

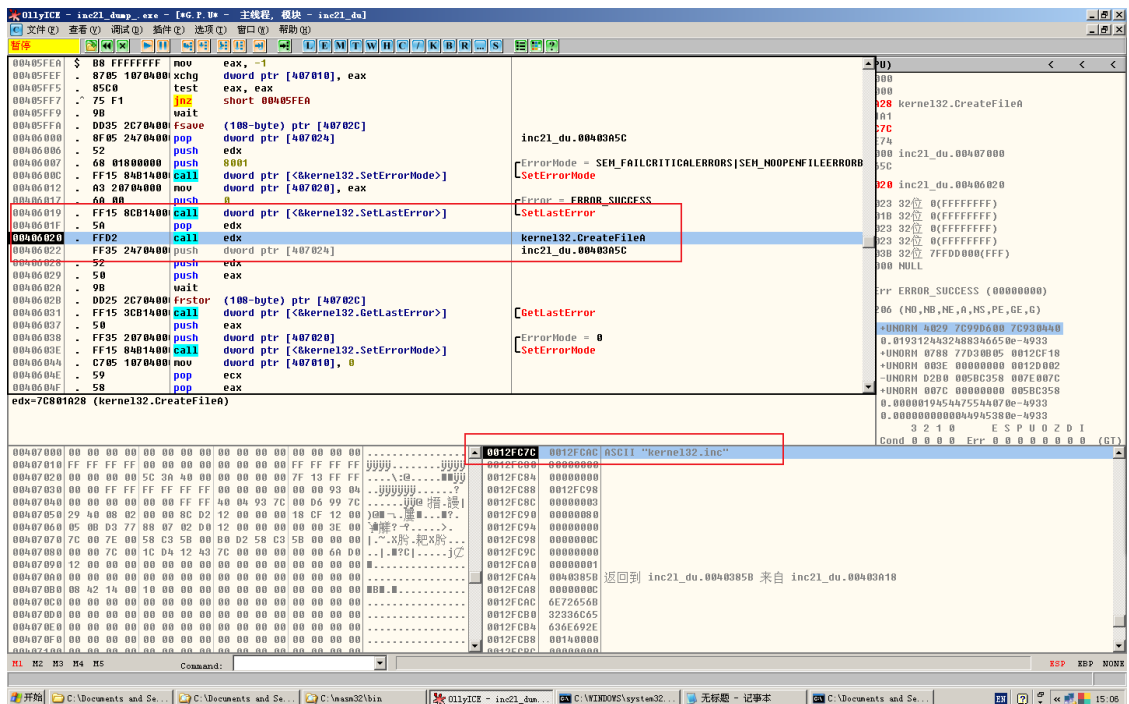
# inc2l程序分析

此程序将 inc 文件转换为 obj 文件，根据前提测试，会调用 m1 和 link 两个子程序来进行编译链接的工作。所以关键函数为 CreateProcess，同时依据 inc 文件进行生成，那么肯定依赖 CreateFile、ReadFile 和 WriteFile 等函数

