1. TCP/IP协议簇是Internet的基础，也是当今最流行的组网形式。[TCP/IP](http://baike.baidu.com/subview/7729/7729.htm)是一组协议的代名词，包括许多别的协议，组成了TCP/IP协议簇。其中比较重要的有[SLIP](http://baike.baidu.com/subview/32775/20284376.htm)协议、PPP协议、IP协议、ICMP协议、ARP协议、TCP协议、UDP协议、[FTP协议](http://baike.baidu.com/subview/1157060/1157060.htm)、[DNS协议](http://baike.baidu.com/subview/2901527/2901527.htm)、[SMTP协议](http://baike.baidu.com/subview/21931/21931.htm)等。[TCP/IP协议](http://baike.baidu.com/subview/7649/7649.htm)并不完全符合OSI的七层参考模型。
2. ;**TCP/IP协议族中的加密协议有什么**，我说了个**ssl，她问我是哪一层的**；
3. SSL(Secure Sockets Layer [安全套接层](http://baike.baidu.com/view/525499.htm)),及其继任者[传输层安全](http://baike.baidu.com/view/831898.htm)（Transport Layer Security，TLS）是为[网络通信](http://baike.baidu.com/view/538641.htm)提供安全及[数据完整性](http://baike.baidu.com/view/702953.htm)的一种安全协议。TLS与SSL在[传输层](http://baike.baidu.com/view/239605.htm)对网络连接进行加密。
4. 男老师2：说一说**客户端和服务器**

客户和服务器都是指在通信中所涉及的两个应用进程, 他们之间是服务与被服务的关系

：**客户端和服务器用英文怎么说**？

Client客户 [ˈklaɪənt]

Server服务器[ˈsɜ:rvə(r)]

P2P: 电驴, P2(to)P种子搜索器,

**CS和BS是什么**

C/S Client/Server客户/服务器模型 是双向的通讯，任何一方都可以随时向对方发送信息。  
  
B/S Browser/Server，[浏览器](https://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8)/[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)模式是「查询」式的通讯，只有客户端向服务器查询时，服务器才能向客户端发送信息.

C/S 是双向通讯，B/S 是一问一答。

1. 浏览器的英文browser([ˈbraʊzɚ])

**端到端指的**进程到进程，**通过套接字(IP地址+端口号)**



**数据结构**

链路怎么判断环,

设置两个指针fast和slow，刚开始都指向链表头部，fast每次移动两个位置，slow每次移动一个位置，若两个指针相遇，说明有环。若fast为null（nʌl）, 则无环(**null只能是地址，不能是int型，表示地址为空，即不存在**)

**在链表里怎么找到环的入口**，

**一解：** 从链表头、与相遇点分别设一个指针，每次各走一步，两个指针再次相遇点就是环的入口。

a,是链表头到入口的距离。 b,是从入口到相遇掉的距离

2s= a+ b + mr

S = a + b + nr

S= (m-n)r=a+b+nr----------------->a = ox-b

**二解**第一问得出相遇点后，再发出一个指针，统计这个指针再次回到这个点的距离，也就是环的距离。   
然后从起点再发出两个指针，一个指针在另一个前面，两个指针的距离就是环的距离，当两个指针再次相遇的时候就是环的入口。

1. **算法里的背包问题以及贪心算法**。
2. **0**-1背包用动态规划算法
3. **0**-1背包：一组物品，每种物品有自己的重量和价格，且每种物品仅一个，在限定的总重量内，如何选择，使得物品的总价格最高
4. 动态规划算法与分治法类似，其基本思想也是将待求解问题分解为若干个子问题，先求解子问题，然后从这些子问题的解得到原问题的解。
5. 与分治法最大的差别是：适用于动态规划求解的问题，经分解后得到的子问题往往不是互相独立的（即下一个子阶段的求解是建立在上一个子阶段的解的基础上，进行进一步的求解）
6. 背包问题(Knapsack problem)是一种组合优化的[NP完全问题](https://baike.baidu.com/item/NP%E5%AE%8C%E5%85%A8%E9%97%AE%E9%A2%98)。问题可以描述为：给定一组物品，每种物品都有自己的重量和价格，在限定的总重量内，我们如何选择，才能使得物品的总价格最高

NP完全问题：[多项式](https://baike.baidu.com/item/%E5%A4%9A%E9%A1%B9%E5%BC%8F)复杂程度的非确定性问题。

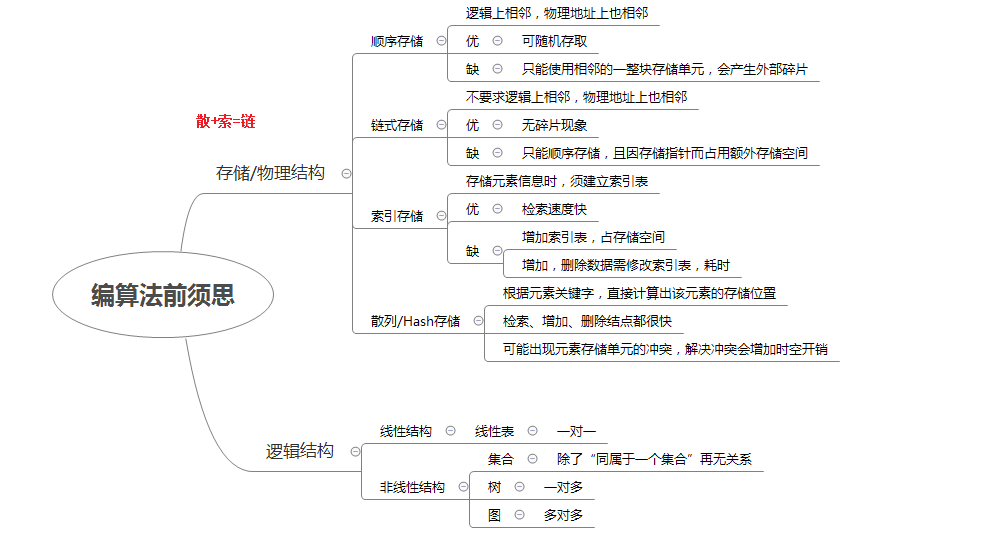
12.怎么判断环？（忘了是链表还是图了）

数据**结构里面编算法之前需要做什么/数据结构三要素**

**逻辑结构+存储/物理结构+数据运算**

**逻辑结构：数据之间的逻辑关系。**

**存储结构：数据的逻辑关系在计算机中的表示。**



**问判断算法优劣的标准：时空 ，以为自己正确的毒瘤，才健壮。**

**正确性：对任何合法的输入，会得出正确的结果。**

**健壮性：当输入非法数据时，不会导致程序瘫痪。**

**可读性：被理解的难易程度。**

**时间复杂度和空间复杂度**：算法运算所需要的时间，算法运算时所需要的存储空间  
常见的排序算法（了解它们的算法思想和时间及空间复杂度，若能手写代码是最好的了），树的定义，给出一颗树要求写出广度和深度优先序列（这个简单哈）

图的遍历就是按照某种次序访问图的每一顶点一次仅且一次。

实现bfs和dfs都需要解决的一个问题就是如何存储图。一般有两种方法：邻接矩阵和邻接表。这里为简单起

见，均采用邻接矩阵存储，说白了也就是二维数组。

1. 

BFS: 广度优先搜索遍历Breadth First Search

DFS: 深度优先搜索遍历 Depth

在搜索最短路径时，适合用BFS，BFS类似二叉树层次遍历，BFS搜索过程中遇到的解一定是离根最近的，所以遇到一个解一定是最优解

在搜索全部的解时，适合用DFS，BFS在搜索过程中使用队列记录搜索过的状态，而DFS使用递归工作栈，不需要保存搜索过的状态。

两个栈模拟队列,

栈是先入后出，队列是先入先出。用两个栈来模拟队列。栈1用来入队列，栈2用来出队列。

入队列时，将元素压入栈1即可。

出队列时，先看栈2是否为空，若不为空，栈2弹栈。若为空，先将栈1元素弹出，压入栈2，之后栈2弹栈。

堆栈的区别,

栈区（stack）— 由编译器自动分配释放 ，存放方法调用和函数参数值和局部变量 一般速度较快；存储地址是连续且存在有限栈容量，会出现溢出现象

堆区（heap） — 一般由程序员分配释放，通过(malloc和new)动态申请内存, 一般速度较栈慢；存储地址通常是链式的，内存较大不会溢出, 若程序员不释放，程序结束时可能由OS回收

1. 树、二叉树、满二叉树、完全二叉树的区别

1树(递归的非线性数据结构) ：N个结点的有限集合，N=0时，称为空树 满足有且仅有一个特定的称为根的结点 当N>1时，其余节点可分为m（M>0）个互不相交的有限集合，其中每个集合本身又是一棵树,并成为根结点的子树.  
1）二叉树：每个节点至多有两个子树（二叉树中不存在度大于2的结点），二叉树有左右之分，不能相互颠倒  
（1）满二叉树：一个高度为n，并且有2n-1个结点的二叉树称为满二叉树

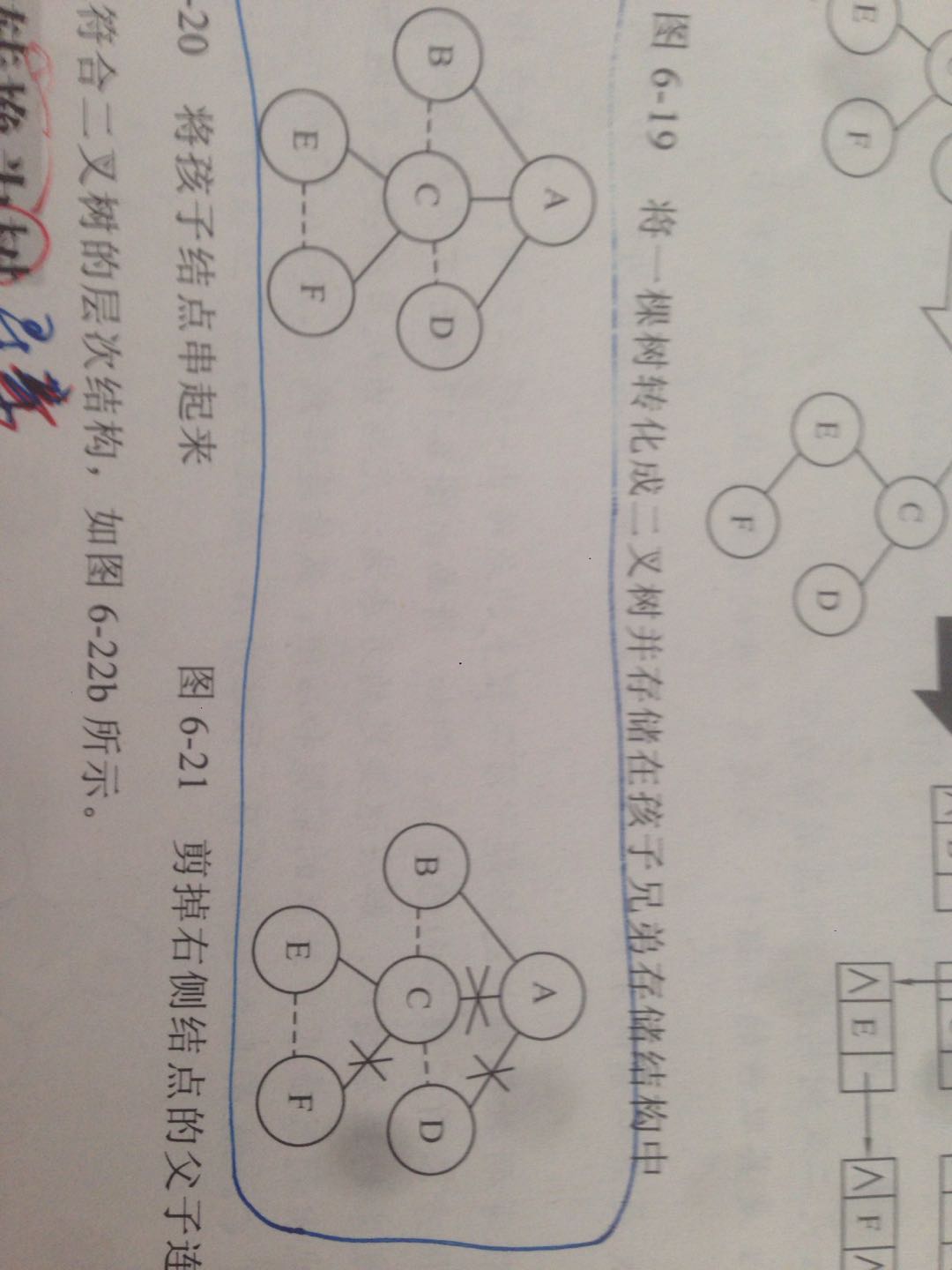
（2）完全二叉树 ：一个高度是n的二叉树，当且仅当其每一个结点都与高度为n的满二叉树中编号为1~2n-1的结点一一对应时，此时称完全二叉树.

