**实验四**

**1.编写程序f.c，不用主函数main**

编译时没有错误，连接出现了错误，undefined symbol '\_main' in module C0S

说明C0S在连接我们的程序，是连接main这个标号。我们程序里没有main 所以在C0S

中的main 就没有定义。

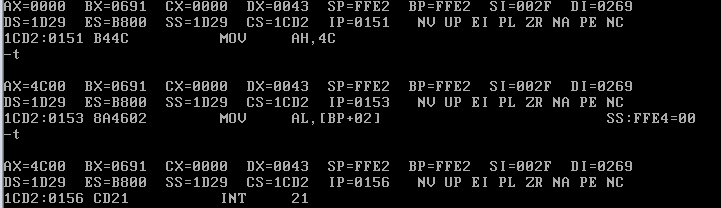
**2**.f.exe debug 调试得出共29个字节，f.exe不能正确返回，因为我们单单

把这个子程序编译，连接，没有返回指令所以不能正确返回，偏移地址 0000

**3.**m.exe debug 调试得出共29个字节，m.exe能正确返回，因为编译环境连接时将我们程序与c0s.obj进行连接，在c0s.obj中调用main函数，main函数返回后到c0s.obj中，c0s.obj中具有返回指令。f.exe与m.exe用户程序汇编代码是相同的。

**4.**main函数的调用指令和程序返回指令是从c0s.obj中来tc.exe 把c0s.obj和用户程序。obj文件一起连接生成.exe文件。

QQ截图20160519130638



**5.编写 c0s.asm**

assume cs:code

data segment

db 128 dup(0)

data ends

code segment

start:mov ax,data

mov ds,ax

mov ss,ax

mov sp,128

call s

mov ax,4c00h

int 21h

s:

code ends

end start

masm 生成c0s.obj 在TC环境中将c0s.obj与我们写的子函数.obj连接

就是把子程序加到s:后面。所以我们主函数的名字就变得很多了。经我调试，果然是这样。

**6.在新的c0s.obj基础上编写f.c**

#define buffer ((char \*)\*(int far \*)0x200) ；定义buffer为char型地址，地址为0000:0200地址处所放的地址

f()

{

buffer=0; //（0000:0200）=0000 偏移地址为0000

buffer[10]=0; //令ds:000ah处的单元赋值为0

while(buffer[10]!=8) //判断ds:000ah处的单元值是否为8 不是就继续执行，是就跳转

{

buffer[buffer[10]]='a'+buffer[10];

buffer[10]++;//经过8次得到 ds:0000h 开始单元一次存放‘abcdefgh’。

}

}