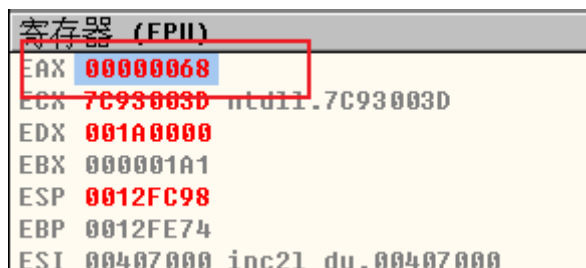
在这大前提下，对这两个函数下断点分析

## 分析

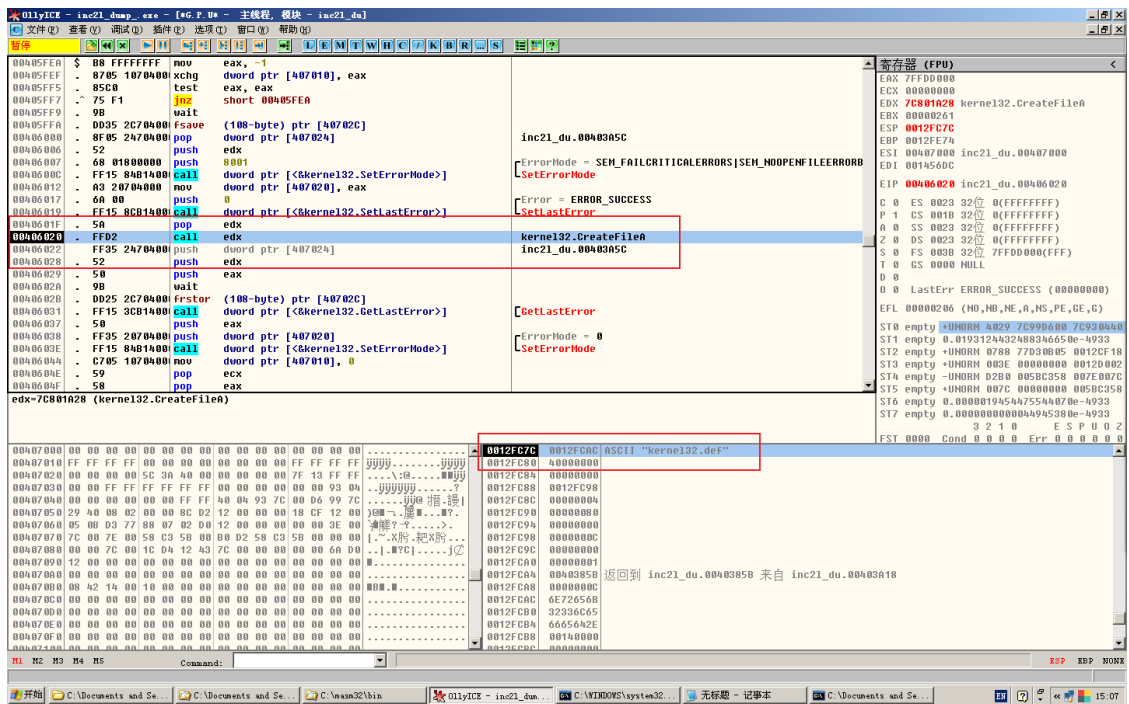
### 1. 打开 inc 文件



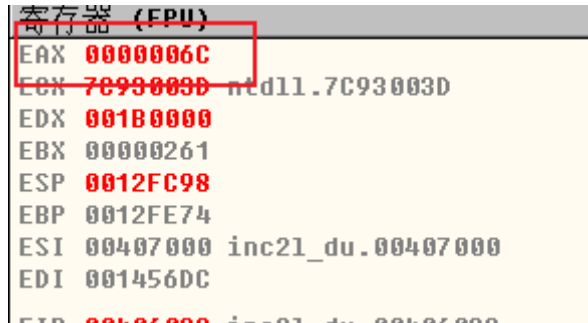
文件句柄 0x68



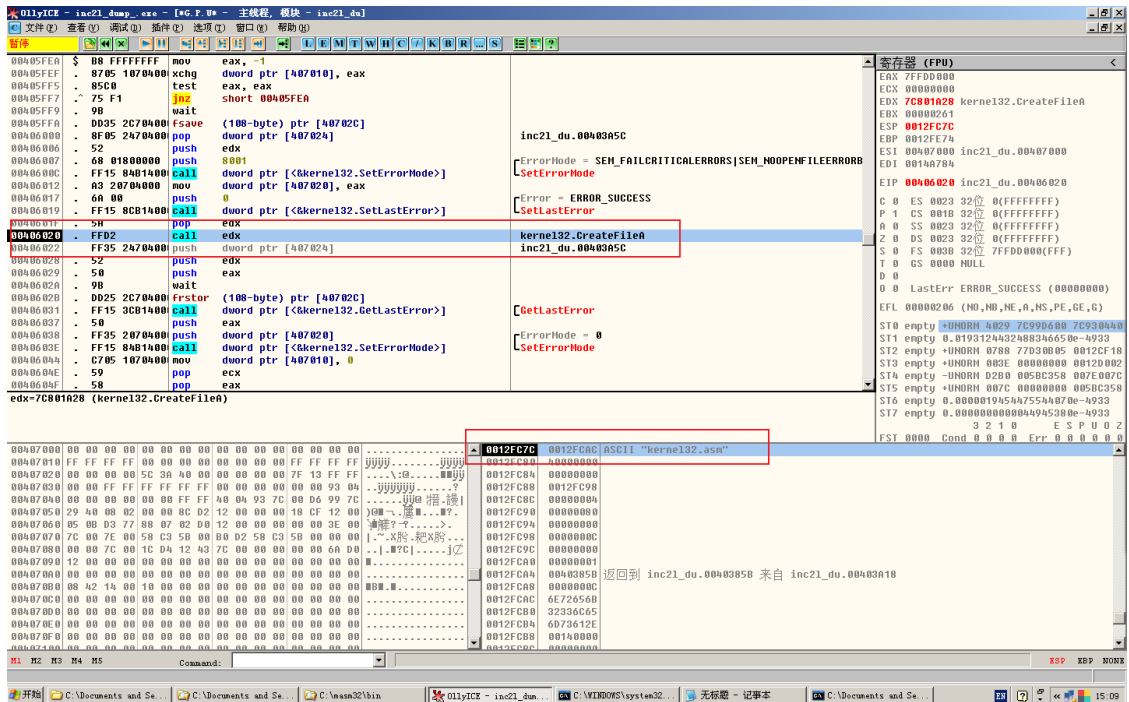
### 2. 创建 def 文件



文件句柄 0x6c



### 3. 创建 asm



文件句柄 0x70

寄存器 (FPU)	
EAX	00000070
ECX	7C93003D ntdll.7C93003D
EDX	001C0000
EBX	00000261
ESP	0012FC98
EBP	0012FE74
ESI	00407000 inc21_du.00407000
EDI	0014A784

