

第一章 概述

2021年3月19日 15:42

1.1 操作系统基础

1. 操作系统是一层特殊的软件，它的职责是管理硬件资源。
2. 操作系统的发行版本一定是与特定硬件结构相关的，比如针对Intel x86的操作系统
3. 常见的操作系统是用C语言来编写的，部分地方会用到汇编语言

1.1.1 计算机系统的硬件资源管理

1. 操作系统的硬件资源分为计算资源和存储资源

- 计算资源：CPU

主流的计算机通常有一个或多个CPU，或者一个CPU中有多个核。操作系统必须合理地安排和调度任务，使得多个CPU或者多核尽可能的利用起来，避免出现竞争或闲置的情形。这一职责又被称为任务调度。在现代操作系统中，由于任务是由进程或线程来完成的，操作系统的这部分功能也称为进程调度或线程调度。

计算机提供的时钟中断可以完成这一点，由此产生了不同的调度算法

- 存储资源：内存(RAM，随机访问存储器)、外存（辅助存储器）

由于外存是通过标准的I/O来管理的，而内存是CPU直接通过系统总线来访问的，所以这里只考虑内存资源的管理。内存是CPU执行一个任务的物质基础，CPU内部的寄存器具备计算的能力，计算的数据来源于系统内存。

现代操作系统中，每个任务都有其独立的内存空间，从而避免任务之间产生不必要的干扰，所以操作系统有责任为每个任务提供相对独立的内存空间。物理内存划分常见的作法是段式内存寻址和页式虚拟内存管理。Intel x86体系结构同时支持段式寻址和页式虚拟内存映射。Windows和Linux都选择了虚拟内存映射作为内存管理的硬件基础。

2. 操作系统如何管理硬件资源

除了计算资源和内存资源的管理外，操作系统对其他资源都通过I/O来管理。

操作系统为了和I/O设备打交道，需要三方面的技术保障：

- a. CPU通过特定的指令来控制I/O设备
- b. I/O设备通知CPU发生了特定的事情
- c. 在系统主内存和设备之间传输数据

通常，CPU直接访问设备的寄存器来操作一个设备，在Intel x86系统上，CPU通过IN/OUT指令来做到这一点。设备寄存器是另一个地址空间，CPU通过I/O端口来访问它们。现代计算机中，I/O端口的分配跟软件和硬件都有关系。

3. CPU如何检测设备的工作状态：

当设备的状态发生变化时，它能够主动地通知CPU，从而操作系统可以采取相应的措施。这即是设备中断机制。当键盘设备接收到按键动作时，它产生一个中断，告诉CPU(和操作系统)，当前哪个键被按下了。中断也有编号，中断的编号被视为系统全局资源。

4. 计算机中的大数据传输

计算机的计算处理能力往往仅限于在CPU内部寄存器和主内存之间进行，但是为了实现基本算术计算以外的其他各种能力，通常有必要让设备中的数据也参与到计算

中来，所以，在设备与CPU寄存器或主内存之间传输数据往往是必要的功能。例如，计算机通过磁盘设备可以实现永久存储，通过显示控制器实现彩色显示甚至三维模拟。实现数据传输的方法有多种，如果设备本身的数据量很小，则可以直接通过in指令来读取设备中的数据，或通过out指令输出到设备中。或者也可以映射一段地址范围到设备中，这样，当CPU访问这块地址范围时，实际上是在访问设备的内存，而不是系统的主内存。另外一种适合于大块数据传输的技术是

DMA (DirectMemory Access, 直接内存访问)。像硬盘控制器和网络控制器就通常采用DMA方式来传输数据，CPU只须设置好数据传输的方向、位置、数量等信息，就可以启动DMA传输了。DMA传输可以与CPU计算同时进行，但是DMA使用的总线不能与CPU使用的发生冲突，它可以趁CPU不用总线的时刻来传送数据，也可能会因此而阻塞CPU指令的执行。**DMA传输影响CPU执行指令的程度取决于DMA控制器的传输策略。**

5. I/O模型

从操作系统的角度来看，考虑到I/O设备的多样性和出现新设备的可能性，操作系统有必要定义一个框架来容纳各种各样的I/O设备，并且允许操作系统发布之后还能够为新的设备提供支持。所以，除了专用操作系统以外，现代操作系统都会提供一个I/O模型，允许设备厂商按照此模型编写设备驱动程序(Device Driver)，并加载到操作系统中。

I/O模型对下提供了控制硬件设备的能力，对上为应用程序访问硬件提供了一个标准接口，同时I/O模型也必须能够让操作系统有效地管理设备驱动程序。

在Windows系统中，第三方厂商可以使用Windows的I/O模型来编写设备驱动程序。Windows本身在发行时，已经随带了大量主流设备的驱动程序，所以，Windows系统在安装阶段可以自动将识别出来的设备的驱动程序安装到系统中。

