

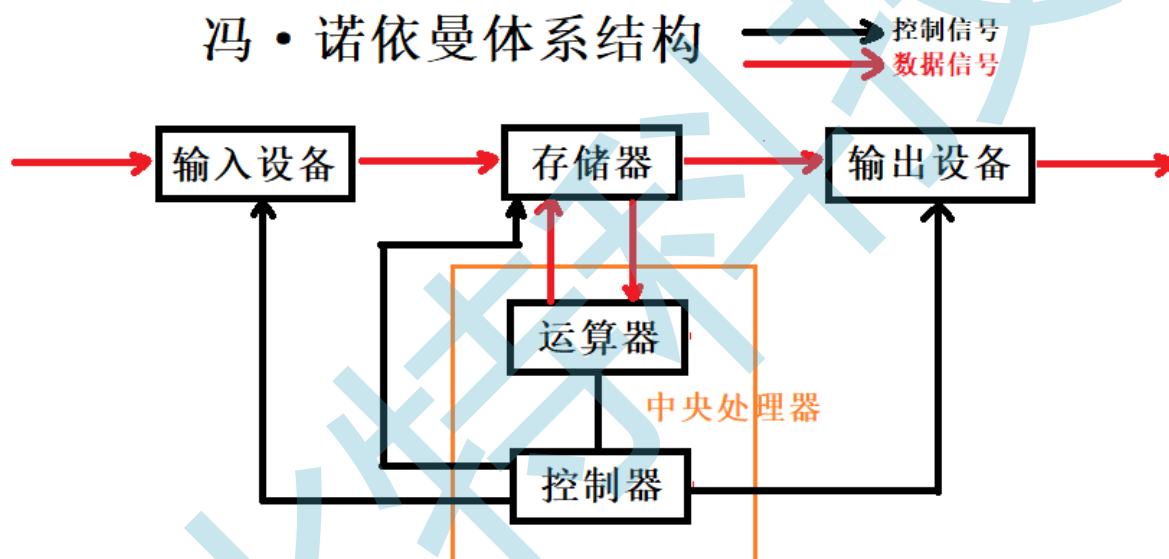
操作系统

本节目标

- 冯诺依曼体系结构
- 操作系统的理解
- 进程
- 线程

1. 冯诺依曼体系结构

我们常见的计算机，如笔记本。我们不常见的计算机，如服务器，大部分都遵守冯诺依曼体系。



https://blog.csdn.net/waves_K

截至目前，我们所认识的计算机，都是有一个个的硬件组件组成

- 输入单元：包括键盘，鼠标，扫描仪，写板等
- 中央处理器（CPU）：含有运算器和控制器等
- 输出单元：显示器，打印机等

关于冯诺依曼，必须强调几点：

- 这里的存储器指的是内存
- 不考虑缓存情况，这里的CPU能且只能对内存进行读写，不能访问外设（输入或输出设备）
- 外设（输入或输出设备）要输入或者输出数据，也只能写入内存或者从内存中读取。
- 一句话，所有设备都只能直接和内存打交道。

