

操作系统

2021/3/19

第一次作业

1. 请分别从系统用户的角度，阐述操作系统的主要功能？

系统：在系统角度，操作系统是资源分配器，是与计算机硬件紧密相连的程序，起到分配 CPU 时间、内存空间、磁盘空间的功能

用户：在用户层面，操作系统是用户与计算机硬件之间的接口，起到了管理应用程序、为应用程序提供服务、结束进程的功能

2. 解释什么是系统调用？阐述系统调用与 API 的逻辑关系。

系统调用是内核提供的一系列管理硬件资源函数的接口，当用户进程必须访问内核或使用某个内核函数时，就得使用系统调用。

API 是程序员在用户空间下可以直接使用的函数接口。这些函数都具有一定功能，说明了如何获得一个给定的服务，跟内核没有必然的联系。提供应用程序与开发人员基于某软件或硬件的以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

逻辑关系：

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。

联系：API 是一个提供给应用程序的接口，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。

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

工作机制：Dual Mode 指用户模式和内核模式，当计算机系统执行用户应用时，系统处于用户模式。当用户通过系统调用，请求操作系统服务时，系统就从用户模式切换到内核模式，以满足请求。具体而言就是，每当操作系统能够控制计算机时，它就处于内核模式。在将控制交回给用户程序前，系统会切换到用户模式。

原因：Dual Mode 提供保护手段，以便防止操作系统和用户程序收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？

●**用户界面：**这种界面可以有多种形式，一种是命令行界面，采用文本命令，并用某一种方法输入；一种是图形用户界面。

●**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，包括正常或不正常（并给出错误）。

●**I/O 操作：**程序的 I/O 操作可能涉及到保护文件特定功能使用，为了提高效率和进行文件保护，用户通常不能直接控制 I/O 设备，因此，操作系统提供 I/O 操作的方法。

●**文件系统操作：**操作系统需要提供文件系统的目录结构和读写文件服务，创建文件删除文件的保护机制，访问权限机制来满足用户需求。

●**通信：**一个进程需要与另一个进程交换信息，这种通信可以通过共享内存来实现，也可以通过信息交换技术实现。通信可以发生在同一台计算机的两个进程之间，也可以发生在网络连接的不同计算机上的两个进程之间。

●**错误检测：**操作系统需要不断检测错误和更正错误。对于每一类错误，操作系统必须采取适当的动作，确保计算的正确和一致。

●**资源分配：**当计算机有多个用户或多个作业运行时，系统就会给他们每一个分配资源，操作系统可以管控多种不同的资源。

●**统计**：记录用户使用资源的种类和数量，用于重新配置系统提高计算服务。

●**保护和安全**：多个进程并发执行时，操作系统会对两个进程之间的访问资源进行控制，确保一个进程不能干预另外一个进程。操作系统需要设置密码用来保护自身不受外界侵犯，记录外界强制破解的企图信息。

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：系统调用接口之下和物理硬件之上的所有部分为内核，将所有功能集中在同一层

优点：性能优势，系统调用接口和内核通信的开销非常小

缺点：难以实现和设计

层次化结构

特点：将操作系统分成若干层，最底层为硬件，最高层为用户接口，每层只能调用更低层的功能和服务

优点：1. 简化了构造和调试

2. 每层为更高层隐藏一定的数据结构、操作和硬件

缺点：1. 难以合理定义各层

2. 与其他方法相比效率稍差

模块化结构

特点：内核提供核心服务，其他服务可在内核运行时动态实现

优点：1. 更加灵活，因为任何模块都可以调用其他模块

2. 更加有效，因为模块无需调用消息传递来进行通信

微内核结构

特点：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：1. 便于扩展操作系统，新的服务可在用户空间内增加而不需要修改内核，当内核确实需要修改时所做的修改也很小

2. 很容易从一种硬件平台移植到另一种硬件平台

3. 提供了更好的安全性与可靠性

缺点：由于增加的系统功能的开销，微内核的性能会受损

6. 举例说明什么是机制与策略分离的设计原则

例 1. 基于微内核的操作系统通过实现一组基本的简单构造块，将机制与策略分离发挥到了极致。这些块几乎不受策略限制，允许通过用户创建的内核模块或用户程序本身添加更高级的机制和策略。

例 2. Windows 已经将机制和策略紧密地编码到系统中，以便在运行 Windows 操作系统的所有设备上实施全局的外观和感觉。全部应用程序具有类似的接口，因为接口本身内置于内核和系统库中。

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

multi-programming: 允许多个程序（作业）同时进入一个计算机系统的内存并启动进行交替计算的方法。计算机中可以同时存放多道程序，从宏观上来看它们是并行的，多道程序都同时处于运行过程中，但都未运行结束，但是微观上是串行的，轮流占用 CPU 交替执行

目的：提高 CPU 的利用率，充分发挥计算机系统部件的并行性。

multi-tasking: 一个计算机配置多个 CPU，同时执行多道程序或多个指令序列

目的：最大限度的利用硬件资源