1. 从程序员写程序，到用户拿鼠标双击 exe 文件打开这个程序，这期间都发生了什么？你能运用本堂课知识，详细描述一下吗?

   ①首先, 程序员根据需求和设计规范, 使用编程语言（如 C++、Java、Python 等）编写代码。一旦代码编写完成，程序员会使用相应的编译器将源代码转换成可执行文件。在这个过程中，源代码被转换成机器可以理解的二进制代码。这些可执行文件通常有不同的格式，例如 Windows 上的.exe 文件、Linux 上的可执行文件等。如果程序使用了其他模块或库，则需要在链接阶段将这些模块或库与可执行文件链接在一起。链接器会将所有需要的代码和资源合并到最终的可执行文件中。在发布之前，程序员会进行各种测试，以确保程序的稳定性、性能和安全性。这包括单元测试、集成测试、系统测试等。

   ②一旦软件通过了测试，程序员通常会将可执行文件与其他必要的文件（如配置文件、资源文件等）打包成一个可供用户安装的软件包。软件包需要部署到用户可以访问的地方。这可以通过多种方式实现，例如：将软件包复制到用户的计算机上, 发布到网站或应用商店, 打包成可安装程序。用户可以通过这些渠道获取软件，有时还会配有安装向导等。用户下载或获取软件后，会执行安装过程。这通常涉及解压文件、复制到适当的目录、注册必要的系统组件等操作, 安装程序可能会在桌面、开始菜单或其他位置创建快捷方式, 方便用户启动程序。

   ③用户在图标上双击 App.exe 的操作被 Explorer.exe 也就是桌面程序获知，在桌面程序体内有一堆等着响应的 DLL，它们被称为 Shell 程序。其中有一个叫 Shell32.dll 的程序, 它表示认领双击工作, 首先它获取被我们点击的文件名字和位置，然后开始通过 ShellExecuteExW 来执行我们的程序, 最终调用了 CreateProcessW。具体流程如下：

```
1  ......
2  _InvokeContextMenu()
3  CDefFolderMenu::InvokeCommand()
4  HDXA_LetHandlerProcessCommandEx()
5  CShellExecMenu::InvokeCommand
6  CShellExecMenu::_InvokeOne()
7  _InvokePid()
8  ShellExecuteExW()
9  _ShellExecuteNormal()
10 ExecuteNormal()
11 _TryInvokeApplication()
12 _DoExecCommand()
13 _SHCreateProcess()
14 CreateProcessW()
```

当 CreateProcess 这个函数被调用，系统就会创建一个“进程内核对象”。进程内核对象可以看作一个操作系统用来管理进程的内核对象，它也是系统用来存放关于进程统计信息的地方。系统为该进程创建 4GB 的虚拟地址空间用来加载 App.exe 和其他必要的 dll 文件。loader 加载器加载 App.exe 及其必要的 DLL 文件数据和代码，把这些代码和数据从硬盘加载到内存中，分析文件头以识别文件的运行环境，根据文件头决定由那个环境进行加载操作；进入内存之后，CPU

按照冯诺依曼的思想，以程序计数器 pc 的值作为地址不断地去内存进行寻址操作，在内存里边里边取出指令，然后对 pc 进行 +1 操作，再对指令进行分析，根据操作码决定进行何种操作，根据操作数去获得相应的数据，再把数据加载到 CPU 的寄存器之中，之后在处理器里边对数据进行相应的操作并将得到的结果进行写回操作，接着再从程序计数器 pc 中得到下一条指令的地址，不断循环直到这个程序所有的指令都执行完为止。

2. 操作系统是程序吗？操作系统管理如何将程序载入内存，那 OS 自己又是如何被放入计算机内存中的呢

①操作系统是程序，但它与普通应用程序不同。操作系统是运行在计算机上的第一个软件，它负责管理计算机的硬件和软件资源，并为其他应用程序提供运行环境。

②操作系统的加载器 loader 会从磁盘或其他存储介质中将程序文件加载到内存中，将程序文件中的代码和数据复制到内存中，并进行必要的重定位。操作系统同时也会为程序分配内存空间，在分配的过程中采用了虚拟内存的概念，让程序以为自己几乎拥有一整个连续的内存单元，永远也用不完，不用担心内存大小，然而实际情况是操作系统采用分段，分页和重定位等技术将无限的虚拟内存映射到有限的物理内存之中，对程序屏蔽掉了实际的物理空间，同时也提高了内存的利用率。如果程序依赖于其他动态链接库（DLL）或共享库，操作系统也会加载这些库到内存中，并将它们的地址映射到程序的地址空间中，以便程序能够调用它们的功能。操作系统会初始化程序的代码和数据。这包括设置程序的寄存器、堆栈和其他数据结构。

③在计算机启动时，BIOS（基本输入/输出系统）会首先运行，会尝试从预设的启动设备（通常是硬盘、固态硬盘、光盘或网络）的引导扇区加载 bootloader 引导加载程序，引导加载程序负责加载操作系统的核心部分内核到内存中。引导程序会将控制权转移给操作系统内核，内核被加载到内存中，它开始执行，内核首先会进行初始化，包

括初始化设备驱动程序、建立内存管理、建立进程管理等操作系统核心功能。操作系统就成功地被放入计算机内存之中