2. 操作系统（Operator System）

2.1 概念

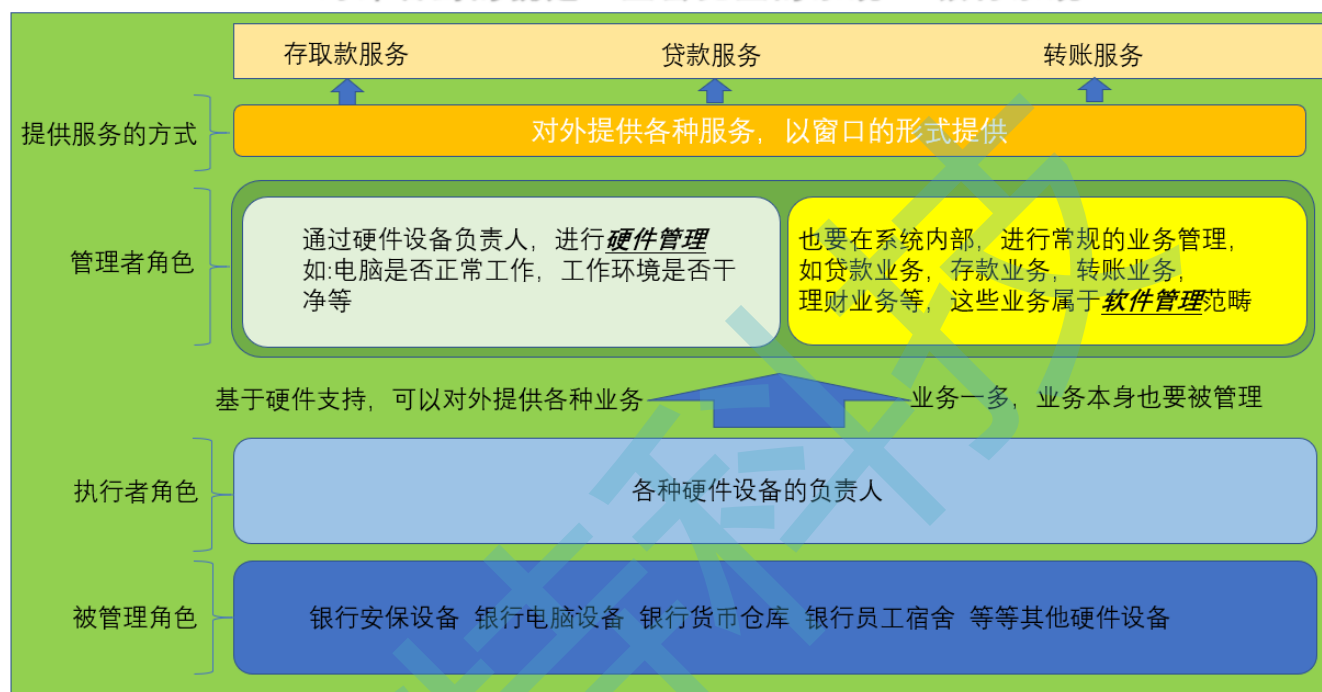
任何计算机系统都包含一个基本的程序集合，称为操作系统（OS）。笼统的理解，操作系统包括：

- 内核（进程管理，内存管理，文件管理，驱动管理）
- 其他程序（例如函数库，shell程序等等）

2.2 OS定位

- 在整个计算机软硬件架构中，操作系统的定位是：一款纯正的“搞管理”的软件
- 如何理解“管理”

