

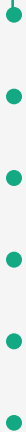


第五讲

操作系统与计算机网络 重点问题盘点

主讲人：Jesse

2017-09-27



1

CHAPTER1

操作系统重点问题盘点

2

CHAPTER2

计算机网络重点问题盘点

CHAPTER 1

操作系统重点问题盘点

操作系统

操作系统概述

操作系统：指控制管理整个计算机系统的软硬件资源，合理组织调度计算机的工作和资源分配，以提供给用户和其他软件方便的接口和环境的程序集合

操作系统特性：并发、共享、虚拟、异步

主要功能：处理器管理、存储器管理、文件管理、设备管理、用户接口

发展：手工操作-->批处理-->分时操作系统-->实时操作系统-->网络/分布式系统-->个人操作系统

运行机制：时钟管理、中断机制、原语、系统控制的数据结构及处理

中断和异常：中断指与当前处理机处理的程序无关，并即刻占用处理机进行处理的事件；异常源自于CPU执行指令内部的事件，一旦出现立即处理（如算数溢出、越界、缺页等）

系统调用：运行在核心态的特殊公共子程序，主要用于：设备、文件、进行、内存等管理

阅读材料 <http://blog.csdn.net/bigpudding24/article/details/48603301>

操作系统

操作系统概述

下列说法正确的是：

分时系统不一定具有人机交互功能

从响应时间角度看，分时系统与实时系统无差别

分时技术让用户独占计算机资源

批处理系统应具有作业控制功能

操作系统

操作系统概述

下述哪种情况会提出中断请求（）

- A. 在键盘输入过程中，每按一次键
- B. 两数相加结果为零
- C. 计算结果溢出
- D. 一条系统汇编指令执行完成

操作系统

进程和线程区别

进程：是进程实体运行的过程，是系统进行资源分配和调度的一个独立单位。使多个程序可以并发的执行，提高系统的资源利用率。特性：动态、并发、独立、异步；状态：就绪、执行、阻塞

线程：进程的一个实体，CPU调度和分配的基本单位，比进程更小且能独立运行，线程不直接拥有系统资源，线程间共享进程的系统资源。

联系：都能够并发执行；进程可以创建进程和线程；一个进程至少有一个线程；线程依赖于进程

区别：1. 线程是更小的调度单位，是CPU调度的基本单位，具有更高的并发性
2. 进程持有系统资源，而线程基本不拥有系统资源（除程序必须的栈、程序指针）。线程可以访问进程的资源。线程间可以共享内存资源
3. 进程创建/销毁/切换，需要创建/回收PCB、系统资源，保存/恢复CPU环境，而线程仅需要保存/恢复少量寄存器，代价更小

阅读材料 <http://www.cnblogs.com/fuchongjundream/p/3829508.html>

进程和线程区别

一个进程中，线程不共享的是：

信号

堆

文件描述符

进程组id

代码段

栈空间

操作系统

进程任务调度

时间片轮转调度算法 (RR)：给每个进程固定的执行时间，根据一定的先后顺序让进程在单位时间片内轮流执行，不考虑进程等待时间和执行时间，属于抢占式，适合分时系统；

先来先服务调度算法 (FCFS)：根据进程到达的先后顺序执行，不考虑等待时间和执行时间，会产生饥饿现象，属于非抢占式调度，不利于短作业；

优先级调度算法 (HPF)：在进程等待队列中选择优先级最高的来执行；

多级反馈队列调度算法：将时间片轮转与优先级调度相结合，把进程按优先级分成不同的队列，先按优先级调度，优先级相同的，按时间片轮转；兼顾长短作业，有较好的响应时间

高响应比优先调度算法：响应比 = (进程执行时间 + 进程等待时间) / 进程执行时间；作业执行的时间越短，响应比越高；响应比会随着等待时间增加而变大；能够避免饥饿现象；计算响应比开销大

