

Chapter 2

2.1 What is the purpose of system calls?

让用户级进程可以请求操作系统所提供的服务

2.2 What are the five major activities of an operating system with regard to process management?

1. 创建和删除用户进程和系统进程
2. 暂停和重启进程
3. 提供进程同步机制
4. 提供进程通信机制
5. 提供死锁处理机制

2.3 What are the three major activities of an operating system with regard to memory management?

1. 记录内存的哪部分正在被使用以及被谁使用
2. 当内存可用时，决定哪些进程加载到内存
3. 根据需要分配和释放内存

2.4 What are the three major activities of an operating system with regard to secondary-storage management?

1. 空闲空间管理
2. 存储空间分配
3. 硬盘调度

2.5 What is the purpose of the command interpreter? Why is it usually separate from the kernel?

命令解释器的主要功能是获取并执行用户指定的下一条命令。用于在用户和操作系统之间提供直接的通讯，从用户或命令文件中读取命令，并创建执行这些命令的进程，通常是将它们转换为一个或多个系统调用。

将命令解释器与内核分开，便于对命令解释器进行修改。

2.6 What system calls have to be executed by a command interpreter or shell in order to start a new process?

在Unix/Linux系统中，需要先执行fork系统调用，然后再执行exec系统调用才能启动新进程。fork调用克隆当前执行的进程，而exec调用基于不同的可执行文件覆盖调用进程上的新进程。

2.7 What is the purpose of system programs?

系统程序充当操作系统的一部分，位于用户界面和系统调用之间，为用户提供基本的系统级功能。

2.8 What is the main advantage of the layered approach to system design? What are the disadvantages of using the layered approach?

Advantage:

1. 模块化

每一层只执行其计划执行的任务。

2. 简化调试与验证

由于各层是离散的，所以很容易调试。假设CPU调度层出现错误，因此开发人员只能搜索要调试的特定层，这与所有服务都集中在一起的单片系统不同。

3. 方便更新

在特定层中所做的修改不会影响其他层

4. 不能直接访问硬件

硬件层是设计中存在的最内层。因此，用户可以使用硬件服务，但不能直接修改或访问它，这与用户可以直接访问硬件的简单系统不同。

5. 抽象

每一层都涉及到它自己的功能。因此，其他层的功能和实现对它是抽象的。

Disadvantage:

1. 实施较为困难

由于上层只能访问下面的层的服务，所以层间的顺序难以确定。例如，后备存储层使用内存管理层的服务，因此，它必须保持在内存管理层以下。

2. 执行速度较慢

如果一层想要与另一层交互，需要发送一个请求，而且该请求必须遍历两个交互层之间存在的所有层，会大幅增加响应时间。

2.9 List five services provided by an operating system, and explain how each creates convenience for users. In which cases would it be impossible for user-level programs to provide these services? Explain your answer.

1. 程序执行

操作系统将可执行文件的内容(或部分)加载到内存中并开始执行。用户级程序难以实现正确分配CPU时间

2. I/O 操作

磁盘、磁带、串行线和其他设备必须以非常低的级别进行通信。用户只需指定设备和要在其上执行的操作，而系统将该请求转换为特定于设备或特定于控制器的命令。用户级程序难以实现只能访问它们有权访问的设备，并且只有在它们未被使用时才能访问它们。

3. 文件系统操作

在文件创建、删除、分配和命名方面有许多用户不应该执行的细节。删除文件需要删除名称文件信息并释放分配的空间，还必须检查保护措施，以确保正确访问文件。用户程序既不能确保遵守保护方法，也不能被信任在文件删除时只分配空闲块和释放块。

4. 通信

系统之间的消息传递需要将消息转换为信息包，通过通信介质传输，发送到网络控制器，并由目的地系统重新组装。必须进行分组排序和数据校正。同样，用户程序可能不会协调对网络设备的访问，或者可能会接收发往其他进程的数据包。

5. 错误检测

错误检测在硬件和软件两个级别上进行。在硬件级别，必须检查所有数据传输，以确保数据在传输过程中没有损坏。必须检查介质上的所有数据，以确保它们在写入介质后没有更改。在软件级别，必须检查介质的数据一致性；例如，已分配和未分配的存储块数量是否与设备上的总数匹配。此时，错误通常与进程无关(例如，磁盘上的数据损坏)，因此必须有一个全局程序(操作系统)来处理所有类型的错误。此外，通过由操作系统处理错误，进程不需要包含代码来捕获和纠正系统上可能出现的所有错误。

2.12 The services and functions provided by an operating system can be divided into two main categories. Briefly describe the two categories, and discuss how they differ.