对外看到的就是一整套完整的系统 - 银行系统

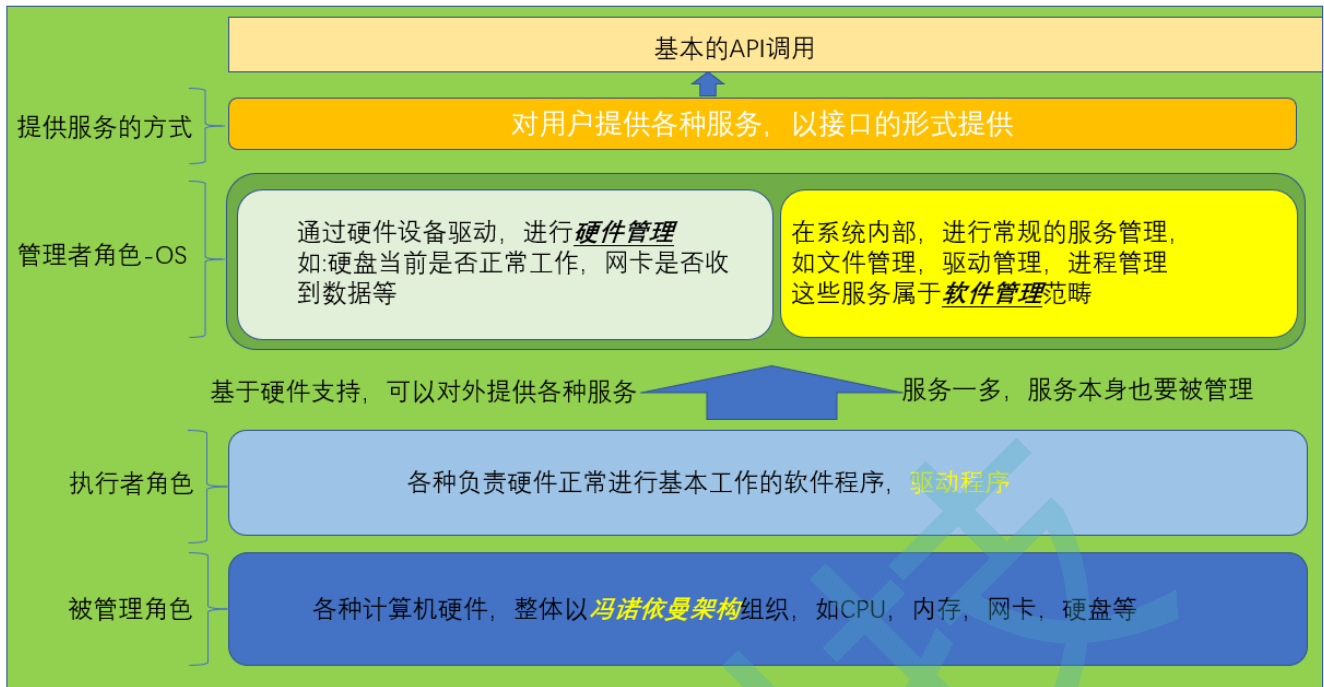


上面的样例，能看到如下结论：

1. 管理者的任务按照种类分两种：硬型基础设备管理和软性业务管理
2. 按照目标对象分两种：一种对内的管理，一种对外业务进行管理

而操作系统，也是一套系统，理解思路于此类似，我们可以类比来看

对看到的就是一整套完整的系统 - 操作系统



上面有关OS的样例，我们也能看到如下结论：

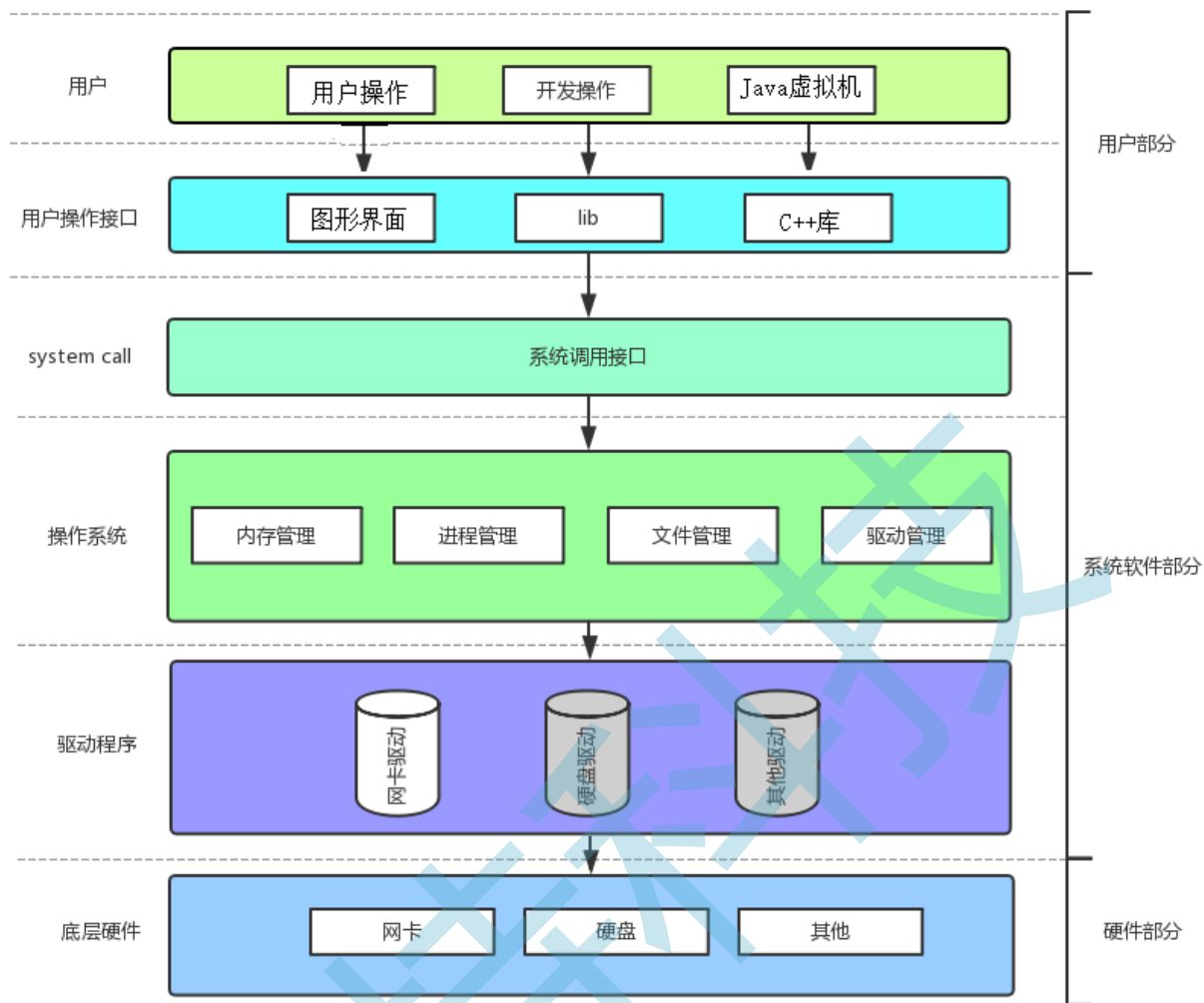
1. 管理者的任务种类分两种：硬件管理软件管理
2. 按照目标对象分两种：一种对内的管理，一种对外业务进行管理

