# 0928\_综合研究1-3补充研究报告

## 综合研究2

通过函数名直接打印出段地址和偏移地址（不使用\_CS）

通过查阅相关资料可以尝试把函数名前强制转换成长整型

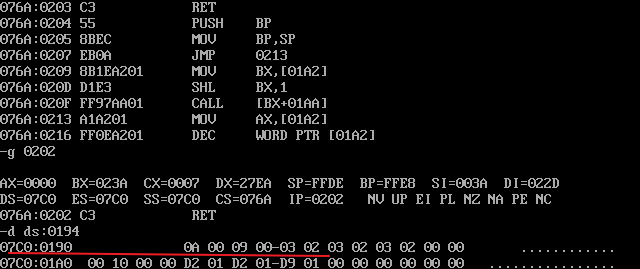
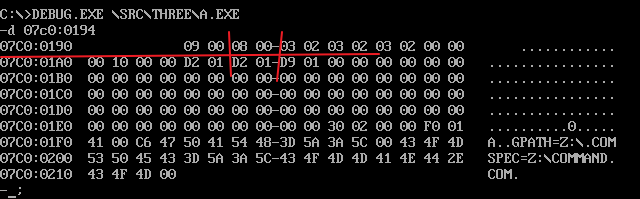
int a;  
void f1(void) { a = 1; }  
void f2(void) { a = 2; }  
void f3(void) { a = 3; }  
main() {  
 char \*string = "--------------------";  
 printf("\nCS: %x\n", \_CS);  
 printf("%s", string);  
 printf("\nf1: %lx\n", (long)f1);  
 printf("\nf2: %lx\n", (long)f2);  
 printf("\nf3: %lx\n", (long)f3);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



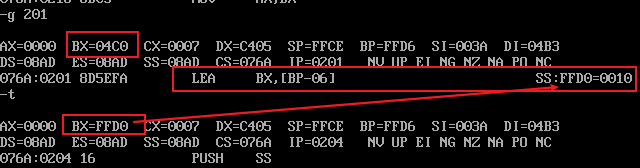
通过cs输出的数字可以检验出 强制转换可以正确输出函数对应的段地址和偏移地址。

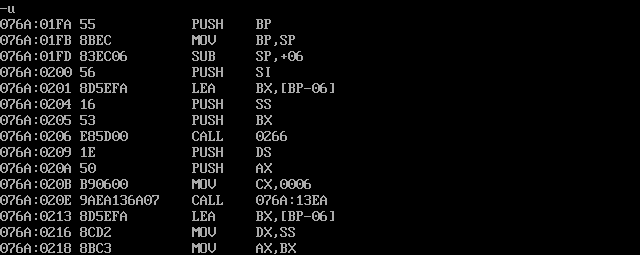
## 综合研究3

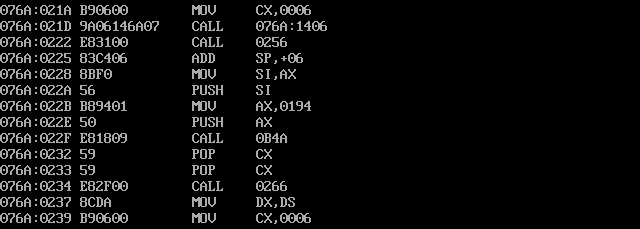
写程序验证全局变量存储空间是加载时分配，还是c0s分配。 @未解决

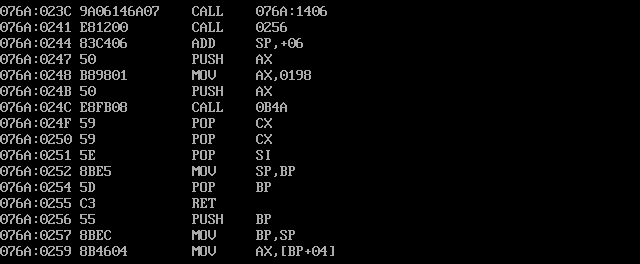
* 编写测试程序
* int b = 9;  
  main() {  
   static int a = 8;  
   a++;  
   b++;  
  }
* 编译链接生成可执行程序通过debug查看变量在数据段的偏移地址
* 
* 可以看见存储全局变量的段地址和偏移地址，然后在重新debug后直接跳转到相应的位置查看
* 
* 可以推断全局变量是在程序加载时给全局变量分配空间