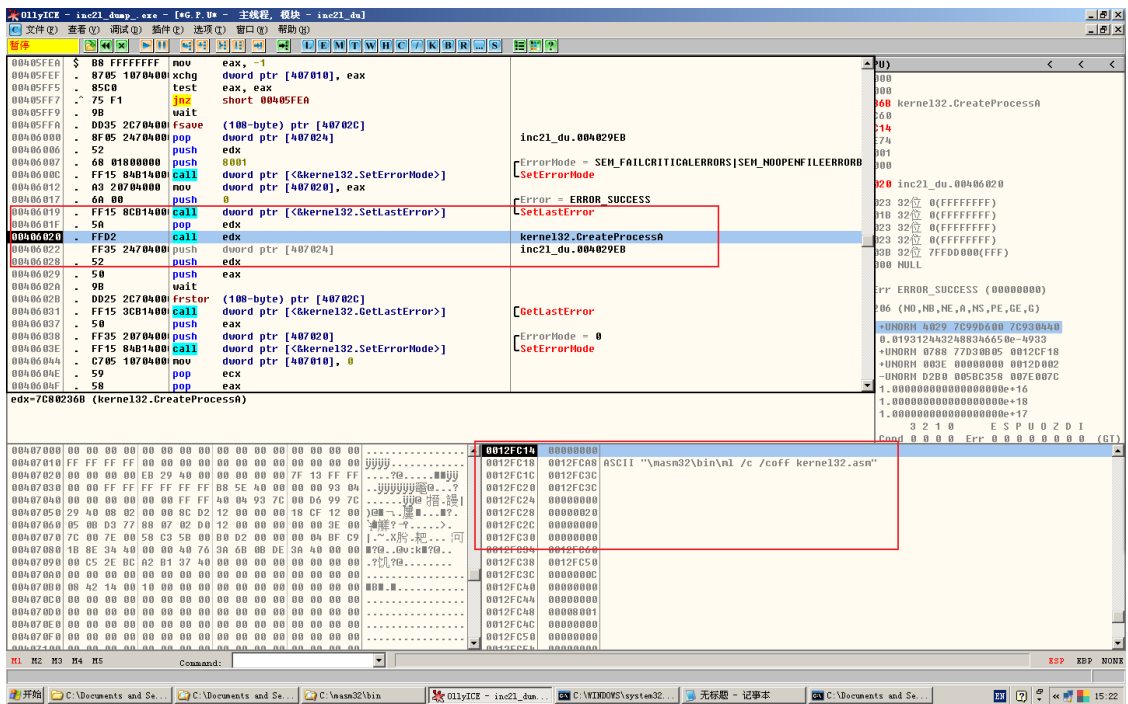
#### 4. 多次向 asm 文件中写数据

The screenshot shows the OllyICE debugger interface. The main window displays assembly code for the file 'inc21\_dmp.exe'. The code includes various instructions such as `mov eax, -1`, `xchg duword ptr [407010], eax`, `test eax, eax`, `jnz short 00405FEA`, `wait`, `fsave (108-byte) ptr [40702C]`, `pop duword ptr [407024]`, `push edx`, `call duword ptr [&kernel32.SetLastError]`, `mov duword ptr [407020], eax`, `call duword ptr [&kernel32.SetLastError]`, `pop edx`, `call duword ptr [407024]`, `push edx`, `push eax`, `wait`, `frstor (108-byte) ptr [40702C]`, `call duword ptr [&kernel32.SetLastError]`, `push eax`, `push duword ptr [407020]`, `call duword ptr [&kernel32.SetLastError]`, `mov duword ptr [407010], 0`, `pop ecx`, and `pop eax`. The CPU register window on the right shows the current state of the registers: EAX is 7FFD0000, ECX is 00000000, EDX is 7C92E4F4, and other registers are zero. The stack window at the bottom shows the current instruction being executed: `edx=7C92E4F4 (ntdll.KiFastSystemCallRet)`.

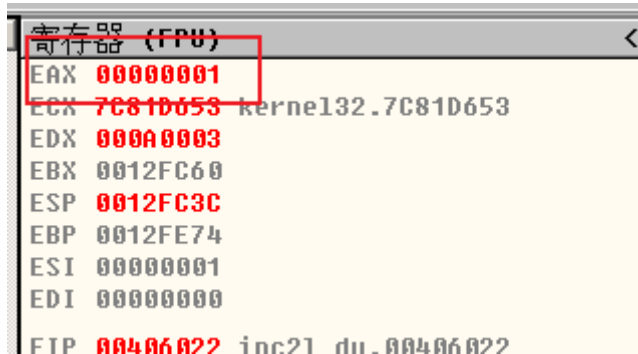