一种是为了提供对用户有帮助的功能，另一种是为了确保系统本身的有效运行。

第一种包括用户界面、程序执行、I/O操作、文件系统操作、通信和错误检测。

第二种包括资源分配、记录和保护以及安全。

不同之处在于其目的一种只是服务于用户，而另一种只是服务于系统

2.13 Describe three general methods for passing parameters to the operating system.

1. 将参数置于寄存器（当参数的个数多于寄存器时不可行）
2. 将块或表中的参数存储在内存汇总，并将其地址作为参数传递给寄存器
3. 将参数压栈，由操作系统弹出

PS：两种方法不限制传递的参数数量和长度

2.15 What are the five major activities of an operating system in regard to file management?

1. 创建和删除文件

文件创建和删除是计算机操作的基础。对于前者，除非以某种形式的文件结构来安排，数据不能以有效的方式存储。对于后者，如果不删除文件并将它们占用的空间重新分配给新文件，存储空间将很快填满。

2. 创建和删除目录

作为在文件中存储数据的需要的必然结果，文件本身需要被安排在目录或文件夹中，以便能够高效地存储和检索。与文件删除非常类似，为了保持系统整洁，需要删除不必要的目录或文件夹。

3. 文件和目录操作指令

由于操作系统允许应用软件使用符号指令执行文件操作，因此操作系统本身需要具有机器级指令集，以便直接与硬件交互。应用程序的符号指令需要由解释器或通过编译应用程序代码转换为机器级指令。操作系统包含用于管理此机器级别文件操作的配置。

4. 映射到外存

操作系统需要能够将文件和文件夹映射到它们在永久存储上的物理位置，以便能够存储和检索它们。这将被记录在某种形式的磁盘目录中，该目录根据操作系统使用的一个或多个文件系统而有所不同。操作系统将包括一种机制，用于定位其已划分文件的单独文件段。

5. 备份文件

计算机的永久存储设备通常包含许多可能发生故障的机械设备，并且存储介质本身可能会降级。操作系统的一个功能是通过在冗余系统中的额外安全和稳定的介质上备份文件来避免数据丢失的风险。

2.16 What are the advantages and disadvantages of using the same system-call interface for manipulating both files and devices?

Advantage:

可以像访问文件系统中的文件一样访问每个设备。由于大多数内核通过此文件接口处理设备，因此通过实现特定于硬件的代码来支持此抽象文件接口，添加新的设备驱动程序相对容易。有利于用户程序代码和设备驱动程序代码的开发，用户程序代码可以被编写来以相同的方式访问设备和文件，设备驱动程序代码可以被编写来支持明确定义的API。

Disadvantage:

难以在文件访问API的上下文中捕获某些设备的功能，从而导致功能损失或性能损失。其中一些问题可以通过使用ioctl操作来克服，ioctl操作作为进程提供通用接口以调用设备上的操作。

2.17 Would it be possible for the user to develop a new command interpreter using the system-call interface provided by the operating system?

可以，命令解释程序允许用户创建和管理进程，还可以确定它们的通信方式(例如通过管道和文件)。而这些功能都可以有系统调用的用户级程序访问，所以用户可以开发新的命令解释程序。

2.18 What are the two models of inter-process communication? What are the strengths and weaknesses of the two approaches?

消息传递模型 (message-passing model) :

strength: 消息可以在进程之间直接交换, 也可以通过公共邮箱间接交换。适用于数据量较小的情况, 并且更容易实现计算机间的通信。

weakness: 它的速度比共享内存模型慢

共享内存模型 (shared-memory model) :

strength: 允许更快的速度和更方便的通信

weakness: 在保护和进程之间的同步方面存在一些问题

2.19 Why is the separation of mechanism and policy desirable?

便于对操作系统进行修改。

将mechanism和policy进行分离, policy可以随意进行更改, 而mechanism不变, 使操作系统更加灵活, 便于操作系统更易满足用户需要。

2.21 What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

Advantage:

1. 添加新的服务不需要修改内核
2. 安全性更高
3. 使操作系统更加可靠

用户程序和系统服务通过使用进程间通信机制(例如消息传递)在微内核体系结构中交互。这些消息由操作系统传递。

Disadvantage:

与进程间通信相关的开销, 以及为了使用户进程和系统服务能够彼此交互而频繁使用操作系统的消息传递功能(上下文切换)。

2.22 What are the advantages of using loadable kernel modules?

loadable kernel modules (LKM) 与普通应用程序之间的区别主要体现在四个方面。

第一, 也是最重要的区别, 普通应用程序运行在用户空间, 而LKM运行在内核空间。通过区分不同的运行空间, 操作系统能够安全地保护操作系统中一些重要数据结构的内容不被普通应用程序所修改, 达到保证操作系统正常运转的目的。

第二, 普通应用程序的目标很明确, 它们从头至尾都是为了完成某一项特定任务。而LKM是在内核中注册并为后续应用程序的请求提供服务的。

第三，普通应用程序可以调用并没有在其中定义的函数，但一个LKM是链接到内核上的，它所能调用的函数只有内核导出来的那些函数。

第四，普通应用程序和LKM处理错误的方式不同。当应用程序中出现错误时并不会给系统造成很大的伤害。LKM则不然，在其中出现的错误对子系统来说通常是致命的，至少对于当前正在运行的进程而言。LKM中的一个错误常常会导致整个系统崩溃。