阅读材料 http://blog.csdn.net/leex_brave/article/details/51638300

操作系统

进程任务调度

以下CPU调度算法中，会引起饥饿的是：

FCFS

SJF

高响应比优先

R

进程任务调度

假设 4 个作业如下表所示同时到达，当使用最高优先权优先调度算法时（优先级数值越大优先级越高），作业的平均周转时间为（ ）小时。

作业号	执行时间（小时）	优先级
1	2	4
2	5	9
3	8	1
4	3	8

操作系统

进程间通信

管道：有名管道和无名管道；无名管道是半双工通信，只能在具有亲缘关系（父子关系）的进程间使用；有名管道也是一种半双工通信，运行无亲缘关系的进程通信

信号量：一种计数器，控制多个线程对共享资源的访问；

信号：一种比较复杂的通信方式，用于通知接收进程某个事件的发生

消息队列：消息队列是消息的链表，存放在内核中并由消息队列标识符标识。消息队列是UNIX中进程之间实现共享资源的一种机制，允许进程将格式化的数据流以消息队列形式发送给任意进程

共享内存：映射一段能被其他进程所访问的内存，往往与其他通信机制（如信号量）配合使用。是最快的进程间通信方式

套接字：源IP地址和目的IP地址以及源端口号和目的端口号的组合称为套接字，是一套数据通信的机制和接口；还能够通过网络与不同机器之间进行通信

阅读材料 http://blog.csdn.net/violet_echo_0908/article/details/51201278

进程间通信

进程间通讯的方式中哪种的访问速度最快？

☐ 管道

☐ 消息队列

☐ 共享内存

☐ 套接字

操作系统

线程/进程同步

临界区：通过对多线程串行化来访问公共资源或代码。在任意时刻只允许一个线程对共享资源进行访问，一个线程进入后，其他试图访问公共资源的线程将被挂起，临界区被释放后其他线程才能抢占

互斥量：采用互斥对象机制，拥有互斥对象的线程才有访问公共资源的权限。互斥对象只有一个，所能保证公共资源不会同时被多个线程访问。互斥能用于同个进程的线程或者不同进程的线程；

信号量：信号量通过一个计数器控制对共享资源的访问，信号量的值是一个非负整数，所有通过它的线程都会将该整数减一。如果计数器大于0，则访问被允许，计数器减1；如果为0，则访问被禁止，所有试图通过它的线程都将处于等待状态。定义成PV操作原语（P操作申请资源/V操作释放资源）

事件：通过通知操作的方式来保持线程的同步

阅读材料 <http://blog.csdn.net/rjlgo/article/details/44061001>

死锁的条件及处理方式

死锁原因：1.竞争资源：请求同一有限资源的进程数多于可用资源数；2.进程推进顺序非法；

产生必要条件：1.互斥条件：进程对所分配的资源进行排他性的使用；2.请求和保持条件：进程被阻塞的时候并不释放所申请到的资源；3.不可剥夺条件：进程对于已经申请到的资源在使用完成之前不可以被剥夺；4.环路等待条件：发生死锁的时候存在的一个“进程-资源”环形等待链

死锁处理：1.预防死锁：破坏死锁的必要条件，实现简单，会降低系统资源利用率和吞吐量；2.避免死锁：在资源的动态分配中，防止系统进入不安全状态（银行家算法）；3.检测死锁：允许系统运行过程中产生死锁，在死锁发生之后，采用一定的算法进行检测，并确定与死锁相关的资源和进程；4.解除死锁：。对检测到的和死锁相关的进程以及资源，通过撤销或者挂起的方式，释放一些资源并将其分配给处于阻塞状态的进程，使其转变为就绪态

