

Ch1 绪论

OS 是硬件和软件之间的软件

设计目标:

方便, 高效, 易维护, 开放

功能:

处理器管理

存储器管理

设备管理

文件管理

用户接口

用户接口

命令接口

命令行

图形化

系统调用

API

发展历史

一代 1946~1950 末期(1954) 电子管

二代 1960-65 晶体管 监督程序 (monitor) 为代表的批处理系统

三代 1965-75 集成电路 多道程序系统

四代 1975-今 大规模集成电路 分时系统、实时系统、嵌入式系统

发展阶段

无操作系统阶段

监督程序阶段

联机批处理系统

脱机批处理系统

多道程序系统(并行)

快速发展阶段

分时操作系统

同时性;独立性;及时性;交互性

实时操作系统

及时性;通用性;操作性;可靠性

网络 OS

分布式 OS

嵌入式 OS

云 OS

Ch2 OS 的结构

机制

需要提供什么功能（机制）（what）

策略

怎样实现这些功能（策略）（how）

Linux 和 Windows 操作系统内核主要用 C 编写

用户接口

命令接口

命令行界面

图形用户界面

系统调用

系统调用就是一种特殊的接口。通过这个接口，用户可以访问内核空间。系统调用规定了用户进程进入内核的具体位置。

具体步骤：用户进程-->系统调用-->内核-->返回用户空间。

内核提供一系列具备预定功能的多内核函数，通过一组称为系统调用（system call）的接口呈现给用户。系统调用把应用程序的请求传给内核，调用相应的内核函数完成所需的处理，将处理结果返回给应用程序。内核模式。

系统调用与 API 的关系

API（应用编程接口）是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码。

API 函数通常为应用程序员调用实际的系统调用。（API 是对系统调用的再封装）

API 使得程序的可移植性好；实际系统调用比 API 更为注重细节且更加难用。

系统调用与 API 的区别与联系(简答题)

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：一个 API 可能会需要一个或多个系统调用来完成特定功能。通俗点说就是如果这个 API 需要跟内核打交道就需要系统调用，否则不需要。

程序员调用的是 API（API 函数），然后通过与系统调用共同完成函数的功能。

因此，API 是一个提供给应用程序的接口，一组函数，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。

并不是所有的 API 函数都一一对应一个系统调用，有时，一个 API 函数会需要几个系统调用来共同完成函数的功能，甚至还有一些 API 函数不需要调用相应的系统调用（因此它所完成的不是内核提供的服务）

向系统传递参数[系统调用时]:

1. 直接通过寄存器
2. 参数保存在内存中（内存地址->寄存器）
3. 参数通过程序压入堆栈，并通过操作系统弹出

OS 结构

1. 整体式
2. 层次式
3. 微内核
 - ① 最基本的功能保留在内核，不需要在核心态执行的功能转移到用户态执行，从而降低了内核的设计复杂性。
 - ② 移出的操作系统代码根据分层的原则被划分成若干服务程序，它们的执行相互独立，交互则都借助于微内核进行通信。
 - ③ **功能：**低级存储管理；进程间通信；I/O 和中断管理
 - ④ **优点：**

有效地分离了内核与服务、服务与服务，使得它们之间的接口更加清晰，维护的代价大大降低，各部分可以独立地优化和演进，从而保证了操作系统的可靠性
 - ⑤ **缺点：**

因为需要频繁地在核心态和用户态之间进行切换，操作系统的执行开销偏大
4. 模块化
 - ① 利用面向对象编程技术来生成模块化的内核。其全称为动态可加载模块化结构。
 - ② **优点：**
 - 1) 将内核映象的尺寸保持在最小，并具有最大的灵活性；
 - 2) 便于检验新的内核代码，而不需重新编译内核并重新引导。
 - ③ **缺点：**
 - 1) 差的内核模块会导致系统崩溃
 - 2) 内核必须维护符号表
 - 3) 内核要维护模块之间的依赖性

虚拟机 (Virtual Machine) 指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。

虚拟机监视器(VMM)，虚拟化技术的核心，处于硬件和虚拟机之间(或者宿主 OS 和虚拟机之间)

5. A microkernel is a kernel (D) .

- A) containing many components that are optimized to reduce resident memory size
- B) that is compressed before loading in order to reduce its resident memory size
- C) that is compiled to produce the smallest size possible when stored to disk
- D) that is stripped (被剥离) of all nonessential(非本质的) components

6. (B) provide(s) an interface to the services provided by an operating system.
A) Shared memory B) System calls C) Simulators D) Communication
7. (D) is not one of the major categories of system calls.
A) Process control B) Communications C) Protection D) Security
8. Microkernels use (A) for communication.
A) message passing B) shared memory C) system calls D) virtualization
9. (A) is/are not a technique for passing parameters from an application to a system call.
A) Cache memory B) Registers
C) Stack D) Special block in memory
10. System calls can be run in either user mode or kernel mode. (F)
11. Application programmers typically use an API rather than directory invoking system calls. (T)
12. Interrupts may be triggered by either hardware or software (T)
13. A dual-core system requires each core has its own cache memory. (F)