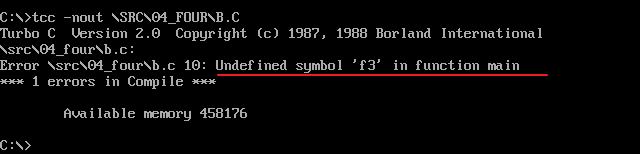
[综合研究4](#header-n3)  
[综合研究11](#header-n12)  
 [1](#header-n13)  
 [2](#header-n21)  
 [3](#header-n33)  
 [4](#header-n43)

# 综合研究4

能否在b.c程序中添加打印f3函数地址的代码来进行验证？

通过修改程序发现编译是就通不过

int func(int, int);  
  
int a, b;  
  
main() {  
 a = f1(1, 2);  
 b = f2(20, 10);  
 a = func(a, b);  
 printf("%d\n", a);  
+ printf("%lx", (long)f3);  
}  
  
int func(int a, int b) { return a \* b; }

* 编译报错
* 

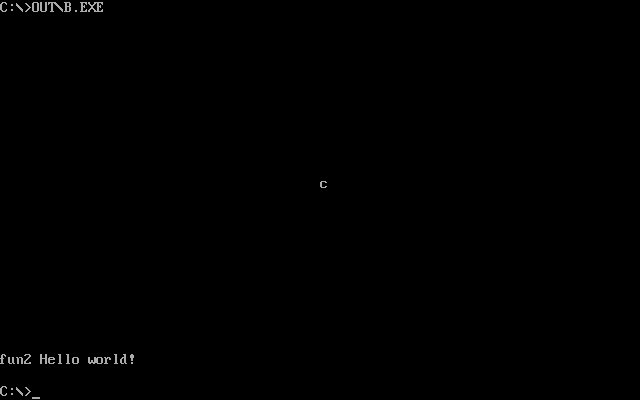
# 综合研究11

## 1

可以用printf函数实现程序11吗？

使用多个printf函数应该可以实现程序11,首先将fun.c生成目标文件,然后加入到cs.lib中，最后将b.c生成可执行文件

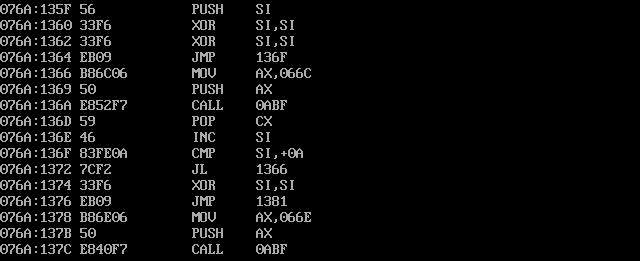
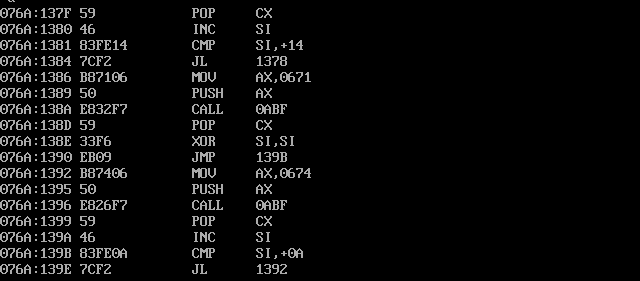
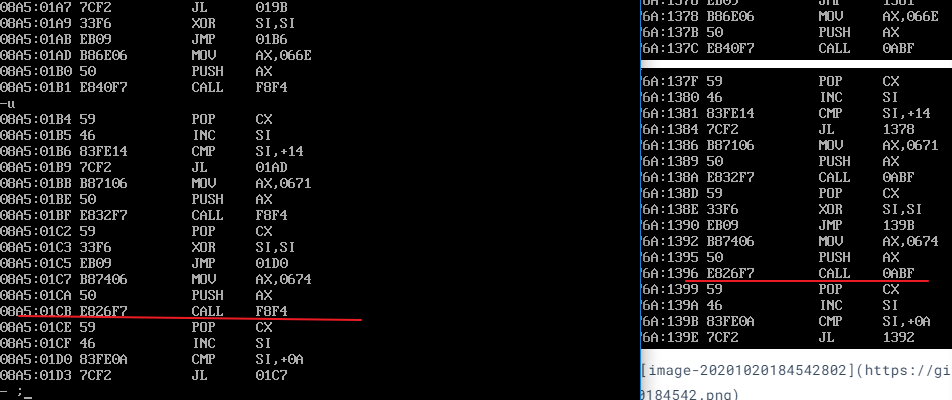
#fun.c  
void fun1() {  
 int n = 0;  
 for (n = 0; n < 10; n++) {  
 printf("\n");  
 }  
 for (n = 0; n < 20; n++) {  
 printf(" ");  
 }  
 printf("c\n");  
 for (n = 0; n < 10; n++) {  
 printf("\n");  
 }  
}  
void fun2() { printf("fun2 Hello world!\n"); }  
#b.c  
main() {   
 fun1();   
 fun2();  
}



