

bennyhuo · 更新于 2021-01-05

上一节2-1 环境搭建2-3 程序调试方法下一节

## HelloWorld

我们程序员最擅长的事儿就是写 HelloWorld。不管是学习什么新技术，我们都要搞得特别有仪式感，先来个 HelloWorld。

<> 代码块

```
1#include <stdio.h>
2
3int main() {
4    printf("Hello, World!\n");
5    return 0;
6}
```

这段程序将在控制台打印 Hello, World!。

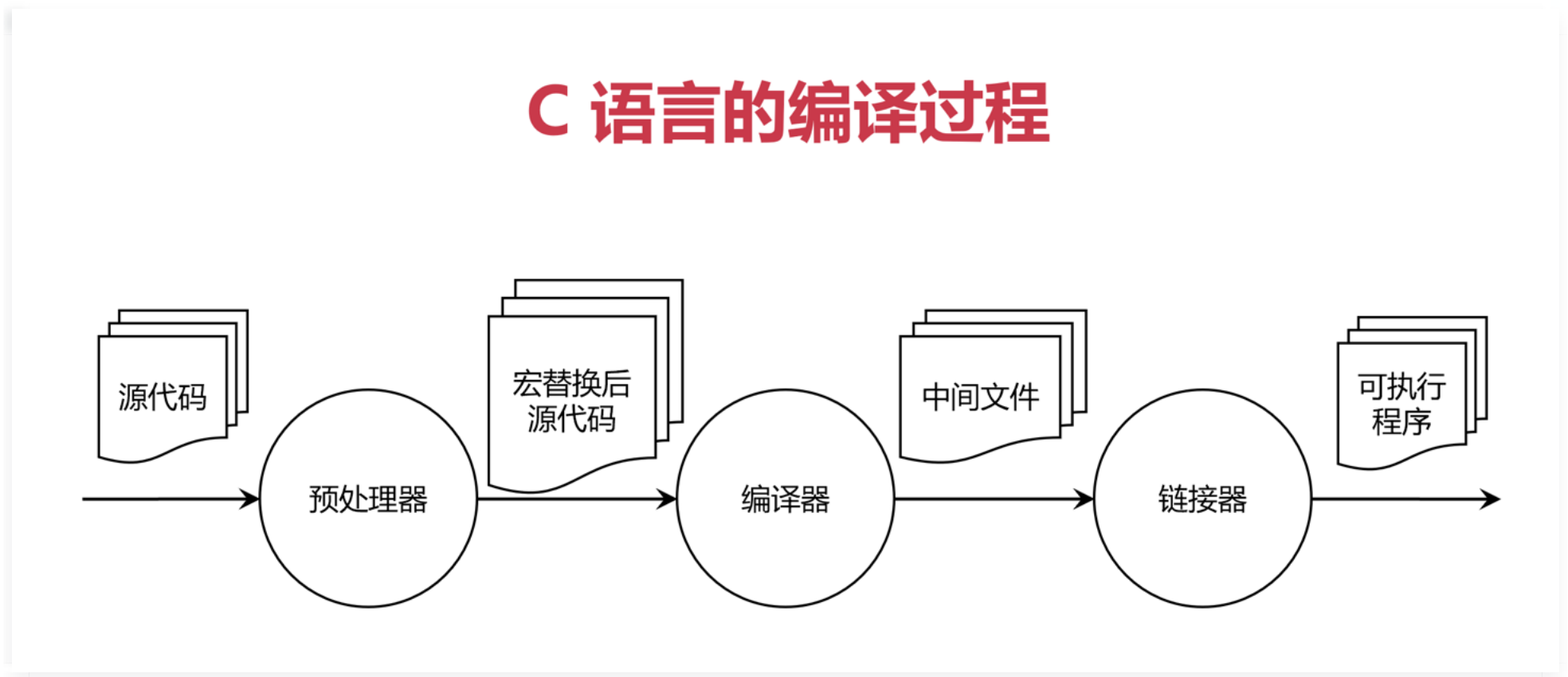
我们在剖析这个程序之前，我们先想想程序究竟是用来干什么的。程序是一组指令，运行程序就是执行一组指令，说得再直白一点儿就是，执行程序就是让计算机听人指挥干活。

那么我们的第一个程序就是希望计算机打印一个 Hello, World! 让我们开心一下，既然明确了我们的目的，我们来简单看下这个程序的结构。这当中会涉及一些概念大家可能不熟悉，没关系，后面我们都会逐一详细介绍。

<> 代码块

```
1#include <stdio.h>
```

第一行是一个预处理指令，啥叫预处理？就是程序编译之前执行的操作。那编译又是啥？编译就是我们将我们编写的源代码翻译成机器能够执行的机器指令的过程，源代码机器是不能直接执行的，不然程序员就要下岗了。我们先大概看下 C 语言的编译过程，有个初步的印象：



预处理指令又叫做宏，这个大家先做了解即可。

接下来我们再来看下面的 main 函数：

<> 代码块

```
1int main() {
2    printf("Hello, World!\n");
3    return 0;
4}
```

我们的程序就是要做一件事儿的指令，我们通常把相关的指令放到一个函数当中，供其他函数调用，这样就能形成一个完成的函数调用的过程。可是这个调用过程的起点在哪里呢？

这就跟公司的 OKR或者KPI或者其他的考核体系一样，大致都是公司的业务目标从大领导层层向下拆解，因此一定有一个起点。

C 语言中这个函数调用的起点就是 main 函数，你可以说它是那个不平凡的人，如果你熟悉黑客帝国，那么你应该想到它就是 The One。

所以你知道了程序就是这么开始执行的，不管多么庞大的 C 语言程序，它一定有一个 main 函数。

<> 代码块

```
1printf("Hello, World!\n");
```

那么下一条就是函数里面的这一句了。printf 也是一个函数，它是由 C 语言标准库定义的，因此我们可以在任何支持 C 标准的平台上调用它，它的意思是 print format，格式化输出，不过具体到格式化的问题，我们后面慢慢聊，今天这里只是输出了一个 "Hello, World!\n"。

等等，那个 \n 是什么东西？哦，这个是换行符啦。我们想要告诉计算机打印完 Hello, World! 之后再换个行，可是换行符又不想字母那样那么直观，因此就跟计算机来了个君子协定，约定 \n 为换行符。

接下来的 return 0; 就是告诉 main 函数我们执行完啦，而且程序运行没什么问题，这里的 0 表示正常退出的意思。

程序运行结束之后有时候我们会看到类似下面的文字：

C:\Users\bennyhuo\Workspace\imooc\BasicC\code\BasicC\cmake-build-debug\BasicC.exe  
Hello, World!  
  
Process finished with exit code 0  
|

这个所谓的“exit code 0”就与我们这里的 return 0; 相对应。不信的话，可以自己改成其他数试试看。

好的，至此，我们终于迈出了学习 C 语言的第一步。它虽然看上去简单，不过我们仍然有些问题没有完全搞清楚，例如 int 是什么？函数怎么定义？怎么键盘输入文字？main 函数是谁调用的？等等等等，这些都是我们课程后续要讲的内容。