第1章 计算机系统漫游

hello world(用C语言打印hello world)

通过跟踪hello程序来对系统的学习

hello的生命周期

代码文件被程序员创建,到在系统上运行,输出hello信息

信息就是 位 + 上下文

源程序(代码文件)实际是由0和1组成的位(比特)序列,8位=1字节,每个字节表示某些文本符号

大部分现在计算机都使用ASCII标准来表示文本字符,即一个字节大小的整数值表示一个字符

如 A ->65 , # -> 35 \n(换行符) -> 10

像这种只由ASCII构成的文件为文本文件,其他文件为二进制文件

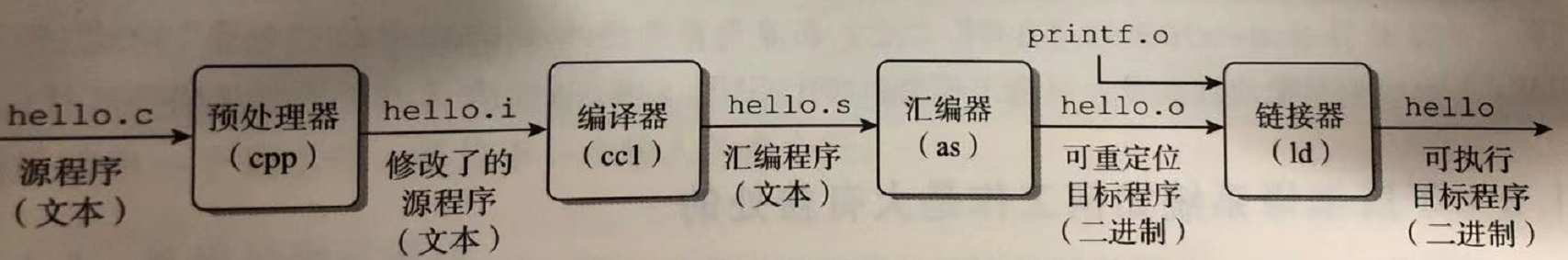
系统中所有的信息包括磁盘文件,内存文件,网络传送的数据都是由一串比特表示的

区别不同数据对象的唯一方法是要看上下文,在不同的上下文中,一个同样的字节序列可能表示是一个整数或浮点数或字符串或机器指令

程序被其他程序翻译成不同的格式

Linux> gcc -o hello hello.c

hello为最终生成的执行程序, hello.c为源代码程序



预处理阶段

预处理器根据以#开头的命令,修改原始的C程序,如 #include <stdio.h>是告诉预处理器充取stdio.h的内容并插入文本中,结果就得到了了以.i作为文件扩展名的另一个C程序

编译阶段

编译器将hello.i文本文件翻译成hello.s文本文件,里面包含一个汇编语言程序(包含函数main的定义)

汇编语言为不同高级语言的不同编译器提供了通用的输出语言(如c,python编译器产生的是一样的汇编语言)

汇编阶段

汇编器将hello.s翻译成机器语言指令,并打包成叫可重定位目标程序的格式保存在hello.o文件(二进制文件),用文本编译器打开是乱码)中

链接阶段

里面调用了prinft函数(标准C库的一个函数,存在于一个叫prinft.o预编好的目标文件中),要以某种方式合并到hello.o中,链接器(ld)就是处理这种合并,最终得到hello执行文件,可加载到内存中由系统执行