**我的系统架构设计师之路**



**系统架构师的定义**

系统架构师是一个最终确认和评估系统需求，给出开发规范，搭建系统实现的核心构架，并澄清技术细节、扫清主要难点的技术人员。主要着眼于系统的“技术实现”。因此他/她应该是特定的开发平台、语言、工具的大师，对常见应用场景能马上给出最恰当的解决方案，同时要对所属的开发团队有足够的了解，能够评估自己的团队实现特定的功能需求需要的代价。 系统架构师负责设计系统整体架构，从需求到设计的每个细节都要考虑到，把握整个项目，使设计的项目尽量效率高，开发容易，维护方便，升级简单等。

**系统架构师技术素质**

* 一线大中型开发项目的总体规划、方案设计以及技术团队管理经验。
* 具备软件行业工作经验，熟悉业务领域的技术和发展。
* 具有项目管理理论基础，并在应用系统开发平台和项目管理上有实战经验。
* 具有面向对象分析(OOA)，设计(OOD),开发(OOP)能力，精通UML和XML，熟练使用Rational Rose,PowerDesigner等Case工具。
* 精通大型数据库
* 对计算机系统，网络安全，应用系统架构有全面认识
* 良好的团队意识和协作精神，有较强的内外沟通能力。

**系统架构师管理素质**

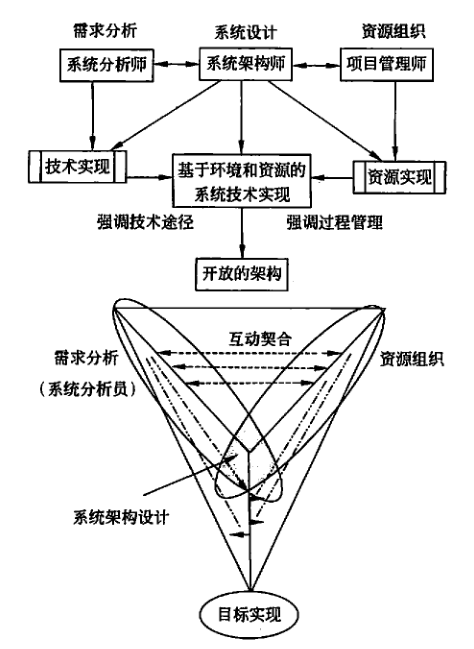
系统架构师的管理素质包括：远见，诚信，果断等，他是一个高效工作团队的建造者，作为核心的高层技术管理人员，必须要为团队制定一个统一的项目计划以及技术解决方案，而且还要在做决定的时候，果断，明确。

**系统架构师与其他团队角色的协调**

项目管理师：负责整个项目的实施和控制，协调各种资源。

系统分析师：获取并分析用户的需求，形成规范化的文档，指导整个项目的开发，需要与客户不断的交流，熟悉应用领域的业务。

系统架构师：负责设计系统整体架构，从需求到设计的每个细节都要考虑到，把握整个项目，使设计的项目尽量效率高，开发容易，维护方便，升级简单等等。



**系统架构师与项目经理关系**

项目经理配合系统架构师提供各个方面的支持，以及与外部沟通，系统架构师则是负责系统的总体架构。

**系统架构师与系统分析师的关系**

系统分析师对业务系统进行分析，建模，系统架构师协同分析师分析系统，给系统分析师提供标准，工具，模式，技术。同时系统分析师给系统架构师整理好需求，让系统架构师更加专注系统架构。

## 操作系统（本章每次约3题）

操作系统是计算机系统中的核心系统软件，负责管理和控制计算机系统中硬件和软件资源，合理组织计算机工作流程和有效利用资源，在计算机与用户之间起接口的作用。

操作系统的类型（依据使用环境和对作业的处理方式）分为批处理、分时、实时、网络、分布式、嵌入式和微内核等。

1. **批处理操作系统（Batch Processing Operating System，BPOS）**：用户将一批作业提交给操作系统后就不再干预，由操作系统控制它们自动运行。批处理操作系统不具有交互性，它是为了提高CPU的利用率而提出的一种操作系统。系统资源的利用率和作业的吞吐量是多道批处理操作系统的主要设计目标，同时也要兼顾作业的周转时间。

