

Malicious Code Analysis

Fangtian Zhong
CSCI 591

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu





Overview

01

PE Structures

02

Examples

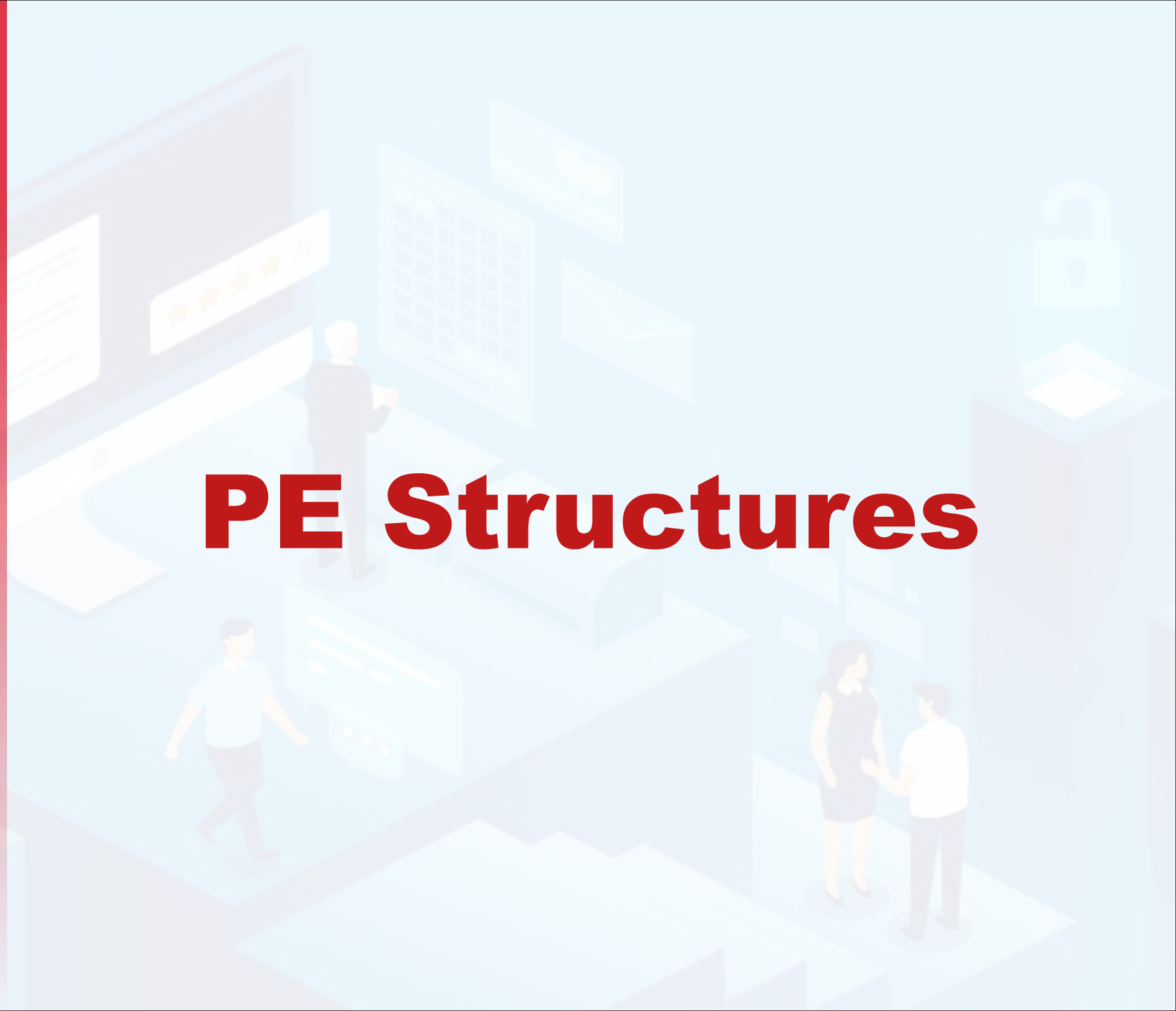


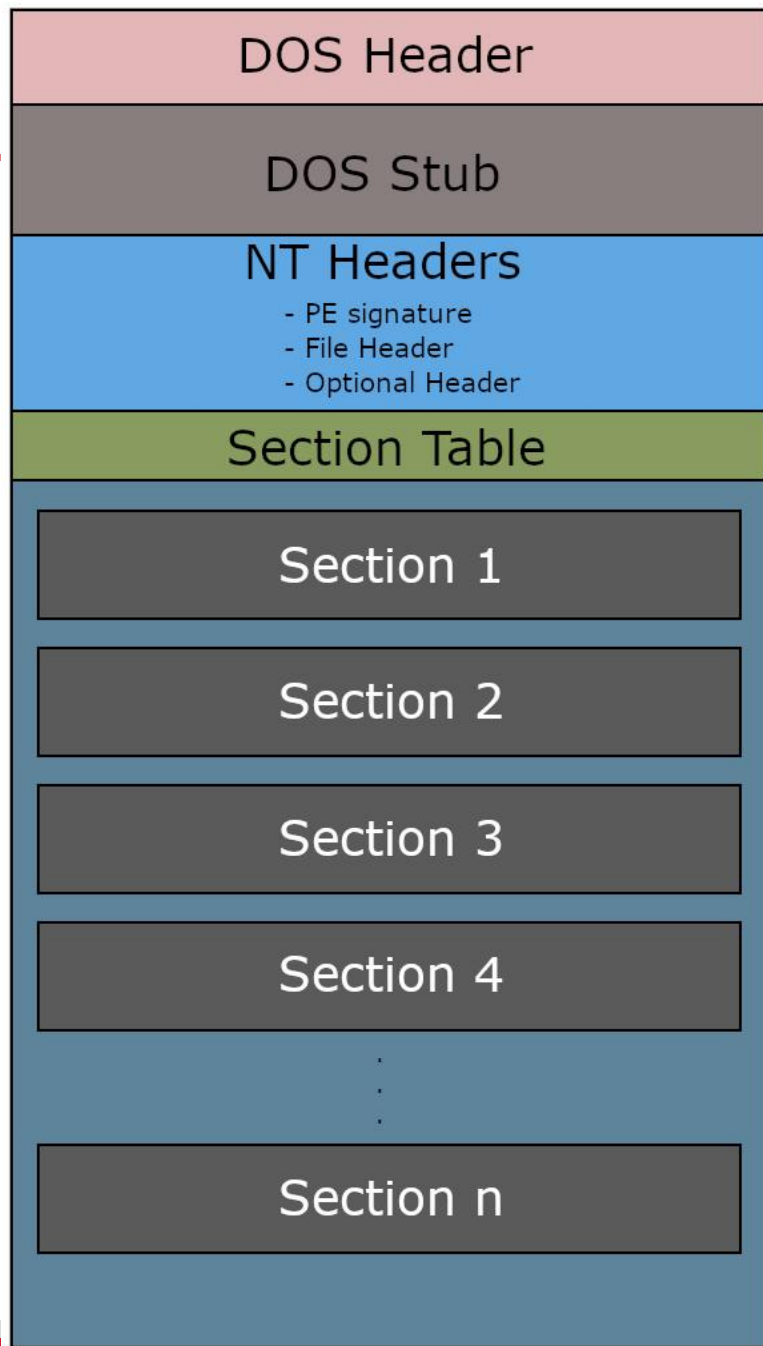
Part One

01

2025-8-26

PE Structures







DOS Header

```
typedef struct _IMAGE_DOS_HEADER {    // DOS .EXE header
    WORD   e_magic;                  // Magic number
    WORD   e_cblp;                   // Bytes on last page of file
    WORD   e_cp;                     // Pages in file
    WORD   e_crlc;                   // Relocations
    WORD   e_cparhdr;                // Size of header in paragraphs
    WORD   e_minalloc;               // Minimum extra paragraphs needed
    WORD   e_maxalloc;               // Maximum extra paragraphs needed
    WORD   e_ss;                     // Initial (relative) SS value
    WORD   e_sp;                     // Initial SP value
    WORD   e_csum;                   // Checksum
    WORD   e_ip;                     // Initial IP value
    WORD   e_cs;                     // Initial (relative) CS value
    WORD   e_lfarlc;                 // File address of relocation table
    WORD   e_ovno;                   // Overlay number
    WORD   e_res[4];                 // Reserved words
    WORD   e_oemid;                  // OEM identifier (for e_oeminfo)
    WORD   e_oeminfo;                // OEM information; e_oemid specific
    WORD   e_res2[10];               // Reserved words
    LONG   e_lfanew;                 // File address of new exe header
} IMAGE_DOS_HEADER, *PIMAGE_DOS_HEADER;
```

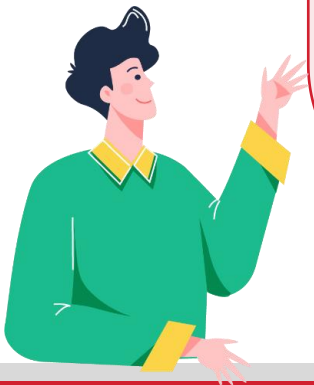




NT Header

```
typedef struct _IMAGE_NT_HEADERS64 {  
    DWORD Signature;  
    IMAGE_FILE_HEADER FileHeader;  
    IMAGE_OPTIONAL_HEADER64 OptionalHeader;  
} IMAGE_NT_HEADERS64, *PIMAGE_NT_HEADERS64;
```

```
typedef struct _IMAGE_NT_HEADERS {  
    DWORD Signature;  
    IMAGE_FILE_HEADER FileHeader;  
    IMAGE_OPTIONAL_HEADER32 OptionalHeader;  
} IMAGE_NT_HEADERS32, *PIMAGE_NT_HEADERS32;
```





COFF Header

```
typedef struct _IMAGE_FILE_HEADER {  
    WORD    Machine;  
    WORD    NumberOfSections;  
    DWORD   TimeDateStamp;  
    DWORD   PointerToSymbolTable;  
    DWORD   NumberOfSymbols;  
    WORD    SizeOfOptionalHeader;  
    WORD    Characteristics;  
} IMAGE_FILE_HEADER, *PIMAGE_FILE_HEADER;
```





Optional Header



There are two versions of the Optional Header, one for 32-bit executables and one for 64-bit executables.

