2021/02/25_PE_第8课_导出表

笔记本:	PE
创建时间:	2021/2/25 星期四 10:16
作者:	ileemi

预留空间

- 使用内敛汇编：__asm{xxxx}
- 自定义dll进行预留空间（dll的空间大小和被加载程序的字节大小一致）
- 移动主模块ImageBase预留空间，动态申请
- 定义节，合并节（防止节之间不连续）

手写一个最小字节的可执行的PE文件时导入表可以取消。

数据目录项保守可以留4项。

文件对齐值小于200，内存对齐值也必须小于200，需要一致，不然操作系统不承认其是一个有效的PE文件。

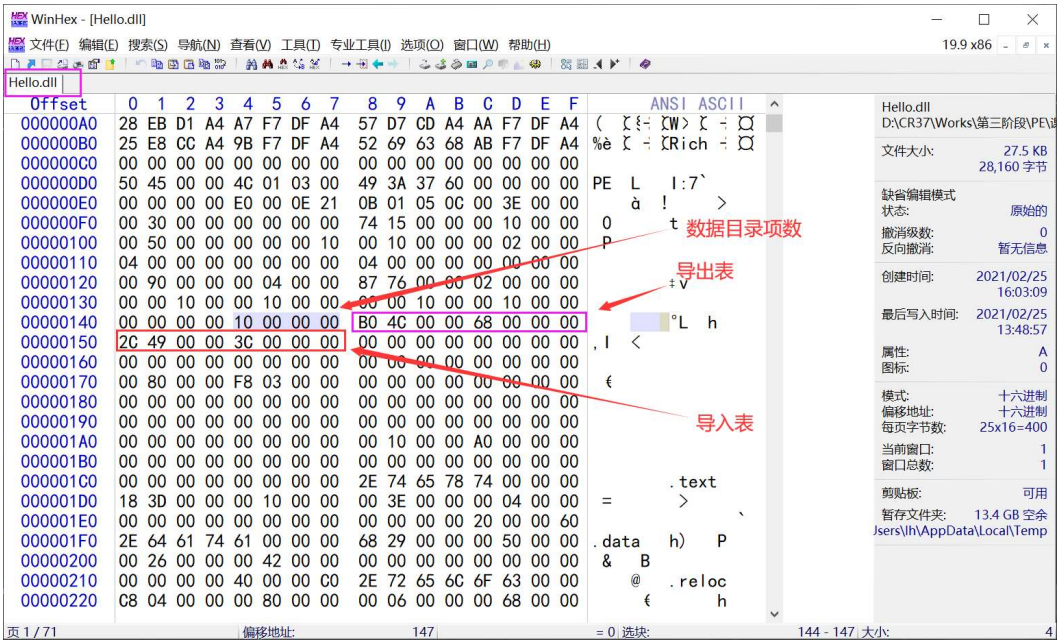
导出表

> IMAGE_DIRECTORY_ENTRY_EXPORT：记录导出函数的RVA的表，将导出表
> 一般讲"dll"，可执行程序很少有导出表（一般不这样做）。

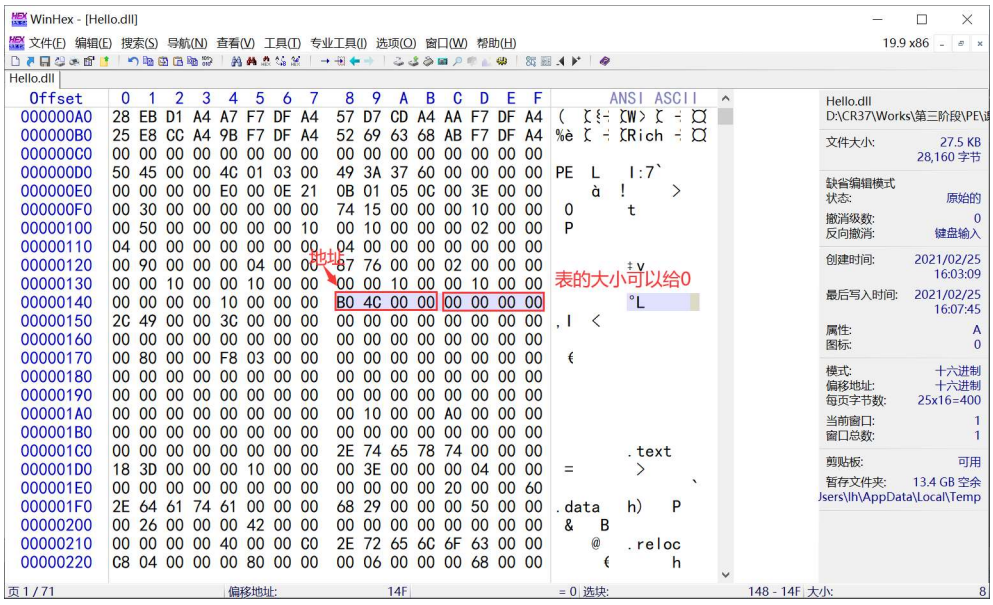
通过导入表可以知道API的地址，通过 LoadLibrary 得到 ImageBase，dll中会记录每个导出函数的 RVA，之后通过GetProcAddress就可以获取导出函数在内存中的地址。

