1. 计算机系统知识

计算机系统由硬件系统和软件系统组成。硬件由运算器、控制器、存储器、输入设备、输出设备 5 部分组成；软件由系统软件、应用软件组成。

运算器：对数据进行处理的部件，主要完成算术和逻辑运算；
控制器：从主存中取出指令，并指出下一条指令在主存中的位置，取出的指令经指令寄存器送往指令译码器，经过对指令的分析发出相应的控制和定时信息；

控制器的组成部分为：
程序计数器
指令寄存器
指令译码器
状态条件寄存器
时序产生器
微信号发生器

计算机硬件的典型结构：单总线、双总线(以 cpu 为中心、以存储器为中心)、采用通道的大型系统。

2、二、八、十、十六进制间的转换方法
十进制转换成二进制：十进制整数转换成二进制整数通常采用除 2 取余法，小数部分乘 2 取整法。
例如，将 30D 转换成二进制数。
2| 30 ….0 ----最右位
2 15 ….1
2 7 ….1
2 3 ….1
1 ….1 ----最左位
∴ 30D=11110B
八、十六进制转二进制方法类似。

二进制数转换成八进制数：对于整数，从低位到高位将二进制数的每三位分为一组，若不够三位时，在高位左面添 0，补足三位，然后将每三 位二进制数用一位八进制数替换，小数部分从小数点开始，自左向右每三位一组进行转换即可完成。例如：将二进制数 1101001 转换成八进制数，则
001 101 001B
| | |
1 5 1O
1101001B = 151O

八进制数转换成二进制数：只要将每位八进制数用三位二进制数替换，即可完成转换，例如，把八进制数(643.503)8，转换成二进制数，则
(6 4 3 . 5 0 3)8
| | | | | |
(110 100 011 . 101 000 011)2
(643.503)8=(110100011.101000011)2
二进制与十六进制之间的转换
(1)二进制数转换成十六进制数：由于 2 的 4 次方=16，所以依照二进制与八进制的转换方法，将二进制数的每四位用一个十六进制数码来表示，整数部分以小数点为界点从右往左每四位一组转换，小数部分从小数点开始自左向右每四位一组进行转换。
(2)十六进制转换成二进制数
如将十六进制数转换成二进制数，只要将每一位十六进制数用四位相应的二进制数表示，即可完成转换。
例如：将(163.5B)16 转换成二进制数，则
(1 6 3 . 5 B)16
| | | | |
(0001 0110 0011. 0101 1011)2
(163.5B)16=(101100011.01011011)2

二进制的算术、逻辑运算

3、数据在计算机中的表示方法：各种数据在计算机中表示的形式称为机器数，其特点是用 0,1 表示，如 0 表示正号，1 表示负号，小数点隐含表示而不占位置。机器数对应的实际数据称为真值。机器数分为无符号数和有符号数。无符号数表示正数。

带符号的机器数可采用原码、反码、补码等码制进行计算。

4、汉字编码：汉字处理包括汉字的编码输入、存储、输出等环节。

输入码(数字编码、拼音码、字形编码)、内部码(简称汉字内码)(GB2312-80 用 2 字节表示一个汉字，Unicode 用 4 字节表

示一个汉字)、字形码(点阵、矢量函数，汉字的输出方式)

5、cpu 的功能：程序控制、操作控制、时间控制、数据处理

6、计算机系统分类：Flynn 分类法（按指令流、数据流分类）、冯式分类法（按最大并行度分类）
指令流：机器执行的指令序列；
数据流：指令调用的数据序列。

7、计算机系统结构和计算机组成的区别：系统结构是指计算机系统在总体上、功能上需要解决的问题；计算机组成是指在逻辑上如何具体实现的问题。

8、计算机并行的发展：不同于同时性的是，并发性是指两个或两个以上事件在同一时间间隔内连续发生；分为存储器操作并行，处理器操作步骤并行（流水线处理机），处理器操作并行（阵列处理机），指令、任务、作业并行（多处理机、分布式处理系统、计算机网络）。

9、存储器的层次结构：高速缓存、主存、辅存。（有人将 cpu 内部的寄存器也作为一个存储层次）

存储器的分类：存储器按位置分为内存（主存）和外存（辅存）；按工作方式分为读写存储器和只读存储器；按访问方式分为按地址访问和按内容访问的存储器；按寻址方式分为随机寻址、顺序、直接寻址存储器。

相连存储器是一种按内容访问的存储器。其工作原理是把数据作为关键字与存储器中的每一单元比较，找出与关键字相同的数据。相连存储器可用在高速缓存中；在虚拟存储器中用来作段表、页表或快表存储器；用在数据库和知识库中。

高速缓存：由控制部分和 cache 部分组成。cache 部分放主存的部分拷贝信息，控制部分判断 cpu 要访问的信息是否在 cache 中命中，并按替换算法决定主存的哪一块信息放到 cache 中的哪一块里面。
一般来说，Cache 的功能全部由硬件实现。
高速缓存与主存的地址映像方法有 3 种，即直接映像，全相连映像，组相连映像（组使用直接相连而组内的块使用全相连方式）
在 Cache 的替换算法中，“近期最少使用 LRU 算法”是命中率最高的一种算法。

10、虚拟存储器，是由主存、辅存、存储管理单元和操作系统的存储管理软件组成的存储系统。它将大容量的外存也纳入存储管理器的管理范围，具体执行程序时要先判断程序是否在主存中，若不在则需从辅存中调入。按工作方式分为：
页式虚拟存储器
段式虚拟存储器
段页式虚拟存储器

11、磁盘阵列 raid，是由多台磁盘存储器组成的，一个大而快速、可靠的外存子系统。
raid0　是不具备容错能力的阵列，N 个磁盘组成的 0 级阵列，其平均故障时间间隔是单个磁盘存储器的 N 分之一；但其数据传输速率是单　的 N 倍。
raid1　使用镜像容错技术
raid2　使用汉明码容错技术
raid3　一般使用一个检验盘
raid4　只使用一个检验盘
raid5　没有专门的检验盘，它在每块盘上都写数据和检验信息。

12、CISC--复杂指令集计算机　RISC--精简指令集计算机

RISC 的特点：　　指令种类少；
指令长度固定、格式少；
寻址方式少，适合于组合逻辑控制器；
设置最少的访问内存指令，访问内存比较花时间；
在 CPU 内部设置大量寄存器，使操作在 CPU 内部快速进行；
适合于流水线操作，容易并行执行。

13、输入输出技术
内存与接口的编址方式分为内存和接口地址独立的编址方式，和内存、接口地址统一的编址方式。
直接程序控制（无条件传送方式、程序查询方式）（整个输入输出过程是在 cpu 执行程序的控制下完成）
中断方式　　（cpu 得用中断方式完成数据的输入输出操作）
直接存储器存取（DMA）方式　，数据直接在内存与 IO 设备间成块传送，cpu 只需在开始和结束时进行处理，过程中无须干涉。

DMA 传送的一般过程为：
1）外设向 DMA 控制器提出 DMA 传送请求；
2）DMA 控制器向 CPU 提出请求；
3）CPU 允许 DMA 工作，处理总路线控制的转交；
4）
输入输出处理机（IOP）方式 ，由一个专用的处理机完成主机的输入输出操作。

14、流水线技术，是将一条指令分解成一连串执行的子过程，在 cpu 中将一条指令的串行执行过程变为若干条指令的子过程重叠执行。

特点是，流水线可分成若干相互联系的子过程；执行每个子过程的时间尽量相等；形成流水处理需要准备时间；指令流发生不能顺序执行时会使流水线中断。

两个指标，吞吐率（单位时间里流水线处理机流出的结果数，对指令而言就是单位时间里执行的指令数）；

建立时间（所有子过程执行一遍用时之和）

15、总线的分类--芯片内总线、元件级总线、内总线（即系统总线）、外总线（即通信总线）

常见的几种内总线：ISA 总线(长短两个插座，分别有 64 个、32 个接点)，EISA 总线，PCI 总线。其中 PCI 总线的工作与处理器的工作是相对独立的，即总线时钟和处理器时间是独立、非同步的，PCI 总线上的设备即插即用。

常见的几种外总线：RS-232C（是一条串行总线），SCSI（是一条并行总线），USB（由 4 条信号线组成，两条用于传送数据，另两条传送+5V 500mA 的电源），IEE1394（是一条串行总线，由 6 条信号线组成，两条传数据两条传控制信号两条传电源，支持即插即用和热插拔）

16、阵列处理机，又称并行处理机，它将重复设置的多个处理单元连成阵列，在控制部件的控制下，对分配给自己的数据进行处理，并行地完成一条指令规定的操作。这是一种单指令多数据流计算机（SIMD）

17、多处理机，是由多台处理机组成的系统。每台处理机有自己的控制部件，可以执行独立的程序，共享一个主存和所有外设。它是多指令流多数据流计算机。

按其构成分为：异构（非对称）型多处理机系统，同构（对称）型多处理机系统，分布式处理系统

4 种多处理机的结构：总线结构，交叉开关结构，多端口存储器结构，开关枢纽式结构

18、并行处理机，与采用流水结构的单机系统都是单指令流多数据流计算机，它们的区别是，并行处理机采用资源重复技术，而流水结构的单机系统使用时间重叠技术。

并行处理机有 2 种典型结构：具有分布式存储器的，具有共享式存储器的。它们的共同点是在系统中设置多个处理单元，各个处理器按一定
接方式交换信息，在统一的控制部件作用下，各自处理分配来的数据，并行的完成同一指令所规定的操作。

19、信息安全的基本要素

机密性
完整性
可用性
可控性
可审查性

20、计算机安全等级：技术安全性、管理安全性、政策法律安全性。一些重要的安全评估准则：“美国国防部和国家标准局的《可信计算机系统评测标准》TCSEC/TDI”、“欧共体的信息技术安全评估准则 ITSEC”、“ISO/IEC 国际标准”、“美国联邦标准”。其中 TCSEC/TDI 分了 4 个组 7 个等级，C2 是安全产品的最低等级。

21、安全威胁与影响数据安全的因素

安全威胁是指某个人、物、事件对某一资源的机密性、完整性、可用性或合法性所造成的危害。典型的安全威胁有很多种。

影响数据安全的因素有内部和外部两种。内部因素：可采取多种技术对数据加密；制定数据安全规划；建立安全存储体系；建立事故应急计划和容灾措施；重视安全管理并建立安全管理规范。

外部因素：按密级划分使用人员的权限；使用多种认证方式；设置防火墙；建立入侵检测、审计和追踪；同时注意物理环境的保护。

22、加密技术包括两个元素：算法和密钥。加解密算法设计的关键是满足 3 个条件“可逆性”，“密钥安全”，“数据安全”。

数据加密技术分为对称加密（以 DES 算法为代表）、非对称加密（以 RSA 算法为代表）、不可逆加密 3 种。

目前常用的对称加密算法有：DES 数据加密标准算法（使用 56 位密钥，对 64 位二进制数据块加密，基本加密运算为置换运算、移位运算 、模加运算）；

3DES（使用 2 个 56 位密钥，加、解、加）；

RC-5；

国际数据加密算法 IDEA（类似于 3DES，使用 128 位密钥，PGP 系统在使用该算法）

比较有名的非对称加密算法：RSA 算法，它是建立在大素数因子分解的理论基础上的算法。其公钥密码长度大于 100 位，算法运算速度较慢，多用于加密信息量小的场合，可以使用 RSA 算法来实现数字签名。

23、密钥管理，主要是指密钥对的管理，包括密钥的产生、选择、分发、更换和销毁、备份和恢复等。多密钥的管理可以使用 KDC。

24、数据完整性保护，是在数据中加入一定的冗余信息，从而能发现对数据的任何增删改。方法是在发送或写入时对所要保护的数据进行检验和作加密处理，产生报文验证码 MAC，附在数据后面。在接受或读出数据时根据约定的密钥对数据进行检验和作加密运算，将所得的结果与 MAC 比较，根据结果是否一致判断数据是否完整。

25、认证技术，主要是解决网络通信双方的身份认可。认证的过程涉及到加密和密钥交换。加密可使用对称加密、不对称加密和二者混合使用的方法。一般有账户名/口令认证、使用摘要算法认证、基于 PKI 公开密钥的认证。

PKI 是一种遵守既定标准的密钥管理平台，它能为所有网络应用提供加密和数字签名等密码服务及必需的密钥和证书管理体系。PKI 的基础技术包括加密、数字签名、数据完整性机制、数字信封、双重数字签名等。完整的 PKI 系统必须包括 CA、数字证书库、密钥备份及恢复系统、证书作废系统、应用接口 API 等基本部分。

PKI 使用证书进行公钥管理，通过 CA 将用户的公钥和用户其它住处绑在一起，以在因特网上验证用户身份。

26、HASH 函数，输入一个不定长的字符串，返回一个固定长度的字符串（即 HASH 值）。单向 HASH 函数用于产生信息摘要；信息摘要简要地描述了一份较长的信息或文件，可以被看作是一份文件的数字指纹，信息摘要用于创建数字签名。

27、数字签名的过程：

信息发送者使用一单向 HASH 函数对信息生成信息摘要；
信息发送者使用自己的私钥加密信息摘要；
信息发送者将信息本身和已签名的信息摘要一并发送出去；
信息接收者使用发送者的公钥对信息摘要解密，再使用同一单向 HASH 算法对信息生成信息摘要并进行验证是否一致。

28、数字加密的过程：

信息发送者先生成一个对称密钥，使用该密钥对信息加密；
信息发送者使用接收者的公钥加密上述对称密钥；
信息发送者将上两步的结果内容都传给接收者（这就是数字信封）；
信息接收者使用私钥解密对称密钥，并使用对称密钥解密信息本身。

29、SSL 安全协议，一个能够保证任何安装了 SSL 的客户和服务器之间事务安全性的协议，主要用于提高应用程序之间数据的安全系数。SSL 提供 3 方面服务：客户和服务器的合法性认证；加密传送的数据；保护数据的完整性。

30、数字时间戳技术，就是数字签名技术的一个变种，不同的是这个要由认证单位 DTS 提供数字签名。它的过程是：先形成需要加时间戳的信息的信息摘要；将信息摘要送到 DTS，DTS 记录收到的日期及时间；DTS 进行数字签名，然后送回用户。

31、计算机病毒的定义，它是一种程序，具有修改别的程序的特性，并使用被修改的程序也具有这样的特性。

病毒的特点：寄生性，隐毕性，非法性，传染性，破坏性。

按病毒的寄生方式和入侵方式分成：系统引导型病毒，文件外壳型，混合型病毒，目录型病毒，宏病毒（也叫数据病毒）。

需要注意的几点：变种、病毒程序加密、多形性病毒、病毒的伪装。

计算机病毒防治的手段：人工预防；软件预防；管理预防。

解决网络安全问题的技术包括：划分网段、局域网交换技术和 VLAN；加密技术、数字签名和认证、VPN 技术；防火墙技术；入侵检测技术；网络安全扫描技术。

32、计算机的 RAS 技术，R（可靠性）、A（可用性）、S（可维修性）。

计算机可靠性的模型有：串联系统模型、并联系统、N 模冗余系统。
串联系统可靠性 R = R1*R2*...Rn 平均故障率 = L1+L2+..Ln
并联系统可靠性 R = 1 - （1-R1）(1-R2）..（1-Rn)
N 模冗余系统由 2n+1 个子系统和一个表决器组成，只要 n+1 个子系统工作正常，系统就工作正常。

提高可靠性的办法：提高元件质量、改进加工工艺与工艺结构、完善电路设计、发展容错讲述。

33、计算机性能评测的常用方法：时钟频率法、指令执行速度法、等效指令执行速度法、数据处理速率法、核心程序法。

基准测试程序有，整数测试程序、浮点测试程序、SPEC 基准测试程序、TPC 基准程序。

34、计算机故障包括永久故障、间歇性故障和偶然故障。故障诊断分为故障检测和故障定位两方面。

容错，就是通过冗余方法来消除故障影响。硬件冗余有时间冗余和器件冗余两种。

故障处理步骤，封闭、检错、重复执行、诊断、重构与恢复、修复、重入。

35、BCD（Binary-Coded Decimal）码又称为“二—十进制编码”，专门解决用二进制数表示十进数的问题。

压缩 BCD 码

每一位数采用 4 位二进制数来表示，即一个字节表示 2 位十进制数。例如：二进制数 10001001B，采用压缩 BCD 码表示为十进制 数 89D。

非压缩 BCD 码

每一位数采用 8 位二进制数来表示，即一个字节表示 1 位十进制数。而且只用每个字节的低 4 位来表示 0～9，高 4 位为 0。

例如：十进制数 89D，采用非压缩 BCD 码表示为二进制数是：

00001000 00001001B

36、ASCII 是 AmericanStandardCodeforInformationInterchange 的缩写，用来制订计算机中每个符号对应的代码，这也叫做计算机的内码(code)。每个 ASCII 码以 1 个字节(Byte)储存，从 0 到数字 127 代表不同的常用符号，例如大写 A 的 ASCII 码是 65，小写 a 则是 97，阿拉伯数字 0 则是 48。由于 ASCII 字节的七个位，最高位并不使用。

第 0～32 号及第 127 号(共 34 个)是控制字符或通讯专用字符，如控制符：LF（换行）、CR（回车）、FF（换页）、DEL（删除）、BEL（振铃）等；通讯专用字符：SOH（文头）、EOT（文尾）、ACK（确认）等；

第 33～126 号(共 94 个)是字符，其中第 48～57 号为 0～9 十个阿拉伯数字；65～90 号为 26 个大写英文字母，97～122 号为 26 个小写英文字 母，其余为一些标点符号、运算符号等。

注意：在计算机的存储单元中，一个 ASCII 码值占一个字节(8 个二进制位)，其最高位(b7)用作奇偶校验位。所谓奇偶校验，是指在代码 传送过程中用来检验是否出现错误的一种方法，一般分奇校验和偶校验两种。奇校验规定：正确的代码一个字节中 1 的个数必须是奇数， 若非奇数，则在最高位 b7 添 1；偶校验规定：正确的代码一个字节中 1 的个数必须是偶数，若非偶数，则在最高位 b7 添 1。

37、按位与的特殊用途：

清零。 方法： 与一个各位都为零的数值相与，结果为零。

取一个数 x 中某些指定位。 方法： 找一个数，此数的各位是这样取值的：对应 x 数要取各位，该数对应位为 1，其余位为零。此数与 x 相就可以得到 x 中的某些位。

例：设 X=10101110

(1)取 X 的低 4 位
(2)取 X 的 bit2、bit4、bit6 位

38、某 EPROM 芯片上有 24 条地址线 A0-A23，8 条数据线 D0-D7，则该芯片的容量为“16M”。

EPROM 芯片上的地址线决定了该芯片有多少个存储单元，数据线数表明每个存储单元所存储的数据位数。24 条地址线则有 16M 个存储单元，8 条数据线决定了每个存储单元存 1 个字节。所以容量为 16M 字节。

39、机内码、国标码、区位码

根据汉字的国家标准，用两个字节（16 位二进制数）表示一个汉字。但使用 16 位二进制数容易出错，比较困难，因而在使用中都将其转换为十六进制数使用。国标码是一个四位十六进制数，区位码则是一个四位的十进制数，每个国标码或区位码都对应着一个唯一的汉字或符号，但因为十六进制数我们很少用到，所以大家常用的是区位码，它的前两位叫做区码，后两位叫做位码。

国标码规定，每个汉字（包括非汉字的一些符号）由 2 字节代码表示。每个字节的最高位为 0，只使用低 7 位，而低 7 位的编码中又有 34 个适用于控制用的，这样每个字节只有 27 - 34 = 94 个编码用于汉字。2 个字节就有 94 94=8836 个汉字编码。在表示一个汉字的 2 个字节中，高字节对应编码表中的行号，称为区号；低字节对应编码表中的列号，称为位号。

国标码与机内码转换关系：为了不与 7 位 ASCII 码发生冲突，把国标码每个字节的最高位由 0 改为 1，其余位不变的编码就是汉字字符的机内码 。也可以理解为国标码加上 8080H 后得到机内码，或是机内码减去 8080H 后得到国标码。

国标码与区位码转换关系：将国标码减去 2020H 后，得到区位码。

如某汉字机内码是 BFF0H，则国标码为 3F70H，区位码为 1F50H。

40、在采用三总线的运算器中，三条总线分别与运算器的两个输入一个输出相连接，各自有自己的通路。因此执行一次操作只需一步即可完成。在运算器的两个输入和一个输出上不再需要设置暂存器。

41、光盘上的信号是记录在光盘表面的凹坑及平面上。凹坑与平面的交接处代表 1，因此在光盘上不允许有连续的两个 1

42、磁盘非格式化容量 = 最大位密度*最内圈周长*总磁道数 --实际上就是使用磁盘的面积乘以位密度

格式化容量 = 每道扇区数*扇区容量*总磁道数

总磁道数为:（外半径 - 内半径）* 磁道密度

常识：有一个多盘片组成的盘组，在向磁盘记录一个文件时，如果超出了一个磁道容量，那么剩下的部分将存于其他盘面的同一编号的磁道上。因为盘组中的多个盘面形成一系列柱面，在向磁盘写入文件时会尽可能记录在同一柱面上，当一个柱面记录不下时，再记录到相邻的柱面上。

43、微指令根据编码方式的不同分为水平微指令和垂直微指令。
水平微指令，长度较长、操作具有高度并行性、编码简单、执行速度快，更多地体现了控制器的硬件细节；
垂直微指令，长度较短、并行度低、功能弱、效率低、编程容易但微程序长。