- The size of the structure itself (or the number of members defined within the structure): `IMAGE_OPTIONAL_HEADER32` has 31 members while `IMAGE_OPTIONAL_HEADER64` only has 30 members, that additional member in the 32-bit version is a `DWORD` named `BaseOfData` which holds an RVA of the beginning of the data section.
- The data type of some of the members: The following 5 members of the Optional Header structure are defined as `DWORD` in the 32-bit version and as `ULONGLONG` in the 64-bit version:

ImageBase

SizeOfStackReserve

SizeOfStackCommit

SizeOfHeapReserve

SizeOfHeapCommit



Optional Header

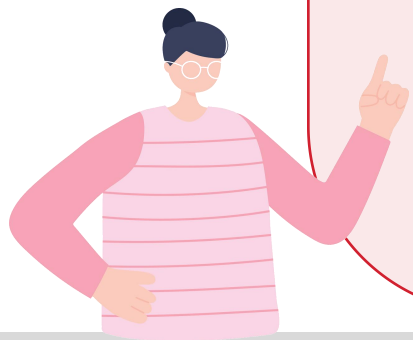
```
typedef struct _IMAGE_OPTIONAL_HEADER64 {  
    WORD    Magic;  
    BYTE    MajorLinkerVersion;  
    BYTE    MinorLinkerVersion;  
    DWORD    SizeOfCode;  
    DWORD    SizeOfInitializedData;  
    DWORD    SizeOfUninitializedData;  
    DWORD    AddressOfEntryPoint;  
    DWORD    BaseOfCode;  
    ULONGLONG ImageBase;  
    DWORD    SectionAlignment;  
    DWORD    FileAlignment;  
    WORD    MajorOperatingSystemVersion;  
    WORD    MinorOperatingSystemVersion;  
    WORD    MajorImageVersion;  
    WORD    MinorImageVersion;  
    WORD    MajorSubsystemVersion;  
    WORD    MinorSubsystemVersion;  
    DWORD    Win32VersionValue;  
    DWORD    SizeOfImage;
```

```
    DWORD    SizeOfHeaders;  
    DWORD    CheckSum;  
    WORD    Subsystem;  
    WORD    DllCharacteristics;  
    ULONGLONG SizeOfStackReserve;  
    ULONGLONG SizeOfStackCommit;  
    ULONGLONG SizeOfHeapReserve;  
    ULONGLONG SizeOfHeapCommit;  
    DWORD    LoaderFlags;  
    DWORD    NumberOfRvaAndSizes;  
    IMAGE_DATA_DIRECTORY  
    DataDirectory[IMAGE_NUMBEROF_DIRECTORY_ENTRIES];  
} IMAGE_OPTIONAL_HEADER64;  
*PIMAGE_OPTIONAL_HEADER64;
```



Section Headers

```
typedef struct _IMAGE_SECTION_HEADER {  
    BYTE    Name[IMAGE_SIZEOF_SHORT_NAME];  
    union {  
        DWORD    PhysicalAddress;  
        DWORD    VirtualSize;  
    } Misc;  
    DWORD    VirtualAddress;  
    DWORD    SizeOfRawData;  
    DWORD    PointerToRawData;  
    DWORD    PointerToRelocations;  
    DWORD    PointerToLinenumbers;  
    WORD     NumberOfRelocations;  
    WORD     NumberOfLinenumbers;  
    DWORD    Characteristics;  
} IMAGE_SECTION_HEADER, *PIMAGE_SECTION_HEADER;
```





Data Directories



```
typedef struct _IMAGE_DATA_DIRECTORY {  
    DWORD   VirtualAddress;  
    DWORD   Size;  
} IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
```



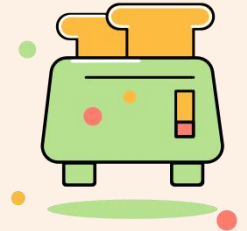
Data Directories

```
#define IMAGE_DIRECTORY_ENTRY_EXPORT      0 // Export Directory
#define IMAGE_DIRECTORY_ENTRY_IMPORT      1 // Import Directory
#define IMAGE_DIRECTORY_ENTRY_RESOURCE    2 // Resource Directory
#define IMAGE_DIRECTORY_ENTRY_EXCEPTION    3 // Exception Directory
#define IMAGE_DIRECTORY_ENTRY_SECURITY    4 // Security Directory
#define IMAGE_DIRECTORY_ENTRY_BASERELOC    5 // Base Relocation Table
#define IMAGE_DIRECTORY_ENTRY_DEBUG        6 // Debug Directory
// IMAGE_DIRECTORY_ENTRY_COPYRIGHT        7 // (X86 usage)
#define IMAGE_DIRECTORY_ENTRY_ARCHITECTURE 7 // Architecture Specific Data
#define IMAGE_DIRECTORY_ENTRY_GLOBALPTR    8 // RVA of GP
#define IMAGE_DIRECTORY_ENTRY_TLS          9 // TLS Directory
#define IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG 10 // Load Configuration Directory
#define IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT 11 // Bound Import Directory in headers
#define IMAGE_DIRECTORY_ENTRY_IAT         12 // Import Address Table
#define IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT 13 // Delay Load Import Descriptors
#define IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR 14 // COM Runtime descriptor
```



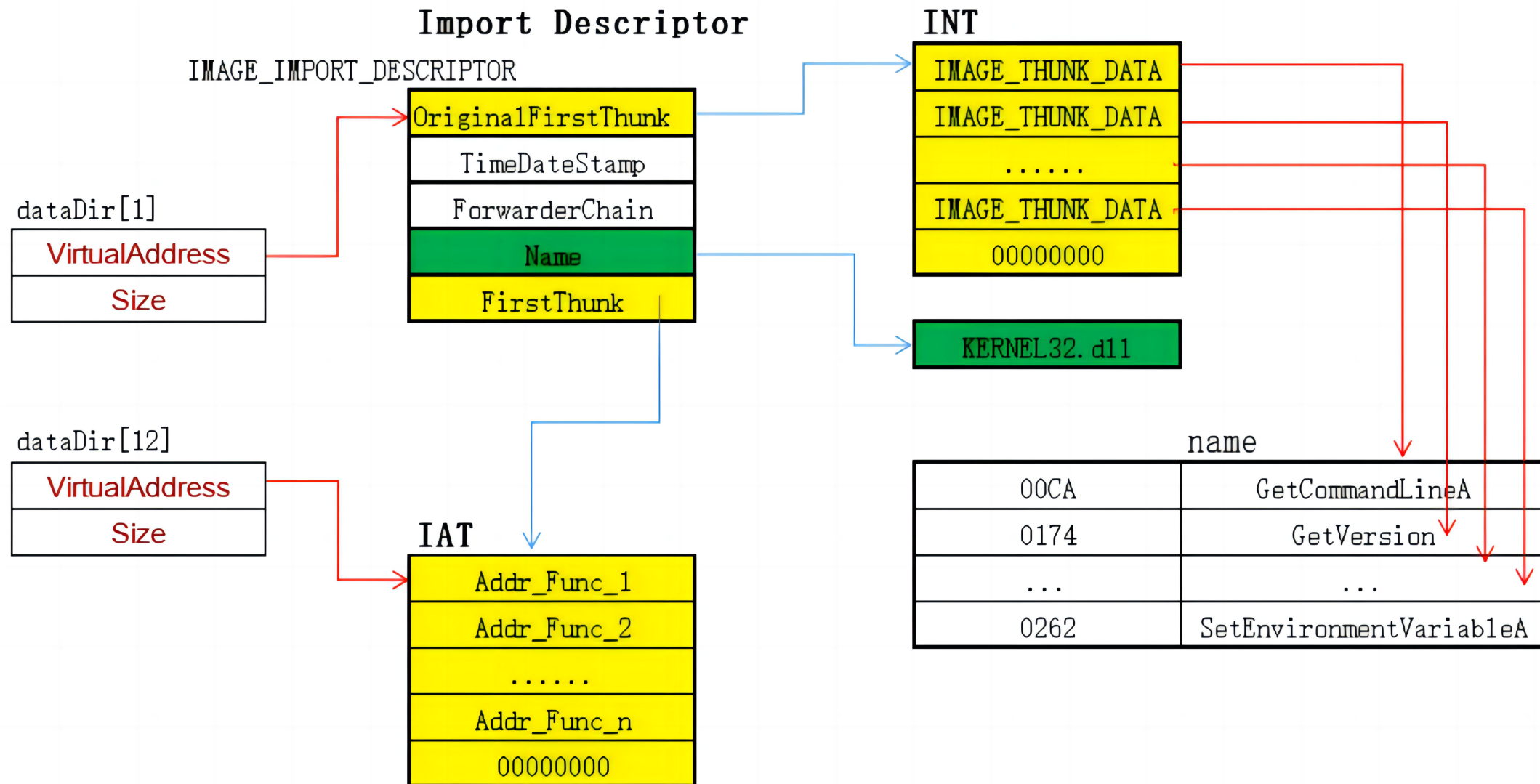
Import Descriptor

```
typedef struct _IMAGE_IMPORT_DESCRIPTOR {  
    union {  
        DWORD Characteristics;  
        DWORD OriginalFirstThunk;  
    } DUMMYUNIONNAME;  
    DWORD TimeDateStamp;  
    DWORD ForwarderChain;  
    DWORD Name;  
    DWORD FirstThunk;  
} IMAGE_IMPORT_DESCRIPTOR;  
typedef IMAGE_IMPORT_DESCRIPTOR UNALIGNED *PIMAGE_IMPORT_DESCRIPTOR;
```





