FileInfo.dll

函数1：输入文件路径，得到文件的MFT记录号、文件所有的簇号和文件内容

|  |  |  |
| --- | --- | --- |
| **函数1：？ FileInfo(？ filepath)** | | |
| **输入：** | | 文件路径 |
| **输出：** | | 文件的MFT记录号，文件所有的簇号，文件内容 |
| 1 | ? FindMFTRecNum(? filepath) // 得到MFT记录号 | |
| 2 | ? DumpClusNum(ULONG index) // 同时得到簇号和文件内容 | |

函数1.1 输入文件路径，得到文件的MFT记录号

|  |  |  |
| --- | --- | --- |
| **函数1.1：? FindMFTRecNum(? filepath)** | | |
| **输入：** | | 文件路径 |
| **输出：** | | 文件的MFT记录号 |
| 1 | ParsePath(filepath); //解析文件路径 | |
| 2 | TransformCode(s); //将分解后的目录名转成unicode编码 | |
| 3 | index = 5; | |
| 4 | FOR (0 … n-1) //分解得到n个目录 | |
| 5 | ReadFileRecord(index, file); //将记录号为index的文件记录读到内存空间file | |
| 6 | FindAttribute(file, AttributeIndexRoot, 0); // 在内存空间file中找90属性 | |
| 7 | ReadAttribute(attr, buf1); // 将内存空间file中的90常驻属性写到buf1中 | |
| 8 | IF (无A0属性) // 90属性第一种情况 | |
| 9 | index = TraverseIndexEntry(AttributeIndexRoot, buf1, 下一个目录名) | |
| 10 | continue; | |
| 11 | ELSE IF (第一个索引项偏移4@0x30 == 0 && 2@3C == 1) // 90属性第二、四种情况，为大索引，有A0、B0属性 | |
| 12 | FindAttribute(file, AttributeIndexAllocation, 0);//在内存file中找A0属性 | |
| 13 | ReadAttribute(attr, buf2); // 将内存空间file中的A0属性写到buf2中，A0是非常驻属性，所以索引节点的数据全被写入buf2（可以不考虑B0属性） | |
| 14 | index = TraverseIndexEntry(AttributeIndexAllocation, buf2, 下一个目录名) // 遍历buf2中索引节点的索引项，找到下一个目录的MFT记录号 | |
| 15 | continue; | |
| 16 | ELSE IF (偏移4@0x30 == 0x10 && 2@3C == 1) // 90属性第三种情况 | |
| 17 | indexTemp = TraverseIndexEntry(AttributeIndexRoot, buf1, 下一个目录名) // 分别遍历90内的索引项和A0指向的索引节点，找目标文件名 | |
| 18 | IF (indexTemp == 0) //在90属性中没找到 | |
| 19 | FindAttribute(file, AttributeIndexAllocation, 0);//在内存file中找A0属性 | |
| 20 | ReadAttribute(attr, buf2); | |
| 21 | index = TraverseIndexEntry(AttributeIndexAllocation, buf2, 下一个目录名) | |
| 22 | ELSE | |
| 23 | index = indexTemp; | |
| 24 | continue; | |

函数1.2 遍历索引节点的索引项，得到目标索引项的MFT记录号（注意长文件名）

|  |  |  |  |
| --- | --- | --- | --- |
| **函数1.2：** | | | **ULONG TraverseIndexEntry (ATTRIBUTE\_TYPE type, PVOID buf, ? indname)** |
| **输入：** | | 目录名 | |
| **输出：** | | 目录的MFT记录号 | |
| 1 | IF (AttributeIndexRoot) //buf中是90属性从第一个索引项开始的属性体，直接遍历索引项 | | |
| 2 |  | | |
| 3 | ELSE IF (AttributeIndexAllocation) //buf中是一个个以INDX开始的索引节点（簇） ，先按索引节点遍历；再在节点内部遍历索引项，当遍历到索引项大小为0x10或0x18时表示该节点的索引项已遍历完，接着遍历下一个节点（簇）。 | | |
|  |  | | |

函数1.3 输入文件路径，得到文件所有的簇号和文件内容

|  |  |  |
| --- | --- | --- |
| **函数1.3：VOID DumpClusNum(ULONG index, PVOID clusbuf, PVOID buf)** | | |
| **输入：** | | 文件路径 |
| **输出：** | | 文件所有的簇号，文件内容 |
| 1 | ReadFileRecord(index, file);//将记录号为index的文件记录读到内存空间file中 | |
| 2 | FindAttribute(file, AttributeData, 0); //在内存空间file中找80属性 | |
| 3 | ReadAttribute(attr, clusbuf, buf); //将内存空间file中的80属性写到buf中,同时将簇号写到clusbuf | |

函数2 输入簇号和簇数，得到簇中的数据

|  |  |  |
| --- | --- | --- |
| **函数2：VOID DumpData(ULONGLONG lcn, ULONG readcount, PVOID buffer)** | | |
| **输入：** | | 簇号，簇数 |
| **输出：** | | 簇中的数据 |
| 1 | lcn -> sector number // 簇号转换为扇区号 | |
| 2 | VOID ReadSector(ULONGLONG sector, ULONG count, PVOID buffer) // 读扇区 | |