自己编写的 "dll" 中有导出函数，操作系统是怎样去识别该 "dll" 中导出函数的地址的，就需要建立一个导出表，用来专门记录导出函数RVA。一般记录在dll中，可执行文件中也可以记录其内部的导出函数（可执行文件中没有导出函数也就没有导出表），使用时，就需要将可执行文件当作 "dll" 去使用，但是这样做维护性比较差。

通过汇编编写的 "dll" 需要写一个 "Dllmain" 函数，通过分析生成的 ".dll" 文件，文件中PE下数两行半为 "OEP"（可以抹掉，应为 "Dllmain" 不是必须写的）。对于一个 "dll" 程序来说，入口地址不是必须的。



导出表的大小可以给 "0"，地址不能给。

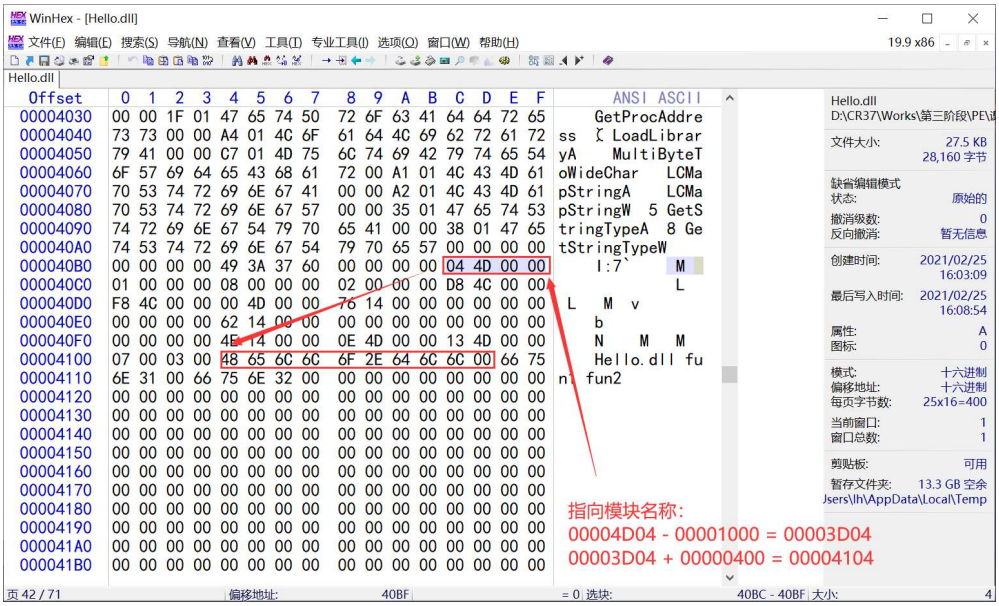


导出表的结构

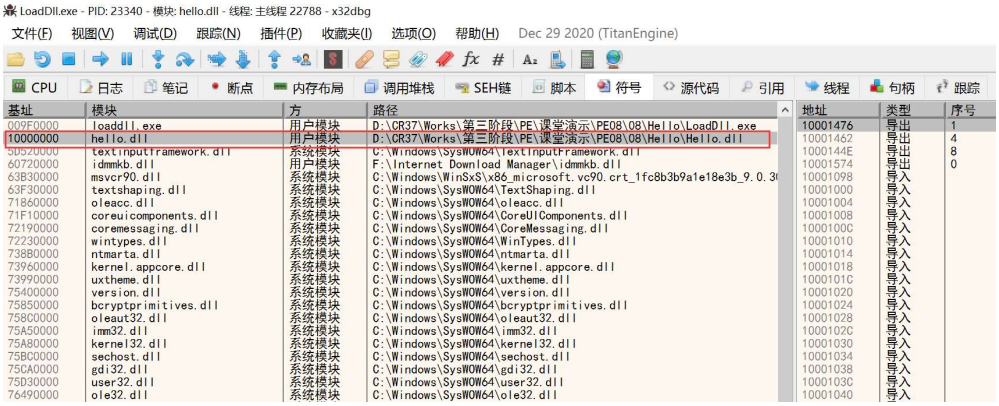
```
//@[comment("MVI_tracked")]
typedef struct _IMAGE_EXPORT_DIRECTORY {
  DWORD Characteristics; // 属性特定，参考
  DWORD TimeDateStamp;  // 日期，参考
  WORD MajorVersion; // 主要版本，参考
  WORD MinorVersion // 次要版本，参考
  DWORD Name; // 模块名称，重要
  DWORD Base; // 序号基址
  DWORD NumberOfFunctions; // 导出函数的总数量
  DWORD NumberOfNames; // 导出函数按照名字导出的数量（导出函数可以有名称以及需要导出，需要进行区分）
  DWORD AddressOfFunctions; // 函数地址表
  DWORD AddressOfNames; // 名称表
```

```
    DWORD AddressOfNameOrdinals;  // 序号表
} IMAGE_EXPORT_DIRECTORY, *PIMAGE_EXPORT_DIRECTORY;
```