多道批处理系统有两个特点：

* 多道：系统内可同时容纳多个作业。
* 成批：系统能成批地自动运行多个作业，在运行过程中不允许用户与其作业发生交互作用。

1. **分时操作系统（Time Share Operating System，TSOS）**：采用分时技术，将处理机时间与内存空间按一定的时间间隔，轮流地切换给各终端用户的程序使用。由于时间间隔很短，每个用户的感觉就像他独占计算机一样。例如UNIX系统就采用剥夺式动态优先的CPU调度，有力地支持分时操作。特征有交互性、多用户同时性、独立性和及时性等。
2. **实时操作系统（Real Time Operating System，RTOS）**：保证在一定时间限制内完成特定功能的操作系统。实时操作系统有硬实时和软实时之分，硬实时要求在规定的时间内必须完成操作，这是在操作系统设计时保证的；软实时则只要按照任务的优先级，尽可能快地完成操作即可。实时系统包括实时过程控制系统（生产过程控制等）和实时信息处理系统（实验数据采集等）。并不强调资源利用率，更关心及时性、可靠性和完整性。
3. **网络操作系统（Network Operating System，NOS）**：网络操作系统是使网络上各计算机能方便而有效地共享网络资源以及为网络用户提供所需的各种服务的软件和有关规程的集合。它提供的服务：文件服务、打印服务、数据库服务、通信服务、信息服务、分布式服务、名字服务、网络管理服务、Internet与Intranet服务等。我们常见的Windows Server2012就属于网络操作系统。
4. **分布式操作系统（Distributed Operating System，DOS）**：分布式计算机系统是由多台计算机组成并满足下列条件的系统：系统中任意两台计算机通过通信方式交换信息；系统中的每一台计算机都具有同等的地位，即没有主机也没有从机； 每台计算机上的资源为所有用户共享；系统中的任意若干台计算机都可以构成一个子系统，并且还能重构；任何工作都可以分布在几台计算机上，由它们并行工作、协同完成。用于管理分布式计算机系统的操作系统称为分布式计算机系统。它是具有高度并行性、故障检测和重构能力的一种高级软件系统。
5. **嵌入式操作系统（Embedded Operating System，EOS）**：将操作系统和功能软件集成于计算机硬件系统之中。嵌入式操作系统多数也是实时操作系统。
6. **微内核操作系统（Microkernel Operating System，MOS）**：由于它能有效地支持多处理机运行，故非常适用于分布式系统环境。当前比较流行的、能支持多处理机运行的操作系统，几乎全部都采用了微内核结构，如当前广泛使用的Windows操作系统，就采用了微内核结构。Linux系统属于单内核，不是微内核。微内核技术的优点：（1）统一的接口；（2）可伸缩性好；（3）可移植性好；（4）实时性好；（5）安全可靠性高，安全是微内核的特性；（6）支持分布式系统、支持多处理器的架构和高度并行的应用程序；（7）真正面向对象的操作系统。

操作系统的5项基本功能，包括进程管理（又称处理机管理）、存储管理、设备管理、文件管理和作业管理。

### 进程管理

进程(process)是资源分配和对立运行的基本单位，进程就是一个程序从开始到结束的一个完整的生命周期，进程就像一艘船，程序就是游客，当游客要过河买票了，就创建了一个进程，船在向对岸行驶的过程中就是进程运行周期，靠岸了游客下船了，那就拜拜了，进程被注销了。