Summary

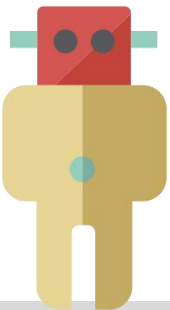




Import Name Table (INT for 32 bit)



```
typedef struct _IMAGE_THUNK_DATA32 {  
    union {  
        DWORD ForwarderString;    // PBYTE  
        DWORD Function;           // PDWORD  
        DWORD Ordinal;  
        DWORD AddressOfData;      //RVA to _IMAGE_IMPORT_BY_NAME  
    } u1;  
} IMAGE_THUNK_DATA32;  
typedef IMAGE_THUNK_DATA32 * PIMAGE_THUNK_DATA32;
```



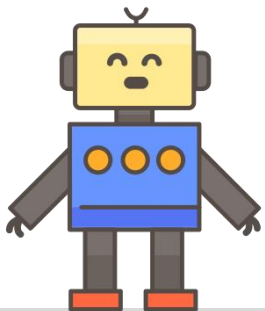


Import Name Table (INT)



Each entry of these entries encodes information as follows:

- ☐ **Bit 31/63 (most significant bit):** This is called the Ordinal/Name flag, it specifies whether to import the function by name or by ordinal.
- ☐ **Bits 15-0:** If the Ordinal/Name flag is set to 1 these bits are used to hold the 16-bit ordinal number that will be used to import the function, bits 30-15/62-15 for PE32/PE32+ must be set to 0.
- ☐ **Bits 30-0:** If the Ordinal/Name flag is set to 0 these bits are used to hold an RVA of a Hint/Name table.





Hint/Name Table

```
typedef struct _IMAGE_IMPORT_BY_NAME {  
    WORD    Hint;  
    CHAR    Name[1];  
} IMAGE_IMPORT_BY_NAME, *PIMAGE_IMPORT_BY_NAME;
```

 **Hint:** A word that contains a number, this number is used to look-up the function, that number is first used as an index into the export name pointer table, if that initial check fails a binary search is performed on the DLL's export name pointer table.

 **Name:** A null-terminated string that contains the name of the function to import.



Data Directories

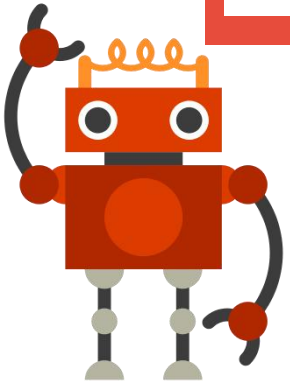


```
typedef struct _IMAGE_DATA_DIRECTORY {  
    DWORD   VirtualAddress;  
    DWORD   Size;  
} IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
```



Import Address Table (IAT)

- ❑ On disk, the IAT is identical to the ILT, however during bounding when the binary is being loaded into memory, the entries of the IAT get overwritten with the addresses of the functions that are being imported.





Data Directories



```
typedef struct _IMAGE_DATA_DIRECTORY {  
    DWORD   VirtualAddress;  
    DWORD   Size;  
} IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
```



Export Directory Table

```
typedef struct _IMAGE_EXPORT_DIRECTORY {  
    DWORD Characteristics;           // Reserved, must be 0.  
    DWORD TimeDateStamp;             //The time and date that the export data was created.  
    WORD MajorVersion;               //The major version number.  
    WORD MinorVersion;               //The minor version number.  
    DWORD Name;                      //The address of the ASCII string that contains the name of the DLL.  
    DWORD Base;                      //The starting ordinal number for exports in this image.  
                                     //It is usually set to 1.  
    DWORD NumberOfFunctions;         //The number of entries in the export address table.  
    DWORD NumberOfNames;             //The number of entries in the name pointer table.  
    DWORD AddressOfFunctions;        //The address of the export address table.  
    DWORD AddressOfNames;            //The address of the export name pointer table.  
    DWORD AddressOfNameOrdinals;     //The address of the ordinal table, relative to the image base.  
} IMAGE_EXPORT_DIRECTORY, *PIMAGE_EXPORT_DIRECTORY;
```



Data Directories



```
typedef struct _IMAGE_DATA_DIRECTORY {  
    DWORD   VirtualAddress;  
    DWORD   Size;  
} IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
```



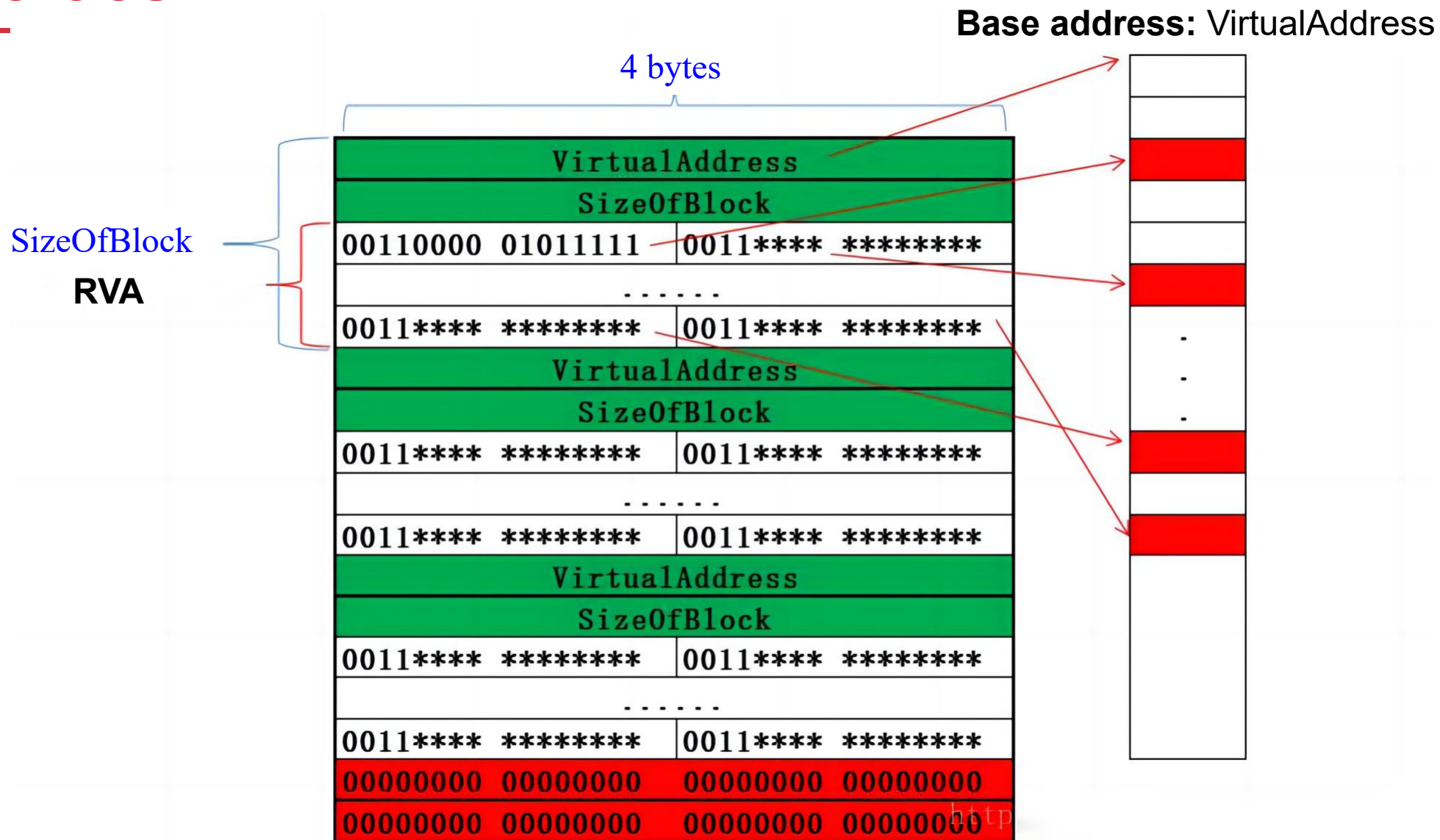
Relocation Table



```
Typedef struct _IMAGE_BASE_RELOCATION{  
    DWORD VirtualAddress  
    DWORD SizeOfBlock  
}IMAGE_BASE_RELOCATION,*PIMAGE_BASE_RELOCATION;
```