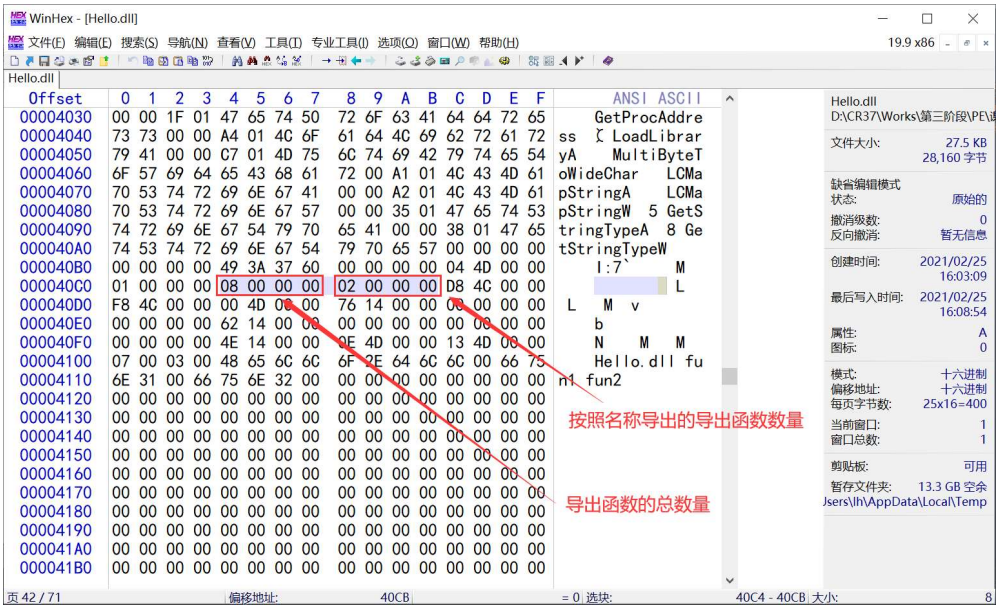
- 模块名称：

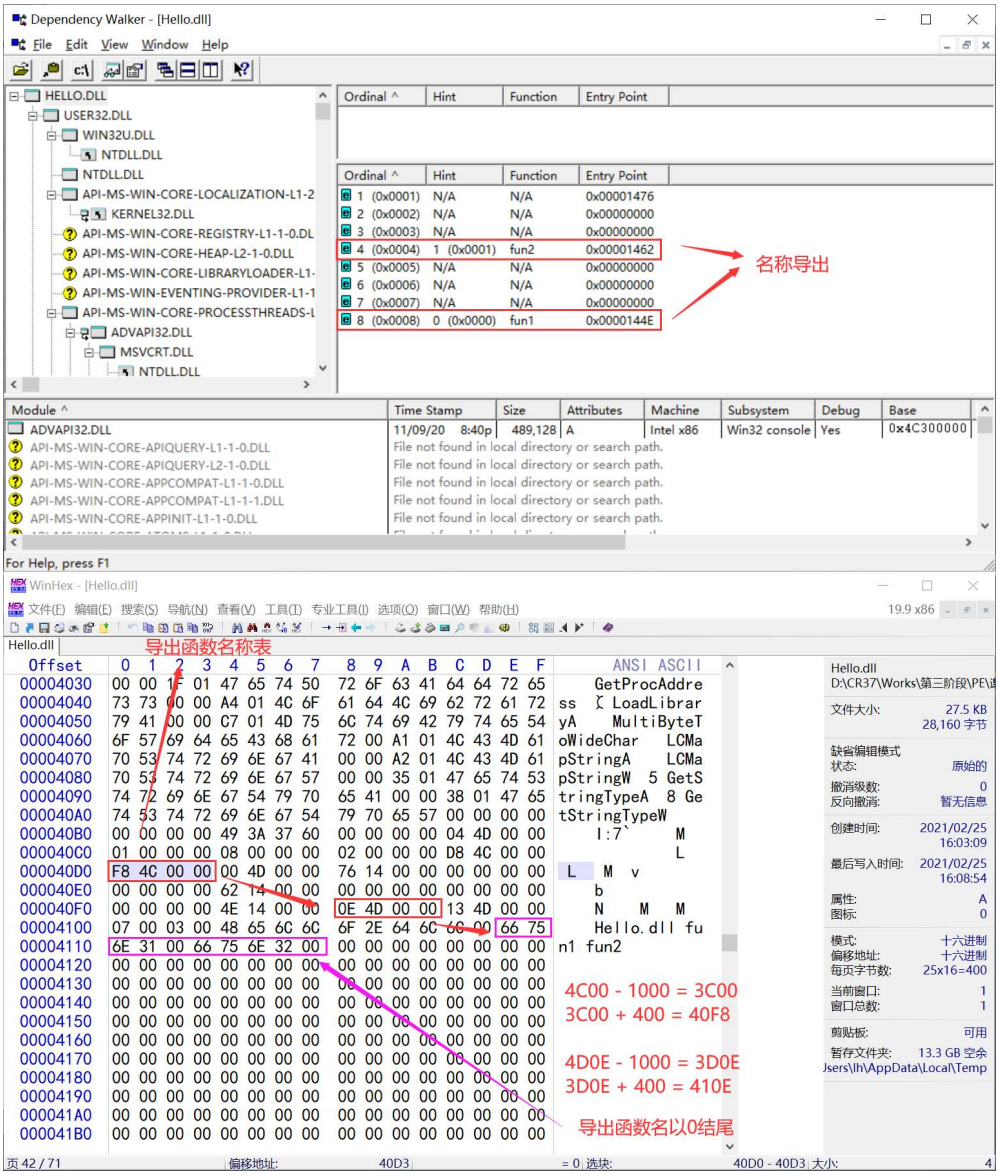


- 模块基址（dll希望被加载的模块基址），操作系统不满足该dll希望被加载的地址时，就需要修正代码，但是一般都会满足：

