1. 计算机组成原理 中断
2. http 浏览器输入网址到显示经历了什么

<https://www.cnblogs.com/linyx/p/3985160.html>

<https://www.jianshu.com/p/d3268aa4ae84>

<https://blog.csdn.net/u014527697/article/details/79340386> 比较全面的介绍

1. 计算机网络 tcp三次握手

1客户端 发送 SYN 1 随机产生一个seq=J

2 服务端 发送SYN 1 和 ack=J+1 随机产生一个seq=K

3 客户端 检测ack 是否等于J+1 然后发送 ack=K+1 然后服务端检测ack是否为K+1从而完成握手

1. Hash 碰撞 原理 开放地址法 负载因子的概念 记录数与哈希表的长度的比值

常见的哈希函数构造方法：

1、线性寻址法

2、除留余数法

3、平方取中法

冲突处理方法：

1、开放地址法（再次散列） *H**i*=（*H*（*k**e**y*）+*d**i*）%*m*

线性探测 主要根据di的不同进行划分

二次探测

伪随机探测

2、链地址法 每个桶用一个链表表示

查找困难，插入和删除容易

3、再哈希法：使用多个哈希函数

https://blog.csdn.net/Carol\_1992/article/details/76735656

1. 线程和进程区别和联系：

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位.

线程是进程的一个实体,是CPU调度和分派的基本单位,它是比进程更小的能独立运行的基本单位.线程自己基本上不拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计数器,一组寄存器和栈),但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源.

进程和线程的主要差别在于它们是不同的操作系统资源管理方式。进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程。

1. 同一进程间的线程究竟共享哪些资源呢，而又各自独享哪些资源呢？

共享的资源有

a. 堆 由于堆是在进程空间中开辟出来的，所以它是理所当然地被共享的；因此new出来的都是共享的（16位平台上分全局堆和局部堆，局部堆是独享的）

b. 全局变量 它是与具体某一函数无关的，所以也与特定线程无关；因此也是共享的

c. 静态变量 虽然对于局部变量来说，它在代码中是“放”在某一函数中的，但是其存放位置和全局变量一样，存于堆中开辟的.bss和.data段，是共享的

d. 文件等公用资源 这个是共享的，使用这些公共资源的线程必须同步。Win32 提供了几种同步资源的方式，包括信号、临界区、事件和互斥体。

独享的资源有

a. 栈 栈是独享的

b. 寄存器 这个可能会误解，因为电脑的寄存器是物理的，每个线程去取值难道不一样吗？其实线程里存放的是副本，包括程序计数器PC

1. 多个进程可以怎么通信？

进程间通信的方法主要有以下几种：

（1）管道（Pipe）：管道可用于具有亲缘关系进程间的通信，允许一个进程和另一个与它有共同祖先的进程之间进行通信。

（2）命名管道（named pipe）：命名管道克服了管道没有名字的限制，因此，除具有管道所具有的功能外，它还允许无亲缘关 系 进程间的通信。命名管道在文件系统中有对应的文件名。命名管道通过命令mkfifo或系统调用mkfifo来创建。

（3）信号（Signal）：信号是比较复杂的通信方式，用于通知接受进程有某种事件发生，除了用于进程间通信外，进程还可以发送 信号给进程本身；linux除了支持Unix早期信号语义函数sigal外，还支持语义符合Posix.1标准的信号函数sigaction（实际上，该函数是基于BSD的，BSD为了实现可靠信号机制，又能够统一对外接口，用sigaction函数重新实现了signal函数）。

（4）消息（Message）队列：消息队列是消息的链接表，包括Posix消息队列system V消息队列。有足够权限的进程可以向队列中添加消息，被赋予读权限的进程则可以读走队列中的消息。消息队列克服了信号承载信息量少，管道只能承载无格式字节流以及缓冲区大小受限。

（5）共享内存：使得多个进程可以访问同一块内存空间，是最快的可用IPC形式。是针对其他通信机制运行效率较低而设计的。往往与其它通信机制，如信号量结合使用，来达到进程间的同步及互斥。

（6）内存映射（mapped memory）：内存映射允许任何多个进程间通信，每一个使用该机制的进程通过把一个共享的文件映射到自己的进程地址空间来实现它。

（7）信号量（semaphore）：主要作为进程间以及同一进程不同线程之间的同步手段。

（8）套接口（Socket）：更为一般的进程间通信机制，可用于不同机器之间的进程间通信。起初是由Unix系统的BSD分支开发出来的，但现在一般可以移植到其它类Unix系统上：Linux和System V的变种都支持套接字。

