

# 编译系统

——张仁发

## 一、源代码

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("hello, world\n");
6  }
```

图 1-1 源程序

## 二、文本表示

|     |      |      |      |      |     |     |      |      |     |     |     |     |     |     |     |
|-----|------|------|------|------|-----|-----|------|------|-----|-----|-----|-----|-----|-----|-----|
| #   | i    | n    | c    | l    | u   | d   | e    | <sp> | <   | s   | t   | d   | i   | o   | .   |
| 35  | 105  | 110  | 99   | 108  | 117 | 100 | 101  | 32   | 60  | 115 | 116 | 100 | 105 | 111 | 46  |
|     |      |      |      |      |     |     |      |      |     |     |     |     |     |     |     |
| h   | >    | \n   | \n   | i    | n   | t   | <sp> | m    | a   | i   | n   | (   | )   | \n  | {   |
| 104 | 62   | 10   | 10   | 105  | 110 | 116 | 32   | 109  | 97  | 105 | 110 | 40  | 41  | 10  | 123 |
|     |      |      |      |      |     |     |      |      |     |     |     |     |     |     |     |
| \n  | <sp> | <sp> | <sp> | <sp> | p   | r   | i    | n    | t   | f   | (   | "   | h   | e   | l   |
| 10  | 32   | 32   | 32   | 32   | 112 | 114 | 105  | 110  | 116 | 102 | 40  | 34  | 104 | 101 | 108 |
|     |      |      |      |      |     |     |      |      |     |     |     |     |     |     |     |
| l   | o    | ,    | <sp> | w    | o   | r   | l    | d    | \   | n   | "   | )   | ;   | \n  | }   |
| 108 | 111  | 44   | 32   | 119  | 111 | 114 | 108  | 100  | 92  | 110 | 34  | 41  | 59  | 10  | 125 |

图 1-2 hello.c 的 ASCII 文本表示

## 三、编译流程

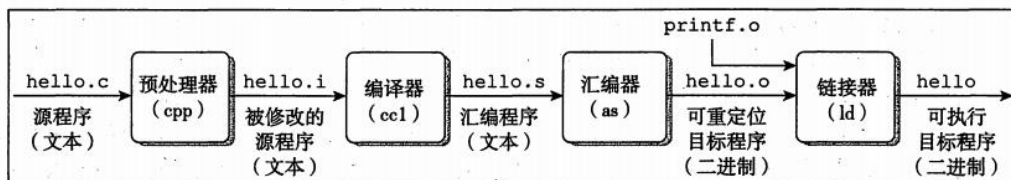


图 1-3 编译系统

- 预处理阶段。预处理器 (cpp) 根据以字符 # 开头的命令，修改原始的 C 程序。比如 hello.c 中第 1 行的 #include <stdio.h> 命令告诉预处理器读取系统头文件 stdio.h 的内容，并把它直接插入到程序文本中。结果就得到了另一个 C 程序，通常是以 .i 作为文件扩展名。
- 编译阶段。编译器 (cc1) 将文本文件 hello.i 翻译成文本文件 hello.s，它包含一个汇编语言程序。汇编语言程序中的每条语句都以一种标准的文本格式确切地描述了一条低级机器语言指令。汇编语言是非常有用的，因为它为不同高级语言的不同编译器提供了通用的输出语言。例如，C 编译器和 Fortran 编译器产生的输出文件用的都是一样的汇编语言。
- 汇编阶段。接下来，汇编器 (as) 将 hello.s 翻译成机器语言指令，把这些指令打包成一种叫做可重定位目标程序 (relocatable object program) 的格式，并将结果保存在目标文件 hello.o 中。hello.o 文件是一个二进制文件，它的字节编码是机器语言指令而不是字符。如果我们在文本编辑器中打开 hello.o 文件，看到的将是一堆乱码。
- 链接阶段。请注意，hello 程序调用了 printf 函数，它是每个 C 编译器都会提供的标准 C 库中的一个函数。printf 函数存在于一个名为 printf.o 的单独的预编译好了的目标文件中，而这个文件必须以某种方式合并到我们的 hello.o 程序中。链接器 (ld) 就负责处理这种合并。结果就得到 hello 文件，它是一个可执行目标文件（或者简称为可执行文件），可以被加载到内存中，由系统执行。