Theory

Virtual Addressing

VA	Virtual Address
RVA	Relative Virtual Adress VA2-VA1
Offset	Difference Virtual - Physcial Address (?)

PE Files (32 bit)

```
IMAGE_DOS_HEADER
+00         WORD e_magic Magic Number MZ ($5A4D)     IMAGE_DOS_SIGNATURE
+02         WORD e_cblp Bytes on last page of file
+04         WORD e_cp Pages in file
+06         WORD     e_crlc     Relocations
+08         WORD     e_cparhdr     Size of header in paragraphs
+0A  (10)   WORD     e_minalloc     Minimum extra paragraphs needed
+0C  (12)   WORD     e_maxalloc     Maximum extra paragraphs needed
+0E  (14)   WORD     e_ss     Initial (relative) SS value
+10  (16)   WORD     e_sp     Initial SP value
+12  (18)   WORD     e_csum     Checksum
+14  (20)   WORD     e_ip     Initial IP value
+16  (22)   WORD     e_cs     Initial (relative) CS value
+18  (24)   WORD     e_lfarlc     File address of relocation table
+1A  (26)   WORD     e_ovno     Overlay number
+1C  (28)   Array[4] of WORD     e_res     Reserved words
+24  (36)   WORD     e_oemid     OEM identifier (for e_oeminfo)
+26  (28)   WORD     e_oeminfo     OEM information; e_oemid specific
+28  (40)   Array[10] of WORD    e_res2     Reserved words
+3C  (60)   DWORD    e_lfanew     File address of new exe header
IMAGE_NT_HEADERS
```

```
typedef struct _IMAGE_NT_HEADERS {
  DWORD                 Signature;
  IMAGE_FILE_HEADER     FileHeader;
  IMAGE_OPTIONAL_HEADER OptionalHEader;  //Not Optional :-)
} IMAGE_NT_HEADERS, *PIAMGE_NT_HEADERS;
```

```
typedef struct _IMAGE_FILE_HEADER {
  WORD  Machine;
  WORD  NumberOfSections;
  DWORD TimeDateStamp;
  DWORD PointerToSymbolTable;
  DWORD NumberOfSymbols;
  WORD  SizeOfOptionalHeader;  //E0h
  WORd  Characteristics;
} IMAGE_FILE_HEADER, *PINMAGE_FILE_HEADER;
```

PE File Header (0x10)

Show

PE Optional Header (0x18)

Show

```
Magic -> 32 or 64
AddressOfEntryPoints -> RVA of Entry Point (EP) ~ location of first
instruction
BaseOfCode, BaseOfData -> Code and Data Sections
ImageBase -> Preferred VA for PE file in memory (default: 0x00400000 for
.exe, 0x10000000 for DLLs)
SectionAlignment, FileAlignment -> Alignment in memory
SizeOfImage -> MemorySize of PE file at runtime, must be multiple of
SectionAlignment
```

DataDirectory Array:

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

* 16 Data Directory Structures per default htat point to RVA and size of
specific data inside PE image on runtime.
* Example: ExportTableAddress (exported functions), ImportTableAddress
(imported functions), ResourceTable (embedded resources), ImportAddressTable
(IAT, runtime addresses of imported functions)
```