**Chapter 1 exercises**

16281262 麻锦涛

1 How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security) system? **（内核态和用户态作为保护系统有什么区别）**

**答：**内核态和用户态的区别有以下方式，某些指令只有当CPU处于内核态时才可以执行。同样地，某些硬件设备只有当程序在内核态下执行才能够被访问。只有在CPU处于内核态时，才能够控制中断。因此，CPU处于用户态时的能力有限，从而强制保护关键的资源。

2 Which of the following instructions should be privileged?

a. Set value of timer.

1. Read the clock.
2. Clear memory.
3. Issue a trap instruction.
4. Turn off interrupts.
5. Modify entries in device-status table.
6. Switch from user to kernel mode.
7. Access I/O device.

**答：**a. Set value of timer.设置计时器的值

c. Clear memory.清除内存

e. Turn off interrupts.

f. Modify entries in device-status table.修改设备状态表中的条目

h. Access I/O device.访问I/O设备

受保护的：a、c、e、f、h，剩下的可以在用户态执行。

1. Some CPU s provide for more than two modes of operation. What are two possible uses of these multiple modes?

（**一些CPU提供了两种以上的操作模式，那么这些多重模式有哪些用途？**）

答：尽管大多数系统只区分用户和内核模式，但有些CPU支持多种模式。可以使用多种模式来提供更细粒度的安全策略。例如，我们可以区分不同类型的用户模式，而不是区分用户和内核模式。也许属于同一组的用户可以执行彼此的代码。当其中一个用户正在运行代码时，机器将进入指定模式。当机器处于这种模式时，组中的成员可以运行属于该组中任何其他人的代码。另一种可能是在内核代码中提供不同的区别。例如，一个特定的模式可以允许USB设备驱动程序运行，这意味着USB设备可以在无需切换到内核模式的情况下进行服务。

1. Timers could be used to compute the current time. Provide a short description of how this could be accomplished.

答：一个程序可以使用灵活的方法来计算当前的时间使用计时器中断。该程序可以在未来一段时间内设置一个定时器，然后进入睡眠状态。当它被中断唤醒时，它可以更新它的本地状态，它用来跟踪它迄今收到的中断的数量。然后，它可以重复此过程，持续设置定时器中断，并在实际引发中断时更新其本地状态。

1. Give two reasons why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device? 5 In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in various security problems. a. What are two such problems？b. Can we ensure the same degree of security in a time-shared machine as in a dedicated machine? Explain your answer.

答:两个或者更多的部件需要交换数据，以及组成部件以不同的速度完成转换时，缓存是十分有用的。缓存通过在个组成部件之间提供一个中间速度的缓冲区来解决转换问题。如果速度较快的设备在缓存中发现它所要的数据，它就不需要再等待速度较慢的设备了。缓存中的数据必须与组成部件中的要一致。如果一个组成部件中的数据值改变了，缓存中的这个数据也必须更新。在多进程系统中，当有不止一个进程可能进入同一个数据时，这就成了一个显著的问题。一个组成部件将会被一个同等大小的组成部件所消除，但是只有当;(a)缓存和组成部件有相同状态存储能力（也就是，当断电的时候，组成部件还能保存它的数据，缓存也一样能保存它的数据），(b)缓存是可以负担的起的，因为速度更快的存储器意味着更高的价格。

1. What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

答：中断是一种硬件生成的系统内部流量变化。调用一个中断处理程序来处理中断的原因；然后将控制返回到中断的上下文和指令。陷阱是一种软件生成的中断。中断可用于指示I/O的完成，以避免对设备轮询的需要。陷阱可以用来调用操作系统例程或捕获算术错误。

1. Some computer systems do not provide a privileged mode of operation in hardware. Is it possible to construct a secure operating system for these computer systems? Give arguments both that it is and that it is not possible.

答：此类机器的操作系统需要始终保持在控制（或监视模式）中。可以通过两种方法来实现：所有用户程序的软件解释（例如一些基本的、APL和Lisp系统）。软件解释器将在软件中提供硬件不提供的内容。

这意味着所有程序都是用高级语言编写的，这样的话所有对象代码都是由编译器生成的。编译器将生成（内联或通过函数调用）保护检查硬件是否丢失。

1. What is the purpose of system calls?

答：让用户级进程可以请求操作系统所提供的服务

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

答：在UNIX系统中，需要先执行fork系统调用，然后再执行exec系统调用，才能启动新的进程。fork调用复制了当前正在执行的进程，而exec调用覆盖基于调用进程上不同可执行文件的新进程

1. What is the purpose of system programs?

答：1.为程序开发和运行提供了方便的环境

2.给用户提供基本的公共功能函数，为用户在不用自己写代码的情况下解决公用问题

1. What is the main advantage of the layered approach to system design? What are the disadvantages of the layered approach?

答：1.在模块化设计所有情况下，在设计操作系统模块化的方式有几个优点。该系统易于调试和修改，因为修改会影响系统的只有有限的部分，而不是触摸操作系统的所有部分。信息保存仅在需要的地方，并且仅在一个限定和限制区域访问，因此任何错误影响该数据必须被限制在一个特定的模块或层。

2. 缺点是模块之间调用关系复杂、相互依赖,从而使分析、移植和维护系统较易出错

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

主要有三种方法：

1.通过寄存器来传递参数

2.寄存器传递参数块的首地址

3.参数通过程序存放或压进堆栈中，并通过操作系统弹出堆栈。

1. 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?

优点主要包括以下几点：

1）增加一个新的服务不需要修改内核

2) 在用户模式中比在内核模式中更安全、更易操作

3) 一个简单的内核设计和功能一般导致一个更可靠的操作系统

用户程序和系统服务通过使用进程件的通信机制在微内核中相互作用，例如发送消息。这些消息由操作系统运送。微内核最主要的缺点是与进程间通信的过度联系和为了保证用户程序和系统服务相互作用而频繁使用操作系统的消息传递功能。