

# 操作系统作业 1

PB19051183 吴承泽

## 1. 请分别从系统和用户的角度，阐述操作系统的主要功能？

系统：

① 操作系统可以看作资源分配器。为了解决问题，计算机系统可能具有许多资源：CPU 时间、内存空间、文件存储空间、I/O 设备等。**操作系统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 操作系统同时也是控制程序。**控制程序管理用户程序的执行，以防止计算机资源的错误或不当的使用，它注重 I/O 设备的运行和控制。**

用户：

① 对于大多数计算机用户，**这类操作系统让单个用户单独使用资源，目的是优化用户进行工作或游戏。**这类系统设计的主要目的是用户使用方便，次要为性能。

② 在一个用户坐在大型机或小型机相连的终端前，其他用户通过其他终端访问同一计算机的情况下。这种操作系统的设计目标是优化资源利用率：**确保所有的 CPU 时间、内存、I/O 能得到有效使用，并确保没有用户使用超过限额以外的资源。**

③ 在用户坐在工作站前，这类工作站与其它工作站和服务器相连的情况下。这类用户不但可以使用专用资源，而且可以使用网络和服务器共享资源，包括文件、计算和打印服务器等。这类操作系统的设计需要兼顾使用方便性和资源利用率。

## 2. 解释什么是系统调用？阐述系统调用与 API 的逻辑关系。

什么是系统调用：系统调用提供操作系统服务接口，是在底层且由内核提供的一系列函数，操作权限高，操作系统为用户态运行的进程和硬件设备进行交互提

供的一组接口。

系统调用与 API 的逻辑关系：一般系统调用是和操作系统相关，并没有较好的可移植性，而根据 API 设计程序，是希望在任何支持同一 API 的系统上编译执行。

当在写程序的时候，无需知道具体系统调用，可以通过 API 来控制系统调用的执行，最终做出在硬件上的操作。API 与内核没有必然的联系，系统调用是在内核上执行，而 API 大多是在函数库中执行。

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

Dual Mode 工作机制：为即至少需要两种单独运行模式：用户模式与内核模式，通过模式位区分为操作系统执行的任务和为用户执行的任务。当计算机系统执行用户应用时，系统处于用户模式，而用户应用通过系统调用时，请求操作系统服务，系统从用户模式切换到内核模式。当系统引导时，硬件从内核模式开始。操作系统加载后在用户模式执行用户程序。出现中断，硬件从用户模式转到内核模式。即，内核模式为操作系统控制计算机，用户模式为用户程序控制计算机。

采用 Dual Mode 的原因：① 确保操作系统正常运行，必须区分操作系统代码和用户代码的执行。② 而且双重模式执行提供保护手段，以便防止操作系统和用户程序受到错误用户程序的影响

### **4. 概述操作系统需要提供的服务有哪些？**

- ① 用户界面
- ② 程序执行
- ③ I/O 操作
- ④ 文件系统操作
- ⑤ 通信

- ⑥ 错误检测
- ⑦ 资源分配
- ⑧ 记账
- ⑨ 保护与安全

## 5. 分别阐述 Monolithic 结构，层次化结构，模块化结构和微内核结构的特点和优劣。

### Monolithic 结构

特点：采用有限结构，由两个独立部分组成：内核与操作系统

优点：内核中包含大量功能且系统调用接口和内核通信的开销非常小

缺点：单片结构使得 UNIX 难以实现与设计

### 层次化结构

特点：操作系统被分成许多层，每一层都基于低层结构实现层层叠加，底层为硬件，顶层为用户接口

优点：简化了构造和调试，所选的层次只能调用更低层的操作和服务。

缺点：相比其他方法效率稍差，每层都会为系统调用增加额外开销，最终导致系统调用需要执行更多时间

### 模块化结构

特点：内核有一组核心组件，在启动或是运行时，内核都可通过模块链入额外服务。内核提供核心服务，而其它服务可在内核运行时动态实现。

优点：比分层系统更加灵活

### 微内核

特点：内核中删除所有不必要的部件，将他们当作系统级与用户级的程序来实现，

其内核较小。

优点：便于扩展操作系统，新服务可在用户空间增加，内核需要修改时修改量也较小，可移植性、安全性与可靠性更好

缺点：微内核性能稍差。

## **6. 举例说明什么是机制与策略分离的设计原则。**

机制与策略分离的设计原则：机制决定如何做，而策略决定做什么。

例如，定时器是一种保护 CPU 的机制，但是为某个特定用户应将定时器设置成多长时间，这是个策略问题。

## **7. 概述 multi-programming 和 multi-tasking 的概念及其设计目的。**

Multi-programming 概念：多道程序设计通过在内存中同时保存多个任务，使得 CPU 总有一个执行作业，从而提高 CPU 利用率。

Multi-programming 设计目的：由于单个程序并不能让 CPU 和 I/O 设备一直保持忙碌状态，设计 Multi-programming 可以使 CPU 始终工作而非空闲，从而最大化 CPU 使用效率。

Multi-tasking 概念：多任务是多道程序设计的自然延伸，CPU 通过切换作业来执行多个作业，由于 CPU 切换频率很高，用户在程序运行时可以进行交互。

Multi-tasking 设计目的：使得用户在程序进行时可以与计算机进行交互，而且分时操作系统允许多用户共享一台计算机。