

南大复试面试题答案汇总

南大复试面试最重要的就是准备下自己在本科做过的项目，比赛，老师一般都会问这些，如果是跨专业的可能会问一些专业性的知识（一般都是书本上的），以下内容是自己根据以前王道上的面试题，加上自己搜集的答案，有不完善的地方还请大家多多改善。希望能对进入南大复试的考生有所帮助。予人玫瑰，手有余香。

By lanyeqiao

2017.3.24

1. cache 的原理和思想

Cache 是一种小容量的高速缓冲存储器，由快速的 SRAM 组成，直接制作在 CPU 芯片中。其根据程序访问的局部性，将主存中活跃的程序块和数据块复制到 cache 中，CPU 直接从 cache 中读取数据，而不必访问低速的内存

Cache 的评价标准：命中率，访存时间

改进方案：选用较好的替换算法，扩大 cache 的容量，采用多级 cache 和数据指令 cache 分类策略。

用到 cache 的思想有：

基地址寄存器，将经常使用的基地址放在一个高速寄存器中，存储器的分层结构。

2. 操作系统中的文件组织

文件组织有(文件的逻辑结构)：有结构的记录文件和无结构的流文件

还可以分为：顺序文件，索引文件，索引顺序文件

操作系统的文件管理（物理结构）：

1. 顺序结构：一种简单的物理结构，将一个逻辑文件存储在外存连续的物理块中
2. 链接结构：将逻辑上连续的文件存放在外存不连续的区域上。通过链表的方式将同一个文件串成一个链表。
3. 索引结构：索引结构将一个逻辑文件的信息存放在外存的若干个物理块中，并为每个文件建立一个索引表，索引表中的每个表目存放文件信息所在的逻辑块号中和与之对应信息存放在的逻辑结构和与之对应的物理块号中，以索引结构存放的文件称为索引文件。

文件的存储空间管理：空闲表和空闲链表，还有位示图

4. 线性表

线性表中数据元素之间的关系是一一对应的关系，即除了第一个和最后一个数据元素之外，其

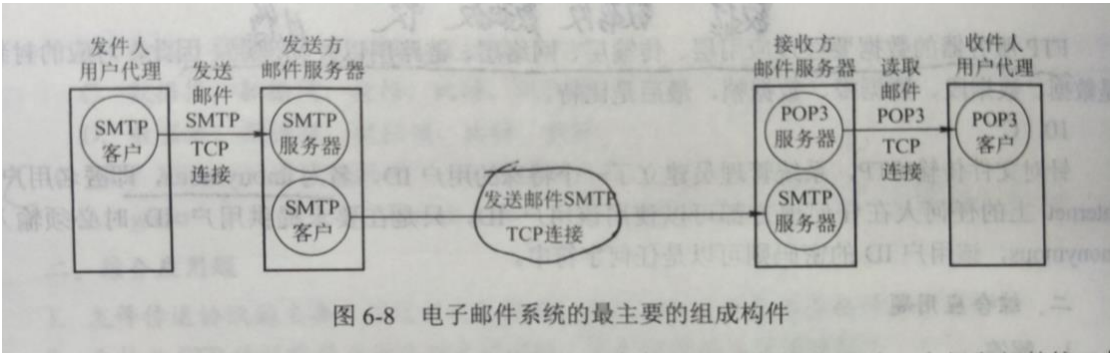
它数据元素都是首尾相接的。在数据结构逻辑层次上细分，线性表可分为一般线性表和受限线性表。一般线性表也就是我们通常所说的“线性表”，可以自由的删除或添加结点。受限线性表主要包括栈和队列，受限表示对结点的操作受限制。

用到了栈的机制：函数的调用，每调用一次，将程序现场入栈保存。程序的递归调用。

5. 电子邮件协议

常用的电子邮件协议有 **SMTP**（简单邮件传输协议）、**POP3**（邮局协议）、**IMAP4**（互联网信息访问协议），它们都隶属于 **TCP/IP** 协议簇。

收发机制：



- 1) 发信人调用用户代理来撰写或编辑要发送的邮件，用户代理用 **SMTP** 协议把邮件传送给发送方的邮件服务器
- 2) 发送发邮件服务器将邮件发送到接收方的邮件服务器中。
- 3) 接收方发现自己的邮件服务器中有未读的邮件，则通过 **POP3** 协议将邮件拉取到收件人用户代理中。

6. P2p 技术

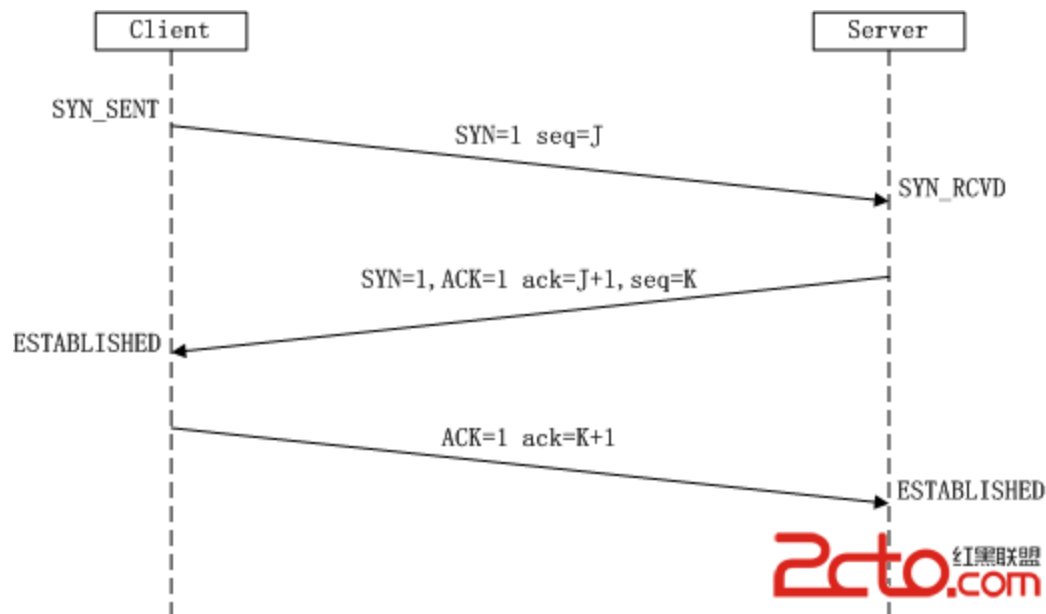
P2p 是点对点技术。网络中的每个主机既是服务器也是客户端。不同于 **C/S** 模式，其没有固定的服务器。纯点对点网络没有客户端或服务器的概念，只有平等的同级节点，同时对网络上的其它节点充当客户端和服务端。这种网络设计模型不同于客户端-服务器模型，在客户端-服务器模型中通信通常来往于一个中央服务器。

