几个重要的概念

汇编语言程序（Assembly language program）：根据汇编语言语法规则编写好的.asm 文件，也称“源代码”，如图 1 所示即为一份示例程序。

汇编程序（Assembler）：将汇编语言程序转换为机器语言的可执行文件。

汇编（Assembling）：汇编程序将汇编语言程序转换为机器语言的过程。

```
1  ; 提示输入英文字母串，进行大小写转换后显示
2  DATA  SEGMENT
3    inputCharInfo DB "Please input a char: $"
4    inputStrInfo DB "Please input a string: $"
5    N = 10
6    buf DB N, ?, N DUP('$')
7  DATA  ENDS
8  CODE SEGMENT
9     ASSUME  CS:CODE, DS:DATA
10     ; 设置数据段
11 START:   MOV AX, DATA
12     MOV DS, AX
13     ; 字符串输出
14     LEA DX, inputStrInfo
15     MOV AH, 09H
16     INT 21H
17     ; 字符串输入
18     MOV DX, OFFSET buf
19     MOV AH, 0AH
20     INT 21H
21     ; 单字符输出(显示换行)
22     MOV AH, 2
23     MOV DL, 10
24     INT 21H
25     ; 初始化指针兼计数器
26     MOV BH, 0
27     MOV BL, buf+1
28     ; 将输入字符依次进行大小写转换
29 CUL:XOR buf[BX+1], BYTE PTR 20H
30     DEC BX
31     JNZ CUL
32     ; 字符串输出
33     LEA DX, buf+2
34     MOV AH, 09H
35     INT 21H
36     ; 程序结束，返回 dos
37     MOV AX,4C00H
38     INT 21H
39 CODE ENDS
40   END START
```

图 1 一份汇编语言程序源代码示例

程序段（Program segment）：内存中，用于存放代码、数据、堆栈的存储区域，分别对应于代码段、数据段、堆栈段。图 1 中 2-7 行为数据段，8-39 行为代码段。

保留字 (Reserved Word)：汇编语言语法确定的具有特殊意义的单词或字母组合。通常包括指令助记符（MOV，ADD，MUL 等，示例程序中深蓝色加粗字母组合）、寄存器名（AX、BX、CL、DH 等，示例程序中浅蓝色加粗黄色背景字母组合）、伪指令助记符（DB、DW、ORG、SEGMET、ENDS 等，示例程序中浅蓝色字母组合）、属性（BYTE、WORD 等，示例程序中蓝黑色加粗字母组合）、运算符（+、-、SEG、OFFSET 等，示例程序中浅蓝色字母组合）、预先定义的符号（$、? 等）。保留字不区分大小写，必须正确使用，不能用作变量等标识符。

标识符 (Identifier)：由程序员选择的名称，用于标识变量（3、4、6 行的 inputCharInfo、inputStrInfo、buf）、常量（5 行的 N）、子程序、指令（11、29 行的 START、CUL）和段（2、8 行的 DATA、CODE）的地址。在汇编语言当中，标识符不区分大小写，可以由大小写字母、下划线、@、?、$、数字构成，但不能以数字开头。标识符不能与保留字相同。

伪指令 (Directive)：嵌入源代码中的指示汇编程序如何汇编的语句。伪指令不会被 CPU 执行，而是被汇编程序识别并执行，通常用来定义变量（3、4、6 行）、程序段（2-7、8-39 行）、子程序，指示汇编程序段寄存器与程序段的关系（9 行）、指令开始地址（40 行）等。伪指令由标号（可选）、伪指令助记符（必选）、操作数（必需）构成。

指令 (Instruction)：汇编语言程序中，指令经汇编程序转换为机器语言后，可以被 CPU 加载并执行。一条指令由四部分组成：标号（可选）、指令助记符（必需）、操作数（可选，通常需要）、注释（可选）。为了区别伪指令，指令也称为“硬指令”。指令有 CPU 识别并执行，一条指令完成一个基本的操作，如数据传送（MOV、LEA 指令）、算术逻辑运算（29、30 行的 XOR 和 DEC）、控制程序转移（31 行的 JNZ）、调用中断服务程序（INT 指令）等。

助记符 (Mnemonic)：用于帮助记忆的指令或伪指令功能的符号。指令助记符有 MOV、ADD、SUB、MUL、JMP、CALL 等，伪指令助记符有 DB、DW、DD、ORG、EVEN、SEGMENT、ENDS 等。对于英语母语程序员来说，助记符的字面含