9、 mysql底层用了什么数据结构 MySQL的存储引擎，有什么区别。

<https://blog.csdn.net/z_ryan/article/details/79685072>

B树适合文件系统的IO B+树主要用于数据库的查询

B+树查找范围是有优势

数据库的聚集索引、非聚集索引 <https://www.cnblogs.com/lyhabc/p/3196479.html>

10、数据库索引 看w3cschool 数据库索引的结构

<https://www.jianshu.com/p/b72d3ab9e54a>

11、md5算法原理 rsa算法原理 ecc算法原理

12、数据库语句 sql语句

13、实现一个高效的定时器？？设计模式

14、单例模式、观察者模式、工厂模式、装饰者模式

15、网络相关：cookies和session的区别？

session是怎么做区分的？

cookie是怎么传到前端的？（M：啊，new Cookie()???    T:你说的语言实现吧，我说的是怎么传的？ M：这个。。。 T：没关系没关系）我觉得他是想问数据的传输过程？这地方求大佬指导

TCP/IP各个层的传输协议

单点登录怎么实现的？

16、tcp三次握手 TIME\_WAIT状态 TCP的四次挥手，time wait状态有什么意义

17、进程和线程通信方式

18、朋友之间的点对点关系用图维护，怎么判断两人是否是朋友，并查集，时间复杂度，过程。没讲清楚 并查集的应用

19、如何实现关键字输入提示，使用字典树，复杂度多少，有没有其他方案，答哈希，如果是中文呢，分词后建立字典树？

Tire字典树 用一个hashmap记录词频

20、代码中遇到进程阻塞，进程僵死，内存泄漏等情况怎么排查。通过ps查询状态，分析dump文件等方式排查。

21、Linux了解么，查看进程状态ps，查看cpu状态 top。查看占用端口的进程号netstat grep

22、Linux的swap了解么，完全不懂

23、Linux的cpu 100怎么排查，top jstack，日志，gui工具

Linux大文件怎么查某一行的内容 问我proc文件系统了解么，答不了解

24、秒杀系统的架构设计

25、判断一棵树是否是对称的  非递归+递归 非递归咋做？

26、怎样去避免竞争条件（锁之类的）

27、中序遍历下一个节点怎么计算

28、一个链表，只知道指向这个节点的指针，怎么删除这个节点（将下一个节点的值复制到当前节点，并删除下一个节点）

29、seclect实现原理（数组、阻塞、轮询） epoll

30、内核态和用户态区别（这个直接说了不会）

31、指针和引用的区别

★ 相同点：

1. 都是地址的概念；

指针指向一块内存，它的内容是所指内存的地址；引用是某块内存的别名。

★ 区别：

1. 指针是一个实体，而引用仅是个别名；

2. 引用使用时无需解引用(\*)，指针需要解引用；

3. 引用只能在定义时被初始化一次，之后不可变；指针可变；

引用“从一而终” ^\_^

4. 引用没有 const，指针有 const，const 的指针不可变；

5. 引用不能为空，指针可以为空；

6. “sizeof 引用”得到的是所指向的变量(对象)的大小，而“sizeof 指针”得到的是指针本身(所指向的变量或对象的地址)的大小；

typeid(T) == typeid(T&) 恒为真，sizeof(T) == sizeof(T&) 恒为真，

但是当引用作为成员时，其占用空间与指针相同（没找到标准的规定）。

7. 指针和引用的自增(++)运算意义不一样；

32、内存

1、栈区（stack）： 由编译器自动分配释放，存放函数的参数值、局部变量的值等，其操作方式类似于数据结构中的栈

2、堆区（heap）：一般由程序员分配释放，若程序员不释放，程序结束时可能由OS回收，注意它与数据结构中的堆是两回事，分配方式倒是类似于链表

3、全局区（静态区 static）： 全局变量和静态变量的存储是放在一块的，初始化的全局变量和静态变量在一块区域，未初始化的全局变量和未初始化静态变量在相邻的另一块区域，程序结束后由系统释放。

4、文字常量区：常量字符串就是存放在该区，程序结束后由系统释放

5、程序代码区：存放函数体的二进制代码

33、红黑树 平衡二叉树

34、http协议和https协议的不同和其大概原理

35、几个sizeof的题，就是int\* a=malloc(10),cout<<sizeof(a)<<endl之类。

36、多线程的同步方法和如何使用 linux的进程间通信与windows的不同点

37、malloc和alloc的区别 malloc的机制，内存管理相关

38、vector何时进行空间的搬运？ 右值引用？?

39、n个整数的无序数组，找到每个元素后面比它大的第一个数，要求时间复杂度为O(N)

40、5种IO模型

41、异步编程的事件循环

42、为什么要有page cache，操作系统怎么设计的page cache

43、STL里resize和reserve的区别

44、撸一个std::lower\_bound，不断优化，直到最坏复杂度也为O(logN)

45、怎么实现线程池

用到哪些C++的新特性

怎么唤醒，调度线程

什么是LRU缓存

怎么设计的LRU缓存，详细

实现一个二叉树的持久化方案，可以伪代码，必须用指针

主要是序列化和反序列化的过程

（卡在指针的持久化，然后他不是很满意，说我应该做得出，然后我只能把我的另一种不用指针的方案给他讲，才算勉强过）