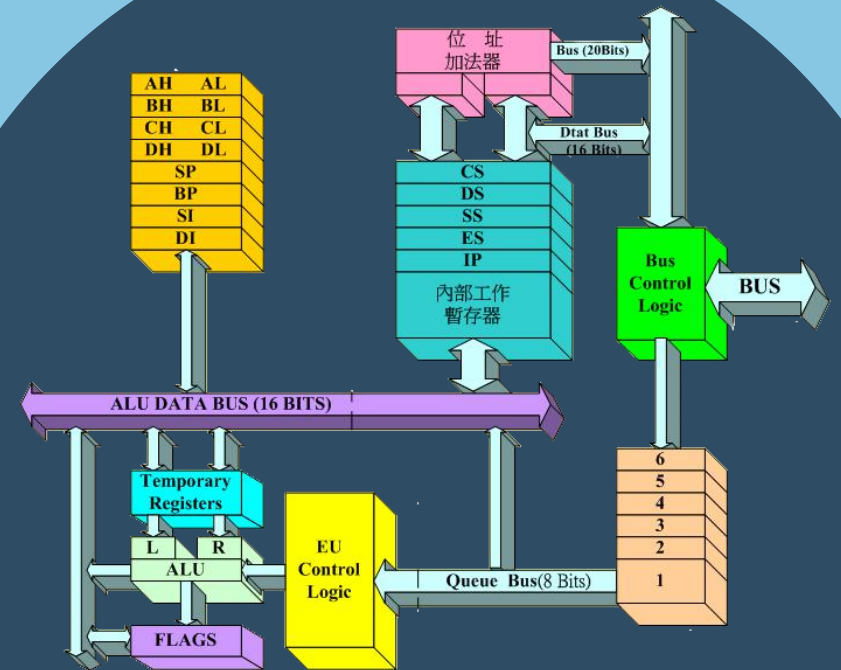


用汇编语言写的源程序

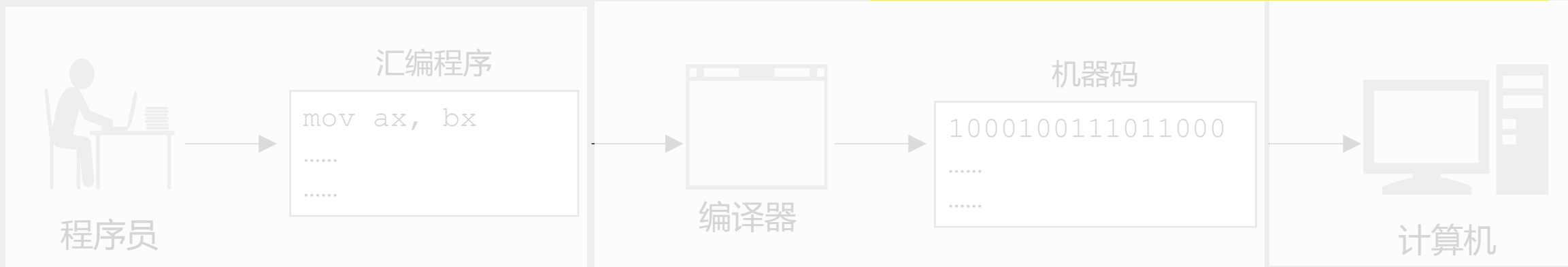
贺利坚 主讲



汇编语言程序设计
Assembly Language

用汇编语言编写程序的工作过程

汇编程序：包含汇编指令和伪指令的文本



```
assume cs:codesg
codesg segment
```

```
mov ax,0123H
mov bx,0456H
add ax,bx
add ax,ax
```

```
mov ax,4c00h
int 21h
```

```
codesg ends
end
```