可由满二叉树从右至左, 从下至上, 挨个删结点形成

（3）二叉排序树：一棵二叉树或者空二叉树，或者具有如下性质的二叉树：左子树上所有结点的关键字均小于根节点上的关键字，右子树上所有结点的关键字均大于根节点的关键字，左子树和右子树又各是一棵二叉排序树  
（4）平衡二叉树(AVL树)：树上任一结点的左子树和右子树的深度之差不超过1  
2）树转换为二叉树：根据左孩子，右兄弟的原则，每个节点左指针指向它的第一个孩子结点，右指针指向它在树中的相邻兄弟节点

树<-->二叉树画法: 兄弟连, 骨肉残

将同一结点的各孩子结点用线串起来, 把每个结点的分支从左往右除了第一个之外, 其余全都剪掉, 最后调整结点使之符合二叉树的层次结构

  
  森林转换为二叉树: 先将森林中的每一棵树转换为二叉树, 然后将第一个树的根作为转换后的二叉树的根, 第一棵树的左子树作为转换后二叉树的左子树, 第二棵树作为转换后二叉树的右子树, 第三棵树作为转换后二叉树的右子树的右子树, 以此类推  
二叉树转换为森林:

若二叉树非空, 则二叉树根及其左子树为第一棵树的二叉树形式, 二叉树的右子树又可以看做是一个由除第一棵树外的森林转换后的二叉树, 应用同样的方法, 直到最后产生一颗棵没有右子树的二叉树为止, 这样就得到了原森林

森林<--->二叉树的画法: 右下皆兄弟, 排排坐, 找妈咪`   
3）哈夫曼树哈夫曼编码

哈夫曼树(也可以3,4叉): 又称最优二叉树, 带权路径长度最小的二叉树.

树的带权路径长度(WPL: weighted path length): 树中所有叶子结点的带权路径长度之和

结点的带权路径长度: 结点具有权值, 从该结点到根之间的路径长度乘以结点的权值

霍夫曼树特点: 1大短xx(霍比特人的gg)

权值越大的结点, 距离根结点越近

树中没有度为1的结点, 此类树又称为正则(严格)二叉树

树的带权路径长度最短

霍夫曼编码:

是一种数据压缩编码, 通过可变长度编码, 产生的是最短前缀码.

前缀码: 任一字符的编码都不是另一字符编码的前缀

.zip压缩文件就用到了霍夫曼编码

常见排序的思想及其时空复杂度。  
2 常见的排序和复杂度  
O(log2n): 欧 lao ge 以2为底n的对数  
1）插入排序  
   （1）直接插入排序：将当前的元素排到前面已经有序的序列中（适用于基本有序和数据量不大的排序表）  
   （2）折半插入排序：用折半查找找出 元素待插入的位置，然后再统一的移动待插入位置之后的所有元素  
   （3）希尔排序（缩小增量排序）：先将待排序表分割成若干个形如L[I,I+d,i+2d~~~]的特殊子表，分别进行直接插入排序，当整个表中元素呈现基本有序时，再对全体记录进行一次直接插入排序（仅适合于当线性表为顺序存贮的情况）空间复杂度为o（1）  
2）交换排序  
  （1）冒泡排序：假设待排序表长为n，从后往前（或从前往后）两两相邻元素的值，若为逆序，则交换他们，直到序列比较完，结果将最小的元素交换到待排序的第一个位置来  
  （2）快速排序：在待排序表中选取一个元素pivot作为基准，通过一趟排序将排序表划分为独立的两部分，前半部分都小于pivot,后半部分都大于pivot,则pivot被放在了最终位置上，然后再对这两个子序列进行划分。  
3）选择排序  
  （1）简单选择排序 ：假设排序表为L[1，2，。。。。。N]，第I趟排序即从L【i。。。。N】中选择最小的元素与L（i）交换，每一趟排序可以确定一个元素的最终位置。  
   （2）堆排序：树形选择排序方法，在排序过程中，将L[1.。。。N]看成是一棵完全二叉树的顺序存贮结构，利用完全二叉树中双亲结点和孩子结点之间的内在关系，在当前无序区选择最大最小元素  
4）归并排序：两个或两个以上的有序表组成一个新的有序表  
5）基数排序：基于关键字排序思想  
每轮排序都会确定一个最终位置：冒泡，快排，选择排序，堆  
3各种链表的算法  
4几种冲突以及解决方法  
5B树：多路平衡查找树，B树中所有结点的孩子结点数的最大值称为B树的阶  
1）树中每个节点至多有M棵子树（即至多含有M-1个关键字）  
2）若根节点不是终端节点，则至少有两棵子树  
3）除根节点以外的所有非叶节点至少有【M/2】棵子树  
4）所有叶节点都出现在同一层  
B树是所有节点的平衡银子都等于0的多路查找树。  
  
B+树的概念：  
1）每个分支最多有M棵子树。  
2）每个非叶节点至少有两棵子树，其他每个分支节点至少有【M/2】棵子树  
3）结点的子树个数与关键字个数相等  
4）所有叶节点包含全部关键字以及指向相应记录的指针，而且叶节点中将关键字按大小顺序排列，并且相邻叶节点按照大小顺序相互连接起来。  
5）所有分支节点（可以看成索引的索引）中仅包含它的各个子节点中的最大值及指向子节点的指针。



函数调用: 先定义并声明函数，之后再根据定义函数的格式调用

函数调用模式: 函数模式,方法模式,构造器模式,

说一下**快速排序的实现过程**，**最好情况和最坏情况的时间复杂度**。

思想(分治思想)：一趟排序，用枢轴值将待排记录分成两块，（较小记录，枢纽值，较大记录），分别对两块记录递归排序，最终，整个序列有序。平均时间复杂度：O(nlog2n)

最好情况（无序列）: O(nlog2n), 此时枢轴值恰好能将待排序列从中间分开

最差情况: O(n^2), 此时待排序列有序，选择的枢轴值最终每次都在序列的两端。

分治思想：将复杂的原问题分解为多个规模较小且性质相同的子问题，求出子问题的解，将其合并，就是原问题的解。如二分法，快排啊

1. “简单介绍下你使用过的数据结构
2. 归并排序, 冒泡与插入

**，一个是单链表倒置的题，**

**新建一个链表，将原链表的从首元结点开始，依次摘下，用头插法插入新链表即可。（每次摘结点之前，要保存其后结点的地址）**

[复制代码](javascript:void(0);)

1 Node\* Reverse(Node\* node)

2 {

3 Node\* prev = NULL; // 用于保存当前链表的头结点

4 Node\* tmp = NULL; // 用于保存当前节点的next

5 while (node != NULL)

6 {

7 tmp = node->\_next; // 保存当前节点的next

8

9 node->\_next = prev; // 将当前节点插入到头结点前

10 prev = node; // 插入之后将当前节点设置为头节点

11

12 node = tmp; // next为下次迭代的当前节点

13 }

14 return prev; // 循环结束后，p即为倒置后的头结点

15 }

1. 求若干序列的公共子序列
2. 查找最大公共子序列（LCS: longest common subsequence）：常用于解决字符串的相似度，
3. 个序列 S ，如果分别是两个或多个已知序列的子序列，且是所有符合此条件序列中最长的，则 S 称为已知序列的最长公共子序列
4. **与最长公共子串**不同的是：子序列不一定在原序列中连续，子串要求连续。

前沿

数据挖掘算法：

PageRank （Google的搜索引擎）。为了决定一些对象和同网络中的其他对象之间的相对重要程度而设计的连接分析算法(link analysis algorithm)。

C4.5 以决策树的形式构建了一个分类器，C4.5 算法在生成信息树的时候使用了信息增益，决策树最好的卖点是他们方便于翻译和解释

The term "big data" often refers simply to the use of [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics), [user behavior analytics](https://en.wikipedia.org/wiki/User_behavior_analytics), or certain other advanced data analytics methods that extract value from data

“大数据”一词通常是指简单地使用预测分析、用户行为分析、或某些其他先进的数据分析方法,从数据中提取价值,很少到一个特定的数据集的大小。

大, 一是指数据量大, 一是指数据源多. 利用不同来源的庞大的数据量进行分析, 来对未来进行预测.