排列组合公式为： 求 n 上数中 m 个数的组合有多少， C = n(n-1)(n-2)..(n-m+1)/m!
例如求 n 个数中每 2 个数组合的可能性，C = n(n-1)/2 种可能性

2. 数据结构与算法

1、线性表的定义及特点

线性表是若干数据元素组成的有限集合；

线性表的特点是，有惟一的起始结点和惟一的终端结点，其它元素都有惟一的直接前驱和惟一的直接后继。

线性表的抽像数据类型定义包括 2 方面，
数据对象、关系的定义；
线性表有关操作的定义；
线性表的数据对象是具有相同性质数据元素的集合。
线性表的有关操作有：
基本操作：初始化线性表、撤消线性表、判/置空表、取表长、取前驱元素、取后继元素、取第 i 个元素、遍历等。
插删操作：在顺序结构下，结点的插入(n/2)和删除[(n-1)/2]主要是进行元素的移动；在链式结构下，结点的插删是调整指针的指向。
查找操作：在顺序表中可以进行折半查找，在链表中只能进行顺序查找。

2、线性表的基本存储结构及特点，线性表有顺序和链式两种存储结构。
顺序存储结构是：用一组地址连续的存储单元依次存储线性表中的数据元素；
链式存储结构是：用一组地址任意的存储单元存储线性表中的数据元素。（存储单元节点可以是连续的，也可以是不连续的）
链式存储结构包括，
单链表（又称线性链表），结点的结构体有两个域，分别存储数据元素和当前元素有关系的其它元素所在结点的指针
双向链表，每个结点包含两个指针，分别指明直接前驱和直接后继元素，可以在两个方向上遍历其后及其前的元素；
循环链表，链表中最后一个结点的指针指向第一个结点，开成环状结构，可以在任意位置上方向不变地遍历全表；
静态链表，借助数组描述线性表的链式存储结构。

3、栈的定义：是只能通过访问它的一端来实现数据存储和检索的一种线性数据结构。
栈的特点：是先进后出(FILO)。在线结构中，允许进行插、删操作的一端称为栈顶，相应另一端称为栈底。不含数据的栈称为空栈。
栈的基本运算有：置空栈、判空栈、元素入栈、出栈和读取栈顶元素的值。

栈的存储结构：顺序栈和链栈。
顺序栈指，用一组连续的存储单元依次存储自栈顶到栈底的元素，同时设置指针 top 指示栈顶元素的位置。顺序栈的空间容量是有限的，要预先定义。顺序栈的入栈和出栈操作是通过修改数组下标来完成。假设栈底对应于数组下标较大的一端，那么在元素入栈时就是下标减 1，而元素出栈时就是下标加 1。

链栈，类似于线性链表，栈顶指针就是链表首结点的位置，元素的插删操作限定在首结点处进行。
栈的应用：表达式计算，数制转换，括号匹配，迷宫问题，递归问题

4、队列的定义：是一种先进先出(FIFO)的线性表。
队列的特点：它只允许在表的一端插入元素而在表的另一端删除元素。在队列中允许插的一端叫队尾(rear)，允许删的一端叫队头(front)。
队列的基本运算：置队空、判队空、入队、出队、读队头元素等。
队列的存储结构：顺序队列和链队列。
顺序队列，又被叫作循环队列，设顺序队列 Q，Q.front 表示队头指针，Q.rear 表示队尾指针，则 Q.front 和 Q.rear

相等且为 0 时为空队列；元素入队时 Q.rear 加 1，元素出队时 Q.front 加 1。因为顺序队列的空间容量是提前设定的，所以当 Q.rear 达到了上限时表示队列满。　为区别队列空和队列满两种情况下可能出现的 Q.front == Q.rear，有两种方法。一个是设置一个标识位，以区别头尾指针相同时队列是空还是满；另一个方法是牺牲一个元素空间，约定以 Q.rear 所指的下一个位置是 Q.front 时表示队列满。

链队列，链队列为空的判定条件是头尾指针相同且均指向头结点。
队列的应用：常用于需要排队的场合，如操作系统中的打印队列，离散事件的复读机模拟等。

5、串的定义：是仅由字符构成的有限序列。是取值范围受限的线性表。一般记为 S = 'a1a2..an'。
串的几个概念：空串、空格串、子串、串相等、串比较。
串的几个操作：赋值操作 StrAssign(s,t)、联接操作 Concat(s,t)、求串长 StrLength(s)、串比较 StrCompare(s,t)、求子串 SubString(s,start,len)。
串的存储：静态存储(顺序存储)，是定长的存储结构。当串超长时，超过部分将被截断。
堆存储，通过程序语言提供的字符数组定义串的存储空间，事先不限定串的长度，在程序执行过程中动态地申请地址连续的串值的空间。
块链存储，使用链表存储串值，每个结点可以存储一个或多个字符，同时每个结点设置一个指针指向后继结点。
串的模式匹配：朴素的模式匹配法、KMP 算法。

6、数组：是定长线性表在维数上的扩张，即线性表中的每个元素又是一个线性表。N 维数组是一种同构的数据结构，其每个数据元素类型相同，结构一致。
数组的特点：　数组元素数目固定。一旦定义了一个数组结构就不再有元素的增减变化；
数据元素具有相同的类型；
数据元素的下标关系受上下界的约束且下标有序。
数组的基本运算：
给定一组下标，存取相应的数据元素；
给定一组下标，修改相应的数据元素中的某个数据项的值。

数组的存储：　数组的固定结构适于使用顺序存储。对于数组，只要知道它的维数和长度，就可以为它分配存储空间。反之，只要给出一组下标就可以求出该数组元素的存储位置。就是说，在数组的顺序存储结构中，数据元素的位置是其下标的线性函数。
以行为主序；　　Loc(Aij) = Loc(Aij) + ((i-1)*n + (j-1))*L
以列为主序；　　Loc(Aij) = Loc(Aij) + ((j-1)*m + (i-1))*L

多维数组的顺序存储计算：例如 3 维数组 A[1..10, 5..8, -3..6]，数组空间的起始位置是 a，每个元素占 4 个存储单元，试以行为主存储和以列为主存储时给出数组元素 A[i,j,k]的存储地址。
解：理解上面给出的以行为主序和以列为主序的两个线性函数公式。把 3 维数组拆开计算，例如以行为主序时先将 3 维数组看成是有一个行和 2 个列的数组，算出此时以行为主占用了多少空间。然后再单独看两个列的组合 B[j,k]又会占用多少空间。前后结果相加就是这个 3 维数组元素在以行为主序存储时的地址。如下，
以行为主序时，A[i,j,k]前面的元素个数是：　(i-1)(8-5+1)(6-(-3)+1) + (j-5)(6-(-3)+1) + k-(-3) = 40i-40 + 10j-50 + k+3 = 40i + 10j + k -87

因此 A[i,j,k]的地址为 a + (40i+10j+k-87)*4
以列为主序时，A[i,j,k]的地址为 a + (40k+10j+i+69)*4

7、特殊矩阵与稀疏矩阵，稀疏矩阵就是非零元素很少的矩阵，而特殊矩阵是非零元素分布有规律的一类矩阵。为节省空间，在存储它们时都使用压缩存储，特殊矩阵有压缩算法，稀疏矩阵使用三元组顺序表或使用十字链表存储矩阵元素。

8、广义表的定义：是由零个或多个单元素或子表所组成的有限序列。广义表的长度是指广义表中元素的个数，深度是指广义表展开后所含的括号的最大层数。

广义表的基本运算：取表头 head(LS)，非空广义表的第一个元素称为表头；
取表尾 tail(LS)，非空广义表中除第一个元素之外，由其余元素构成的表称为表尾。表尾必定是一个表。
Head(LS)=a1, Tail(LS)=(a2,a3,...,an)

9、树的定义：树是 n(n>=0)个结点的有限集合。当 n=0 时称为空树。在任一非空树中，有且仅有一个称为根的结点；其余 m 个结点可分为 m(m>=0)个互不相交的有限集，其中每个子集合又都是一棵树，称为根结点的子树。
树的定义是递归的，树形结构具有明显的层次结构。
树的术语：双亲和孩子，兄弟，结点的度，叶子结点，内部结点，结点的层次，树的高度，有序树和无序树，森林。
树的基本操作是：先根遍历和后根遍历。

10、二叉树的定义：二叉树是另一种树形结构，它的特点是每个结点至多有两棵子树并且有左右之分，且左、右子树的次序不能颠倒。

满二叉树，若二叉树上每一层的结点数目都达到最大值，则称为满二叉树；
完全二叉树，若二叉树的除第 H 层以外，其余各层的结点数目达到了最大值，而第 H 层上的结点集中存放在左侧，则称为完全二叉树；
非完全二叉树，就是完全二叉树的相反情况。

二叉树的性质：　1）二叉树第 i 层(i>=1)上至多有 2^(i-1)个结点；

2）深度为 K 的二叉树至多有 2^k -1 个结点(k>=1)；

3）对任何一棵二叉树，若其终端结点个数为 N0，度为 2 的结点个数为 N2，则 N0 = N2 + 1 ；

4）具有 n 个结点的完全二叉树的深度为 log(2,n)+1;

5）对一棵有 n 个结点的完全二叉树的结点按层次自左至右进行编号，则对任一结点 i (1<=i<=n)有：

若 i=1，则 i 是根结点；若 i>1 则其双亲为 i/2;

若 2i>n,，则结点 i 无左孩子，否则其左孩子为 2i;

若 2i+1>n,则结点 i 无右孩子，否则其右孩子为 2i+1;

例：一棵有 124 个叶结点的完全二叉树，最多有多少结点？

N0=N2+1

N=N0+N1+N2

N1=1

综合上面 3 个表达式可以求解。

例 2：具有 N 个结点的满二叉树，其叶子结点个数为多少？

设其深度为 h，则: N0=2^(h-1)

N = 2^h - 1

所以 N0 = (n+1)/2

二叉树的存储结构：

二叉树的顺序存储结构，若采用二叉树的性质 5 对树中的结点进行编号，即树根结点的编号为 1，若编号为 i 的结点存在左孩子，则其左孩子的编号为 2i；若编号为 i 的结点存在右孩子，则其右孩子的编号为 2i+1，这样利用数组元素的下标作为结点的编号，表示出结点间的关系。

二叉树的链式存储结构，二叉链表（有单向性）和三叉链表（有双向性）。

遍历二叉树，有 4 种方式：先序、中序、后序和层序遍历。

先序遍历二叉树的操作定义为：访问根结点；先序遍历根的左子树；先序遍历根的右子树。（若二叉树为空，则进行空操作）

中序遍历二叉树的操作定义为：中序遍历根的左子树；访问根结点；中序遍历根的右子树.（若二叉树为空，则进行空操作）

后序遍历二叉树的操作定义为：后序遍历根的左子树；后序遍历根的右子树；访问根结点。

层序遍历二叉树的操作定义为：从根结点开始，从或到右依次访问每层上的结点。

二叉树遍历思想的关键：首先在想象中把二叉树补齐为满二叉树，叶子结点也要被想象为有 2 个子结点。然后，画一条路线，从根出发，逆时针沿着二叉树的外缘移动，全程对每个结点均途经三次。若第一次经过时即访问，则是先序遍历；若是第二次经过结点时访问结点，则是中序遍历；若是第 3 次经过时访问则是后序遍历。这 3 种方法的路径相同，但结果不同。

遍历二叉树的基本操作就是，访问结点。--遍历二叉树实质上是按一定规则，将树中的结点排成一个线性序列。

11、线索二叉树：对于有 N 个结点的二叉树的二叉链表存储表示，其中必有 N+1 个空指针。遍历时使结点中原本为空的左孩子指针或（和）右孩子指针指向结点的前驱或（和）后继，这样的处理称为对二叉树的线索化，指向前驱或后继的指针称为线索。加上线索的二叉树称为线索二叉树。

为了区分结点中的指针是孩子还是线索，在结点结构中增加标志域 ltag, rtag。两个标志取值 0，则 lchild,rchild 域分别指向左孩子和右孩子；两个标志取值 1，则 lchild,rchild 域分别指向直接前驱和直接后继。

访问线索二叉树时，如何查找结点的前驱和后继？以中序线索二叉树为例，令 P 指向树中的某个结点，

当 p->ltag = 0 时，P 的中序直接前驱一定是其左子树进行中序遍历得到的最后一个结点，也可以沿 P 的左子树根结点出发沿右孩子指针向下查找，直到找到一个没有右孩子的结点时为止，该结点就是 P 的直接前驱结点，也称为 P 的左子树中“最右下”的结点。

当 P->rtag = 0 时，P 的中序直接后继一定是其右子树进行中序遍历得到的第一个结点，　也可以沿 P 的右子树根结点出发沿左孩子指针向上查找，直到找到一个没有右孩子的结点时为止，该结点就是 P 的直接后继结点，也称为 P 的右子树中“最左下”的结点。

12、二叉树的应用：最优二叉树（又称霍夫曼树），是一种带权路径长度最短的树。

路径，是从树中一个结点到另一个结点之间的通路，路径上的分支数目称为路径长度。

树的路径长度，是从根到每一个叶子结点之间的路径长度之和。

结点的带权路径长度，是从该结点到树根之间的路径长度与该结点权的乘积。

树的带权路径长度，是树的所有叶子结点的带权路径长度之和，记为 WPL 。

如何构造最优二叉树？使用霍夫曼算法如下：

1）将给定的 N 个结点的权值构成 N 棵二叉树的集合 F，其中每棵树 Ti 只有一个权为 Wi 的根结点，其左右子树为空；

2）在 F 中选取两棵根结点的权值最小的树作为左右子树，并新生成一个根结点，根结点的权值为左右子树的权值和；

3）从 F 中删除被取出的两棵树并将新生成的树放入 F；

4）重复 2，3 步骤到只剩一棵树为止，这棵树就是最优二叉树。最优二叉树的形式不唯一，但其 WPL 值却是唯一确定的。

霍夫曼编码：若要设计长度不等的编码，则任一字符的编码都不是其他字符编码的前缀，这种编码称为“前缀编码”。要设计总长最短的二进制前缀编码，应以 N 种字符出现的频率作为权来构造一棵霍夫曼树，由此得到的二进制前缀编码称为霍夫曼编码。树的左右分枝分别标上 0 和 1（或相反）。从根到叶子路径上的 0，1 组成的串就是每个字符的二进制编码。

13、树的存储结构

1）树的双亲表示法，用一组连续的存储单元存储树的结点，并在每个结点中附设一个指示器，指示其双亲结点在该存储结构中的位置；

2）树的孩子表示法，是在存储结构中用指针指出结点的每个孩子。要为树的每个结点的孩子建立一个链表，则 N 个结点的树具有 N 个单链表，这 N 个单链表的头指针又排成了一个线性表（头指针即树的存储结构中每个结点的指示器）。

将上两种方法结合起来可以形成树的双亲孩子表示法。

3）树的孩子兄弟表示法，是指用二叉链表表示树。在链表的结点中设置两个指针域，分别指向该结点的第一个孩子和下一个兄弟。 |firstchild| data |nextbrother| 若将树的孩子指针解释为左孩子、兄弟指针解释为右孩子，则可以得到这棵树的二叉树结构。

14、树的遍历：

先根遍历；

后根遍历。

树进行先根遍历也就是对转换得到的二叉树进行先序遍历；对树进行后根遍历也就是对转换得到的二叉树进行中序遍历。

（先根遍历的顺序是：由根出发从左至右遍历每棵子树。后根遍历的顺序是从左至右从每棵子树的叶子结点向根的方向访问子树，最后访问根结点。）

15、森林的遍历：

先序遍历森林；

中序遍历森林。

先序遍历森林，若森林非空，访问森林中第一棵树的根结点，先序遍历第一棵子树根结点的子树森林，再先序遍历除第一棵树之外的树所构成的森林。

中序遍历森林，若森林非空，中序遍历森林中第一棵树的子树森林，再访问第一棵树的根结点，再中序遍历除第一棵树以外的树所构成的森林。

16、树、森林和二叉树的转换

利用树的孩子兄弟表示法可以由一棵树转成唯一的一棵二叉树。

森林如何转换成二叉树呢？因为树根没有兄弟，所以树转换成二叉树后一定没有右子树，所以森林转换成二叉树的方法是：

1）先将森林中的每棵树全转成二叉树；

2）用第一棵树的根做新二叉树的根，第一棵树转为二叉树后得到的左子树做为新二叉树的左子树，第二棵树作为新二叉树的右子树，第三棵树作为新二叉树的右子树的右子树，依此类推，森林便转为了一棵二叉树。

17、图的定义：在数据结构中，图是一个由顶点集合和边集合构成的二元组，其中边表示顶点之间的关系。

图的主要术语：

有向图，图中每条边都是有方向的，弧、弧尾、弧头；

无向图，图中的边是没有方向的，边；

无向完全图，图中的 N 个结点之间每两个结点间都有边，共有 n(n-1)/2 条边；

有向完全图，图中的 N 个结点之间每两个结点间都有方向相反的两条弧，共有 n(n-1)条弧；

度、入度、出度，顶点 v 的度是指关联于该顶点的边的数目，记作 D(v)。若是有向图则以该顶点为终点的有向边数目称为入度，从该顶点出发的有向边的数目称为出度，有向图的度是入库和出度的和。

路径，两个顶点之间由边组成的一条通路。若是有向图则路径也有方向。路径长度是路径上边或弧的数目。第一个顶点和最后一个顶点相同的路径称为回路。若首尾顶点以外的顶点均不相同则是简单路径，若只有首尾顶点相同则称为简单回路。

子图，一个图的顶点集合与边集合都从属于另一个图，则称之为另一个图的子图；

连通图与连通分量，在无向图中若两个顶点之间有路径则称为这两个顶点是连通的。若无向图中任两个顶点间都是连通的则称其为连通图。该无向图的最大连通子图称为它的连通分量。

强连通图与强连通分量，是有向图的连通概念；

网，边（弧）带权值的图称为网；

生成树，是一个极小的连通子图，它包括图中的全部顶点，但只有构成一棵树的 n-1 条边；

有向树和生成森林，一个有向图恰有一个顶点的入度为 0 其它顶点的入度均为 1，则这是一棵有向树。生成森林是一个有向图中的若干棵有向树组成，特点是含有全部顶点但只有足以构成若干棵不相交的有向树的弧。

图的存储结构：

邻接矩阵表示法，用于表示图有顶点之间的关系。对于个有 n 个顶点的图 G＝（V，E）来说，其邻接矩阵就是一个 n 阶方阵。依靠判断图的两顶点间是否存在边或弧来决定 Aij=1 或 Aij=0；网的邻接矩阵，当两顶点间存在边或弧时 Aij 等于权值否则 Aij 等于无穷。

邻接链表表示法，为图的每个顶点建立一个单链表，单链表中的结点表示依附于相应顶点的边或弧，有表头结点和表结点两种结构类型。

图的遍历：深度优先搜索；广度优先搜索。一个类似于先根遍历，一个类似于层序遍历。

生成树的概念：生成树是连通图的一个子图，它由全部顶点和一次遍历图所经过的边组成。图的生成树不惟一，按深度优先搜索得到深度优先生成树，按广度优先搜索得到广度优先生成树。一个非连通图，每个连通分量中的顶点集和遍历时走过的边集一起构成若干棵生成树，称为非连通图的生成树森林。

18、最小生成树：连通网的边是带有权值的，将生成树的各边权值和称为生成树的权。其中权值最小的生成树称为最小生成树。

构造最小生成树的两种算法：

普里母算法：以一个顶点集合 U 作为初态，不断寻找与 U 中顶点相邻且代价最小的边的另一个顶点，扩充 U 至 U＝V 时为止。例如初始只给 U 一个顶点且边的集合 TE＝｛｝；这种算法的时间复杂度为 O（n^2），因为它由顶点推算出的，所以适合于边稠密的网的最小生成树。

克鲁斯卡尔算法：假设连通网 N＝（V，E），令最小生成树的初始状态为只有 n 个顶点而无边的非连通图 T＝（V，｛｝），图中每个顶点自成一个连通分量。在 E 中选择代价最小的边，若该边依附的顶点落在 T 中不同的连通分量上，则将此边加入到 T 中，否则舍去此边而选择下一条代价最小的边。信此类推，直至 T 中所有顶点都在一个连通分量上为止。这种算法与顶点数无关，所以适合计算顶点多而边稀疏的网的最小生成树。

19、AOV 网(active on vertex)：在有向图中，以顶点表示活动，用有向边表示活动之间的优先关系，这样的网称为 AOV 网。在 AOV 网中不应出现有向环。

拓朴排序：是将 AOV 网中所有顶点排成一个线性序列的过程，并且该序列满足：若在 AOV 网中从顶点 Vi 到 Vj 有一条路径，则在该线性序列中，顶点 Vi 必然在 Vj 之前。

拓朴排序的方法：在 AOV 网中选一个入度为 0 的顶点并输出它；从网中删除该顶点及与其有关的边；重复前两步至网中不存在入度为 0 的顶点为止。这样操作会有两种结果：一个是所有顶点已输出，也就是拓朴排序完成，说明网中不存在回路；另一个可能结果是尚有未输出的结点，剩余顶点均有前驱顶点，表明网中存在回路！

也可以进行逆拓朴排序，即计算出度为 0 的顶点。拓朴算法的时间复杂度为 O(n+e)。

AOE 网(active on edge)：，在带权有向图中，以事件表示顶点，以边表示活动，以边上的权值表示活动持续的时间，则这种网称为用边表示活动的网，简称 AOE 网。