#### 进程状态

进程状态分为三态模型和五态模型。

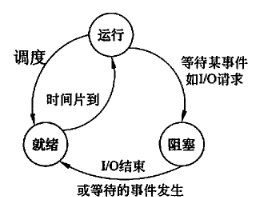
**三态模型**：包括运行、就绪和阻塞。

运行态🡪就绪态：运行时间片到；出现更高优先权进程。

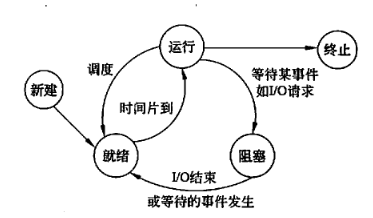
运行态🡪阻塞态：等待使用资源，如等待外设传输；等待人工干预。

就绪态🡪运行态：CPU空闲时调度选择一个就绪进程。

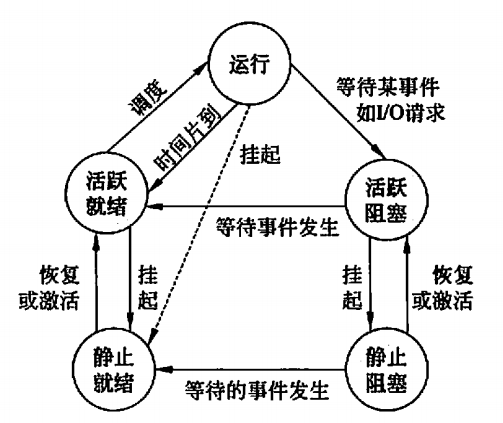
阻塞态🡪就绪态：资源得到满足，如外设传输结束；人工干预完成。



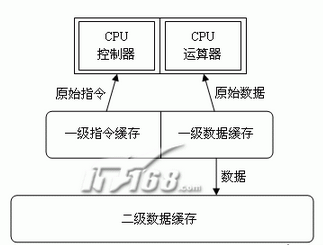
**五态模型**：对于一个实际的系统，进程的状态及其转换更为复杂。引入新建态和终止态构成了进程的五态模型。



具有挂起状态的进程状态转换如下图所示：



进程了解一下后，看到了进程的5态模型，看到了这个模型，让我有一种豁然开朗的感觉，以前老是随口所说的 CPU的一级，二级缓存原来基本原理是这样的。



二级缓存就是进程模型中的“静止就绪”、“静止阻塞”两种状态，就是将优先级较低的，或者需要等待的进程先暂时挪移到二级缓存中。一级缓存就是进程模型中的“活跃就绪”、“活跃阻塞”，就是优先级高的，或者已经具备运行条件等待执行的。而运行就是CPU的运行过程。 这也说明了一些事情，为什么CPU ，磁盘等等二级缓存越大就越贵了，因为缓存越大，可供中间等待的任务就越多，那么任务间的切换就越快，速度就越快了。

#### 进程互斥与同步

互斥原则

* 有空让进
* 忙则等待
* 多种选一
* 有限等待
* 让权等待

信号量：semaphore[英][ˈseməfɔ:(r)] 它负责协调各个线程, 以保证它们能够正确、合理的使用公共资源。

P原语： Passeren(占有)，阻塞原语，负责把当前进程由运行状态转换为阻塞状态，操作为：申请一个空闲资源（把信号量减1），若成功，则退出；若失败，则该进程被阻塞；

V原语： V是荷兰语Vrijgeven（释放）的首字母。为唤醒原语，负责把一个被阻塞的进程唤醒，它有一个参数表，存放着等待被唤醒的进程信息。操作为：释放一个被占用的资源（把信号量加1），如果发现有被阻塞的进程，则选择一个唤醒之。

在PV原语执行期间不允许有中断的发生，根本原理是机器指令。

信号量包含两类，一类是公用信号量，他实现进程间的互斥，初值为1或资源数目；另一类是私用信号量，它实现进程间的同步，初值为0或某个正数。

PV操作是对信号量的操作。

P是给信号量减1、V是给信号量+1。

P操作的是自己的私有信号量、V是操作别人的私有信号量。公有信号量每个进程的PV都能操作。

#### 前趋图

#### 进程调度

调度算法

* 先来先服务
* 优先数调度
* 轮转法

#### 死锁

死锁是指多个进程之间互相等待对方的资源，而在得到对方资源之前又不释放自己的资源，这样，造成循环等待的一种现象。

1、死锁发生的必要条件

产生死锁的根本原因在于系统提供的资源个数少于并发进程要求的该类资源数。

2、银行家算法

指在分配资源之前先看清楚，资源分配后是否会导致系统死锁。如果会死锁，则不分配，否则就分配。

3、解决死锁的策略

（1）死锁预防。破坏导致死锁必要条件中的任意一个就可以预防死锁。

（2）死锁避免。指进程在每次申请资源时判断这些操作是否安全。

（3）死锁检测。判断系统是否处于死锁状态，如果是，则执行死锁解除策略。

（4）死锁解除。将某进程所拥有的资源强行收回，分配给其他的进程。

#### 线程

线程可共享进程的资源与地址空间，通过线程的活动，进程可以提供多种服务或实行子任务并行。多线程实现的并行避免了进程间并行的缺点：创建线程的开销比创建进程要小，同一进程的线程共享进程的地址空间。

### 存储管理

#### 存储管理的方式

##### 分区存储管理

##### 页式存储管理

##### 段式存储管理

##### 段页式存储管理