1. 向用户程序提供抽象，如文件；管理计算机资源。

3.多道程序系统主要是为了解决因 I/O 操作造成的 CPU 算力的浪费，虽然它在宏观上可以同时运行多个程序，但无法保证每个程序都能被及时处理。分时系统会将CPU分配给若干个需要计算的作业轮流使用，因此可为多个终端提供快速的交互式服务。

4.经验证据表明，存储器访问表现出引⽤局部原则，即如果读取某⼀个位置，则接下来访问附近位置的概率⾮常⾼，尤其是紧随其后的内存位置。 因此，通过缓存整个缓存⾏，接下来缓存命中的概率会增加。 此外，现代的硬件可以将32或64字节块整个传输到⾼速缓存⾏，因此一次读取64字节要远快于读取单个字节 64 次。

1. 便于操作系统对 I/O 设备进行抽象。
2. 资源利用和及时性会互相矛盾，例如分时系统，为了及时性会将CPU的使用权在多个程序间不断切换，切换操作带来了不必要的开销。
3. 在内核态运行时，CPU可以执行指令集中的所有指令。在用户态运行时，只能执行指令集的部分命令。存在两种模式，可以将部分权限和操作都约束在操作系统内部实现，便于实现资源的管理和向用户提供抽象。
4. TRAP 指令可以将 CPU 从用户态切换至内核态。用户程序可以借助该指令进行系统调用。
5. 在分时系统中，允许多个进程轮流使用 CPU，因此需要进程表来保存各个进程的状态。Unix 或 Windows 也同时会有多个进程轮流使用 CPU，因此也需要进程表。

21.多路复用：在时间或空间维度上，多个进程共享资源。

时间多路复用：CPU(分时系统)，网卡，键盘(快捷键)，显示器

空间多路复用：内存，磁盘，显示器

26.除了文档意义上的名称外，系统调用实际上没有名称。

当库过程向内核读取trap时，它将系统调用的编号放入寄存器或堆栈中。这个数字用于索引到表中。在任何地方都没有使用名字。

另一方面，标准库过程的名称非常重要，因为它是程序中出现的内容。

27.提供了运行比机器物理内存大的程序的能力。

使得程序可以在运行时动态地链接库，而不是必须在编译时链接。

31.机制可以理解为处理数据的固定流程。策略可以理解为构造数据的方法。

通过机制和策略的分离，使得内核在决定处理流程的前提下足够精简，同时将策略交给用户指定，也保证了一定的自由度。

32.虚拟化层增加了内存使用和处理器开销，也增加了性能开销。