※ABeen※ 汇编语言

学习摘要

一、 汇编语言计算机基础

从事计算机科学方面的工作，汇编语言是我们必不可少缺的基础。我们的工作平台、研究对象都是机器，而汇编语言是人和计算机沟通的最直接方式，它描述了机器最终所要执行的指令序列。学习汇编语言可以让你充分获得底层编程的体验，深刻理解机器运行程序的机理。

二、 基础知识

1. 汇编语言的产生

计算机从本质上来讲，就是中央处理器(CPU）连接一堆外设。CPU 是计算机的核心部件，它控制整个计算机的运作并进行运算。要想 CPU 工作就必须提供指令和数据。这些指令和数据在存储器(严格来说应该是逻辑存储器)中存放，也就是平台我们所说的内存。指令和数据是应用上的概念，在内存或磁盘上，它们没有区别都是二进制信息。CPU 在工作的时候把有的信息看作指令，有的信息看作数据，为同样的信息赋予了不同的意义。

只要 CPU 一加电，它就从预设的地址开始一直执行下去。在执行程序的时候 CPU 是从 CS:IP 指向的某个地址开始，自动向下读取指令执行。CPU 读取的指令就是机器语言。

机器语言是机器指令的集合。机器指令就是一台机器可以正确执行的命令。电子计算机的机器指令是一列二进制数字。计算机将之转变成一列高低电平，从而使计算机的电子器件受到驱动，进行运算。由于机器指令难于辨别和记忆，给整个产业的发展带来了障碍，于是汇编语言产生了。汇编语言的主体是汇编指令，汇编指令和机器指令是一一对应的，汇编指令是机器指令便于记忆的书写格式。

汇编语言的核心是汇编指令，它决定了汇编语言的特性。

汇编语言主要由 3 类指令组成。

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。

2. 存储单元和总线

存储器被化分成若干个存储单元，每个存储单元从 0 开始编号。电子计算机的最小信息单位是 bit,也就是一个二进制位，8 个 bit 组成一个 Byte 字节,微机存储器的容量是以字节为最小单位计算的。

CPU 要从编号的存储单元中读取或写入数据，必须进行下面 3 类信息的交互。

- 存储单元的地址(地址信息);
- 器件的选择，读或写的命令(控制信息);
- 读或写的数据(数据信息);

电子计算机处理、传输的信息都是电讯号，电信号是用导线传送的，专门连接 CPU 和其他芯片的导线，称为总线。从物理上来讲，总线就是一根根导线的集合。根据传送信息的不同，总线从逻辑上分为：地址总线、控制总线、数据总线。

如 CPU 从 3 号单元中读取数据的过程

- CPU 通过地址总线将地址信息 3 发出
- CPU 通过控制线发出内存读命令，选中存储器芯片，并通知它将要从中读取数据
- 存储器将 3 号单元中的数据，通过数据总线传入 CPU

➢ 地址总线和寻址能力

CPU 是通过地址总线来指定存储器单元的，可见地址总线上能传送多少个不同的信息，CPU 就可以对多少个存储单元进行寻址。在计算机中一根导线可以传送和稳定状态只有两种：高电平或是低电平。用二进制表示就是 1 或 0。

所以一个 CPU 有 N 要地址线，则可以说这个 CPU 的地址总线的宽度为 N, 最多可以寻找 2 的 N 次方个内存单元。

➢ 数据总线

CPU 与内存或其他器件之间的数据传送是通过数据总线传送的。数据总线的宽度决定了 CPU 和外界的数据传送速度。8 根数据总线一次可传送一个 8 位二进制数，１６根数据总线则可传两个字节。

➢ 控制总线

CPU 对外部器件的控制是通过控制总线进行的。所以有多少要控制总线，就意味着 CPU 提供了对外部器件的多少种控制。控制总线决定了 CPU 对外部器件的控制能力。