7. 握手问题

（离散数学）握手定理：任何无向图中所有顶点的度数之和等于变数的 2 倍。

三次握手：

所谓三次握手 (**Three-Way Handshake**) 即建立 **TCP** 连接，就是指建立一个 **TCP** 连接时，需要客户端和服务端总共发送 3 个包以确认连接的建立。在 **socket** 编程中，这一过程由客户端执行 **connect** 来触发，整个流程如下图所示：



(1) 第一次握手: Client 将标志位 SYN 置为 1, 随机产生一个值 $\text{seq}=\text{J}$, 并将该数据包发送给 Server, Client 进入 SYN_SENT 状态, 等待 Server 确认。

(2) 第二次握手: Server 收到数据包后由标志位 $\text{SYN}=1$ 知道 Client 请求建立连接, Server 将标志位 SYN 和 ACK 都置为 1, $\text{ack}=\text{J}+1$, 随机产生一个值 $\text{seq}=\text{K}$, 并将该数据包发送给 Client 以确认连接请求, Server 进入 SYN_RCVD 状态。

(3) 第三次握手: Client 收到确认后, 检查 ack 是否为 $\text{J}+1$, ACK 是否为 1, 如果正确则将标志位 ACK 置为 1, $\text{ack}=\text{K}+1$, 并将该数据包发送给 Server, Server 检查 ack 是否为 $\text{K}+1$, ACK 是否为 1, 如果正确则连接建立成功, Client 和 Server 进入 ESTABLISHED 状态, 完成三次握手, 随后 Client 与 Server 之间可以开始传输数据了。

8. 抽象数据类型

抽象数据类型(Abstract Data Type 简称 ADT)是指一个数学模型以及定义在此数学模型上的一组操作。ADT 包括数据数据元素, 数据关系以及相关的操作。

即 ADT

```
{
    数据对象: (数据元素集合)
    数据关系: (数据关系二元组结合)
    基本操作: (操作函数的罗列)
}
```

抽象数据类型 (ADT) 是一个实现包括储存数据元素的存储结构以及实现基本操作的算法。在这个数据抽象思想中, 数据类型的定义和它的实现是分开的, 这在软件设计中是一个重要的概念。这使得只研究和使用它的结构而不用考虑它的实现细节成为可能。

优先级队列的实现: 可以使用最大堆或者最小堆来实现, 优先级最高的在堆顶处。或者用链表, 但是需要每次对元素的优先级进行排序。

9. 什么是计算机? 语法, 语义

计算机（computer）俗称电脑，是现代一种用于高速计算的电子计算机器，可以进行数值计算，又可以进行逻辑计算，还具有存储记忆功能。是能够按照程序运行，自动、高速处理海量数据的现代化智能电子设备。由硬件系统和软件系统所组成，没有安装任何软件的计算机称为裸机。可分为超级计算机、工业控制计算机、网络计算机、个人计算机、嵌入式计算机五类，较先进的计算机有生物计算机、光子计算机、量子计算机等。

词法：定义组成程序的单词

语法：描述了该语言的程序的正确形式

语义：定义了程序的含义

10. 拉格朗日定理及其证明

群论中的拉格朗日定理

设 G 是有限群， H 是 G 的子群， $[G:H]$ 是 H 在 G 中的指数--即陪集个数。

那么我们有 $[G:H] | H| = |G|$ 即 H 的阶整除 G 的阶。

这里 $|G|$ 是群的阶数，即元素个数。

证明：设 G 和 H 的元数分别为 n 和 r ，设 H 有 s 个右陪集，但 G 等于所有右陪集的并集，不同的右陪集没有公共元素，而且，每个右陪集的元数等于 H 的元数 r ，一共是 s 个右陪集，故所有右陪集的并集有元数 rs ，它等于 G 的元数 n ： $n=rs$ ，或者说， r 整除 n ，商为 s 。

11. 数据库的查询优化有哪些？

- 1.对查询进行优化，要尽量避免全表扫描，首先应考虑在 `where` 及 `order by` 涉及的列上建立索引。
- 2.应尽量避免在 `where` 子句中对字段进行 `null` 值判断，否则将导致引擎放弃使用索引而进行全表扫描，如：
- 3.应尽量避免在 `where` 子句中使用 `!=` 或 `<>` 操作符，否则将引擎放弃使用索引而进行全表扫描。
- 4.应尽量避免在 `where` 子句中使用 `or` 来连接条件，如果一个字段有索引，一个字段没有索引，将导致引擎放弃使用索引而进行全表扫描
- 5.任何地方都不要使用 `select * from t`，用具体的字段列表代替“*”，不要返回用不到的任何字段。

12. n 条直线最多把平面分成几份？

由题意，平面内 n 条直线，任何两条不平行，任何三条不过同一点时，将平面分成的区域最多

设前 k 条直线把平面分成了 $f(k)$ 部分，第 $k+1$ 条直线与原有的 k 条直线有 k 个交点，这 k 个交点将第 $k+1$ 条直线分为 $k+1$ 段，这 $k+1$ 段将平面上原来的 $f(k)$ 部分的每一部分分成了 2 个部分，共 $2(k+1)$ 部分，相当于增加了 $k+1$ 个部分，

∴第 $k+1$ 条直线将平面分成了 $f(k+1)$ 部分，则 $f(k+1) - f(k) = k+1$ ，

令 $k=1, 2, 3, \dots, n$ 得 $f(2) - f(1) = 2, f(3) - f(2) = 3, \dots, f(n) - f(n-1) = n$ ，

把这 $n-1$ 个等式累加，得 $f(n) - f(1) = 2+3+\dots+n = \frac{(n+2)(n-1)}{2}$

∴ $f(n) = 2 + \frac{(n+2)(n-1)}{2} = \frac{n^2+n+2}{2}$

故选 C。



13. 谷歌

谷歌创始人之一拉里·佩奇指出：“完美的搜索引擎需要做到确解用户之意，且返用户之需”。就搜索技术的现状而言，我们需要通过研究、开发和革新来实现长远的发展。谷歌致力于成为这一技术领域的开拓者。尽管谷歌已是全球公认的业界领先的搜索技术公司，但其目标是为所有信息搜寻者提供更高标准的服务。

在谷歌，要求工程师们每周都花一天时间在个人感兴趣的项目上。这种近乎强制性的要求造成 Google News 之类的新服务品种出现，根据 Nielsen NetRatings 的数据显示，这项服务每个月都能吸引 710 万浏览者，同时也导致了社区网络站点 Orkut 的出现，已经被整合到整个搜索网站之中。谷歌里的每一个人都充满了故事：与你共进午餐的人或许发明了你在使用的编程语言；坐在你隔壁的同事或许为你的研究生课程编写过教材；和你一起打台球的那个人或许开发过你的桌面浏览器。

公司还提供免费的班车和渡轮服务接载雇员上班，这些交通工具都有无线互联网服务，方便员工在上下班时也可以工作。

14. 解释

FTP 是 File Transfer Protocol（文件传输协议）

超文本传输协议（HTTP, HyperText Transfer Protocol）

ACM (Association for Computing Machinery) 中文：国际计算机学会

电气和电子工程师协会(IEEE, 全称是 Institute of Electrical and Electronics Engineers)

