可执行文件即将被执行时会经历：磁盘文件 -> 创建进程 -> 加载到虚拟内存 -> （动态链接） -> 执行。

开始时文件储存在磁盘中，包括各种数据信息（代码段，数据段，文件信息等）。文件在Hex软件中打开查看到的地址就是此刻在磁盘存储时的线性偏移地址，从0000 0000开始，一个数据距离文件开头多少字节地址就显示为多少。

创建进程会为程序在硬件上开辟一个独立的空间供其使用，不同进程互不干扰，也就解释了为何不同程序可以使用相同的ImageBase且互不冲突。可执行文件加载到虚拟地址的位置和在磁盘中的地址实际上并无大的关系，只是需要根据在磁盘中的偏移地址来定位关键信息，从而根据文件信息来分配虚拟地址。

虚拟地址也就是打开普通调试器看到的地址。而实际地址通常来讲只有操作系统自己知道[ASLR（地址空间布局随机化）启用，为了安全]，虚拟地址的分配并非从0000 0000开始，而是随机设定一个符合地址格式的值，因此即使程序按照头部信息分配，多次执行时看到的虚拟地址也不同。

PE文件地址层级关系整理大概是这样：

**磁盘文件偏移**

PointerToRawData（节区头）：节区在磁盘文件中的起始偏移（文件指针）

SizeOfRawData（节区头）：节区在磁盘中的实际大小（可能因对齐填充与实际代码大小不同）

FileAlignment（可选头）：磁盘中节区数据的对齐粒度（通常512字节或4KB）

**内存虚拟地址**

ImageBase（可选头）：模块的预设加载基址（可能被ASLR随机化）

SectionAlignment（可选头）：内存中节区的对齐粒度（通常4KB）

RVA字段（如AddressOfEntryPoint、BaseOfCode等）：所有以RVA形式存储的地址都是内存层级的偏移量

VirtualAddress（节区头）：节区在内存中的起始RVA（需加上ImageBase得到VA）

VirtualSize（节区头）：节区在内存中的实际大小（可能大于磁盘大小，如未初始化数据）