操作系统HW2

1. 依然有指导意义。其提出的系列计算机思想使得所有计算机拥有相同的体系结构和指令集，方便不同计算机之间互相连接、交换数据，这种思想也促进了计算机硬件、软件开发的标准化，控制了成本，促进技术发展。
2. 重要，特别是在优化性能和软件调试的时候。因为系统调用是操作系统提供的接口，有时调用库函数是实际上是使用系统调用。由于系统调用往往会在用户态和内核态之间切换，故了解库函数实际使用的系统调用，可能可以避免多余的调用进而优化性能；同理，程序调用库函数出现bug时，可能是因为系统调用没有正确的执行，了解系统调用可以帮助定位问题并修复。
3. 不必须有相同的名字，一般而言库函数名字更重要。因为系统调用通常被封装成库函数，而编写程序往往使用库函数实现功能。库函数的名字与其功能相关，与系统调用无关，提供一些高层次的、易于使用的接口，知道其名字有利于程序员了解函数功能并使用。
4.
    1. 顺序执行时，A周转时间70s；B周转时间70+80=150s；cpu利用率70/150=46.67%；设备利用率80/150=53.33%
    2. 并发执行时，两个程序同时加载到内存。0-20s，A使用cpu计算，B使用设备；20-40s，B计算，A使用10s后等待；40-50s，A计算，B使用设备；50-60s，B计算，A使用设备；60-70s，A计算，B使用设备；70-90s，A计算10s后等待，B使用设备。综上所述，A周转时间80s；B周转时间90s；cpu利用率70/90=77.78%；设备利用率80/90=88.89%
5. linux采用了`fork()` `exec()`两个函数。`fork()`可以创建一个与当前进程几乎一样的拷贝，但与当前进程完全独立，但可能导致安全问题；而`exec()`则可以动态地改变进程执行的程序，更加灵活。两个函数兼顾了独立性与灵活性。但是两个函数需要额外的处理以实现进程间的通信和同步，并且可能因此导致额外的性能开销。而Windows的`CreateProcess()`则简单易用，无需额外操作即可创建进程并执行程序，也无需额外进行进程间的通信同步，开销更小，但也因此缺少了独立性和灵活性。