Relocs





Part Two

02

2025-8-26

Examples

An isometric illustration of a modern office environment. It features several people: a man in a suit standing near a large screen displaying a calendar, a man in a white shirt walking, and a woman in a dark dress talking to a man in a white shirt. There are various digital screens and floating icons, including a padlock, a calendar, and a document, all in a light blue and white color scheme.



DOS Header



```
typedef struct _IMAGE_DOS_HEADER {    // DOS .EXE header
    WORD   e_magic;                  // Magic number
    WORD   e_cblp;                   // Bytes on last page of file
    WORD   e_cp;                     // Pages in file
    WORD   e_crlc;                   // Relocations
    WORD   e_cparhdr;                // Size of header in paragraphs
    WORD   e_minalloc;               // Minimum extra paragraphs needed
    WORD   e_maxalloc;               // Maximum extra paragraphs needed
    WORD   e_ss;                     // Initial (relative) SS value
    WORD   e_sp;                     // Initial SP value
    WORD   e_csum;                   // Checksum
    WORD   e_ip;                     // Initial IP value
    WORD   e_cs;                     // Initial (relative) CS value
    WORD   e_lfarlc;                 // File address of relocation table
    WORD   e_ovno;                   // Overlay number
    WORD   e_res[4];                 // Reserved words
    WORD   e_oemid;                  // OEM identifier (for e_oeminfo)
    WORD   e_oeminfo;                // OEM information; e_oemid specific
    WORD   e_res2[10];               // Reserved words
    LONG   e_lfanew;                 // File address of new exe header
} IMAGE_DOS_HEADER, *PIMAGE_DOS_HEADER;
```



PE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00
10	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00
40	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68
50	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F
60	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20
70	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00
80	50	45	00	00	64	86	0B	00	67	C0	A3	63	00	00	00	00
90	00	00	00	00	F0	00	2E	02	0B	00	00	00	00	00	00	00
A0	00	44	00	00	00	02	00	00	B0	10	00	00	00	00	00	00
B0	00	00	00	40	01	00	00	00	00	10	00	00	00	00	00	00
C0	04	00	00	00	00	00	00	00	05	00	00	00	00	00	00	00
D0	00	D0	00	00	00	04	00	00	0E	BA	00	00	02	00	00	01
E0	00	00	20	00	00	00	00	00	00	10	00	00	00	00	00	00
F0	00	00	10	00	00	00	00	00	00	10	00	00	00	00	00	00

WORD e_lfarlc: 0040

WORD e

WORD e_oeminfo: 0000

WORD e_res2[10]: 0000 0000 0000 00 LONG e_lfanew: 0000 0080

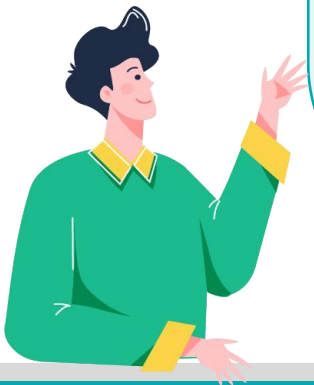
150	60	40	00	00	28	00	00	00	00	00	00	00	00	00	00	00
160	00	00	00	00	00	00	00	00	24	82	00	00	C0	01	00	00
170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
180	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00	00
190	C8	1A	00	00	00	10	00	00	00	1C	00	00	00	04	00	00
1A0	00	00	00	00	00	00	00	00	00	00	00	00	60	00	00	60
1B0	2E	64	61	74	61	00	00	00	B0	00	00	00	00	30	00	00
1C0	00	02	00	00	00	20	00	00	00	00	00	00	00	00	00	00
1D0	00	00	00	00	40	00	00	C0	2E	72	64	61	74	61	00	00
1E0	00	0B	00	00	00	40	00	00	00	0C	00	00	00	22	00	00
1F0	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00	40



NT Header

```
typedef struct _IMAGE_NT_HEADERS64 {  
    DWORD Signature;  
    IMAGE_FILE_HEADER FileHeader;  
    IMAGE_OPTIONAL_HEADER64 OptionalHeader;  
} IMAGE_NT_HEADERS64, *PIMAGE_NT_HEADERS64;
```

```
typedef struct _IMAGE_NT_HEADERS {  
    DWORD Signature;  
    IMAGE_FILE_HEADER FileHeader;  
    IMAGE_OPTIONAL_HEADER32 OptionalHeader;  
} IMAGE_NT_HEADERS32, *PIMAGE_NT_HEADERS32;
```





COFF Header

```
typedef struct _IMAGE_FILE_HEADER {  
    WORD    Machine;  
    WORD    NumberOfSections;  
    DWORD   TimeDateStamp;  
    DWORD   PointerToSymbolTable;  
    DWORD   NumberOfSymbols;  
    WORD    SizeOfOptionalHeader;  
    WORD    Characteristics;  
} IMAGE_FILE_HEADER, *PIMAGE_FILE_HEADER;
```





Stardust.EXE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00
10	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00
40	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68
50	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F
60	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20
70	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00
80	50	45	00	00	64	86	0B	00	67	C0	A3	63	00	00	00	00
90	00	00	00	00	F0	00	2E	02	0B	02	02	27	00	1C	00	00
A0	00	44	00	00	00	02	00	00	B0	14	00	00	00	10	00	00
B0	00	00	00	40	01	00	00	00	00	10	00	00	00	02	00	00
C0	04	00	00	00	00	00	00	00	05	00	02	00	00	00	00	00
D0	00	D0	00	00	00	04	00	00	00	00	00	00	00	00	00	00
E0	00	00	20	00	00	00	00	00	00	00	00	00	00	00	00	00
F0	00	00	10	00	00	00	00	00	00	00	00	00	00	00	00	00
100	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
110	00	80	00	00	DC	07	00	00	00	00	00	00	00	00	00	00
120	00	50	00	00	40	02	00	00	00	00	00	00	00	00	00	00
130	00	C0	00	00	80	00	00	00	00	00	00	00	00	00	00	00
140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
150	60	40	00	00	28	00	00	00	00	00	00	00	00	00	00	00
160	00	00	00	00	00	00	00	00	24	82	00	00	C0	01	00	00
170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
180	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00	00
190	C8	1A	00	00	00	10	00	00	00	1C	00	00	00	04	00	00
1A0	00	00	00	00	00	00	00	00	00	00	00	00	60	00	00	60
1B0	2E	64	61	74	61	00	00	00	B0	00	00	00	00	30	00	00
1C0	00	02	00	00	00	20	00	00	00	00	00	00	00	00	00	00
1D0	00	00	00	00	40	00	00	C0	2E	72	64	61	74	61	00	00
1E0	00	0B	00	00	00	40	00	00	00	0C	00	00	00	22	00	00
1F0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	40

DW DW WOR WORD Characteristics: 022E



Optional Header

```
typedef struct _IMAGE_OPTIONAL_HEADER64 {  
    WORD        Magic;  
    BYTE        MajorLinkerVersion;  
    BYTE        MinorLinkerVersion;  
    DWORD       SizeOfCode;  
    DWORD       SizeOfInitializedData;  
    DWORD       SizeOfUninitializedData;  
    DWORD       AddressOfEntryPoint;  
    DWORD       BaseOfCode;  
    ULONGLONG   ImageBase;  
    DWORD       SectionAlignment;  
    DWORD       FileAlignment;  
    WORD        MajorOperatingSystemVersion;  
    WORD        MinorOperatingSystemVersion;  
    WORD        MajorImageVersion;  
    WORD        MinorImageVersion;  
    WORD        MajorSubsystemVersion;  
    WORD        MinorSubsystemVersion;  
    DWORD       Win32VersionValue;  
    DWORD       SizeOfImage;  

