

C 程序结构

在我们学习 C 语言的基本构建块之前，让我们先来看看一个最小的 C 程序结构，在接下来的章节中可以以此作为参考。

C Hello World 实例

C 程序主要包括以下部分：

- 预处理器指令
- 函数
- 变量
- 语句 & 表达式
- 注释

让我们看一段简单的代码，可以输出单词 "Hello World"：

实例

```
#include <stdio.h>
int main()
{
    /* 我的第一个 c 程序 */
    printf("Hello, World! \n");
    return 0;
}
```

接下来我们讲解一下上面这段程序：

1. 程序的第一行 `#include <stdio.h>` 是预处理器指令，告诉 C 编译器在实际编译之前要包含 `stdio.h` 文件。
2. 下一行 `int main()` 是主函数，程序从这里开始执行。
3. 下一行 `/* ... */` 将会被编译器忽略，这里放置程序的注释内容。它们被称为程序的注释。
4. 下一行 `printf(...)` 是 C 中另一个可用的函数，会在屏幕上显示消息 "Hello, World!"。
5. 下一行 `return 0;` 终止 `main()` 函数，并返回值 0。

编译 & 执行 C 程序

接下来让我们看看如何把源代码保存在一个文件中，以及如何编译并运行它。下面是简单的步骤：

1. 打开一个文本编辑器，添加上述代码。
2. 保存文件为 `hello.c`。
3. 打开命令提示符，进入到保存文件所在的目录。
4. 键入 `gcc hello.c`，输入回车，编译代码。

5. 如果代码中没有错误，命令提示符会跳到下一行，并生成 *a.out* 可执行文件。
6. 现在，键入 *a.out* 来执行程序。
7. 您可以看到屏幕上显示 "Hello World"。

```
$ gcc hello.c  
$ ./a.out  
Hello, World!
```

请确保您的路径中已包含 gcc 编译器，并确保在包含源文件 *hello.c* 的目录中运行它。

如果是多个 c 代码的源码文件，编译方法如下：

```
$ gcc test1.c test2.c -o main.out  
$ ./main.out
```

test1.c 与 *test2.c* 是两个源代码文件。

[← C 环境设置](#)[C 基本语法 →](#)**3 篇笔记**** 写笔记**