

# 面试技巧

---

对于一些已经知道的题目，一定要说的很透彻很深入，比如tcp/ip，https，内存管理，这些都是有评分的，不要想当然的说，要有理有据，条理清晰，逻辑缜密。

# 算法

---

☑ LRU的实现(LinkedHashMap/(hashmap+双向链表))

# cpp基础

---

## c语言编译流程

---

### 预处理

预处理用于将所有的#include头文件以及宏定义替换成其真正的内容，预处理之后得到的仍然是文本文件，但文件体积会大很多。gcc的预处理是预处理器cpp来完成的。

### 编译

这里的编译不是指程序从源文件到二进制程序的全部过程，而是指将经过预处理之后的程序转换成特定汇编代码(assembly code)的过程。

### 汇编

汇编过程将上一步的汇编代码转换成机器码(machine code)，这一步产生的文件叫做目标文件，是二进制格式。

### 链接

链接过程将多个目标文以及所需的库文件(.so等)链接成最终的可执行文件(executable file)。

## 静态链接和动态链接的区别

---

- 静态：库的代码和原程序绑定在一起，执行速度快，内存有冗余
- 动态：执行的时候才取找库的代码的逻辑地址，执行速度慢，节省内存

## define和const的区别

---

- define是预处理阶段，不检查类型，不安全
- const是编译阶段，检查类型，相对安全

## 结构体字节对齐

---

参考《后台开发核心技术》相关章节和笔记

## 网络

---

### 拥塞控制

---

- 慢启动
- 拥塞避免
- 快恢复
- 快重传

## url从输入到页面显示的过程

---

域名解析的流程：

1. **查找浏览器缓存**——我们日常浏览网站时，浏览器会缓存DNS记录一段时间。如果以前我们访问过该网站，那么在浏览器中就会有相应的缓存记录。因此，我们输入网址后，浏览器会首先检查缓存中是否有该域名对应的IP信息。如果有，则直接返回该信息供用户访问网站，如果查询失败，会从系统缓存中进行查找。
2. **查找系统缓存**——从hosts文件中查找是否有存储的DNS信息（MAC端，可在“终端”中输入命令cat etc/hosts找到hosts文件位置），如果查询失败，可从路由器缓存中继续查找。
3. **查找路由器缓存**——如果之前访问过相应的网站，一般路由器也会缓存信息。如果查询失败，可继续从ISP DNS缓存查找。查找ISP DNS缓存——从网络服务商（例如电信）的DNS缓存信息中查找。
4. 如果经由以上方式都没找到对应IP的话，则向根域名服务器查找域名对应的IP地址，根域名服务器把请求转发到下一级，逐层查找该域名的对应数据，直到获得最终解析结果或失败的相应。根域名服务器，根服务器主要用来管理互联网的主目录。是互联网域名解析系统（DNS）中最高级别的域名服务器。

## HTTP/HTTPS

---

### HTTP

HTTP协议无论是请求报文还是回应报文，都分为以下4个部分：

1. 报文头(initial line)
2. 0个或多个请求头(header line)
3. 空行(作为header line的结束)
4. 可选的消息体

HTTP协议是基于行的协议，每一行以\r\n作为分隔符。报文头通常表明报文的类型(例如请求类型)，且报文头只占一行；请求头附带一些特殊信息，每个请求头占一行，其格式为name:value，即以分号作为分割；空行也就以一个\r\n分隔；可选body通常包含数据，例如服务器返回的某个静态HTML文件的内容

常见的请求头：

- Host
- Connection(keep-alive和close)
- Accept
- Cache-Control
- Cookie

返回码：

- 1xx：指示信息，表示请求已接收，继续处理
- 2xx：成功，表示请求已被成功接收、理解、接受
- 3xx：重定向(301-永久重定向，302-暂时重定向，304 Not Modified)
- 4xx：客户端错误，请求有语法错误或请求无法实现
- 5xx：服务端错误，服务器未能实现合法的请求

## HTTPS

<http://www.ruanyifeng.com/blog/2014/09/illustration-ssl.html>

HTTP有以下安全性问题：

- 使用明文进行通信，内容可能会被窃听
- 不验证通信方的身份，通信方的身份有可能遭遇伪装
- 无法证明报文的完整性，报文有可能被篡改

对称密钥加密

- 优点：运算速度快
- 缺点：密钥管理复杂(每一个连接都需要保存一个密钥)、无法安全地将密钥传输给通信方

非对称密钥加密

又称公开密钥加密（Public-Key Encryption），加密和解密使用不同的密钥。

公开密钥所有人可以获得，通信发送方获得接收方的公开密钥之后，就可以使用公开密钥进行加密，接收方收到通信内容后使用私有密钥解密。

非对称密钥除了用来加密，还可以用来进行签名。因为私有密钥无法被其他人获取，因此通信发送方使用其私有密钥进行签名，通信接收方使用发送方的公开密钥对签名进行解密，就能判断这个签名是否正确。

- 优点：安全
- 缺点：运算速度慢

SSL协议的握手过程：

1. 客户端给出协议版本号、一个客户端生成的随机数、以及客户端支持的加密方法。
2. 服务器确认双方使用的加密方法、并给出数字证书、以及一个服务器生成的随机数。
3. 客户端确认数字证书有效，然后生成一个新的随机数，并使用数字证书中的公钥加密这个随机数，发给服务器。
4. 服务器使用自己的私钥，获取客户端发来的随机数。
5. 客户端和服务器根据约定的加密方法，使用前面的三个随机数，生成“对话密钥”，用来加密接下来的整个对话过程。