```

```
    DWORD       SizeOfHeaders;  
    DWORD       CheckSum;  
    WORD        Subsystem;  
    WORD        DllCharacteristics;  
    ULONGLONG   SizeOfStackReserve;  
    ULONGLONG   SizeOfStackCommit;  
    ULONGLONG   SizeOfHeapReserve;  
    ULONGLONG   SizeOfHeapCommit;  
    DWORD       LoaderFlags;  
    DWORD       NumberOfRvaAndSizes;  
    IMAGE_DATA_DIRECTORY  
    DataDirectory[IMAGE_NUMBEROF_DIRECTORY_ENTRIES];  
} IMAGE_OPTIONAL_HEADER64;  
*PIMAGE_OPTIONAL_HEADER64;
```



PE

Stardust.EXE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00
10	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00
40	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68
50	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F
60	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20
70	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00
80	50	45	00	00	64	86	0B	00	67	C0	A3	63	00	00	00	00
90	00	00	00	00	F0	00	2E	02	0B	02	02	27	00	1C	00	00
A0	00	44	00	00	00	02	00	00	B0	14	00	00	00	10	00	00
B0	00	00	00	40	01	00	00	00	00	10	00	00	00	02	00	00
C0	04	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D0	00	D0	00	00	00	04	00	00	00	00	00	00	00	00	00	00
E0	00	00	20	00	00	00	00	00	00	00	00	00	00	00	00	00
F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
150	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
180	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00	00
190	C8	1A	00	00	00	10	00	00	00	1C	00	00	00	04	00	00
1A0	00	00	00	00	00	00	00	00	00	00	00	00	60	00	00	60
1B0	2E	64	61	74	61	00	00	00	B0	00	00	00	00	30	00	00
1C0	00	02	00	00	00	20	00	00	00	00	00	00	00	00	00	00
1D0	00	00	00	00	40	00	00	C0	2E	72	64	61	74	61	00	00
1E0	00	0B	00	00	00	40	00	00	00	0C	00	00	00	22	00	00
1F0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	40

DWORD BaseOfCode:0000 1000

DWORD Win32VersionValue: 0000 0000

DWORD NumberOfRvaAndSizes:0000 0010



Data Directories



```
typedef struct _IMAGE_DATA_DIRECTORY {  
    DWORD  VirtualAddress;  
    DWORD  Size;  
} IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
```



PE

Stardust.EXE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00
10	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00
40	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68
50	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F
60	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20
70	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00
80	50	45	00	00	64	86	0B	00	67	C0	A3	63	00	00	00	00
90	00	00	00	00	F0	00	2E	02	0B	02	02	27	00	1C	00	00
A0	00	44	00	00	00	02	00	00	B0	14	00	00	00	10	00	00
B0	00	00	00	40	01	00										
C0	04	00	00	00	00	00										
D0	00	D0	00	00	00	04										
E0	00	00	20	00	00	00										
F0	00	00	10	00	00	00										
100	00	00	00	00	10	00										
110	00	80	00	00	DC	07	00	00	00	B0	00	00	E8	04	00	00
120	00	50	00	00	40	02	00	00	00	00	00	00	00	00	00	00
130	00	C0	00	00	80	00	00	00	00	00	00	00	00	00	00	00
140	00	00	00	00	00											
150	60	40	00	00	28											
160	00	00	00	00	00											
170	00	00	00	00	00											
180	00	00	00	00	00											
190	C8	1A	00	00	00											
1A0	00	00	00	00	00	00										
1B0	2E	64	61	74	61	00	00									
1C0	00	02	00	00	00	20	00	00	00	00	00	00	00	00	00	00
1D0	00	00	00	00	40	00	00	C0	2E	72	64	61	74	61	00	00
1E0	00	0B	00	00	00	40	00	00	00	0C	00	00	00	22	00	00
1F0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	40

IMA IMAGE_
DWORD

DWORD Size:0000 0000

JG
)

IMAGE_DIRECTORY_ENTRY_EXPORT
DWORD Vi DWORD Size: 0000 0000
DWORD



PE

Stardust.EXE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00
10	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00
40	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68
50	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F
60	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20
70	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00
80	50	45	00	00	64	86	0B	00	67	C0	A3	63	00	00	00	00
90	00	00	00	00	F0	00	2E	02	0B	02	02	27	00	1C	00	00
A0	00	44	00	00	00	02	00	00	B0	14	00	00	00	10	00	00
B0	00	00	00	40	01	00	00	00	00	10	00	00	00	02	00	00
C0	04	00	00													
D0	00															
E0	00															
F0	00															
100	00															
110	00															
120	00															
130	00	C0	00	00	80	00	00	00	00	00	00					
140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
150	60	40	00	00	28	00	00	00	00	00	00	00	00	00	00	00
160	00	00	00	00	00	00	00	00	24	82	00	00	C0	01	00	00
170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
180	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00	00
190	C8	1A	00	00	00	10	00	00	00	1C	00	00	00	04	00	00
1A0	00	00	00	00	00	00	00	00	00	00	00	00	60	00	00	60
1B0	2E	64	61	74	61	00	00	00	B0	00	00	00	00	30	00	00
1C0	00	02	00	00	00	20	00	00	00	00	00	00	00	00	00	00
1D0	00	00	00	00	40	00	00	C0	2E	72	64	61	74	61	00	00
1E0	00	0B	00	00	00	40	00	00	00	0C	00	00	00	22	00	00
1F0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	40

IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG

IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR

DWORD Size:0000 0000

DWORD Size:0000 0000



Section Headers

```
typedef struct _IMAGE_SECTION_HEADER {  
    BYTE    Name[IMAGE_SIZEOF_SHORT_NAME];  
    union {  
        DWORD    PhysicalAddress;  
        DWORD    VirtualSize;  
    } Misc;  
    DWORD    VirtualAddress;  
    DWORD    SizeOfRawData;  
    DWORD    PointerToRawData;  
    DWORD    PointerToRelocations;  
    DWORD    PointerToLinenumbers;  
    WORD     NumberOfRelocations;  
    WORD     NumberOfLinenumbers;  
    DWORD    Characteristics;  
} IMAGE_SECTION_HEADER, *PIMAGE_SECTION_HEADER;
```





PE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00
10	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00
40	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68
50	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F
60	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20
70	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00
80	50	45	00	00	64	86	0B	00	67	C0	A3	63	00	00	00	00
90	00	00	00	00	F0	00	2E	02	0B	02	02	27	00	1C	00	00
A0	00	44	00	00	00	02	00	00	B0	14	00	00	00	10	00	00
B0	00	00	00	40	01	00	00	00	00	10	00	00	00	02	00	00
C0	04	00	00	00	00	00	00	00	05	00	02	00	00	00	00	00

IMAGE_SECTION_HEADER

BYTE Name[IMAGE_DWORD Characteristics: 6000 0060

Start	140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	150	60	40	00	00	28	00	00	00	00	00	00	00	00	00	00
	160	00	00	00	00	00	00	00	00	24	82	00	00	C0	01	00
	170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	180	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00
	190	C8	1A	00	00	00	10	00	00	00	1C	00	00	00	04	00
	1A0	00	00	00	00	00	00	00	00	00	00	00	00	60	00	60
	1B0	2E	64	61	74	61	00	00	00	B0	00	00	00	00	30	00
	1C0	00	02	00	00	00	20	00	00	00	00	00	00	00	00	00
	1D0	00	00	00	00	40	00	00	C0	2E	72	64	61	74	61	00
	1E0	00	0B	00	00	00	40	00	00	00	0C	00	00	00	22	00
	1F0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	40



PE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	2E	70	64	61	74	61	00	00	40	02	00	00	00	50	00	00
210	00	04	00	00	00	2E	00	00	00	00	00	00	00	00	00	00
220	00	00	00	00	40	00	00	40	2E	78	64	61	74	61	00	00
230	B8	01	00	00	00	60	00	00	00	02	00	00	00	32	00	00
240	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	40
250	2E	62	73	73	00	00	00	00	80	01	00	00	00	70	00	00
260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
270	00	00	00	00	80	00	00	C0	2E	69	64	61	74	61	00	00
280	DC	07	00	00	00	80	00	00	00	08	00	00	00	34	00	00
290	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0
2A0	2E	43	52	54	00	00	00	00	60	00	00	00	00	90	00	00
2B0	00	02	00	00	00	3C	00	00	00	00	00	00	00	00	00	00
2C0	00	00	00	00	40	00	00	C0	2E	74	6C	73	00	00	00	00
2D0	10	00	00	00	00	A0	00	00	00	02	00	00	00	3E	00	00
2E0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0

BYTE Name[IMA DW C DWORD Characteristics: C000 0040

340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00



PE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	2E	70	64	61	74	61	00	00	40	02	00	00	00	50	00	00
210	00	04	00	00	00	2E	00	00	00	00	00	00	00	00	00	00
220	00	00	00	00	40	00	00	40	2E	78	64	61	74	61	00	00
230	B8	01	00	00	00	60	00	00	00	02	00	00	00	32	00	00
240	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	40
250	2E	62	73	73	00	00	00	00	80	01	00	00	00	70	00	00
260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
270	00	00	00	00	80	00	00	C0	2E	69	64	61	74	61	00	00
280	DC	07	00	00	00	80	00	00	00	08	00	00	00	34	00	00
290	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0
2A0	2E	43	52	54	00	00	00	00	60	00	00	00	00	90	00	00
2B0	00	02	00	00	00	3C	00	00	00	00	00	00	00	00	00	00
2C0	00	00	00	00	40	00	00	C0	2E	74	6C	73	00	00	00	00
2D0	10	00	00	00	00	A0	00	00	00	02	00	00	00	3E	00	00
2E0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0