AOE 网特点：

1）顶点所表示的事件是指该顶点的所有进入边所表示的活动已完成，所有发出边表示的活动可以开始的一种状态。

2）对一个工程来说，要有一个开始状态和一个结束状态，所以在 AOE 网中有一个入度为 0 的开始顶点，称为源点；有一个出度为 0 的结束顶点，称为汇点。AOE 网中也不允许存在回路。

3）完成整个工程的时间是从开始顶点到结束顶点间的最长路径的长度（指该路径上的权值和）。

活动的松驰时间：用活动的持续时间和该活动两侧的两个事件的关键路径时间，二者取差。

关键路径：从源点到汇点的路径长度最长路径称为关键路径。关键路径上的所有活动均是关键活动。

最短路径？？？

20、查找的基本概念

1）查找是一种常用的基本运算。查找表是由同一类型数据元素构成的集合；

2）静态查找表，指在进行查找运算时不再修改的已经构造好的查找表。静态查找表只进行两种操作：查询某个特定的数据元素是否在查找表中；检索某个特定的数据元素的各种属性。

3）动态查找表，是可以进行另两种操作的查找表，即在查找表中插入一个数据元素；从查找表中删除一个数据元素。

4）关键字，是数据元素的某个数据项的值，用它来识别这个数据元素；

5）主关键字，指能唯一标识一个数据元素的关键字；

6）次关键字，指能标识多个数据元素的关键字；

7）查找，根据给定的某个值，在查找表中确定是否存在一个其关键字等于给定值的记录，并返回结果。

顺序查找，从表的一端开始，逐个进行记录的关键字和给定值的比较，若找到一个记录的关键字和给定值相等则查找成功；若整个表均比较过仍未找到则查找失败。若要查找的记录不在表中则需进行 n+1 次查找。

平均查找长度为(n+1)/2。

折半查找（二分查找），可以用二叉树进行分析，以中间记录为根，左子表为左子树，右子表为右子树，依此类推。关键字的比较次数即为被查找结点在树中的层数。查找成功或失败时所比较的关键字数不超过树的层数。折半查找只适用于有序顺序表（以数组方式存储的有序表）。 n=2^k - 1, k=log(n+1)

分块查找，又称为索引顺序查找，综合使用上面两种方法。

将长度为 n 的表均匀分为 b 块，每块含有 s 个记录，按顺序查找确定元素所在的块，则 ASL = Lb + Lw， 即块内查找与索引查找之和。

ASL=(b+1)/2 + (s+1)/2，当 s 取 n 的平方根时，ASL 取得最小值“n 的平方根加 1"。

21、二叉排序树（又称二叉查找树）：二叉查找树或者是一棵空树，或者具有这样的特性，

1）若二叉查找树的左子树非空，则左子树上的结点值均小于根结点的值；

2）若它的右子树非空，则右子树上的结点值均大于根结点的值；

3）左、右子树的子身就是一棵二叉查找树。

二叉查找树的查找过程从根结点开始，过程类似于折半查找（二分查找）。二叉查找树的插入操作按它的特性法则进行插入，若是空树则作根结点，否则会成为一个新的叶子结点。在二叉查找树中删除一个结点时不能把该结点的子树也删掉，只能删除这个结点但仍要保持二叉查找树的特性，相当于是从一个有序序列中删除一个元素，不能破坏其它元素的有序性。一种方法是，如果删除结点＊P，则可以用＊P 的直接前驱或直接后继代替＊P，同时删除它的直接前驱（或直接后继）。

二叉排序树顺序存储在一地址连续的空间内，则序列按中序递增存储。

22、平衡二叉树：它或者是一棵空树，或者具有这样的性质：它的左右子树都是平衡二叉树，且左右子树的深度之差的绝对值不超过 1。

平衡因子： 某结点的平衡因子定义为该结点的左子树深度减去它的右子树深度。平衡二叉树上的所有结点的平衡因子只可能是－1、0 和 1。

为了得到树形均匀的二叉排序树，在构造二叉排序树的过程中可以使用几种办法让它保持为一棵平衡二叉树。每插入一个新结点时，就检查是否打破了平衡。若是，则找出最小不平衡二叉树，在保持二叉排序树特性的情况下，调整最小不平衡二叉树中结点间关系，达到新的平衡。

最小不平衡二叉树是指离插入结点最近且以平衡因子的绝对值大于 1 的结点作为根的子树。

平衡二叉树上的插入操作引起不平衡的解决方法：

1）单向右旋平衡处理 －－用于在根的左子树根结点的左子树上插入新结点情况

2）单向左旋平衡处理 －－用于在根的右子树根结点的右子树上插入新结点情况

3）双向旋转（先左旋后右旋）操作 －－用于在根的左子树根结点的右子树上插入新结点的情况

4）双向旋转（先右旋后左旋）操作 －－用于在根的右子树根结点的左子树上插入新结点的情况

B 树，有几个比较鲜明的特点。如：一棵 m 阶的 B 树中每个结点至多有 m 棵子树；非终结点（根除外）至少有 m/2 棵树；根至少有两棵子树（当根不是叶子时）；所有叶子结点出现在同一层次上。

23、哈希表的定义：根据设定的哈希函数 H（key）和处理冲突的方法，将一组关键字映射到一个有限的连续地址集上，并以关健字在地址集中的“像”作为记录在表中的存储位置，这种表称为哈希表。这一映射过程称为哈希造表或散列，所得的存储位置称为哈希地址或散列地址。哈希函数是从关键字集合到地址集合的映像。

对于哈希表主要考虑两个问题：一是如何构造哈希函数；一是如何解决冲突。

构造哈希函数要解决好两个问题：首先哈希函数是一个压缩映像函数；其次哈希函数应具有较好的散列性。前者为节省空间，后者为减少冲突。常用的哈希函数构造方法有直接定址法、数字分析法、平方取中法、折叠法、随机数法和除留余数法。

处理冲突的方法：

1）开放地址法 Hi = (H(key) + Di)%m i=1,2,...,k (k<=m-1)

H(key)为哈希函数；m 为哈希表的表长；Di 为增量序列。

Di=1,2,3,...,m-1 称为线性探测再散列；

Di=1^2,-1^2,2^2,-2^2...,k^2 (k<=m/2) 称为二次探测再散列；

Di＝伪随机序列，称为随机探测再散列。

最简单的产生探测序列的方法是线性探测，即当冲突时顺序对下一单元进行探测并存储。在用线性探测法解决冲突构造的哈希表中进行查找时有 3 种可能结果：一是在某一位置上找到关键字等于 key 的记录，查找成功；一是按探测序列查找不到而又遇到了空单元，查找失败，此时可进行插入操作；一是查遍全表，未查到指定关键字且存储区已满，要进行溢出处理。

线性探测法的缺点是“溢出处理需别编程序”，“很容易产生聚集现象”。

2）链地址法 ，它在符号表的每一个记录增加一个链域，链域中存放下一个有相同哈希函数值的记录的的存储地址。

3）再哈希法 ，Hi = RHi(key) i= 1,2,..,k RHi 均是不同的哈希函数，即在同义词发生地址冲突时计算另一个哈希函数地址，直到解决。

4）建立一个公共溢出区，一溢出全放这里去；

哈希表的装填因子， a = (表中添入的记录数/哈希表长度） ——a 越小，发生冲突的可能越小

虽然哈希表在关键字与记录的存储位置之间建立了直接映像，但由于“冲突”的产生，使得哈希表的查找过程仍是一个给定值和关键字进行比较的过程。仍须以平均查找长度衡量哈希表的查找效率。

在查找过程中须与给定值进行比较的关键字的个数取决于 3 个因素：哈希函数、处理冲突方法、哈希表的装填因子。

24、排序：稳定的排序、不稳定的排序。
内部排序、外部排序。

简单排序法：包括直接插入排序、冒泡排序和简单选择排序。它们的算法复杂度为 O（n ^ 2），在元素已经基本有序的情况下，使用直接排序方法可获得较高的效率 O(n)。直接插入排序和冒泡排序是稳定的排序方法，简单选择排序是不稳定的排序方法。

直接插入排序适用于”在文件局部有序或文件长度较小的情况下的一种最佳内部排序方法“。直接插入排序的时间复杂度为 O(n*n)，若记录序列为正序时其时间复杂度可提高到 O(n)。正序？？

冒泡排序算法：
```
void bubblesort(int data[], int n){
    int i,j,tag,temp;
    for(i=0,tag=1;tag==1&&i<n-1;i++){
        tag = 0;
        for(j=0;j<n-i-1;j++)
            if(data[j]>data[j+1]){
                temp=data[j];data[j]=data[j+1];data[j+1]=temp;
                tag=1;
            }
    }
    }
```

简单选择排序算法：
```
void selectsort(int data[], int n){
    int i,j,k,temp;
    for(i=0;i<n-1;i++){
        k=i;
        for(j=i+1;j<n;j++)
        if(data[j]<data[k])k=j;
        if(k!=j){
        temp=data[i];data[i]=data[k];data[k]=temp;
        }
    }
    }
```

希尔排序，又称为缩小增量排序。它是在直接插入排序的基础上加以改进得到的排序方法。基本思想就是：设定一个初始间隔 d,d<n，按间隔 d 将元素分组，在每一组内进行直接插入排序，可以使得最小元素跳跃式向前移动。然后缩小增量 d 的值，重复上述操作到 d=1 时为止。

快速排序，基本思想是通过一趟排序将待排序的记录分割为独立的两部分，其中一部分的关键字均比另一部分小，然后再分别对这两部分记录继续进行排序。具体做法：在头尾设两个指针 low,high，分别指向第一个元素和最后一个元素。设枢轴记录为正向（返向）的第一个记录。当初始序列有序时，快速排序蜕变为冒泡排序，此时算法的时间复杂度为 n*n。

例如，对 50 个整数进行快速排序时，因为初始序列有序，所以排序过程退化为冒泡排序，总过程中的比较次数为 49+48+...+1 = 49*50/2

堆排序，基本思想是对一组待排序记录的关键字，首选把它们按堆的定义排成一个序列，从而输出堆顶的最小关键字。然后将剩余关键字再调整成堆，便得到次小关键字，反复进行，直至全部关键字排成有序序列。

归并排序，是将两个或多个有序表合并成一个新的有序表。是一种稳定的排序。

基数排序，是一种借助多关键字排序思想对单逻辑关键字进行排序的方法。它不是基于关键字比较的排序方法，其平均时间复杂度为 O(d*n)，适合于 n 值很大而关键字较少的序列。

基于关键字比较的内部排序方法的时间复杂度的下限为 O(nlogn)，简单排序、希尔排序、快速排序、堆排序和归并排序是要熟练掌握的排序方法。（重要）

排序方法好坏的两条因素：执行算法的时间；执行算法所需要的附加空间。

常用的外部排序方法是归并排序。

25、分治法，将一个规模为 n 的问题逐步分解为 k 个规模更小的子问题，这些子问题互相独立且与原问题性质相同，逐个解决分解出的子问题，由这些子问题的解构造出原问题的解，当 k=2 时称为二分法。如解决棋盘覆盖问题。

分治法适用的问题一般具有这些特征：

原问题可以分解成多个子问题，这些子问题与原问题相比只是规模的下降而其结构和求解方法与原问题相同；

若子问题的规模足够小，则直接求解，否则递归地求解子问题；

在得到各子问题的解后，能采用某种方法构造出原问题的解。

动态规划法，与分治法类似也是先将待求解的问题分解成若个个子问题，先求解子问题，然后从子问题的解得到原

问题的解。不同的是适合于动态规划法求解的问题所分解得到的子问题之间往往不是独立的。动态规划法常用来求解具有最优性质的问题。即问题的最优解包含了其子问题的最优解。如求解多边形游戏问题。

设计动态规划法的步骤为：

1）找出最优解的性质，并刻画其结构特征；
2）递归地定义最优值；
3）以向前或向后处理方式计算出最优值；
4）根据计算最优值得到的信息，构造最优解。

贪心法，通过一系列的选择来得到一个问题的解，它所做的每一次选择都是当前情况下某种意义的最好选择，即贪心选择。如果待求解的问题具有最优子结构特征，也就是原问题的最优解包含子问题的最优解，并且可以通过局部的贪心选择来达到问题的全局最优解时，可通过贪心法进行求解。贪心标准的选择和问题的结构决定是否可以使用贪心法。如用于二分最小覆盖问题。

回溯法，又被称为通用解题法，用它可以系统地搜索问题的所有解。回溯法是一个既带有系统性又带有跳跃性的搜索算法。它在问题的解空间中按深度优先策略，从根结点出发搜索解空间树。算法搜索到解空间树的任意结点时，首先判断该结点是否包含问题的解。如果不包含则跳过对以该结点为根的子树的搜索，逐层向其祖先结点回溯；否则进入这棵子树继续按深度优先搜索。如收费公路重建问题。

分支限界法，它类似于回溯法，也是在解空间中搜索问题解的算法，但分支限界法求解的目标是找出满足条件的一个解，如有多个解则要找出某种意义下的最优解。分支限界法以广度优先或以最小耗费优先的方式搜索解空间树。如最大完备子图问题。

26、算法的几个基本特征

有穷性，确定性，能行性，输入，输出。

程序 ＝ 数据结构 ＋ 算法

内容关键字：

线性表、栈、队、串、数组

树、二叉树、森林、线索二叉树、霍夫曼树

图、有向图、无向图、最小生成树、拓朴排序、关键路径

查找、静态查找（顺序、折半、分块）、动态查找（二叉排序树、平衡二叉树、哈希表）

排序、直接插入、简单选择、冒泡、希尔、快速、堆、归并、基数

3. 操作系统知识

1、操作系统的定义：是管理计算机中各种软件、硬件资源的程序和相关文档的集合，是一种系统软件。

操作系统能有效的组织和管理系统中的各种软、硬件资源，合理地组织计算机工作流程，控制程序的执行，并且向用户提供一个良好的工作环境和友好的接口。

操作系统的两个重要作用：

通过资源管理，提高系统的使用效率；
改善人机界面，向用户提供友好的工作环境。

操作系统的 4 个特征：并发性、共享性、虚拟性、不确定性。

操作系统的 5 个管理功能：进程管理、文件管理、存储管理、设备管理、作业管理

操作系统的分类：

批处理系统，计算机自动、顺序地执行作业流产生的每一个作业，以节省人工操作时间和提高机器的使用效率。分为单道批处理系统和多道批处理系统。优点是同一批内的各作业次次执行，改善了 cpu,io 的使用效率，提高了吞吐量。缺点是磁盘需要人工装卸，作业需要人工分类，监督程序易受用户程序破坏，缺少交互性。

分时系统， 具有如下特征：多路性、独立性、交互性、及时性。

实时系统，分为实时控制系统和实时信息处理系统。主要特点有：快速的响应时间、有限的交互能力、高可靠性

网络操作系统，使得计算机更有效地共享网络资源，为网络用户提供所需各种服务的软件和有关协议的集合。

分布式操作系统，是由多个分散的计算机经网络连接而成，各主机无主次之分。为分布式计算机配置的操作系统

称为分布式操作系统。
微机操作系统
嵌入式操作系统

2、研究操作系统的观点
资源管理的观点：从这种观点看，操作系统的管理对象是计算机系统的资源，操作系统则是管理计算机系统的程序集合。这种观点是在共享的前提下以资源分配、使用和回收为出发点，考虑操作系统各部分程序的功能和算法。

虚拟机的观点：操作系统加裸机构成虚拟计算机。虚拟机的观点是从功能分解的角度出发，考虑操作系统的结构，将操作系统分成若干层次，每一层完成特定的功能。

3、顺序程序执行时的特征：顺序性、封闭性、可再现性；

并发程序执行时的特征：非封闭性、程序和机器执行程序的活动不在一一对应、并发程序间的相互制约性。

引入进程的原因：由于程序并发执行破坏了程序的封闭性和可再现性，使得程序和执行程序的活动不在一一对应，此时用静态的程序概念已经不能描述系统中程序动态执行的过程，所以引入了进程。

4、进程的定义：就是程序的一次执行，该程序可以和其它程序并发执行。

进程的组成：进程通常是由程序、数据及进程控制块（PCB）组成的。
进程的程序部分是进程执行时不可修改部分，它描述了进程需要完成的功能；
进程的数据部分是进程的可修改部分；
进程控制块是进程的描述信息和控制信息，是进程存在的惟一标志。

进程和程序的区别是：进程具有状态而程序没有。

5、进程的状态及状态间的切换

三态模型：运行、就绪、阻塞。

五态模型：新建态、终止态、运行、就绪、阻塞。

新建态：对应于进程刚刚被创建时还没有被提交，并等待系统完成创建进程的所有必要信息的状态。整个过程分为两个阶段，一是为一个新建进程创建必要的管理信息，另一是让进程进入就绪状态。因为有了新建态，操作系统可以根据系统的性能和主存的容量限制而推迟新建态的提交。

终止态也分为两个阶段，一是等待操作系统进行善后处理，另一是释放主存。

具有挂起状态的进程状态：当系统资源不能满足所有进程的运行要求时，必须将某些进程挂起，放在磁盘对换区，暂时不参加调度，以平衡系统负载。有这样几个状态：活跃就绪、静止就绪、活跃阻塞、静止阻塞。

6、进程的控制，就是对系统中所有进程从创建到消亡的全过程实施有效的控制。操作系统的内核为系统实现进程控制和存储管理提供了有效的控制机制。
大多数操作系统内核均包含支撑功能和资源管理功能。
支撑功能：中断处理、时钟管理、原语操作。
原语是由若干条机器指令构成的，用于完成特定功能的一段程序。内核在执行某些基本操作时往往是通过原语操作实现的。原语在执行过程中不可分割。内核中包含的原语有进程控制、进程通信、资源管理等。

资源管理功能：进程管理、存储器管理、设备管理。

7、进程间通信
进程间的同步：一般来说，一个进程相对于另一个进程的运行速度是不确定的，即进程是在异步环境下运行。每个进程都以各自独立的不可预知的速度向前推进，但相互合作的进程需要在某些确定点上协调它们的工作，当一个进程到达了这些点后，除非另一进程已完成了某些操作，否则就不得不停下来等等这些操作结束。
进程间的互斥：在多道程序系统中，各进程可以共享各类资源，但有些资源一次只能供一个进程使用，称为临界资源(critial resource)。同步是进程间的直接制约问题，互斥是进程间的间接制约问题。

临界区(critial section)是对临界资源实施操作的那段程序。互斥临界区管理的原则为：有空即进、无空则等、有限等待、让权等待。

8、整形信号量与 PV 操作

整形信号量是一个整形变量，根据控制对象的不同赋不同的值。信号量分为两类：
公用信号量：实现进程间的互斥，每个相关进程即可对它施行 P 操作也可以进行 V 操作，初值为 1 或资源的数目；
私用信号量：实现进程间的同步，只有一个进程可以对它施行 P 操作，其它进程只能做 V 操作，初值为 0 或某个正整数。

信号量 S 的物理意义：S＞＝0 表示某资源的可用数，S＜0 则其绝对值表示阻塞队列中等待该资源的进程数。

PV 操作是实现进程同步与互斥的常用方法。PV 操作是低级通信原语，其中 P 操作表示申请一个资源，V 操作表示释放一个资源。

P 操作定义：S:=S-1，若 S＞＝0，则执行 P 操作的进程继续执行；否则若 S＜0，则该进程为阻塞状态，并将其插入阻塞队列。
V 操作定义：S:=S+1，若 S＞0，则执行 V 操作的进程继续执行；否则，若 S＜＝0，则从阻塞状态唤醒一个进程，并将其插入就绪队列，执行 V 操作的进程继续执行。

利用 PV 操作实现进程的互斥：令信号量 mutex 的初值为 1，当进入临界区时执行 P 操作，临界区时执行 V 操作。
P(mutex)
临界区
V(mutex)

怎样利用 PV 操作实现进程的同步：可用一个信号量与消息联系起来，当信号量的值为 0 时表示希望的消息未产生，当信号量的值为非 0 时表示希望的消息已经存在。假定用信号量 S 表示某条消息，进程可以通过调用 P 操作测试消息是否到达，调用 V 操作通知消息已准备好。最典型的是单缓冲区的生产者和消费者的同步问题。如果采用 PV 操作来实现进程 PA 和进程 PB 间的管道通信，并且保证这两个进程并发执行的正确性，则至少需要 2 个信号量，信号量的初值分别为 0、1。

9、高级通信原语，因为 PV 操作不足以描述复杂的进程间的信息交换，所以引入高级通信原语。

高级通信原语有这么几种：共享存储系统、消息传递系统、管道通信。

进程通信有直接和间接两种方式。间接方式是以信箱以为媒介。

10、管程(monitor)：另一种同步机制，采用资源集中管理的方法，将系统中的资源用某种数据结构抽象地表示出来。由于临界区是访问共享资源的代码段，因而建立一个管程来管理进程提出的访问请求。采用这种方式对共享资源的管理就可以借助数据结构及在其上实施操作的若干过程来进行。对共享资源的申请和释放可以通过过程在数据结构上的操作来实现。

11、进程调度，在某些系统中一个作业从提交到完成需要经历高、中、低三级的调度。
高级调度（又称长调度、作业调度或接纳调度），它决定输入池中的哪个后备作业可以调入主系统做好运行的准备，成为一个或一组就绪进程。
中级调度（又称对换调度），它决定处于交换区中的哪个就绪进程可以调入主存，以便直接参与 CPU 的竞争。
低级调度（又称进程调度），它决定处于主存中的哪个进程使用 CPU。

