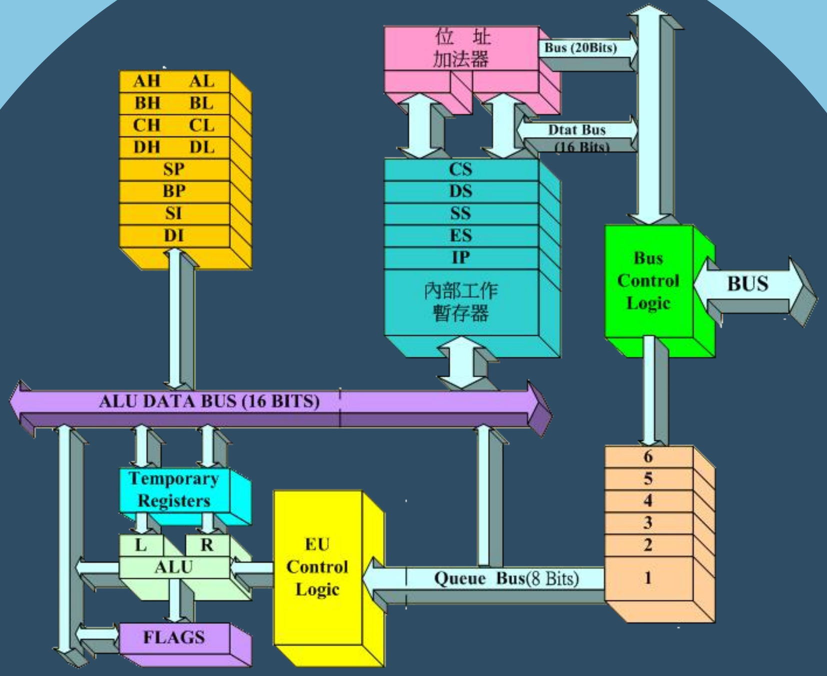
子程序的另一种写法

贺利坚　主讲



汇编语言程序设计
Assembly Language

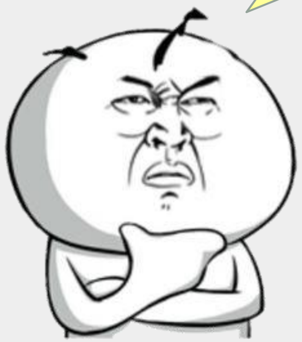
初始的程序

```
assume cs:code, ss:stack
stack segment
      db  16 dup (0)
stack ends
code segment
start: mov ax,stack
      mov ss,ax
      mov sp,16
      mov ax,1000
      call s       ;调用子程序
      mov ax,4c00h
      int 21h
   s: add ax,ax     ;子程序开始
      ret          ;子程序返回
code ends
end start
```

```
#include <stdio.h>
int max(int x,int y);
int main()
{
    int a, b, c;
    scanf(%d %d", &a, &b);
    c = max(a, b);
    printf("max=%d", c);
    return 0;
}
int max(int x, int y)
{
    return(x>y?x ： y);
}
```

```
名称 proc
    ……  ;实现逻辑功能的指令
    (ret)
名称 endp
```

能否有可读性更好的写法？



程序新结构

```
assume cs:code, ss:stack
stack segment
        db  16 dup (0)
stack ends
code segment
start: mov ax,stack
        mov ss,ax
        mov sp,16
        mov ax,1000
        call s        ;调用子程序
        mov ax,4c00h
        int 21h
    s: add ax,ax      ;子程序开始
        ret           ;子程序返回
code ends
end start
```

```
assume cs:code, ss:stack
stack segment
        db  16 dup (0)
stack ends
code segment
main proc
start: mov ax,stack
        mov ss,ax
        mov sp,16
        mov ax,1000
        call subp
        mov ax,4c00h
        int 21h
main endp
subp proc
    s: add ax,ax
        ret
subp endp
code ends
end start
```

```
assume cs:code, ss:stack
stack segment
        db  16 dup (0)
stack ends
code segment
main proc
start: mov ax,stack
        mov ss,ax
        mov sp,16
        mov ax,1000
        call far ptr subp
        mov ax,4c00h
        int 21h
main endp
subp proc
    s: add ax,ax
        retf
subp endp
code ends
end start
```