3. 内存地址空间

一台 PC 机中，装有多个存储器芯片，从读写属性上分为：随机存储器(RAM)和只读存储器(ROM). 如主板上的 RAM 和插在扩展槽上的 RAM、显卡、网卡等都有相应的存储器芯片。这些芯片都和 CPU 总线相连，CPU 对它们进行读或写的时候都是通过控制线发出的命令。可以说，CPU 把它们全当作内存来对待，把它们总的看作一个由若干存储单元组成的逻辑存储器，这个逻辑存储器就是我们所说的内存地址空间。于是，每个物理存储器在这个逻辑存储器中占有一个地址段，既一段地址空间。CPU 对这段地址空间中读写数据，实际上就是在对相应的物理存储器中读写数据。

内存地址空间的大小受 CPU 地址总线宽度的限制，若地址总线宽度为 20，则可以定位 2 的 20 次方个内存单元，内存地址空间大小为 1M, 若地址总线宽度为 32,则内存地址空间最大为 4GB。现在明白为什么 XP 系统不能完成利用 4GB 内存空间了吧。

4. 物理地址与寄存器

存储器被化分成若干个存储单元，每个存储单元从 0 开始编号，所用的内存单元构成的存储空间是一个一维的线性空间，每一个内存单元在这个这间中都有唯一的地址，这个地址称为物理地址。

CPU 通过地址总线送入存储器的，必须是一个内存单元的物理地址。所以 CPU 在发出物理地址之前必须先在内部形成这个物理地址，不同的 CPU 可以有不同的形成方式。对于 16 位结构的 CPU 来说，若地址总线为 20 位，可以传送 20 位地址，达到 1MB 寻址能力。但 16CPU 表现出的寻址能力只有 64KB。为了得到更大的寻址能力，有的 CPU 在内部用两个 16 位地址合成的方法来形成一个 20 位的物理地址。

比如 8086CPU 要读写内存时：

- CPU 中的相关部件提供两个 16 位地址，一个称为段地址，另一个称为偏移地址；

- 段地址和偏移地址通过内部总线送入一个称为地址加法器的部件；
- 地址加法器将两个 16 位地址合成为一个 20 位的物理地址；
- 地址加法器通过内部总线装 20 位物理地址送入输入输出控制电路；
- 输入输出控制电路将 20 位物理地址送上地址总线；
- 20 位物理地址被地址总线传送到存储器。

地址加法器采用 **物理地址＝段地址 * 16 + 偏移地址** 的方法用段地址和偏移地址合成物理地址。

注意段地址并不是说内存被分成许多小段了，这里段地址是对于 CPU 来说的，由于地址加法器采用 物理地址＝段地址 * 16 + 偏移地址的方法来表示物理地址，我们在编程的时候可以，将若干地址连续的内存单元看作一个段，用段地址*16 定位段的起始地址，用偏移地址定位段中的内存单元。

注意：

1、段地址*16 肯定是 16 的倍数，所以一个段的起始地址也一定是 16 的倍数；

2、偏移地址为 16 位，16 位寻址能力为 2 的 16 次方为 64KB,所以一个段的长度最大为 64KB.

现在你可以想象，程序在内存中是如何布局的？

为什么好多工具分析时会有程序基地址、偏移地址等？

CPU 是怎么区分数据和指令的呢？CPU 内部有好多寄存器。

比如：

- CS 代码段寄存器，IP 为指令指针寄存器,任意时刻，CPU 将 CS:IP 指向的内容当作指令执行。
- DS 数据段寄存器，DS:[0]指向 DS*16+0 的内存单元。
- SS 栈段寄存器，SP 存栈偏移地址，任意时刻 SS:SP 指向栈顶元素。

5. 栈

栈空间也是内空间的一部分，它只是一段可以以一种特殊方式进行访问的内存空间。其特殊性在于，最后进入这个空间的数据，最选出去。栈有两个基本操作：入栈和出栈。栈的操作规则被称为：LIFO(Last In First Out，后进先出).

