

## 2.1 操作系统提供的服务和功能可以分为两大类。简要描述这两个类别，并讨论它们的区别

答：

操作系统提供的一类服务是在系统中不同并发运行的进程之间强制保护。进程只允许访问与它们关联的内存位置。此外，进程不允许损坏与其他进程相关联的文件。进程也不允许在没有操作系统介入的情况下直接访问设备。

操作系统提供的第二类服务是提供低层硬件不能直接支持的功能。虚拟内存和文件系统就是操作系统提供的两类服务的例子。

或者说分为用户功能和系统程序两大类。

用户功能包括用户界面、I/O 操作、程序执行、文件系统、资源分配、保护与安全等

系统程序包括文件管理、信息状态、文件修复、程序加载与执行、后台服务等

区别在于一个服务于用户，一个服务于较底层的程序部分。

## 2.5 对于操作文件和设备。采用同样的系统调用接口有什么优点和缺点

答：

优点

可以像访问文件系统中的文件一样访问每个设备。因为大多数内核都是通过文件接口来处理设备的，所以相对容易通过实现具体硬件的代码来支持抽象的文件接口来增加一个新的设备驱动器。因此，这个有利于双方用户代码的发展，它可以以同样的方式来访问设备和文件，并且设备驱动代码可以支持一个定义明确的 API。

缺点

使用相同的接口的弊端在于它可能很难捕捉在文件访问 API 上下文中某些设备的功能，从而导致功能损失或性能损失。

## 2.7 进程间通信的两个模型是什么？这两种方案各有何长处短处。

答：

一个是消息传递模型，一个是共享内存模型。

消息传递模型：消息可以在进程之间直接或间接地通过一个公共的部分进行交换。对小数量的数据很有用，因为不需要避免冲突，对计算机之间的通信也很容易实现。但是速度较慢。

共享内存模型：速度更快，通信更加便利，因为可以在内存中进行传输。但是不利于进程之间的保护和同步。

## 2.10 采用微内核设计系统的主要优点是什么？用户程序和系统服务在微内核架构内如何交互？采用微内核设计的缺点是什么？

答：

优点：（1）添加新的服务不需要修改内核。（2）操作更安全，因为在用户模式下进行的操作比在内核模式下进行的操作多。

交互方式：用户程序和系统服务在微内核架构中通常使用进程间通信机制例如消息传递来交互。这些消息由操作系统传递。

缺点：微内核架构的主要缺点是与进程间通信相关的开销有关，需要频繁的使用操作系

统消息传递功能来使用户进程和系统服务可以彼此交互。