

1、操作系统的定位

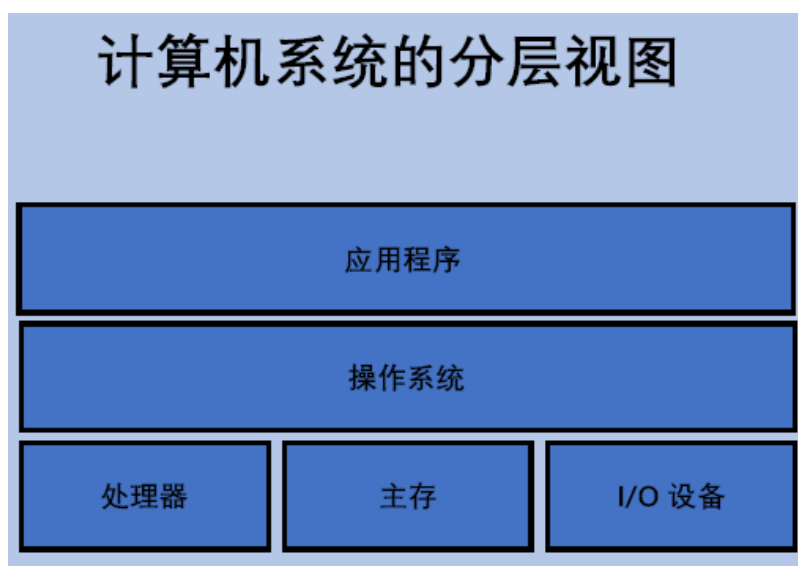
2、什么是进程/任务 (Process/task)

4、CPU分配---进程调度 (Process Scheduling)

5、内存分配—内存管理 (Memory Manage)

6、进程间通信 (Inter Process Communication)

1、操作系统的定位



1.操作系统就是一个“管理软件”。（所谓管理，两个方面：描述 + 组织）

1.管理硬件设备

2.管理软件资源

2.两个基本功能

1.防止硬件被时空的应用程序滥用

2.像应用程序提供简单一致的机制来控制复杂而又大庭的低级硬件设备

2、什么是进程/任务 (Process/task)

进程是操作系统对一个正在运行的程序的一种抽象，换言之，可以把进程看做程序的一次运行过程；同时在操作系统内部，**进程又是操作系统进行资源分配的基本单位。**

| 应用 (3) | 状态 | PID | CPU | 内存 | 磁盘 | 网络 | GPU | GPU 内存 |
|------------------------------------|----|-------|------|----------|----------|----------|------|--------|
| C:\Clock (32 位) (4) | | | 9.9% | 227.2 MB | 0.1 MB/s | 0.9 Mbps | 0% | |
| Google Chrome (8) | | | 0.1% | 112.1 MB | 0.1 MB/s | 0 Mbps | 0% | |
| QQ音乐 - 让音乐充满生活 (32 ...) | | | 3% | 10.0 MB | 0 MB/s | 0 Mbps | 0% | |
| Typora (2) | | | 0.4% | 21.7 MB | 0 MB/s | 0 Mbps | 0% | |
| Windows 资源管理器 (2) | | 10660 | 0.1% | 18.4 MB | 0 MB/s | 0 Mbps | 0% | |
| 桌面 | | 10868 | 0.5% | 168.4 MB | 0 MB/s | 0 Mbps | 0% | |
| 任务管理器 (2) | | | 3.8% | 35.2 MB | 0 MB/s | 0 Mbps | 0% | |
| 腾讯QQ (32 位) (2) | | 10152 | 2.8% | 135.6 MB | 0 MB/s | 0.1 Mbps | 0.1% | |
| 后台进程 (86) | | | | | | | | |
| Activation Licensing Service (...) | | 4740 | 3% | 5.7 MB | 0 MB/s | 0 Mbps | 0% | |
| Antimalware Service Executa... | | 4764 | 0.3% | 152.7 MB | 0 MB/s | 0 Mbps | 0% | |
| AnyDesk (32 位) | | 14492 | 3% | 1.1 MB | 0 MB/s | 0 Mbps | 0% | |

进程就是一个正在运行的“程序”

1.进程在系统中是如何被管理的？（最核心的问题）

管理 = 描述（PCB） + 组织。

PCB：进程控制块，这是一个类/C语言的结构体，一个结构体就对应一个进程

组织：使用一定的数据结构来组织，一种常用的做法就是使用双向链表

注：

查看进程列表,本质上就是遍历操作系统内核中这个链表,并显示其中的属性;

