

本小节内容

函数的声明与定义
函数的分类与调用

1 函数的声明与定义

函数间的调用关系是，由主函数调用其他函数，其他函数也可以互相调用。同一个函数可以被一个或多个函数调用任意次，如图 1.1 所示。

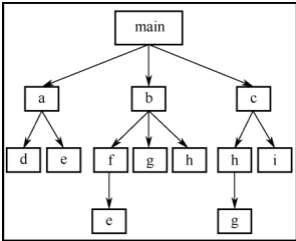


图 1.1 函数间的调用关系

下面来看例 1.1。例中有两个 c 文件，func.c 是子函数 printstar 和 print_message 的实现，也称定义；main.c 是 main 函数，func.h 中存放的是标准头文件的声明和 main 函数中调用的两个子函数的声明，如果不在头文件中对使用的函数进行声明，那么在编译时会出现警告。

【例 1.1】函数嵌套调用。

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

int printstar(int i); //函数声明
void print_message();

func.c
#include "func.h"

int printstar(int i) //i 即为形式参数
{
    printf("*****\n");
    printf("printstar %d\n",i);
    return i+3;
}
```

```

void print_message() //可以调用 printstar
{
    printf("how do you do\n");
    printstar(3);
}

main.c
#include "func.h"
int main()
{
    int a=10;
    a=printstar(a);
    print_message();
    printstar(a);
    return 0;
}

```

C 语言的编译和执行具有以下特点。

(1) 一个 C 程序由一个或多个程序模块组成，每个程序模块作为一个源程序文件。对于较大的程序，通常将程序内容分别放在若干源文件中，再由若干源程序文件组成一个 C 程序。这样处理便于**分别编写、分别编译，进而提高调试效率（复试有用）**。一个源程序文件可以为多个 C 程序共用。

(2) 一个源程序文件由一个或多个函数及其他有关内容（如命令行、数据定义等）组成。一个源程序文件是一个编译单位，在程序编译时是以源程序文件为单位而不是以函数为单位进行编译的。`main.c` 和 `func.c` 分别单独编译，在链接成为可执行文件时，`main` 中调用的函数 `printstar` 和 `print_message` 才会通过链接去找到函数定义的位置。

(3) C 程序的执行是从 `main` 函数开始的，如果在 `main` 函数中调用其他函数，那么在调用后会返回到 `main` 函数中，在 `main` 函数中结束整个程序的运行。

(4) 所有函数都是平行的，即在定义函数时是分别进行的，并且是互相独立的。一个函数并不从属于另一函数，即函数不能嵌套定义。函数间可以互相调用，但不能调用 `main` 函数。`main` 函数是由系统调用的，例 1.1 的 `main` 函数中调用 `print_message` 函数，而 `print_message` 函数中又调用 `printstar` 函数，我们把这种调用称为**嵌套调用**。

函数的声明与定义的差异如下。

(1) 函数的定义是指对函数功能的确立，包括指定函数名、函数值类型、形参及其类型、函数体等，它是一个完整的、独立的函数单位。

(2) 函数的声明的作用是把函数的名字、函数类型及形参的类型、个数和顺序通知编译系统，以便在调用该函数时编译系统能正确识别函数并检查调用是否合法。

隐式声明：C 语言中有几种声明的类型名可以省略。例如，**函数如果不显式地声明返回值的类型，那么它默认返回整型**；使用旧风格声明函数的形式参数时，如果省略参数的类型，那么编译器默认它们为整型。然而，依赖隐式声明并不是好的习惯，因为隐式声明容易让代码的读者产生疑问：编写者是否是有意遗漏了类型名？还是不小心忘记了？显式声明能够清楚地表达意图！

2 函数的分类与调用

从用户角度来看，函数分为如下两种。

(1) 标准函数：即库函数，这是由系统提供的，用户不必自己定义的函数，可以直接使用它们，如 `printf` 函数、`scanf` 函数。不同的 C 系统提供的库函数的数量和功能会有一些不同，但许多基本的函数是相同的。

(2) 用户自己定义的函数：用以解决用户的专门需要。

从函数的形式看，函数分为如下两类。

(1) 无参函数：一般用来执行指定的一组操作。在调用无参函数时，主调函数不向被调用函数传递数据。

无参函数的定义形式如下：

```
类型标识符  函数名()
```

```
{  
    声明部分  
    语句部分  
}
```

在例 1.1 中，`print_message` 就是无参函数。

(2) 有参函数：主调函数在调用被调用函数时，通过参数向被调用函数传递数据。

有参函数的定义形式如下：

```
类型标识符  函数名(形式参数表列)
```

```
{  
    声明部分  
    语句部分  
}
```

在例 1.1 中，`printstar` 就是有参函数，`int i` 对应的 `i` 为形式参数，主调函数和被调用函数之间存在数据传递关系。