现今的 CPU 都有栈的设计，8086CPU 也提供相关的指令来以栈的方式访问内存空间。我们编程的时候，可以将一段内存当作栈来使用。基本操作为 PUSH 和 POP。那么 CPU 怎么会知道我们想把哪段空间当栈使用呢？CPU 中有两个寄存器，段寄存器 SS 和寄存器 SP,栈顶的段地址放在 SS 中，偏移地址放在 SP 中。任意时刻，SS:SP 指向栈顶元素。那么 PUSH 和 POP 执行进自然是从 SS:SP 得到栈顶的地址。

三、 汇编指令

高级语言经编译器编译生成后,最终会生成机器语言,汇编指令与机器语言又是对应的。所以汇编指令与高级语言语言定义存在一定的关系。

如：CALL、RET 组合相当于方法调用。
LOOP、LOOPE、LOOPZ、LOOPNE、LOOPNZ、JCXZ、JECXZ 循环执行指令。
INT、INTO、IRET 中断，与并发执行和调试等有关。

四、 中断

任何一个通用的 CPU,都具备一种能力，可以在执行完当前正在执行的指令后，检测到从 CPU 内部或外部送过来的一种特殊信息，并且可以立即对所接收到的信息进行处理。这种特殊的信息，称为中断信息。中断的意思是,CPU 不再接着(刚执行完的指令)向下执行，而是转去处理这个特殊的信息。中断信息是要求 CPU 马上进行某种处理，并向所要进行的该处理提供了必备的通知信息。中断信息可以来自 CPU 内部和外部。

内中断的产生

1. 除法错误
2. 单步执行
3. 执行 into 指令
4. 执行 int 指令

当 CPU 收到中断信息后，从中提取中断类型码，利用中断类型码在中断向量表中找到中断处理程序的入口地址，然后用入口地址设置 CS 和 IP,使 CPU 执行中断处理程序。用中断类型码找到中断向量，并用它设置 CS 和 IP, 这个工作是由 CPU 的硬件自动完成的，这个过程称为**中断过程**。中断向量表指定放在内存地址 0 处，从内存 0000:0000 到 0000：03FF

的 1024 个单元中存放，这是规定。

8086CPU 中断过程如下

1. (从中断信息中)取得中断类型码
2. 把标志寄存器入栈(中断过程要改变标志位，为了恢复现场)
3. 设置标志寄存器的第 8 位 TF 和第 9 位 IF 的值为 0
4. CS 入栈
5. IP 入栈
6. 从内存地址为中断类型码*4 和中断类型码*4+2 的两个字单元中读取中断处理程序的入口地址设置 IP 和 CS;

五、 计算机开机运行

现在我们知道了计算机是怎么样运行、区分指令和数据、读写指令和数据的。那么计算机中的程序是怎么运行的呢？

在系统主板的 ROM 中存放着一套程序，称为 BIOS(基本输入输出系统)，主要包含以下几部分内容。

1. 硬件系统的检测和初始化程序
2. 外部中断和内部中断的中断例程
3. 用于硬件设备进行 I/O 操作的中断例程
4. 其他和硬件系统相关的中断例程

操作系统也提供了中断例程，从操作系统来看，操作系统的中断例程是向程序员提供的编程资源。

1. 开机后，CPU 一加电，初始化 CS=0FFFFH,IP=0,自动从 FFFF:0 单元开始执行程序。FFFF:0 处有一条跳转指令，转去执行 BIOS 中的硬件系统检测和初始化程序。
2. 初始化程序将建立 BIOS 所支持的中断向量，将 BIOS 提供的中断例程入口地址登记在中断向量表中。这里只需要将入口地址登记在中断向量表中即可，不需要考程序数据，因为它们是固化到 ROM 中的程序，一直在内在中存在。
3. 硬件系统检测和初始化完成后，调用 int 19h 进行操作系统的引导。从此将计算机

交由操作系统控制。

4. 操作系统启动后，除完成其他工作外，还将它所提供的中断例程装入内存，并建立相应的中断向量。

六、 程序实例分析说明