15 集合，关系，图，树

集合：集合是一组无序的对象（集合中元素互异）

关系：集合中的元素之间的关系被表示为一种结构，这种结构就是关系，一般用有序对来表示元素之间的关系。

图：图是由顶点和连接顶点的边构成的离散结构。按照边是否有方向可以分为有向图和无向图

树：不包含简单回路的连通图称为树

15 子曰：君子怀德，小人怀土；君子怀刑，小人怀惠。

君子怀德，君子所思的是道德，他的居处，必然选择在有仁德之人所居的邻里。小人怀土，土是地利，小人只选择有利可图之地，如升官发财等，以为居处，定居后，则安安而不迁徙。此为上二句，就择居而辨君子小人。

君子怀刑，刑即典刑，经典法则。例如书经大禹谟：「惠迪吉，从逆凶。」迪，孔传为道。顺道而行则吉。逆，是相反。反其道而行则凶。君子有所行动，就想到是否合乎此类典刑。小人怀惠，中庸：「小人行险以侥幸。」小人之行，冒险以求其幸，不思虑后果，只贪图眼前的小惠。此为下二句，就行动而辨君子小人。

16. 连续函数

连续函数：函数连续的一个定义：

$$\lim_{\Delta x \rightarrow 0} \Delta y = 0$$

这就是说，如果自变量在某一点处的增量趋于 0 时，对应函数值的增量也趋于 0，就把 $f(x)$ 称作是在该点处连续的。

一致连续：若定义在实数区间 A （注意区间 A 可以是闭区间，亦可以是开区间甚至是无穷区间）上的任意函数 $f(x)$ ，对于任意给定的正数 $\epsilon > 0$ ，总存在一个与 x 无关的实数 $\zeta > 0$ ，使得当区间 A 上的任意两点 x_1, x_2 ，满足 $|x_1 - x_2| < \zeta$ 时，总有 $|f(x_1) - f(x_2)| < \epsilon$ ，则称 $f(x)$ 在区间 A 上是一致连续的。

17 平台无关性

就是在不同的系统上面都可以运行了

windows

linux

unix

都可以

楼上说的对

java 程序都是跑在虚拟机上面的

sun 给不同的系统都作了虚拟机

所以 java 跨平台

Java 主要靠 Java 虚拟机（JVM）在目标码级实现平台无关性。JVM 是一种抽象机器，它附着在具体操作系统之上，本身具有一套虚机器指令，并有自己的栈、寄存器组等。但 JVM 通常是在软件上而不是在硬件上实现。（目前，SUN 系统公司已经设计实现了 Java 芯片，主要使用在网络计算机 NC 上。另外，Java 芯片的出现也会使 Java 更容易嵌入到家用电器中。）JVM 是 Java 平台无关的基础，在 JVM 上，有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员在编写完软件后，通过 Java 编译器将 Java 源程序编译为 JVM 的字节代码。任何一台机器只要配备了 Java 解释器，就可以运行这个程序，而不管这种字节码是在何种平台上生成的（过程如图 1 所示）。另外，Java 采用的是基于 IEEE 标准的数据类型。通过 JVM 保证数据类型的一致性，也确保了 Java 的平台无关性。

18 计算机中如何表示数据

使用二进制来表示数据，有原码，补码，反码，移码

19 cache 的两种更新机制

回写法和直写法

- 1) write through: CPU 向 cache 写入数据时，同时向 memory 也写一份，使 cache 和 memory 的数据保持一致。优点是简单，缺点是每次都要访问 memory，速度比较慢。
- 2) write back: CPU 更新 cache 时，只是把更新的 cache 区标记一下，并不同步更新 memory。只是在 cache 区要被新进入的数据取代时，才更新 memory。这样做的原因是考虑到很多时候 cache 存入的是中间结果，没有必要同步更新 memory。优点是 CPU 执行的效率提高，缺点是实现起来技术比较复杂。

20 windows 漏洞

漏洞描述：服务拒绝。

解释 Windows XP 支持点对点的协议（PPTP）作为远程访问服务实现的虚拟专用网技术。由于在其控制用于建立、维护和拆开 PPTP 连接的代码段中存在未经检查的缓存，导致 Windows XP 的实现中存在漏洞。通过向一台存在该漏洞的服务器发送不正确的 PPTP 控制数据，攻击者可损坏核心内存并导致系统失效，中断所有系统中正在运行的进程。

该漏洞可攻击任何一台提供 PPTP 服务的服务器，对于 PPTP 客户端的工作站，攻击者只需激活 PPTP 会话即可进行攻击。对任何遭到攻击的系统，可通过重启来恢复正常操作。

21 进程和线程的区别

简而言之，一个程序至少有一个进程，一个进程至少有一个线程。

线程的划分尺度小于进程，使得多线程程序的并发性高。

另外，进程在执行过程中拥有独立的内存单元，而多个线程共享内存，从而极大地提高了程序的运行效率。

线程在执行过程中与进程还是有区别的。每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口。但是线程不能够独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制。

从逻辑角度来看，多线程的意义在于一个应用程序中，有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用，来实现进程的调度和管理以及资源分配。这就是进程和线程的重要区别。

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。

一个线程可以创建和撤销另一个线程；同一个进程中的多个线程之间可以并发执行。

22 路由协议

路由协议通过在路由器之间共享路由信息来支持可路由协议。路由信息在相邻路由器之间传递，确保所有路由器知道到其它路由器的路径。总之，路由协议创建了路由表，描述了网络拓扑结构；路由协议与路由器协同工作，执行路由选择和数据包转发功能。

边界网关协议（BGP）是运行于 TCP 上的一种自治系统的路由协议。BGP 是唯一一个用来处理像因特网大小的网络的协议，也是唯一能够妥善处理好不相关路由域间的多路连接的协议。

OSPF(Open Shortest Path First 开放式最短路径优先)是一个内部网关协议(Interior Gateway Protocol, 简称 IGP), 用于在单一自治系统 (autonomous system, AS) 内决策路由。是对链路状态路由协议的一种实现, 隶属内部网关协议 (IGP), 故运作于自治系统内部。著名的迪克斯加算法(Dijkstra)被用来计算最短路径树。

RIP 是一种基于距离矢量的路由协议, 以路由跳数作为计数单位的路由协议。适合用于比较小型的网络环境。

23 带冲突检测的载波监听

CSMA/CD 是英文 carrier sense multiple access/collision detected

在 CSMA/CD 协议中, 一旦检测到冲突, 为降低再冲突的概率, 需要等待一个随机时间, 然后再使用 CSMA 方法试图传输。为了保证这种退避维持稳定, 采用了二进制指数退避算法的技术, 其算法过程如下:

