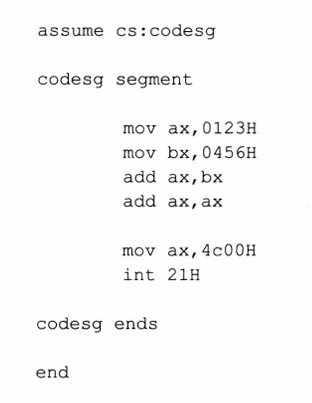
\_04\_第一个程序

**1.源程序**

（1）基本语法：



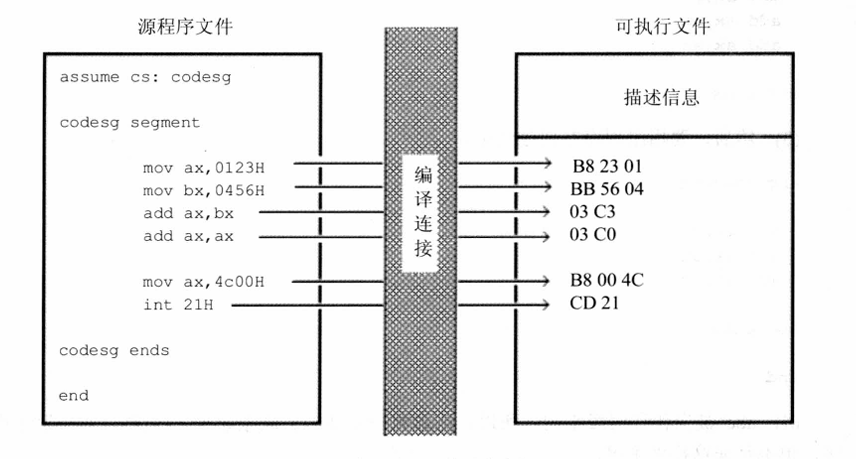
（2）汇编指令与伪指令

一个完整的汇编程序，其中会包含汇编指令和伪指令。

①汇编指令：和机器指令一一对应，最终会被CPU执行。

②伪指令：没有对应的机器指令，最终不会被CPU执行。

伪指令在编译阶段由编译器来解释，为汇编程序的编译提供信息。



（3）段名（Segment Name）和标号（Label）

①段名（Segment Name）

段名用于标识不同的段，如代码段、数据段、堆栈段等。

一个段名对应一个**段地址**。

②标号（Label）

**标号，是伪指令中的一个地址标签，它对应着源程序中的一个指定位置。**

一个标号唯一对应一个**内存地址**。

源程序中的标号，经过编译的过程后，会被转换成对应的内存地址，用以服务汇编程序。

③段名和标号的区别

Ⅰ段名唯一对应一个段地址，而标号唯一对应一个内存地址。

Ⅱ段名一般用在段定义时，而标号一般用于代码段中的随意哪条指令前。

（4）几条常见的伪指令

①assume指令

（英文：assume=假设）

语法：assume 段寄存器: 段名

功能：为段指定与之对应的段寄存器。

e.g. 若seg\_name1段为代码段，则assume cs: seg\_name1

②segment、ends指令

（英文：segment=段，ends=end segment=结束段）

语法：段名 segment

段中的内容....

段名 ends

功能：用于定义一个段，segment表示段的开始，ends表示段的结束。

注意：若一个源程序中定义了多个段，其中必须至少有一个段是代码段。

③end指令

语法：end [标号]

功能：用于结束整个汇编源程序。若end后添加标号，该标号就是程序开始的入口。（由于一个程序只有一个end，因此CPU寻址程序的入口是从end后面的标号开始寻址找）

e.g. ……

段名 segment

start: ……

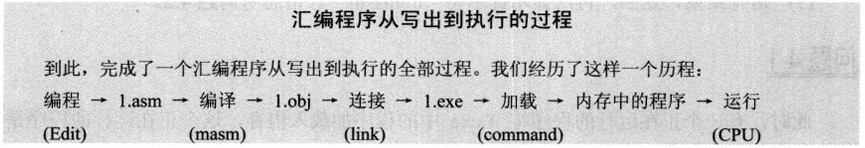
……

段名 ends

end start; （这里指明程序从start标号开始执行，即初始cs:ip指向start）

**2.汇编程序的执行过程**

以下是利用MASM执行汇编程序的过程：



（1）编译

概念：编译阶段将.asm结尾的汇编源文件编译为.obj结尾的二进制文件。

masm.exe：在DOS系统中，我们需要用MASM提供的masm.exe对源文件进行编译。

语法：masm 源文件名.asm

快速编译语法：masm 源文件名.asm;（加分号，自动忽略可选流程）

（2）连接

概念：连接阶段将一个或多个.obj文件进行链接，生成.exe结尾的可执行文件。

link.exe：在DOS系统中，我们需要用MASM提供的link.exe对目标文件进行连接。

语法：link 目标文件名.obj

快速连接语法：link 目标文件名.obj;（加分号，自动忽略可选流程）

（3）运行

源文件经过编译、连接后生成的.exe文件，我们可以在DOS系统中直接运行。

语法：可执行文件名.exe

（4）调试

debug.exe：在DOS系统中，我们需要用MASM提供的debug.exe对可执行文件进行调试。

语法：debug 可执行文件名.exe