   金融大数据

集合海量非结构化数据，通过对其进行实时分析，通过分析和挖掘客户的交易和消费信息掌握客户的消费习惯，并准确预测客户行为，使金融机构在营销和风险控制方面有的放矢

云计算基于Hadoop平台,分布式基础构架，Google开发，其核心是HDFS（分布式文件系统Hadoop Distributed File System）为海量的数据提供了存储，利用mapreduce进行计算。

云计算：按使用量付费商业计算模式。

1.云计算是一种通过网络提供高性能的计算资源的，可以弹性变化的，共享的一种服务模式。它综合应用了多种技术，但不是一种新技术。

2.说它是“云”，因为它很难找到边界，而且是弹性变化的，像天空的云一样。另外，提供云计算服务的中心有一堆服务器，像 “云”一样不易计数。

计算能力和存储能力向互联网端的迁移和抽象。迁移指计算和存储都不在本机甚至本地服务器而迁移到了互联网的服务器集群上，抽象指最终用户无需关系真正的存储和计算发生在何处。

云计算包含两方面内容，一是在网络上提供的为计算服务的应用，例如以前被称为SaaS(Software as a Service)的那一类应用；二是提供这些服务的在数据中心的硬件和系统软件，这部分也就是我们通常所称呼为“云”的东西

**通过*网络*以*自助服务*的方式获得所需要的*IT资源*的模式**

云计算就是租！将计算力作为服务进行销售

非本地计算，都可以算作云计算。小到一个程序应用，大到整个计算架构

云计算的本质是一种服务提供模型，通过这种模型可以随时随地、按需的通过网络访问共享资源池的资源，这个资源池的资源内容包括计算资源、网络资源、存储资源等，这些资源能够动态的分配和调整，在不同的用户之间;灵活切换划分。

你可以直接形象化的理解为云计算就是生活中的水电厂，之前每家每户都需要自己买发动机自己打水井，现在有一个公司供水供电还便宜，这个公司就是云计算厂家。

云计算是一种按使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问， 进入可配置的计算资源共享池（资源包括[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C)，服务器，存储，[应用软件](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E8%BD%AF%E4%BB%B6)，服务），这些资源能够被快速提供，只需投入很少的管理工作，或与服务供应商进行很少的交互。

云，就是互联网上的提供计算服务的计算集群，提供云计算服务的中心的服务器集群。很多，像云一样，无边无界

表示[互联网](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91)和底层基础设施的抽象

Cloud Computing）

Refers to the provision of computational resources on demand via a computer network.云计算指通过计算机网络，根据需要提供计算资源。

老板，我们现在人太多了，人不能闲在这里。

啪，你不会考虑一下把这些人卖出去，给别人开发项目。

但是，别人只要半个人，或者几个小时的工作量，我们一个人一个人的太多了。

啪，你不会把一个人拆成几个虚拟的人啊，或者按工时卖就行了。

想要管理这么多机器，做这么多事情，还真不简单，还要有人管协调分工（并行计算），有人管理谁多谁少（负载均衡），还需要容许别人生病、辞职（热备冗余），还需要解决卖半个人，甚至十分之一个人的问题（虚拟化）。。。

[人工智能](https://baike.baidu.com/item/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD/9180)（Artificial Intelligence），[英文](https://baike.baidu.com/item/%E8%8B%B1%E6%96%87)缩写为AI

人工智能就是研究如何使计算机去做过去只有人才能做的智能工作

云[安全](https://baike.baidu.com/item/%E5%AE%89%E5%85%A8)（Cloud Security）：通过网状的大量客户端对网络中软件行为的异常监测，获取互联网中木马、恶意程序的最新信息，推送到Server端进行自动分析和处理，再把病毒和木马的解决方案分发到每一个客户端。

**机器学习 ： machine learning在不直接针对问题进行编程的情况下，赋予计算机学习能力，让计算机程序随着经验积累自动提高性能**

**互联网金融**Internet Finance（ITFIN） ：传统[金融机构](https://baike.baidu.com/item/%E9%87%91%E8%9E%8D%E6%9C%BA%E6%9E%84)与互联网企业利用互联网技术和信息通信技术实现资金[融通](https://baike.baidu.com/item/%E8%9E%8D%E9%80%9A)、[支付](https://baike.baidu.com/item/%E6%94%AF%E4%BB%98)、[投资](https://baike.baidu.com/item/%E6%8A%95%E8%B5%84)和信息中介服务的新型金融业务模式



VR(virtual reality，虚拟现实)：可以让用户沉浸其中的由计算机生成的三维虚拟环境，并与现实环境相隔绝。

AR(augmented reality，增强现实)：在真实环境中增添或者移除由计算机实时生成的可以交互的虚拟物体或信息。

MR(mixed reality，混合现实)：将现实环境与虚拟环境相互混合，也可以看成是VR与AR的混合。

区块链Blockchain ： （[blɑk tʃen]）

**用共识算法来对于新增数据达成共识，采用密码学的方法来保证已有数据不可能被篡改的去中心化的分布式数据库。**

系统结构(操作系统体系结构): 微内核和宏(大)内核

宏内核: 将操作系统的主要功能模块都作为一个整体运行在内核态, 不易于扩展，但是功能较强，效率比较高。  
微内核: 把一些必要的功能集成在内核里，易于扩展，功能相对较弱, 效率没有宏内核高(要频繁的在用户态和核心态间进行切换)。

windows 是微内核系统, Linus, unix是宏内核系统

物联网: 即是物物相连的互联网. 物联网是互联网的基础上进行扩展和延伸的网络, 其用户端扩展和延伸到了任何物品与物品之间, 进行信息交换和通信.

软工

软件开发模型(软件设计中的模型)：  
1)**瀑布模型** 前一阶段的工作完成后，才能开始下一阶段的工作，前一阶段的输出文档就是后一阶段的输入文档。  
2)**快速原型模型** 快速建立起可以运行的程序，完成的功能是最终产品的一个子集   
3)增量模型 在各个阶段并不交付一个可运行的完整产品，而是交付用户满足用户需求的一个子集的可运行产品  
4)螺旋模型 结合快速原型和瀑布模型，每个阶段之前都增加了风险分析的快速原型模型   
5)喷泉模型以用户需求为动力，以对象作为驱动的模型。具有迭代性和**无间隙性**

模块独立程度：（1）耦合 一个软件结构内不同模块之间互联程度的度量 数据耦合 控制耦合 特征耦合 公共环境耦合 内容耦合 （2）内聚 一个模块内各个元素彼此结合的紧密程度 偶然内聚 逻辑内聚 时间内聚（必须在同一段时间内完成）过程内聚（顺序） 通信内聚（必须使用同一个输入数据和产生同一个输出数据） 顺序内聚（一个模块的输出作为下一个模块的输入） 功能内聚（所有处理元素属于同一整体，完成单一的功能）

11 面向对象

1）客观世界是由各种对象组成的，任何事物都是对象，复杂的对象可以由简单的对象以某种方式组合而成  
2）把所有的对象都划分成各种对象类，每个对象类都定义了一组数据和一种方法。  
3）按照子类与父类的关系，把若干个对象类组成一个层次结构的系统。  
4）对象之间仅能通过传递消息互相联系。  
优点：与人类习惯的思维方法一致，稳定性好，可重用性好，较易开发大型软件  
对象：由描述该对象属性的数据以及可以对这些数据施加的所有操作封装在一起构成的统一体  
特点：以数据为中心 对象是主动的 实现了数据封装 本质上具有并行性 模块独立性好  
  
类：具有相同数据和相同操作的一组相似对象的定义，也就是说，类是对具有相同属性和行为的一个或多个对象的描述  
  
  
  
重载：1）函数重载是在同一作用域内的若干个参数特征不同的函数可以使用相同的名字  
     2）运算符重载是指同一个运算符可以施加于不同类型的操作数上面  
面向对象设计的准则：模块化，抽象 信息隐藏 弱耦合 强内聚 可重用  
  
  
  
24面向对象建模的三种模型  
1）对象模型：描述系统数据结构  
2）动态模型：系统控制结构，表示瞬时的，行为化的系统控制性质，规定了对象模型中对象的合法变化序列  
3）功能模型：系统功能（数据流图，用例图）

**计算机网络**

4层结构, 可看做五层(TCP/IP模型)

物理层, 数据链路层, 网络层, 传输层, 应用层  
1 TCP/IP（Transmission Control Protocol/Internet Protocol）即传输控制协议/网间协议  
应用层：  
运输层：TCP （面向连接，报文段）UDP（用户数据报）  
网际层：IP（IP数据报====分组）  
网络接口层：

OSI/RM（open system interconnection reference model）开放系统互连基本参考模型

Physical layer, data link layer, network layer, Transport Layer, session layer, presentation layer, application layer  
应用层：

DHCP（动态主机配置协议 dynamic host configuration protocol）基于udp  
     
        RIP(路由信息协议 routing information protocol)距离向量(udp)  
        BGP（边界网关协议 border gateway protocol）(tcp)  
        FTP（文件传输协议 file transfer protocol）(控制端口21 数据连接20)  
        SMTP(简单邮件传输协议 simple mail transfer protocol)  
        Pop3(邮局协议 post office protocol)端口号110  
        HTTP(超文本传输协议 HyperText  Transfer  Protocol)  
  
表示层：不同的表示方法的数据和信(不同通信系统))之间能相互交换(数据格式转换)  
会话层：允许不同主机上进程间的会话

传输层: 进程之间的逻辑通信,(可同一主机, 也可不同主机).

一个进程一个端口

端到端: 套接字(ip地址+端口号)唯一标识端口

网络层：

IP数据报首部中的检验和只检验首部不检查数据部分     
    ARP（地址解析协议 address resolution protocol）  
      RARP（逆地址解析协议 reverse）  
     ICMP（网际控制报文协议 internet control message protocol）允许主  机或路由器报告差错和异常情况  
      IGMP(网际组管理协议 intnet group management protocol)  
     IP（网际协议 internet protocol）   [ˈprəʊtəkɒl]

   OSPF(开放最短路径优先)

数据链路层：ALOHA(additive link online hawaii system)  
            CSMA(载波侦听多路访问 carrier sense multiple acess)  
        CSMA/CD(载波侦听多路访问/碰撞协议 ~~with collision dection)  
           CSMA/CA（/碰撞避免）  
           令牌传递协议  
（以上都属于随机访问介质访问控制）  
          HDLC   PPP     
物理层：  
协议是水平的，服务是垂直的  
协议数据单元pdu(protocol data unit)对等层次间数据单元的传递  
服务数据单元sdu(service data unit) 层间数据的传递  
分层的好处：各层之间相互独立，易于实现和维护  
复用：多个应用层进程同时使用下面传输层的服务  
分用：传输层把收到的信息分别交给上面应用层中相应的进程  
  
  
2 TCP与UDP的区别

都有: 复用和分用

复用：多个应用层进程同时使用下面传输层的服务  
分用：传输层把收到的信息分别交给上面应用层中相应的进程

Tcp和udp都检验: 首部+数据  
  TCP: （transmission control protocol 传输控制     
        协议） 使用TCP的应用层协议：SMTP TELNET    
        HTTP FTP  
面向连接 每一条tcp连接只能由两个端点 提供可靠交付 全双工 面向字节流  
  UDP:（user datagram protocol 用户数据报协议）  
         使用UDP的应用层协议：DNS TFTP RIP DHCP   
         SNMP,IP电话，流式多媒体的通信  
无连接 尽最大努力交付 面向报文, 首部开销小(tcp报文段首部=IP数据报首部=20B, UDP用户数据报首部=8B)  不提供流量控制, 没有拥塞控制 (两个都是连接的事) 不保证顺序接受支持一对一，一对多，多对一，多对多的交互通信,

优点：首部开销小, 发送之前不需建立连接，减少时延, 快速. 不保证可靠交付，主机不需维持复杂的连接状态表 无拥塞控制有利于实时应用 tcp只支持一对一,udp支持一对多, 多对一.