2.3 设计OS的目的

那么，设计操作系统的目的是什么呢？类比银行系统，我们就能理解

- 与硬件交互，管理所有的软硬件资源（对内且对下）
- 为用户程序（应用程序）提供一个良好的执行环境（对外且对上）

那我们的所学的java在哪里呢？



3. 进程

3.1 进程的初步认识

对于操作系统来说，一个任务就是一个进程（Process）。

- 课本概念：程序的一个执行实例，正在执行的程序等。
- 内核观点：担当分配系统资源（CPU时间，内存）的实体。
- 如何理解进程？

如：张三去银行转账，就要使用银行的资源（接待，转账人员，银行电脑，必要时保安也可以跟着:））为张三完成转账,而在银行内部，这一整套过程的执行，我们统称为办理业务。

类比，一个程序刚开始就是一个在硬盘上的程序，加载运行时，是为了完成某些任务（如迅雷完成下载任务），而要完成任务，就需要操作系统为该任务提供足够做的资源（如内存和CPU资源），而这一整套任务的执行，我们可以叫做进程

所以：进程是担当分配系统资源（CPU时间，内存）的实体，是具有动态特性的，如张三在办理转账业务期间，这样任务一直都是存在且状态变更的。

而张三去银行转账也不是立马就能进行，需要先抽号，在大厅等待，因为当时可能会有很多人在大厅等待，所以就有了进程等待队列。而张三抽到的号，可以称之为进程ID

下面，我们尝试查看系统进程

编写一个java进程：

```
public class Test {  
    public static void main(String[] args) throws Throwable {  
        Thread.sleep(99999999);  
    }  
}
```