1. 确定基本退避时间, 一般为端到端的往返时间为 $2t$, $2t$ 也成为冲突窗口或征用期。
2. 定义参数 k , k 与冲突次数有关, 规定 k 不能超过 10, $k = \text{Min}[\text{冲突次数}, 10]$ 。在冲突次数大于 10, 小于 16 时, k 不再增大, 一直取值为 10。
3. 从离散的整数集合 $[0, 1, 2, \dots, (2k-1)]$ 中随机的取出一个数 r , 等待的时延为 r 倍的基本退避时间, 等于 $r \times 2t$ 。 r 的取值范围与冲突次数 k 有关, r 可选的随机取值为 $2k$ 个、这也是称为二进制退避算法的起因。
4. 当冲突次数大于 10 以后, 都是从 $0 \sim 2^{10}-1$ 个 $2t$ 中随机选择一个作为等待时间。
5. 当冲突次数超过 16 次后, 发送失败, 丢弃传输的帧, 发送错误报告。

24 MFC

MFC(Microsoft Foundation Classes)是微软基础类库的简称, 是微软公司实现的一个 c++类库, 主要封装了大部分的 windows API 函数, vc++是微软公司开发的 c/c++的集成开发环境, 所谓集成开发环境, 就是说利用它可以编辑, 编译, 调试, 而不是使用多种工具轮换操作, 灵活性较大。vc 也指它的内部编译器, 集成开发环境必须有一个编译器内核, 例如 DevC++其中一个编译器内核就是 gcc。MFC 除了是一个类库以外, 还是一个框架, 在 vc++里新建一个 MFC 的工程, 开发环境会自动帮你产生许多文件, 同时它使用了 mfcxx.dll。xx 是版本, 它封装了 mfc 内核, 所以你在你的代码看不到原本的 SDK 编程中的消息循环等等东西, 因为 MFC 框架帮你封装好了, 这样你就可以专心的考虑你程序的逻辑, 而不是这些每次编程都要重复的东西, 但是由于是通用框架, 没有最好的针对性, 当然也就丧失了一些灵活性和效率。但是 MFC 的封装很浅, 所以效率上损失不大。

25 八人过河

一家六口, 包括爸爸、妈妈、两个女儿及两个儿子在远足途中迷路, 不幸遇上一个逃狱的犯人, 正好被一个远足的休班警员逮捕。一家六口才得保住性命。但荒郊野外无线电接收不到, 所以他们不能与外界联系, 只有一条河是必经之路, 河上有一艘木筏, 木筏能载两人, 只有警员、爸爸、妈妈能控制木筏。可警员离去, 逃犯伤害一家六口; 当妈妈看见爸爸离去, 就会教训儿子; 当爸爸看见妈妈离去, 就会教训女儿。请问怎样才能让这八个人过河?

- 1, 让警察和小偷过去
- 2, 在对岸, 留下小偷, 警察回来 (对岸是小偷)
- 3, 警察带 1 男孩过去, 在对岸, 留下男孩, 警察带小偷回来 (对岸是 1 男孩)
- 4, 让男人带另 1 个男孩过去, 男人回来, (对岸是 2 男孩)
- 5, 男人带女人过去, 女人回来 (对岸是 1 男人 2 男孩)

- 6, 猎人带小偷过去, 男人回来 (对岸是 1 猎人 1 狗 2 男孩)
- 7, 男人带女人过去, 女人回来 (对岸是 1 小偷 1 警察 1 男人 2 男孩)
- 8, 女人带 1 女孩过去, 留下女孩, 女人和男人回来 (对岸是 1 警察 1 小偷 2 男孩 1 女孩)
- 9, 女人带另 1 个女孩过去, 女人回来 (对岸是 1 警察 1 小偷 2 男孩 2 女孩)
- 10, 女人和男人一起回来 (全部过岸)

26

1. 艾伦·麦席森·图灵 (Alan Mathison Turing, 1912 年 6 月 23 日—1954 年 6 月 7 日), 英国数学家、逻辑学家, 被称为计算机科学之父, 人工智能之父。

由美国计算机协会 (ACM) 于 1966 年设立一年一度的图灵奖, 以表彰在计算机科学中做出突出贡献的人, 图灵奖被喻为“计算机界的诺贝尔奖”。

理查德·卡普 NP-完全性理论

彼得·诺尔 算法语言 60 特点是局部性、动态性、[递归](#)性和严谨性

姚期智 计算理论, 包括伪随机数生成, 密码学与通信复杂度 2017 年 2 月, 姚期智教授放弃外国国籍成为中国公民, 正式转为中国科学院院士

(2)

停机问题 (halting problem)。其本质问题是: 给定一个图灵机 T , 和一个任意语言集合 S , 是否 T 会最终停机于每一个。通俗的说, 停机问题就是判断任意一个程序是否会在有限的时间之内结束运行的问题。

(3) 图灵的基本思想是用机器来模拟人们用纸笔进行数学运算的过程, 他把这样的过程看作下列两种简单的动作: 在纸上写上或擦除某个符号; 把注意力从纸的一个位置移动到另一个位置;

一台图灵机是一个七元组, $\{Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}\}$, 其中 Q, Σ, Γ 都是有限集合, 且满足

1. Q 是状态集合;
2. Σ 是输入字母表, 其中不包含特殊的空白符 \square ;
3. Γ 是带字母表, 其中 $\square \in \Gamma$ 且 $\Sigma \in \Gamma$;
4. $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ 是转移函数, 其中 L, R 表示读写头是向左移还是向右移;
5. $q_0 \in Q$ 是起始状态;
6. q_{accept} 是接受状态。
7. q_{reject} 是拒绝状态, 且 $q_{\text{reject}} \neq q_{\text{accept}}$

图灵机 $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$

27 什么是数学证明

在数学上, 证明是在一个特定的公理系统中, 根据一定的规则或标准, 由公理和定理推导出某些命题的过程。反证法, 数学归纳法 (证 1 到 n 的和), 演绎法 (演绎推理有三段论、假言推理、选言推理、关系推理等形式。)

28 罗素悖论

罗素悖论: 设集合 S 是由一切不属于自身的集合所组成, 即 “ $S = \{x | x \notin x\}$ ”。那么问题是: S 包含于 S 是否成立? 首先, 若 S 包含于 S , 则不符合 $x \notin S$, 则 S 不包含于 S ; 其次, 若 S 不包含于 S , 则符合 $x \notin S$, S 包含于 S

解决这一悖论主要有两种选择，ZF 公理系统和 NBG 公理系统。

1908 年，策梅罗（Ernst Zermelo）在自己这一原则基础上提出第一个公理化集合论体系，后来这一公理化集合系统很大程度上弥补了康托尔朴素集合论的缺陷。这一公理系统在通过弗兰克尔（Abraham Fraenkel）的改进后被称为 ZF 公理系统。在该公理系统中，由于分类公理（Axiom schema of specification）： $P(x)$ 是 x 的一个性质，对任意已知集合 A ，存在一个集合 B 使得对所有元素 $x \in B$ 当且仅当 $x \in A$ 且 $P(x)$ ；因此 $\{x \mid x \text{ 是一个集合}\}$ 并不能在该系统中写成一个集合，由于它并不是任何已知集合的子集；并且通过该公理，存在集合 $A = \{x \mid x \text{ 是一个集合}\}$ 在 ZF 系统中能被证明是矛盾的，因此罗素悖论在该系统中被避免了。

