第1章

汇编语言基础知识

教学重点

第1章是用汇编语言进行程序设计所需要了解的基本知识。

重点掌握几个内容：

§认识汇编语言

§数据表示

§基本位操作

程序设计语言

§ 机器语言

§ 汇编语言

§ 高级语言

什么是汇编语言（1）

§ 汇编语言是一种面向机器的低级程序设计语言

§ 汇编语言以助记符形式表示每一条指令

- Ø 助记符（mnemonic）是便于人们记忆、并能描述指令功能和指令操作数的符号
- Ø 助记符一般就是表明指令功能的英语单词或其缩写

什么是汇编语言（2）

- 用助记符表示的指令就是汇编语言中的**汇编格式指令**
- **汇编格式指令**以及使用它们**编写程序的规则**就形成**汇编语言**（Assembly Language）
- 用汇编语言书写的程序就是**汇编语言程序**，或称汇编语言源程序（.ASM）
- **汇编程序**将汇编语言程序“汇编”成机器代码目标模块（.OBJ）

汇编语言程序与汇编程序是两个概念

什么是汇编语言（3）

§ 汇编语言的主要特点：

Ø 汇编语言程序与处理器指令系统密切相关

Ø 程序员可直接、有效地控制系统硬件

Ø 形成的可执行文件运行速度快、占用主存容量少

汇编语言和高级语言（1）

§ 汇编语言与处理器密切相关

↘ 汇编语言程序的通用性、可移植性较差

§ 高级语言与具体计算机无关

↗ 高级语言程序可以在多种计算机上编译后执行

汇编语言：✗

高级语言：√

汇编语言和高级语言（2）

§ 汇编语言功能有限、涉及硬件细节

↘ 编写程序比较繁琐，调试起来也比较困难

§ 高级语言提供了强大的功能，不必关心琐碎问题

↗ 类似自然语言的语法，易于掌握和应用

汇编语言：✗

高级语言：√

汇编语言和高级语言（3）

§ 汇编语言本质上就是机器语言

↗ 可以直接、有效地控制计算机硬件

↗ 易于产生速度快、容量小的高效率目标程序

§ 高级语言不针对具体计算机系统

↘ 不易直接控制计算机的各种操作

↘ 目标程序比较庞大、运行速度较慢

汇编语言：√

高级语言：×

汇编语言和高级语言（4）

§ 汇编语言的优点：

Ø 直接控制计算机硬件部件

Ø 可以编写在“时间”和“空间”两方面最有效的程序

§ 汇编语言的缺点：

Ø 与处理器密切有关

Ø 需要熟悉计算机硬件系统、考虑许多细节

Ø 编写繁琐，调试、维护、交流和移植困难

汇编语言：？

高级语言：？

汇编语言和高级语言（5）

- 汇编语言的优点使得它在程序设计中占有重要的位置，是不可被取代的
- 汇编语言的缺点使得人们主要采用高级语言进行程序开发工作
- 有时需要采用高级语言和汇编语言混合编程的方法，互相取长补短，更好地解决实际问题

混合编程　　取长补短

汇编语言的意义

§ 速度：对于同一个问题，用汇编语言设计出的程序能达到“运行速度最快”。

§ 空间：对于同一个问题，用汇编语言设计出的程序能达到“占用空间最少”。

§ 功能：汇编语言可以实现高级语言难以胜任甚至不能完成的任务。

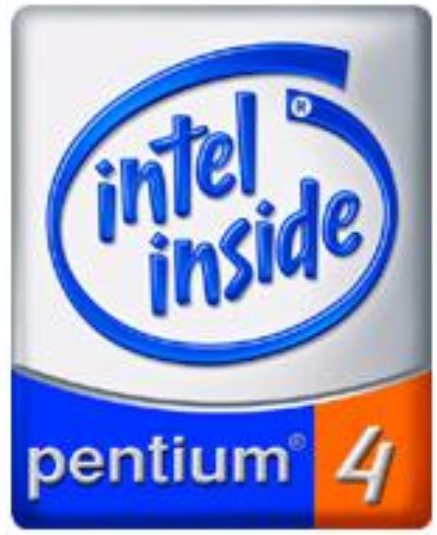
§ 知识：学习汇编语言，有助于对计算机系统的理解、写出更好的程序。

汇编语言的应用场合

- 程序要具有较快的执行时间，或者只能占用较小的存储容量
- [illegible]
- [illegible]
- 没有合适的高级语言、或只能采用汇编语言的时候
- 分析具体系统尤其是该系统的低层软件、加密解密软件、分析和防治计算机病毒等等

汇编语言的作用实在不小！

微处理器飞速发展



Sandy Bridge
Core i3
Core i5
Core i7
IA-64（安腾）
奔腾4代
奔腾3代
奔腾2代
奔腾
80386
80486
80286
8086
4004