在cmd中编译并运行：

```
javac Test.java  
java Test
```

运行之后会启动java程序（进程），可以在任务管理器中查看到该进程：



任务管理器							
文件(F) 选项(O) 查看(V)							
进程 性能 应用历史记录 启动 用户 详细信息 服务							
名称	PID	状态	用户名	CPU	内存(专用...)	描述	
InputPersonalizatio...	12944	正在运行	Frank	00	3,024 K	输入个性化服务器	
IntelCpHDCPSvc.exe	4220	正在运行	SYSTEM	00	1,276 K	Intel HD Graphics Driv...	
IntelCpHeciSvc.exe	4948	正在运行	SYSTEM	00	1,304 K	IntelCpHeciSvc Execut...	
java.exe	21148	正在运行	Frank	00	9,348 K	Java(TM) Platform SE b...	
jhi_service.exe	1352	正在运行	SYSTEM	00	924 K	Intel(R) Dynamic Appli...	
LCD_Service.exe	4308	正在运行	SYSTEM	00	10,640 K	LCD_Service	
LdsLite.exe	23320	正在运行	Frank	00	6,964 K	鲁大师Lite版	
lsass.exe	808	正在运行	SYSTEM	00	7,292 K	Local Security Authorit...	
MateBookService....	4316	正在运行	SYSTEM	00	6,312 K	MateBookService.exe	
MSASCuiL.exe	8312	正在运行	Frank	00	1,512 K	Windows Defender no...	

这里的PID代表这个进程的代号，而我们传说中的java虚拟机，本质上只是一个系统中的进程而已。具体可以在看看之前的样例图

3.2 时间片：

现代操作系统比如Mac OS X，UNIX，Linux，Windows等，都是支持“多任务”的操作系统。所谓的多任务，就是操作系统可以同时运行多个任务。

操作系统（如Windows、Linux）的任务调度是采用**时间片轮转**的抢占式调度方式，也就是说一个任务执行一小段时间后强制暂停去执行下一个任务，每个任务轮流执行。

任务执行的一小段时间叫做**时间片**，任务正在执行时的状态叫**运行状态**，任务执行一段时间后强制暂停去执行下一个任务，被暂停的任务就处于**就绪状态**等待下一个属于它的时间片的到来。

这样每个任务都能得到执行，由于CPU的执行效率非常高，时间片非常短，在各个任务之间快速地切换，给人的感觉就是多个任务在“同时进行”，这也就是我们所说的**并发**。

如何理解？

假设张三准备办理转账业务，但当他把所有资料给工作人员之后，工作人员告诉他，你现在办理不了，因为他现在需要填写一张申请表，此时，工作人员将他的资料保存起来，让他去一边填写资料，填写完毕之后，再回来继续办理，同时，张三去填表了，而工作人员继续给别人提供服务。这个过程叫做进程切换。

张三表填完了，继续回到柜台，工作人员拿出他之前的资料，继续给张三办理业务，这叫做进程的上下文保护与恢复（想想，为什么要这么做？因为进程的运行是在CPU上的，CPU有寄存器，保存的是进程运行的各种临时数据，为了达到切换和便于恢复的目的，就有了将CPU寄存器保存和恢复的做法，归根结底是为了接着上次的位置继续运行）

后来，银行出台了规定，每个人在柜台办理任务的时间不能超过10分钟（以防止其他人长时间等待），所以为了更好的服务各个人员，银行工作人员将上面的切换与恢复的思路应用到各种业务中，所以长期来看，即便只有一个工作人员，也能同时服务多个客户，这种机制叫做基于时间片的进程轮转管理机制，而上面的10分钟，就是银行轮转的时间片，只要时间到了，客户就得下去等待，让其他用户来办理业务

而上面的所有轮转与管理工作，在操作系统中，统一由一个叫做调度器的内核模块完成

而对每个人来说，在一段时间之内，可能所有人的业务都得以推进（即便没完成），而不至于大家长时间等待，这种机制就叫做并发

如果银行财大气粗，工作人员比客户都多，那就好办了，一人一个工作人员，所有的任务真正同时处理，这种机制叫做并行。

下面，我们来看看并发和并行的详细概念

3.3 并发与并行

现在，多核CPU已经非常普及了，由于任务数量远远多于CPU的核心数量，所以，操作系统也会自动把很多任务轮流调度到每个核心上执行。对于

- 并发：多个进程在一个CPU下采用时间片轮转的方式，在一段时间之内，让多个进程都得以推进，称之为并发。
- 并行：多个进程在多个CPU下分别，同时进行运行，这称之为并行。