调度方式，是指当有更高优先级的进程来到时如何分配 CPU。调度的方式分为可剥夺式和不可剥夺式两种。

常用的调度算法：先来先服务，主要用于宏观调度，有利于长作业，有利于 CPU 繁忙的作业；
时间片轮转，主要用于微观调度，提高了并发性和响应时间，最终提高了资源利用率；
优先级调度，　分为静态和动态两种；
多级反馈调度，是在时间片轮转和优先级算法的基础上改进得到。其特点是：照顾了短进程以提高系统吞吐量，照顾 I/O 型进程以获得较好的 I/O 设备利用率并缩短响应时间，不必估计进程的执行时间和动态调节优先级。

12、死锁：就是指两个以上的进程相互请求对方已经占有的资源时而导致无法继续运行下去的现象。

几种会产生死锁的情况：进程推进程顺序不当，同类资源分配不当，PV 使用不当。

进程资源有向图：由方框、圆圈和有向边 3 部分组成。其中资源用方框表示，进程用圆圈表示。在方框中每一个小圆圈代表一个资源。有向边分别代表请求资源和分配资源。

死锁产生的原因：因为竞争资源或进程推进顺序非法。进程推进顺序仍是关于进程请求和释放资源的顺序。

死锁产生的 4 个必要条件：互斥条件、请求保持条件、不可剥夺条件、环路条件。
互斥是说进程对所要求的资源有排它性控制。请求保持是说进程断续地请求资源，但后续的资源被阻塞。环路是指在发生死锁时在进程资源有向图中，每个进程都占有了下一个进程请求的一个或多个资源。

死锁的 4 种处理：鸵鸟策略；
预防策略，即破坏死锁产生的 4 个必要条件之一；
避免策略，即精心分配资源，主动回避死锁；
检测与解除死锁

13、线程
传统的进程有两个基本属性，即可拥有资源的独立单位，和可独立调度、分配的基本单位。引入线程后，将传统进程的两个属性分开，线程作为可独立调度和分配的基本单位，进程作为独立拥有资源的单位。因此，用户可以通过创建

线程来完成任务，以减少程序并发执行时的时空开销。

14、存储器的结构：(寄存器)--缓存－主存－辅存。

虚拟地址，又称为逻辑地址、相对地址、程序地址。它是从 0 号单元开始编址，并顺序分配所有的符号名所对应的地址单元，它不是主存中的真实地址。
地址空间，又称逻辑地址、虚地址。
存储空间，又称物理地址空间，是物理地址的集合。相对地址空间通过地址再定位机构转换到绝对地址空间。

重定位：程序的逻辑地址被转换成主存的物理地址的过程称为地址重定位。分为静态重定位和动态重定位。

静态地址重定位的优点是无需硬件地址变换机构的支持，它的缺点是必须为程序分配连续的存储区域且执行期间不能扩充不能移动并难以共享；
动态地址重定位要依赖于硬件的地址变换机构。它解决了静态重定位的各种缺点。

进行存储管理的目的是：对主存空间进行分配和管理；主存扩充；存储保护；提高空间的利用率。
主存扩充技术，通过交换和覆盖实现，其中交换是由操作系统实现，覆盖是由操作系统提供覆盖机制但由用户进行控制。

15、分区存储管理，按分区方式的不同分为固定分区、可变分区、可重定位分区。

可变分区有 4 种请求和释放分区的算法：最佳适应算法、最差适应算法、首次适应算法、循环首次适应算法。

为减少分区碎片而使用的可重定位算法，基本思想是移动所有已分好的分区，使其靠拢成为连续区域。

分区保护管理：有 2 种方法。一是“上界/下界寄存器”，另一种是“基址/限长寄存器”的方法。其中上界寄存器和基址寄存器都是放的作业的装入地址。下界寄存器放作业的结束地址，限长寄存器放作业的长度。因此调入作业所需要的物理地址必需满足：

上界寄存器＜＝物理地址＜＝下界寄存器
或　基址寄存器＜＝物理地址＜＝物理地址＋限长寄存器

分区管理方案是解决多道程序共享主存的可行方案，但它要求用户的程序必须装入地址连续的空间中。

16、页式存储管理

分页原理：将一个进程的地址空间划分成若干大小相等的区域，称为页。相应地将主存空间划分成与页相同大小的若干物理块，称为块或页框。在为进程分配主存时，将进程中若干页分别装入多个不相邻的块中。

地址结构由 2 部分组成：页号＋页内地址

页表：又称为页面映射表。作用是实现从页号到物理块号的地址映射。

快表：是页表方式的改良，是在地址映射机构中增加一个联想存储器（是由一组高速存储器组成），这就是所谓的快表。它用来保存当前访问频率最高的少数活动页的页号及相关信息。　另外还有一种方法是增加高速寄存器来保存页表，但这样的成本太大。

两级页表机制：是为了减少页表占用的连续地址空间，而提出的方法。使用两级或多级页表机制来存储页表。

17、分段存储管理

原理：在分段式存储管理系统中，为每个段分配一个连续的分区，而进程中的各个段可以离散地分配到主存的不同分区中。在系统中为每个进程建立一张段映射表，简称段表。每个段在表中占有一个项，记录该段在主存中的起始地址（基址）和段的长度。进程在执行时，通过查段表来找到每个段所对应的主存区。因此，段表实现了逻辑段到物理主存区的映射。

分段系统的地址结构：段号(名)＋段内地址

特点：段是信息的逻辑单位，因此分段的一个突出优点是易于实现段的共享，即若干个进程共享一个或多个段，而且对段的保护也很简单。在分页系统中，虽然也能实现程序和数据的共享，但远不如分段系统方便。

段页式存储管理，原理是先将主存划分为大小相等的存储块（页框），再将用户程序按程序的逻辑关系分为若干个段，为每个段命名，然后将每个段划分为若干个页，以页架为单位离散分配。

段页式系统的地址结构：段号＋段内页号＋页内地址

18、虚拟存储管理

程序的局部性：时间局限性和空间局限性。前者指程序中的某条指令或某个存储单元一旦被执行或访问，则在不久的将来可能会再次发生（因为程序中存在着大量的循环操作）；后者指一旦程序访问了某个存储单元，则不久的将来该存储单元附近的存储单元也最有可能被访问（因为程序是顺序执行的）。

虚拟存储器，从用户的角度看，是这样一个系统，它所具有的主存容量比实际主存容量大得多。它是根据局部性原理，在一个作业运行之前只把部分程序和数据装入主存，其余部分留在磁盘上。如果要访问的页或段未在主存中(称为缺页或缺段)则将它们调入主存。

虚拟存储器的实现：
请求分页系统，它是在分页系统的基础上，增加了请求调页和页面置换功能后所形成的页式虚拟存储系统。

请求分段系统，它是在分段系统的基础上，增加了请求调段和段置换功能后所形成的段式虚拟存储系统。

请求段页式系统，它是在段页式基础上，增加了请求调页和页面置换功能后所形成的段页式虚拟存储系统。

其中请求分页系统是目前常用的一种虚拟存储器方式。其页面置换算法的好坏直接影响系统性能，不当的置换算法可能会导致系统“抖动”。常用的页面置换算法有：最佳置换算法、先进先出置换算法、最近最久未使用置换算法和最近未用置换算法。

虚拟存储器的特征：离散性、多次性、对换性、虚拟性。工作集的概念是指在某段时间间隔里，进程实际要访问的页面的集合。
虚存容量不是无限的，它受主存和外存可利用的总容量限制；虚存还受计算机总线地址结构限制。虚存的扩大是以牺牲 CPU 工作时间和主存与外存交换时间为代价的。虚存是由操作系统调度，采用主存外存交换技术，各道程序在必须使用时调入主存，不用的程序则调出主存。

19、设备管理，包括各种设备分配、缓冲区管理和实际物理 I/O 设备操作，通过管理达到提高设备利用率和方便用户使用的目的。

设备的分类
按数据组织分为：块设备 ，如磁带、磁盘
字符设备，如打印机、交互式终端
按资源分配分为：独占设备，如打印机
共享设备，如磁盘
虚拟设备，如利用假脱机技术将一台独占设备变为多个用户共享的逻辑设备。
按数据传输速率：低速设备，如键盘、鼠标
中速设备，如打印机
高速设备，如磁盘

设备管理的目标是如何提高设备的利用率，为用户提供方便统一的界面。
设备管理的任务是保证在多道程序环境下，当多个进程竞争使用设备时，按一定策略分配和管理各种设备，控制设备的各种操作，完成 I/O 设备与主存之间的数据交换。

20、I/O 软件

IO 设备管理软件分为 4 层：由低到高为中断处理程序－－设备驱动程序－－与设备无关的系统软件－－用户级软件

设备驱动程序是直接同硬件打交道的软件模块，它与 IO 设备的硬件结构有密切的联系。它的任务就是接受来自与设备无关的上层软件的抽象请求，进行与设备有关的处理。

设备的 IO 方式：
通道 ， 使数据的传输独立于 CPU，CPU 只须向通道发出 IO 命令，由通道完成 IO 任务后再向 CPU 发出中断信号。
DMA ， 是指数据在主存和 IO 设备之间直接传送，CPU 只需要在首尾做些处理。
缓冲技术，缓冲区技术可提高外设利用率，使外设尽可能处于忙状态。分为硬件缓冲(由硬件寄存器实现)和软件缓冲(由操作系统实现)。缓冲技术的优点是：可以缓和 CPU 与 IO 设备间速度不匹配的矛盾；减少 CPU 的中断频率，放宽对中断响应时间的限制；提高 CPU 和 IO 设备之间的并行性。

21、Spooling 技术

Spooling 是外围设备联机操作的简称，又称为假脱机系统。Spooling 实际上是用一类物理设备模拟另一类物理设备的技术，是使独占使用的设备变成多台虚拟设备的技术，是一种速度匹配技术。
Spooling 由预输入程序、缓输出程序、井管理程序、输入井输出井组成。

Spooling 系统中拥有一张作业表来登记进入系统的所有作业的作业名、状态、预输入表位置等信息。每个作业拥有一张预输入表来登记该作业的各个文件的情况，包括设备类、信息长度及存放位置等。（包括图）

输入井中的作业有 4 种状态：提交、后备、执行、完成。

22、磁盘调度，分为移臂调度和旋转调度两种。并且是先进行移臂调度，然后再进行旋转调度。因为访问磁盘最耗时的

是寻道时间，所以磁盘调度的目标是减少磁盘的平均寻道时间。

磁盘驱动调度， 常用的磁盘调度算法有先来先服务 FCFS、最短寻道时间 SSTF、扫描算法 SCAN(又称为电梯调度算法)、单向扫描调度算法 CSCAN、N-Step-SCAN 算法(磁臂粘着)、FSCAN 算法。

FCFS 的优点是简单，缺点是平均寻道时间太长；SSTF 的优点是每次的寻道时间最短，缺点是不能保证平均寻道时间最短，且有高度局部化的倾向，会推迟某些请求以致引起饥饿；SCAN 的优点是避免了饥饿现象，缺点是可能有个别请求被严重延迟；C－SCAN 为的是避免 SCAN 的缺点

旋转调度算法， 该算法用来计算，当移动臂定位后，有多个进程等待访问该柱面时，这些进程的访问顺序。系统应该选择延迟时间最短的进程对磁盘的扇区进行访问。

23、文件：具有符号名的、在逻辑上具有完整意义的一组相关信息项的集合。文件是一种抽象机制，它隐藏了硬件和实现细节。
文件管理系统：就是操作系统中实现文件统一管理的一组软件和相关数据的集合，是专门负责管理和存取文件信息的软件机构，简称文件系统。
文件系统的功能：按名存取、统一的用户接口、并发访问和控制、安全性控制、优化性能、差错恢复。

文件的结构和组织：文件的结构是指文件的组织形式。从用户的角度看到的文件组织形式称为文件的逻辑结构；从实现的角度看文件在存储器上的存放方式，称为文件的物理结构。

文件的逻辑结构分为 2 类：一是有结构的记录式文件；另一是无结构的流式文件。

文件的物理结构，决定了文件的逻辑块号到物理块号的转换方式。常见的物理结构有：连续结构(顺序结构)、链接结构、索引结构、多个物理块的索引表（链接、多重索引表、unix 的索引结构）。索引顺序文件既适合于交互方式应用，也适合于批处理方式应用。

文件目录，就是文件控制块的有序集合。文件控制块 FCB 是用于描述和控制文件的数据结构。常见的目录结构有 3 种：一级目录结构，二级目录结构，多级目录结构。

文件的存取方法有顺序和随机两种。

磁盘分配表，就是外存进行空间管理的数据结构。

常用的空闲空间管理方法：位示图、空闲表法、空闲链表及成组链接法。

文件的使用：文件系统为每个文件与该文件在磁盘上的存放位置建立了对应关系。文件系统通过用户给出的文件名查找对应文件的存放位置并读出内容。在多用户环境下，操作系统为每个文件建立和维护关于访问权限等方面的信息。为此操作系统在操作级和编程级为用户提供文件服务。

文件共享：是指不同用户使用同一文件。有多种共享形式，采用文件名与文件说明分离的目录结构有利于实现文件共享。

在 Unix 系统中允许多用户基于索引结点的共享，或利用符号链接共享同一个文件。基于索引结点的共享方式又有静态共享和动态共享两种方式。这样子，会在打开文件表、系统打开文件表、内存 i 结点表及磁盘间形成一副关系图。这种关系图在辅导教材的 155 页的几个例子中有图解，可以体味。

符号链接会增加系统的读盘次数，而硬链接的共享文件的目录文件表目中已包括了共享文件的索引结点号。

文件保护：文件系统对文件的保护采用存取控制方式进行。存取控制就是不同的用户对文件的访问规定不同的访问权限。常用的存取控制方式有，存取控制矩阵、存取控制表、用户权限表、密码。

存取控制矩阵，就是一个二维矩阵，一维列出全部用户，另一维列出全部的文件，每个矩阵元素表示某个用户对某个文件的存取权限。
存取控制表，就是按用户对文件的访问权力的差别对用户进行分类，该存取控制表可存放在每个文件的文件控制块中。UNIX 使用的这种方式，用 9 位二进制数表示三类用户对文件的存取权限，该权限存在文件索引节点的 di_mode 中。
用户权限列表，以用户或用户组为单位将用户可存取的文件集中起来存入表中，表中的每个条目表示该用户对相应文件的存取权限。这相当于把存取控制矩阵简化为一行。

系统的安全性：分为 4 个级别，系统级、用户级、目录级和文件级。

文件系统的可靠性：转储与恢复，日志文件，文件系统的一致性。

24、作业，是系统为完成一个用户的计算任务所做的工作总和。作业中的每个步骤又称为作业步。

作业控制：分为脱机控制和联机控制两种方式。在脱机控制中用户必须使用作业控制语言（JCL）编写作业说明书，并同作来一同提高给系统

作业控制块 JCB：是记录作业各种有关信息的登记表。JCB 是作业存在的惟一标志，其中包括用户名、作业名和状态标志等信息。JCB 被用于在输入井中形成作业后备队列。

作业的 4 种状态：提交、后备、执行和完成。注意它们的状态转换图。

作业调度算法：先来先服务算法、短作业优先、响应比高者优先、优先级、均衡调度算法。其中响应比是取值于“作业响应时间除以作业执行时间”，作业响应时间是作业时间与作业等待时间之和。

作业周转时间 ＝ 作业完成时间－作业提交时间 ，N 个作业的平均周转时间就是取 N 个作业的周转时间平均值。
作业带权周转时间 ＝ （作业完成时间－作业提交时间）/ 作业执行时间

25、UNIX 操作系统

UNIX 系统的结构：它是一种多用户、多任务的分时操作系统，一般由存储管理、进程管理、设备管理和文件系统管理几个部分组成。

unix 文件系统的目录结构是树形带交叉勾连的，根目录记为"/"。目录是一个包含目录项的文件。进程可以通过系统调用访问文件。unix 文件系统的布局如图所示：
丨引导块丨超级块丨索引结点区丨数据存储区丨

Unix 进程的组成：由控制块 PCB、正文段和数据段组成。

Unix 进程的控制：有一个进程控制子系统，提供了如 fork,exec,exit,wait,signal,kill,msgsnd,msgrcv 等系统调用，以完成进程的同步、通信、存储及调度。

Unix 进程的调度：采用优先数算法，进程的优先数随进程的运行情况而变化。

Unix 进程的存储：早期采用对换技术；高版本的 Unix 的主存管理采用的分页式虚拟存储机制，以对换技术作为辅助手段。

Unix 的设备管理：Unix 上包括两类设备，即块设备和字符设备。Unix 设备管理有这样的特点，
块设备与字符设备具有相同的层次结构（对它们的控制方法和所采用的数据结构、层次结构相同）；
将设备作为一个特殊文件并赋予一个文件名（文件存取与对设备的使用，具有了统一的接口）；
采用完善的缓冲区管理技术（预先读、异步写、延迟写）。

26、Windows 操作系统

Windows 的体系结构：通过硬件实现了核心态和用户态两种特权状态。核心组件使用了面向对象的设计原则，一般不能直接访问某个数据结构中由单个组件维护的消息，这些组件只能使用外部接口传送参数访问或修改这些数据。

Windows 的核心态模块有：核心、执行体、硬件抽象层、设备驱动程序、图形引擎。

Windows 的文件系统：NTFS 使用 64 位簇进行索引，NTFS 的特征有可恢复性、安全性、大磁盘和大文件、多数据流和通用索引功能。

在 Windows 中进程是资源分配的单位，并将进程作为对象来进行管理。Windows 的线程是内核线程，是处理机的调度单位。

存储管理，Windows 默认使用二级页面表结构来转换物理地址和虚拟地址。

Windows 的设备管理，建立了广义的资源管理概念，并统一地用对象模型来描述和规范化，大大降低了系统的复杂性。在输入输出上，建立了一个一致的高层界面－－IO 设备虚拟界面。将所有的读写数据看成直接送往虚拟文件的字节流。

4. 程序设计语言基础

1、程序设计语言的基本概念

低级语言和高级语言

编译程序和解释程序：解释程序会直接解释执行源程序或者将源程序翻译成某种中间表示形式后再执行。
编译程序则会将源程序翻译成目标语言程序，然后在计算机上运行目标程序。

二者的根本区别是，在编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序都不参加目标程序的执行过程；而在解释方式下，解释程序和源程序要参与到程序的运行过程中，运行程序的控制权在解释程序。解释器翻译源程序时不生成独立的目标程序，而编译器则需将源程序翻译成独立的目标程序。

程序设计语言的定义：一般地，程序设计语言涉及 3 个方面，语法、语义和语用。语言的实现有个语境问题，包括编译环境和运行环境。

2、程序设计语言的分类：按程序设计方法的不同分为 4 种。分别是命令式程序设计语言和结构化设计语言、面向对象的程序设计语言、函数式程序设计语言、逻辑型程序设计语言。

命令式程序设计语言：它是基于动作的语言，也称为过程式语言。随着函数、库、模块的使用，出现了结构化程序设计技术。在结构化程序设计中任何程序段的编写都基于 3 种基本结构，就是顺序、选择、循环。典型的实例有 Pascal,C。

面向对象的程序设计语言：面向对象的语言一般包括这 3 个概念，对象、类、继承。对象是人们要研究的任何事物，它具有状态和操作；类是由用户定义的数据类型，它将具有相同状态、操作和访问机制的多个对象抽象成一个对象类，属于这种类的一个对象叫作类实例或类对象，类代表一般而该类的一个对象代表具体；继承，类与类之间可以组成继承层次，以达到概念复用和代码重用。

函数式程序设计语言：是一种面向值的语言，其基本概念来自 LISP。主要应用于符号数据处理，如微积分、数理逻辑、游戏推演以及人工智能。

逻辑型程序设计语言：是陈述式语言，其基本概念来自 PROLOG，不是严格的通用程序设计语言。PROLOG 的基本运算单位是 Horn 子句。主要用在人工智能领域，也用在自然语言处理、数据库查询、算法描述等，尤其适合作为专家系统的开发工具。

FORTRAN 是世界上最早出现的高级程序设计语言，是由一个主程序或一个主程序与若干个子程序组成，且都是独立的程序单位；

COBOL 是一种面向事务处理的高级语言，主要用于情报检索、商业数据处理等管理领域；

ALGOL 是另一个较早出现的高级语言，是一个分程序结构语言，每个分程序由 begin 和 end 括起来；

PASCAL 语言体现了结构化程序设计风格，将分程序和过程这两个概念合并为“过程”。一个 PASCAL 程序本身可看成是一个操作系统所调用的过程；

C 语言在系统应用和实时处理应用中成为主要的开发语言；

C＋＋中最主要的是增加了类机制，成为一种面向对象的设计语言，并最大限度的与 C 兼容；

JAVA 是一种新型的面向对象的 Internet 编程语言，扩充了对分布式及 C/S 结构的支持，是一种强类型语言，隐含了指针以避免由于指针引起的问题；

LISP 是基于表处理的函数语言，该语言中的程序和数据的形式是等价的，数据结构可以作为程序执行，程序也可以作为数据修改

3、程序设计语言的基本成分：包括数据、运算、控制和传输。

数据成分，是程序操作的对象，具有存储类别、类型、名称、作用域和生存期等属性，使用时要为它分配内存空间。常量、变量、全局量、局部量。

运算成分，指明允许使用的运算符号及运算规则。

函数：函数的定义，函数的声明，函数的调用。函数的定义包括函数首部和函数体。函数应先声明后引用。函数调用时实参与形参间交换信息的方法有传值调用和引用调用两种。

传值调用中，若函数调用时以实参向形式参数传递相应类型的值，这种方式下，形式参数将不能向实际参数返回信息；除非使用用指针作形参，在调用时先对实参进行取地址运算，然后将实参地址传递给指针形参，这样才可以实现被调用函数对实际参数的修改。