然后把fun1对应的机器码写入数组中，接着看问题2

## 2

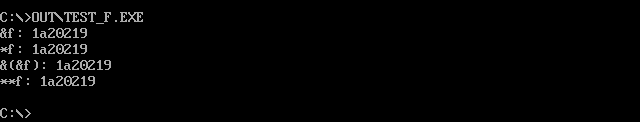
在数组a中添加printf机器码，编译器是否会将该函数连接到可执行文件中？猜测是：并不会将a数组中的数据当做代码去处理，将调用的函数连接到可执行文件中。 写程序对猜想进行验证。

* 程序
* char a[600] = {0x56, 0x33, 0xF6, 0x33, 0xF6, 0xEB, 0x09, 0xB8, 0x6C, 0x06,
    
   0x50, 0xE8, 0x57, 0xF7, 0x59, 0x46, 0x83, 0xFE, 0x0A, 0x7C,
    
   0xF2, 0x33, 0xF6, 0xEB, 0x09, 0xB8, 0x6E, 0x06, 0x50, 0xE8,
    
   0x40, 0xF7, 0x59, 0x46, 0x83, 0xFE, 0x14, 0x7C, 0xF2, 0xB8,
    
   0x71, 0x06, 0x50, 0xE8, 0x32, 0xF7, 0x59, 0x33, 0xF6, 0xEB,
    
   0x09, 0xB8, 0x74, 0x06, 0x50, 0xE8, 0x26, 0xF7, 0x59, 0x46,
    
   0x83, 0xFE, 0x0A, 0x7C, 0xF2, 0x5E, 0xCB};
    
     
  main() {
    
   ((void(far \*)())(long)a)();
    
   fun1();
    
  }
* 
* 
* 
* 
* 有个疑问是机器码一样对应的指令却不一样，再一个程序中一个通过a来调用fun1,一个直接函数名调用，按照猜测来说printf和fun1都应该加载上了，只要机器码一致应该执行效果来说是一致的，但因为机器码对应的机器指令不一样可以看到通过a来执行fun1 中没有调用printf
* 后来的想起call 后面是偏移地址，所以不一样但是调用的应该是一致的都是多少个字节后，而不是调用一个具体的函数,但是a因为段地址和不是代码段的所以调用不上。所以就不会把数组中的内容当成代码来处理。

## 3

编写函数f，打印f、&f、\*f、&（&f）、\*\*f ，对运行结果进行分析。

* code
* void f();  
  main() {  
   printf("&f: %lx\n\*f: %lx\n&(&f): %lx\n\*\*f: %lx\n", (long)&f, (long)\*f,  
   (long)&(&f), (long)\*\*f);  
  }  
  void f() { puts("hello world"); }



可以发现四者打印出来的数据完前一样

## 4

((**void**(far \*)())(**long**)a)()；从c语言角度理解该语句。

(void(far \*)())表示参数为空返回值为空的函数指针类型

(long)a将a强转成和函数指针大小匹配的类型

((void(far \*)())(long)a)();将a转换为函数指针