《漏洞利用及渗透测试基础》实验报告

姓名：王亦辉 学号：2310764 班级：计科1班

**实验名称：**

**格式化字符串漏洞**

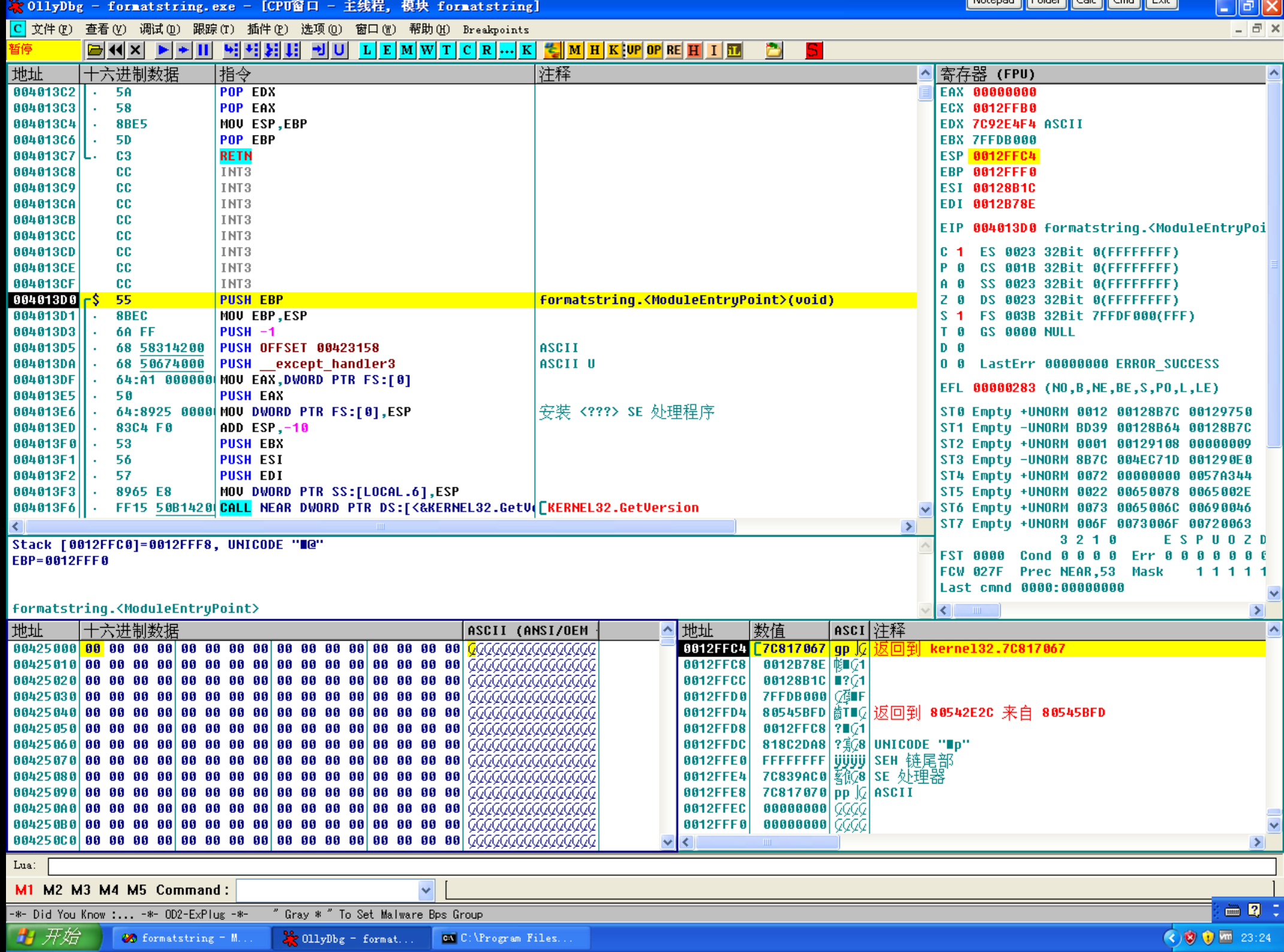