4、汇编程序的基本原理

汇编语言是面向机器的符号化程序设计语言。计算机需要使用汇编程序对汇编源程序进行翻译才能运行。一般汇编语言都提供指令语句、伪指令语句、宏指令语句进行编程。
指令语句，　又称机器指令语句，汇编后能产生相应的机器代码，可以被 CPU 直接识别执行。
伪指令语句，指示汇编程序在汇编源程序时完成某些工作，如给变量分配存储单元地址，给某个符号赋值。
宏指令语句，允许用户将多次重复使用的程序段定义为宏，宏指令语句就是对宏的引用。

指令语句与伪指令语句的区别：指令语句经汇编后将产生相应的机器代码，而伪指令语句不产生机器代码；指令语句是在程序运行时完成，而伪指令语句只能在源程序被汇编时完成。

汇编程序：它的基本工作是将每一条可执行汇编语句转换成对应的机器指令；处理源程序中出现的伪指令和宏指令。汇编程序一般需要扫描源程序 2 次才能完成翻译过程，第一次主要是计算符号的值，第二次才产生目标程序。

5、编译程序的工作阶段，编译程序的过程分为 6 个阶段，另有 2 个辅助的管理程序。分为是：词法分析、语法分析、语义分析、中间代码生成器、代码优化、目标代码生成 6 个阶段和符号表管理、出错处理程序。中间代码的特征是与具体的机器无关。

代码优化和中间代码生成两个阶段并不是每种编译程序都必须的。

语法分析中的预测分析法是自顶向下的一种语法分析方法。

编译器在语义分析阶段进行表达式的类型检查及类型转换。

编译过程的各个阶段都会涉及到表格管理和出错处理。

5. 网络基础知识

1、网络的拓朴结构

计算机网络拓朴主要是指通信子网的拓朴构型。网络拓朴影响着网络的性能，以及整个网络的设计、功能、可靠性和通信费用

总线结构：只有一条双向通路，便于采用广播式传送信息；总线拓朴结构属于分布式控制，无须中央处理器，结构简单；节点的增删和位置的变动容易，扩充性能好；节点的接口通常采用无源线路，可靠性高；设备少价格低，安装使用方便；但因为电气信号通路多，干扰较大，对信号的质量要求高。负载重时线路的利用率低。网上的信息延迟不确定，故障隔离和检测困难。

星状结构：维护容易，配置灵活；故障隔离和检测容易；网络延迟时间短节点与中央交换单元直接连通，各节点之间的通信必须经过中央单元转换；网络共享能力差；线路利用率低，中央单元负荷重。

环状结构：环形网中信息流动方向是固定的，两个节点间只有一条通路，路径控制简单；有旁路设备，节点一旦发生故障，系统自动旁路，可靠性高；信息要串行穿行多个节点，传输效率低，系统响应速度慢；环路封闭，扩充较难。

树状结构：是总线结构的扩充形式，主要用于多个网络组成的分组结构中。

分布式结构：无严格的布点规定，各节点之间有多条线路相连。有较高的可靠性，资源共享方便，网络响应时间短；因为节点与多个节点相连，故节点的路由选择和流量控制难度大，管理复杂；硬件成本高。

2、网络的协议和标准

一个网络协议主要包括 3 个要素：语法、语义和时序。协议是一组约定的规则，它有助于通信实体间的相互理解和正确通信。协议中有 3 个要素，其中语法定义数据的表示形式；语义则能使数据管理所需的信息得到正确理解；时序则规定了通信应答信号之间的间隔和先后关系。

在 IEEE802 局域网标准中只定义了物理层和数据链路层。其中又把数据链路层分为了逻辑链路控制 LLC 和介质访问控制 MAC。

以太网 IEEE802.3 采用的是带冲突检测的载波监听多路访问协议技术 CSMA/CD。802.3,802.3u,802.3z。

802.3u 采用非屏蔽双绞线，并使用与 802.3 一样的介质访问控制（MAC）层；802.3z 对 MAC 规范进行了重定义，并重定义了物理层标准。

令牌环网 IEEE802.5，FDDI 类似于令牌环网协议，但是采用双环技术。

PPP 协议，具备用户验证能力，可以解决 IP 分配。PPP 和其他协议共同派生出了 PPPoE 和 PPPOA，主要用于 ADSL。

ADSL，非对称用户数据线，速率上行 1M 下行 8M，把线路按频段分成语音、上行和下行 3 个信道。

数字专线 DDN，综合业务数字网 ISDN

帧中继 FR，双工并保持顺序不变。是一种基于可变帧长的数据传输网络。适用于带宽需求 64K－2M，通信距离较长，数据有突发性时。

异步传输模式 ATM，是一种面向分组的快速分组交换模式，使用异步时分复用技术，将信息流分割成定长的信息元。共有 4 层，用户层、ATM 适配层、ATM 层、物理层。有 2 种连接类型，永久虚电路和交换虚电路。

TCP/IP 协议是 internet 协议的核心，分为 5 方面：逻辑编址、路由选择、域名解析、错误检测、流量控制及对应用程序的支持。

TCP/IP 分层模型由 4 层组成，应用层（FTP,telnet,smtp,nfs,snmp）、传输层（TCP，UDP）、网际层（IP,icmp,arp,rarp）、网络接口层(802.3,802.5,FDDI,ppp）。

TCP/IP 的传输层任务是提供应用程序之间的通信服务，这种通信又叫端到端的通信。网际层又叫 IP 层，它接收传输层的请求，传送某个具有目的地址信息的分组。网络接口层又称为数据链路层。

3、构建网络

网络互联的设备有：中继器（及集线器）、网桥（及交换机）、路由器、网关。

在构建一个网络的过程中，主要考虑服务器、客户机、网络设备、通信介质、、网络软件等，以及协议的选择和设备的连接方式。

4、关于 IP 地址

IP 地址分为 5 类：A，B，C，D，E。

A 类网络地址占有 1 个字节，定义最高位为 0 来标识此类地址。余下 7 位为真正的网络地址，支持 1－126 个网络。第一个字节的 10 进制表示为 000－127。后面 3 个字节作为主机地址，提供 2^24 - 2 个端点的寻址。

B 类网络地址占有 2 个字节，定义最高位为 10 来标识此地址。余下 14 位为真正的网络地址，第一个字节的 10 进制表示为 128－191。

C 类网络地址占有 3 个字节，定义最高位为 110 来标识此地址。余下 21 位为真正的网络地址，第一个字节的 10 进制表示为 192－223。

D 类网络地址用于组播，定义最高位为 1110 来标识此类地址。第一个字节的 10 进制表示为 224－239。

E 类网络地址为实验保留，定义最高位为 1111 来标识此类地址。第一个字节的 10 进制表示为 240－255。

可变长子网掩码 VLSM，就是在 IP 地址后面加上“/网络号及子网络号编址比特数”。如：193.168.125.0/27，就表示前 27 位为网络号。

5、网络的信息安全：主要就是信息的存储安全和传输安全。信息的存储安全包括信息的使用安全（用户的标识与验证、用户存取权限限制、安全问题跟踪），计算机病毒的防治，系统安全监控，数据的加密，防止非法攻击。

WindowsNT 的网络结构中，包括的两个接口是 NDIS 和 TDI。通过边界定义了各个层次间的统一接口，两个主要的边界层为 NDIS 和 TDI。
NDIS，网络设备接口规范；
TDI，传输驱动程序接口。

防火墙技术经历了包过滤、应用代理网关和状态检测三个发展阶段。

包过滤路由器是最简单常用的防火墙。一般工作在网络层，对经过网络的信息进行分析并按策略进行限制，其核心是包过滤的算法设计。优点是速度快、实现方便；缺点是安全性差、兼容差，日志记录能力差。

双宿主主机防火墙，由具有两个以上网口的堡垒主机构成，通过代理服务器软件从一个子网访问另一个子网。优点是加强了日志功能；缺点是若堡垒主机被攻破意味着失去了网络的安全。

屏蔽主机网关防火墙，是由过滤路由器和应用网关组成。过滤路由器的作用是进行包过滤；应用网关的作用是代理服务。共建立了两道安全屏障。优点是安全性高；缺点是配置复杂。

被屏蔽子网防火墙，由两个包过滤路由器和一个应用网关（堡垒主机）组成。两个包过滤路由器中间形成一个 DMZ 区。

6、重发器也称为中继器或转发器，是一种在物理层上互联网段的设备。

网关也称为信关，工作在应用层，实现网络间协议转换的功能，也被称为协议转换器。

Kerberos 是分布式环境下的身份认证系统。为了防止 relay 攻击，它使用了一次性的 ticket 和时间戳。常用的数字证书格式有 PGP 和 X.509 证书。

SSL 是要建立一条安全的连接。是传输层安全协议。

HTTPS 用于安全地传送单个报文，属于应用层协议。

SOCKS5 是增加了认证功能的 SOCKS 协议。SOCKS 用于代理基于 TCP/IP 的网络应用。SOCKS 服务器端实现于应用层，SOCKS 客户机实现于应用层和传输层之间。协议的作用是在两个没有直接 IP 联系的主机之间实现通信。

SNMP 是一种广泛使用的网络管理协议，用来收集网络上设备信息。其对应的管理信息库为 MIB-2。

7、OSI 参考模型的三个主要概念是 Service, Interface, Protocol。
OSI/RM 中的 1－3 层负责通信功能，称为通信子网。5－7 层属于资源子网的功能范围，称为资源子网层。传输层起着承接作用。
物理层，只是为它的上一层提供一个物理连接，在这一层数据还没有被组织；
数据链路层，负责两个相邻结点间的线路上无差错地传送以帧为单位的数据，并进行流量控制。数据链路层要负责建立、维持和释放数据链路的连接；
网络层，为传输层提供端到端的交换网络数据传送功能，屏蔽传输细节，为传输层建立、维持和拆除一条或多条通信路径。在这一层帧被组成数据包；
传输层，为会话层提供透明可靠的数据传输服务，保证端到端的数据完整性。选择网络层的最适宜服务，提供建立、维护、拆除传输链接的功能。在这一层传输的是报文；
会话层，为表示层实体提供建立、维护、结束会话连接的功能。完成通信进程的逻辑名字与物理名字间的对应，提供会话管理服务；
表示层，为应用层提供能解释所交换信息含义的一组服务，提供格式化的表示和转换数据服务，数据的压缩、解压、加密和解密工作也是由表示层完成；
应用层，提供 OSI 用户服务，提供网络与用户应用软件间的接口服务。

8、ISDN 为了使通信网络内部的变化对终端用户是透明的、不可见的，它必须提供一个标准的用户接口。

宽带 ISDN 可以提供许多业务，其中会议电视属于会话型业务。窄带 ISDN 向用户提供基本速率 144Kb/s 的基本速率接口 BRI，和速率 2Mb/s 的一次群速率接口 PRI。

双绞线多用于 10BASE－T 和 100BASE－T 的以太网中，一段双绞线的最大长度为 100m，只能连接一台计算机。双绞线的每端需要一个 RJ45 插头，各段双绞线通过集线器相连，利用双绞线最多可连接 64 个结点到中继器。

屏蔽双绞线 STP，非屏蔽双绞线 UTP。10BASE-T, 10BASE-F 的最后一个字母是以线缆类型命名的，T 表示双绞线，F 表示光纤。

以太网遵循 IEEE802.3 标准。采用粗缆的标准称为 10Base5，规定每段粗缆的长度不超过 500 米。采用细缆的标准称为 10BASE2，工作距离为 185 米。否则要使用重发器（即中继器）相连。整个网的长度不能超过 2500 米。若超过该长度则要分成两个网，网间使用网桥相连。这是在数据链路层的连接。

千兆以太网支持 3 种传输介质。多模光纤工作距离为 500 米，单模光纤的工作距离为 2000 米；宽带同轴电缆的工作距离只有 25 米；5 类 UTP 双绞线仍然是最大传输 100 米。

符合以太网标准的物理地址采用连续编码方法，它使用的地址长度是 48bit。

域名解析的两种主要方式是反复解析和递归解析。

从网络高层协议角度看，网络攻击可以分为服务攻击与非服务攻击。

防火墙一般可提供 4 种服务，它们是服务控制、方向控制、行为控制和用户控制。

防火墙是一种被动的网络安全措施。

6. 多媒体基础

1、媒体可分为感觉媒体、表示媒体、表现媒体、存储媒体、传输媒体。通常所说的媒体包括两个含义：一是指信息的物理载体；二是指承载信息的载体，即信息的表现形式，即 CCITT 定义的存储媒体和表示媒体。其中的表示媒体又可以分为 3 种：视觉类、听觉类、触觉类。

多媒体体就是指多种信息载体的表现形式和传递方式。

超文本是一种文本管理技术，它以结点为单位组织信息。结点、链和网络是超文本的 3 个基本要素。

超媒体，具体表现为用超文本方式组织和处理多媒体信息。

2、声音 3 要素：音强、音调、音色。音色由混入基音的泛音所决定。人耳听觉范围是 20HZ-20KHZ。

声音信号的数字化：取样－量化法。有 3 个步骤，即取样、量化、编码。

波形声音，是一个用来表示声音强弱的数据序列，它是由模拟声音经采样、量化和编码后得到的一种数据格式。

有 3 种声音编码方法：波形编码，参数编码，混合编码。

声音合成，分为语音合成和音乐合成。

MIDI，是乐器数字接口的缩写，目前已成为数字音乐的国际标准。它规定了乐器间及与计算机进行数据传输的协议规范。

声音文件的格式：Wave，Module(.mod)记录音色样本，MPEG(mp3)，RealAudio(.ra)，MIDI，Voice(.voc)每个 voc 文件由文件头块和音频数据块组成，Sound(.snd)，Audio(.au)，AIF,CMF。

3、色彩 3 要素：亮度、色调、色饱和度。

彩色空间：指彩色图像所使用的颜色描述方法。RGB 彩色空间，用于计算机显示器；CMY 彩色空间，是相减混色，典型应用有油墨、颜料；YUV 彩色空间，Y 表示亮度，U,V 是两个色度变量，典型应用有电视图像。

图形与图像：图形是用一系列计算机指令来描述和记录的一幅图的内容，典形应用有矢量图；图像是用像素点来描述的图。图形采用光栅化（即点阵化）技术可以转换为图像，图像采用图形跟踪技术可以转化为图形。

现实图片数字化过程为采样－量化－编码。

图像的主要属性包括分辨率、像素深度、真/伪彩色。像素深度指存储每个像素所使用的位数。真彩色，图像的每个像素的颜色由 R、G、B 三个基色分量所决定。伪彩色，是在生成图像时对图像中的不同色彩采样并生成一个彩色查找表，图像的每像素值存储的是色彩的索引。

图像的无损压缩方法有：行程编码 RLE、增量调制编码、霍夫曼编码。增量调制法只记录每行上第一个像素的值，其后的像素只记录与行首记录的增量。霍夫曼法是根据像素出现的频率定义像素编码，进而生成该图像的霍夫曼码表。

图像的文件格式：BMP 深度可选且不做压缩，GIF 采用了无损压缩，TIFF 适用于扫描仪和桌面出版，PCX 像素深度可选且使用 RLE 编码，PNG 是 gif 的替代品，JPEG 采用有损压缩，Target 彩色丰富供专业用户使用，WMF 保存的是函数调用信息(windows)，EPS 是 PostScript 图形打印机专用，DIF 是 AutoCAD 格式，CDR。

4、电视信号通过光栅扫描的方法显示到屏幕上，彩色电视采用相加混色，使用 RGB 作为三基色。

几种视频文件格式：GIF；Flic(.FLI/.FLC)一种彩色动画文件格式，使用 RLE 和 Delta 算法编码；AVI 支持 256 色和 RLE 压缩；QuickTime；MPEG 是运行图像压缩算法，方法是单位时间内采集并保存第一帧信息，然后只存储其余帧对第一帧发生变化的部分，进而实现压缩(平均压缩比 50:1)；RM 一种新型流式视频文件格式。

MPEG－4 是一套多媒体通信标准，主要由音频编码、视频编码、数据平面、描述符接口、缓冲共管理和实时识别等部分构成。
MPEG－1 应用于电视图像和伴音信息的通用编码；
MPEG－2 应用于高数据速率数字存储媒体的电视图像和伴音编码；
MPEG－7 是一套多媒体内容描述符接口标准。

5、虚拟现实技术的特征：多感知，沉浸感，交互，构想。

大概有 4 种虚拟现实技术：桌面虚拟现实，完全沉浸的高级虚拟现实，增强现实性的虚拟现实，分布式虚拟现实系统。

6、VCD 上的数据文件是以 MPEG－1 标准的格式存储的，它将图像分为了 3 种：帧内图像、预测图像和插补图像构成。其中帧内图像采用 JPEG 压缩方法来去掉冗余信息，也称作空间压缩。预测图像使用的帧间编码模式，也称作时间压缩。

对动态图像进行压缩处理的基本条件是：动态图像中帧与帧之间具有相关性。

动态图像有 3 个基本特点：具有时间连续特性，具有实时性，帧与帧之间具有相关性。相关性是指动态图像的连续前后两帧信息变化很小的特点。

人们把无损压缩称为熵编码。

CD 盘光道的结构与磁盘的磁道不同，它的光道不是同心圆，而是螺旋型光道，一张 CD 的光道大概有 5 公里。光盘的随机存储性变得较差。

DVD 盘是将光盘间距缩小，将记录信息的最小凹凸坑长度缩小，以提高存储容量。因此 DVD 刻录机和播放机需要采用波长更短的激光，同时为提高接收盘片反射光的能力，要增大光学物镜数值孔径。

MIDI 文件包含音符、定时和 16 个通道的演奏定义。

脉冲编码调制是最简单、最基本的一种波形编码。

如果两种色光混合而成白光，则这两种色光互为补色。　红色＋青色　＝　绿色＋品红　＝　蓝＋黄　＝　白色

使用 RGB3:3:2 表示一个像素时，像素深度为 8，即 3+3+2。如果使用 RGB8:8:8 表示一个像素，那么像素深度为 24。

图像分辨率，是指组成一幅图像的像素密度，即用每英寸多少点(dpi)表示数字化图像的大小，也用水平和垂直的像素表示。例如用 200dpi 来扫描一幅 2*2.5 英寸的照片，则可以得到 400*500 的图像。

视盘与 CD 盘的信息存储和读出原理有结构一样，但视盘是以模拟量的方式记录视频信号的。激光束读取视盘上信息的方向是从盘的内圈向外圈读的。

7. 数据库技术基础

1、数据库系统 DBS，是由数据库、硬件、软件和人员组成的。

数据库 DB

软件包括操作系统、数据库管理系统 DBMS 和应用程序。

数据库技术的发展经历了 3 个阶段：人工管理阶段、文件系统阶段、数据库系统阶段。

文件系统的最大特点是解决了应用程序和数据之间的一个公共接口问题，使得应用程序使用统一的存取方法来操作数据。

数据库系统与文件系统的区别是：数据的充分共享、交叉访问及应用程序的高度独立性。数据库对数据的存储是按照同一结构进行的，不同的应用程序可直接操作这些数据。

2、DBMS 的功能，主要实现对共享数据有效的组织、管理和存取。有以下几个方面的功能：

数据定义：DBMS 提供数据定义语言 DDL，用户可以对数据库的结构进行描述，包括外模式、模式和内模式的定义，数据库的完整性定义，安全保密定义等。这些存储在数据字典中，是 DBMS 运行的基本依据。

数据库操作：数据操纵语言 DML，实现对数据库中数据的基本操作，如检索、插、改、删。DML 双分为宿主型和自含型。

数据库运行管理：多用户环境下的并发控制、安全性检查和存取控制、完整性检查和执行、运行日志的组织管理、事务管理和自动恢复都是 DBMS 的重要组成。

数据组织、存储和管理：DBMS 分类组织、存储和管理各类数据，包括数据字典、用户数据、存取路径等，并要确定以何种文件结构和存取方式在存储级别上组织这些数据。DBMS 实现数据间的联系、数据组织和存储的基本目标是提高存储空间的利用率。

数据库的建立和维护

RDBS，关系数据库系统，实体与实体间的关系的集合构成一个 RDBS，也有型、值之分。关系数据库的型称为关系数据库模式，是对数据库的描述。关系数据库的值也称为关系数据库，是关系的集合。统称为 RDBS。

OODBS，是面向对象的数据库系统。支持以对象形式进行数据建模，且要符合 2 个条件，首先要是一个 DBMS，其次必须是面向对象的。

ORDBS，对象关系数据库，提供了元组、数组和集合等更丰富的数据类型以及处理新的数据类型的能力。

3、数据模型的基本概念

数据描述的三个领域：现实世界、信息世界、机器世界。

现实世界，指客观存在的各种报表、图表、原始数据；在信息世界中数据库常用的术语有属性、实体、实体集和码；机器世界是按机器的观点对建模，主要使用的术语有字段、记录、文件和记录码。信息世界与机器世界的几个术语可以一一对应。

数据模型的 3 要素：数据结构、数据操作、数据的约束条件。

常用的数据模型分为概念型数据模型和基本数据模型。概念数据模型也称为信息模型，是按用户的观点对数据和信息建模，是从现实世界到信息世界的第一层抽象，强调语义表达功能，用于数据库设计，这类模型中最著名的是 E－R 模型。基本数据模型是按计算机观点对数据进行建模，用于实现 DBMS。基本的数据模型有层次模型、网状模型和关系模型。目前随着新应用的发展，面向对象模型更广泛的被使用。

4、E－R 模型，它所采用的 3 个主要概念是：实体、联系、属性。E－R 模型只能说明实体间的语义联系。
E－R 方法：　矩形－－表示实体
菱形－－－－联系

椭圆－－－－属性
双椭圆－－－多值属性
虚椭圆－－－派生属性
线段－－－－－将属性与相关实体或实体与联系连接起来
双线－－－－－表示一个实体全部参与到联系集中
双线矩形－－弱实体