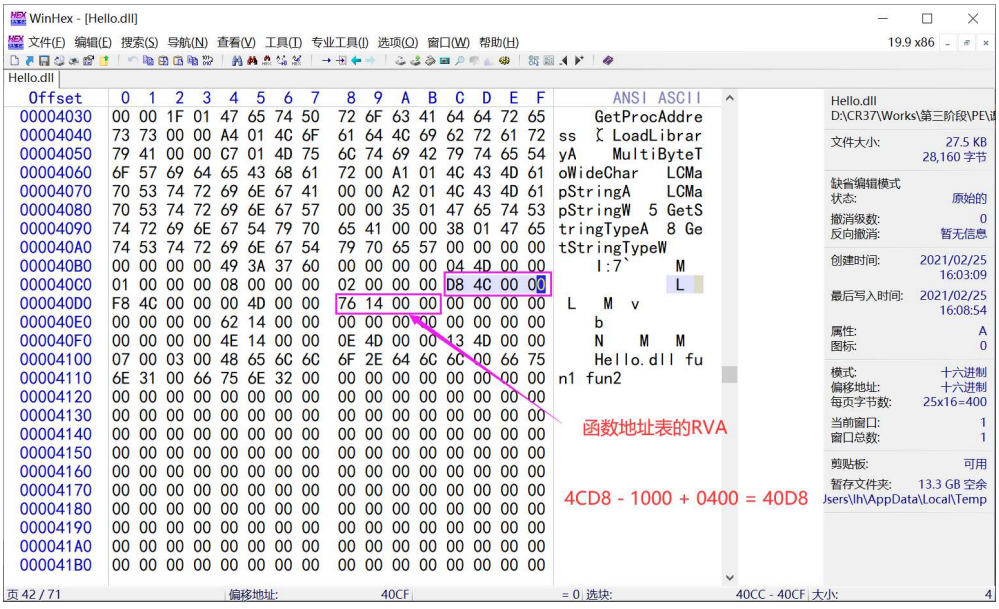


- 导出函数按照名字导出的数量（导出函数可以有名称以及需要导出，需要进行区分），**序号导出的导出函数数量可以由总的导出函数数量减去名字导出的导出函数的数量得出。**

