0924_综合研究2研究报告

研究问题

> 在main函数中添加代码，打印出下列函数的段地址和偏移地址

```
int a;
void f1(void) { a = 1; }
void f2(void) { a = 2; }
void f3(void) { a = 3; }
main() {
    .....
}
```

研究过程

1. 因为代码段地址是存储在`cs`中的所以我们可以通过`printf("\nCS: %x\n", _CS);`来打印出程序运行时的段地址
2. 函数的标号就是对应的偏移地址我们可以通过`printf("\nf1: %x\n", f1);`来打印出函数的偏移地址
3. 所以修改`a.c`添加相应打印输出语句

```
int a;
void f1(void) { a = 1; }
void f2(void) { a = 2; }
void f3(void) { a = 3; }
main() {
    char *string = "--------------------";
    printf("\nCS: %x\n", _CS);
    printf("%s", string);
    printf("\nf1: %x\n", f1);
    printf("\nf2: %x\n", f2);
    printf("\nf3: %x\n", f3);
}
```

4. 编译链接完后执行可执行文件可以分别看出3个函数的偏移地址和段地址

5. 然后在DEBUG中验证输出结果是否正确

```
C:\>DEBUG.EXE \SRC\A.EXE
-r
AX=FFFF  BX=0000  CX=180E  DX=0000  SP=0080  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=08F0  CS=076A  IP=0000   NV UP EI PL NZ NA PO NC
076A:0000 BAA508        MOV     DX,08A5
-g

CS: 76a  和上次运行结果不一样?
--------------------
f1: 1fa

f2: 201

f3: 208
```

```
-u 1fa
076A:01FA C7065E040100  MOV     WORD PTR [045E],0001   f1
076A:0200 C3            RET
076A:0201 C7065E040200  MOV     WORD PTR [045E],0002   f2
076A:0207 C3            RET
076A:0208 C7065E040300  MOV     WORD PTR [045E],0003    f3
076A:020E C3            RET
076A:020F 56            PUSH    SI
076A:0210 BE9401        MOV     SI,0194
076A:0213 8CC8          MOV     AX,CS
076A:0215 50            PUSH    AX
076A:0216 B8A901        MOV     AX,01A9
076A:0219 50            PUSH    AX
```

6. 发现 `DEBUG` 中执行 `a.exe` 打印出的 `CS` 的值和直接运行的结果不一致，剩下三个函数的偏移地址都一致但是 `DEBUG` 中打印的 `cs` 的值是正确的，所以推测直接运行 `a.exe` 的结果也是正确的。可能是应为 `DEBUG` 本身就是一个可执行程序所以在运行时系统就已经分配给 `DEBUG` 一块内存然后 `DEBUG` 再分配内存给 `a.exe` ,而直接运行 `a.exe` 的话就系统直接分配的内存所以两次的结果会不同。

研究结果

- 函数的名称就好像汇编程序里面的标号起到一个定位的作用，方便程序嵌套和跳转