阅读材料 <http://www.cnblogs.com/changyaohua/p/4642526.html>

死锁的条件及处理方式

对资源编号，要求进程按照序号顺序申请资源，是破坏了死锁必要条件中的哪一条？（ ）

☐ 互斥

☐ 请求与保持

☐ 不可剥夺

☐ 循环等待

虚拟内存及页面置换

虚拟内存：具有请求调入功能和置换功能，可以从逻辑上对内存容量加以扩充的一种存储器系统；虚拟存储器都是建立在离散内存管理的基础上。

页面置换：

- 1.先进先出置换算法：每次淘汰最早调入的页面
- 2.LRU算法：记录上次页面被访问到现在经历的时间 t ，把 t 最大的页面置换出去
- 3.LFU算法：设置寄存器记录页面被访问次数，每次置换的时候置换当前访问次数最少的
- 4.时钟算法clock (NRU)：页面设置一个访问位，并将页面链接为一个环形队列，页面被访问的时访问位设为1。页面置换的时，若当前所指页面访问位为0，则置换，否则将其置为0，循环直到遇到一个访问位为0的页面
- 5.改进型clock算法：在clock算法的基础上添加一个修改位，优先替换访问位和修改位都是0的页面
- 6.页面缓冲算法：设置一个缓冲区，将被置换的页面表项放入空闲链表（若未修改），或修改页面链表中（若已修改），积累到一定数量再写入到磁盘

CHAPTER 2

计算机网络重点问题盘点

OSI参考模型

物理层：该层为上层协议提供了一个传输数据的可靠的物理媒体

数据链路层：为网络层提供可靠的数据传输，基本数据单位为帧，基于以太网协议。作用：物理地址寻址、数据的成帧、流量控制、数据的检错、重发

网络层：实现两个端系统之间的数据透明传送。具体功能包括寻址和路由选择、连接的建立、保持和终止等

传输层：负责将上层数据分段并提供端到端的、可靠的或不可靠的传输。此外，传输层还要处理端到端的差错控制和流量控制问题

会话层：管理主机之间的会话进程，即负责建立、管理、终止进程之间的会话。会话层还利用在数据中插入校验点来实现数据的同步

表示层：对上层数据或信息进行变换保证一个主机应用层信息可以被另一个主机的应用程序理解

应用层：为操作系统或网络应用程序提供访问网络服务的接口

计算机网络

典型协议

网络层：IP协议：无连接数据报传输、数据报路由选择和差错控制；地址解析协议ARP；逆地址解析协议RARP；路由选择协议RIP、OSPF；因特网报文协议ICMP；因特网组管理协议IGMP

传输层：UDP, TCP

应用层：FTP（文件传送协议）、Telnet（远程登录协议）、DNS（域名解析协议）、SMTP（邮件传送协议），POP3协议（邮局协议），HTTP协议（Hyper Text Transfer Protocol）

阅读材料 <http://www.cnblogs.com/maybe2030/p/4781555.html>

计算机网络

IP地址

网络地址：IP地址由网络号（包括子网号）和主机号组成，网络地址的主机号为全0，网络地址代表着整个网络

子网掩码：子网掩码是标志两个IP地址是否同属于一个子网的。如果两个IP地址在子网掩码的按位与的计算下所得结果相同，即表明它们共属于同一子网中

广播地址：广播地址与网络地址的主机号正好相反，广播地址中，主机号为全1

几类地址：A类地址：以0开头，第一个字节作为网络号，地址范围为：0.0.0.0~127.255.255.255；子网掩码：255.0.0.0；B类地址：以10开头，前两个字节作为网络号，地址范围是：128.0.0.0~191.255.255.255；子网掩码：255.255.0.0；C类地址：以110开头，前三个字节作为网络号，地址范围是：192.0.0.0~223.255.255.255；子网掩码：255.255.255.0；D类地址：以1110开头，地址范围是224.0.0.0~239.255.255.255，D类地址作为组播地址（一对多的通信）；E类地址以1111开头，为保留地址

ARP/RARP协议

ARP: 主机发送信息时将包含目标IP地址的ARP请求广播到网络上的所有主机，并接收返回消息，以此确定目标的物理地址（MAC地址）。收到返回消息后将该IP地址和物理地址存入本机ARP缓存中并保留一定时间，下次请求时直接查询ARP缓存以节约资源

RARP: 功能和ARP协议相对，其将局域网中某个主机的物理地址转换为IP地址，比如局域网中有一台主机只知道物理地址而不知道IP地址，那么可以通过RARP协议发出征求自身IP地址的广播请求，然后由RARP服务器负责回答。

路由选择协议

RIP: 底层是贝尔曼福特算法，它选择路由的度量标准 (metric) 是跳数，最大跳数是15跳，如果大于15跳，它就会丢弃数据包

OSPF: 开放式最短路径优先，底层是迪杰斯特拉算法，是链路状态路由选择协议，它选择路由的度量标准是带宽，延迟

TCP三次握手、四次挥手

TCP连接建立过程：首先Client端发送连接请求报文，Server端接受连接后回复ACK报文，并为这次连接分配资源。Client端接收到ACK报文后也向Server端发送ACK报文，并分配资源，这样TCP连接就建立了。