29 王国维的人生三种境界

古今之成大事业、大学问者，必经过三种之境界："昨夜西风凋碧树。独上高楼，望尽天涯路。"此第一境也。"衣带渐宽终不悔，为伊消得人憔悴。"此第二境也。"众里寻他千百度，蓦然回首，那人却在灯火阑珊处。"此第三境也。此等语皆非大词人不能道。然遽以此意解释诸词，恐为晏欧诸公所不许也。

第一境界在王国维此句中解成，做学问成大事业者，首先要有执着的追求，登高望远，瞰察路径，明确目标与方向，了解事物的概貌。

第二境界王国维则别有用心，以此两句来比喻成大事业、大学问者，不是轻而易举，随便可得的，必须坚定不移，经过一番辛勤劳动，废寝忘食，孜孜以求，直至人瘦带宽也不后悔。

第三境界做学问、成大事业者，要达到第三境界，必须有专注的精神，反复追寻、研究，下足功夫，自然会豁然贯通，有所发现，有所发明，就能够从必然王国进入自由王国。

30 图灵测试

图灵提出了一种测试机器是不是具备人类智能的方法。即假设有一台电脑，其运算速度非常快、记忆容量和逻辑单元的数目也超过了人脑，而且还为这台电脑编写了许多智能化的程序，并提供了合适种类的大量数据，那么，是否就能说这台机器具有思维能力？

图灵肯定机器可以思维的，



图 1 图灵测试示意图

图灵测试

他还对智能问题从行为主义的角度给出了定义，由此提出一假想：即一个人在不接触对方的情况下，通过一种特殊的方式，和对方进行一系列的问答，如果在相当长时间内，他无法根据这些问题判断对方是人还是计算机，那么，就可以认为这个计算机具有同人相当的智力，即这台计算机是能思维的。这就是著名的“图灵测试”（Turing Testing）。当时全世界

只有几台电脑，其他几乎所有计算机根本无法通过这一测试。^[7]

要分辨一个想法是“自创”的思想还是精心设计的“模仿”是非常难的，任何自创思想的证据都可以被否决。图灵试图解决长久以来关于如何定义思考的哲学争论，他提出一个虽然主观但可操作的标准：如果一台电脑表现（act）、反应（react）和互相作用（interact）都和有意意识的个体一样，那么它就应该被认为是有意意识的。

为消除人类心中的偏见，图灵设计了一种“[模仿游戏](#)”即图灵测试：远处的人类测试者在一段规定的时间内，根据两个实体对他提出的各种问题的反应来判断是人类还是电脑。通过一系列这样的测试，从电脑被误判断为人的几率就可以测出电脑智能的成功程度。

图灵预言，在 20 世纪末，一定会有电脑通过“图灵测试”。2014 年 6 月 7 日在[英国皇家学会](#)举行的“2014 图灵测试”大会上，举办方英国雷丁大学发布新闻稿，宣称俄罗斯人弗拉基米尔·维西罗夫（Vladimir Veselov）创立的人工智能软件[尤金·古斯特曼](#)（Eugene Goostman）通过了图灵测试。虽然“尤金”软件还远不能“思考”，但也是人工智能乃至计算机史上的一个标志性事件。

人机测试

图灵采用“问”与“答”模式，即观察者通过控制[打字机](#)向两个测试对象通话，其中一个是人，另一个是机器。要求观察者不断提出各种问题，从而辨别回答者是人还是机器。图灵还为这项测试亲自拟定了几个示范性问题：

问：请给我写出有关“第四号桥”主题的十四行诗。

答：不要问我这道题，我从来不会写诗。

问：34957 加 70764 等于多少？

答：（停 30 秒后）105721

问：你会下[国际象棋](#)吗？

答：是的。

问：我在我的 K1 处有棋子 K；你仅在 K6 处有棋子 K，在 R1 处有棋子 R。轮到你走，你应该下哪步棋？

答：（停 15 秒钟后）棋子 R 走到 R8 处，将军！

图灵指出：“如果机器在某些现实的条件下，能够非常好地模仿人回答问题，以至提问者在相当长时间里误认它不是机器，那么机器就可以被认为是能够思维的。”

从表面上看，要使机器回答按一定范围提出的问题似乎没有什么困难，可以通过编制特殊的程序来实现。然而，如果提问者并不遵循常规标准，编制回答的程序是极其困难的事情。例如，提问与回答呈现出下列状况：

问：你会[下国际象棋](#)吗？

答：是的。

问：你会下国际象棋吗？

答：是的。

问：请再次回答，你会下国际象棋吗？

答：是的。

你多半会想到，面前的这位是一部笨机器。如果提问与回答呈现出另一种状态：

问：你会下国际象棋吗？

答：是的。

问：你会下国际象棋吗？

答：是的，我不是已经说过了吗？

问：请再次回答，你会下国际象棋吗？

答：你烦不烦，干嘛老提同样的问题。

那么，你面前的这位，大概是人而不是机器。上述两种对话的区别在于，第一种可明显地感到回答者是从知识库中提取简单的答案，第二种则具有分析综合的能力，回答者知道观察者在反复提出同样的问题。“图灵测试”没有规定问题的范围和提问的标准，如果想要制造出能通过试验的机器，以我们的技术水平，必须在电脑中储存人类所有可以想到的问题，[储存](#)对这些问题的所有合乎常理的回答，并且还需要理智地作出选择

32 编译器与解释器的区别

解释器 - 直接利用用户提供的输入，执行源程序 中指定的操作 - 不生成目标程序，而是根据源程序的语义直接运行

33 什么是瀑布模型

瀑布模型核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作，即采用结构化的分析与设计方法将逻辑实现与物理实现分开。将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动，并且规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。

34 智力测验：有两柱不同的且质地都不均匀香，但烧完都要用 1

小时。问如何用这两柱香来表示 15 分钟？

把一支香点燃一端 同时把另一支香点燃两端 当第二支香先烧完时也就是过了半小时 那么再把第一支香没点燃的那端点燃 当第一支香烧完时正好 15 分钟

35 class 和 struct 的区别：

- 1、class 是引用类型，struct 是值类型；
- 2、class 可以继承类、接口和被继承，struct 只能继承接口，不能被继承；
- 3、class 有默认的无参构造函数，有析构函数，struct 没有默认的无参构造函数，且只能声明有参的构造函数，没有析构函数；
- 4、class 可以使用 abstract 和 sealed，有 protected 修饰符，struct 不可以用 abstract 和 sealed，没有 protected 修饰符；
- 5、class 必须使用 new 初始化，结构可以不用 new 初始化；
- 6、class 实例由垃圾回收机制来保证内存的回收处理，而 struct 变量使用完后立即自动解除内存分配；
- 7、从职能观点来看，class 表现为行为，而 struct 常用于存储数据；
- 8、作为参数传递时，class 变量以按址方式传递，而 struct 变量是以按值方式传递的。

