第六章 多段程序

使用数据段

问题：编程计算以下8个数据的和，结果存在ax寄存器中。

```
0123h, 0456h, 0789h, 0abch, 0defh, 0fedh, 0cbah, 0987h
```

按照以学的知识在代码中应该有一下类型的数据。

```
mov ax, 0123h
add ax, 0456h
....
add ax, 0987h副了
```

非常的麻烦，可以使用数据段将这些数据存放在某一段内存空间中。

```
assume cs:code

code segment
        dw 0123h, 0456h, 0789h, 0abch, 0defh, 0fedh, 0cbah, 0987h ; 定义数据

        mov bx, 0
        mov ax, 0

        mob cx, 8
a:
        add ax, cs:[bx]
        add bx, 2 ; 处理的是字型数据，每次向后移动两个字节

```

```
        loop a

        mov ax, 4c00h
        int 21h
code ends
end
```

`dw`是`define word`的缩写，用来定义字型数据。位于程序的最开始的地方，也就是cs段寄存器执行的位置，此时我们可以使用cs作为他的段前缀。

但是以上程序有问题，就是第一行不是程序语句而是数据。在计算机内部，指令和数据都是二进制数据，所以会将数据作为指令来执行。(cs:ip指向这些数据)。

我们可以通过一下的方式修改ip初始的指向。

```
assume cs:code

code segment
        dw 0123h, 0456h, 0789h, 0abch, 0defh, 0fedh, 0cbah, 0987h ; 定义数据
start: ; 添加start标号，定位指令的入口
        mov bx, 0
        mov ax, 0

        mob cx, 8
a:
        add ax, cs:[bx]
        add bx, 2 ; 处理的是字型数据，每次向后移动两个字节
        loop a

        mov ax, 4c00h
        int 21h
code ends
end start ; 修改此处为 end start
```

简单程序的框架

```
assume cs:code
code segment
        定义数据
start:
        代码
code ends
end start
```

使用栈

定义栈的关键是要有一段内存空间可以u作为栈来使用。这和定义数据是一样的。

```
assume cs:code

code segment
        dw 0123h, 0456h, 0789h, 0abch, 0defh, 0fedh, 0cbah, 0987h ; 定义数据
        dw 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;定义栈，大小为16个字
start: ; 添加start标号，定位指令的入口
        mov ax, cs
        mov ss, ax
        mov sp, 30h ; 将栈顶设置为cs:30h

        mov ax, 4c00h
        int 21h
code ends
end start ; 修改此处为 end start
```

将数据，代码，栈放入不同的段

上面将数据，栈，代码全都放到了代码段中，看起来十分的拥挤，而且使用起来也不舒服。

通过下面的方式可以将数据，栈，代码放入不同的段中。

例： 将给定的八个数据倒转

使用栈可以非常容易的解决这个问题

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

data segment
        dw 0123h, 0456h, 0789h, 0abch, 0defh, 0fedh, 0cbah, 0987h
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
start:
        mov ax, stack
        mov ss, ax
        mov sp, 20h

        mov ax, data
        mov ds, ax

        mov bx, 0
        mov cx, 8
a:
        push [bx]
        add bx, 2
        loop s ; 将这八个数据入栈

        mov bx, 0
```

```
        mov cx, 8
b:
        pop [bx]
        add bx, 2
        loop b ; 将八个数据出栈，此时完成倒转

        mov ax, 4c00h
        int 21h
code ends
end start
```

多段程序的框架

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

data segment
        定义数据
data ends

stack segment
        定义栈空间
stack ends

code segment
start:
        mov ax, data
        mov ds, ax ; 数据段

        mov ax, stack
        mov ss, ax
        mov sp, xxh ; 栈段

        程序代码

        mov ax, 4c00h
        int 21h ; 程序返回
```

```
code ends
end start
```