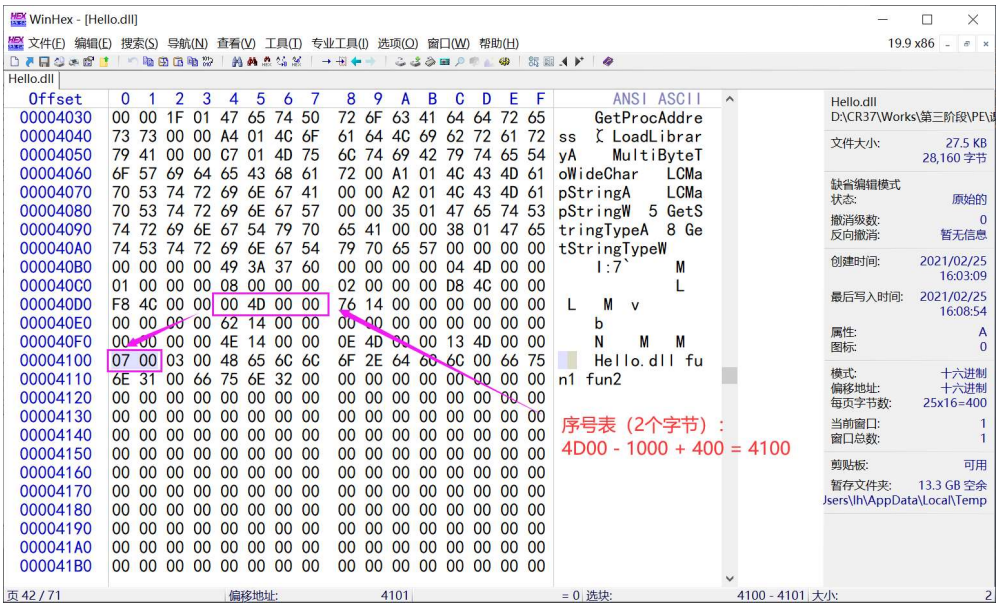




- 函数地址表RVA为（地址表上的值指向的位置上的值作为RVA），名称表中导出函数在表中的项数和函数地址表中的项数一一对应：

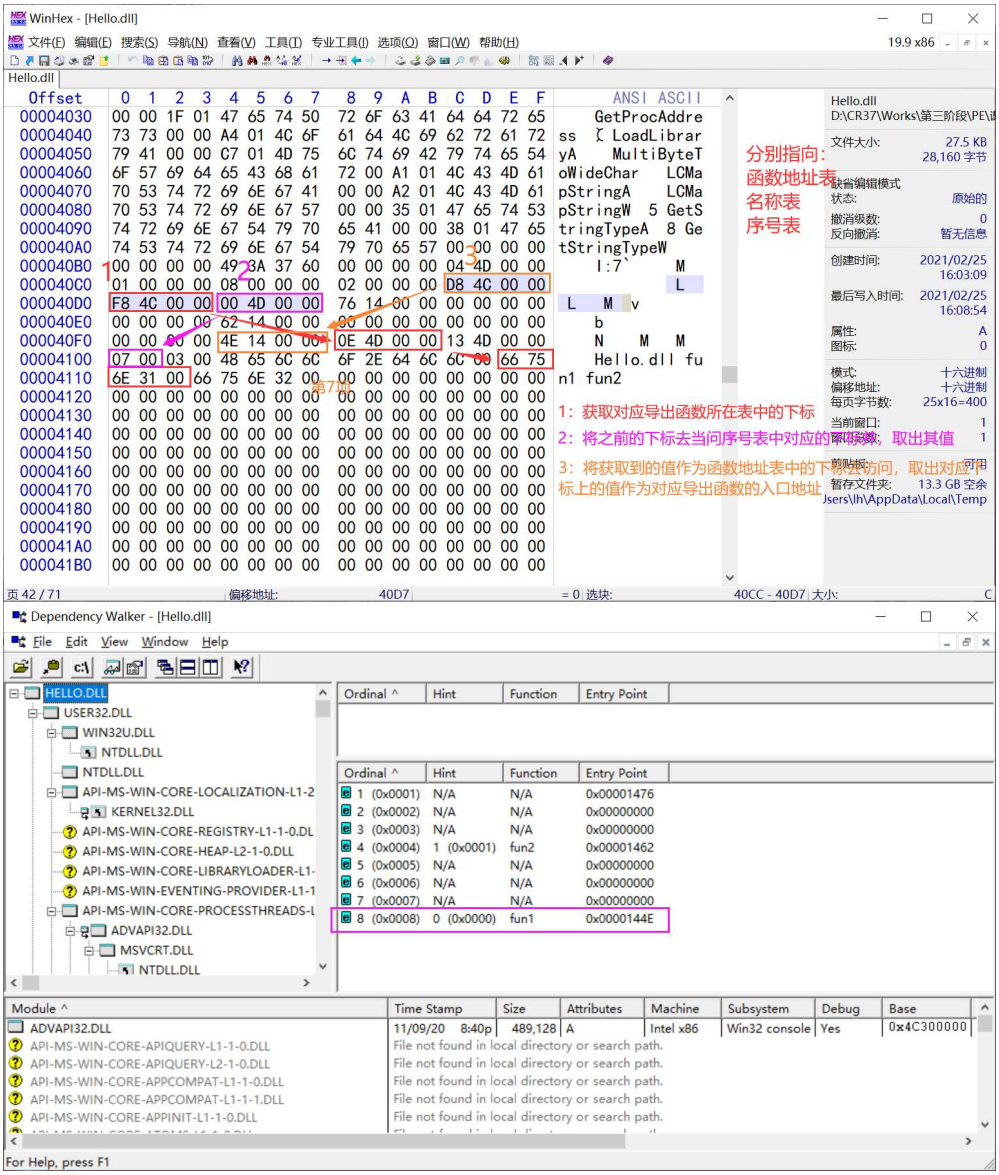


- 序号表（表中每项为2个字节）：

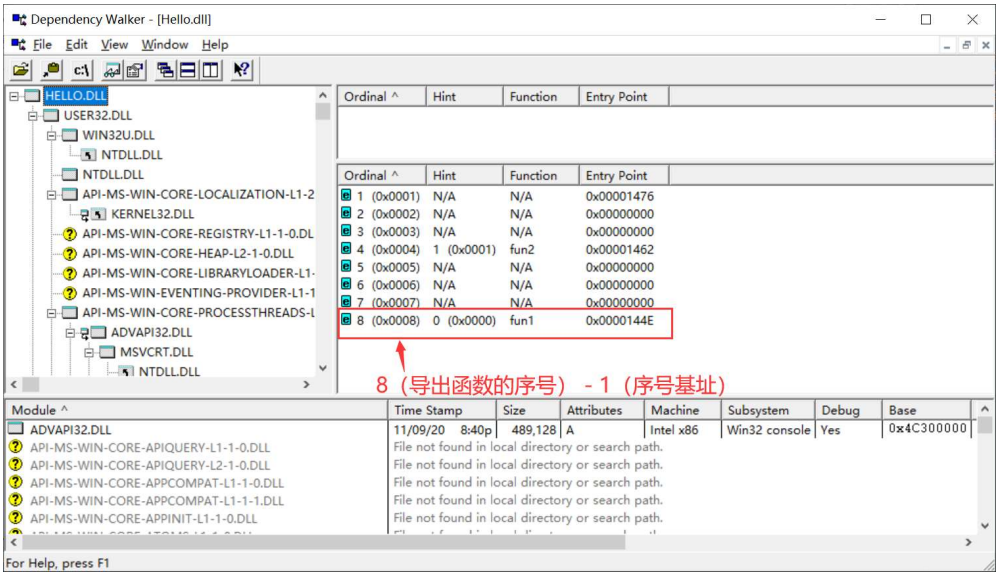


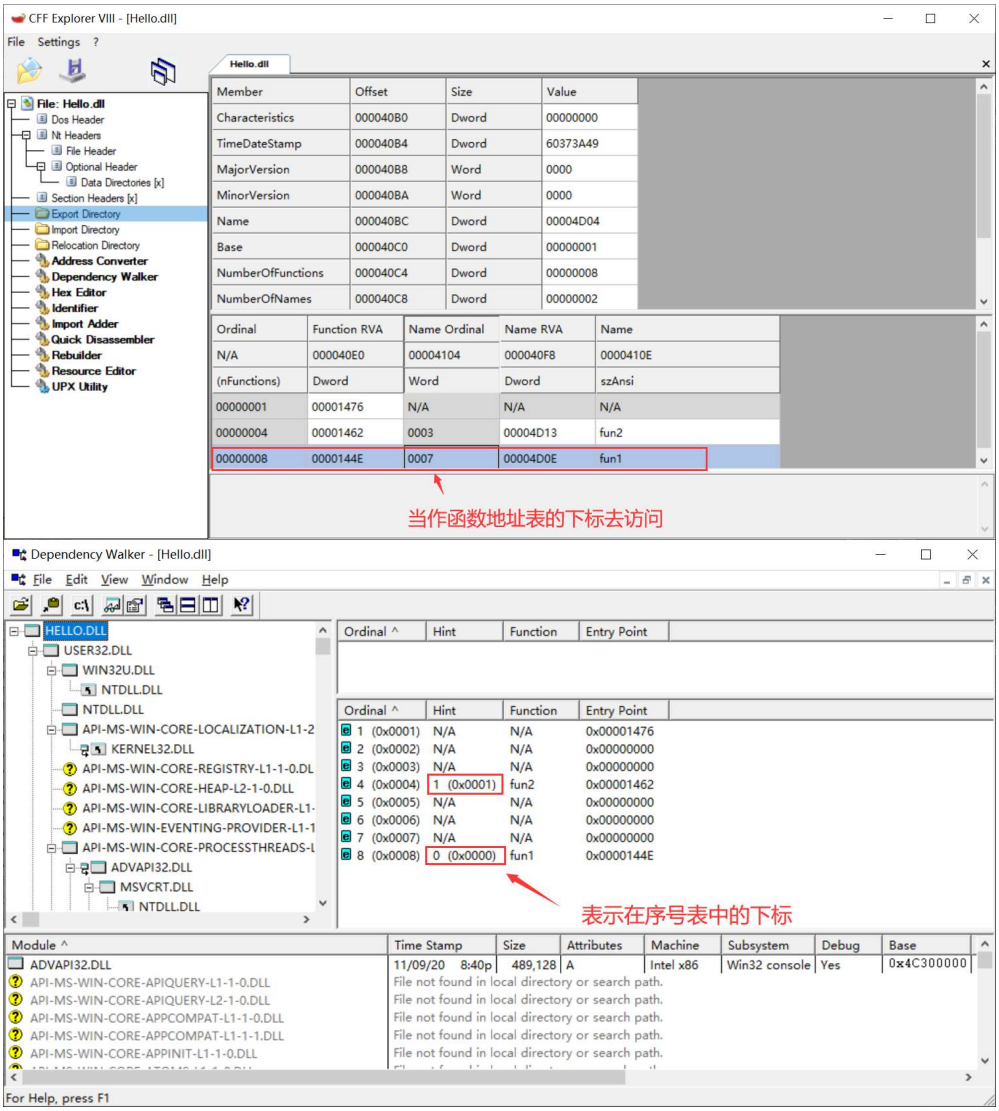
> 正常查询：
> 名称导出：名称表 --> 序号表 --> 函数地址表
> 序号导出：序号表 --> 函数地址表

- 名称导出：通过名称表查询指定的导出函数名称所在表中的下标，再去查询序号表，通过名称表中的下标去获取序号表中对应的下标上的值作为函数地址表中的下标，之后去访问函数地址表，将之前的值作为下标去访问，取出对应下标上的值作为对应导出函数的入口地址，如下入所示：



- 序号导出：通过 "Dependency Walker" 工具查看对应的导出函数的序号，将其序号减去导出表中的序号基址，得出的值当作 "函数地址表" 的下标去访问。