36 C++ 构造函数和析构函数可以重载吗

构造函数可以被重载，因为构造函数可以有多个且可以带参数。

析构函数不可以被重载，因为析构函数只能有一个，且不能带参数。

37 c++中多态

多态指同一个实体同时具有多种形式。它是面向对象程序设计（OOP）的一个重要特征。如果一个语言只支持类而不支持多态，只能说明它是基于对象的，而不是面向对象的。C++中，实现多态有以下方法：虚函数，抽象类，覆盖，模板（重载和多态无关）。

38 数据库的恢复方法

日志和备份

39 数据库接口

数据库接口是与数据库建立连接的技术。这个接口提供了最大限度的相互可操作性：一个应用程序可以通过一组通用的代码访问不同的数据库管理系统。一个软件开发者开发的客户/服务器应用程序不会被束定于某个特定的数据库之上。曾今在学习 web 的时候，对数据库实现增删改查的时候使用过数据库的接口。在 myeclipse 中实现数据库的连接。

40 研究人工智能最重要的是什么

算法具有较强的学习能力

41 专家系统

专家系统是一个智能计算机程序系统，其内部含有大量的某个领域专家水平的知识与经验，能够利用人类专家的知识和解决问题的方法来处理该领域问题。也就是说，专家系统是一个具有大量的专门知识与经验的程序系统，它应用人工智能技术和计算机技术，根据某领域一个或多个专家提供的知识和经验，进行推理和判断，模拟人类专家的决策过程，以便解决那些需要人类专家处理的复杂问题，简而言之，专家系统是一种模拟人类专家解决领域问题的计算机程序系统。

知识库用来存放专家提供的知识。专家系统的问题求解过程是通过知识库中的知识来模拟专家的思维方式的，因此，知识库是专家系统质量是否优越的关键所在，即知识库中知识的质量和数量决定着专家系统的质量水平。

42 Windows 系统为什么容易受到攻击

1.在微软的 Windows 系统中，自动更新程序只会升级那些由微软公司官方所提供的组件。而第三方的应用程序却不会得到修补。从而，第三方的应用程序可能会给你的系统带来大量的安全隐患。

2.window 系统用户人多，攻击的人也多

3.windows 代码的不公开，很多安全问题都被掩盖起来

43

（1）FTP 是 File Transfer Protocol（文件传输协议）的英文简称，而中文简称为“文传协议”。用于 Internet 上的控制文件的双向传输。

（2）超文本传输协议（HTTP, HyperText Transfer Protocol)是互联网上应用最为广泛的一种网络协议。所有的 WWW 文件都必须遵守这个标准。

FTP 和 HTTP 一样都是 Internet 上广泛使用的协议,用来在两台计算机之间互相传送文件。相比于 HTTP,FTP 协议要复杂得多。复杂的原因,是因为 FTP 协议要用到两个 TCP 连接,一个是命令链路,用来在 FTP 客户端与服务器之间传递命令;另一个是数据链路,用来上传或下载数据。

44 OSI 七层模型:

OSI 采用了分层的结构化技术,共分七层,物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

45 进程调度的 3 态图, 5 态图, 7 态图

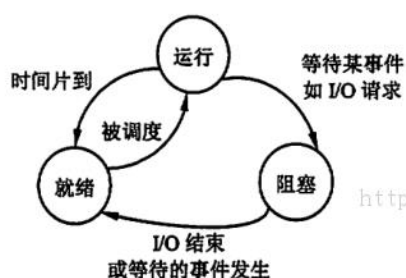


图 2-3 进程三态模型

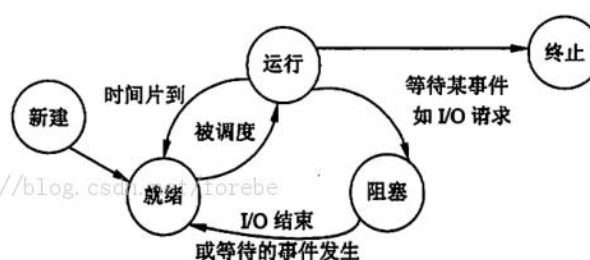
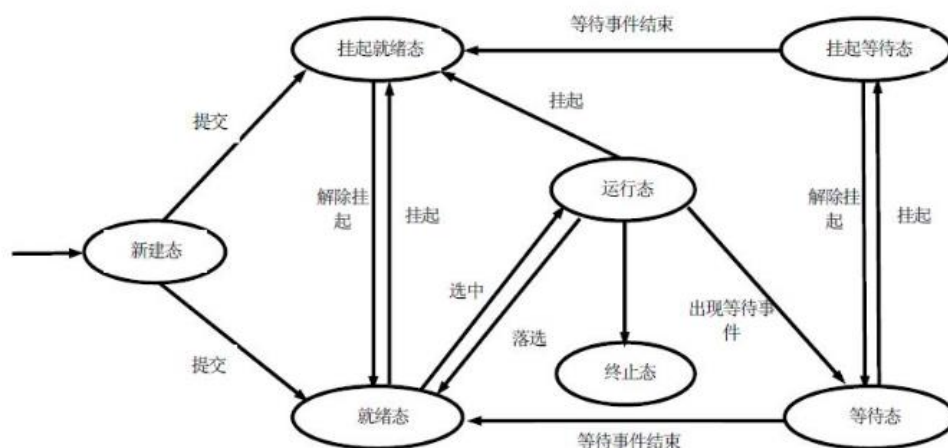


图 2-4 进程五态模型



46 系统调用和库函数的区别

系统调用：是操作系统为用户态运行的进程和硬件设备(如 CPU、磁盘、打印机等)进行交互提供的一组接口，即就是设置在应用程序和硬件设备之间的一个接口层。可以说是操作系统留给用户程序的一个接口。再来说一下，linux 内核是单内核，结构紧凑，执行速度快，各个模块之间是直接调用的关系。放眼望整个 linux 系统，从上到下依次是用户进程->linux 内

核->硬件。其中系统调用接口是位于 Linux 内核中的，如果再稍微细分一下的话，整个 linux 系统从上到下可以是：用户进程->系统调用接口->linux 内核子系统->硬件，也就是说 Linux 内核包括了系统调用接口和内核子系统两部分；或者从下到上可以是：物理硬件->OS 内核->OS 服务->应用程序，其中操作系统起到“承上启下”的关键作用，向下管理物理硬件，向上为操作系统服务和应用程序提供接口，这里的接口就是系统调用了。

一般地，操作系统为了考虑实现的难度和管理的方便，它只提供一部分的系统调用，这些系统调用一般都是由 C 和汇编混合编写实现的，其接口用 C 来定义，而具体的实现则是汇编，这样的好处就是执行效率高，而且，极大的方便了上层调用。

