汇编语言

寻址方式

立即寻址方式Immediate addressing

操作数直接存放在指令中，紧跟在操作码直后，它作为指令的一部分存放在代码段里，这种操作数称为立即数。

MOV AL， 5

MOV AX， 3064H

寄存器寻址方式Register addressing

操作数在寄存器中，指令指定寄存器号。

MOV AX， BX

直接寻址方式Direct addressing

有效地址EA(Effective Address)在指令中，它存放在代码段中指令操作码之后，但操作数一般存放在数据段中，所以必须先求出操作数的物理地址，然后再访问存储器才能取得操作数。

MOV AX,[2000H]

MOV AX,ES:[2000H]

寄存器间接寻址方式Register indirect addressing

操作数的有效地址在基址寄存器BX，BP或变址寄存器SI，DI中，而操作数则在存储器中。

DS:[BX], DS:[SI], DS:[DI]

SS:[BP]

MOV AX,[BX]

MOV AX,ES:[BX]

寄存器相对寻址方式Register relative addressing

直接变址寻址方式

操作数的有效地址是一个基址或变址寄存器的内容和指令中指定的位移量displacement之和。

MOV AX,3000H[SI]

MOV AX,[3000H+SI]

MOV AX,ES:3000H[SI]

基址变址寻址方式Based indexed addressing

操作数的有效地址是一个基址寄存器和一个变址寄存器的内容之和，两个寄存器均由指令指定。

MOV AX，[BX][DI]

MOV AX,[BX+DI]

段内直接寻址

JMP NEAR PTR LABEL位移量=16

JMP SHORT LABEL位移量=8

段内间接寻址

JMP BX

JMP WORD PTR [BX+TABLE]

段间直接寻址

JMP FAR PTR LABEL

段间间接寻址

JMP DWORD PTR [n + BX]

选择编译器

nasm？fasm？yasm？还是masm、gas或其他？

前面三个是免费开源的汇编编译器，总体上来讲都使用Intel的语法。yasm是在nasm的基础上开发的，与nasm同宗。由于使用了相同的语法，因此nasm的代码可以直接用yasm来编译。

yasm虽然更新较慢，但对nasm一些不合理的地方进行了改良。从这个角度来看，yasm比nasm更优秀些，而nasm更新快，能支持更新的指令集。在Windows平台上，fasm是另一个不错的选择，平台支持比较好，可以直接用来开发Windows上的程序，语法也比较独特。在对Windows程序结构的支持上，fasm是3个免费的编译器里做得最好的。

masm是微软发布的汇编编译器，现在已经停止单独发布，被融合在Visual Studio产品中。gas是Linux平台上的免费开源汇编编译器，使用AT&T的汇编语法，使用起来比较麻烦。

masm

MASM是微软公司开发的汇编开发环境，拥有可视化的开发界面，使开发人员不必再使用DOS环境进行汇编的开发，编译速度快，支持80x86汇编以及Win32Asm是Windows下开发汇编的利器。它与windows平台的磨合程度非常好，但是在其他平台上就有所限制，使用MASM的开发人员必须在windows下进行开发。

masm32

MASM32 是一个为那些对学习或者编写 32 位微软汇编（ MASM ）感兴趣 的程序员提供的工作环境，它最流行的版本是 8.0 版，而我所知道最新版本是 9.0 版。但是请大家一定要注意，这里的 8.0 版和 9.0 版不是指 MASM 编译器 ( 即 ML) 的版本，而是 MASM32 开发环境的版本。

那么 MASM(ML) 编译器的版本到底有哪些呢？教学上常用的版本是 5.0 版的， 5.0 版之后有个 5.1 版本，在这之后，从 6.0 版开始 MASM 就改名 ML 了，因为它把编译和连接组合在一起了。而且 6.0 版还对汇编语言进行了大幅改造，使之支持高级语言的很多特性。 6.0 版之后常看到的就是 6.11 、 6.13 、 6.14 MASM32(V8.0 和 V9.0) 就是包含有 ML6.14 版 , 而不是 ML8.0 版。 6.14 之后就是 6.15

但不常见 , 我手头上有 6.15 的 ML 。但 Microsoft 从 6.15 版开始好象就不再单独发布 MASM 的开发工具包了。

以后的 ML 还有 7.0 、 7.1 版，最新的是 ML8.0 版。但它们都只包含在 Visual Studio.Net 和 Visual Studio 2005 里面。而且如果你 安装了 Visual Studio 2005 的 64 位开发环境，你将得到一个MASM for x64 ( ml64 .exe) 哈哈，各位这可是支持 Intel 和 AMD 的 64 位 CPU 的汇编语言编译器！有志于 64 位底层开发的赶快去试用看看吧。

nasm fasm yasm

这三个都是免费开源的汇编编译器，总体上都是采用的Intel的语法。yasm是从nasm的基础上开发出来的，属于同宗，使用了相同的语法，所以nasm的代码可以用yasm编译。

yasm虽然更新较慢，但对nasm一些不合理的地方进行了改良。从这个角度来看，yasm比nasm更优秀些，而nasm更新快，能支持更新的指令集，而且还支持Gas语法和AMD64（EM64T）架构，跨平台，支持多种目标文件格式。在Windows平台上，fasm是另一个不错的选择，平台支持比较好，可以直接用来开发Windows上的程序，语法也比较独特。在对Windows程序结构的支持上，fasm是3个免费的编译器里做得最好的。

yasm的官方地址：http://yasm.tortall.net/Download.html。使用很简单：将下载下来的yasm-1.2.0-win32.exe重命名为yasm.exe，放到D:\Program Files\Microsoft Visual Studio 9.0\VC\bin目录下即可。

nasm的官方地址：http://www.nasm.us/

gas

Gas即GNU assembler 是linux下的汇编编译器，使用的AT&T的语法。