计算机操作系统中把并行性和并发性明显区分开，主要是从微观的角度来说的，具体是指进程的并行性（多处理机的情况下，多个进程同时运行）和并发性（单处理机的情况下，多个进程在同一时间间隔运行的）。

并发与并行类似于工厂中的流水线，要扩大产量，1是考虑建造多个工厂，这就是并行，2是考虑每个工厂中新增流水线，这就类似并发。

3.4 内核态与用户态

一般的操作系统（如Windows、Linux）对执行权限进行分级：用户态和内核态。

- 如上图所示，操作系统内核作为直接控制硬件设备的底层软件，权限最高，称为**内核态**，或**核心态**。
- 用户程序的权限最低，称为**用户态**。

如何理解？

就好比上面的例子，张三去填表，自己写姓名，电话，邮箱等等，做着自己的事情，这叫做用户态

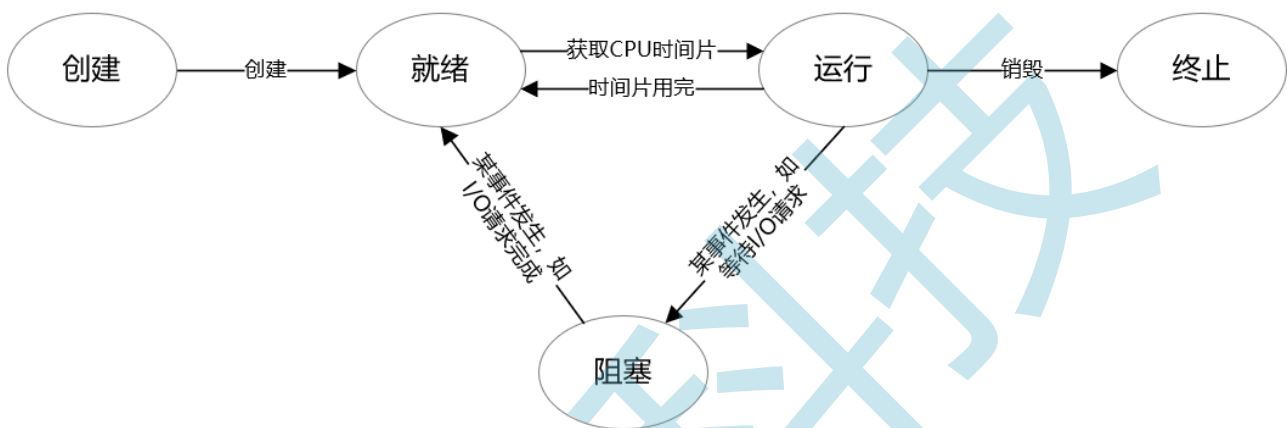
而张三通过窗口的工作人员，把自己的需求给工作人员，自此，张三在等，银行工作人员在忙，对张三来讲，就叫做陷入内核。那么内核态是什么意思？就是工作人员在帮你办理业务时的状态

3.5 进程中的上下文

上下文简单说来就是一个环境，进程在时间片轮转切换时，由于每个进程运行环境不同，就涉及到转换前后的上下文的切换

- 就是一个进程在执行的时候，CPU的所有寄存器中的值、进程的状态以及堆栈上的内容。
- 切换时需要保存当前进程的所有状态，即保存当前进程的进程上下文，以便再次执行该进程时，能够恢复切换时的状态，继续执行。

3.5 进程状态



就绪：进程处于可运行的状态，只是CPU时间片还没有轮转到该进程，则该进程处于就绪状态。

运行：进程处于可运行的状态，且CPU时间片轮转到该进程，该进程正在执行代码，则该进程处于运行状态。

阻塞：进程不具备运行条件，正在等待某个事件的完成。

可以对比张三去银行办理业务的状态

4. 重点内容总结

1. 站在硬件角度，理解冯诺依曼体系结构
2. 理解操作系统的作用，和对“管理”的理解
3. 理解进程的概念，理解进程相关内容，如时间片，切换，调度，并行，并发，内核态用户态等后期java需要理解的概念