**实验要求：**

**以第四章示例4-7代码，完成任意地址的数据获取，观察Release模式和Debug模式的差异，并进行总结。**

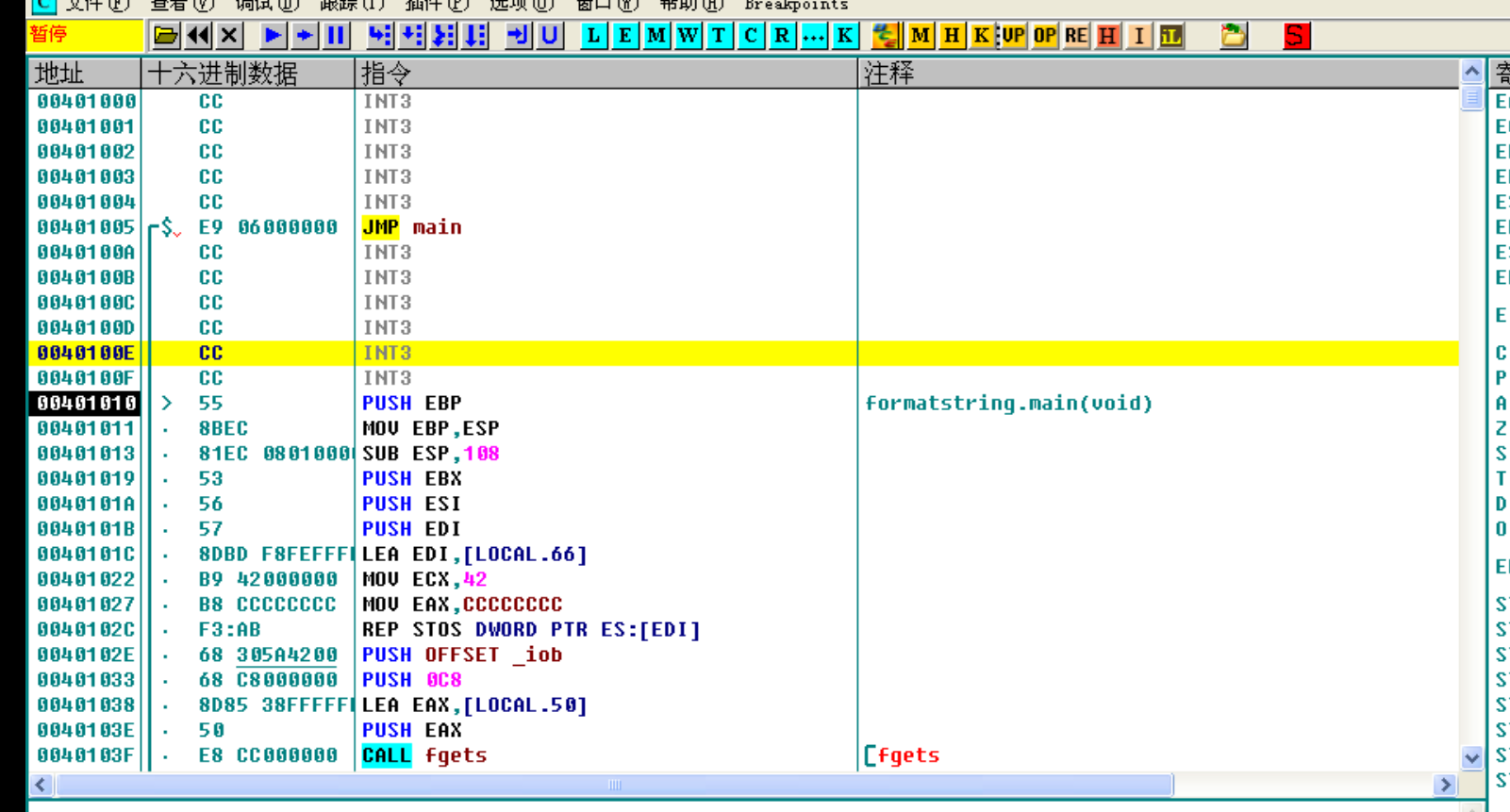
