# 实验二 搜索API函数地址

## 一、实验目的

1）理解API函数搜索原理；

2）掌握搜索kernel32.dll地址的方法；

3）掌握搜索LoadLibrary（）及GetProcAddress()地址的方法；

## 二、实验内容和要求

1) 编程实现求动态链接库kernel32.dll的虚拟地址。

2）编程实现求LoadLibrary（）及GetProcAddress()虚拟地址。

## 三、实验思考

1）为什么PEB，TEB中能找到kernel32.dll的地址；

2）dll文件的导出表的结构是怎样的；

3）搜索得到LoadLibrary（）及GetProcAddress()虚拟地址后，怎样得到例如printf()等函数的地址？

## 四、实验开设方式

本实验开设方式为个人实验；实验授课时间1学时，上机3学时，学生分为一人一组，每人一台PC。

## 五、实验流程

按照实验内容逐步完成。

## 关键步骤

1. 定位kernel32.dll地址

(1) 通过段选择字FS在内存中找到当前的线程控制模块TEB

线程控制块中偏移位置为0x30的地方存放着指向进程控制块PEB的指针

(2) 进程控制块中偏移地址0x0c的地址存放着指向PEB\_LDA\_DATA结构体的指针，其中存放着已经被装载的动态链接库信息。

(3) PEB\_LDA\_DATA结构体偏移地址为0x1c的地方存放着指向模块初始化链表的头指针InInitializationOrderModulelist。

(4) 模块初始化链表InInitializationOrderModulelist中按顺序存放着PE装入运行时初始化模块信息，第一个链表节点是ntdll.dll，第二个链表节点就是kernel32.dll。

(5) 属于kernel32.dll的节点后，在此基础上再便宜0x08就是kernel32.dll在内存中的价值基地址。

1. 定位LoadLibrary（）及GetProcAddress()地址
   1. 从kernel32.dll的加载基地址开始偏移0x3c的地方就是其PE头
   2. PE头偏移0x78的地方存放着指向函数导出表的指针。
   3. 按以下方式在导出表中算出所需函数的入口地址

I 导出表偏移0x1c处的指针指向存储导出函数偏移地址（RVA）的列表

Ⅱ 导出表偏移0x20处的指针指向存储导出函数函数名的列表

Ⅲ 函数的RVA地址和名称按顺序放在RVA列表及函数名列表中，根据函数名在函数名称列表中搜索函数序号，再根据函数序号在RVA列表中搜索函数对应的RVA

Ⅳ 函数对应的RVA加上动态链接库的加载地址得到该函数的虚拟地址。