如何确定udp可靠  
（1）比对目的端口号是否存在(UDP首部: 源端口, 目的端口, 长度, 检验和)，不存在就丢弃，并由icmp发送端口不可达给发送方  
（2）udp检验和, 检验首部和数据部分是否正确

3 ip的作用

ip协议定义了, 数据传送的基本单元——IP分组及其明确的数据格式。

/IP协议的作用, 就是向传输层提供统一的IP数据报/分组，即将各种不同类型的MAC帧转换为统一的IP数据报/分组， 并将MAC帧的物理地址变换为全网统一的逻辑地址（IP地址）。

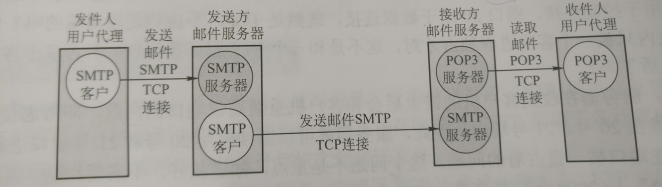
4 ipv4和ipv6的区别，联系

  Ipv6：更大的地址空间 扩展的地址层次 支持即插即用  更小的路由表 增加了增强的组播和对流的支持 具有更高的安全性  
Ipv6只有在源结点才能分片，是端到端的，传输路径中不能分片 ipv6首部是长度必须是8B的整数倍, 而IPV4是4B的整数倍 增加了组播功能 简化了报文头部 对于可选项有了更大的支持  
  Ipv6跟TCP UDP ICMP IGMP等都兼容  
  过渡采用了：双协议栈，隧道技术  
5 简述数据链路层  
（1）封装成帧：在网络层传下来的ip数据报的前面和后面分别添加首部和尾部，首部和尾部用于帧定界   
（2）透明传输：字符计数法，字符填充，比特填充，违规编码法（数据链路层成帧方法）  
（3）差错检测:无比特差错，主要使用CRC

6 DNS服务器的作用与原理

DNS属于客户/.服务器模型(UDP协议, 端口号53)  
作用：把域名转换为ip地址  
原理：当某个应用进程需要把主机名解析为ip地址时，调用解析程序，成为DNS的一个客户，把待解析的域名放在请求报文中，以UDP用户数据报方式发给本地域名服务器，查找域名后，把ip放在响应报文中返回，如果本地域名服务器不能回答，就向其他域名服务器发出请求，向本地查询时使用递归，向根域名服务器，顶级域名服务器，权限域名服务器通常使用迭代  
DNS迭代和递归的区别：  
递归：如果主机所询问的本地域名服务器不知道被查询域名的ip地址，那么本地域名服务器就以DNS客户的身份，向根域名服务器继续发出查询请求报文（即替该主机查询），而不是让该主机自己进行下一步查询  
迭代：当根域名服务器收到本地域名服务器发出的迭代查询请求报文时，要么给出所要查询的ip地址，要么告诉本地域名服务器应该去 哪一个域名服务器进行下一步的查询

7电子邮件的发送过程

用户代理: QQ邮箱, 163邮箱  
（1）发件人发信, 调用用户代理, 以SMTP客户的身份与发送方邮件服务器建立TCP连接, 使用smtp协议（简单邮件传输协议）发给邮件.  
（2）发送方邮件服务器收到邮件后，以 smtp客户的身份与接收方邮件服务器建立TCP连接，使用smtp发送邮件  
(3)收件人打算收信, 调用用户代理, 以Pop3客户身份与收件方邮件服务器建立TCP连接，使用pop3接收邮件  
**HTTP的持久连接和非持久连接**  
http1.0非持久连接: 传输一个网页元素对象, 一次TCP连接

每请求一个文档=传输文档时间+2倍的往返时间(一个RTT,用于tcp连接, 一个用于HTTP请求和接收文档)  
http1.1持久连接: 传输多个网页元素对象, 一次TCP连接

万维网服务器在发送响应后仍保持连接，使同一个客户和该服务器可以继续在这条连接上传送报文

简述网卡的作用: 又称网络适配器（网络接口卡NIC network interface card）, 局域网中连接计算机和传输介质的的接口.( 一块网卡包括物理层和数据链路层, 每块网卡有唯一MAC地址（物理地址）)

封(封装解封)码(编码译码)路(链路管理), 有网卡的放行(并串行)

.数据的封装与解封：发送时将上一层交下来的数据加上首部和尾部，成为以太网的帧。接收时将以太网的帧剥去首部和尾部，然后送交上一层；

链路管理：主要是CSMA/CD（Carrier Sense Multiple Access with Collision Detection ，带冲突检测的载波监听多路访问）协议的实现；

编码与译码：即曼彻斯特编码与译码。

进行串行/并行转换: 网卡和局域网之间通信通过电缆或双绞线, 以串行传输方式进行的。网卡和计算机之间通信通过计算机主板上I/O总线, 以并行传输方式进行  
RIP和OSPF  
  RIP:距离向量算法 15 收敛速度慢  
特点：（1）仅和相邻路由器交换信息  
      （2）路由器交换的信息是当前路由器知道的全部信息，即自己的路由表  
      （3）按照固定的时间间隔交换路由信息  
  Ospf:分布式链路状态路由  
特点：（1）向本自治系统的所有路由器发送信息，洪泛法  
      （2）发送的信息是与本路由器相邻的所有路由器的链路状态（本路由器与哪些路由器相邻，以及该链路的度量）  
      （3）网络层协议，不使用tcp或udp直接使用ip数据报传送（ip数据报协议字段首部89）  
      （4）只有当链路状态发生变化时，才用洪泛法。更新过程收敛的快  
  
10 通信的同步异步  
同步：通信双方必须先建立同步，即双方的时钟要调到同一个频率，收发双方不停的发送和接收连续的同步比特流  
异步：所发送的字符之间的时间间隔可以是任意的，但接收方必须做好时刻接收的准备  
  
11 码元传输率 ：单位时间内传输的码元个数（脉冲次数/信号变化次数）  
信息传输速率：单位时间内传输的二进制码元个数  
  
12 基带频带宽带传输

基带：不经调制就在信道上直接传输，通常用于局域网  
频带：用数字信号对特定频率的载波进行调制，将其变成适合于传送的信号在传输（远距离传输 或 无线传输）  
宽带：借助频带传输，将链路容量分解为两个或多个信道，每个信道可以携带不同的信号 宽带中所有的信道可以同时互不干扰的发送信号  
  
13 hub 路由器 交换机 网桥等都有什么作用  
1）物理层：  
    （1）中继器（转发器）：将信号整形并放大再转发出去，消除信号因为某些原因造成的失真和衰减，使信号的波形的强度达到要求（将衰弱的数字信号整形再生)  
    （2）集线器（多个端口的中继器 hub ）整形放大传送到其他端口（属于同一个冲突域）  
2）数据链路层：  
    （1）网桥：连接以太网，成为一个覆盖范围更大的以太网，原来每个以太网就可以成为一个网段（隔离碰撞域）  
   特点：网桥必须具备寻址和路径选择能力，以确定帧的传输方向  
         从源网络接收帧，以目的网络的介质访问控制协议向目的地转发该帧  
         网桥在不同或相同类型的LAN之间都可以存贮并转发帧，必要时还进行协议转换  
   优点：过滤通信量 扩大物理范围 可使用不同的物理层 可互联不同类型的局域网 提高了可靠性 性能得到改善  
   缺点：增加时延 MAC层没有流量控制功能 网桥只适合于用户数不多和通信量不大的局域网，否则会因为传播过多广播信息产生网络拥塞  
  
  （2）局域网（以太网）交换机：多端口的网桥：在某一时刻可以执行多个帧的转发操作（检测从以太网端口来的数据帧的源和目的地址的MAC(介质访问层)地址，然后与内部的动态查找表进行比较，若数据帧的MAC地址不在查找表中，则将该地址加入查找表中，并将数据帧发送给相应的目的端口）  
特点：全双工 即插即用 独占带宽   
  
-  
  
14 cookie 的作用  
HTTP是无状态的，同一个用户多次访问时，服务器并不记得曾经访问过的这个用户，也不记得为该客户服务过多少次。称为HTTP的无状态特性，使得服务器更容易支持大量并发的HTTP请求。在实际应用中，通常使用cookie加数据库的方式来跟踪用户的活动。Cookie是一个存贮在用户主机的文本文件，WEB服务器根据cookie就能从数据库中查询到该用户的活动记录，进而执行个性化的活动。  
  
15 介质访问控制  
  1）信道划分介质访问控制  
多路复用：通过在一条介质上同时携带多个传输信号的方法来提高传输系统的利用率  
频分多路复用（FDM）：将基带信号调到不同的频率载波上再进行叠加形成一个复合信号的多路复用技术（充分利用的传输介质的带宽，效率高，系统成熟，实现容易）  
时分多路复用（TDM）：将一条物理信路按时间分成若干个时间片，轮流的分配给多个信号使用  
波分多路复用（WDM）：光的频分多路复用  
码分多路复用（CDMA）：不同的编码来区分各路原始信号的一种复用方式  
2）随机访问介质访问控制  
ALOHA协议 纯ALOHA 当网络中的任何一个节点需要发送数据时，可以不进行任何检测就发送数据，如果早一段时间内没有收到确认，该站点就认为传输过程发生了冲突，发送站点需要等待一段时间再发送数据，直到成功  
时隙ALOHA 把所有各站的时间都同步起来，分0成一段段等长的时隙，规定只能在每个时隙开始时才能发送一个帧hu  
CSMA协议：在发送前侦听一下共用的信道，信道空闲后再发送  
CSMA/CD（总线型 半双工）先听后发 边听边发 冲突停发 随机重发  
CSMA/CA(无线 碰撞避免)  
3）轮询访问介质访问控制：令牌传递协议  
16TCP/IP协议族中的加密协议:

应用层：s-http，ssh，ssl，pgp；传输层：ssl，tls；网络层：ipsec；网络接口层：pptp

17客户机服务器（client/server C/S）：总是有一个打开的服务器，服务于许多来自其他客户机的主机请求 客户机一般不需要特殊的硬件和复杂的操作系统，而服务器上运行的软件则是专门用来提供某种服务的程序，可同时处理多个远程和本地用户的要求  
P2P（peer to peer）模型：每个结点即作为客户访问其他节点的资源，也作为服务器提供资源给其他节点访问  
18 ppp和HDLC  
PPP：串行链路通信的面向字节的协议 HDLC：面向比特  
通过拨号或专线方式建立点对点连接发送数据，使其成为各种主机，网桥，路由器之间连接的一种共同解决方案。  
包括：1)链路控制协议LCP：一种扩展链路控制协议，用于建立，配置，测试和管理数据链路。  
2）网络控制协议NCP：PPP允许同时采用多种网络层协议  
3）一个将IP数据报封装到串行链路的方法。  
19 电路交换和分组交换的联系与区别  
电路交换：  
1） 基于位置，即在某一位置的比特经交换后变更到另一个位置上。  
2） 面向连接的电路交换必定是面向连接的，但面向连接的却不一定是电路交换。  
3） 通话过程中始终占用端到端的固定传输带宽。  
分组交换：  
1） 采用存储转发技术  
2）基于标记的，将欲发送的整块数据称为报文(message)，将较长的报文划分为一个个更小等长数据段（比如 1024bit） ，在每个数据段前面加上首部(header)，后就构成一个分组(packet)，分组又称为包，分组的头部称为包头。分组首部包含了目的地址和源地址等重要的控制信息，每一个分组才能在分组交换网中独立地选择路由。  
3）无连接的  
20 解决ipv4地址用完  
分类的IP地址{<网络号><主机号>}  
分类的划分{<网络号><子网号><主机号>}  
构造超网{<网络前缀><主机号>}  
ipv6