汇编指令，对应
有机器码的指令，
可以被编译为机
器指令，最终被
CPU执行。

伪
指
令

伪指令

没有对应的机器码的指令，最终不被CPU所执行。

谁来执行伪指令呢？

伪指令是由编译器来执行的指令，编译器根据伪指令来进行相关的编译工作。

程序返回（套路！）：程序结束运行后，将CPU的控制权交还给使它得以运行的程序（常为DOS系统）。

程序中的三种伪指令

assume cs:codesg

codesg segment

mov ax,0123H

mov bx,0456H

add ax,bx

add ax,ax

mov ax,4c00h

int 21h

codesg ends

end

段定义

- 📄 一个汇编程序是由多个段组成的，这些段被用来存放代码、数据或当作栈空间来使用。
- 📄 一个有意义的汇编程序中至少要有有一个段，这个段用来存放代码。
- 📄 定义程序中的段：每个段都需要有段名
段名 **segment** ——段的开始
....
段名 **ends** ——段的结束

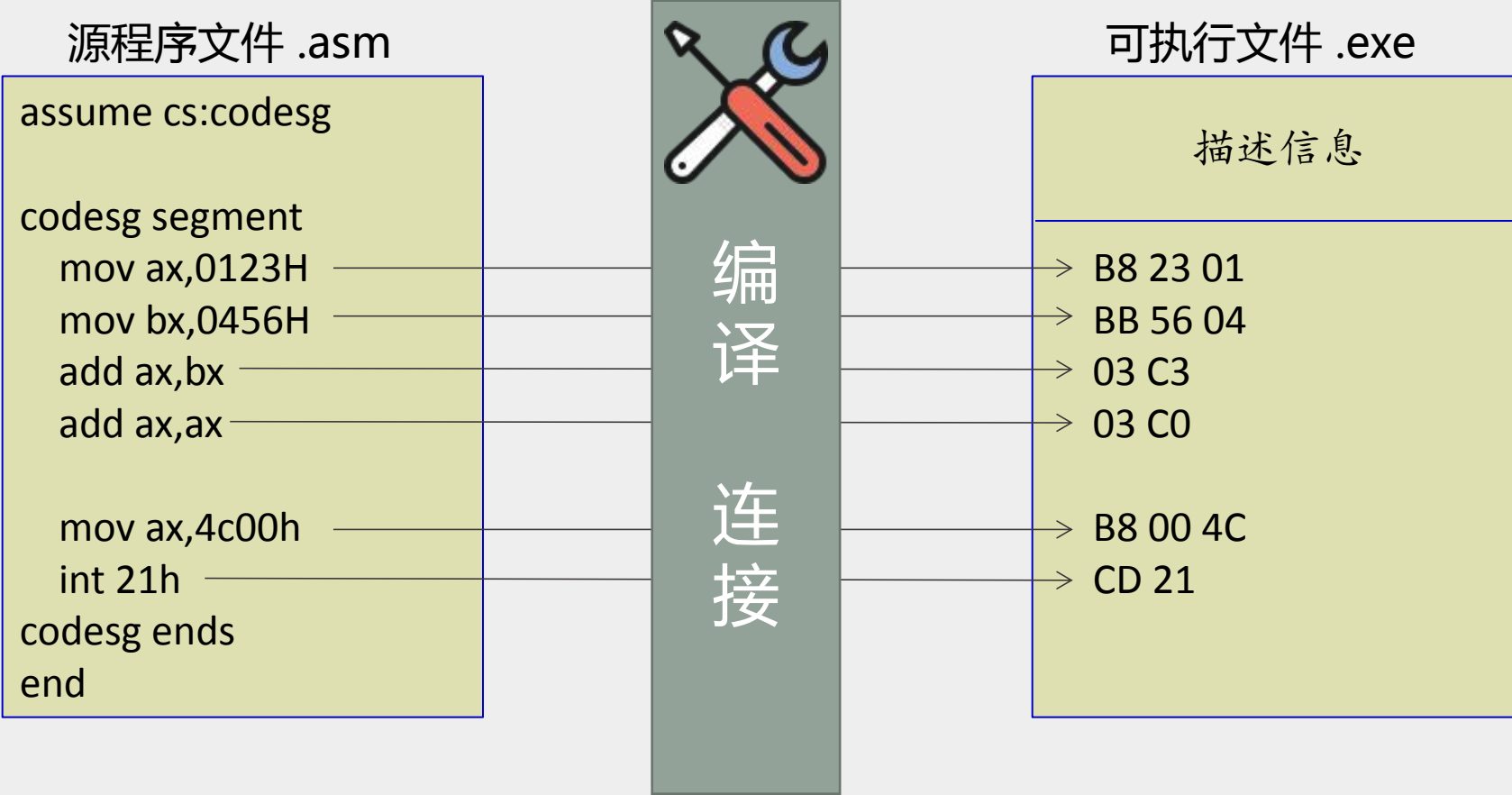
end (不是ends)