#### 5. 多次向 def 文件中写数据

The screenshot shows the OllyICE debugger interface. The main window displays assembly code for the file 'inc21\_dmp.exe'. The code includes various instructions such as `mov eax, -1`, `xchg duword ptr [407010], eax`, `test eax, eax`, `jnz short 00405FEA`, `wait`, `fsave (108-byte) ptr [40702C]`, `pop duword ptr [407024]`, `push edx`, `call duword ptr [&kernel32.SetLastError]`, `mov duword ptr [407020], eax`, `call duword ptr [&kernel32.SetLastError]`, `pop edx`, `call duword ptr [407024]`, `push edx`, `push eax`, `wait`, `frstor (108-byte) ptr [40702C]`, `call duword ptr [&kernel32.SetLastError]`, `push eax`, `push duword ptr [407020]`, `call duword ptr [&kernel32.SetLastError]`, `mov duword ptr [407010], 0`, `pop ecx`, and `pop eax`. The CPU register window on the right shows the current state of the registers: EAX is 7FFD0000, ECX is 00000000, EDX is 7C810E27, and other registers are zero. The stack window at the bottom shows the current instruction being executed: `edx=7C810E27 (kernel32.WriteFile)`.

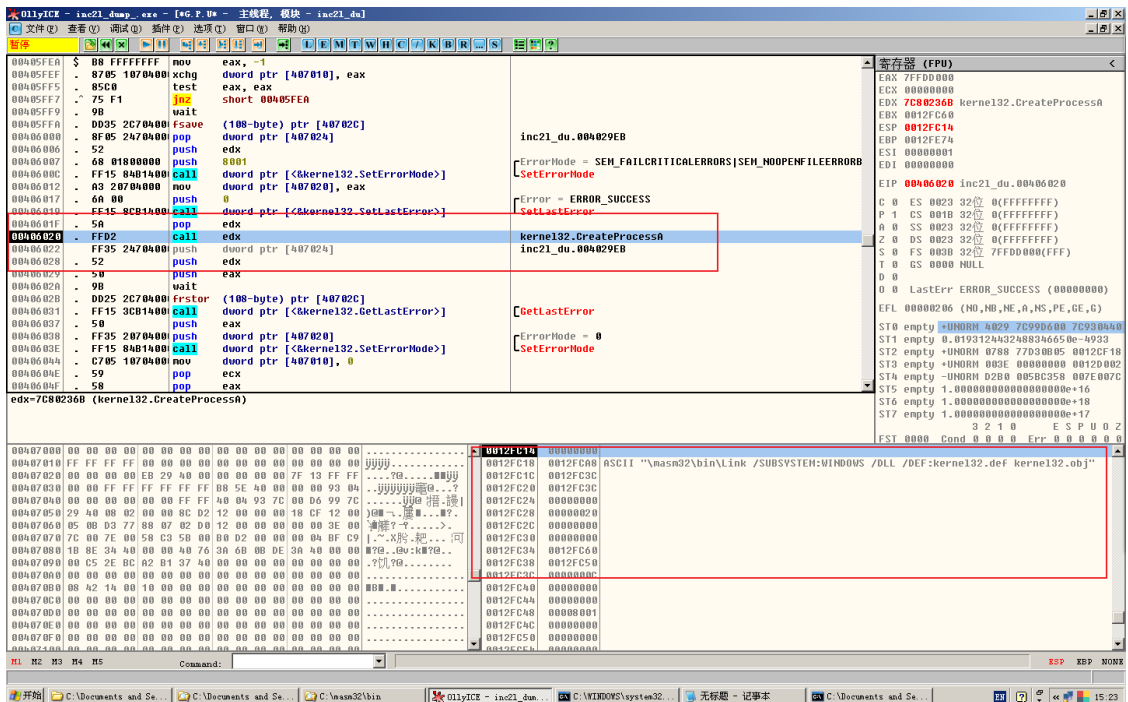
#### 6. 创建进程 \masm32\bin\ml /c /coff kernel32.asm