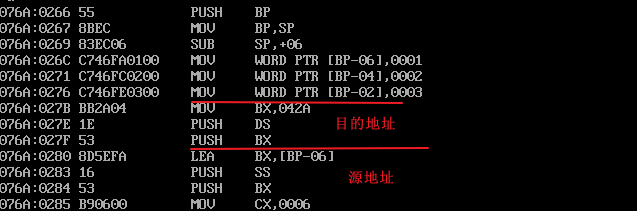
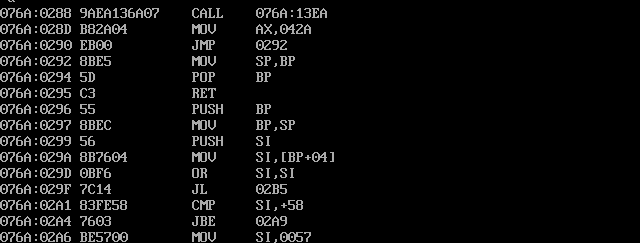
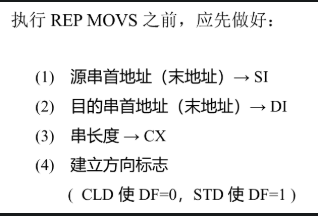
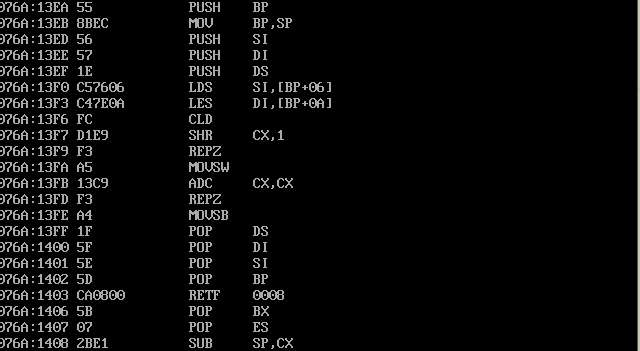
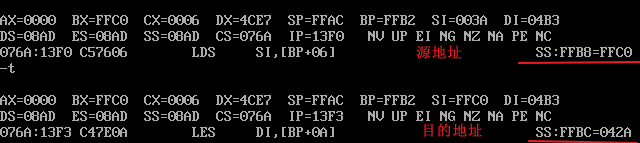
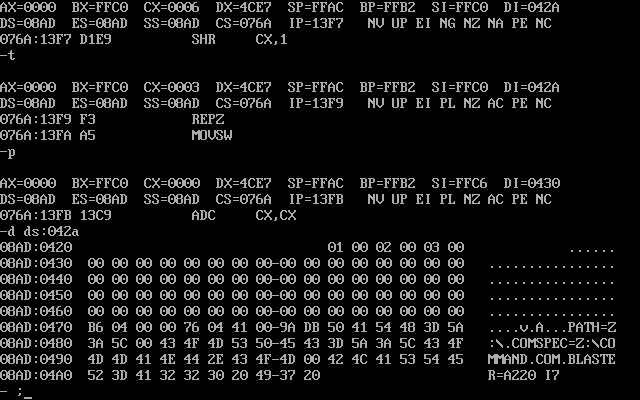
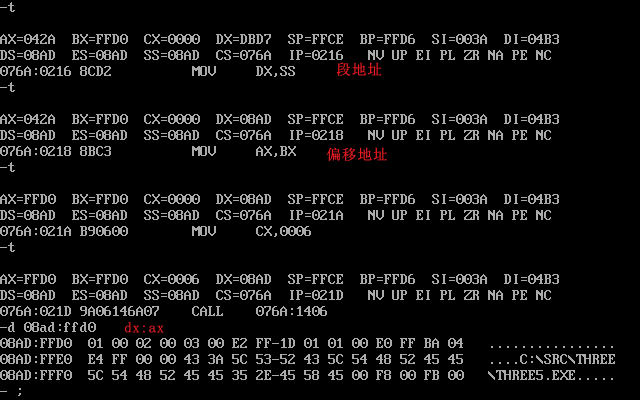
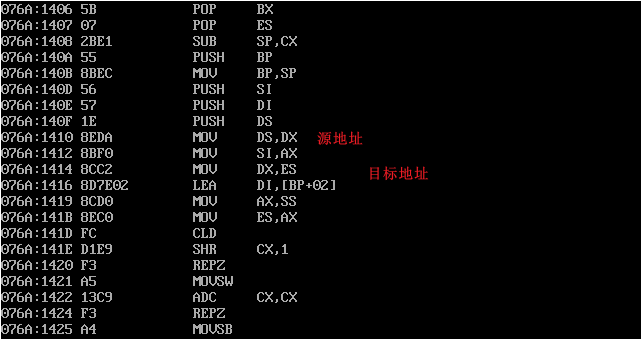
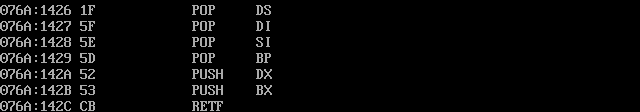
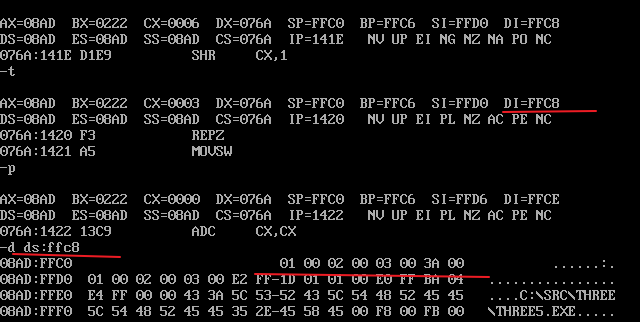
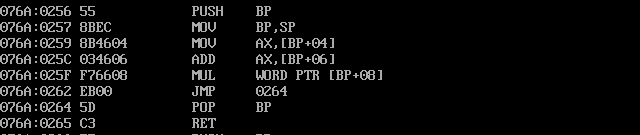
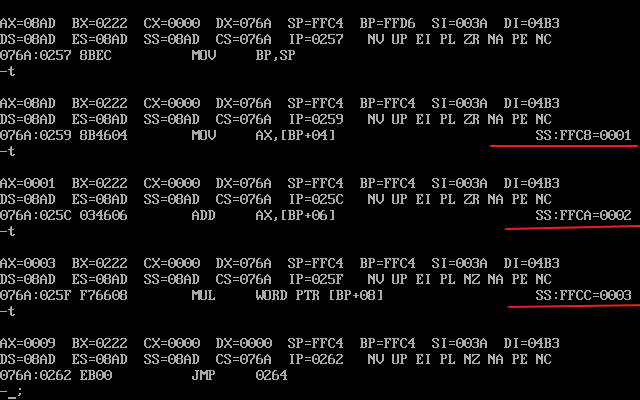
程序5首先要找到每条C语句对应的汇编代码，然后单步跟踪，跟踪时要将栈中的情况都画出来（这部分自己纸上画就可以，不必体现在研究报告中，但是必须通过这个过程去完全理解程序）。跟踪一遍后，回答教材中的两个问题。 @未解决