1.1.2 为应用程序提供执行环境

操作系统之所以要管理各种硬件资源，是为了更好地为上层应用程序提供服务。应用程序并不直接与机器的各种硬件设备及资源打交道，而是运行在一个抽象层上，即操作系统提供的功能语义层。**尽管不同的操作系统提供的这一功能语义层的接口可能不一致，甚至相距甚远，但现代操作系统的语义层在概念上仍然比较一致。**

进程与线程的概念：

尽管不同的操作系统对于进程和线程的定义不尽相同，但是每个任务都应该有它自己的执行环境，即该任务的控制流以及函数调用的层次递进痕迹。在UNIX中，一个进程代表了一个应用任务，它记录了该任务的执行状态；在WINDOWS中，一个线程代表了一个任务，其中也记录了该任务的执行状态。多任务操作系统得使用适当的算法决定什么时候让哪个任务执行，这就就是任务调度，或称为**进程调度或线程调度**。Windows的线程调度方案属于**基于优先级的抢占式调度算法，甚至根据任务的各种特性来进行优先级的动态微调。**

现代操作系统通常会提供多种同步机制，例如**互斥体(mutex)、信号量(semaphore)、临界区(critical section)**等。

应用程序的独立执行空间：

在现代操作系统中，进程也代表了一个应用程序和它的执行空间。不同进程的执行空间是相互隔离的，这是现代操作系统的基本需求。操作系统必须在处理器的硬件特性基础之上，实现一套行之有效的空间隔离方案。每个进程有它自己的内存空间，并且无法访问其他进程的内存空间。进程之间如果要共享数据，则必须通过操作系统提供的机制来进行。Intel x86体系结构上的操作系统基本上都利用硬件的虚拟内存映射机制来隔离每个进程的内存空间。所以，**操作系统的职责是为每个进程维护好从虚拟地址到物理地址的映射关系，并且管理好物理内存的分配和回收。**

应用程序的通信：

除了进程的空间隔离性，操作系统还必须提供相应机制让不同的进程可以相互通信，毕竟很多软件需要进程之间的协作来完成一些上层功能。**同步机制和跨进程共享内存是典型的进程间通信(IPC, Inter-Process Communication)手段**

什么是操作系统中间模块：

我们通常所指的操作系统的范畴包括上面提到的资源管理功能，以及为应用程序提供的各种抽象概念和抽象接口的具体实现。但是在应用程序和操作系统核心模块之间，往往还存在很多预封装好的模块，以便于应用程序开发人员可以复用这些模块，从而高效地开发应用软件，缩短应用软件的开发周期。这些模块既可能是操作系统厂商提供的，也可能是独立软件厂商提供的。它们既可能构成一个通用的中间件，也可能只是随编译器而提供的一个库模块。

例如：windows系统的发行版本包含大量的中间模块(例如COM/COM+、Winsock2、.NET等)。

1.2 学习操作系统必备知识

体系结构、汇编语言、编译原理和C/C++语言

1.3 Windows操作系统发展历史

看看就好，没啥用

MS-DOS => Windows 95/98/Me

WINDOWS NT => Windows NT 3.1/3.5/3.51/4.0 => Windows XP/Server 2003 => win7 => win10

1.4 Windows内核的版本

| 表1.1 Windows内核的版本列表 | | |
|---------------------|-------------------------|---------|
| NT 内核版本 | 发布的操作系统 | 发布日期 |
| NT 3.1 | Windows NT 3.1 | 1993.7 |
| NT 3.5 | Windows NT 3.5 | 1994.9 |
| NT 3.51 | Windows NT 3.51 | 1995.5 |
| NT 4.0 | Windows NT 4.0 | 1996.7 |
| NT 5.0 | Windows 2000 | 1999.12 |
| NT 5.1 | Windows XP | 2001.8 |
| NT 5.2 | Windows Server 2003 | 2003.3 |
| | Windows XP SP2 | 2004.8 |
| | Windows XP 64 位版本 | 2005.4 |
| | Windows Server 2003 SP1 | |
| NT 6.0 | Windows Vista | 2007.1 |
| | Windows Server 2008 | 2008.2 |
| | Windows Vista SP1 | |
| NT 6.1 | Windows 7 | 2009.10 |
| | Windows Server 2008 R2 | |

NT 5.2版本 就是WRK公开的内核，我们可以来研究它。

1.5 操作系统的研究与发展

看看就好，对学windows内核没啥实际帮助。

1.6 总结

操作系统是一种特殊的软件，它一方面管理系统中的硬件资源和设备，另一方面为应用软件提供基本的系统服务。由于它需要直接跟硬件打交道，所以，为了学习像Windows这样的复杂操作系统，基本的硬件体系结构知识和汇编语言、c语言的编程节能是必不可少的。