库函数：顾名思义是把函数放到库里。是把一些常用到的函数编完放到一个文件里，供别人用。别人用的时候把它所在的文件名用#include<>加到里面就可以了。一般是放到 lib 文件里的。一般是指编译器提供的可在 c 源程序中调用的函数。可分为两类，一类是 c 语言标准规定的库函数，一类是编译器特定的库函数。(由于版权原因，库函数的源代码一般是不可见的，但在头文件中你可以看到它对外的接口)

libc 中就是一个 C 标准库，里面存放一些基本函数，这些基本函数都是被标准化了的，而且这些函数通常都是用汇编直接实现的。

库函数一般可以概括的分为两类，一类是随着操作系统提供的，另一类是由第三方提供的。随着系统提供的这些库函数把系统调用进行封装或者组合，可以实现更多的功能，这样的库函数能够实现一些对内核来说比较复杂的操作。比如，read()函数根据参数，直接就能读文件，而背后隐藏的比如文件在硬盘的哪个磁道，哪个扇区，加载到内存的哪个位置等等这些操作，程序员是不必关心的，这些操作里面自然也包含了系统调用。而对于第三方的库，它其实和系统库一样，只是它直接利用系统调用的可能性要小一些，而是利用系统提供的 API 接口来实现功能(API 的接口是开放的)。部分 Libc 库中的函数的功能的实现还是借助了系统调用，比如 printf 的实现最终还是调用了 write 这样的系统调用；而另一些则不会使用系统调用，比如 strlen, strcat, memcpy 等。

实时上，系统调用所提供给用户的是直接而纯粹的高级服务，如果想要更人性化，更符合特定情况的功能，那么就要我们用户自己来定义，因此就衍生了库函数，它把部分系统调用包装起来，一方面把系统调用抽象了，一方面方便了用户级的调用。系统调用和库函数在运行的效果上很相似（当然库函数会更符合需求），但是系统调用是运行于内核状态；而库函数由用户调用，运行于用户态。

系统调用是为了方便使用操作系统的接口，而库函数则是为了人们编程的方便。

47 目态和管态

大多数计算机系统将 CPU 执行状态分为管态和目态。管态又称为特权状态、系统态或核心态。通常，操作系统在管态下运行。目态又叫做常态或用户态，用户程序只能在目态下运行，如果用户程序在目态下执行特权指令，硬件将发生中断，由操作系统获得控制，特权指令执行被禁止，这样可以防止用户程序有意或无意的破坏系统。从目态转换为管态的唯一途径是中断。

48 面包店算法

该算法的基本思想源于顾客在面包店中购买面包时的排队原理。顾客在进入面包店前，首先

抓一个号, 然后按照号码由小到大的次序依次进入面包店购买面包. 这里, 面包店发放的号码是由小到大的, 但是两个或两个以上的顾客却有可能得到相同的号码(使所抓号码不同需要互斥), 如果多个顾客抓到相同的号码, 则规定按照顾客名字的字典次序进行排序, 这里假定顾客是没有重名的. 在计算机系统中, 顾客就相当于进程, 每个进程有一个唯一的标识, 我们用 P 的下面加一个下标来表示. 例如: 对于 P_i 和 P_j , 如果有 $i < j$, 则先为 P_i 服务, 即 P_i 先进入临界区

49 页表的实现

页表是一种特殊的数据结构, 放在系统空间的页表区, 存放逻辑页与物理页帧的对应关系. 每一个进程都拥有一个自己的页表, PCB 表中有指针指向页表
页表里面存放的是物理页号, 有效位, 脏位等

50. 什么是操作系统

操作系统 (Operating System, 简称 OS) 是管理和控制计算机硬件与软件资源的计算机程序, 是直接运行在“裸机”上的最基本的系统软件, 任何其他软件都必须在操作系统的支持下才能运行. 操作系统是用户和计算机的接口, 同时也是计算机硬件和其他软件的接口. 操作系统的功能包括管理计算机系统的硬件、软件及数据资源, 控制程序运行, 改善人机界面, 为其它应用软件提供支持, 让计算机系统所有资源最大限度地发挥作用, 提供各种形式的用户界面, 使用户有一个好的工作环境, 为其它软件的开发提供必要的服务和相应的接口等. 实际上, 用户是不用接触操作系统的, 操作系统管理着计算机硬件资源, 同时按照应用程序的资源请求, 分配资源, 如: 划分 CPU 时间, 内存空间的开辟, 调用打印机等。

51 排序有哪几种

选择排序, 插入排序, 冒泡排序, 快速排序, 堆排序, 归并排序, 基数排序。
快速排序平均时间效率最高。

52 PV 操作

PV 操作与信号量的处理相关, P 表示获取的意思, V 表示释放的意思. PV 信号量机制可以用来解决进程的同步与互斥的问题. 一般来说, 信号量 $S \geq 0$ 时, S 表示可用资源的数量. 执行一次 P 操作意味着请求分配一个单位资源, 因此 S 的值减 1; 当 $S < 0$ 时, 表示已经没有可用资源, 请求者必须等待别的进程释放该类资源, 它才能运行下去. 而执行一个 V 操作意味着释放一个单位资源, 因此 S 的值加 1; 若 $S < 0$, 表示有某些进程正在等待该资源, 因此要唤醒一个等待状态的进程, 使之运行下去。

53 死锁的概念

所谓死锁: 是指两个或两个以上的进程在执行过程中, 由于竞争资源或者由于彼此通信而造成的一种阻塞的现象, 若无外力作用, 它们都将无法推进下去. 此时称系统处于死锁状态或系统产生了死锁, 这些永远在互相等待的进程称为死锁进程。

54 死锁产生的四个必要条件

- 1) 互斥条件: 指进程对所分配到的资源进行排它性使用, 即在一段时间内某资源只由一个进程占用. 如果此时还有其它进程请求资源, 则请求者只能等待, 直至占有资源的进程用毕释放。
- 2) 请求和保持条件: 指进程已经保持至少一个资源, 但又提出了新的资源请求, 而该资源

已被其它进程占有，此时请求进程阻塞，但又对自己已获得的其它资源保持不放。

3) 不剥夺条件：指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。

4) 循环等待条件：指在发生死锁时，必然存在一个进程——资源的环形链，即进程集合{P0, P1, P2, ..., Pn}中的 P0 正在等待一个 P1 占用的资源；P1 正在等待 P2 占用的资源，……，Pn 正在等待已被 P0 占用的资源。

55 死锁产生的原因

1. 竞争不可剥夺的资源引起的死锁
2. 进程推进顺序不当

56 死锁的处理方法

1) 预防死锁。

这是一种较简单和直观的事先预防的方法。方法是通过设置某些限制条件，去破坏产生死锁的四个必要条件中的一个或者几个，来预防发生死锁。预防死锁是一种较易实现的方法，已被广泛使用。但是由于所施加的限制条件往往太严格，可能会导致系统资源利用率和系统吞吐量降低。

2) 避免死锁。

该方法同样是属于事先预防的策略，但它并不须事先采取各种限制措施去破坏产生死锁的四个必要条件，而是在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免发生死锁。

3) 检测和解除死锁。

先检测：这种方法并不须事先采取任何限制性措施，也不必检查系统是否已经进入不安全区，此方法允许系统在运行过程中发生死锁。但可通过系统所设置的检测机构，及时地检测出死锁的发生，并精确地确定与死锁有关的进程和资源。检测方法包括定时检测、效率低时检测、进程等待时检测等。

