

## 32 位汇编语言编程步骤演示：

<http://10.71.45.100/bhh/sum.rar>

①下载上述链接中的压缩包，解压缩后，里面有一个 `sum.asm` 是 32 位汇编程序的源代码；

<http://10.71.45.100/bhh/masm32.zip>

②下载上述链接中的压缩包，解压缩后，生成 `masm32` 文件夹，里面包含了 32 位汇编语言的编译程序；

③双击 `masm32\qeditor.exe` 运行 32 位汇编语言集成环境

④在集成环境中打开 `sum.asm`；

⑤点菜单 `project->build all` 进行编译，生成 `sum.exe`

⑥点菜单 `project->run program` 运行 `sum.exe`

## 16 位汇编与 32 位汇编的区别：

### (1) 宽度的区别

16 位汇编是在 `dos` 环境下的汇编语言；

32 位汇编是在 `windows` 或 `linux` 环境下的汇编语言；

16 位汇编的寄存器是 16 位的，地址也是 16 位的；

而 32 位汇编的寄存器是 32 位，地址也是 32 位。

### (2) 调用操作系统函数时有区别

例如在屏幕上输出一句话，32 位汇编可以调用

`MessageBox` 这个函数，通过弹框输出；

16 位汇编要调用 `int 21h` 的 9 号功能；

使用 16 位的寄存器及地址与使用 32 位的寄存器及地址在语法上差别不大，例如：

`mov ax, 1`；16 位寄存器

`mov eax, 1`；32 位寄存器

`mov word ptr ds:[1000h], 1234h`；16 位地址

`mov word ptr ds:[00401000h], 1234h`；32 位地址

## 16 位汇编语言编程步骤演示：

dosbox86 中如何对 h02.asm (<http://10.71.45.100/bhh/h02.asm>) 进行编译?

### 1. 集成环境编译

- (1) 先复制 h02.asm 到 dosbox86\masm
- (2) 运行 dosbox86
- (3) file->open->h02.asm
- (4) assemble->assemble 编译
- (5) assemble->link 连接
- (6) run->run
- (7) debug->user screen 查看运行结果

### 2. 命令行编译

file->dos shell

masm h02; 编译: 把 h02.asm 编译成 h02.obj

link h02; 连接: 把 h02.obj 连接成 h02.exe

h02            运行

exit           返回集成环境

如何调试汇编语言程序?

(1) 在集成环境中选菜单 debug 可以对 16 位汇编语言程序进行调试;也可以在命令行中输入以下命令可以对 h02.exe 进行调试:

td h02.exe

其中 td 的全名为 Turbo Debugger。

寄存器 ax 分成左 8 位(名为 ah)及右 8 位(名为 al)。

(2) windows 环境下的 32 位汇编语言程序用 qeditor 集成环境编译成 exe 后, 可以用 OllyDbg 打开该 exe 进行调试。

(3) linux 环境下的 32 位汇编语言程序可以用 gdb 调试:

演示程序: 把字符数组中的元素逐个取出并输出

(<http://10.71.45.100/bhh/outs.asm>)

源程序 `outs.asm` 中，以下语句

```
mov dl, s[bx]
```

编译后，变成：

`mov dl, [bx+0003]`；其中变量的段地址 `ds` 是隐含的  
这条机器语言指令其实等价于以下指令：

```
mov dl, ds:[bx+0003]
```

`ds:[bx+0003]` 表示指针 `ds:bx+0003` 所指的对象，其中 `ds` 是变量的段地址，`bx+0003` 是变量的偏移地址。

根据上述分析，只要程序中引用了 `data` 段内的变量，则必须在程序一开始就把 `ds` 赋值为 `data`，这样可以保证变量的段地址及偏移地址精确地指向该变量。