* LEA 取有效地址指令
* 
* 程序的汇编代码







* 执行 call 266可以看到把结构体的数据的首地址偏移地址和要返回结果的段地址的偏移地址存储到栈中然后调用 call 076a:13ea
* 
* 
  + call 076a:13ea
  + 
  + 
    - 执行到076a:13f0 c57606可以看到传入栈中的结构体数据的源地址和要返回结果的目的地址
    - 
    - 执行到adc cx,cx前后查看目的地址中的数据可以发现数据已经由源地址传送到目的地址
    - 
* 从函数返回的结构体数据类型存储在数据段中，在函数中先把返回结果的段地址和偏移地址压入栈中然后把结构体的段地址和偏移地址压入栈中，接着调用子程序（参数就是两个偏移地址）把栈中的数据移动到目标地址的内存中，子程序完成返回目标地址的偏移地址。
* 执行到call 076a:1406
  + 可以看到dx:ax为 076a:1406的参数
  + 
  + 子程序 076a:1460
  + 
  + 
    - 可以看到此程序的功能是把结构体数据复制到目的地址
  + 执行 call 0256
  + 
  + 可以看到子程序通过栈来使用传过来的结构体数据
  + 
* 向函数传递结构体数据
  + 通过调用子程序来把结构体数据复制到栈中
  + 然后函数在通过栈来使用结构体数据