## http1.0/1.1/2.0的区别

---

<https://blog.csdn.net/zhglance/article/details/76162176>

### HTTP 1.1新特性

#### 支持持久链接

一个TCP链接可以传送多个http请求和响应，减少了TCP建立链接和关闭链接的消耗。另外http1.1允许客户端不用等待上一次请求结果返回，就可以发出下一次请求，但服务器端必须按照接收到客户端请求的先后顺序依次回送响应结果，以保证客户端能够区分出每次请求的响应内容。

#### 支持http管道

不使用管道的http请求，在使用持久链接时，必须严格满足先进先出的队列顺序（FIFO），即发送请求，等待响应完成，再发送客户端队列中的下一个请求。管道可以让我们把 FIFO 队列从客户端（请求队列）迁移到服务器（响应队列），即客户端可以并行，服务端串行。客户端可以不用等待前一个请求返回，发送请求，但服务器端必须顺序的返回客户端的请求响应结果。缺点：一个请求响应阻塞，就会阻塞后续所有请求

## 使用多个tcp链接

http1.1 在客户端排队所有请求，让后通过一个TCP持久链接，一个接一个的发送请求（如果有http管道还必须顺序等待服务端的顺序返回结果）。但实际中，浏览器的开发者不会这么笨，浏览器允许我们打开N个TCP链接（大多说浏览器是6个TCP链接，这个数字越大，客户端和服务器的资源占用越多）

## HTTP 2.0新特性

HTTP 2.0把解决性能问题的方案内置在了传输层，通过多路复用来减少延迟，通过压缩 HTTP首部降低开销，同时增加请求优先级和服务器端推送的功能。

### 多路复用

同一时间可以发送多个http请求，只使用一个tcp链接

### 新的二进制格式（Binary Format）

http1.x诞生的时候是明文协议，其格式由三部分组成：start line（request line或者status line），header，body。要识别这3部分就要做协议解析，http1.x的解析是基于文本。基于文本协议的格式解析存在天然缺陷，文本的表现形式有多样性，要做到健壮性考虑的场景必然很多，二进制则不同，只认0和1的组合。基于这种考虑http2.0的协议解析决定采用二进制格式，实现方便且健壮。

### 首部压缩

http 2.0支持DEFLATE和HPACK 算法的压缩

### 服务器推送

指客户端请求之前发送数据的机制，在 HTTP 2.0 中，服务器可以对客户端的一个请求发送多个响应。

# 操作系统

---

## 内存管理

---

### 段页式

参考课本

## 进程和线程

---

### 进程

进程是具有独立功能的程序在某个数据集合上的一次运行活动，也是操作系统进行资源分配和保护的基本单位。进程具有以下属性：

- 动态性。进程是程序在数据集合上的一次执行过程，是动态概念，同时它有生命周期，由创建而生、由调度而执行、由事件而等待、由撤销而消亡；而程序是一组有序指令序列，是静态概念。
- 共享性。同一程序同时运行于不同数据集合上时构成不同进程，即多个不同进程可执行相同的程序，所以进程和程序不是一一对应的关系。
- 独立性。每个进程是操作系统中的一个独立实体，有自己的虚拟内存空间、程序计数器和内部状态。
- 制约性。进程因共享资源或协同工作产生相互制约关系，造成进程执行速度的不可预测性，必须对进程的执行次序或相对执行速度加以协调。
- 并发性。多个进程的执行在时间上可以重叠，在单处理器系统中可以并发执行；在多处理器环境下可以并行执行。

### 线程

线程是进程中能够并发执行的实体，是进程的组成部分，也是处理器调度和分派的基本单位。线程的组成部分有线程唯一标识符及线程状态信息(运行态、就绪态、阻塞态和终止态)。线程是一条执行路径，有独立的程序计数器，未运行时保护线程上下文。线程有执行栈和存放局部变量的私用存储空间。线程可以访问所属进程的内存和资源，并与该进程中的其他线程共享这些资源。

# 进程和线程的区别

## I 拥有资源

进程是资源分配的基本单位，但是线程本身不被单独分配资源，线程可以访问隶属进程的资源。

## II 调度

线程是独立调度的基本单位，在同一进程中，线程的切换不会引起进程切换，从一个进程中的线程切换到另一个进程中的线程时，会引起进程切换。

## III 系统开销

由于创建或撤销进程时，系统都要为之分配或回收资源，如内存空间、I/O 设备等，所付出的开销远大于创建或撤销线程时的开销。类似地，在进行进程切换时，涉及当前执行进程 CPU 环境的保存及新调度进程 CPU 环境的设置，而线程切换时只需保存和设置少量寄存器内容，开销很小。

## IV 通信方面

线程间可以通过直接读写同一进程中的数据进行通信，但是进程通信需要借助 IPC。

# 进程间通信

- 管道(无名管道和有名管道)
- 消息队列
- 共享内存
- 信号量
- 套接字

# 项目

---

## 秒杀系统

---

难点



1. 系统安全性：两次加密、接口防刷
2. 限流：分布式限流
3. Redis扣减库存防止商品超卖
4. 异步下单

## 大数据平台

---

1. Ambari接管线上Hadoop安全数据
2. 实现Ambari多集群管理
3. Ambari汉化
4. HDFS HA

## 场景设计题

---

### 无锁队列的实现

---

<https://coolshell.cn/articles/8239.html>

## 聊人生

---

### 最大的缺点是什么

---

- 1) 喜欢追求细节导致项目 / 作业未能按期完成。通过时间管理能力改变工作方式，先完成框架再改善细节得以解决；
- 2) 不知如何拒绝，同事要求帮忙一概揽下，影响自身工作进度。通过多任务处理能力设定优先顺序，以该优先顺序表向求助同事展示自己手上工作，并给其一个自己在何时可以给予帮助的时间估计，让求助人自行决定是否求助，问题解决。