BYTE Name[IMAC

DW

DWORD Characteristics: C000 0040

350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00



PE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	2E	70	64	61	74	61	00	00	40	02	00	00	00	50	00	00
210	00	04	00	00	00	2E	00	00	00	00	00	00	00	00	00	00
220	00	00	00	00	40	00	00	40	2E	78	64	61	74	61	00	00
230	B8	01	00	00	00	60	00	00	00	02	00	00	00	32	00	00
240	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	40
250	2E	62	73	73	00	00	00	00	80	01	00	00	00	70	00	00
260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
270	00	00	00	00	80	00	00	C0	2E	69	64	61	74	61	00	00
280	DC	07	00	00	00	80	00	00	00	08	00	00	00	34	00	00
290	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0
2A0	2E	43	52	54	00	00	00	00	60	00	00	00	00	90	00	00
2B0	00	02	00	00	00	3C	00	00	00	00	00	00	00	00	00	00
2C0	00	00	00	00	40	00	00	C0	2E	74	6C	73	00	00	00	00
2D0	10	00	00	00	00	A0	00	00	00	02	00	00	00	3E	00	00
2E0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0
2F0	2E	72	73	72	63	00	00	00	E8	04	00	00	00	B0	00	00
300	00	06	00	00	00	40	00	00	00	00	00	00	00	00	00	00
310	00	00	00	00	40	00	00	C0	2E	72	65	6C	6F	63	00	00
320	80	00	00	00	00	C0	00	00	00	02	00	00	00	46	00	00

BYTE Name[IMAGE]

0000

DWORD PointerToRawData:0000 4600

390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	2E	70	64	61	74	61	00	00	40	02	00	00	00	50	00	00
210	00	04	00	00	00	2E	00	00	00	00	00	00	00	00	00	00
220	00	00	00	00	40	00	00	40	2E	78	64	61	74	61	00	00
230	B8	01	00	00	00	60	00	00	00	02	00	00	00	32	00	00
240	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	40
250	2E	62	73	73	00	00	00	00	80	01	00	00	00	70	00	00
260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
270	00	00	00	00	80	00	00	C0	2E	69	64	61	74	61	00	00
280	DC	07	00	00	00	80	00	00	00	08	00	00	00	34	00	00
290	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0
2A0	2E	43	52	54	00	00	00	00	60	00	00	00	00	90	00	00
2B0	00	02	00	00	00	3C	00	00	00	00	00	00	00	00	00	00
2C0	00	00	00	00	40	00	00	C0	2E	74	6C	73	00	00	00	00
2D0	10	00	00	00	00	A0	00	00	00	00	00	00	00	00	00	00
2E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2F0	2E	72	73	72	63	00	00	00	00	00	00	00	00	00	00	00
300	00	06	00	00	00	40	00	00	00	00	00	00	00	00	00	00
310	00	00	00	00	40	00	00	00	00	00	00	00	00	00	00	00
320	80	00	00	00	00	C0	00	00	00	02	00	00	00	46	00	00
330	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	42
340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

DWOI

DWORD Characteristics: 4200 0040



Data Directories



```
typedef struct _IMAGE_DATA_DIRECTORY {  
    DWORD   VirtualAddress;  
    DWORD   Size;  
} IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
```



Section Tables

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
▼ .text	400	1C00	1000	1AC8	60000060	0	0	0
>	2000	^	2AC8	^	r-x			
▼ .data	2000	200	3000	B0	C0000040	0	0	0
>	2200	^	30B0	^	rw-			
> .rdata	2200	C00	4000	B00	40000040	0	0	0
> .pdata	2E00	400	5000	240	40000040	0	0	0
> .xdata	3200	200	6000	1B8	40000040	0	0	0
> .bss	0	0	7000	180	C0000080	0	0	0
> .idata	3400	800	8000	7DC	C0000040	0	0	0
> .CRT	3C00	200	9000	60	C0000040	0	0	0
> .tls	3E00	200	A000	10	C0000040	0	0	0
> .rsrc	4000	600	B000	4E8	C0000040	0	0	0
> .reloc	4600	200	C000	80	42000040	0	0	0

IMAGE_DIRECTORY_ENTRY_IMPORT

DWORD VirtualAddress:0000 8000

DWORD Size:0000 07DC

$8000 - 8000(\text{VA of Section}) + 3400 (\text{FOA of Section}) = 3400$





Import Descriptor

```
typedef struct _IMAGE_IMPORT_DESCRIPTOR {  
    union {  
        DWORD Characteristics;  
        DWORD OriginalFirstThunk;  
    } DUMMYUNIONNAME;  
    DWORD TimeDateStamp;  
    DWORD ForwarderChain;  
    DWORD Name;  
    DWORD FirstThunk;  
} IMAGE_IMPORT_DESCRIPTOR;  
typedef IMAGE_IMPORT_DESCRIPTOR UNALIGNED *PIMAGE_IMPORT_DESCRIPTOR;
```



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3400	64	80	00	00	00	00	00	00	00	00	00	00	20	87	00	00
3410	24	82	00	00	FC	80	00	00	00	00	00	00	00	00	00	00
3420	98	87	00	00	BC	82	00	00	D4	81	00	00	00	00	00	00
3430	00	00	00	00	A8	87	00	00	94	83	00	00	E4	81	00	00
3440	00	00	00	00	00	00	00	00	D0	87	00	00	A4	83	00	00
3450	00	00	00													
3460	00	00	00													
3470	00	00	00													
3480	00	00	00	00	2C	84	00	00	00	00	00	00	44	84	00	00
3490	00	00	00	00	58	84	00	00	00	00	00	00	72	84	00	00
34A0	00	00	00	00	82	84	00	00	00	00	00	00	94	84	00	00
34B0	00	00	00	00	B0	84	00	00	00	00	00	00	C8	84	00	00
34C0	00	00	00	00	E6	84	00	00	00	00	00	00	EE	84	00	00
34D0	00	00	00	00	FC	84	00	00	00	00	00	00	0E	85	00	00

8064-8000(VA of Section) + 3400 (FOA of Section)=3464224



Section Tables

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
▼ .text	400	1C00	1000	1AC8	60000060	0	0	0
>	2000	^	2AC8	^	r-x			
▼ .data	2000	200	3000	B0	C0000040	0	0	0
>	2200	^	30B0	^	rw-			
> .rdata	2200	C00	4000	B00	40000040	0	0	0
> .pdata	2E00	400	5000	240	40000040	0	0	0
> .xdata	3200	200	6000	1B8	40000040	0	0	0
> .bss	0	0	7000	180	C0000080	0	0	0
> .idata	3400	800	8000	7DC	C0000040	0	0	0
> .CRT	3C00	200	9000	60	C0000040	0	0	0
> .tls	3E00	200	A000	10	C0000040	0	0	0
> .rsrc	4000	600	B000	4E8	C0000040	0	0	0
> .reloc	4600	200	C000	80	42000040	0	0	0

$8064 - 8000(\text{VA of Section}) + 3400 (\text{FOA of Section}) = 3464$

3460	00	00	00	00	E4	83	00	00	00	00	00	00	F4	83	00	00
3470	00	00	00	00	02	84	00	00	00	00	00	00	14	84	00	00
3480	00	00	00	00	2C	84	00	00	00	00	00	00	44	84	00	00
3490	00	00	00	00	58	84	00	00	00	00	00	00	72	84	00	00
34A0	00	00	00	00	82	84	00	00	00	00	00	00	94	84	00	00
34B0	00	00	00	00	B0	84	00	00	00	00	00	00	C8	84	00	00
34C0	00	00	00	00	E6	84	00	00	00	00	00	00	EE	84	00	00
34D0	00	00	00	00	FC	84	00	00	00	00	00	00	0E	85	00	00
34E0	00	00	00	00	1E	85	00	00	00	00	00	00	34	85	00	00
34F0	00	00	00	00	00											
3500	00	00	00	00	5C											
3510	00	00	00	00	76											
3520	00	00	00	00	9C											
3530	00	00	00	00	B4	85	00	00	00	00	00	00	BE	85	00	00
3540	00	00	00	00	CA	85	00	00	00	00	00	00	D4	85	00	00
3550	00	00	00	00	E0	85	00	00	00	00	00	00	EA	85	00	00
3560	00	00	00	00	F2	85	00	00	00	00	00	00	FC	85	00	00
3570	00	00	00	00	04	86	00	00	00	00	00	00	0E	86	00	00
3580	00	00	00	00	16	86	00	00	00	00	00	00	20	86	00	00
3590	00	00	00	00	2A	86	00	00	00	00	00	00	34	86	00	00
35A0	00	00	00	00	3E	86	00	00	00	00	00	00	48	86	00	00
35B0	00	00	00	00	52	86	00	00	00	00	00	00	5C	86	00	00