NAT技术：将专用网络内部使用的IP地址转换为有效的外部使用的全球ip地址，使得整个专用网只需要一个IP地址就可以与因特网联通  
  
  
  
  
  
简述OSI模型与TCP/IP模型有什么不同” (网传, 签个, 协议, 山鸡37也可以变凤凰, 不用再 xie fu jie了)

一个有7层, 一个有4层

一个有3个重要概念, 协议服务接口(xie fu jie), 一个没有区分协议服务接口

一个产生在协议发明之前, 一个产生在协议发生之后

一个网络层有面向连接和无连接, 一个网络层仅有无连接

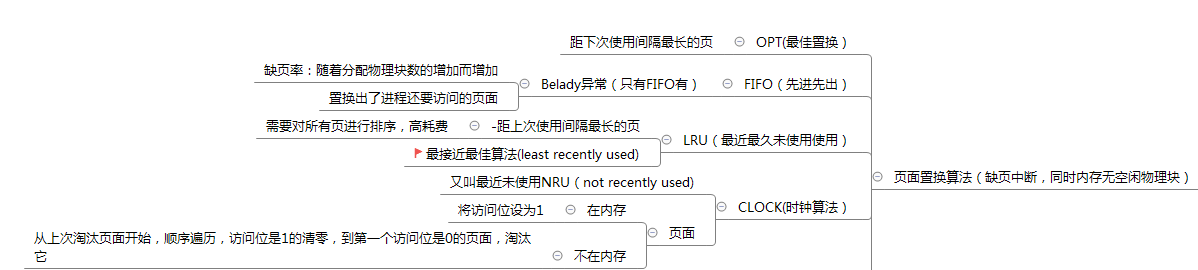
一个传输层仅有面向连接, 一个传输层有面向连接和无连接

奇偶校验区别

奇偶校验码都是由n-1位信息元和1位校验元组成. 如果是奇校验码, 在附加上一个校验元之后, 码长为n的码字中”1”的个数为奇数. 如果是偶校验码, 在附加一个校验元之后, 码长为n的码字中”1”的个数为偶数  
 **操作系统**

9虚拟内存 ：

基于局部性原理，可以将程序的一部分装入内存，而将其余部分留在外存。程序运行时, 当所访问的信息不在内存时，可以将暂时不用的信息调出内存, 将所需部分调入内存，然后继续执行程序，这样，系统好像为用户提供了一个比实际内存大得多的存储器，称为虚拟存储器。  
虚拟存储器特征：（1）多次性（2）对换性（3）虚拟性(多次对换达到虚拟)  
虚拟内存实现方式：请求分页存储管理，请求分段存储管理，请求段页式存储管理  
页面置换算法：FLOC废棉絮要置换掉



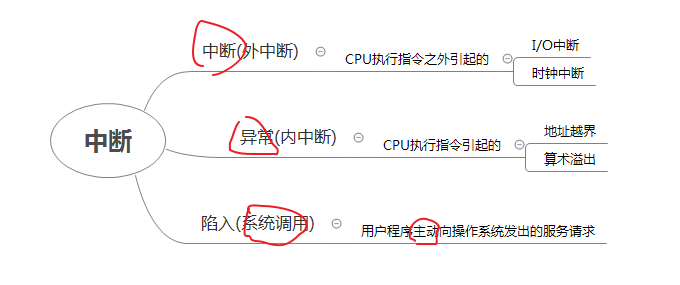
页面分配策略：固定分配局部置换 可变分配全局置换 可变分配局部置换

11 I/O控制方式  
（1）程序直接控制  
（2）中断驱动：允许i/o设备主动打断cpu的运行并请求服务，解放cpu，使得其向I/O控制器发送读命令后可以继续做其他工作  
（3）DMA：在i/o和内存之间直接开辟了一条通路，传送单位数据块  
（4）通道控制方式：把一个数据块改为一组数据块  
12 磁盘调度算法  
（1）FCFS(first come first service )先来先服务  
（2）SSTF (short seek time first)最短寻道时间优先  
（3）SCAN扫描算法 电梯算法  
（4）C-SCAN 循环扫描  
  
13系统调用：用户在程序中调用操作系统所提供的一些子功能。系统中的共享资源都由操作系统统一掌管，因此在用户程序中，凡是与资源相关的操作（存储分配，进行I/O传输以及管理文件），都须通过系统调用方式向操作系统提出服务请求，并由操作系统代为完成.

系统调用过程：操作系统接收到系统调用请求后，会让处理器进入内核模式, 而当处理完系统调用内容后，操作系统会让处理器返回用户模式，来执行用户代码

核心态: 操作系统关系程序执行时, 机器所处的状态, 有较高的特权, 可以执行包括特权指令的一起指令, 访问所有的寄存器和存储区

用户态: 用户程序执行时, 机器所处的状态, 有较低的特权, 只能执行规定的指令, 访问指定的寄存器和存储区  
14 中断过程  
中断请求 中断响应 断点保护 执行中断服务程序 断点恢复 中断返回  
17 中断隐指令(即硬件机制)：CPU在中断周期内由硬件自动完成的一条指令，其功能包括关中断, 保存PC(断点)，寻找中断服务程序入口地址并送至pc.



系统调用与子程序调用区别:系统调用是在核心态(即管态)下执行，需从用户态转换为核心态，而子程序调用在用户态(即目态)下执行的,没有状态变化.

中断和子程序调用的区别:

硬件+软件处理过程 软件处理过程

入口地址: 有中断隐指令根据中断向量得到, 有调用程序根据寻址方式得到

保存环境: 保存PC, PSW,通用寄存器, 保存PC,通用寄存器

进程状态: 从用户态转换为和心态, 没有状态变化

中断和系统调用的区别:

中断是由外设产生, 无意的, 被动的

系统调用是由应用程序请求操作系统提供服务产生, 有意的, 主动的

系统调用运行在核心态, 其通过中断机制, 来完后用户态到核心态的转换

14 程序链接方式：静态 装入时动态 运行时动态   
程序装入：绝对装入 可重定位装入 动态运行装入  
  
  
  
15 内存分配方式管理方法  
1.单一分配  
2.固定分区分配  
3.动态分区分配 （1）首次适应 （2）循环首次适应 最佳适应 最坏适应  
  
  
  
  
18 操作系统的层次结构   
硬件层  
操作系统层  
实用程序层：由一组操作系统控制下的实用程序组成  
应用层  
  
21数据库系统，操作系统，应用软件之间关系  
  
23分页和分段的区别：   
1.页是信息的物理单位，是系统的需要；段则是信息的逻辑单位，是为了满足用户的需要 2.页的大小固定且有系统决定，而段长度可变，取决于用户编写的程序   
3.分页的作业地址空间一维，单一线性地址空间；分段的作业地址空间是二维的，要给出段名和段内地址  
  
24管程：一组数据以及定义在这组数据之上对这组数据的操作组成的软件模块，这组操作能初始化并改变管程中的数据和同步进程。  
组成：1）局部于管程的共享结构数据说明  
      2）对该数据结构进行操作的一组过程  
      3）对局部于管程的共享数据设置初始值的语句  
特性：局部于管程的数据只能被局部于管程内的过程所访问  
      一个进程只能通过调用管程内的过程才能进入管程访问共享数据  
      每次仅允许一个进程在管程内执行某个内部过程。  
  
**数据库**

事务：一个单元的工作，要么全完成，要么全不做。

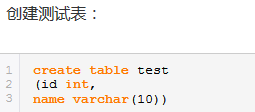
2 ACID:

原子性（Atomicity）整个事务中的所有操作，要么全部完成，要么全部不完成，不可能停留在某个中间环节  
一致性（consistency）在事务开始之前和事务结束之后，数据库的完整性约束没有被破坏  
隔离性(isolation)：两个事务的执行是互不干扰的，一个事务不可能看到其他事务运行时，中间某一刻的数据  
持久性(durablity)：在事务完成以后，该事务所对数据库所做的更改便持久的保存在数据库中，并不会回滚。

4数据库的armstrong 定理及其应用  
从已知的一些函数依赖，可以推导出另外一些函数依赖，这就需要一系列推理规则。这些规则常被称作“Armstrong 公理  
设U 是关系模式R 的属性集，F 是R 上成立的只涉及U 中属性的函数依赖集。函数依赖的推理规则有以下三条：  
1）自反律：若属性集Y 包含于属性集X，属性集X 包含于U，则X→Y 在R 上成立。(此处X→Y是平凡函数依赖)  
2）增广律：若X→Y 在R 上成立，且属性集Z 包含于属性集U，则XZ→YZ 在R 上成立。  
3）传递律：若X→Y 和 Y→Z在R 上成立，则X →Z 在R 上成立。  
Armstrong公理系统的有效性和完备性  
①Armstrong公理系统的有效性指的是：由R出发根据Armstrong公理系统推导出来的每一个函数依赖一定是R所逻辑蕴含的函数依赖。  
②Armstrong公理系统的完备性指的是：对于R所逻辑蕴含的每一函数依赖，必定可以由R出发根据Armstrong公理系统推导出来。  
根据上面三条推理规则，又可推出下面三条推理规则：  
④ 合并规则：若X→Y，X→Z，则X→YZ为F所蕴含；  
⑤ 伪传递规则：若X→Y，WY→Z，则XW→Z为F所蕴含；  
⑥ 分解规则：若X→Y，Z⊆Y，则X→Z为F所蕴含。

5 **什么是数据库中的索引**  
索引是对数据库表中,一列或多列的值进行排序的一种结构

主键: 一个列或多列的组合，其值能唯一地标识表中的每一行.



