movss -24(%rbp), %xmm0 | 返回浮点数，使用寄存器 xmm0。 |
| 函数调用 | float ret = func_sum_num(age, &age2); | movq %rdx, %rsi<br>movl %eax, %edi<br>call func_sum_num<br>movss %xmm0, -20(%rbp) | 汇编指令 call 实现函数调用。<br>call 上方，使用 edi、rsi 传递入参。<br>call 下方，使用 xmm0 接收返回值。 |