8064-8000(VA of Section) +

0000 0000 0000 8534



Import Descriptor

```
typedef struct _IMAGE_IMPORT_DESCRIPTOR {  
    union {  
        DWORD Characteristics;  
        DWORD OriginalFirstThunk;  
    } DUMMYUNIONNAME;  
    DWORD TimeDateStamp;  
    DWORD ForwarderChain;  
    DWORD Name;  
    DWORD FirstThunk;  
} IMAGE_IMPORT_DESCRIPTOR;  
typedef IMAGE_IMPORT_DESCRIPTOR UNALIGNED *PIMAGE_IMPORT_DESCRIPTOR;
```



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F							
3400	64	80	00	00	00	00	00	00	00	00	00	00	20	87	00	00							
3410	24	82	00	00	FC	80	00	00	00	00	00	00	00	00	00	00							
3420	98	87	00	00	BC	82	00	00	D4	81	00	00	00	00	00	00							
3430	00	00	00	00	A8	87	00	00	94	83	00	00	E4	81	00	00							
3440	00	00	00	00	00	00	00	00	D0	87	00	00	A4	83	00	00							
3450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
3460	00	00	00	00	E4	83	00	00	00	80FC-8000(VA of Section) + 3400													
3470	00																						
3480	00																						
3490	00																						
34A0	00	00	00	00	82	84	00	00	00	00	00	00	94	84	00	00							
34B0	00	00	00	00	B0	84	00	00	00	00	00	00	C8	84	00	00							
34C0	00	00	00	00	E6	84	00	00	00	00	00	00	EE	84	00	00							
34D0	00	00	00	00	FC	84	00	00	00	00	00	00	0E	85	00	00							

80FC-8000(VA of Section) + 3400 (FOA of Section)=34FC



Section Tables

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
▼ .text	400	1C00	1000	1AC8	60000060	0	0	0
>	2000	^	2AC8	^	r-x			
▼ .data	2000	200	3000	B0	C0000040	0	0	0
>	2200	^	30B0	^	rw-			
> .rdata	2200	C00	4000	B00	40000040	0	0	0
> .pdata	2E00	400	5000	240	40000040	0	0	0
> .xdata	3200	200	6000	1B8	40000040	0	0	0
> .bss	0	0	7000	180	C0000080	0	0	0
> .idata	3400	800	8000	7DC	C0000040	0	0	0
> .CRT	3C00	200	9000	60	C0000040	0	0	0
> .tls	3E00	200	A000	10	C0000040	0	0	0
> .rsrc	4000	600	B000	4E8	C0000040	0	0	0
> .reloc	4600	200	C000	80	42000040	0	0	0

$80FC - 8000(\text{VA of Section}) + 3400(\text{FOA of Section}) = 34FC$

34F0	00	00	00	00	00	00	00	00	00	00	00	00	4C	85	00	00
3500	00	00	00	00	5C	85	00	00	00	00	00	00	68	85	00	00
3510	00	00	00	00	76	85	00	00	00	00	00	00	88	85	00	00
3520	00	00	00	00	9C	85	00	00	00	00	00	00	A6	85	00	00
3530	00	00	00	00	B4	85	00	00	00	00	00	00	BE	85	00	00
3540	00	00	00	00	CA	85	00	00	00	00	00	00	D4	85	00	00
3550	00	00	00	00	E0	85	00	00	00	00	00	00	EA	85	00	00
3560	00	00	00													
3570	00	00	00		80EC-8000(VA of Section) + B4											
3580	00	00	00													
3590	00	00	00	00	2A	86	00	00	00	00	00	00	34	86	00	00
35A0	00	00	00	00	3E	86	00	00	00	00	00	00	48	86	00	00
35B0	00	00	00	00	52	86	00	00	00	00	00	00	5C	86	00	00
35C0	00	00	00	00	66	86	00	00	00	00	00	00	00	00	00	00
35D0	00	00	00	00	72	86	00	00	00	00	00	00	00	00	00	00
35E0	00	00	00	00	80	86	00	00	00	00	00	00	8E	86	00	00
35F0	00	00	00	00	9C	86	00	00	00	00	00	00	AA	86	00	00

0000 0000 0000 8666



Import Descriptor

```
typedef struct _IMAGE_IMPORT_DESCRIPTOR {  
    union {  
        DWORD Characteristics;  
        DWORD OriginalFirstThunk;  
    } DUMMYUNIONNAME;  
    DWORD TimeDateStamp;  
    DWORD ForwarderChain;  
    DWORD Name;  
    DWORD FirstThunk;  
} IMAGE_IMPORT_DESCRIPTOR;  
typedef IMAGE_IMPORT_DESCRIPTOR UNALIGNED *PIMAGE_IMPORT_DESCRIPTOR;
```



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3400	64	80	00	00	00	00	00	00	00	00	00	00	20	87	00	00
3410	24	82	00	00	FC	80	00	00	00	00	00	00	00	00	00	00
3420	98	87	00	00	BC	82	00	00	D4	81	00	00	00	00	00	00
3430	00	00	00	00	A8	87	00	00	94	83	00	00	E4	81	00	00
3440	00	00	00	00	00	00	00	00	D0	87	00	00	A4	83	00	00
3450	00	8000-8000(VA of [redacted]) 81D4-8000(VA of Section) + 3400 (FOA of Section)=35D4														
3460	00															
3470	00															
3480	00	00	00	00												
3490	00	00	00	00	58	84	00	00	00	00	00	00	72	84	00	00
34A0	00	00	00	00	82	84	00	00	00	00	00	00	94	84	00	00
34B0	00	00	00	00	B0	84	00	00	00	00	00	00	C8	84	00	00
34C0	00	00	00	00	E6	84	00	00	00	00	00	00	EE	84	00	00
34D0	00	00	00	00	FC	84	00	00	00	00	00	00	0E	85	00	00



Section Tables

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
✓ .text	400	1C00	1000	1AC8	60000060	0	0	0
>	2000	^	2AC8	^	r-x			
✓ .data	2000	200	3000	B0	C0000040	0	0	0
>	2200	^	30B0	^	rw-			
> .rdata	2200	C00	4000	B00	40000040	0	0	0
> .pdata	2E00	400	5000	240	40000040	0	0	0
> .xdata	3200	200	6000	1B8	40000040	0	0	0
> .bss	0	0	7000	180	C0000080	0	0	0
> .idata	3400	800	8000	7DC	C0000040	0	0	0
> .CRT	3C00	200	9000	60	C0000040	0	0	0
> .tls	3E00	200	A000	10	C0000040	0	0	0
> .rsrc	4000	600	B000	4E8	C0000040	0	0	0
> .reloc	4600	200	C000	80	42000040	0	0	0

$81D4 - 8000(\text{VA of Section}) + 3400(\text{FOA of Section}) = 35D4$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
35D0	00	00	00	00	72	86	00	00	00	00	00	00	00	00	00	00
35E0	00	00	00	00	80	86	00	00	00	00	00	00	8E	86	00	00
35F0	00	00	00	00	9C	86	00	00	00	00	00	00	AA	86	00	00
3600	00	00	00	00	B8	86	00	00	00	00	00	00	C2	86	00	00
3610	00	00	00	00	CA	86	00	00	00	00	00	00	00	00	00	00
3620	00	00	00	00	E4	83	00	00	00	00	00	00	F4	83	00	00
3630	00	00														
3640	00	00														
3650	00	00														
3660	00	00	00	00	82	84	00	00	00	00	00	00	94	84	00	00
3670	00	00	00	00	B0	84	00	00	00	00	00	00	C8	84	00	00
3680	00	00	00	00	E6	84	00	00	00	00	00	00	EE	84	00	00
3690	00	00	00	00	FC	84	00	00	00	00	00	00	0E	85	00	00
36A0	00	00	00	00	1E	85	00	00	00	00	00	00	34	85	00	00
36B0	00	00	00	00	00	00	00	00	00	00	00	00	4C	85	00	00
36C0	00	00	00	00	5C	85	00	00	00	00	00	00	68	85	00	00

81D4-8000(VA of Section) + 3400 0000 0000 0000 8672

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3400	64	80	00	00	00	00	00	00	00	00	00	00	20	87	00	00
3410	24	82	00	00	FC	80	00	00	00	00	00	00	00	00	00	00
3420	98	87	00	00	BC	82	00	00	D4	81	00	00	00	00	00	00
3430	00	00	00	00	A8	87	00	00	94	83	00	00	E4	81	00	00
3440	00	00	00	00	00	00	00	00	D0	87	00	00	A4	83	00	00
3450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3460	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3470	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3480	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3490	00	00	00	00	58	84	00	00	00	00	00	00	72	84	00	00
34A0	00	00	00	00	82	84	00	00	00	00	00	00	94	84	00	00
34B0	00	00	00	00	B0	84	00	00	00	00	00	00	C8	84	00	00
34C0	00	00	00	00	E6	84	00	00	00	00	00	00	EE	84	00	00
34D0	00	00	00	00	FC	84	00	00	00	00	00	00	0E	85	00	00