创建成功



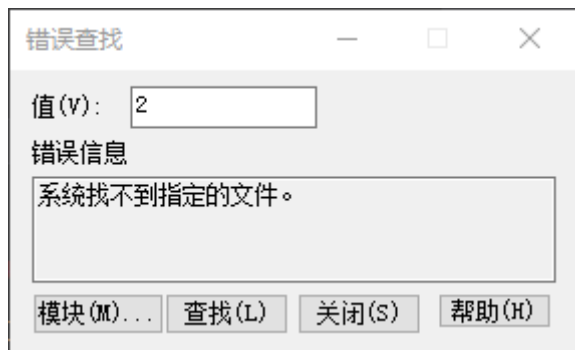
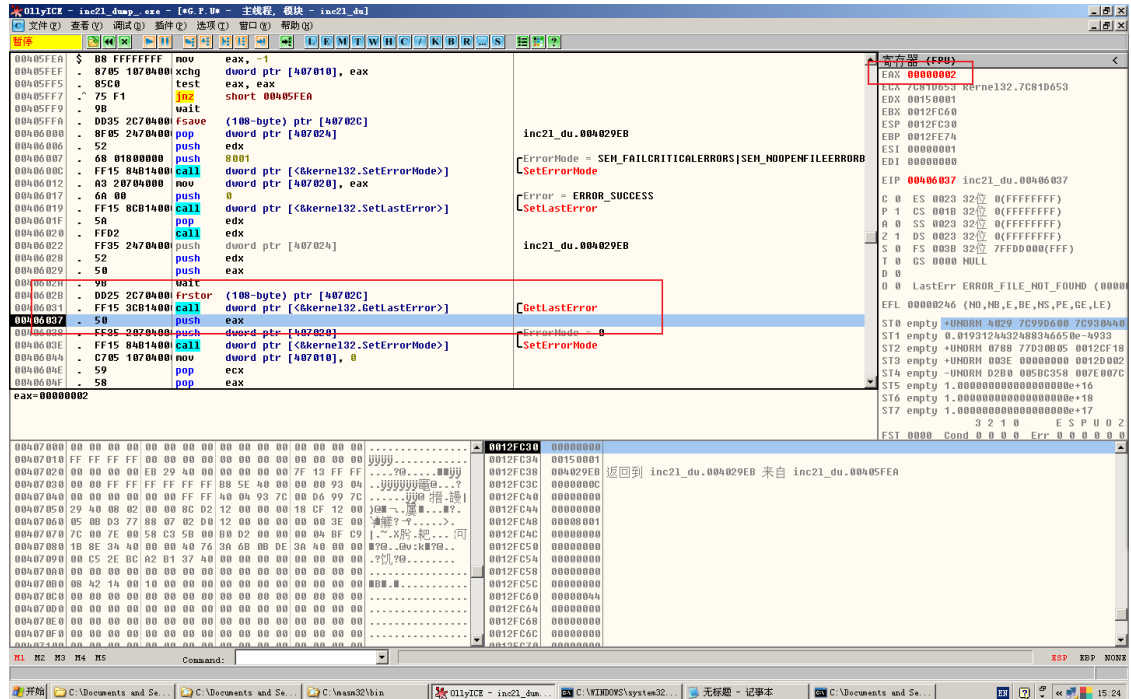
7. 创建进程 \masm32\bin\Link /SUBSYSTEM:WINDOWS /DLL /DEF:kernel32.def kernel32.obj