TCP连接断开过程：Client端发起中断连接请求，也就是发送FIN报文；Server端接到FIN报文后，先发送ACK，Client端就进入FIN_WAIT状态；Server端确定数据已发送完成，则向Client端发送FIN报文；Client端收到FIN报文后，发送ACK后进入TIME_WAIT状态；Server端收到ACK后，关闭连接；Client端等待了2MSL后依然没有收到回复，则证明Server端已正常关闭，Client端也关闭连接。

TCP重传机制

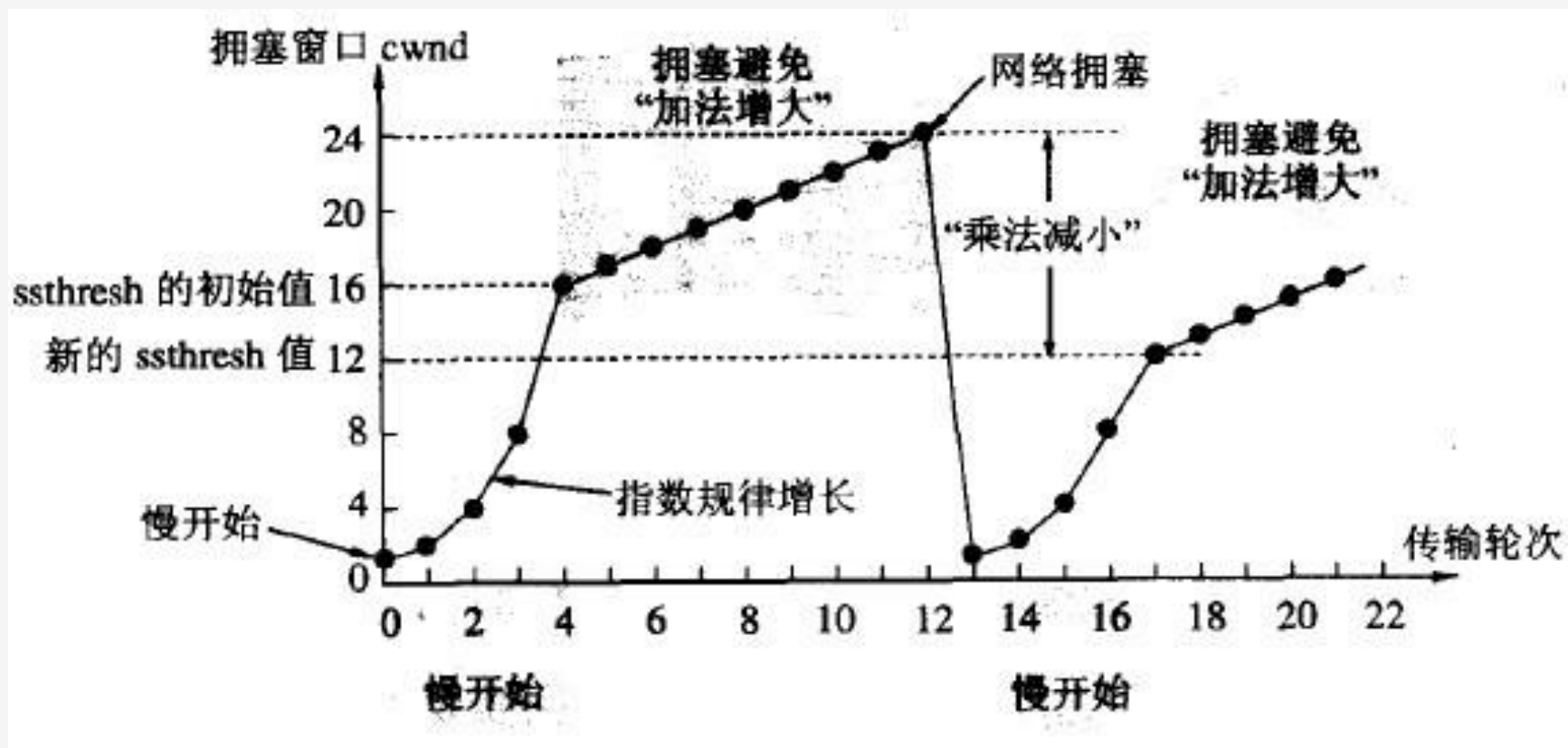
超时重传：请求包发出去的时候，开启一个计时器，当计时器达到时间之后，没有收到ACK，则就进行重发请求的操作，一直重发直到达到重发上限次数或者收到ACK

