**一、基本概念和基础知识**

1.汇编语言的一般概念

汇编语言（Assembly Language）是任何一种用于电子计算机、微处理器、微控制器或其他可编程器件的低级语言，亦称为符号语言。通俗来讲，汇编语言是直接在硬件之上工作的编程语言。

2.学习汇编语言的用处

汇编语言是一门程序设计语言，在数以千计的计算机语言中，有着不可替代的重要地位，广泛地用于开发操作系统内核、设备驱动程序等。同时，汇编语言是计算机提供给用户最快也最有效的语言，也是利用计算机所有硬件特性并能直接控制硬件的唯一语言，因此在对程序的空间和时间要求很高的场合，汇编语言是必不可少的。而在很多需要直接控制硬件的应用场合，汇编语言也是不可或缺的。

3.什么是机器语言

机器语言是机器能直接识别的程序语言或指令代码，无需经过翻译，每一操作码在计算机内部都有相应的电路来完成它，或指不经翻译即可为机器直接理解和接受的程序语言或指令代码。比如：1000100111011000。

4.为什么会产生汇编语言

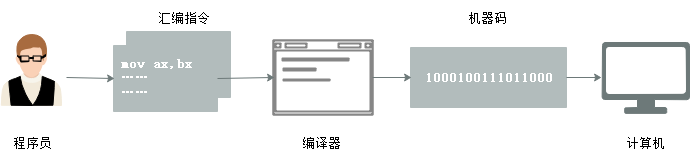
早期的程序设计均使用机器语言，但是机器语言难以辨别和记忆，给产业发展带来了障碍，于是产生了汇编语言。

5.汇编语言与机器语言的区别

机器语言是机器能直接识别的程序语言或指令代码，是用二进制代码表示的语言；而汇编语言是面向机器的程序设计语言，用易于理解和记忆的名称和符号来表示机器指令中的操作码。他们的差别在于指令的表示方法上，汇编指令是机器指令便于记忆的书写格式。例如：机器指令1000100111011000表示把寄存器BX的内容送到AX中，汇编指令则写成mov ax,bx。这样的写法与人类语言接近，便于阅读和记忆。

6.用汇编语言编写程序的工作过程

程序员用汇编语言写出源程序，再用汇编编译器将其编译为机器码，由计算机最终执行。如下图所示。



7.汇编语言的指令组成

汇编指令：机器码的助记符，有对应的机器码。

伪指令：没有对应的机器码，由编译器执行，计算机并不执行。

其他符号：如+、-、\*、/等，由编译器识别，没有对应的机器码。

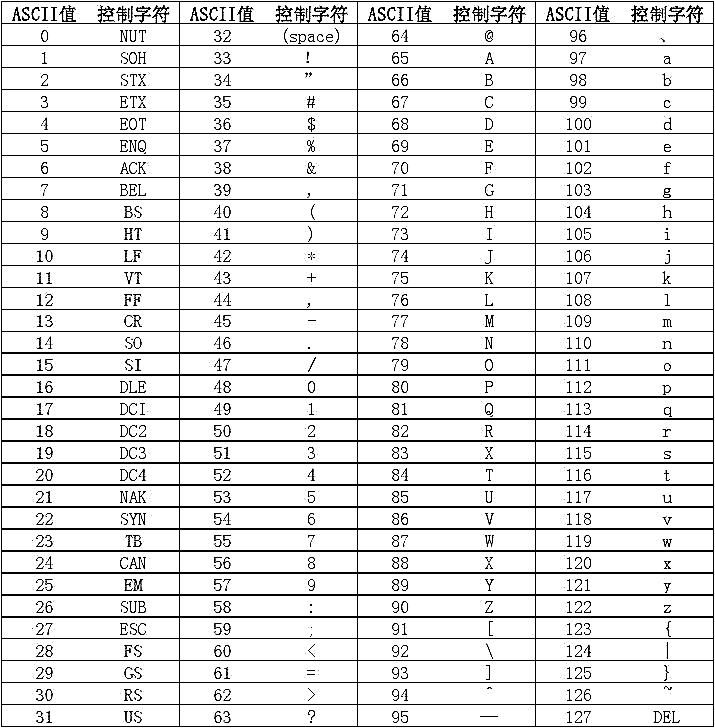
8.二进制、十进制以及十六进制数的转换。

|  |  |  |
| --- | --- | --- |
| 二进制数 | 十进制数 | 十六进制数 |
| 0000 | 0 | 0 |
| 0001 | 1 | 1 |
| 0010 | 2 | 2 |
| 0011 | 3 | 3 |
| 0100 | 4 | 4 |
| 0101 | 5 | 5 |
| 0110 | 6 | 6 |
| 0111 | 7 | 7 |
| 1000 | 8 | 8 |
| 1001 | 9 | 9 |
| 1010 | 10 | A |
| 1011 | 11 | B |
| 1100 | 12 | C |
| 1101 | 13 | D |
| 1110 | 14 | E |
| 1111 | 15 | F |

9.补码表示法如何表示机器数

补码表示法中正数采用符号-绝对值表示，即数的最高有效位为0表示符号为正，数的其余部分则表示数的绝对值。例如，假设机器字长为8位，则[+1]补=00000001，[+127]补=01111111，[+0]补=00000000。而对于负数，先写出与该负数相对应的正数的补码表示，然后将其按位求反，最后在末位加1，就可以得到该负数的补码表示了。