如何建主键: 

数据库索引分为，唯一索引、主键索引和聚集索引。  
聚簇索引 按数据存放的物理位置为顺序的

唯一索引, 不允许其中任何两行具有相同索引值的索引。  
  
主键索引(默认索引)

数据库主键，一个列或多列的组合，其值能唯一地标识表中的每一行. 主键索引是唯一索引的特定类型。该索引要求主键中的每个值都唯一

聚集索引 表中行的物理顺序与键值的逻辑（索引）顺序相同。(表中行的物理顺序与键值的逻辑顺序不匹配)。与非聚集索引相比，聚集索引通常提供更快的数据访问速度  
优点：   
加快数据的检索速度  
缺点：创建和维护索引要耗费时间, 且索引占物理空间

3 数据库模型：关系模型 层次模型 网状模型

6数据库的模式 内模式 外模式 模式  
1）模式．   
  模式又称概念模式或逻辑模式，对应于概念级。它是由数据库设计者综合所有用户的数据，按照统一的观点构造的全局逻辑结构，是对数据库中全部数据的逻辑结构和特征的总体描述，是所有用户的公共数据视图(全局视图)  
  2）外模式   
  外模式又称子模式，对应于用户级。它是某个或某几个用户所看到的数据库的数据视图，是与某一应用有关的数据的逻辑表示。  
  3）内模式   
  内模式又称存储模式，对应于物理级，它是数据库中全体数据的内部表示或底层描述，是数据库最低一级的逻辑描述，它描述了数据在存储介质上的存储方式翱物理结构，对应着实际存储在外存储介质上的数据库  
8关系数据库都有那些操作,特点是什么?    
◇查询:选择、投影、连接、除、并、交、差  ◇数据更新:插入、删除、修改    
关系操作的特点:集合操作方式,即操作的对象和结果都是集合。  
9数据库的几个键，几个码  
主码：主关键字(primary key)是表中的一个或多个字段，它的值用于惟一地标识表中的某一条记录。

1. 范式(NF Normal Form): 一张数据表中的表结构所符合的某种设计标准的级别.

**关系数据库: 采用的关系模型来组织数据的数据库. (**简单的说, 用二维表来保存数据**)**

“关系模式”和“关系”的区别，类似于面向对象程序设计中”类“与”对象“的区别。”关系“是”关系模式“的一个实例，你可以把”关系”理解为一张带数据的表，而“关系模式”是这张数据表的表结构

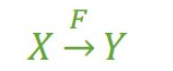
第一范式: **符合1NF的关系中的每个属性都不可再分(所有关系型数据库的最基本要求)**

**码：关系中的某个属性或者某几个属性的组合，**之外的所有属性都完全函数依赖于它,**用于区分每个元组**（可以把“元组”理解为一张表中的每条记录，也就是每一行）**。(**通常可以理解为：**假如当 K 确定的情况下，该表除 K 之外的所有属性的值也就随之确定，那么 K 就是码**一张表中可以有超过一个码。（实际应用中为了方便，通常选择其中的一个码作为**主码**）)

**非主属性** 包含在任何一个码中的属性成为主属性

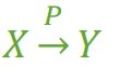
**第二范式: 关系中的每个属性都不可再分(1NF)属于第一范式，同时消除了非主属性对于码的部分函数依赖**

**完全函数依赖(full): Y函数依赖于x, 但是x的任一真子集(B是A的子集, A中至少一个元素不属于B, B为真子集)都不能唯一确定y的值.**

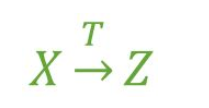


部分函数依赖(partly): Y函数依赖于x, 但是不完全依赖于x,



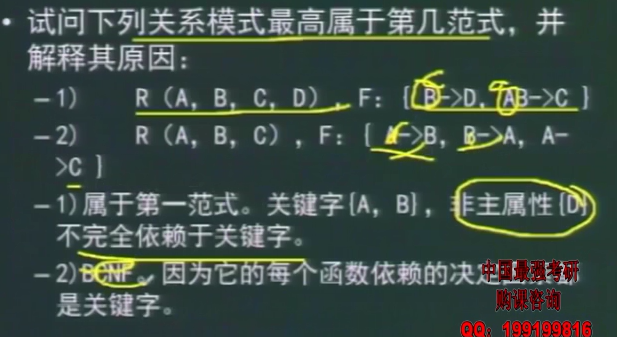


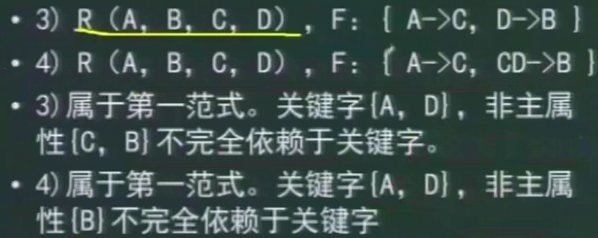
传递函数依赖(transfer): z函数依赖于Y, Y函数依赖于x, 则z函数依赖于x



第三范式: 关系中每个属性不可再分(1NF), 消除了非主属性对码的部分函数依赖(2NF), 同时消除了非主属性对码的传递函数依赖

巴斯-科德范式(**BC范式**): 关系中每个属性不可再分, 消除了**所有属性(非主属性和主属性)**对于码的部分与传递函数依赖。





**计算机组成**  
1冯诺依曼计算机  
1)计算机硬件系统由运算器，控制器，存储器，输入和输出5大部分组成。  
2)指令和数据以同等地位存于存储器中，并按地址寻访。  
3）指令和数据均用二进制代码表示。  
4）指令由操作码和地址码组成，操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置。  
5）指令在存储器内按顺序存放，通常，指令是顺序执行的，在特定条件下，可根据运算结果或根据设定的条件改变执行顺序。  
6）早起的冯诺依曼计算机以运算器为中心，输入/输出设备通过运算器与存储器交换信息。  
2cisc &risc  
CISC（Complex Instruction Set Computer）：复杂指令系统计算机  
1）指令系统复杂庞大，指令长度不固定，指令格式多，寻址方式多  
2）可以访存的指令不受限制，各种指令使用的相差频度大。执行时间相差大  
3）采用微程序控制，难以用优化编译生成高效的目标代码程序。  
RISC(reduced Instruction Set Computer):精简指令系统计算机  
1）选取使用频率最高的一些简单指令，复杂指令由简单指令构成  
2）指令长度固定，指令格式种类少，寻址方式种类少   
3）只有取数存数指令访存  
4）cpu中通用寄存器数量多  
5）采用流水线  
6）硬布线 特别注意编译优化  
  
3中断过程  
关中断 保存断点 中断服务程序寻址 保存现场和屏蔽字 开中断 执行中断服务程序 关中断 恢复现场和屏蔽字 开中断 中断返回  
  
4寻址方式  
常见的数据寻址方式  
1）隐含寻址：规定在累加器ACC  
2）立即寻址：  
3）直接寻址  
4）立即寻址  
5）寄存器寻址  
6）寄存器间接寻址  
7）相对寻址（将程序计数器pc的值加上形式地址A）  
8）基址寻址（CPU中基址寄存器（不可变）加上指令格式中的形式地址A（可变））  
9）变址寻址（形式地址A（不变）与变址寄存器的内容之和）  
10）堆栈寻址

 2.什么叫做程序？

适合于计算机处理的语句序列。

面向过程程序：算法+数据结构    程序最小单元是：函数/接口

面向对象程序：对象+消息                ：类

面向构件程序：构件+构架                ：构件

3.什么是对象？通信方式？

对象是对问题域中某个或某一类实体的抽象。

通信方式：消息

7.什么是类？实例？消息？方法？属性？

类是对具有相同数据和相同操作的一组相似对象的定义。类是对具有相同属性和行为的一个和多个对象的描述，通常在这种描述中也包括对怎样创建该类的新对象的说明。

实例就是由某个特定的类所描述的一个具体的对象。

消息就是要求某个对象执行在定义它的那个类中所定义的某个操作的规格说明。通常，一个消息由3各部分组成：接收消息的对象，消息选择符，零个或多个变元。

方法就是对象所能执行的操作，也就是类中所定义的服务。方法描述了对象执行操作的算法，相应消息的方法。在C++语言中吧方法称为成员函数。

属性就是类中所定义的数据，它是对客观世界实体所具有的性质的抽象。类的每个实例都有自己特有的属性值。

8.接口(功能)和类的区别？

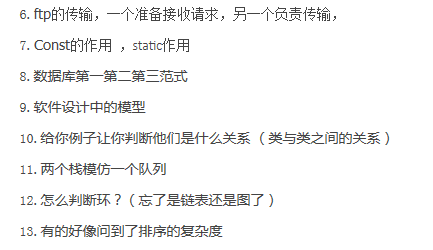
分别是面向过程程序和面向对象程序的最小程序单位。

TLB; translation look aside buffer高速缓冲存储器, 又称联想存储器, 也称快表技术.存放进程正在访问的页表项,为了加快地址映射速度.

有并行查找能力: 多 CPU 让查询更快的查询

呀，你是做java的，讲一下java反射机制的底层原理

反射让Java具有了动态的特性，   
 反射的核心是JVM在运行时才动态加载类或调用方法/访问属性，它不需要事先（写代码的时候或编译期）知道运行对象是谁。



网卡的几种工作模式:

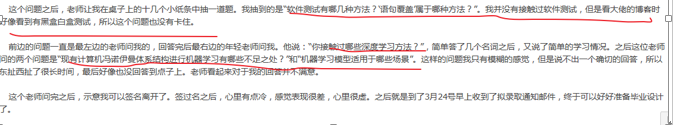
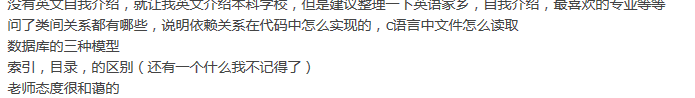
广播模式(broad cast model): 接收广播传送帧. 物理地址(MAC) 0xFFFFFF,

多播模式(multicast model): 接收多播传送帧

以多播传送地址为目的物理地址可以被组内其他主机接收, 而组外主机接收不到,,但把网卡设为多播模式, 可以接收所有多播传送帧, 不管它是不是组内成员

直接模式(direct model): 只接收目地址是自己 Mac地址的帧

混杂: 接收所有的流过网卡的帧



没有配置软件的计算机是裸机

系统软件: 负责管理计算机系统中各种独立的硬件，使得它们可以协调工作,如WINDOWS、UNIX(操作系统)、[汇编语言](https://baike.baidu.com/item/%E6%B1%87%E7%BC%96%E8%AF%AD%E8%A8%80)[汇编器](https://baike.baidu.com/item/%E6%B1%87%E7%BC%96%E5%99%A8)，C语言编译、连接器(语言处理程序)

应用软件: 为了某种特定的用途而被开发的软件,如爱奇艺, 优酷

操作系统属于系统软件

操作系统：控制整个计算机系统的硬件和软件资源，合理的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

操作系统作用: 处理器管理, 存储器管理, 文件管理, 设备管理, 提供用户接口

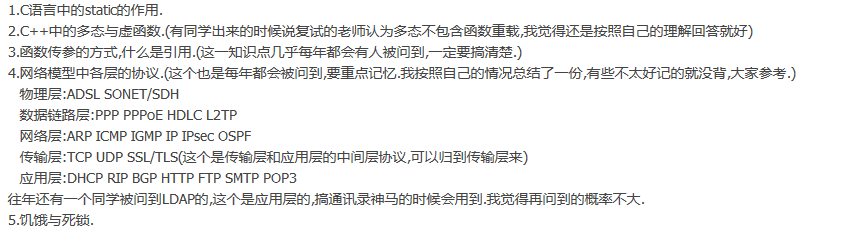
**数据库系统，操作系统，应用软件之间关系**

**核心是操作系统, 其之上式数据库系统, 再上是应用软件**

**最底层是硬件, 之上是操作系统(管理硬件资源),其他系统软件运行在操作系统的基础上, 再之上是应用程序**

透明, 指的是看不见

什么东西符合”透明”特征: 数据链路层帧的传输, 虚拟存储器对程序员透明, 分页对程序员透明, 联想寄存器的地址变换对程序员透明, 交换对程序员透明(覆盖不是)