快速重传：当接收方收到的数据包是不正常的序列号，那么接收方会重复把应该收到的那一条ACK重复发送，这个时候，如果发送方收到连续3条的同一个序列号的ACK，那么就会启动快速重传机制。

TCP拥塞控制

慢开始和拥塞避免：当主机开始发送数据时，由小到大逐渐增大发送窗口；在刚刚开始发送报文段时，先把拥塞窗口 $cwnd$ 设置为一个最大报文段MSS的数值；每收到一个对新的报文段的确认后，把拥塞窗口增加至多一个MSS的数值，因此每经过一个传输轮次，拥塞窗口 $cwnd$ 就加倍；设置一个慢开始门限 $ssthresh$ 状态变量，当 $cwnd > ssthresh$ 时，停止使用慢开始算法而改用拥塞避免算法，让拥塞窗口 $cwnd$ 缓慢地增大，即每经过一个往返时间RTT就把发送方的拥塞窗口 $cwnd$ 加1；无论在慢开始阶段还是在拥塞避免阶段，只要发送方判断网络出现拥塞（丢包），就要把慢开始门限 $ssthresh$ 设置为出现拥塞时的发送方窗口值的一半（但不能小于2）。然后把拥塞窗口 $cwnd$ 重新设置为1，执行慢开始算法。