义与其表示的指令或伪指令的功能具有某种相关性。所有的指令助记符和伪指令助记符都是保留字，不能用作程序员自定义的标识符。

标号 (label)：一种标记指令或数据地址的标识符，也可以用来表示常量。

标号位于指令前，以冒号隔开，表示指令的地址。示例程序中第 11、29 行的 START、CUL。

标号位于数据定义伪指令前，表示数据的地址。示例程序中 3、4、6 行的 inputCharInfo、inputStrInfo、buf。定义数据通常被称为定义变量，所以表示数据地址的标号也被称为“变量名”，我们使用的教材即使用“变量名”这个术语，且常常省略为“变量”。另外，由于指令和数据的长度是不定的，标号所表示的是指令或数据的首地址（即最低的地址）。教材中，变量特指数据标号，标号特指指令标号。

标号位于常量定义伪指令符前，表示常量的值。示例程序中第 5 行的 N。

变量 (Variable)：存储器中某个数据区的名称，表示所定义数据的地址。用标识符表示的变量，称作数据标号或变量名。由于变量名在指令中可以作为存储器操作数，表示数据的偏移地址，所以变量名也称“符号地址”。变量出现在指令当中被当作数据的存储器寻址方式，可以省略表示存储器寻址方式的“[]”。如第 14 行的指令 `LEA DX, inputStrInfo`，在汇编过程中，源操作数会被当作[0016]，即直接寻址方式。第 18 行的 `MOV DX, OFFSET buf`，buf 是存储器的直接寻址，OFFSET 作用于 buf 前，取该变量的有效地址，最后源操作数 OFFSET buf 会被当作 002EH，即立即数寻址方式。

常量 (Literal)：固定的值，分为数值常量、字符串常量。通常我们会用一个标识符表示的常量，称为“符号常量”，MASM 使用“等价 EQU”和“等号=”伪指令来定义符号常量。常量在汇编过程中即计算出对应的值，不占用存储空间。如第 5 行的 N 就是符号常量，它在汇编过程会被当作 10，且不占用内存空间。

数值常量表示数值，有多种进制形式。第 19 行的 0AH 与第 23 行的 10，分别是十六进制和十进制的 10。

字符串常量是用单引号或双引号括起来的单个字符或多个字符，其数值是每

个字符的 ASCII 码。

运算符 (Operator)：作用于常量，用于对常量进行算术、逻辑、移位、关系运算等。如+、-、MOD、AND、OR、EQ、SHL 等。

操作符 (Operator)：作用于变量和标号，用于求变量和标号的某些属性，或强制转换数据类型，如 OFFSET、SEG、LENTHOF、PTR 等。由于操作符常用于求变量和标号的某些属性，因此操作符也称“属性操作符”。如第 18 行 offset 求变量 buf 的有效地址，第 29 行 PTR 强制将立即数 20H 转换为字节型数据。

运算符和操作符的英文名称都是“Operator”，教材中将与常量运算相关的称为“运算符”，与变量和标号属性相关的称为“操作符”。

表达式 (Expression)：由寄存器、运算符、操作符、常量、立即数、变量和标号等构成的式子，在汇编的过程进行计算，并将结果作为指令的操作数。MASM 中的表达式分为“数值表达式”和“地址表达式”。注意，寄存器仅用于存储器寻址方式。如第 29 行 buf[BX+1]所示，其中[BX+1]构成了相对寄存器寻址方式，寄存器与运算符组合只能出现在方括号内，表示存储器寻址方式。

数值表达式 (Number expression)：由立即数、常量和数值运算符，变量或标号与操作符，两个变量相减构成的，表示数值的表达式。

举例 1：N SHL 1 + 2。该数值表达式中 N 是常量，1 和 2 是立即数，+是运算符，它们在汇编过程会被“常量 N 乘以 2 再加上 2（22）”代替。

举例 2：示例程序第 18 行的 OFFSET buf。该数值表达式中 offset 是属性操作符，buf 是变量，它们在汇编过程中会被 buf 的有效地址（002E）代替。

举例 3：inputStrInfo-inputCharInfo。该表达式是两个变量相减的形式。由于变量表示所标识数据的地址，所以该表达式表示 inputStrInfo 的地址减去 inputCharInfo 的地址，即这两个变量所标识的数据的距离。又因为这两个变量是紧挨着的，它们的距离实际上也是变量 inputCharInfo 所占存储单元的个数。

地址表达式 (Address expression)：由变量或标号、加减运算符、地址计数器$构成的，表示地址的表达式。示例程序中第 27、33 行的 buf+1 和 buf+2 即为由变量、加法运算符、立即数构成的地址表达式，其表示的是 buf 所指存储单元的地址加 1 或加 2。