**一、汇编语言简介**

**1汇编语言层**

汇编语言：数字型机器指令的符号表示(mov移动；push压栈等)

汇编语言相关的翻译器：

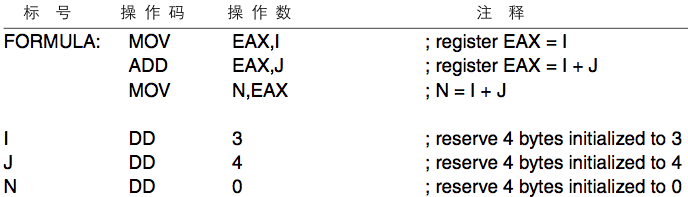
·汇编器 一个程序，将汇编语言翻译成数字型的机器指令

·编译器 一个程序，将高级语言(C、JAVA)转化为汇编语言

高级语言的翻译过程：先编译成汇编语言，在汇编成数字型机器指令

**2汇编语句格式**

语句的组成：由标号(内存地址的符号表示)、操作码、操作数、注释组成



**3汇编语句类型**

机器指令语句：数字型机器指令对应的语句

伪指令语句：汇编本身使用的命令(分配一些存储空间、把一个新页放入链表等)



**4宏**

宏定义：给一段程序取名，汇编器遇到宏定义时将其保存在宏定义表中



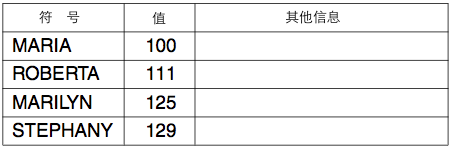
宏调用：把宏名作为操作码使用。

宏扩展：编译时将宏名称替换为宏定义的程序

**二、汇编过程**

**1第一趟扫描**

建立符号表：解决程序向前引用未定义的符号



·符号 带符号的语句和伪指令分配的符号(逻辑上的变量、方法等)

·值 符号的数值或指向符号的指针

·其他 符号值的长度、能否在过程之外访问

建立宏定义表：解决宏调用时实现宏的扩展

**2第二趟扫描**

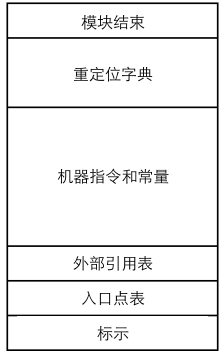
生成目标程序：

·将汇编指令翻译成数字型的机器指令

·用符号表中的值替换程序中出现的符号

·将伪指令(段定义等)翻译成机器指令(在GDT、LDT中初始化描述符)

生成链接器所需信息(目标模块)：该信息不被引入内存，链接完成后将抛弃



·标示：模块名称、模块不同部分的长度、汇编日期等

·入口点表：对其他模块公开的符号表(变量、过程)，由public伪指令定义

·外部引用表：引用其他模块的符号表(变量、过程)，由extern伪指令声明

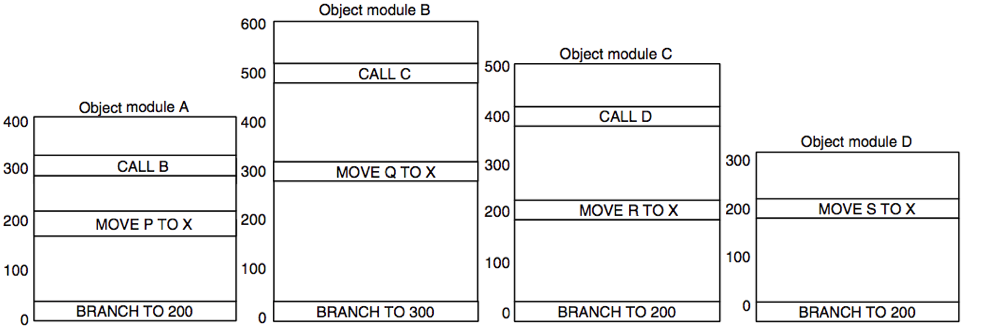
·机器指令和常量：目标程序，既由第二趟扫描生成的可执行程序

·重定位字典：引用内存地址的操作数被记录，链接重定位时修正为正确的地址

·结束模块：设置一个校验等

**三、链接目标模块**

将多个单独汇编的过程链接成一个可执行过程

****

**1第一趟扫描模块** 建立目标模块信息表：通过模块的标示部分，设置各模块长度、起始地址



建立全局符号表：将各模块的入口点表整合成一张

**2第二趟扫描模块**

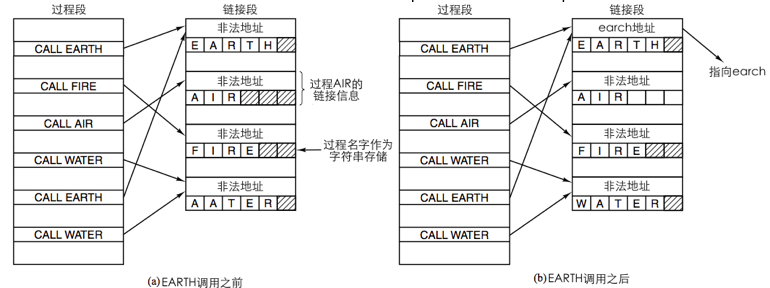
设置模块重定位地址：内存地址的操作数 + 重定位常量(模块的起始地址)

设置模块外部引用地址：全局符号表修改模块内的外部引用地址

将模块的程序部分存入二进制执行文件

**四、动态链接**

**1程序中的链接段**



地址：方法或符号对应的地址，初始为非法地址(产生陷阱/异常)

方法/符号名称：用字符串存储的名称

**2动态链接过程**

程序调用外部符号：产生一个异常，然后被动态链接器捕获处理

动态链接器捕获并处理异常

·遍历动态链接库寻找对应符号的过程的地址

·修正链接段对应符号的非法地址为上述地址

程序继续执行：再次执行产生异常的方法