10.常用字符的7位ASCII值



11.四种基本逻辑运算是什么

与运算、或运算、非运算和异或运算。如有A、B两个逻辑变量（每个变量只能有0或1两种取值），在各种取值条件下得到的逻辑运算结果如下表所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *A* | *B* |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

**二、80x86计算机组织**

1.程序可见寄存器的分类。

程序可见寄存器可以分为通用寄存器、专用寄存器和段寄存器三类。

2.通用寄存器有哪些？各自有什么作用？

通用寄存器包括AX、BX、CX、DX、SP、BP、SI、DI。其中AX、BX、CX、DX用来暂时存放计算过程中所用到的操作数、结果或其他信息，也可称为数据寄存器；SP、BP、SI、DI可以在存储器寻址时，提供偏移地址，也被称为指针或变址寄存器。

3. AX、BX、CX、DX四个寄存器的具体作用

AX：可分为AH和AL，并且作为累加器使用，所以是算术运算的主要寄存器。在乘、除等指令中指定用来存放操作数。另外，所有的I/O指令都使用这一寄存器与外部设备传送信息。

BX：可分为BH和BL，作为通用寄存器使用，此外在计算存储器地址时，它经常用作基址寄存器。

CX：可分为CH和CL，作为通用寄存器使用，此外常用来保存计数值，如在移位指令、循环和串处理指令中用作隐含的计数器。

DX：可分为DH和DL，作为通用寄存器使用。一般在做双字长运算时把DX和AX组合在一起存放一个双字长数，DX用来放高位字。此外，对某些I/O操作，DX可用来存放I/O的端口地址。

4. SP、BP、SI、DI四个寄存器的具体作用。

SP：堆栈指针寄存器，用来指示段项的偏移地址。

BP：基址指针寄存器，可作为堆栈区中的一个基地址以便访问堆栈中的信息，也可以与堆栈段寄存器SS联用来确定堆栈段中的某一存储单元的地址。

SI：源变址寄存器，一般与数据段寄存器DS联用，用来确定数据段中某一存储单元的地址。在串处理指令中，SI和DS联用，达到在数据段中寻址的目的。

DI：目的变址寄存器，一般与数据段寄存器DS联用，用来确定数据段中某一存储单元的地址。在串处理指令中，DI和附加段寄存器ES联用，达到在附加段中寻址的目的。

5.32位的通用寄存器有哪些？

EAX、EBX、ECX、EDX、ESP、EBP、EDI和ESI。

6.专用寄存器有哪些？它们分别有什么作用？

专用寄存器包括IP、SP和FLAGS三个16位寄存器。其中IP为指令指针寄存器，用来存放代码段中的偏移地址，在程序运行的过程中，它始终指向下一条指令的首地址。SP为堆栈指针寄存器，它与堆栈段寄存器联用来确定堆栈段中的栈顶地址，也就是用来存放栈顶的偏移地址。FLAGS为标志寄存器，又称程序状态寄存器，这是一个存放条件码标志、控制标志和系统标志的寄存器。

7.EFLAGS中主要标志位及其含义。

（1）条件码标志位。

用来记录程序中运行结果的状态。包括以下六位：

溢出标志（OF）：在运算过程中，如操作数超出了机器能表示的范围称为溢出，此时OF置1，否则置0。

符号标志（SF）：记录运算结果的符号，结果为负时置1，否则置0。

零标志（ZF）：运算结果为0时ZF置1，否则置0。

进位标志（CF）：记录运算时从最高有效位产生的进位值。最高有效位有进位时置1，否则置0。

辅助进位标志（AF）：记录运算时第三位产生的进位值，有进位时置1，否则置0。

奇偶标志（PF）：用来为机器中传送信息时可能产生的代码出错情况提供检验条件。当结果操作数中1的个数为偶数时置1，否则置0。

（2）控制标志位。

方向标志（DF），在串处理指令中控制处理信息的方向。

（3）系统标志位。

用于I/O、可屏蔽中断、程序调试、任务切换和系统工作方式等的控制。包括陷阱标志（TF）、中断标志（IF）、I/O特权级（IOPL）等。

8.测试标志位时，标志位的符号表示。

|  |  |  |
| --- | --- | --- |
| 标志名 | 标志为1 | 标志为0 |
| OF 溢出（是/否） | OV | NV |
| DF 方向（减量/增量） | DN | UP |
| IF 中断（允许/关闭） | EI | DI |
| SF 符号（负/正） | NG | PL |
| ZF 零（是/否） | ZR | NZ |
| AF 辅助进位（是/否） | AC | NA |
| PF 奇偶（偶/奇） | PE | PO |
| CF 进位（是/否） | CY | NC |

9.段寄存器有哪些？它们的作用是？

段寄存器专用于存储器寻址，用来直接或间接地存放段地址。包括代码段（CS）、数据段（DS）、堆栈段（SS）和附加段（ES）。从80386起，增加了FS和CS两个段寄存器，它们也属于附近的数据段。

10.（未完，待整理）

参考资料：

[汇编语言（面向机器的程序设计语言）\_百度百科 (baidu.com)](https://baike.baidu.com/item/%E6%B1%87%E7%BC%96%E8%AF%AD%E8%A8%80/61826)

[汇编语言程序设计\_电子科技大学\_中国大学MOOC(慕课) (icourse163.org)](https://www.icourse163.org/course/UESTC-1002047009?tid=1468253441)

[Assembly 基础介绍\_w3cschool](https://www.w3cschool.cn/assembly/wyxd1s2a.html)

《汇编语言（第3版）》王爽

《IBM-PC汇编语言程序设计（第2版）》沈美明 温冬婵