**实验过程：**

**一．Debug模式**

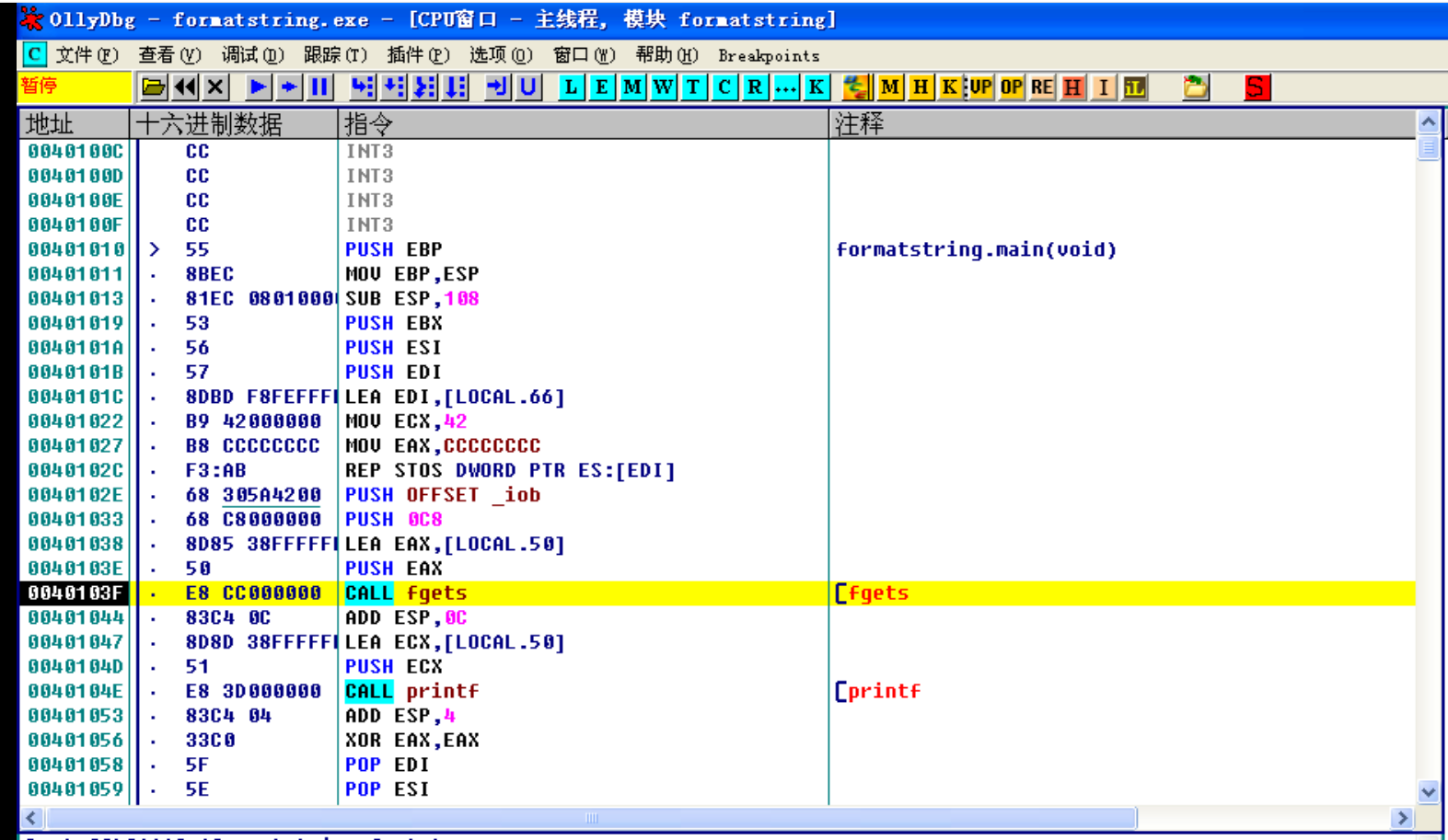
1. **将示例代码再Debug模式下编译，用Ollydbg打开编译产生的可执行文件。**