Internet协议(簇): HTTP, FTP, SMTP, POP3

应用层协议: DNS(domain name system域名解析系统), FTP(file transfer protocol 文件传输协议), HTTP(超文本传输协议), 电子邮件(SMTP简单邮件传输协议simple mail transfer protocol, POP3邮局协议post office Protocol), 路由信息协议RIP（Routing Information Protocol）,BGP边界网关协议Border Gateway Protocol

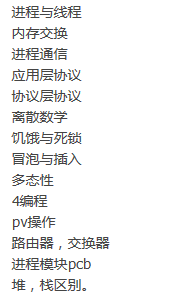
DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）

传输层协议: TCP, UDP, :传输层安全协议SSL/TLS(Transport Layer Security)

网络层协议: ARP（Address Resolution Protocol地址解析协议）, ICMP（Internet Control Message Protocol）Internet控制[报文](https://baike.baidu.com/item/%E6%8A%A5%E6%96%87)协议, IP, OSPF(Open Shortest Path First[开放式最短路径优先](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E5%BC%8F%E6%9C%80%E7%9F%AD%E8%B7%AF%E5%BE%84%E4%BC%98%E5%85%88)）

数据链路层: PPP协议(point-to-point protocol), HDLC(High-Level Data Link Control)

CSMA, CSMA/CD, CSMA/CD, ALOHA



数据链路层功能

为网络层提供服务: 无连接无确定服务, 无连接有确定服务, 有连接有确定服务(连接即发生关系, 要对女同志负责)

链路管理: 负责数据链路层连接的建立,维持和释放,用于面向连接的服务

帧同步: 让接收方可以确认比特流中一帧的开始位置和结束位置

透明传输: 使得不管什么样的比特组合, 都能在链路上传送

流量控制: 调节发送方的发送速率, 使其不超过接收方的接受能力

差错控制: 使接收方确定接受的数据就是由发送方发送的数据

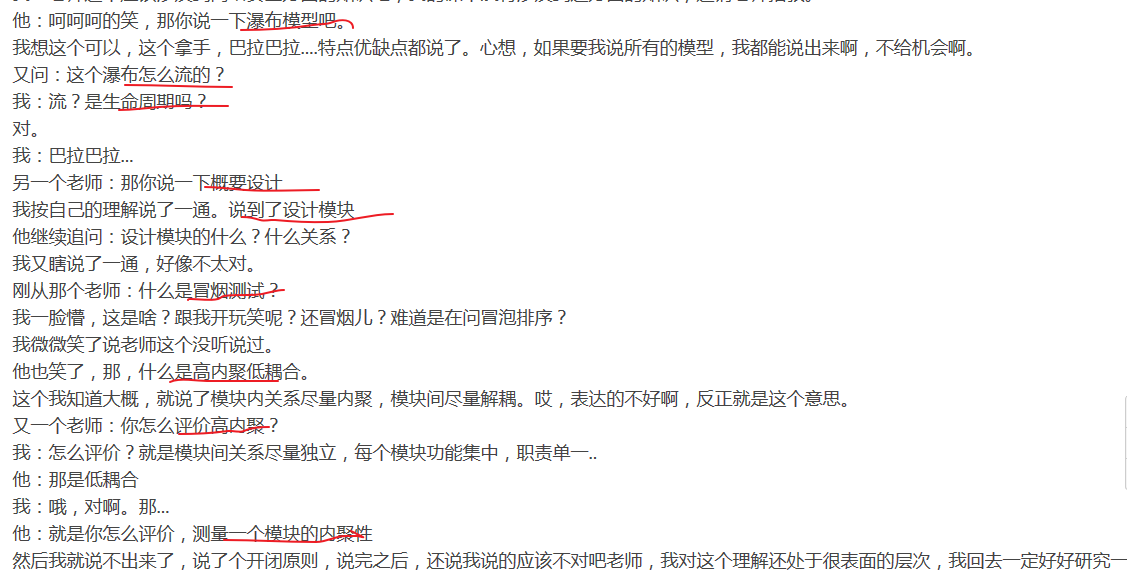
数据链路层四种组帧方法:

字符计数法

字节填充的首尾界符法,ASCII码128个,可输入字符95个,从剩下33中选两个来充当帧开始符和结束符

比特填充的首尾界符法: 以01111110为开始和结束, 每5个1后加一个0

物理编码违例法



**OS**

2chapter ProcessManagement

进程和线程(thread)的区别

线程是独立调度的基本单位, 本身没有资源, 进程是资源分配的基本单位, 同一进程中的线程共享进程的地址空间(放着各种资源: 进程代码段, 全局变量, 打开的文件), 但是, 一个线程的栈指针对其他线程透明(心里的客栈, 只对自己开放)

透明: 即看不到其存在.

2.进程与程序的区别  
进程是一个程序对某个数据集的一次运行活动. 进程是动态的, 程序时静态的.

进程有一定的生命周期，程序是一组代码的集合，进程可以创建进程，程序不能创建程序

进程由程序段+数据段+PCB(进程控制块process control block): 一个数据结构，进程存在的唯一标志. 系统通过PCB来了解进程的状态信息,以便控制和管理

3 作业和进程的区别  
一个进程是一个程序对某个数据集的执行过程，是分配资源的基本单位。  
作业是用户需要计算机完成的某项任务，是要求计算机所做工作的集合

4, 并行和并发的区别

并行(并排行走): 两个或多个事件在同一时刻发生

并发(病发, 一块全来就死了): 两个或多个事件在同一时间间隔内发生,同一时刻只处理一个任务, 通过分时来实现

进程中的通信

(低级通信方式PV操作)

高级通信方式: 共享存储, 消息传递, 管道通信(管道是一个共享文件)

PCB中有两个标识符, PID(进程标识符process identification)，用户标识符(UID).

**进程控制块英文process contral block**，进程存在的唯一标志, 操作系统通过PCB 表来管理和控制进程.

在多道程序环境下扩充内存可用(逻辑上）：覆盖+对换+虚拟存储器技术

覆盖：进程内

把暂时用不到的程序部分调到外存，把需要运行的程序换入，把用户空间分为固定区，覆盖区（类似固定分区存储管理），可用于单一连续存储管理和固定分区管理

内存交换(对换)/对换(交换)：多个进程之间

把处于等待状态的程序从内存中移出, 把准备好竞争CPU的程序移入内存.

虚拟内存: 其实不存在, 系统通过部分装入, 请求调用和置换从逻辑上扩充了内存容量.

其大小有计算机的地址结构决定

1. PV操作是什么  
   我把PV操作的实现原理和使用过程说了一遍,老师说我说的不准确.事后翻了翻书,估计老师想要我回答信号量.

信号量S是一个整数，S大于等于零是代表可使用的资源数目，当小于零时则其绝对值为在该信号量上等待的进程个数。

我们通过信号量机制, 来解决进程同步和互斥的问题.而pv操作是可以访问信号量的两个原语. P[原语](https://baike.baidu.com/item/%E5%8E%9F%E8%AF%AD)操作的动作是：S减1；若S减1后仍大于或等于零，则进程继续执行；若S减1后小于零，则该进程被阻塞后进入等待队列。

V原语操作的动作是：S加1；若相加结果大于零，则进程继续执行；若相加结果小于或等于零，则从待队列中唤醒一等待进程.

PV操作对于每一个进程来说，都只能进行一次，而且必须成对使用

原语: 能实现某种功能, 且不能被中断执行的程序段.

OSI/RM开放系统互连参考模型的全称  
Open System Interconnection Reference Model..

死锁: 两个或多个进程无限等待一个事件, 而该事件只能由这些等待进程之一来产生

死锁例子: 哲学家就餐问题. 一张桌子, 5个人, 每两个人之间放一根筷子, 为了吃饭,每个人拿起自己右手边的筷子, 此时陷入了死锁.

饥饿: 进程等待永远不会分配给自己的资源.

相同点: 都是因为竞争资源引起的

不同点: 死锁至少涉及两个进程, 一定有循环等待, 且等待永远不会释放的资源

饥饿可以只涉及一个进程, 不一定有循环等待, 等待会释放但不会给自己的资源.

`

死锁产生的必要条件: (在)循环(中), 互请(求资源)(而)不(得)

互斥条件: 在一段时间内, 某种资源仅被一个进程所占有

不剥夺条件: 进程所占有的资源在为使用完毕前, 不能被其他进程强行夺走.

请求和保持条件: 进程在等待分配新资源时, 继续占有已经分配到的资源

循环等待条件: 存在进程资源循环等待链, 链中的每一个进程已经获得的资源同时被链中的下一个进程所请求.

避免死锁方法:

静态策略(预防死锁), 破坏产生死锁的四个必要条件中的一个或几个

预先静态分配法破坏请求和保持条件, 资源剥夺法破坏不剥夺条件, 资源的按序分配破坏循环等待条件

动态策略(避免死锁), 在资源动态分配时,用银行家算法防止系统进入不安全状态.

安全状态, 在某一时刻, 系统能按某种顺序为每个进程分配其所需资源, 直至最大需求, 使每个进程都可以顺利完成的状态

不安全状态, 不存在这样的安全序列.

死锁的解除：资源剥夺法，撤销进程法

10.什么是局部性原理？

大多数程序执行时, 在一个较短的时间内仅使用程序代码的一部分, 相应地, 程序所访问的存储空间也局限于某个区域.

局部性原理又表现为：时间局部性和空间局部性。  
　　时间局部性: 一条指令的一次执行和下次执行都集中在一个较短的时间内

空间局部性: 当前指令和邻近的几条指令都集中在一个较小的区域内

进程调度算法:

先来先服务: FCFS

短进程优先SPF

优先级调度算法

高响应比优先调度算法

多级反馈队列调度算法

**C, C++, java**

C语言中的static的作用.

让一个变量长期有效，而不管在什么地方被申明, 相当于一个全局变量(被调用后值会保存)

避免多个文件使用了相同的变量名而导致冲突. 在不同文件的相同的全局变量申明前加上static修饰符。系统就为他们分配不同的内存，互不影响了.