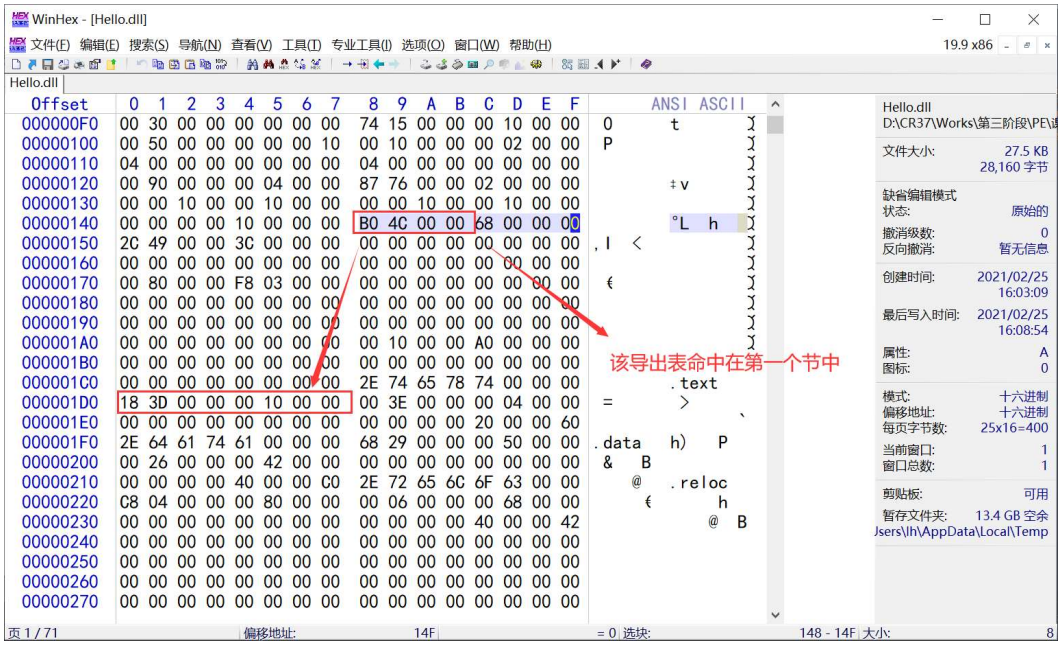


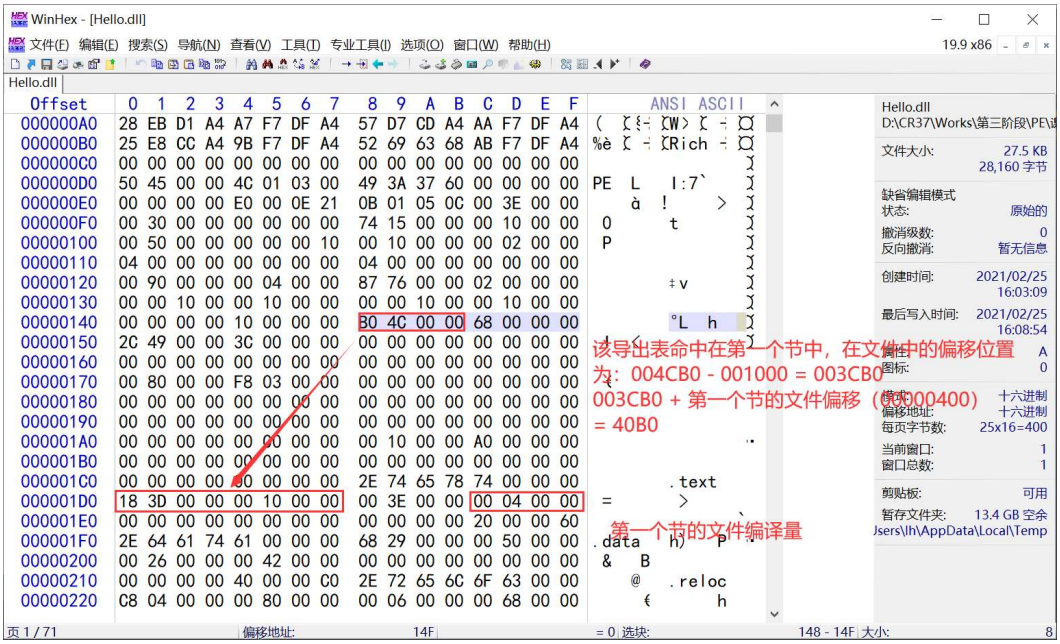


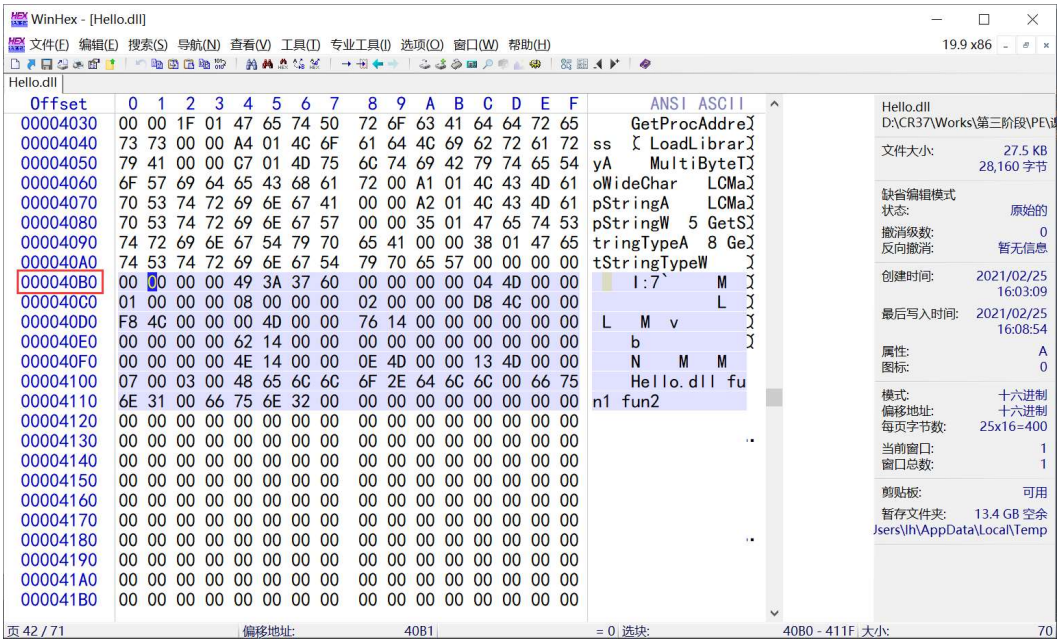
> Dependency Walker 工具就是通过遍历目标 ".dll" 中的导出表显示对应的导出函数信息的