- 📄 汇编程序的结束标记。若程序结尾处不加end，编译器在编译程序时，无法知道程序在何处结束。


assume(假设)

- 📄 含义是假设某一段寄存器和程序中的某一个用 segment ... ends 定义的段相关联——assume cs:codesg指CS寄存器与codesg关联，将定义的codesg当作程序的代码段使用。

源程序经编译连接后变为机器码




汇编程序的结构


 在Debug中直接写入指令编写的汇编程序

```
C:\>debug
-a
073F:0100 mov ax, 27A1
073F:0103 add ax, 892C
073F:0106
```


 适用于功能简单、短小精悍的程序

 只需要包含汇编指令即可

 单独编写成源文件后再编译为可执行文件的程序

 适用于编写大程序

 需要包括汇编指令，还要有指导编译器工作的伪指令

 源程序由一些段构成，这些段存放代码、数据，或将某个段当作栈空间。

 ; ---注释

```
assume cs:code,ds:data,ss:stack
```

```
data segment
```

```
    dw 0123H,0456H,0789H,0abcH,0defH
```

```
data ends
```

```
stack segment
```

```
    dw 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
stack ends
```

```
code segment
```

```
    mov ax,stack
```

```
    mov ss,ax
```

```
    mov sp,20h    ;设置栈段
```

```
    mov ax,data
```

```
    mov ds,ax    ;设置数据段
```

```
    mov bx,0
```

```
    mov cx,8
```

```
s: push [bx]
```

```
    add bx,2
```

```
    loop s
```

```
    ....
```

```
code ends
```

```
end
```

如何写出一个程序来？

🖥️ 例：编程求 2^3 。

- ① 定义一个段
- ② 实现处理任务
- ③ 指出程序在何结束
- ④ 段与段寄存器关联
- ⑤ 加上程序返回的代码



```
abc segment
```

```
abc ends
```

①

```
abc segment
```

```
mov ax,2
```

```
add ax,ax
```

```
add ax,ax
```

```
abc ends
```

②

```
abc segment
```

```
mov ax,2
```

```
add ax,ax
```

```
add ax,ax
```

```
abc ends
```

```
end
```

③

```
assume cs:abc
```

```
abc segment
```

```
mov ax,2
```

```
add ax,ax
```

```
add ax,ax
```

```
abc ends
```

```
end
```

④

```
assume cs:abc
```

```
abc segment
```

```
mov ax,2
```

```
add ax,ax
```

```
add ax,ax
```

```
mov ax,4c00h
```

```
int 21h
```

```
abc ends
```

```
end
```

⑤

程序中可能的错误


语法错误

 程序在编译时被编译器发现的错误；

 容易发现下面程序中错误

```
aume cs:abc
abc segment
    mov ax,2
    add ax,ax
    add ax,sx
end
```

逻辑错误

 程序在编译时不能表现出来的、在运行时发生的错误；

 不容易发现下面程序中的错误

```
assume cs:abc
abc segment
    mov ax,2
    add ax,ax
    add ax,bx
    mov ax,4c10H
    int 21H
abc ends
end
```

求 2^3