创建失败

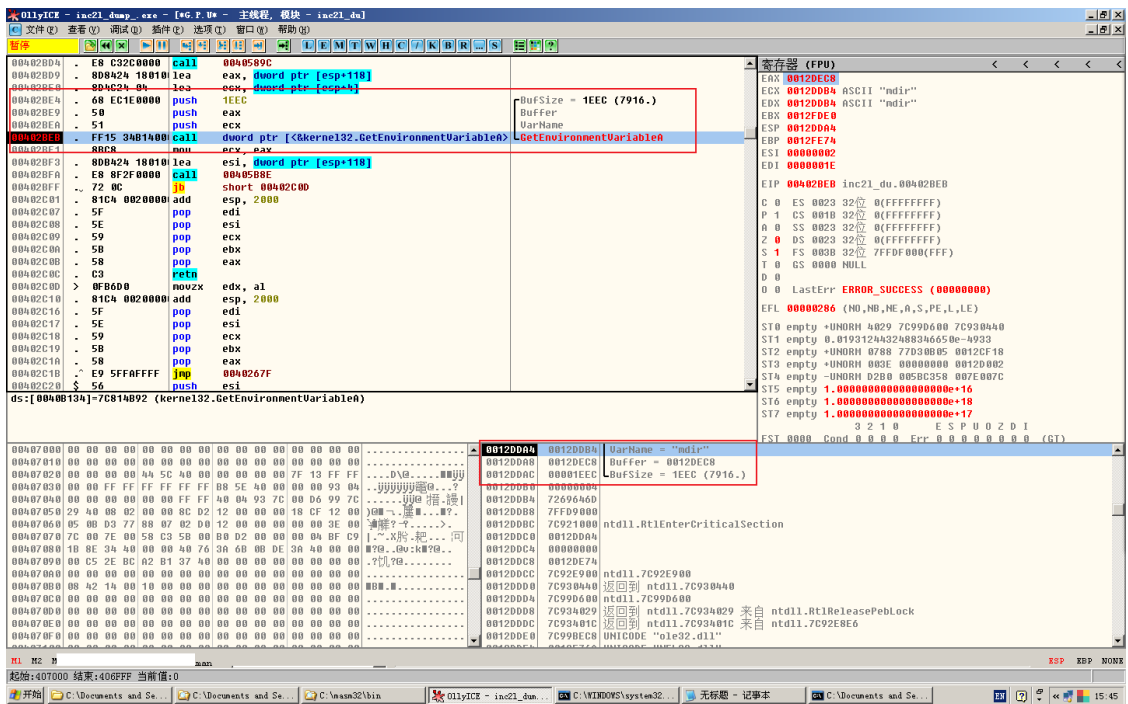
寄存器 (FPU)	
EAX	00000000
ECX	7C81D653 kernel132.7C81D653
EDX	00150001
EBX	0012FC60
ESP	0012FC3C
EBP	0012FE74
ESI	00000001