56. 计算机系统中如何表示小数点

计算机系统中有定点数和浮点数，定点数通过固定小数点的位置来表示小数点，定点整数的小数点一般默认为最后一位处，定点小数的小数点一般在符号位的后面。浮点数通过阶码，尾数，符号位来表示，阶码和尾数决定小数点的位置。

57. 什么是虚函数

简单地说，那些被 `virtual` 关键字修饰的成员函数，就是虚函数。虚函数的作用，用专业术语来解释就是实现多态性（Polymorphism），多态性是将接口与实现进行分离；用形象的语言来解释就是实现以共同的方法，但因个体差异而采用不同的策略。

58 模板

模板是 C++ 支持参数化多态的工具，使用模板可以使用户为类或者函数声明一种一般模式，使得类中的某些数据成员或者成员函数的参数、返回值取得任意类型。

模板是一种对类型进行参数化的工具；

通常有两种形式：函数模板和类模板；

函数模板针对仅参数类型不同的函数；

类模板针对仅数据成员和成员函数类型不同的类。

使用模板的目的就是能够让程序员编写与类型无关的代码。比如编写了一个交换两个整型 `int` 类型的 `swap` 函数，这个函数就只能实现 `int` 型，对 `double`，字符这些类型无法实现，要实现这些类型的交换就要重新编写另一个 `swap` 函数。使用模板的目的就是要让这程序的实现与类型无关，比如一个 `swap` 模板函数，即可以实现 `int` 型，又可以实现 `double` 型的交换。模板可以应用于函数和类。下面分别介绍。

59. 继承

通过继承机制，可以利用已有的数据类型来定义新的数据类型。所定义的新的数据类型不仅拥有新定义的成员，而且还同时拥有旧的成员。我们称已存在的用来派生新类的类为基类，又称为父类。由已存在的类派生出的新类称为派生类，又称为子类。

在 `C++` 语言中，一个 [派生类](#) 可以从一个 [基类](#) 派生，也可以从多个基类派生。从一个基类派生的继承称为单继承；从多个基类派生的继承称为 [多继承](#)。

派生类的定义格式

单继承的定义格式如下：

1	<code>class<派生类名>:<继承方式><基类名></code>
2	<code>{</code>
3	<code><派生类新定义成员></code>
4	<code>};</code>

其中，`class` 是关键词，`<派生类名>` 是新定义的一个类的名字，它是从 `<基类名>` 中派生的，并且按指定的 `<继承方式>` 派生的。`<继承方式>` 常使用如下三种关键字给予表示：

`public` 表示公有继承；

`private` 表示私有继承；

`protected` 表示保护继承；

[多继承](#) 的定义格式如下：

1	<code>class<派生类名>:<继承方式 1><基类名 1>,<继承方式 2><基类名 2>,...</code>
2	
3	<code>{</code>
4	
5	<code><派生类新定义成员></code>
6	
7	<code>};</code>

可见，多继承与单继承的区别从定义格式上看，主要是多继承的 [基类](#) 多于一个。

60 数据库关系模型

关系模型允许定义三类完整性约束：实体完整性、参照完整性和用户定义的完整性。其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件，体现了具体领域中的语义约束。

实体完整性规则：若属性 `A` 是基本关系 `R` 的主属性，则属性 `A` 不能取空值。实体完整性规则规定基本关系的所有主属性都不能取空值，而不仅是主码整体不能取空值。

(1) 实体完整性规则是针对基本关系而言的。一个基本表通常对应现实世界的一个实体集。例如学生关系对应于学生的集合。

(2) 现实世界中的实体是可区分的, 即它们具有某种唯一性标志。

(3) 相应地, 关系模型中以主码作为唯一性标志。

(4) 主码中的属性即主属性不能取空值。所谓空值就是“不知道”或“无意义”的值。如果主属性取空值, 就说明存在某个不可标识的实体, 即存在不可区分的实体。这与第(2)点相矛盾, 因此这个规则成为实体完整性。

参照完整性规则: 若属性(或属性组) F 是基本关系 R 的外码, 它对于基本关系 S 的主码 K 相对应(基本关系 R 和 S 不一定是不同的关系), 则对于 R 中的每个元组在 F 上的值必须为: 或者取空值 (F 的每个属性值均为空值); 或者等于 S 中某个元组的主码值。

用户定义的完整性就是针对某一具体关系数据库的约束条件。它反映某一具体应用所涉及的数据必须满足的语义要求。例如某个属性必须取唯一值、某些属性值之间应满足一定的函数关系、某个属性的取值范围在 $0 \sim 100$ 之间等。关系模型应提供定义和检验这类完整性的机制, 以便于用统一的系统的方法处理他们, 而不要由应用程序承担这一功能。

61 C 语言中指针的概念

指针即为地址。指针值是该对象所在存储空间的地址, 通过指针可以直接访问该对象。比如数组中, 数组名即为指针, 指针一般通过取地址符 $\&$ 来获取。使用指针具有很强的灵活性。

62 . 类与对象的关系

对象是对客观事物的抽象, 类是对对象的抽象。类是一种抽象的数据类型。

它们的关系是, 对象是类的实例, 类是对象的模板。对象是动态的, 类是静态的。

63. 什么是虚拟存储器

虚拟存储器是指具有请求调入功能和置换功能, 能从逻辑上对内存容量进行扩充的一种存储器系统。在虚拟存储器系统中, 作业无需全部装入, 只要装入一部分就可运行。

程序员可以在一个不受物理内存的限制, 并且比物理内存空间大的多的虚拟的逻辑地址空间中编写程序, 就好像每个程序都有一个巨大的存储空间一样。CUP 在运行时使用的是虚拟地址, 需要用存储器管理部件将虚拟地址转换成物理地址。才能访问主存。

64. 虚拟存储器, 内存, cache 之间的关系

虚拟存储器是为了解决内存不够的问题才出现的, 通过在内存上使用请求调入和置换的功能来从逻辑上扩充内存, 其解决的是内存空间大小不足的问题。Cache 是一种小容量的高速缓冲寄存器吧内存中频繁访问的程序和数据复制到其中, 主要是根据程序的访问局部性原理来实现 CPU 直接从 cache 中取得指令和数据, 不必访问慢速的主存。Cache 与主存之间的映射有直接映射, 全相连映射, 组相连映射。

65. 特征向量和特征值

定义 设 A 是 n 阶方阵, 如果数 λ 和 n 维非零列向量 x 使关系式

$$Ax = \lambda x$$

成立, 那么这样的数 λ 称为矩阵 A 特征值, 非零向量 x 称为 A 的对应于特征值 λ 的特征向量。

特征向量不是唯一的, 有无穷多个。

66 贝叶斯公式

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^n P(B_j)P(A|B_j)}$$

公式描述：

公式中，事件 B_i 的概率为 $P(B_i)$ ，事件 B_i 已发生条件下事件 A 的概率为 $P(A|B_i)$ ，事件 A 发生条件下事件 B_i 的概率为 $P(B_i|A)$ 。