定位导出表的文件偏移位置

定位导出表的位置，所处在哪个节中：




WinHex - [Hello.dll]
文件(F) 编辑(E) 搜索(S) 导航(N) 查看(V) 工具(T) 专业工具(I) 选项(O) 窗口(W) 帮助(H)
19.9 x86
Hello.dll
Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F ANSI ASCII
000000A0 28 EB D1 A4 A7 F7 DF A4 57 D7 CD A4 AA F7 DF A4
000000B0 25 E8 CC A4 9B F7 DF A4 52 69 63 68 AB F7 DF A4
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000D0 50 45 00 00 4C 01 03 00 49 3A 37 60 00 00 00 00
000000E0 00 00 00 00 E0 00 0E 21 0B 01 05 0C 00 3E 00 00
000000F0 00 30 00 00 00 00 00 00 74 15 00 00 00 10 00 00
00000100 00 50 00 00 00 00 00 10 00 10 00 00 00 02 00 00
00000110 04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00
00000120 00 90 00 00 00 04 00 00 87 76 00 00 02 00 00 00
00000130 00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00
00000140 00 00 00 00 10 00 00 00 B0 4C 00 00 68 00 00 00
00000150 2C 49 00 00 3C 00 00 00 00 00 00 00 00 00 00 00
00000160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000170 00 80 00 00 F8 03 00 00 00 00 00 00 00 00 00 00
00000180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001A0 00 00 00 00 00 00 00 00 00 10 00 00 A0 00 00 00
000001B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001C0 00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00
000001D0 18 3D 00 00 00 10 00 00 00 3E 00 00 00 04 00 00
000001E0 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60
000001F0 2E 64 61 74 61 00 00 00 68 29 00 00 00 50 00 00
00000200 00 26 00 00 00 42 00 00 00 00 00 00 00 00 00 00
00000210 00 00 00 00 40 00 00 C0 2E 72 65 6C 6F 63 00 00
00000220 C8 04 00 00 00 80 00 00 00 06 00 00 00 68 00 00
该导出表命中在第一个节中，在文件中的偏移位置为：004CB0 - 001000 = 003CB0
003CB0 + 第一个节的文件偏移（00000400）= 40B0
第一个节的文件编译量
Hello.dll
D:\CR37\Works\第三阶段\PE\
文件大小: 27.5 KB 28,160 字节
缺省编辑模式
状态: 原始的
撤消级数: 0
反向撤消: 暂无信息
创建时间: 2021/02/25 16:03:09
最后写入时间: 2021/02/25 16:08:54
属性: A
图标: 0
模式: 十六进制
偏移地址: 十六进制
每页字节数: 25x16=400
当前窗口: 1
窗口总数: 1
剪贴板: 可用
暂存文件夹: 13.4 GB 空余
Users\lh\AppData\Local\Temp
页 1 / 71 偏移地址: 14F = 0 选块: 148 - 14F 大小: 8


WinHex - [Hello.dll]
文件(F) 编辑(E) 搜索(S) 导航(N) 查看(V) 工具(T) 专业工具(I) 选项(O) 窗口(W) 帮助(H)
19.9 x86
Hello.dll
Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F ANSI ASCII
00004030 00 00 1F 01 47 65 74 50 72 6F 63 41 64 64 72 65 GetProcAddre
00004040 73 73 00 00 A4 01 4C 6F 61 64 4C 69 62 72 61 72 ss LoadLibrar
00004050 79 41 00 00 C7 01 4D 75 6C 74 69 42 79 74 65 54 yA MultiByteT
00004060 6F 57 69 64 65 43 68 61 72 00 A1 01 4C 43 4D 61 oWideChar LCMa
00004070 70 53 74 72 69 6E 67 41 00 00 A2 01 4C 43 4D 61 pStringA LCMa
00004080 70 53 74 72 69 6E 67 57 00 00 35 01 47 65 74 53 pStringW 5 GetS
00004090 74 72 69 6E 67 54 79 70 65 41 00 00 38 01 47 65 tringTypeA 8 Ge
000040A0 74 53 74 72 69 6E 67 54 79 70 65 57 00 00 00 00 tStringTypeW
000040B0 00 00 00 00 49 3A 37 60 00 00 00 00 04 4D 00 00 I:7` M
000040C0 01 00 00 00 08 00 00 00 02 00 00 00 D8 4C 00 00 L
000040D0 F8 4C 00 00 00 4D 00 00 76 14 00 00 00 00 00 00 L M v
000040E0 00 00 00 00 62 14 00 00 00 00 00 00 00 00 00 00 b
000040F0 00 00 00 00 4E 14 00 00 0E 4D 00 00 13 4D 00 00 N M M
00004100 07 00 03 00 48 65 6C 6C 6F 2E 64 6C 6C 00 66 75 Hello.dll fu
00004110 6E 31 00 66 75 6E 32 00 00 00 00 00 00 00 00 00 n1 fun2
00004120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000041A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000041B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Hello.dll
D:\CR37\Works\第三阶段\PE\
文件大小: 27.5 KB 28,160 字节
缺省编辑模式
状态: 原始的
撤消级数: 0
反向撤消: 暂无信息
创建时间: 2021/02/25 16:03:09
最后写入时间: 2021/02/25 16:08:54
属性: A
图标: 0
模式: 十六进制
偏移地址: 十六进制
每页字节数: 25x16=400
当前窗口: 1
窗口总数: 1
剪贴板: 可用
暂存文件夹: 13.4 GB 空余
Users\lh\AppData\Local\Temp
页 42 / 71 偏移地址: 40B1 = 0 选块: 40B0 - 411F 大小: 70