错误码为2



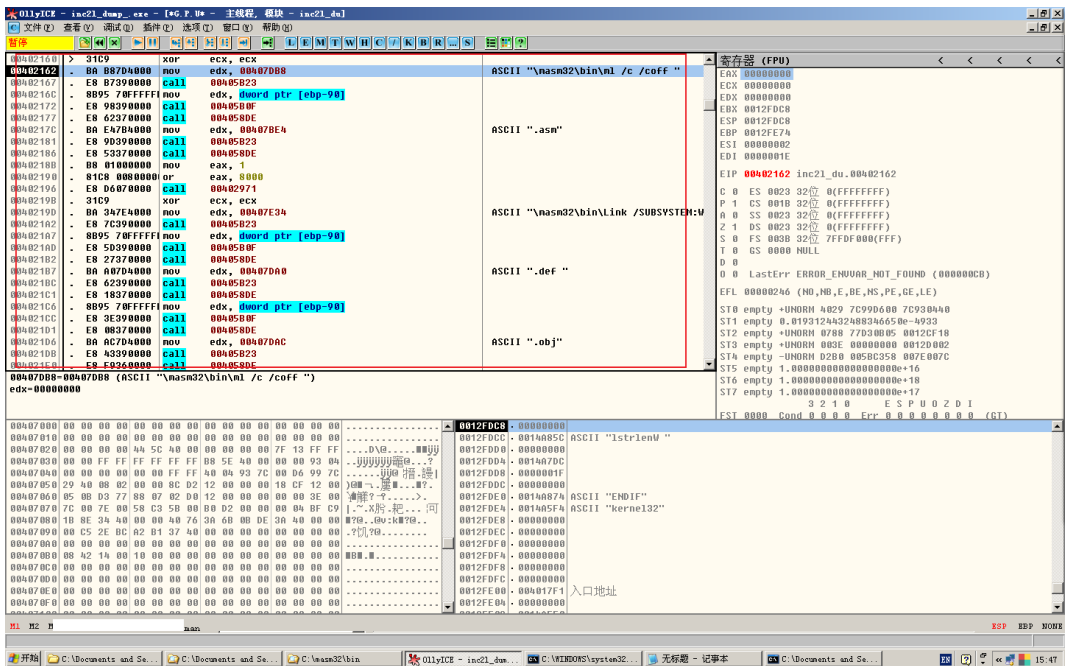
从这里就可以看出，创建进程的时候并没有指定盘符，猜测跟环境变量有关，于是断点于 `GetEnvrionmentStrings` 上再次执行

## 8. 获取环境变量 `mdir`

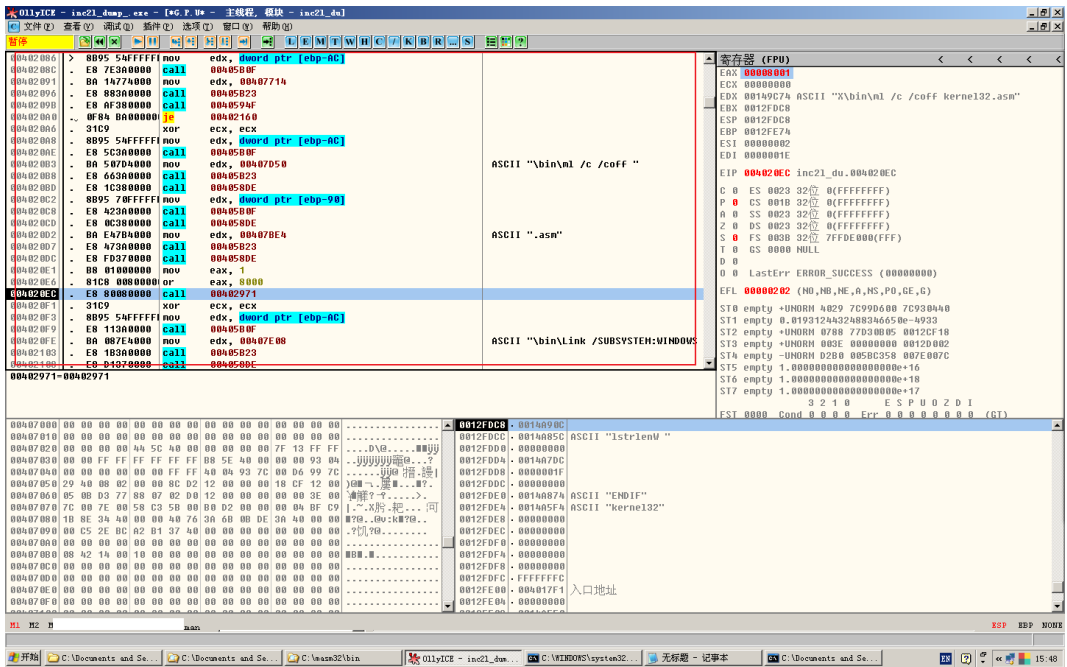


此时如果 mdir 存在，则以此拼接路径，不存在则使用默认路径即 \masm32\xxx

- o mdir 不存在的情况



- o 修改 GetEnvrionmentStrings 的返回值，因为 mdir 不存在，所以拼接之后只有后面的部分



## 修复方案

方案一：修改默认路径的字符串

方案二：添加环境变量 `mdir`