#include<stdio.h>

int main()

{

printf("Hellow world");

}

程序生命周期是从一个“源文件”开始的，有程序员写到文本并保存，文件名\*\*\*.c.源程序实际上是由0和1组成的位。这些位8个一组，称为字节。每一个字节表示程序中的某个字符。例如第一个字节的整数值是35，对应的字符是“#”，第二个字符对应的是105对应i，依次类推。只由ASCII字符组成的文件称为文本文件，其他的是二进制文件！

为了能让机器读懂我们语言，所以需要进行一系列的转换，转换为机器能读懂的低级语言指令，然后这些指令按照一种称为可执行目标程序的格式打包好，以二进制磁盘文件形式存放起来，目标程序也称为可执行程序，

从源文件到目标文件的转换是由编译器驱动程序完成的

分为以下几个阶段：

预处理阶段：预处理器根据以字符#开头的命令修改原始的C程序，比如hello.c中第一行#include<stdio.h>指令告诉预处理器读取系统文件stdio.h里面的内容，并把它直接插入到程序文本里面，得到另外一个C程序，通常以.i作为文件扩展名。

除此之外预处理阶段还做一下事情：

1.进行宏替换过程，比如define指令定义的内容(只是简单的替换)。（const只在编译运行时才会检查）

2.删除掉注释的乱七八糟。

编译阶段：将hello.i翻译成了文本文件hello.s，包含一个汇编语言程序。汇编语言程序中每条语句都以标准的文本格式确切描述了一条低级机器语言指令。汇编语言很有用，因为不同高级语言最后都转换为通用汇编语言了。主要是进行了语法检查，语义分析，并将代码翻译成了汇编语言。

汇编阶段：汇编阶段是将hello.s翻译成了机器语言，可重定位的的目标文件，保存在hello.o中，它是一个二进制文件，字节编码是机器语言指令，而不是字符。如果在编译器里面打开的话将是一串乱码。

链接阶段：这个阶段主要是将静态库和动态库链接，打包到一起称为可执行文件。比如printf存在于一个pintf.o程序中，链接器就负责他们的链接，合并为一个可执行文件。