创建一个进程,本质就是创建一个PCB对象,加入到内核的链表中;

销毁一个进程,本质就是把这个PCB对象从内核链表中删除掉

2.PCB中有哪些属性呢？

2.1 pid：一个进程的身份标识，一个机器在同一个时刻，不可能有两个进程的pid相同。

2.1 内存指针：描述了这个进程使用的内存空间在哪个范围（虚拟地址空间）

2.3 文件描述符：描述这个进程都打开了那些文件

系统中打开一个文件，其实就得到了一个“文件描述符”，这个文件描述符就像一个“遥控器”一样,文件数据是在磁盘上的,代码中操作磁盘数据不像操作内存数据那么方便,所以就往往采用这种“遥控器”的方式来操作.(文件描述符是一个整数)

2.4 进程的状态

2.5 进程的优先级

进程的上下文:安排时间表的时候,优先给谁安排

2.6 进程的上下文

进程的上下文:需要记住上次运行到那个指令了,方便下次调度的时候就能够继续从这个位置来运行(存档和读档)

进程的上下文,主要是存储 调度处出CPU之前 **寄存中的信息**(把寄存信息保存到内存中),等到这个进程下次恢复到 CPU 上执行的时候,就把内存保存好的数据恢复到寄存中

2.7 进程的记账信息

进程的记账信息:记录这个进程在CPU上**执行了多久了**,用来辅助决定这个进程是继续在CPU上执行,还是说调度出去了

1.以上的四个属性就是用来辅助进行进程调度的!

一个机器上有上百个进程在运行,而这个机器只有一个CPU(8 core CPU),多核的意思就是把多个CPU打包在一起了,8 core就相当于CPU有8 个分身.

2.多任务的系统其实就是基于:"进程调度"这样的机制来实现的

3.操作系统管理进程的方式:

并发式执行:由于CPU的运行速度极快,虽然CPU在一直进行切换,但是电脑用户,是感知不到切换的过程的

并行式执行:多个CPU,运行多个程序.CPU1运行进程1,CPU2运行进程2,进程1和进程2不论是宏观和微观,都是同时执行的

真实的计算机,真实的操作系统在进行进程调度的时候是并行并发两种策略综合使用的

3、进程控制块抽象 (PCB Process Control Block)

计算机内部要管理任何现实事物,都需要将其抽象成一组有关联的、互为一体的数据。在 Java 语言中,可以通过类/对象来描述这一特征。

```
1 // 以下代码是 Java 代码的伪码形式,重在说明,无法直接运行
2 class PCB {
3   // 进程的唯一标识 — pid;
4   // 进程关联的程序信息,例如哪个程序,加载到内存中的区域等
5   // 分配给该资源使用的各个资源
6   // 进度调度信息(留待下面讲解)
```

每一个 PCB 对象，就代表着一个实实在在运行着的程序，也就是进程。

操作系统再通过这种数据结构，例如线性表、搜索树等将 PCB 对象组织起来，方便管理时进行增删查改的操作。

4、CPU分配——进程调度（Process Scheduling）

操作系统对CPU资源的分配，采用的是**时间模式**——不同的进程在不同的时间段去使用 CPU 资源。

5、内存分配—内存管理（Memory Manage）

操作系统对内存资源的分配，采用的是**空间模式**——不同进程使用内存中的不同区域，互相之间不会干扰。

一个进程要想运行,就需要给它分配一些系统资源,其中内存就是一个很重要的资源

虚拟地址空间的作用:

- 1.一定程度的减少内存访问越界,让进程和进程之间隔开,
- 2.尽量不互相影响,这对于系统的稳定性来说,是至关重要的

6、进程间通信（Inter Process Communication）

如上所述，进程是操作系统进行资源分配的最小单位，这意味着各个进程互相之间是无法感受到对方存在的，这就是操作系统抽象出进程这一概念的初衷，这样便带来了进程之间互相具备“隔离性（Isolation）”。

但现代的应用，要完成一个复杂的业务需求，往往无法通过一个进程独立完成，总是需要进程和进程进行配合地达到应用的目的，如此，进程之间就需要有进行“信息交换”的需求。进程间通信的需求就应运而生。

目前，主流操作系统提供的进程通信机制有如下：

1. 管道
2. 共享内存
3. 文件
4. 网络
5. 信号量
6. 信号

其中，网络是一种相对特殊的 IPC 机制，它除了支持同主机两个进程间通信，还支持同一网络内部非同一主机上的进程间进行通信。

