

# 《计算机系统》理论课—程序机器级表示：过程

## 课堂要点

本次课堂教学主要内容为：**过程的特点、程序栈的使用、过程调用的细节、嵌套与递归、通过传递指针实现变量交换的 SWAP 函数**。作为第三章最重要的内容，本节课内容着重展现了程序在机器中执行时相对完整的过程。

**1. 过程的特点：**过程即高级语言中的函数，是程序模块化的象征。通过过程，程序变得简洁明了，易于维护和复用，开发效率也更高。在机器中执行时，过程调用将程序的控制和数据从调用者 P 传递给被调用者 Q，为每一个过程分配相应的栈帧空间并在过程退出时释放这些空间。

**2. 程序栈：**过程调用通过程序栈来实现，利用程序栈来提供一个不同过程之间参数传递的平台。在栈中还为过程中的局部变量开辟存储空间并且保存会在被调用者过程中被覆盖的寄存器；调用者的栈帧和被调用者的栈帧是紧密相连的，以此实现便捷的参数传递操作

**3. 过程调用演示。**演示中重点展示了栈的构成和变化。

a. 当调用者 P (main) 要调用被调用者 Q (sum) 时，它在栈中做了两步准备工作：i) 将 sum 需要的参数拷贝到栈顶位置（构造参数传递区）；ii) 将返回地址保存到栈顶。然后是 Q(sum) 开始构造自己的栈帧，首先保存 P 的 ebp 值，为结束调用返回 main 的栈帧做准备，然后为自己的局部变量准备空间并赋值，再从传递参数区取得参数进行运算并将获得的结果（返回值）保存到默认返回值保存寄存器 %eax。

b. Q 运算完毕返回时，首先是恢复 P 的栈帧，即将保存的 old ebp 出栈送给 %ebp，使得 %ebp 重新指向 P 的栈底位置。然后将返回地址出栈送给 %eip，使得机器可以继续执行 P 中调用 Q 之后的第一条指令。

**4. 嵌套与递归：**嵌套调用就是 Q 中还有对其它过程的调用。当 Q 调用一个新的过程 R 时，栈帧变化与 P 调用 Q 时基本一致，返回时亦然。递归则是 Q 中有对自身的调用，当递归结束条件满足时，再逐级返回，因此此时栈帧变化是一直向

下生长直到递归结束条件满足的最后一次自身调用，然后栈帧从下往上逐级释放，最终回到 P。

#### 5. swap 函数。

swap 函数利用指针作为传递参数，有效地实现了对调用者的实际局部变量 x, y 的数值交换。如果直接使用 x, y 作为函数的参数，那么交换只会发生在传递参数区的那两个副本 x, y 之间，而不会改变实际的 x, y 值。