扩充的 E－R 模型：增加了弱实体、特殊化、概括以及聚集等概念。

弱实体，在现实世界中有一种联系代表实体间的 ownership 关系，即一个实体依赖于另一个实体而存在，这类实体被称为弱实体。

特殊化，设有实体集 E，如果 S 是 E 的某些真子集的集合，则称 S 为 E 的一个特殊化，E 是 Si 的超类，Si 称为 E 的子类。若两个子集 Si 与 Sj 没有交集，则 S 称为 E 的不相交特殊化，否则称为重叠特殊化。在扩充的 E－R 图中使用特殊化圆圈（而不是菱形）和连线的方式来表示超类－子类关系模型。超类到圆圈有连线，双线表全特殊化，单线表部分特殊化。子类用双竖边矩形表示。圆圈到子类的线用符号“U”标识为特殊化。圆圈内的"d"表示不相交特殊化，"O"表示重叠特殊化。

从 E－R 模型向关系模型转换时，所有“实体”和“联系”都要转换成相应的关系模式。

5、层次模型，采用树型结构表示数据与数据间的联系，每个节点表示一个实体。根节点之外的结点都有且仅有一个双亲。
特点：记录间的联系通过指针实现，简单、效率高。
缺点：只适于表示 1:n 的联系。

网状模型 DBTG，采用网络结构表示数据与数据的联系。允许一个以上结点无双亲，也允许一个结点有多个双亲。

网状模型与层次模型的区别：
1)网状模型中子女结点与双亲结点的联系不惟一，需要为每个结点命名，见图示；
2)网状模型允许复合链，两结点间可以存在两种以上的联系；
3)网状模型不能表示记录之间的多对多的联系，解决办法是引入联结记录来表示多对多的联系，见图示。

特点是：更直接地描述现实世界，存取效率高；缺点是：结构复杂。

关系模型，在关系模型中用表格结构表达实体集及实体集间的联系，其最大的特色是描述的一致性。关系模型就是若干个关系模式的集合。一个关系模式相当于一个记录型，对应于程序设计语言中的类型定义的概念。关系是一个实例，也是一张表，对应于程序设计语言中变量的概念。 区别： 与前两种模型的最大区别是使用主键而不是指针导航数据，表格简单直观。

在关系数据库中，若关系模式中的每个关系的属性值均是不可分解的，则该关系模式属于－－第一范式 1NF，这也是关系数据库的最基本要求。

关系代数运算是以关系作为运算对象的一组高级运算集合，关系定义为元素相同的元组的集合。因此，关系代数运算是以集合操作为基础的运算，其 5 种基本运算是并、差、笛卡尔积、投影和选择。

6、从数据库管理的角度看，数据库系统体系结构一般采用三级模式结构。外模式、概念模式、内模式。

数据库的行和值：行，是指对某一数据的结构和属性的说明；值，是行的一个具体赋值。

概念模式，也称模式，是数据库中全部数据的逻辑结构和特征的描述。它由若干个概念记录类型组成，只涉及行的描述，不涉及具体的值。概念模式不仅要描述概念记录类型，还要描述记录间的联系、操作以及数据的完整性和安全性。概念模式不涉及存储结构和访问技术等，因此做到了物理数据独立性。概念模式的数据定义语言称为“模式 DDL”。

外模式，也称用户模式或子模式，是用户与数据库系统的接口，是用户用到的那部分数据的描述。它由若干个外部记录类型组成，用户使用数据操纵语言 DML 对数据库进行操作。描述外模式的数据定义语言称为”外模式 DDL“。

内模式，也称为存储模式，是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式，定义所有的内部记录类型、索引和文件的组织方式，以及数据控制方面的细节。但是内部记录也不涉及物理记录，那是操作系统负责的相关机制。“内模式 DDL”。

总之，数据按外模式的描述提供给用户，按内模式的描述存储在磁盘上，而概念模式则提供了连接这两级模式的相对稳定的中间点，且使得两级模式的任一级的改变都不受另一级的牵制。

数据库系统在三级模式之间提供了两级映像，这保证了数据库中的数据具有较高的逻辑独立性和物理独立性。“模式/内模式”，“外模式/模式”。

数据的独立性包括数据的物理独立性和数据的逻辑独立性。物理逻辑性是指，当数据库的内模式发生改变时，数据的逻辑结构不变；逻辑独立性，是指用户的应用程序和数据库的逻辑结构相独立。

7、数据库系统的体系结构

1）集中式数据库系统：不但数据是集中的，数据的管理也是集中的，就是说从形式的用户接口到 DBMS 核心都集中在 DBMS 所在的计算机上。

2）C/S 数据库体系结构：安排某些任务在服务器上执行，另一些任务在客户机上执行。数据库系统功能分为前端和后端。前端主要包括图形用户界面、表格生成和报表处理等；后端负责存取结构、查询计算和优化、并发控制以及故障恢复。前端与后端通过 SQL 或应用程序来接口。

注：数据库服务器一般可分为事务服务器和数据服务器。其中事务服务器又被称为查询服务器，典型的事务服务器中有多个在共享内存中访问数据的进程，包括服务器进程、锁管理进程、写进程、监视进程和检查点进程。

3）并行数据库系统：并行体系结构的数据库系统是由多个物理上连在一起的 CPU 组成，分为共享内存式多处理器和无共享式并行体系结构。前者多个 CPU 共享一个内存与磁盘接口，后者每个 CPU 都有自己的内存和磁盘。

4）分布式数据库系统：分布式 DBMS 包括物理上分布、逻辑上集中的分布式结构和物理上逻辑上都分布的分布式数据结构 2 种。

5）WEB 数据库，又称为网络数据库，即网站上的后台数据库。

8、事务：是一个操作序列，这些操作要么都做，要么都不做。它是数据库环境中不可分割的逻辑工作单位。事务的 4 个特性 ACID，原子性、一致性、隔离性、永久性。

9、数据库的 4 类故障：事务内部故障、系统故障、介质故障、计算机病毒。

故障恢复的基本原理是“建立数据冗余”。建立冗余数据的方法是进行数据转储和登记日志文件。

数据的转储分为：静态转储和动态转储、海量转储和增量转储。

日志文件：DBMS 在事务处理的过程中，把事务开始、事务结束以及对数据库的插入、删除和修改的每一步操作写入日志文件。当发生故障后，DBMS 可以利用日志文件撤销事务对数据库的改变，回退到事务的初始状态。DBMS 可以利用日志文件来进行事务故障恢复和系统故障恢复，并可协助后备副本进行介质故障恢复。

在发生数据库故障后，把数据库恢复到故障发生前的状态的方法：定期对数据库作后备文件；在进行事务处理时，对数据更新的全部有关内容写入日志文件；存储系统正常运行时，按一定的时间间隔，设立检查点文件，把内存缓冲区内容还未写入到磁盘中去的有关状态记录到检查点文件中；当发生故障时，根据现场数据内容、日志文件的故障前映像和检查点文件来恢复系统的状态。

事务恢复的 3 个步骤：1）反向扫描文件日志，查找该事务的更新操作；
2）对事务的更新操作执行逆操作；
3）继续反向扫描日志文件，查找该事务的其他更新操作，并作同样的处理，直到事务的开始标志。

在 SQL 中定义事务的语句有 3 条：BEGIN　TRANSACTION；COMMIT；ROLLBACK。

10、并发控制

并发操作带来的问题是数据的不一致性，有 3 种：丢失更新、不可重复读、读脏数据。主要原因是事务的并发操作破坏了事务的隔离性。

并发控制的主要技术是封锁，有 2 种封锁类型：排他锁（又称为 X 锁或写锁），共享锁（又称为 S 锁或读锁）。

排他锁：特征是独占性；共享锁：特征是共享可读，在锁释放前该数据对象不可写。

三级封锁协议：一级封锁协议是指事务在修改数据前对其加 X 锁，直到事务结束才释放，这样就解决了丢失更新的问题；

二级封锁协议是指在一级封锁协议的基础上，事务 T 在读取数据 R 之前先对其加 S 锁，读完后立即释放 S 锁，这样就解决　　了读脏数据的问题；

三级封锁协议是指在一级封锁协议的基础上，事务 T 在读取数据 R 之前先对其加 S 锁，直到事务结束时释放 S 锁，三级封　　锁协议能够解决丢失更新、读脏数据、不可重复读这 3 个问题。

活锁与死锁

并发调度的可串行性：一个正确的多个事务并发执行，当且仅当其结果与某一次序串行地执行它们时的结果相同，则这种调度策略是可串行化的调度。可串行性是并发事务正确性的准则，一个给定的并发调度，当且仅当这旨可串行化的才认为是正确的调度。

两段封锁协议：是指事务必须分两个阶段对数据进行加锁和解锁。事务在第一个阶段只能获得封锁，在第二个阶段只能释放锁。

封锁的粒度：封锁对象的大小称为封锁的粒度。封锁的对象可以是逻辑单元也可以是物理单元。

事务的嵌套：事务是不能嵌套的，这样就违背了事务的原子性。相当于当且仅当当前没有事务在运行时，程序才能执行 BEGIN　TRANSACTION 操作。

11、数据库安全性：为保护数据库，我们要在多个层次上采取安全性措施。
数据库系统层次；操作系统层次；网络层次；物理层次；人员层次。

数据库的完整性：是指数据的正确性和相容性（有效性）。这包括用户定义完整性、参照完整性、实体完整性。实体完整性规则指主码的任何组成部分都不可以是空值，引用完整性规则则不允许引用不存在的实体。

授权：read, insert, update, delete ；index, resource, alteration, drop 。后者操作的对象是涉及数据库模式的表、表属性及索引。

权限授予图：表现授权从一个用户到另一个用户的传递。用户具有授权的条件是，当且仅当存在从授权图的根到代表该用户节点的路径。为安全着想，我们要求授权图中的所有边都必须是某条从 DBA 开始的路径的一部分。

角色：在数据库中建立一个角色集，将权限授予角色，通过为用户授予角色实现赋权的管理。

审计追踪: 它是一个记录对数据库所有更改的日志。可以在关系更新操作上定义适当的触发器来建立一个审计追踪。很多的数据库系统都提供了内置机制来建立审计追踪。

12、数据仓库 DW（Data Warehouse）：在数据库基础上产生的能满足决策分析需要的数据环境。

数据仓库的基本特征：数据是面向主题的，数据是集成的，数据是相对稳定的，数据是反映历史变化的。

数据仓库的数据模式：星型模式，雪花模式，事实星型模式。典型的数据仓库具有为数据分析而设计的模式，使用 OLAP 工具进行联机分析处理。其数据通常是多维的，包括维属性和度量属性。包含多维数据的表称为事实表。
一个事实表、多维表以及从事实表到多维表的参照外码的模式，称为星型模式;
更复杂的数据仓库含有多级维表，这种模式称为雪花模式;
复杂的数据仓库也可能含有不止一个事实表，这种模式称为事实星型模式。

数据仓库的体系结构：它通常采用三层结构，底层为数据仓库服务器、中间层是 OLAP 服务器、顶层为前端工具。

底层的数据仓库服务器一般是一个关系数据库系统。中间层的 OLAP 可以是关系型 OLAP（即扩充的 DBMS）也可以是多维的 OLAP 服务器。顶层的前端工具是各种查询、报表、分析、挖掘工具。

从结构的角度看，有 3 种数据仓库模型：企业仓库、数据集市、虚拟仓库。

数据集市是企业范围数据的一个子集；虚拟仓库是操作型数据库上视图的集合。

13、数据挖掘：从海量数据中挖掘信息的技术。支持 DM 的 3 种基础技术是海量数据搜索、强大的多处理器计算机、数据挖掘算法。几中常用的数据挖掘技术是：人工神经网络、决策树、遗传算法、近邻算法、规则推导。

数据挖掘与数据仓库的关系：数据仓库不仅是集成数据的一种方式，而且它的 OLAO 联机分析功能还为数据挖掘提供了一个很好的操作平台。

与传统数据分析工具比较，数据挖掘工具更注重于预测未来的情况。

数据挖掘技术的应用过程：确定挖掘对象，准备数据，建立模式，数据挖掘，结果分析，知识应用。

将数据转换成一个分析模型，建立一个真正适合挖掘算法的分析模型是数据挖掘的关键。

14、SQL 语言中不提供地使用索引的功能，这支持了物理数据的独立性。

8. 关系数据库

1、关系数据库的基本概念

属性与域

第一范式条件 1NF：在关系数据模型中，所有的域都应是原子数据。

笛卡尔积：设 D1,D2,...,Dn 为任意集合，则定义 D1,D2,...,Dn 的笛卡尔积为：

D1*D2*...*Dn = {(d1,d2,...,dn)|di 属于 Di, i=1,2,...,n}
其中每一个元素(d1,d2,...,dn)叫做一个 n 元组，元组的每一个值 di 叫做一个分量。笛卡尔积可用二维表来表示。

例如：D1={0,1}, D2={a,b}, D3={c,d}，求 D1*D2*D3。

D1*D2*D3 = {(0,a,c), (0,a,d), (0,b,c), (0,b,d), (1,a,c), (1,a,d), (1,b,c), (1,b,d)}

D1	D2	D3
0	a	c
0	a	d
0	b	c
0	b	d
1	a	c
1	a	d
1	b	c
1	b	d

笛卡尔积与关系：D1*D2*..*Dn 的子集叫做在域 D1,D2,..,Dn 上的关系，记为 R(D1,D2,..,Dn)，称关系 R 为 n 元关系。关系中属性的个数称为元数，元组的个数称为基数。

术语对应：　属性－－－字段
　　关系模式－记录类型
　　元组－－－记录

关系的一些名词：
　　目或度－－常用 R 表示关系的名字，用 n 表示关系的目或度
　　候选码－－能够做主码的属性或属性组都是候选码
　　主码－－－候选码中的一个
　　主属性－－包含在个选码中的属性
　　外码－－－对于关系 R 来讲，外码就是指它的某个属性或属性组，不是该关系的码，而是其它关系的码
　　全码－－－一个关系的全属性称为这个关系的全码

3 种基本的关系类型：基本关系（即基本表或基表），查询表，视图表。

2、关系模式：关系的描述称之为关系模式，表示为 R(U,D,dom,F)。通常简记为 R(U)或 R(A1,A2,..,An)，Ai 为属性名或域名，一般在主码属性下加下划线以标识。
　　其中，R 表示关系名，U 表示属性名集合，D 是属性的域，dom 是属性向域的映像的集合，F 是属性间数据的依赖关系的集合。

3、关系的完整性分为 3 类：

　　实体完整性，　　关系的主属性不能为空值；
　　参照完整性，　　设 F 是关系 R 的外码，与关系 S 的主码相对应，则 F 或均取空值，或等于 S 中某个元组的主码值。
　　用户定义完整性，反映某一具体应用涉及的数据必须满足的语义要求。

4、关系运算：关系运算的特点是操作对象和操作结果都是集合。关系数据语言分为 3 类，关系代数语言、关系演算语言、和具有前二者特点的语言（如 SQL）。关系演算语言又分为元组关系演算语言和域关系演算语言。

关系代数运算符有 4 类：集合运算符（并、差、交、笛卡尔积），专门的关系运算符（选择、投影、连接、除），算术比较符（大于、小于等），逻辑运算符（与、或、非）。

并

差

广义笛卡尔积，两个元数分别是 n 目和 m 目的关系 R 和 S 的广义笛卡尔积是一个(n+m)列的元组的集合。元组的前 n 列是关系 R 的一个元组，后 m 列是关系 S 的一个元组，记作 R*S。

投影，从关系 R 中选择出若干属性列 A 组成新的关系，记作 PaiA(R)。

选择，是从关系的水平方向进行运算，是从关系 R 中选择满足给定条件的诸元组，记作　　。

扩展的关系运算符：

交，关系 R 与 S 具有相同的关系模式，关系 R 与 S 的交是由属于 R 同时属于 S 的元组构成的集合，记作　　。

连接，是从两个关系 R 和 S 的笛卡尔积中选取满足条件的元组，可以认为笛卡尔积是无条件的连接。连接又可以分为 3 种：　　连接、等值连接、自然连接。
　　连接：
　　可以由基本的关系运算符笛卡尔积和选择导出：
　　等值连接：

自然连接：是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且在结果集中将重复属性列去掉。自然连接不仅要从关系的水平方向，而且还要从关系的垂直方向运算。

除，同时从关系的水平方向和垂直方向进行运算。给定关系 R（X,Y）和 S（Y,Z），X,Y,Z 都是属性组。R 除 S 应当满足元组在 X 上的分量值 x 的象集 Yx 包含关系 S 在属性组 Y 上投影的集合。

广义投影，就是带有条件的投影运算，记为 。

外连接， 连接的扩展，为了处理缺失的信息。分为 3 种左外、右外、全连接。以左外连接为例，取出左侧关系中所有与右侧关系中任一元组都不匹配的元组，用空值填充所有来自右侧关系的属性，构成新的元组，将其加入自然连接的结果中。

聚集函数，要注意一些聚集函数的表达方式。分组的表达方法。

5、元组演算

元组演算是非过程化查询语言。它只描述所需信息，而不给出获得该信息的具体过程。元组表达式中的变量是以元组为单位的，其一般形式为{t|P(t)}。t 是元组变量，P(t)是元组关系演算公式，公式是由原子公式组成的。

原子公式，即原子命题函数。有 3 种形式。一种是 R(t)。

全称量词和存在量词。

A＝＞B，表示若 A 为真则 B 为真。

6、域演算，在域演算中表达式中的变量是表示域的变量，可将关系的属性名视为域变量。一般表示为{t1,...,tk|P(t1,...,tk)}，其中 t1,..,tk 是域变量。

原子公式，有 3 种形式。一种是 R(t1,..,ti,...,tk)。

7、函数依赖

依赖的闭包，属性的闭包

8、优化查询

9、规范化

10、模式分解

无损连接，保持函数依赖

9. SQL 语言

1、SQL 是集数据定义和数据操纵为一体的数据库语言。

数据定义子语言 DDL，用来定义数据库模式。DDL 包括数据库模式定义，数据库存储结构和存取方法定义，以及数据库模式的修改删除功能。

数据定义子语言的处理程序也分为了数据库模式定义处理程序，和数据库存储结构和存取方法处理程序。前者接收用 DDL 表示的数据模式定义，翻译成内部表示形式，存储到数据字典中；后者接收数据库存储结构和存取方法定义，在存储设备上创建相关的数据库文件，建立物理数据库。

数据操纵子语言，通常有这样几种操作：查询、插入、删除、修改。后三种应该可以都纳入更新的范畴。

嵌入式 SQL，宿主语言

2、SQL 是一种通用的、功能强大的关系数据库语言，它的主要功能包括数据查询、数据操纵、数据定义和数据控制。

SQL 的特点有：
综合统一
高度非过程化
面向集合的操作方式
两种使用方式， 一是在终端上键入 SQL 命令直接操作数据库，另一种是将 SQL 嵌入到高级语言中去。
简洁、易用，完成核心功能只用了 9 个动词，包括了 4 类：数据查询（SELECT）、数据定义（CREATE、DROP、ALTER）、数据操纵（INSERT、UPDATE、DELETE）、数据控制（GRANT、REVOKE）。

SQL 支持关系数据库的三级模式结构，其中视图对应外模式，基本表对应模式，存储文件对应内模式。

SQL 的基本组成：DDL、DML、事务控制、嵌入式 SQL 和动态 SQL、完整性、权限管理。

3、数据库定义

（一）创建表： CREATE TABLE ＜表名＞（＜列名＞＜数据类型＞[列级完整性约束]
[，＜列名＞＜数据类型＞[列级完整性约束]]...
[，＜表级完整性约束条件＞]）；

其中列级完整性约束条件有：NULL 和 UNIQUE。

例 9.1 建立一个供应商和零件数据库。其中供应商表 S（Sno,Sname,Status,City）的属性分别表示供应商代码、姓名、状态、所在城市。“零件”表 P（Pno,Pname,Color,Weight,City）的属性分别表示零件号、零件名、颜色、重量及产地。其中数据库要满足这样的要求：

1）供应商代码不能为空，且值是惟一的，供应商名也是惟一的；
2）零件号不能为空，且值是惟一的。零件名不能为空；
3）一个供应商可以供应多个零件，而一个零件可以由多个供应商供应。

解：供应商与零件之间需要建立一个关系模式，二者之间是一个多对多的关系，新生成的关系模式的码应该是供应商的码和零件表的码，以及二者联系的属性构成。如 SP（Sno,Pno,Qty） Qty 表示数量。

```
CREATE TABLE S(Sno    CHAR(5) NOT NULL UNIQUE,
      Sname    CHAR(30) UNIQUE,
      Status    CHAR(8),
      City    CHAR(20)
      PRIMARY KEY (Sno));


CREATE TABLE P(Pno    CHAR(6) NOT NULL UNIQUE,
      Pname    CHAR(30) NOT NULL,
      Color    CHAR(8),
      Weight    NUMERIC(6,2),
      City    CHAR(20)
      PRIMARY KEY(Pno));

CREATE TABLE SP(Sno    CHAR(5),
      Pno    CHAR(6),
      Qty    NUMERIC(9)
      PRIMARY KEY(Sno,Pno),
      FOREIGN KEY(Sno) REFERENCES S(Sno),
      FOREIGN KEY(Pno) REFERENCES P(Pno));
```

（二）修改表和删除表

```
ALTER TABLE ＜表名＞[ADD＜新列名＞＜数据类型＞[完整型约束条件]]
      [DROP ＜完整性约束名＞]
      [MODIFY ＜列名＞＜数据类型＞]

DROP TABLE ＜表名＞
```

（三）定义和删除索引

数据库中的索引就是某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的指针清单。作用如下：

通过创建惟一的索引，可以保证数据记录的惟一性；
加快数据检索速度；
加速表与表之间的连接，由其在实现数据的参照完整性方面有特别意义；
在使用 ORDER BY ，GROUP BY 语句时可以明显地减少计算时间；
使用索引可以在检索数据的过程中使用优化隐藏器，提高系统性能。

分为聚集索引和非聚集索引

聚集索引，对表的物理数据页中的数据按列进行排序，然后再重新存储到磁盘上，叶子节点中存储的是实际数据；
非聚集索引，具有完全独立于数据行的结构，不必对物理数据页中的数据按列排序，叶子节点存储的是组成非聚集索引的关键字和行定位器。

建立索引：CREATE [UNIQUE][CLUSTER] INDEX ＜索引名＞

ON ＜表名＞（＜列名＞[＜次序＞][，＜列名＞[＜次序＞]]...）；

其中次序可选 ASC，DSC，默认为 ASC
UNIQUE 表明此索引的每一个索引值只对应惟一的数据记录。
CLUSTER 表示要建立的索引是聚簇索引，即索引项的顺序是与表中记录的物理顺序一致的索引组织。

几个例子： CREATE UNIQUE INDEX S-SNO ON S(Sno);
CREATE UNIQUE INDEX P-PNO ON P(PNO);
CREATE UNIQUE INDEX SPJ-NO ON SPJ(SNO ASC,PNO DESC, JNO ASC)。

删除索引： DROP INDEX ＜索引名＞

（四）定义、删除、更新视图

视图的创建：CREATE VIEW 视图名 （列表名）
AS SELECT 查询子句
[WITH CHECK OPTION]；

其中查询子句可以是任意复杂的 select 语句，但一般不能出现 order by , distinct。
WITH CHECK OPTION，表示在对视图进行更新、插入或删除操作时，要保证满足子查询中的条件表达式。

例如 CREATE VIEW CS－STUDENT
AS SELECT SNO，SNAME，SAGE，SEX
FROM STUDENDS
WHERE SD＝‘CS’
WITH CHECK OPTION；

其中使用了 with check option，所以在对视图插删操作时，要保证 SD＝‘CS’的条件成立。

视图的删除：DROP VIEW 视图名

4、数据操作，SELECT，INSERT，DELETE，UPDATE

（一）SELECT 基本结构
SELECT [ALL|DISTINCT] <目标列表达式> [,<目标列表达式>]...
FROM <表名或视图名>[,<表名或视图名>]
[WHERE <条件表达式>]
[GROUP BY <列名 1> [HAVING <条件表达式>]]
[ORDER BY <列名 2> [ASC|DESC]...]