8000-8000(VA of Section) + 3400 (FOA of Section) = 35E4

81E4-8000(VA of Section) + 3400 (FOA of Section) = 35E4

A4



Section Tables

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
▼ .text	400	1C00	1000	1AC8	60000060	0	0	0
>	2000	^	2AC8	^	r-x			
▼ .data	2000	200	3000	B0	C0000040	0	0	0
>	2200	^	30B0	^	rw-			
> .rdata	2200	C00	4000	B00	40000040	0	0	0
> .pdata	2E00	400	5000	240	40000040	0	0	0
> .xdata	3200	200	6000	1B8	40000040	0	0	0
> .bss	0	0	7000	180	C0000080	0	0	0
> .idata	3400	800	8000	7DC	C0000040	0	0	0
> .CRT	3C00	200	9000	60	C0000040	0	0	0
> .tls	3E00	200	A000	10	C0000040	0	0	0
> .rsrc	4000	600	B000	4E8	C0000040	0	0	0
> .reloc	4600	200	C000	80	42000040	0	0	0

$81E4 - 8000 (\text{VA of Section}) + 3400 (\text{FOA of Section}) = 35E4$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
35E0	00	00	00	00	80	86	00	00	00	00	00	00	8E	86	00	00
35F0	00	00	00	00	9C	86	00	00	00	00	00	00	AA	86	00	00
3600	00	00	00	00	B8	86	00	00	00	00	00	00	C2	86	00	00
3610	00	00	00	00	CA	86	00	00	00	00	00	00	00	00	00	00
3620	00	00	00	00	E4	83	00	00	00	00	00	00	F4	83	00	00
3630	00	00	00	00	02	84	00	00	00	00	00	00	14	84	00	00
3640	00	00	00	00	2C	84	00	00	00	00	00	00	44	84	00	00
3650	00	00	00	00	58	84	00	00	00	00	00	00	72	84	00	00
3660	00	00														
3670	00	00														
3680	00	00														
3690	00	00														
36A0	00	00	00	00	1E	85	00	00	00	00	00	00	34	85	00	00
36B0	00	00	00	00	00	00	00	00	00	00	00	00	4C	85	00	00
36C0	00	00	00	00	5C	85	00	00	00	00	00	00	68	85	00	00
36D0	00	00	00	00	76	85	00	00	00	00	00	00	88	85	00	00
36E0	00	00	00	00	9C	85	00	00	00	00	00	00	A6	85	00	00
36F0	00	00	00	00	B4	85	00	00	00	00	00	00	BE	85	00	00
3700	00	00	00	00	CA	85	00	00	00	00	00	00	D4	85	00	00
3710	00	00	00	00	E0	85	00	00	00	00	00	00	EA	85	00	00
3720	00	00	00	00	F2	85	00	00	00	00	00	00	FC	85	00	00

81E4-8000(VA of Section) + 3400

0000 0000 0000 86CA

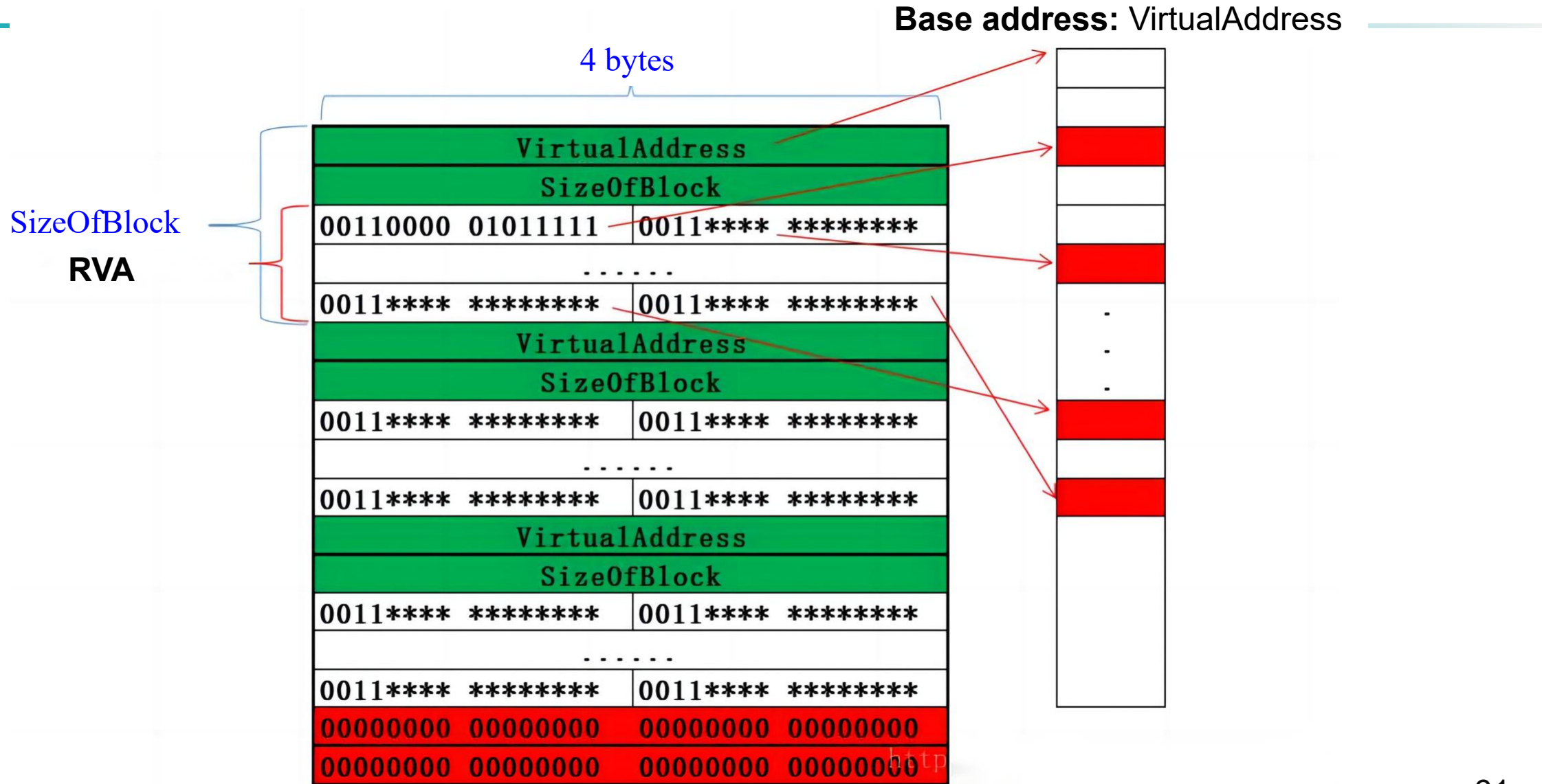


Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
▼ .text	400	1C00	1000	1AC8	60000060	0	0	0
>	2000	^	2AC8	^	r-x			
▼ .data	2000	200	3000	B0	C0000040	0	0	0
>	2200	^	30B0	^	rw-			
> .rdata	2200	C00	4000	B00	40000040	0	0	0
> .pdata	2E00	400	5000	240	40000040	0	0	0
> .xdata	3200	200	6000	1B8	40000040	0	0	0
> .bss	0	0	7000	180	C0000080	0	0	0
> .idata	3400	800	8000	7DC	C0000040	0	0	0
> .CRT	3C00	200	9000	60	C0000040	0	0	0
> .tls	3E00	200	A000	10	C0000040	0	0	0
> .rsrc	4000	600	B000	4E8	C0000040	0	0	0
> .reloc	4600	200	C000	80	42000040	0	0	0

```
Typedef struct _IMAGE_BASE_RELOCATION{  
    DWORD VirtualAddress  
    DWORD SizeOfBlock  
}IMAGE_BASE_RELOCATION,*PIMAGE_BASE_RELOCATIO  
N;
```

```
IMAGE_DIRECTORY_ENTRY_BASERELOC  
DWORD VirtualAddress:0000 C000  
DWORD Size:0000 0080
```

$C000 - C000(\text{VA of section}) + 4600(\text{FOA of section}) = 4600$



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4600	00	20	00	00	0C	00	00	00	A8	AA	00	00	00	30	00	00
4610	18	00	00	00	10	A0	60	A0	70	A0	80	A0	90	A0	98	A0
4620	A0	A0	00	00	00	40	00	00	4C	00	00	00	40	A0	60	A0
4630	68	A0	70	A0	78	A0	70	A3	80	A3	90	A3	A0	A3	B0	A3
4640	C0	A3	D0	A3	E0	A3	F0	A3	00	A4	10	A4	20	A4	30	A4
4650	40	A4	50	A4	60	A4	70	A4	80	A4	90	A4	A0	A4	B0	A4
4660	C0	A4	C000 C000 VA of 1 DWORD				C000 C000 VA of 1 DWORD				C000 C000 VA of 1 DWORD				A040 000 0010	
4670	00	90														
4680	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4690	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
46A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
46B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
46C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
46D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
46E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

THE END

Fangtian Zhong

CSCI 591

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

09/18/2025