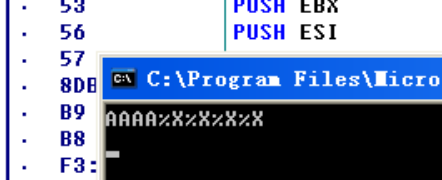
1. 找到main函数入口并步入



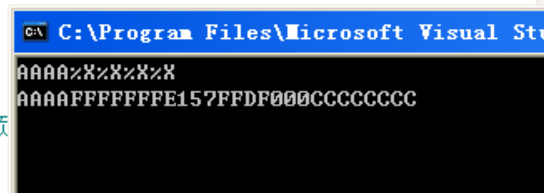
1. 进入函数后，进入进行栈帧调整，给函数的局部变量提供空间。



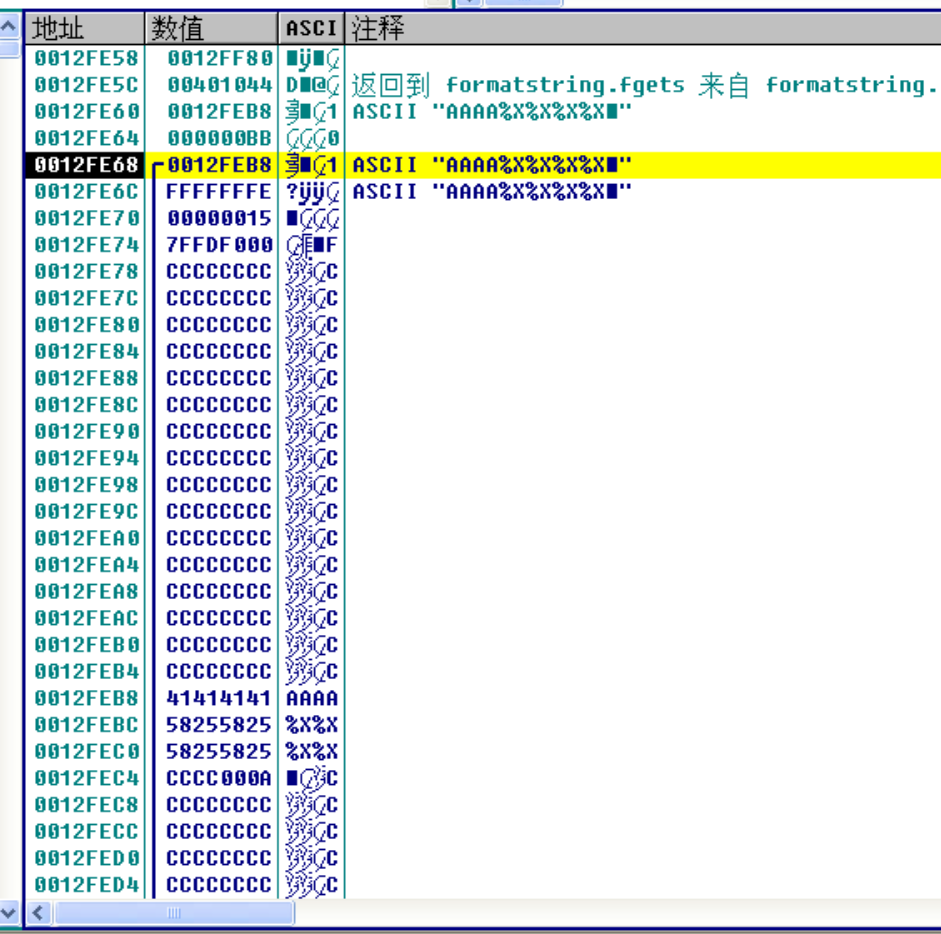
1. 运行fgets 输入AAAA%X%X%X%X



1. 运行到printf，可以看到打印出了额外的字符串。

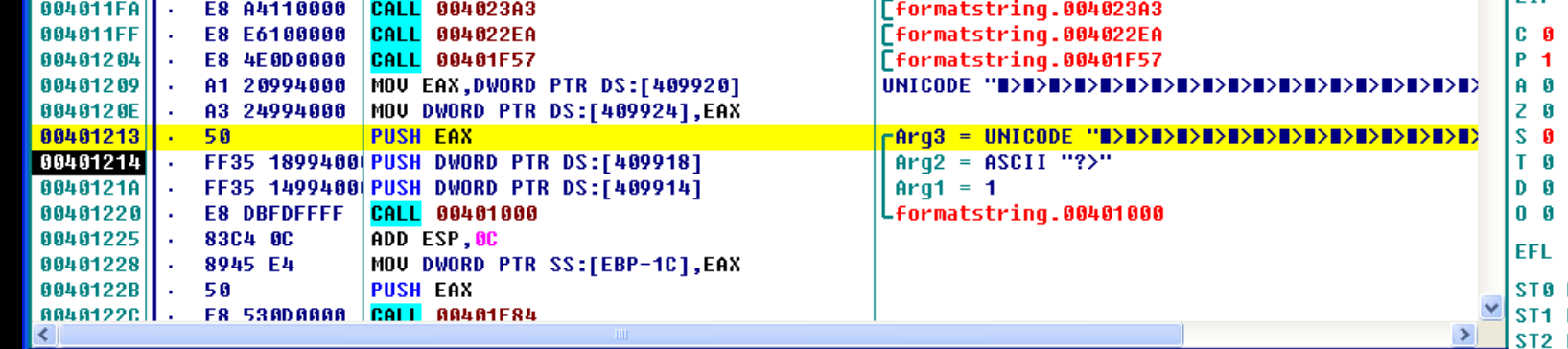


1. 检查栈上的内存，可以发现%X将后面四个地址的值打印出来了。此外在Debug模式下，函数被额外给了很多空间，从字符串地址到字符串之间有十几个地址用全C填充。可能是Debug模式下，为了便于调试出这样问题，或者为了让检测程序能检测到这些意外的打印或改变而添加的。

