

目 录

1	在代码段中使用数据	2
2	在代码段中使用栈	2
3	将数据、代码、栈放入不同的段	3
3.1	使用多个段的经验总结	4

1 在代码段中使用数据

计算 8 个数据: 0123H,0456H,0789H,0abcH,0defH,0fedH,0cbaH,0987H 它们的和, 结果存在 ax 寄存器, 实现代码如下:

```
1  assume cs:code
2  code segment
3      dw 0123H,0456H,0789H,0abcH,0defH,0fedH,0cbaH,0987H
4  start:
5      mov bx, 0
6      mov ax, 0
7      mov cs, 8
8
9  s:
10     add ax, cs:[bx]
11     add bx, 2
12     loop s
13
14     mov ax, 4c00h
15     int 21h
16 code ends
17 end start
```

如果要在代码段中使用数据, 就用如下框架:

```
1  assume cs:code
2  code segment
3      ; ...
4      ; 数据
5  start:
6      ; ...
7      ; 代码
8  code ends
9  end start
```

2 在代码段中使用栈

利用栈, 将 0123H,0456H,0789H,0abcH,0defH,0fedH,0cbaH,0987H 逆序存放, 实现代码如下:

```
1  assume cs:codesg
2  codesg segment
3      dw 0123H,0456H,0789H,0abcH,0defH,0fedH,0cbaH,0987H
4      dw 0, 0, 0, 0, 0, 0, 0, 0
5
6  start:
7      mov ax, cs
8      mov ss, ax
9      mov sp, 32
```

```
10      mov bx, 0
11      mov cx, 8
12  s:
13      push cs:[bx]
14      add bx, 2
15      loop s
16
17      mov bx, 0
18      mov cx, 8
19
20  s0:
21      pop cs:[bx]
22      add bx, 2
23      loop s0
24
25      mov ax, 4c00h
26      int 21h
27  codesg ends
28  end
```

3 将数据、代码、栈放入不同的段

首先看一个程序，实现了将数据、代码、栈放入不同的段，程序如下：

```
1      assume cs:code, ds:data, ss:stack
2
3      data segment
4          dw 0123H,0456H,0789H,0abcH,0defH,0fedH,0cbaH,0987H
5      data ends
6
7      stack segment
8          dw 0, 0, 0, 0, 0, 0, 0, 0
9      stack ends
10
11     code segment
12     start:
13         mov ax, stack
14         mov ss, ax
15         mov sp, 16
16         mov ax, data
17         mov ds, ax
18         mov bx, 0
19         mov cx, 8
20  s:
21         push [bx]
22         add bx, 2
23         loop s
24
25         mov bx, 0
26         mov cx, 8
27
28  s0:
```

```
29     pop [bx]
30     add bx, 2
31     loop s0
32
33     mov ax, 4c00h
34     int 21h
35 code ends
36
37 end start
```

3.1 使用多个段的经验总结

程序有多个段的时候，我们可以借助段名去访问段中的数据。一个段名将被编译器处理为一个段地址，然后将段地址送入相应的段寄存器，然后使用偏移地址去访问。需要注意的是，8086CPU 不允许将一个数值直接送入段寄存器中，需要使用如下方式：

```
1     mov ax, data
2     mov ds, ax
```

CPU 对我们定义的段中的内容，是当作指令执行、当作数据访问还是当做栈空间，完全由程序中具体的汇编指令决定。我们通过对 CS:IP、SS:SP 和 DS 这几个寄存器进行设置，从而决定 CPU 对代码的执行。