硬件抽象层(HardwareAbstraction Layer) (HAL) (hal.dll)  
最底层。直接和硬件打交道，是硬件和软件的桥梁，帮助操作系统和鼠标，打印机，显示器等硬件交流。

配图：（打印机通常需要额外安装驱动）

内核(Kernel)  
实现操作系统的一些底层服务。比如线程调度，多处理器的同步，中断/异常处理等。我们的电脑通常只有四个以内CPU核心，但是我们却可以同时运行Dev-C++，Edge浏览器，QQ，Microsoft Word，网易云音乐，搜狗拼音输入法等等非常多的软件，靠的是Kernel层调度。（可以配个图：开一堆的软件，然后开一下任务管理器，截个图）

Kernel将CPU资源分配给不同的软件轮流使用。但是因为切换地非常快，我们感觉似乎所有软件在同时运行。

对称地，运行单单一个软件，也未必一直在同一个CPU内核运行，这些都是Kernel层的工作。

所以，当一个软件卡住的时候，整个系统并不会卡住（配图：一个未响应的截图）

运行体(Executive)（ntoskrnl.exe）  
实现主要的操作系统服务，比方主要的线程进程管理，内存管理， IO及进程间通讯等。

运行在Kernel层之上，具体地负责各个对各个软件地调度，并且负责软件之间的通信。比如，我们在C语言里面include了stdlib.h 之后可以通过malloc函数像操作系统申请内存；比如：很多软件占用了内存之后就不主动还给操作系统了，操作系统会在这个软件被关闭的时候自动回收。比如当你先登录了QQ，再安装打开QQ音乐的时候，就可以自动登录了，也是操作系统帮助软件进行进程间通讯（等等！是不是会自动登录我不知道，就算会，QQ音乐是不是靠这个实现的我也不知道。不过我知道的是“进程间通讯”确实可以被用来做“一个软件登录了，另外一个软件自动登录”这件事情）

窗体图形子系统(Windows Graphics Subsystem)  
由win32K.sys在内核层实现,用户界面相关都依赖该层,User32.dll的大部分功能都由该层实现。

来几个漂亮的窗口的图片，还有不同软件里面比较统一的风格（比如右键菜单）

//以上参考资料：<https://www.cnblogs.com/mengfanrong/p/5178588.html>

然后还有特殊的Windows Subsystem for Linux

（截图：应用商店搜索Linux，里面截图。话说有不少有意思的评论也可以截图一下。）

Unix是历史比Windows还悠久的系统。当时很火。

Unix操作系统的内核可以给其他人写的程序提供很多现成的接口（正如printf差不多的意思），这些接口被称为POSIX。

Windows刚开始的时候不仅仅发明了自己的接口：WIN32，而且也重新实现了POSIX接口。

后来，Windows发达了，就把这个给去掉了

最近，被誉为Unix-Like的Linux越来越火，Windows10中，修改了一下POSIX接口，做成了Windows Subsystem for Linux，然后，我们就可以在Windows上运行Linux系统了啦。

（为嘛我不截图？因为这几天我在疯狂试用各种版本的Linux。。。没有用WIndows系统）