**2021/01/18_32位汇编_第6课_inc2l程序bug的调试、混合编程、联合编程的使用**

| | |
|---|---|
| **笔记本:** | 32位汇编 |
| **创建时间:** | 2021/1/18 星期一 10:43 |
| **作者:** | ileemi |


- 混合编程（dll）
- 汇编语言编写dll
- 联合编译 (link obj)


# 混合编程（dll）

> 高级语言和汇编语言混合编写（提高程序的执行效率）。

# 汇编语言编写dll

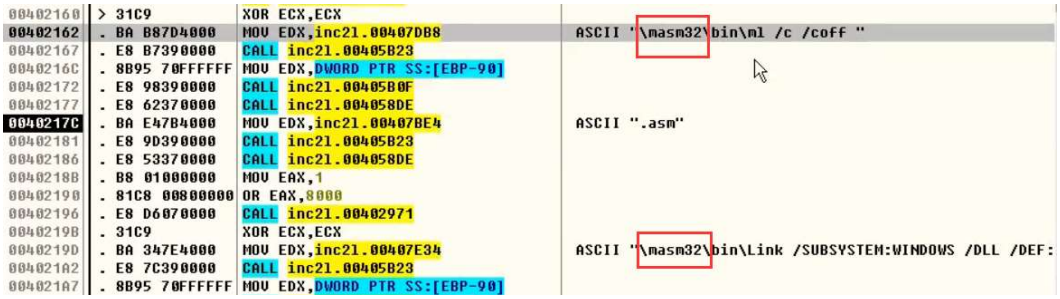
> 将功能程序进行封装（供高级语言使用）。

使用RadASM 编写动态链接库链接时需要添加 "/dll" 链接选型。同时需要添加入口函数（DllMainCRTStartup函数 在 "LIBC.lib" 文件中）。

Editplus搜索文件中的内从容："路径 content:搜索的内容"

DllMainCRTStartup函数（LIBC.lib） 需要调用 GetEnvironmentStrings函数（kernel32.lib），GetEnvironmentStrings 函数在RadASM中的 **kernel32.inc**中定义有问题，需要进行修改。GetEnvironmentStrings 函数在LIBC.lib库中为GetEnvironmentStringsA。

使用 "inc2l.exe" 可以将头文件（.inc）反生成对应的静态库（.lib），inc2l.exe 工具默认版本有一个Bug（链接路径问题），会导致生成 ".lib" 文件失败。




修复 inc2l.exe 存在的Bug：

通过程序的功能分析内部实现的大概过程（定位转换的函数位置），通过调试器进行调试分析（调试 inc2l.exe 程序时，需要添加一个命令行参数（要转换的 ".inc" 文件））：

- 程序转换时需要打开目标文件，就需要调用系统的API（CreateFile（分A和W版本） 属于 kernrl32.dll，OpenFile 只支持16位操作系统）
- 查看dll中是否存在API（Executable modules --> 目标dll -->View names）
- 通过 View names 找到 CreateFile（分A和W版本）后，将两个版本都下断点，程序运行，会自动停到断点处
- 调用link程序需要创建进程（CreateProcessA 函数下断点运行程序进行分析）
- x64dbg："符号" 在目标dll中搜索对应的API下断点，添加命令行参数，F9运行程序进行调试。"Alt + F9" 返回到用户代码

> 通过调试程序可以观察到 "inc2l.exe" 程序通过扫描 ".inc" 文件后，生成一个对应的 ".def" 文件，之后又创建一个对应的 ".asm" 文件，接着将 ".asm" 文件通过编译链接生成对应的动态链接库（同时也会生成一个".lib"文件），最后将生成的所有文件全部删除（".lib"文件保留）。

修改汇编代码：同时将修改后的数据保存到可执行程序中。
修改内存数据：

通过调试器修改可执行程序（修改汇编代码），调试起支持将可执行文件中修改后的代码保存到可执行文件中，不需要考虑文件格式的问题，操作实例：

- ollydbg：右键（Copy to Executable）--> All modifications --> Copy All --> 右键（Save File）--> 从新保存一份（patch）
- x64dbg：调试程序，修改汇编代码，右键 "不补丁" --> 修补文件 --> 选择要修改的文件

当程序又壳的时候直接通过调试起修改可执行程序中的汇编代码在保存是不可行的（会被覆盖）。想要直接修改可执行文件的汇编代码需要将可执行文件中的壳脱掉。
脱壳：从文件中重建内存格式，方便更好的打补丁。
补丁：从内存中将代码同步到文件中。

加壳程序会自动抬栈，程序运行起来的时候，解压缩代码之后，将代码写到指定的内存位置中，脱壳：将解密后程序的代码重新建立一个文件并写入，新建可执行文件的代码区数据区按照解密后程序的格式来写，程序的入口代码要和解密后程序的入口位置一致，脱壳后的可执行文件，就可以修改其内部的汇编代码并进行保存（打补丁）。

使用没有bug的 "inc2l.exe" 将修改后的 "kernel32.inc" 生成对应的 ".lib" 文件，并替换RadASM\masm32中对应的 "kernel32.lib" 文件即可。

# 联合编译 (link obj)

> 将不同语言的 ".obj" 文件合在一起进行编译。

- C程序中使用汇编代码（链接汇编代码的obj）：在链接C程序时和汇编代码生成的 ".obj" 文件一同链接（静态链接，不使用dll以及lib）。使用IDE编译，只需要将汇编生成的 ".obj" 文件添加到VS工程中，之后编译运行程序即可。
- 汇编程序中使用C代码（链接C代码的obj）：在C代码中编写功能函数（函数名前添加 "extern "C" int __stdcall" 不进行名称粉碎）编译即可。汇编代码中使用：添加函数声明，使用 "invoke" 调用C编写的函数，还需要将C对应的 ".obj"

文件添加到 汇编程序中（可在工程选项中的链接中添加，也可以将对应的 ".obj" 文件直接拖到工程中，之后编译链接汇编程序即可）。

> 主程序使用高级语言编写（界面），核心程序使用汇编代码进行编写（性能要求高）。