

HW1

姓名：王艺洲

学号：PB21111722

1.请分别从系统和用户的角度，阐述操作系统的功能，并具体描述操作系统需要提供哪些服务。

系统角度：操作系统作为资源分配器，管理系统可能用到的比如CPU时间，内存文件。文件存储空间，I/O设备等资源。

用户角度：让单个用户单独使用资源，优化用户进行的工作，使用户使用方便；在一些情况下同时让用户使用网络和服务器的共享资源，并允许用户共享资源以及交换信息。

操作系统提供的服务：

用户界面：提供用户界面，可有多种形式：命令行界面，批处理界面，图形用户界面。有些系统甚至提供了两种甚至所有三种界面。

程序执行：系统应能加载程序到内存，并加以运行。程序应能结束执行，包括正常和不正常。

I/O操作：为了效率和保护，用户通常不能直接控制I/O设备，因此操作系统必须提供手段以便执行I/O。

文件系统操作：需要提供读写文件和目录，也需要根据文件名称来创建或删除文件，搜索某个文件，列出文件信息等。

通信：不同进程之间可以互相交换信息，可以通过共享内存或者信息交换来实现。

错误检测：操作系统需要不断检测错误和更正错误。

资源分配：当多个用户或者作业同时运行时，每个都应分配资源，操作系统管理许多不同的资源，负责资源分配和调度。

记账：需要记录用户使用资源的类型和数量。

保护和安全：当多个独立进程并发执行时，一个进程不应干预其他进程或操作系统本身，操作系统要确保可以控制系统资源的所有访问。要求用户向系统认证自己，以获取系统资源的访问权限，保护外部I/O设备不受非法访问，并记录所有非法的闯入企图。

2.请阐述mlti-programming和multi-tasking的概念与设计目的。

多道程序设计（mlti-programming）指在主存中保留多个程序，通过安排作业（编码和数据）使得CPU总有一个执行作业，从而提高CPU利用率。设计目的是加快用户同时运行多个程序的速度，提高CPU利用率。

多任务（multi-tasking）是多道程序设计的自然延伸，指CPU通过切换作业来同时执行多个作业，切换频率很高，用户可以在程序运行时与其交互。设计目的是满足用户同时进行多任务的需求。

3.请阐述缓存的思想以及工作原理。

缓存是通过将信息临时写入更快存储系统，以提高读取时的速度。工作原理如下：信息通常保存在一个存储系统中，当使用时，会被临时复制到更快存储系统，即缓存；当需要特定信息时，首先检查它是否位于缓存中，若是，则直接调用，否则使用源地的信息并将信息放入缓存中以便下次再用。

4.请阐述什么是系统调用，以及系统调用与API的逻辑关系。

系统调用是用户程序向操作系统请求的一种特殊接口，用户可以通过这种接口让操作系统执行某种操作比如文件的处理。

API为方便应用程序员规定了一组函数，包括每个函数的输入参数和返回值，这是可以和程序员直接交互的。程序员通过操作系统提供的函数库来调用API，并进行系统调用，API是用户和系统调用之间的一个媒介。

5.阐述Dual Mode的工作机制，以及采用DualMode的原因。

工作机制：

计算机系统至少需要两种单独工作模式：用户模式和特权模式。对应CPU的两种工作态：用户态和内核态。用户态下CPU只能执行一些受限的指令，不能直接访问硬件设备。而在内核态下可以访问任何设备，执行任何指令。

采用原因：

双重模式执行提供了保护手段，将可能引起损害的机器指令作为特权指令，即只有在内核态下才允许执行的指令，以便防止操作系统和用户程序受到错误用户程序的影响。

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

Monolithic大内核结构：

一种操作系统内核架构，特点是整个内核程序是一个单一的二进制执行文件，运行在核心空间的监管者模式下。Monolithic大内核结构将各种功能模块都集成在一个大的程序中。

优点是性能高，因为各个模块直接调用函数或共享数据。缺点是可扩展性差，因为任何修改都需要重新编译整个内核；而且可靠性低，因为任何一个模块出错都可能导致整个系统崩溃。

层次化结构：

将一个大型复杂的系统分解成若干单向依赖的层次，即每一层都提供一组功能且这些功能只依赖该层以内的各层。最低层为硬件，最高层为用户接口

优点是简化了构造和调试，可逐层调试，且发现错误时缩小了错误范围。缺点是需要仔细规划，需要合理定义各层。效率相对于其他方法较低，需要逐层调用，执行更长时间。

模块化结构：

将操作系统划分为若干个功能不同的模块，每个模块都有明确的接口和功能。模块之间可以相互调用，但不能直接访问对方的内部数据。。

优点是比分层系统更加灵活，性能更高，直接调用函数或者共享数据。缺点是可扩展性差，任何修改都需要重新编译其他模块。

微内核结构：

从内核中删除所有不必要的部件，而将它们当作系统级与用户级的程序来实现。

优点为便于扩展操作系统，所有新服务可在用户空间内增加，不需要修改内核，当需要修改时修改也很小。缺点是由于增加的系统功能的开销，微内核的性能会受损。

7.举例说明什么是机制与策略分离的设计原则，并说明该设计的好处。

设计原则：

机制决定如何做，而策略决定做什么。将设计与策略分离，把实现功能的基本模型和如何具体实现分离开。

例子：

Linux内核，它遵循了机制与策略分离的原则，在内核中提供了一些基本功能（如进程管理、内存管理、文件系统）作为机制，在用户空间中提供了各种应用程序（如shell、编辑器、编译器等）作为策略。

好处：

增加了系统的灵活性与可移植性，使一个机制可以用于多种策略，一个策略可以用于多种机制。