

操作系统作业一

范翔宇 PB18000006

1. 从系统的角度理解，操作系统主要负责哪两大功能？

①操作系统是个资源分配器，面对许多甚至冲突的资源请求，操作系统考虑如何为各个程序和用户分配资源，以便计算机系统能有效且公平地运行。

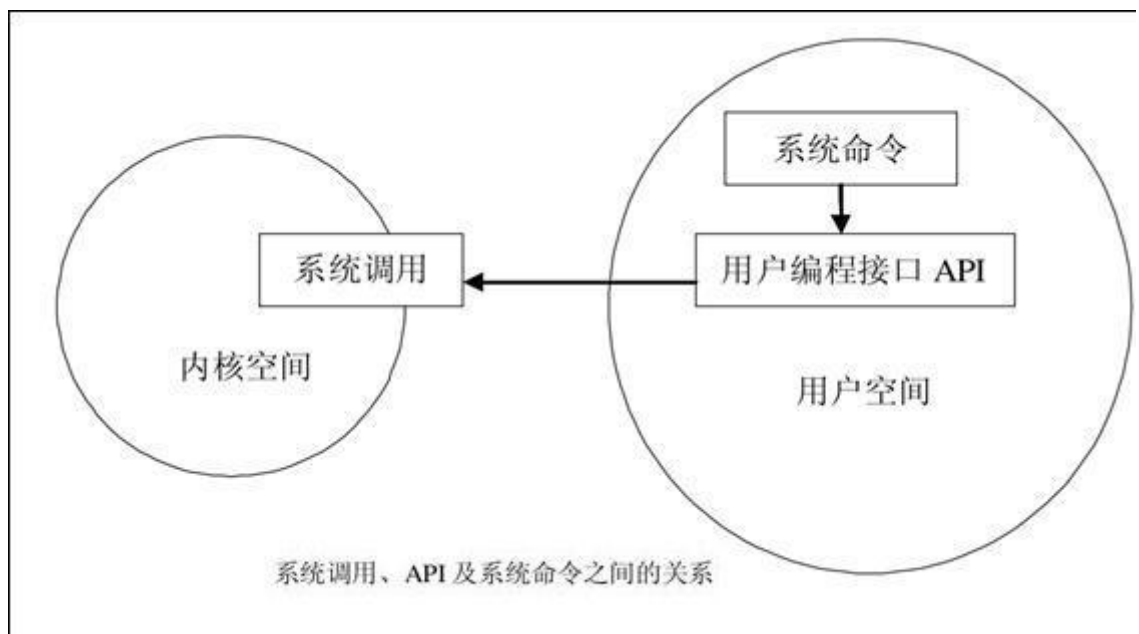
②操作系统是个控制程序，控制各种 I/O 设备，管理用户程序的执行，确保计算机资源的正常使用。

2. 什么是系统调用？阐述系统调用与 API 的区别和逻辑关系。

系统调用是操作系统提供的服务编程接口。

区别：API 是函数的定义，规定了这个函数的功能，跟内核无直接关系。而系统调用是通过中断向内核发请求，实现内核提供的某些服务。

逻辑关系：程序员调用的是 API（API 函数），然后通过系统调用共同完成函数的功能。因此，API 是一个提供给应用程序的接口，一组函数，是与程序员进行直接交互的。系统调用则不与程序员进行交互的，它是根据 API 函数，通过一个软中断机制向内核提交请求，以获取内核服务的接口。可通过该图来理解：



3. 阐述 Dual Mode 的工作机制，以及采用 Dual Mode 的原因。

工作机制：计算机硬件可以通过一个模式位来表示当前模式：内核模式（0）和用户模式（1）。系统可以通过模式位来区分为系统执行的任务和用户执行的任务。某些指令被指定为特权指令，只能在内核模式下执行。当计算机系统执行用户应用时，系统处于用户模式。然而，当用户应用通过系统调用，请求操作系统服务时，系统必须从用户模式切到内核模式，以满足请求。当系统引导时，硬件从内核模式开始。操作系统接着加载，然后开始在用户模式下执行用户程序。一旦有陷阱或中断，硬件会从用户模式切到内核模式。

原因：双重模式能够区分操作系统代码和用户代码，允许操作系统保护自身和其他系统组件，确保操作系统的正确运行。

4. 分析 Monolithic 结构，层次化结构，模块化结构和微内核结构的优劣。

Monolithic 结构：性能比较好，开销比较小，但难以实施和维护；

层次化结构：简化了构造和调试，但难在合理定义各层且与其他结构相比效率相对稍差；

模块化结构：任何模块都可以调用任何其他模型，更灵活；无需调用消息传递，更高效，但设计时对各模块间的接口规定很难满足在模块完成后对接口的实际需求而且各模块的设计齐头并进，无法寻找到一个可靠的决定顺序；

微内核结构：便于扩展操作系统，并且提供了更好的安全性和可靠性，但由于增加的系统功能的开销，微内核的性能会受损。

5. 举例说明采用机制与策略分离的设计原因。

在最坏情况下，每次策略的改变都可能需要改变底层机制，那么我们希望策略改变不敏感的通用机制。例如现有一种机制，可赋予某些类型的程序相对更高的优先级，如果机制能与策略分离开，那么它可用于支持I/O密集型程序应比CPU密集型程序具有更高优先级的策略，或者支持相反策略。