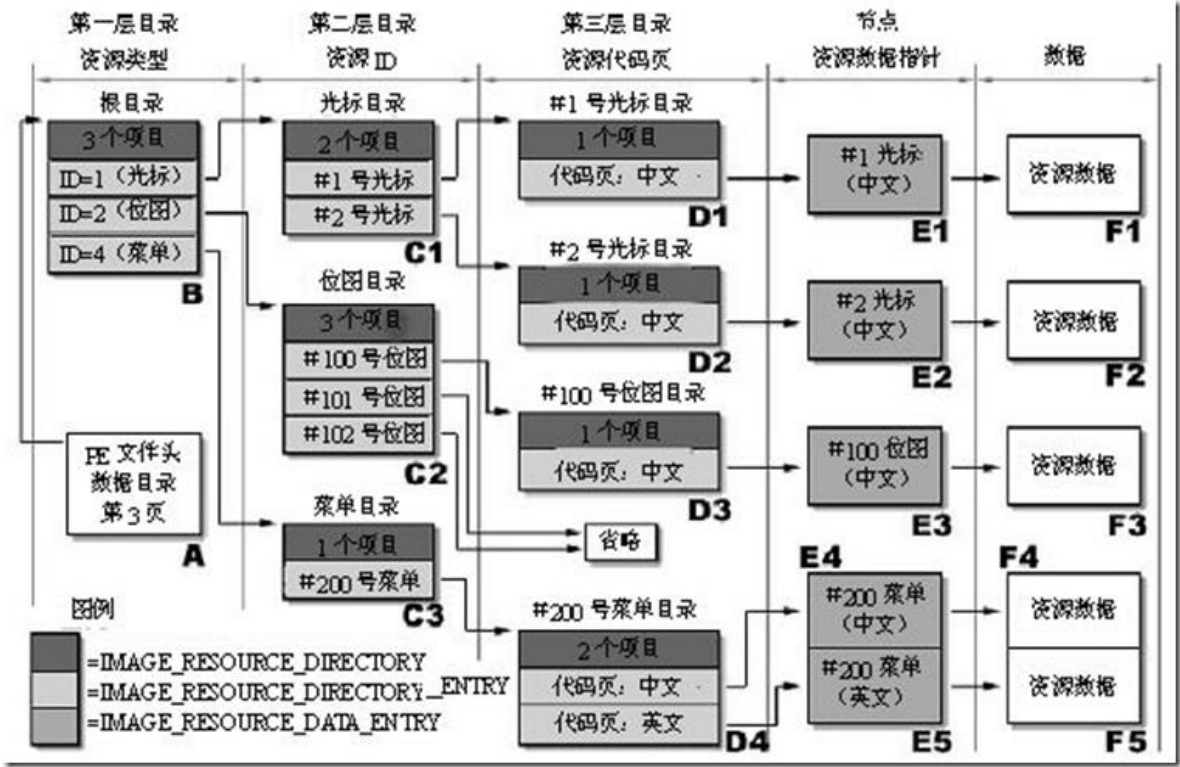


资源表

资源在PE中的存储是按照目录存储的一般是三层，每一层都是有一个 `IMAGE_RESOURCE_DIRECTORY` 作为一个索引后边跟着 `IMAGE_RESOURCE_DIR_ENTRY`



IMAGE_RESOURCE_DIRECTORY

```
typedef struct _IMAGE_RESOURCE_DIRECTORY {
    DWORD Characteristics;
    DWORD TimeDateStamp;
    WORD MajorVersion;
    WORD MinorVersion;
    WORD NumberOfNamedEntries; // 以名称（字符串）命名的入口数量
    WORD NumberOfIdEntries; // 以ID（整型数字）命名的入口数量
    // IMAGE_RESOURCE_DIRECTORY_ENTRY DirectoryEntries[];
} IMAGE_RESOURCE_DIRECTORY, *PIMAGE_RESOURCE_DIRECTORY;
```

紧随其后的是 `IMAGE_RESOURCE_DIRECTORY_ENTRY` 结构

IMAGE_RESOURCE_DIRECTORY_ENTRY

```
typedef struct _IMAGE_RESOURCE_DIRECTORY_ENTRY {
    union {
        struct {
            DWORD NameOffset : 31; // 资源名偏移
            DWORD NameIsString : 1; // 资源名为字符串，这两个字段最高位为1则是名称
        };
    };
};
```

```

        DWORD    Name;        // 资源/语言类型
        WORD     Id;          // 资源数字ID
    };
    union {
        DWORD    OffsetToData;        // 数据偏移地址
        struct {
            DWORD    OffsetToDirectory : 31;        // 子目录偏移地址
            DWORD    DataIsDirectory : 1;           // 数据为目录，这两个字段最高位为1
        }
    };
} IMAGE_RESOURCE_DIRECTORY_ENTRY, *PIMAGE_RESOURCE_DIRECTORY_ENTRY;

```

则是目录

偏移均为节内偏移

资源类型详见<https://docs.microsoft.com/en-us/windows/win32/menurc/resource-types>

IMAGE_RESOURCE_DIR_STRING_U

```

typedef struct _IMAGE_RESOURCE_DIR_STRING_U {
    WORD    Length; // 字符串长度
    WCHAR    NameString[1]; // 字符串数组
} IMAGE_RESOURCE_DIR_STRING_U, *PIMAGE_RESOURCE_DIR_STRING_U;

```

IMAGE_RESOURCE_DATA_ENTRY

```

typedef struct _IMAGE_RESOURCE_DATA_ENTRY {
    DWORD    OffsetToData;    // 资源数据的RVA
    DWORD    Size;            // 资源数据的长度
    DWORD    CodePage;        // 代码页
    DWORD    Reserved;        // 保留字段
} IMAGE_RESOURCE_DATA_ENTRY, *PIMAGE_RESOURCE_DATA_ENTRY;

```