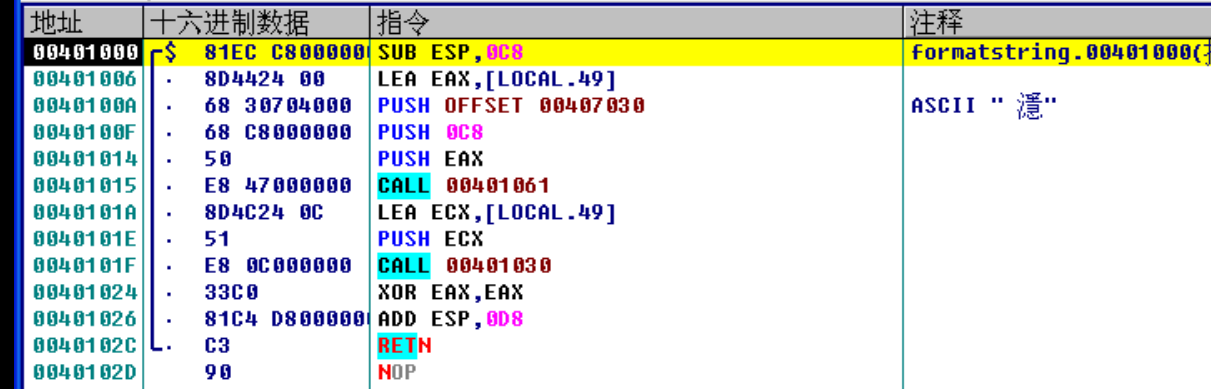


二．Release模式

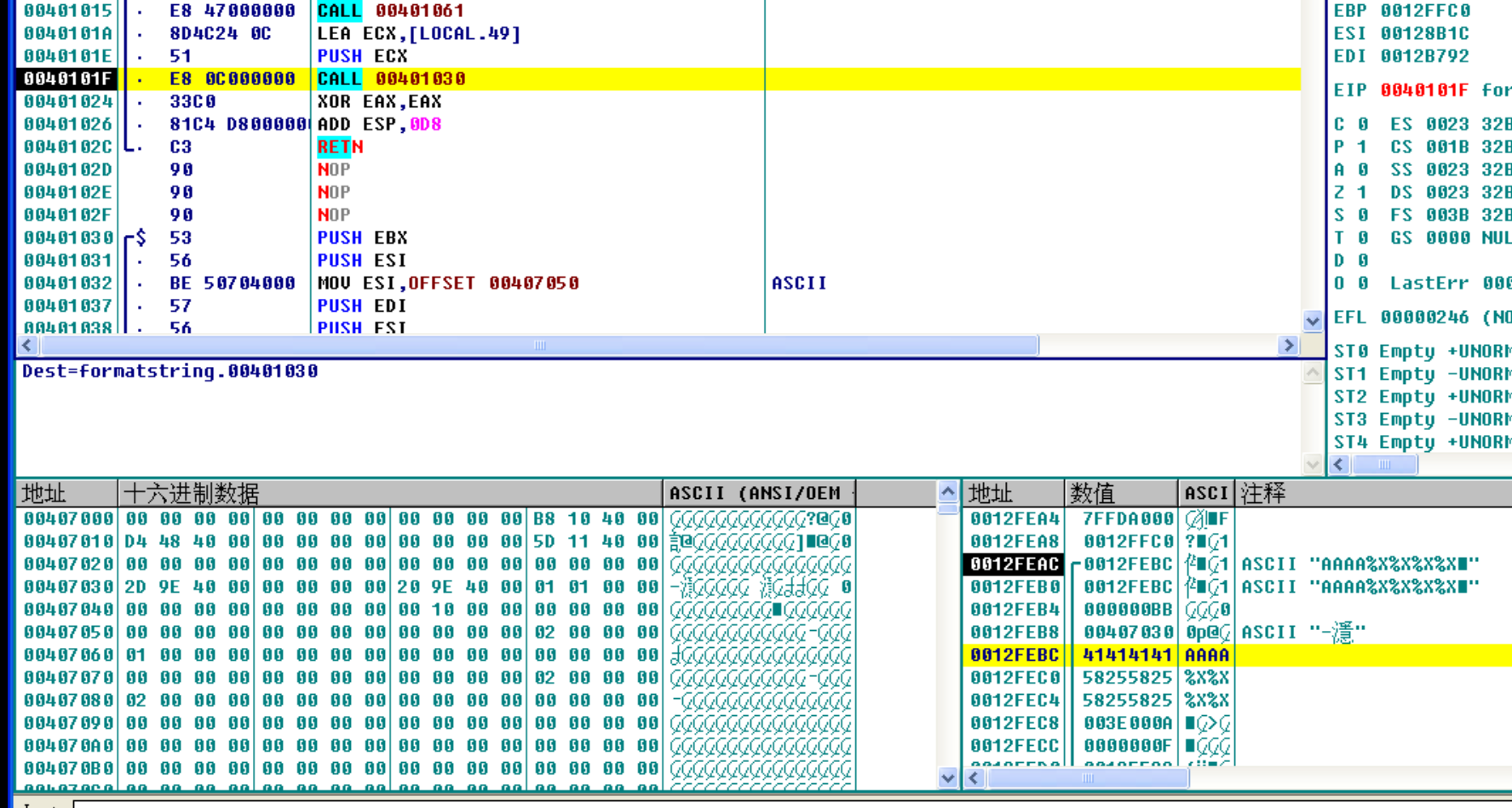
1. 以Release模式构建项目，然后用ollydbg打开，找到主函数入口点进入。



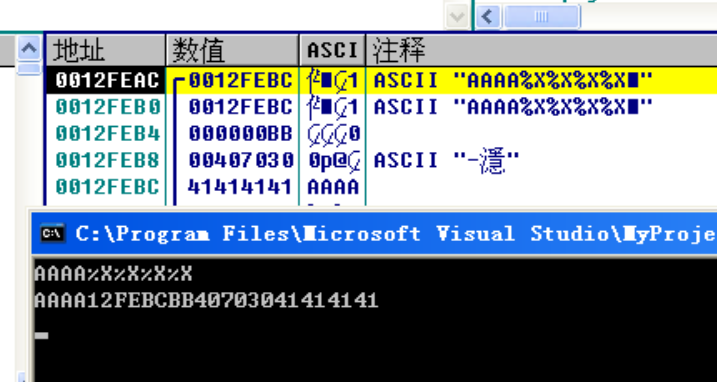
2. 可以看到Release模式下，栈只抬高了0xC8，不像Debug模式下，抬高了0x108，Release模式下并没有给出多余的空间。



3. 运行到fgets，同样输入AAAA%X%X%X%X，可以看到字符串与待传给参数的地址之间没有用CCCCCCCC填充的空隙。



4. 执行到printf时，从输出可以看出4个%X 把后面4个地址的值以十六进制打印出来了。



**心得体会：**

1. 对于C/C++编写的程序，需要非常小心地进行边界检查以及用户输入检查，因为这些事情都默认交给程序员做了。因此我们最好别使用C/C++写这些东西，靠人去保证完全不会有这样的漏洞是不可能的，应该有更好的自动的机制去保障。
2. 这种格式化字符串使用起来非常危险，如这里最后输入的是%s, 并且将AAAA换成自己想要读取的地址，就可以将这个任意地址的数据读出来，而这本应是用户做不到的，会造成隐私数据泄露等风险。