其中，子句顺序是：SELECT、FROM、WHERE、GROUP BY、HAVING 和 ORDER BY。HAVING 只能和 GROUP BY 搭配使用。

SELECT 对应关系代数运算中的投影运算；FROM 对应笛卡尔积；WHERE 对应选择。

SELECT 查询中没有全程量词，也没有逻辑蕴涵，但可以通过谓词转换来实现。？

（二）简单查询
（三）连接查询
检索至少选修了课程号为 C1 和 C3 的学生号：SELECT Sno FROM SC SCX, SC SCY WHERE SCX.Sno = SCY.Sno AND SCX.Cno = 'C1' AND SCY.Cno = 'C3';

（四）子查询与聚集函数， 子查询也叫嵌套查询。

例：检索选修课程名为 MS 的学生号和学生姓名。

```
SELECT Sno, Sname
    FROM Students
    WHERE Sno IN
        (SELECT Sno FROM SC
            WHERE Cno IN
                (SELECT Cno FROM C WHERE Cname = 'MS'));
```

聚集函数：AVG, MIN, MAX, SUN, COUNT

使用谓词 ANY 和 ALL 必须同时使用比较运算符，其含义与等价的转换关系如下：
＞ANY －－－ ＞MIN
＞ALL －－－ ＞MAX
＜ANY －－－ ＜MAX
＝ANY －－－ IN

＜＞ALL－－－　NOT　IN

几个例子：查询其他系比计算机系 CS 所有学生年龄都要小的学生姓名及年龄。

```
SELECT Sname, Sage
    FROM Students
    WHERE Sage < ALL
        (SELECT Sage
            FROM Students
            WHERE SD = 'CS')
        AND SD<>'CS';
```

用＜MIN 代替上面的＜ALL：

```
SELECT Sname, Sage
    FROM Students
    WHERE Sage <
        (SELECT MIN(Sage)
            FROM Students
            WHERE SD = 'CS')
        AND SD<>'CS';
```

（五）分组查询

GROUP　BY　子句

HAVING　子句，如果在元组被分组之前需要按某种方式加以限制，使不需要的分组为空，可以在 GROUP　BY 子句后面加一个 HAVING 子句。

注意：空值在任何聚集操作中都会被忽视，COUNT（＊）是计算某个关系中所有元组数目之种，但 COUNT（A）是计算 A 属性中非空的元组个数之和。

例：针对供应商数据库中的 S、P、J。SPJ 关系，查询哪一个工程至少用了 3 家供应商（包含 3 家）供应的零件的平均数量，并按工程号降序排列。

```
SELECT　JNO，AVG（QTY）
    FROM　SPJ
    GROUP　BY　JNO
    HAVING　COUNT（DISTINCT（SNO））＞2
    ORDER　BY　JNO　DESC;
```

（六）别名运算

（七）字符串操作，　使用操作符 LIKE 的模式匹配。%匹配任意字符串，_ 可以匹配任意一个字符。

在 Like 中可以使用转义字符，将特殊字符当作普通字符处理，如反斜杠“\"。

（八）集合操作，　保留字 UNION，INTERSECT，EXCEPT 分别对应并、交、差。保留字用于两个查询时，其两侧应用括号括起来。

例：学生和教师的关系模式如下，查询既是女研究生又是教师且工资大于 1500 元的名字和地址。

```
(SELECT Name, Address
    FROM Students
    WHERE SEX='女' AND Type='研究生')
    INTERSECT
(SELECT Name, Address
    FROM Teachers
    WHERE Salary >=1500)
```

查询不是教师的学生：

```
(SELECT Name, Address FROM Students)
    EXCEPT
    (SELECT Name, Address FROM Teachers)
```

（九）视图的查询和删除

视图的查询：当查询视图表时，通常先将其转换成等价的对基本表的查询，然后执行查询语句。即系统先从数据字典中取出该视图的定义，然后与视图中的查询语句结合起来，形成一个修正的查询语句。

视图更新要遵守的规则：

从多个基本通过连接操作导出的视图不允许更新；
对使用了分组、集函数操作的视图不允许更新；
若视图是从单个基本表通过投影、选取操作导出的，则允许进行更新操作。

WITH 子句，将一个复杂的查询分解成一小视图？？

（十）插入、删除和修改语句

插入： INSERT INTO 表名（字段名[，字段名]...)
VALUES(常量[，常量]...);

删除： DELETE FROM 表名
WHERE 条件表达式；

修改： UPDATE 表名
SET 列名＝值表达式
[WHERE 条件表达式]

5、SQL 中的授权

数据库中的完整性是指数据库的正确性和相容性。

（一）主键约束 PRIMARY KEY

完整性约束条件：完整性约束条件作用的对象有关系、元组、列 3 种，每种又分为静态、动态两类。

完整性控制：有 3 方面的功能，定义功能、检测功能、处理功能。这样来保证实现对数据的完整性控制。检查是否违背完整性约束的时机 有两个：立即执行约束和延迟执行约束。前者在一条语句执行完后立即检查，后者在整个事务执行完成后进行。

实体完整性（PRIMARY KEY 子句），关系中只能有一个主键，声明主键的方法有两个，就是 primary key 放的位置不同。

如， CREATE TABLE Students
(Sno CHAR(8),
Sname CHAR(10),
Sex CHAR(1),
Sdept CHAR(20),
Sage NUMBER(3),
PRIMARY KEY(Sno));

或 CREATE TABLE Students
(Sno CHAR(8) PRIMARY KEY,
Sname CHAR(10),
Sex CHAR(1),
Sdept CHAR(20),
Sage NUMBER(3));

（二）外键约束 FOREIGN KEY（参照完整性）

格式： FOREIGN KEY（属性名）REFERENCES 表名（属性名）
[ON DELETE[CASCADE|SET NULL]]

ON DELETE CASCADE 指明删除参照关系的元组时，同时删除参照关系中的元组。

（三）属性值上的约束 NULL 和 CHECK

如果要求某属性为空，在定义时在数据类型的后面加上 NOT NULL。
如， CREATE TABLE Students
(Sno CHAR(8),
Sname CHAR(10) NOT NULL,
Sex CHAR(1),
Sdept CHAR(20),
Sage NUMBER(3),
PRIMARY KEY(Sno));

在 Students 表中，要求男生的年龄在 15－25 之间，女生的年龄在 15－24 之间。
如， CREATE TABLE Students
(Sno CHAR(8),
Sname CHAR(10) NOT NULL,
Sex CHAR(1),
Sdept CHAR(20),

```
    Sage    NUMBER(3),
    PRIMARY KEY(Sno))
    CHECK(Sage >=15 AND ((SEX='M' AND Sage<=25) OR
          (SEX='F' AND Sage<24)));
```

（四）全局约束　CREATE　ASSERTIONS

全局约束是指一些较复杂的完整性约束，会涉及到多个属性间的联系或多个关系间的联系。分为两种：基于元组的检查子句和断言。

1）使用 CHECK 子句对单个关系的元组值加以约束，可以在关系的定义中的任何地方加上 CHECK 及约束条件；

2）断言：　CREATE　ASSERTION　＜断言名＞　CHECK（＜条件＞）

例如，在教学数据库模式 Students,SC,C 中加一个约束，不允许男同学选修“张勇”教师的课。

```
CREATE　ASSERTION　ASSE－SC1　CHECK
    （NOT　EXISTS
      （SELECT　＊　FROM　SC　WHERE　Cno IN
        (SELECT Cno FROM C WHERE TEACHER='张勇')
        AND Sno IN
        (SELECT Sno FROM Students WHERE SEX='M')));
```

又如，在 Students,SC,C 中有一个约束，每门课最多允许 50 名男同学选修。

```
CREATE　ASSERTION　ASSE－SC2　CHECK
    （50＞＝ALL（SELECT　COUNT（SC.Sno)
          FROM Students,SC
          WHERE Students.Sno=SC.Sno AND SEX='M'
          GROUP BY Cno));
```

（五）授权与销权，DBMS 数据控制应具有这样的功能，通过 GRANT 和 REVOKE 将授权通知系统并存入数据字典；当用户提出请求时，检查其授权情况。

授权语句格式：

```
GRANT　＜权限＞[，＜权限＞]...
[ON＜对象类型＞＜对象名＞]
TO　＜用户＞[，＜用户＞]...
[WITH GRANT OPTION];
```

PUBLIC 与 WITH GRANT OPTION：PUBLIC 参数可以将权限授给所有用户；后者使获得授权的用户还可以将此权限授给其它用户。

例如，将对供应商 S、零件 P 及项目 J 的所有操作权限授给用户 User1 及 User2。

```
GRANT ALL PRIVILEGES ON TABLE S, P, J TO User1,User2;
```

将 S 的插入权限授组 User1，并允许将此权限授给其他用户。

```
GRANT INSERT ON TABLE S TO User1 WITH GRANT OPTION;
```

DBA 把数据库 SPJ 中建立表的权限授给用户 User1。

```
GRANT CREATETAB ON DATABASE SPJ TO User1;
```

收回授权语句格式：

```
REVOKE <权限>[,<权限>]...
    [ON <对象类型><对象名>]
    FROM <用户>[,<用户>]...;
```

例如，　REVOKE ALL PRIVILEGES ON TABLE S , P, J FROM User1,User2;

```
REVOKE INSERT ON TABLE S FROM User1 WITH GRANT OPTION;

REVOKE SELECT ON TABLES S FROM PUBLIC;

REVOKE UPDATE(Sno) ON TABLE S FROM User1;
```

－－将权限的控制定位在某一个属性上

6、触发器，触发器是一种特殊类型的存储过程，它是通过事件触发而执行的。主要特点是，当被声明的事件发生时触发器被激活；触发器激活后不会立即执行，而是先测试触发条件；如果触发条件满足，则由 DBMs 执行与该触发器相连的动作。

创建触发器，不同数据库使用的触发器语法不同。

例：假定银行数据库关系模式为：　Account(Account-no, branch-name,balance)

Loan(Loan-no, branch-name, amount)
depositor(customer-name, Account-no)

假定银行在处理透支时，不是将账户余额设成负值，而是将账户余额设置为零，并且建立一笔贷款，其金额为透支金额。这笔贷款的贷款号应该等该透支帐户的账户号。采用 SQL－99 标准创建触发器如下：

```
CREATE TRIGGER overdraft_trigger after update on Account
Refferencing new row as nrow
For each row
When nrow.balance<0
Begain atomic
    Insert into borrower
    (SELECT customer-name, Account-no
        FROM depositor
        Where nrow.account-no=depositor.account-no);
    Insert into values
        (nrow.account-no, nrow.branch-name, nrow.balance);
    update account set balance=0
        Where account.account-no=nrow.account-no
End
```

When nrow.balance<0 是触发条件；

Begain atomic ... End 子句用来将多行 SQL 语句集成为一个复合语句。其中前两条 Insert into 语句表示在 borrower 和 loan 关系中建立新的贷款业务，update 语句用来将账户余额清零。

Referencing old row as 子句建立一个变量，用来存储已经被更新或删除的行的旧值。Referencing new row as 可以被 update 和 Insert 语句使用，可以存放经过更新的新值。

Referencing old table as 或 Referencing new table as 子句可以用来指向临时表，使之容纳所有被影响的行。临时表不能使用 before 触发器，但可以用 after 触发器。

触发器在事件之前被激发，可以避免非法更新。

例 9.45：仓库管理数据库中有如下关系，
inventory(item, level)，表示仓库中某种商品的现有量。
minlevel(item, level)，表示仓库中存有某种商品的量小量。
reorder(item, amount)，表示某种商品小于最小量时要订购的数量。
orders(item, amount)，表示定购某种商品的量。

```
CREATE TRIGGER reorder_trigger after update of amount on inventory     --我怀疑 amount 应该是 level
Referencing old row as orow, new row as nrow
For each row
When nrow.level <= (SELECT level
            FROM minlevel
            Where minlevel.item = orow.item)
And orow.level > (SELECT level
            FROM minlevel
            Where minlevel.item = orow.item)
Begin
    Insert into orders
        (SELECT item, amount
            FROM reorder
            Where reorder.item = orow.item)
End
```

删除触发器：　DROP TRIGGER {trigger}[,...,n]

10. 系统开发与运行

1、软件工程知识

软件工程是指应用计算机科学、数学及管理科学等原理，以工程化的原则和方法来解决软件问题的工程。其目的是提高软件生产率，提高软件质量，降低软件成本。

在经历 60 年代的软件开发危机后，人们开展了软件开发模型、开发方法、工具与环境的研究，提出了瀑布模型、深化模型、螺旋模型和喷泉模型等开发模型，出现了面向数据流方法、面向数据结构的方法和面向对象方法等开发方法，以及一批计算机辅助的软件工程工具和环境。

2、软件生存周期：可以分为 6 个阶段。计划制定、需求分析、设计、编码、测试、运行维护。

计划制定：　确定待开发软件系统的总目标，对其进行可行性分析，并对资源分配，合理安排进度计划。

参加人员有，用户、项目负责人和系统分析员。

该阶段产生的文档有，可行性分析报告和项目计划书。

需求分析：　确定系统功能、性能、数据及界面等要求，从而确定系统的逻辑模型。

参加人员有，用户、项目负责人和系统分析员。

产生的文档有，需求规格说明书。

软件设计：　分为概要设计和详细设计。

概要设计参加人员为，系统分析员和高级程序员；详细设计参加人员有，高级程序员和程序员。

该阶段产生的文档有，设计规格说明书（可以分为概要设计说明书和详细设计说明书）。

编码：　产生的文档为源程序清单。

测试：　文档为测试计划和测试报告。

运行及维护

3、软件开发项目管理基础知识

成本管理：有两种方法。　开发费用　＝　人月数　＊　每个人月的代价
　　开发费用　＝　源代码行数　＊　每行平均费用

风险分析：涉及 3 个概念，一是关心未来，第二是关系变化，第三是要解决选择问题。风险分析实际包括 4 个活动：风险识别、风险预测、风险评估和风险控制。

进度管理：有两种安排方式，一是交付日期已确定，另一个是仅确定了大致的日期，最终交付日期由开发部门确定。常用两种图形描述方法。

Gantt 甘特图，横轴表示时间，纵轴表示任务，水平线表示任务的进度安排。它可以很好的描述任务间的并行性，但不能反映任务间的依　　赖关系，不能确定整个项目的关键；
PERT 图，　是一个有向图，图中的箭头表示任务，图中的结点称为事件，表示流入结点的任务的结点和流出结点的任务的开始。仅当流入结点的任务都结束时，该事件才出现，流出结点的任务才能开始。每个任务有一个松驰时间。为了表示任务间的关系，图中还可以加入一些空任务（虚线表示）。一个事件有事件号、出现该事件的最早时刻、最迟时刻。松驰事件为 0 的任务构成了关键路径。PERT 图不能反映任务的并行性。

人员管理：主程序员组、无主程序员组、层次式程序员组。

4、软件开发方法：主要掌握 3 种方法，分别是结构化方法、面向对象方法和原型法。

结构化方法：是目前最成熟的开发方法之一，分为结构化分析和结构化设计。

面向对象方法：从现实世界中客观存在的事物出发来构造软件系统。软件系统适用的业务范围称作软件的问题领域，把问题领域中事物的特征抽象地描述成类，由类建立的对象作为系统的基本构成单位，它们的内部属性与服务描述了客观存在的事物的静态和动态特征。对象类之间的继承关系、聚集关系、消息和关联反映了问题域中事物之间实际存在的各种关系。

原型法：在获得一组基本需求后，快速地加以实现，随着用户和开发人员对系统理解的加深而不断进行补充和细化，是一种动态定义技术。

5、软件开发环境：是指支持软件产品开发的软件系统，它由软件工具集和环境集成机制构成。环境集成机制为工具集成和软件开发、维护及管理提供统一的支持，通常包括数据集成、控制集成和界面集成。有几个特征，环境的服务是集成的；环境的服务可用于各种软件开发活动；环境应支持小组工作方式。

6、ISO/IEC9126 软件质量模型

由 3 个层次组成，分别是：质量特性－－质量子特性－－量度指标。

质量特性（质量子特性）：
　　功能性（适合性、准确性、互用性、依从性、安全性）
　　可靠性（成熟性、容错性、易恢复性）

易使用性（易理解性、易学性、易操作性）
效率　（时间特性、资源特性）
可维护性（易分析性、易改变性、稳定性、易测试性）
可移值性（适应性、易安装性、一致性、易替换性）

MC　Call 软件质量模型，从软件产品的运行、修正和转移 3 个方面确定了 11 个质量特性。
产品运行（正确性、可靠性、易使用性、效率、完整性）
产品修正（可维护性、灵活性、可测试性）
产品转移（可移值性、复用性、互用性）

7、软件质量保证：是指为提高软件质量而进行的有计划、有组织的活动。

软件质量保证包括的 7 个主要活动相关的任务：应用技术方法、进行正式的技术评审、软件测试、标准的实施、控制变量、量度、记录保存和报告。

8、软件过程能力评估

软件产品的质量取决于软件开发过程。

软件过程评估，是软件改进和软件能力评价的前提。

软件过程评估的意义：是软件过程改进的需要。软件过程不断改进是软件工程的基本原理之一；软件过程改进是软件生存周期的基本过程之一。
是降低软件风险的需要。

软件能力成熟度模型 CMM：是对软件组织进化阶段的描述。分为 5 个成熟度级别，初始级－可重复级－已定义级－已管理级－优化级。比较有名的一个基于 CMM 模型的产品是成熟度调查表，可以用于一个机构软件过程实力、弱点和风险。

8、系统分析的目的和任务：对现行系统做进一步的详细调查，将调查所得到的文档资料集中，对组织内部整体管理善和信息处理过程进行分析，为系统开发提供所需资料，并提交系统方案说明书。

系统分析的主要步骤是：由现实系统得出物理模型－－抽象出逻辑模型－－优化出新的逻辑模型－－逻辑模型具体化，得到新系统的物理模型。最终编写系统方案说明书。

9、结构化分析方法：采用“自顶向下、逐层分解”的开发策略。

数据流程图 DFD：在逻辑上描述系统的功能、输入、输出和数据存储。DFD 的基本成分有，数据流、加工、数据存储、外部实体。它们各有特定的图形表示。

分层数据流图的画法：　1）画系统的输入和输出；　　－－称为顶层图
2）画系统的内部，将顶层图的加工分解成若干个加工，并用数据流连接；　－－称为 0 层图

确定加工的方法：在数据流的组成或值发生变化的地方应画一个加工；也可根据系统功能确定加工。
确定数据流方法：用户把若干个数据看作一个单位来处理时，可把这些数据看成是一个数据流。
3)画加工的内部；
4)重复第 3 步的分解过程至所有的加工都足够简单。