C语言中Const的作用

一个变量被const修饰, 意味着只可以被访问, 不可以修改

数组与指针：数组名就是这块连续内存单元的首地址

多级指针：当指针变量所指向的也是一个指针变量时，这个指针就是一个多级指针(地址的地址)

5  构造方法(有参, 无参), 在创建对象时初始化对象 public 类名() {}

12指针和引用的区别  
★ 相同点：  
都是地址的概念；  
指针指的是某块内存的地址；引用是一个变量的别名。  
★ 区别：

1.从内存分配上看：程序为指针变量分配内存区域，而引用不需要分配内存区域。

1. 引用不能为空，指针可以为空；
2. 引用只能在定义时被初始化一次，之后不可变；指针可变；

4. 指针和引用的自增(++)运算意义不一样；  
5. “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；

6. 引用没有 const，指针有 const；

7. 引用使用时无需解引用(\*)，指针需要解引用；

8. 指针是一个实体，而引用仅是个别名；

函数传参方式有哪些？

C函数调传参：传递值、传递指针（传递的是地址）

C++函数传参：传递值、传递指针、传递引用(传指针和引用都是传地址)

Java中函数传参方式: 传递值, 传递引用

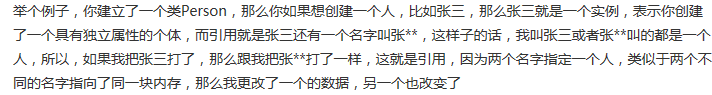
传递值: 形参是实参的一个副本, 形参的改变不影响实参

传递指针: 传入地址, 形参实参都是地址, 形参是实参的副本(值传递的一种), 通过形参间接改变实参所指向的变量的值.

传递引用: 形参相当于实参的别名，改变形参就是改变实参（引用就是起一个别名）

什么是引用

引用是一个变量的别名, 对引用的操作等于对其指定的对象进行操作.



传递指针和传递引用的区别

使用引用传递函数的参数，在内存中并没有产生实参的副本，它是对实参操作； 使用指针作为函数的参数要给形参分配存储单元

引用就是给对象起个别名, 引用的地址就是该对象的地址, 对引用的操作等于对其指定的对象进行操作

引用就是给变量起个别名, 引用的地址就是该变量的地址, 对引用的操作就是对变量的操作.

传入指针的拷贝, 在内存中开辟空间存储变量的地址, 此时, 形参的地址是变量地址的地址,

形参的地址就是变量的地址, 对引用的操作就是对变量的操作.

形参的地址是变量地址的地址, 需另从内存中开辟空间,

(形参和实参是指针)传入实参(指针)的副本, 形参的地址和实参地址不同, 需另开辟内存空间

(形参和实参是变量)形参的地址就是实参的地址, 对形参的操作就是对实参的操作

j**ava和C++有什么区别**

1）内存管理：java具有内存自动回收机制，而C++没有；

2）Java不在所有类之外定义全局变量,在某个类中定义公用静态的变量来完成C++中全局变量的功能；

3）异常处理，java使用try-catch-finally处理异常；C++ 异常处理涉及到三个关键字：try、catch、throw。

4）Java不支持头文件，面C和C++语言中都用头文件来定义类的原型,全局变量,库函数等；Java不支持宏定义,而是使用关键字final来定义常量,在C++中则采用宏定义来实现常量定义；

C++ [宏定义](http://baike.baidu.com/view/2076445.htm)将一个[标识符](http://baike.baidu.com/view/390932.htm)定义为一个字符串，[源程序](http://baike.baidu.com/subview/546605/546605.htm)中的该标识符均以指定的字符串来代替。前面已经说过，[预处理命令](http://baike.baidu.com/view/1334643.htm)不同于一般C++语句。因此预处理命令后通常不加分号。这并不是说所有的预处理命令后都不能有分号出现。由于宏定义只是用宏名对一个字符串进行简单的替换，因此如果在宏定义命令后加了分号，将会连同分号一1起进行置换。

5）Java对每种数据类型都分配固定长度，java具有跨平台可移植性，而C++则有限制；

6）类型转换不同.在C和C++中,可通过指针进行任意的类型转换,常常带来不安全性,而在Java中,运行时系统对对象的处理要进行类型相容性检查,以防止不安全的转换；

7）结构和联合的处理.在C和C++中,结构和联合的所有成员均为公有,这就带来了安全性问题,而在Java中根本就不包含结构和联合,所有的内容都封装在类里面；

8）Java不再使用指针，指针是C和C++中最灵活,也最容易产生错误的数据类型.由指针所进行的内存地址操作常会造成不可预知的错误,同时通过指针对某个内存地址进行显式类型转换后,可以访问一个C++中的私有成员,从而破坏安全性.而Java对指针进行完全地控制,程序员不能直接进行任何指针操作.

**Java有指针吗？为什么？**

9虚函数：在某[基类](https://baike.baidu.com/item/%E5%9F%BA%E7%B1%BB)中声明为 virtual, 允许被一个或多个[派生类](https://baike.baidu.com/item/%E6%B4%BE%E7%94%9F%E7%B1%BB)中重写的[成员函数](https://baike.baidu.com/item/%E6%88%90%E5%91%98%E5%87%BD%E6%95%B0)

  Virtual 函数类型 函数名

作用是在基类中为其派生类保留一个函数的名字，以便派生类重写, 实现多态

1. 抽象类：带有纯虚函数的类是抽象类 抽象类只能作为基类来使用，抽象类无法具体实例化。  
         抽象类不可以实例化(如动物, 实例化成什么样?不知道)，即不能创建一个抽象类的对象，但是可以声明一个抽象类的指针和引用，通过指针和引用，就可以指向并访问派生类的对象，进而访问派生类成员。  
   函数重载和运算符重载是一种静态多态，虚函数是动态多态
2. 在同名函数中寻找匹配函数的过程称为关联  
     
   10重载和重写  
   重载：从overload翻译过来，是指同一可访问区内被声明的几个具有不同参数列（参数的类型，个数，顺序不同）的同名函数，根据参数列表确定调用哪个函数，重载不关心函数返回类型。  
   重写：翻译自override，也翻译成覆盖（更好一点），是指派生类中存在重新定义的函数。其函数名，参数列表，返回值类型，所有都必须同基类中被重写的函数一致。只有函数体不同（花括号内），派生类调用时会调用派生类的重写函数，不会调用被重写函数。重写的基类中被重写的函数必须有virtual修饰。
3. 类包含的内容  
   属性：这个类的数据成员  
   方法：这个类的成员函数

6 接口和类的区别  
（1）接口不能实例化，类可以实例化  
（2）接口没有构造函数，类有构造函数  
（3）接口不能进行运算符重载，类可以运算符重载  
（4）接口的成员没有任何修饰符，成员都是公共的，类的成员有修饰符  
（5）接口的成员没有执行方式，它只是方法，属性，事件和索引的组合，类还可以有别的成员（如字段

13多重继承：类A可以继承类B, 类B又可以继承类C,如此

   单继承：一个子类最多只有继承一个父类

多继承: 一个子类可以继承多个父类.

2.C++中的多态与虚函数.(有同学出来的时候说复试的老师认为多态不包含函数重载,我觉得还是按照自己的理解回答就好)

11死代码： 1. 执行不到的代码. 2. 执行得到， 但没有任何作用的代码。就是“不产生实际作用”的代码  
死代码消除: **Dead code elimination(DEC),** [编译器](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8)可以判断出某些代码不影响输出，会消除这些代码。还可以使用优化作用进行消除。

通过AQtime覆盖率的分析就可以发现所有的永远都不会执行到的代码，但是与程序无关的代码却不一定能够全部发现。  
注释算死代码

1. 左手男老师2：学过java吧？容器/集合: 专门用来存放其他类的对象的一个类  
   男老师2：**EJB** (Enterprise JavaBean)**,** JavaEE服务器端[组件模型](https://baike.baidu.com/item/%E7%BB%84%E4%BB%B6%E6%A8%A1%E5%9E%8B), 即是把已经编写好的程序（即：类）打包放在服务器上执行  
   我：这个我知道，它又分为实体Bean和会话Bean  
   男老师2：那你说说**实体Bean和会话Bean的区别**

实体bean 实实在在的东西，比如说一本书，有书名，编号，多少页等等。没有逻辑性。可以被持久化的, 好比这本书应该放在图书馆里！   
  
会话bean 指一次会话。客户与店家的对话想买哪本书, 有逻辑性的。 它的逻辑性最终决定了会从书库里拿到哪一本书。不能持久化（你怎么可以把顾客的对话保存到图书架上呢？不合真实情况）

EJB容器可以接受三类EJB

* 1. 会话Bean（Session Beans）
  2. 实体Bean（Entity Beans）
  3. 消息驱动Bean（Message Driven Beans ，MDBs）

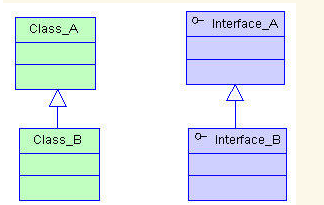
用过的**软件开发工具有哪些**，sublime Text3, EditPlus, ecplise。

c++, java都是面向对象语言, 一切操作以对象为基础, 对底层操作不是很方便.

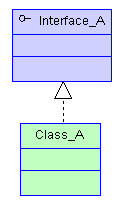
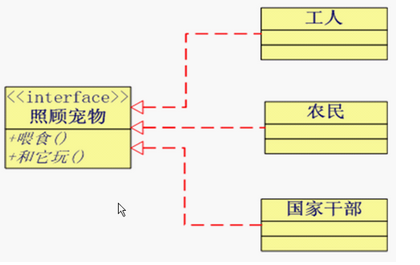


类(对象)间关系(UML图（unified modeling language）统一建模语言),及其具体实现(举例子)

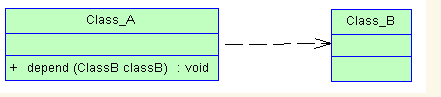
1. 继承关系 是一个类（称为子类、子接口）继承另外的一个类（称为父类、父接口）的功能，并可以增加它自己的新功能的能力。在Java中继承关系通过关键字extends明确标识



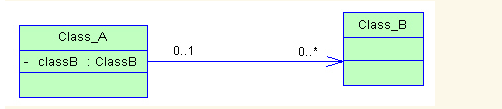
泛化: 泛化和继承其实是一个逆过程 泛化就是有子类抽象出一个父类 而继承就是由父类具体化一个子类

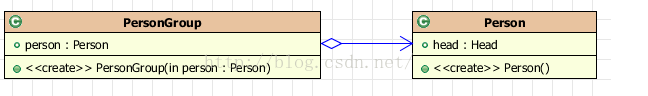
二、实现关系 一个类实现接口（可以是多个）的**功能**。Java中通过关键字implements明确标识。  
   
三、依赖关系(人和书) 一个类A使用到了另一个类B，而这种使用关系是具有偶然性的、临时性的、非常弱的，但是类B的变化会影响到类A。在代码层面如何实现?

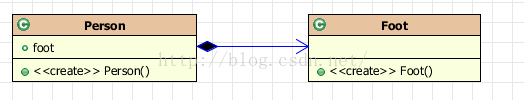
类B作为参数被类A在某个方法中使用

  
四、关联关系(老师和学生) 一种强依赖关系.在代码层面如何实现?

为被关联类B以类的属性形式出现在关联类A中，也可能是关联类A引用了一个类型为被关联类B的全局变量。

  
五、聚合关系(聚在一起,大难临头各自飞,不共死) 关联的一种特例，体现整体与部分的关系，如大雁和雁群.(体现**has-a**的关系)

  
六、组合关系(同生共死) 关联的一种特例，体现整体与部分间的关系,单比聚合更强, 此时整体和部分不可分, 如大雁和翅膀.( 体现一种 **contains-a**的关系)



强弱程度依次为：组合>聚合>关联>依赖。

`