不是我不明白，
这世界变化太快。
扎扎实实掌握知识，
以不变应万变！

教学重点

第1章是用汇编语言进行程序设计所需要了解的基本知识。

重点掌握几个内容：

§认识汇编语言

§数据表示

§基本位操作

数据表示

§ 数制的基本知识

- 10进制
- 2进制
- 16进制

§ 说明：

- 前导0可以忽略，不影响取值。
- 结尾用D（10进制数）、B（2进制数）、H（16进制数）。缺省为十进制数。

数据组织

位、字节、字、双字

§ 位（Bit）：1个二进制位。

计算机是在特定位数下工作的，如8位、16位、32位等。

§ 字节（Byte）：8位。

位编号从右到左为0～7，第0位为最低位，第7位为最高位。

数据组织

§ 字（Word）：16位。

位编号从右到左为0～15，第0位为最低位，第15位为最高位。位0～7为低字节，位8～15为高字节。

§ 双字（Double Word）：32位。

位编号从右到左为0～31，第0位为最低位，第31位为最高位。位0～15为低字，位16～31为高字。

带符号数与无符号数（1）

1．带符号数的补码表示

补码的表示规则：

§ 以最高位作为符号位（0表示正数，1表示负数）。

§ 正数的补码是其本身。

§ 负数的补码是对其正数“各位求反、末位加1”后形成的。

N+1位二进制补码数可以表示的带符号数范围为-2^N～2^N-1。

例如，8位二进制数可以表示 -128～127，16位二进制数可以表示-32768～32767。

带符号数与无符号数（2）

2.补码的特性

求补

$[x]_{补} \Longleftrightarrow [-x]_{补}$

$[x + y]_{补} = [x]_{补} + [y]_{补}$　　$[x - y]_{补} = [x]_{补} + [-y]_{补}$

说明：

§ 在计算机内部，补码减法是通过对减数求补后将减法转换为加法进行的。

§ 一个带符号数在不同位数下，其二进制补码表示可能是不同的。例如，8位数-1的补码表示是0FFH，16位数-1的补码表示是0FFFFH。

带符号数与无符号数（3）

3. 符号扩展与零扩展

§ 符号扩展是将原符号位填入扩展的每一位，使得在带符号数意义下取值不变。

§ 零扩展是将0填入扩展的每一位，使得在无符号数意义下取值不变。

4. 无符号数

N位二进制数可以表示的无符号数范围为0～2^N -1。例如，8位二进制数00H～0FFH表示0～255，16位二进制数0000H～0FFFFH表示0～65535。

字符的ASCII码表示

§ ASCII码字符集采用一个字节表示字符。

§ 常用字符的ASCII码。

数字'0'～'9'：30H～39H

字母'A'～'Z'：41H～5AH

字母'a'～'z'：61H～7AH

空格：20H　　回车CR：0DH

换行LF：0AH

§ 注意回车与换行的差别：

CR用来控制光标回到当前行的最左端；

LF用来移动光标到下一行，而所在列不变。

BCD码

§ 压缩BCD码

以4个二进制位表示1个十进制位，用0000B～1001B表示0～9。 例如，十进制数6429的压缩BCD码表示为

0110 0100 0010 1001 B（即6429H）

§ 非压缩BCD码

以8个二进制位表示1个十进制位，低4位与压缩BCD码相同，高4位无意义。 例如，十进制数6429的非压缩BCD码表示为

xxxx0110 xxxx0100 xxxx0010 xxxx1001 B

有时，要求非压缩BCD码的高4位为0，这时，6429的非压缩BCD码为06040209H。

注解

§ 同一个二进制数可以表示多种含义，其具体含义由使用者解释。

例如，二进制数00110000B，即30H，可以当作十进制数48的二进制表示，字符'0'的ASCII码，30的压缩BCD码，0的非压缩BCD码，等等。甚至将其当作现实世界的任一物理对象也未尝不可。

§ 带符号数的二进制补码表示与位数密切相关。

例如0FFH，若作为8位带符号数，则表示-1；若作为16位带符号数，则表示255。再如0FFFFH，若作为16位带符号数，则表示-1；若作为32位带符号数，则表示65535。

教学重点

第1章是用汇编语言进行程序设计所需要了解的基本知识。

重点掌握几个内容：

§认识汇编语言

§数据表示

§基本位操作

基本位操作

1. 逻辑操作：AND、OR、XOR、NOT

§ AND操作可以使某些位清0。

24H AND F0H=20H

§ OR操作可以使某些位置1。

20H OR 0FH=2FH

§ XOR操作可以使某些位取反。

25H XOR 25H=00H

基本位操作

2、移位与循环移位

§ 左移：最低位移入0。在不溢出的情况下，左移1位相当于乘以2。

§ 逻辑右移：最高位移入0。逻辑右移1位约等于无符号数除以2。

§ 算术右移：最高位不变。算术右移1位约等于带符号数除以2。

§ 循环左移与循环右移：从一端移出的位要移入到另一端。

本章小结

§ 汇编语言是机器语言的符号表示，与机器语言无本质区别。

§ 现代计算机系统使用2进制表示数据。为了描述方便，书写时常采用16进制形式。

§ 以补码表示的带符号数在基于不同位数时，其二进制形式可能完全不同。

§ 二进制只是一种表示，一个二进制数的具体含义由使用者解释。

§ AND、OR、NOT和XOR等二进制逻辑操作，可用于有选择地对某些位进行处理。