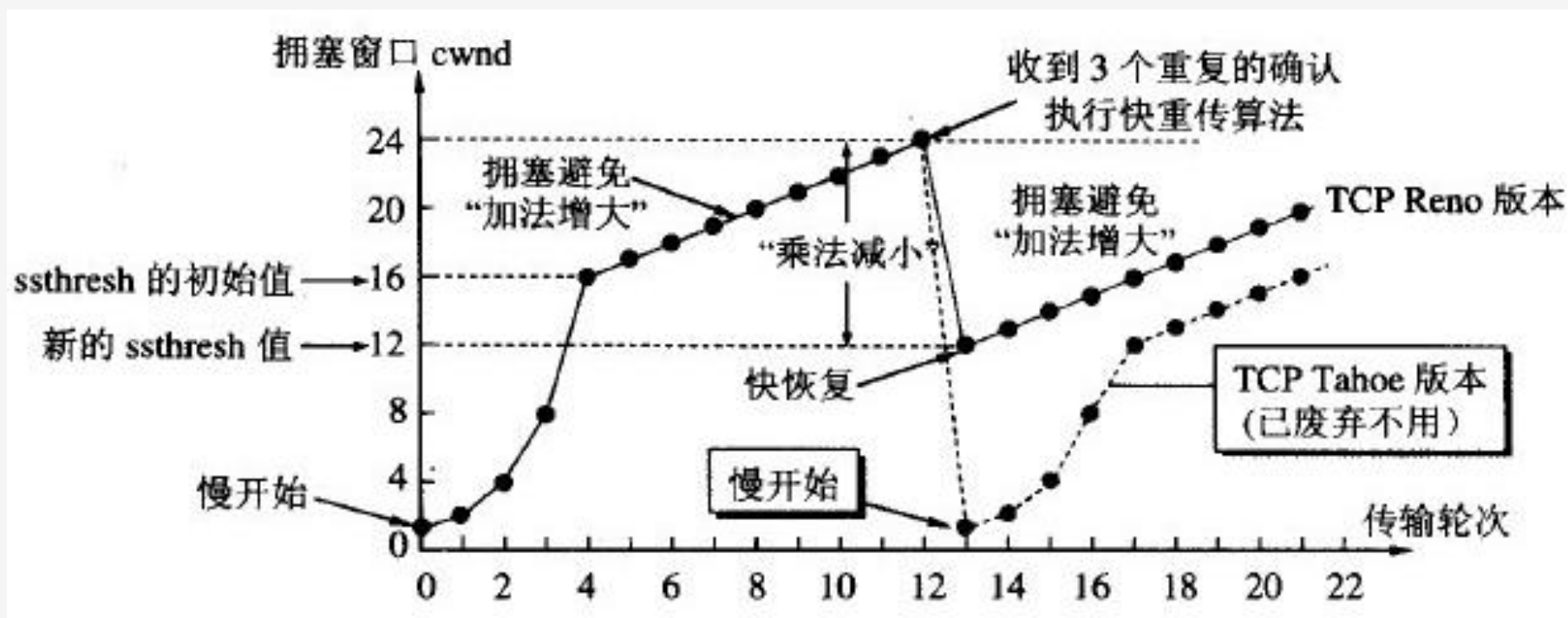
TCP拥塞控制



TCP拥塞控制

快重传和快恢复：快重传算法首先要求接收方每收到一个失序的报文段后就立即发出重复确认，发送方只要一连收到三个重复确认就应当立即重传对方尚未收到的报文段；发送方连续收到三个重复确认，就执行“乘法减小”算法，把慢开始门限ssthresh减半；发送方认为网络很可能没有发生拥塞，把cwnd值设置为慢开始门限ssthresh减半后的数值，然后开始执行拥塞避免算法；

TCP拥塞控制



阅读材料 <http://blog.csdn.net/yechaodechuntian/article/details/25429143>

其他协议

DNS协议：用于命名组织到域层次结构中的计算机和网络服务，可以简单地理解为将URL转换为IP地址；每一个域名都对应一个惟一的IP地址，DNS就是进行域名解析的服务器

DHCP协议：使用UDP协议工作，给内部网络或网络服务供应商自动分配IP地址，给用户或者内部网络管理员作为对所有计算机作中央管理的手段

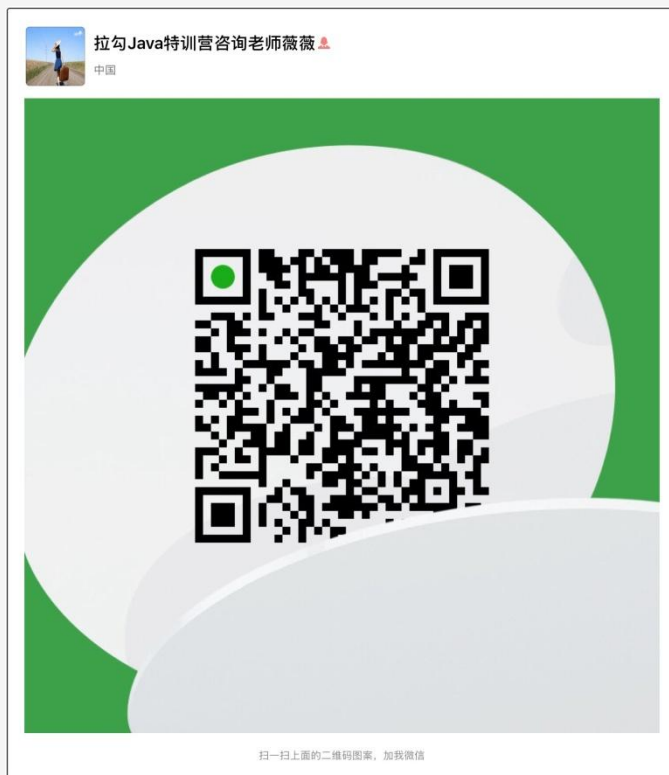
HTTP协议：联网上应用最为广泛的一种网络协议，是基于TCP的应用层协议

GET：请求读取由URL所标志的信息

POST：给服务器添加信息

PUT：在给定的URL下存储一个文档

DELETE：删除给定的URL所标志的资源。



**添加拉勾课程咨询老师薇薇微信，获得更多课程信息；
关注互联网offer之路，获取海量互联网求职干货。**

Thank You