对图和加工进行编号：　顶图不必编号；
0 层图只有一张，图中的加工号可以是 0.1, 0.2, ...或是 1, 2, ..
子图号就是父图中被分解的加工号
子图中的加工号同样由图号、圆点和序号组成

注意分析教材 513 页的实例：某考务处理系统有如下功能：对考生送来的报名单进行检查；
对合格的报名单进行检查；
对阅卷站送来的成绩清单进行检查，并根据考试中心指定的合格标准审定合格者；
制作考生通知单送给考生；
生成各种报表。

画图中应注意的问题：适当地为数据流、加工、数据存储及外部实体命名；
画数据流而不是控制流；
一个加工的输出数据流不应与输入数据流同名；
允许一个加工有多条数据流流向另一个加工，也允许一个加工有两个相同的输出数据流流向两个不同的加工；
保持父图与子图平衡；
在自顶而下的分析过程中，如果一个数据存储首次出现时只与一个加工有关，那么可以把它作为内部文件而不必画出
保持数据守恒，即一个加工的输出数据流中的数据必须直接从该加工的输入数据流中获得或产生；
每个加工都必须有输入、输出数据流；

在整套数据流图中，每个数据存储都必须有读的数据流和写的数据流。

数据词典 DD：就是为数据流图中的每个数据流、文件、加工，以及更细节的数据项做出说明。其中对加工的说明称为“小说明”或“加工逻辑说明”。常用的加工逻辑说明方法有 3 种：结构化语言、判定表和判定树。结构化语言分为内外两层，外层有严格的语法，而内层语法灵活，接近于自然语言。

10、统一建模语言 UML：它提供了 9 种基本元素的图形，分别是：类图、对象图、用例图、序列图、协作图、状态图（活动图、构件图、部署图）。

UML 由 3 个要素构成：UML 的基本构造块、支配这些构造块如何放置在一起的规则、运用于整个语言的一些公共机制。

在 UML 提供的图中，可以采用类图，对逻辑数据库模式建模；状态图，用于接口、类和协作的行为建模，并强调对象行为的事件顺序；活动图，用于系统的功能建模，并具强调对象间的控制流。

11、系统分析报告：数据流图、数据字典和加工说明应该成为系统分析报告的主体。并且一份完整的系统分析报告应该包括如下内容。

组织情况概述
现行系统概述
系统逻辑模型
新系统在各个业务处理环节拟采用的管理方法、算法或模型
与新系统相配套的管理制度和运行体制的建立
系统设计和实施的初步计划
用户领导审批意见

12、系统设计的目的和任务：主要目的是为系统制定蓝图，在各种技术和实施方法中权衡利弊，精心设计，合理使用种种资源，最终形成系统的详细设计方案。

系统设计的任务分为两个步骤：首选是把总任务分解为许多基本的、具体的任务。合理地组织这些具体任务可以构成总任务，称为总体结构设计，也称为概要结构设计；　　其次是为各个具体任务选择适当的技术手段和处理方法，即详细设计。

系统总体结构设计原则：分解－协调原则
　　自顶而下原则
　　信息隐蔽、抽象原则
　　一致性原则
　　明确性原则
　　模块间耦合尽可能小，模块内组合尽可能紧凑
　　模块的扇入系数和扇出系数要合理
　　模块的规模要适当

模块化设计：　模块是组成系统的基本单位，应该具备 4 个元素，分别是，输入和输出、处理功能、内部数据、程序代码。

模块结构图，　是采用 HIPO 图（分层输入－处理－输出）形式绘制而成的框图。它主要关心模块的外部属性，即上下级模块、同级模块之间的数据传递和调用关系。它主要由 5 种基本符号表示：模块、调用、数据、控制和转接。

存储设计：首先要解决数据的整体结构设计，然后要确定数据资源分布和安全保密属性。

13、系统详细设计

代码设计

输出设计　－－确定输出内容、选择输出设备与介质、确定输出格式

输入设计　－－输入原则：最小量、简单性、早检验、少转换

处理过程设计　－－总体结构设计将系统分解成许多模块，并决定每个模块的外部特征，即功能和界面。计算机处理过程的设计则是要确定每个模块的内部特征，包括局部的数据组织、控制流、每一步的具体加工要求及实施细节等。

处理过程的关键是，用一种合适的表达方法来描述每个模块的执行过程。常用的描述方式有图形、语言和表格 3 类。例如，程序流图、盒图 NS、形式语言、决策树、决策表。盒图就是用一个盒子表示一个步骤，可以嵌套，只能从上头进入下头输出，因此限制了控制转移，保证了程序的良好结构。

用户界面设计　－－包括菜单方式、会话方式、操作提示以及操作权限管理方式等。权限管理一般是通过入网口令和建网时定义该节点级别来实现的。

安全控制设计　——包括数据处理和环境两方面

系统设计说明书

一份完整的系统设计说明书应包括：

1）引言

背景——摘要——工作条件/限制——参考和引用资料——专门术语定义

2）系统总体设计方案

模块设计——代码设计——输入设计——输出设计——数据库设计说明——模型库及方法库设计——网络设计——安全保密设计——实施方案说明书。

从系统调查、系统分析到系统设计是信息系统开发的主要工作，它们的工作量应占到总开发量的 70%。

14、系统实施的任务：　按总体设计方案购置和安装计算机网络系统；

软件准备——其中编写程序是一个重要任务；

人力培训；

数据准备；

试运行。

程序设计：主要依据是系统设计阶段的 HIPO 图、数据库结构和编码设计。

结构化程序设计方法：适用于某些过程不规范、模块划分不细或有特殊业务处理需要模块程序量较大时。主要强调 3 点规则：模块内部程序各部分要按自顶向下的结构划分；各程序部分应按功能组合；各程序间联系尽量使用调用子程序实现。

快速原型式方法：首先将 HIPO 图中具有普遍性的功能模块集中实现，构造系统原型，再对一些特定的功能和模块进行补充。

面向对象的程序设计方法：OOD?

15、软件测试方法：分为人工测试和机器测试。

人工测试，又称为代码审查。

机器测试，分为黑盒测试、白盒测试。

黑盒测试——也称为功能测试，主要测试软件的外部特性。

白盒测试——也称为结构测试，根据程序内部结构、逻辑，以程序的路径和过程进行测试。

软件测试步骤：可以分为 4 步，如下

1）单元测试，即模块测试

2）组装测试，即集成测试。又分为非增量式集成和增量式集成。前者可以对模块进行并行测试，后者使测试更彻底。

3）确认测试，进一步检查软件的功能和性能是否与用户要求的一致。以系统方案说明书为基础，检查软件有效性。

确认测试首先要进行有效性测试以及软件配置审查，然后进行验收测试和安装测试，最后经各部门认可后交付使用。

4）系统测试，将已经确认的软件、硬件、外设及网络结合起来，进行系统的各种组装测试和确定测试。

调试：试探法、回溯法、对分查找法、归纳法、演绎法。

16、系统转换

实际运行，是对系统最好的检验和测试方法。这个阶段的工作有：

对系统进行初始化、输入各原始数据记录；

记录系统运行数据和状况；

核对新、老系统的输出结果；

考察输入方式（方便、效率、误操作）

测试响应速度

系统转换方式：直接转换、并行转换、分段转换。

17、系统维护，系统的可维护性可以定义性的定义为维护人员理解、改正、改动和改进这个软件的难易程序。

系统可维护性的评价指标：可理解性、可测试性、可修改性。

文档，是软件可维护性的决定因素。

系统维护主要包括硬件设备的维护、应用软件的维护和数据的维护。

18、系统评价，分为广义和狭义两种。广义的评价是指从系统开发的开始到结束的每一阶段都需要进行评价。狭义的评价是指在系统建成并投入运行之后进行的全面和综合的评价。

从总体上可以将广义评价分为立项评价、中期评价、结项评价。

19、系统运行管理

运行管理制度：包括种类机房安全运行管理制度，和信息系统的其他管理制度。

日常运行管理内容：包括系统运行情况记录、审讯追踪、审查应急措施落实、系统资源管理、软件及文档管理

11. 数据库设计

1、数据库设计概述：数据库设计属于系统设计的范畴。参照软件工程对生命周期的定义，也把数据库设计分为 6 个步骤：数据库规划、需求收集分析、数据库设计与应用程序设计、实现、测试、运行与维护。

数据库设计：数据库的设计是对用户数据的组织和存储设计；应用程序的设计是在数据库设计的基础上对数据操作及业务实现的设计，包括事务设计和用户界面设计。

实现：依照设计使用 DBMS 支持的 DDL 语言实现数据库的定义，用高级程序语言编写应用程序。

2、系统需求分析，是用户和设计人员对数据库应用系统所涉及的内容和功能的理解和描述。

用户对系统的需求包括：数据需求、围绕这些数据的业务处理需求、数据安全性需求、数据完整性需求。

需求分析阶段是以调查和分析为主要手段的，以此获得用户对系统的信息要求和处理要求。

需求分析阶段要完成的主要工作是建立数据字典和数据流图。

需求分析的方法和步骤：使用数据字典描述用户的信息要求，使用数据流图描述业务处理过程。

数据字典包括，数据项、数据结构、数据流、数据存储、处理过程。

数据项：是数据的最小单位，一般包括项名、含义和说明、别名、类型、长度、取值范围及该项与其它项的逻辑关系。如采购单号；

数据结构：是若干有意义的数据项的集合，包括数据结构名、含义和组成成分。如采购单；

数据流：既可以是数据项也可以是数据结构，它表示某一次处理的输入输出数据，包括数据流名、说明、数据来源和去向及需要的数据项和数据结构。如采购计划数据流；

数据存储：加工中需要存储的数据，包括数据存储名、说明、输入数据流、输出数据流、组成成分、数据量、存取方式以及存取频度等。如原材料的价目表，在计算成本和支付采购费用的处理过程中要用到这些数据；

处理过程：是加工处理过程的定义和说明，包括处理名称、输入数据、输出数据、数据存储及响应时间等，如采购支付处理。

－－－－－－－－－－－－－－－－－－－－－－－－－－－－
处理过程名：　采购支付
说明：　根据采购单、原材料价目表，计算出应付原材料采购费用
输入数据：　采购单
数据存储：　原材料价目表
输出数据：　支付费用表
－－－－－－－－－－－－－－－－－－－－－－－－－－－－

3、概念结构设计，是在需求分析的基础上，对用户信息加以分类、聚集和概括，建立信息模型，并依照选定的数据库

管理系统软件把它们转换为数据的逻辑结构，再依照软硬件环境，最终实现数据的合理存储。这一过程被称为“数据建模”。

数据建模的过程，可以分为 3 个阶段：概念结构设计、逻辑结构设计、物理结构设计。

概念结构设计的策略有 4 种：自顶而下、自底而上、逐步扩张、混合策略。

概念结构设计最常用的方法是 1976 年由一位华人学者提出的 E－R 方法。将现实世界的信息结构统一由实体、属性及实体之间的联系来描述。

使用 E－R 方法时，需要对现实事物抽象并以 E－R 图的形式描述出来，有 3 种抽象的方法：

分类，　将现实世界中具有共同特征和行为的事物定义为一种类型。个体与类型关系是 is member of

聚集，　定义某一类型所具有的属性。各个属性是所属类型的一个成分，is part of

概括，　由一种已知类型定义新的类型。已知类称为超类，新定义类称为子类，关系为 is subset of

4、用 E－R 方法建立概念模型

步骤：选择局部应用；逐一设计分 E－R 图；E－R 图合并。

注意属性与实体的区别：属性不可再分；属性不能与其它实体发生联系。

分 E－R 图的合并方法就是将具有相同实体的两个或多个 E－R 图合而为一。合并过程中可能会发生的冲突有：属性冲突、命名冲突、结构冲突。

对合并后的 E－R 进行优化的方法有 3 个：

1）实体类型的合并，　凡具有 1：1 或 1：n 联系的实体都可以合并，减少实体个数；

2）冗余属性的消除；

3）冗余联系的消除，　合并后的 E－R 图中可能会出现实体联系的环状结构，消除直接联系，保留间接联系。

5、逻辑结构设计，是在概念结构设计基础上进行的数据模型设计，可以是层次、网状和关系模型。逻辑结构设计的主要任务是：

确定数据模型；

将 E－R 图转换为指定的数据模型；

确定完整性约束；

确定用户视图。

6、E－R 图向关系模式的转换：

1）实体向关系模式的转换
将 E－R 图中的实体逐一转换为一个关系模式，其中实体名对应关系模式的名称，实体的属性转换成关系的属性，实体标识符就是关系的码。

2）联系向关系模式的转换

一对一联系的转换：有 2 种方式。

一种方式，是将联系转换成一个独立的关系模式，关系模式的名称取联系的名称，关系的属性包括该联系所关联的两个实体的码和联系的属性，关系的码可以取自任一方实体的码；

另一种方式，是将联系归并到关联的两个实体的任一方，在一方实体属性集中增加另一方实体的码和该联系的属性，归并后的实体码保持不变。

一对多联系的转换：有 2 种方式。

第一种方式，是将联系转换成一个独立的关系，关系的名称取联系的名称，关系的属性包括该联系所关联的两个实体的码和联系的属性，关系的码是多方实体的码；

第二种方式，是将联系归并到关联的两个实体的多方，在待归并的多方实体属性集中增加一方实体的码和该联系

的属性，归并后的多方实体的码保持不变。

多对多联系的转换：只有 1 种方式。

那就是将该联系转换成一个独立的关系，关系的名称取联系的名称，关系的属性包括该联系所关联的两个多方实体的码及该联系的属性，关系的码是两个多方实体的码构成的属性组。

7、关系模式的规范化

由 E－R 图转换得来的初始关系模式可能会有数据冗余或更新异常，需要进一步得进行规范化处理：

1）根据语义确定各关系的数据依赖；

2）根据数据依赖确定关系的范式；

3）对不合要求的范式进行分解，达到 3NF、BCNF 或 4NF；

4）对关系进行评价和修正。因为最规范的关系不一定是最合适的关系。

关系的完整性约束有：主码约束、检查约束、参照性约束

8、数据库的物理设计

物理设计一般应做这些工作：

确定数据分布；

确定存储结构；

确定存取方式。

存储结构是指数据文件中记录之间的物理结构，可以是顺序存储、哈希存储、堆存储或 B＋树存储等。要根据数据的处理要求和变更频度，选定合理的物理结构。

为提高数据的访问速度，会采用索引技术。同样也要根据数据处理和修改要求，选择恰当的索引字段和类型。

数据的存取方式，是由其存储结构决定了的。

9、数据系统的实现，是根据设计，由开发人员编写代码程序来完成的，包括数据库的操作程序和应用程序。

数据库的操作程序使用 SQL 语言实现，主要有：DDL、DML、事务处理程序、存储过程、触发器。

嵌入式 SQL 因为其复杂性，已逐渐被 ODBC、ADO 接口技术取代。

10、数据库系统的实施方法：

建立实际的数据库结构（DDL）

装载测试数据试运行

装载数据，即卸载实验数据，加载用户数据，正式运行。

11、数据库的保护，是通过数据库的恢复、安全性控制、完整性控制、并发控制，来实现的。

事务，是数据库处理的基本逻辑单位，事物的原子性、一致性、隔离性和持久性（简称 ACID）保证了数据更新的正确性。面向数据更新的应用程序的编写，必须以事务为单位进行数据的操作。

数据库的备份与恢复：

数据备份与日志备份是数据库恢复技术的主要依据。数据备份又称为数据转储，分为静态和动态两种方式。日志备份用来记录对数据库系统的更新操作，写日志的次序严格按照并发事务执行的时间次序，必须先写日志后写数据库。

数据库系统中的故障类型：事务故障、系统故障、介质故障。

恢复策略：有 2 种操作，分别是撤销事务（UNDO）和重做事务。

事务故障的恢复：可以 UNDO 产生故障的事务，回到该事务执行前的正确状态；

系统故障的恢复：系统故障会导致数据库不一致，恢复方法是先 UNDO 未完成的事务，再 REDO 已提交的事务；

介质故障的恢复：需要 DBA 参与，重装数据库、装入数据库的备份和日志文件的副本，再由系统完成 UNDO、REDO 操作。

数据库的安全性，是保证数据库不被非法用户访问和破坏的机制。包括：权限机制（GRANT）、视图机制、数据加密。数据加密可以防止数据在存储和传输过程中失密。

数据库的完整性，是保证数据库不被合法用户的错误操作而破坏。完整性是指数据的正确性和相容性。

数据库的并发控制：

1）并发操作，可能会带来数据的不一致性有 3 种，丢失修改、不可重复读和读脏数据。

2）加锁：控制的手段就是加锁。排他锁（写锁 X）和共享锁（读锁 S）。X 锁将独占数据，数据上有 S 锁时事务只能加 S 锁读而不能加 X 锁写。

3）封锁协议：

一级封锁协议，事务 T 在修改数据 A 前必须先对 A 加 X 锁，直到事务结束才能释放 X 锁。这样解决了丢失修改的问题；

二级封锁协议，在一级封锁协议基础上，事务 T 在读数据 A 前必须对其加上 S 锁，读完即释放 S 锁。这样使得一个事务不能读取其他事物修改中的数据，解决了读脏数据问题；

三级封锁协议，在一级封锁协议基础上，事务 T 在读数据 A 前必须对其加上 S 锁，直到事务 T 结束才释放 S 锁。这样使得一个事务在读取数据期间，其他事务只能读取该数据而不能修改，所以解决了不可重复读的问题；

两段锁协议，对任何数据进行读写前都必须加锁，在释放一个封锁后，事务不再申请和获得任何其它封锁。这样可以缩短持锁时间，提高并发性，同时解决了数据的不一致性。

12. 数据库运行与管理

1、数据库系统的运行策略：　从物理环境上保障系统的稳定运行；

从对人员的要求上做保障；

应用数据库的安全性策略；

做好数据库的备份与恢复工作。

2、数据库系统的监控对象和监控方式

监控对象有 3 个，性能监控、故障监控、安全监控。

监控方式有 2 种，系统监控和应用程序监控。

3、数据库维护

因为某些原因需要修改数据库的结构，称为数据重构，包括表结构的修改和视图的修改。

视图机制一方面可以实现数据的逻辑独立性，另一方面可以实现数据的安全性。

文档是对系统结构和实现的描述，必须与系统保持调度的一致性。数据库重构过程中的所有修改必须在文档中体现出来。

4、数据库系统的运行统计

系统监控和运行统计是 DBA 掌握数据库系统运行状态最有效的手段。系统监控用来保障系统的稳定运行，系统统计则用来了解系统性能，作为性能调整的依据。

5、数据库系统的审计，是一种 DBMS 工具，它记录数据库资源和权限的使用情况。审计是被动的。

6、数据库系统的管理

（1）数据字典的管理：数据字典是存储在数据库中的所有对象信息的知识库，其中存储的数据称为元数据。数据字典是只读的。

（2）数据完整性维护和管理：作用对象有列、行、表 3 种。列级约束、主码约束和参照完整性约束是在数据库定义过程中定义的，存在数据字典中。更为复杂的约束可以编写触发器程序实现。

因此，由 DBMS 管理的约束，可通过修改数据库定义完成维护和管理；

由应用程序实现的复杂的完整性约束，要通过分析修改程序（触发器程序）来实现。

（3）数据库的存储管理：数据库中的数据是以文件形式存储在物理存储设备上的，程序通过 DBMS 完成 I/O 操作来访问数据。提高系统访问效率的有效手段就是提高 I/O 操作的效率。使用这样几种手段管理数据的存储，可以有效地提高性能：

1）索引文件和数据文件分开存储，事务日志文件存储在高速设备上；

2）适时修改数据文件和索引文件的页面大小；

3）定期对数据进行排序；

4）增加必要的索引项。

也可以增加计算机内存，引入调整存储设备等外部方式提高系统的访问效率。

（4）数据库备份与恢复的管理：

设定合理的备份周期和备份时间；

把事务日志文件保存在最稳定的存储设备上；

定期在事务日志文件中加入检查点（checkpoint），检查点记录数据库的正确状态点。在数据库恢复过程中，可以反向扫描日志文件找到第一个检查点，执行 UNDO、REDO 操作。

（5）数据库的并发控制与死锁管理：多用户数据库 DBMS 都提供并发控制机制。在实际运行过程中，死锁的产生多是因为事务程序的错误引起。管理员需要使用系统监控工具和系统日志，找出频繁产生死锁的事务。分析原因，修改事务程序，减少死锁。

（6）数据库的安全管理：

建立网络安全（防火墙）

操作系统级安全（登录用户管理）

DBMS 级安全（访问 DB 的用户验证密码）

角色和用户授权管理

使用视图和存储过程

使用审计功能

7、数据库系统的性能调整

1）SQL 语句的编码检验：通过 DBMS 提供的监控和统计功能，找出频繁执行的 SQL 语句并对其进行优化。步骤为，

（1）尽可能地减少多表查询或建立物化视图；

（2）以不相关子查询代替相关子查询；

（3）只检索需要的列；

（4）用带 IN 的条件子句等价替换 OR 子句；

（5）经常提交 COMMIT，以尽早释放锁。

2）表设计的评价：首先要求关系都能符合 3NF 或 BCNF，然后还要根据实际运行情况对表进行调整。

调整的原则是：　如果频繁地访问涉及的是对两个相关表进行连接操作，则将这两个表合并；

如果频繁地访问只是在表中的一部分字段上进行，则考虑分解表或将该部分单独拿出作为一个表；

对于很少更新的表，引入物化视图。

3）索引的改进：索引的调整原则如下，

如果查询是瓶颈－－在关系上新建适当的索引，通常在作为查询条件的属性上建立索引可以提高查询效率；

如果更新是瓶颈－－因为每次更新都会重建表上的索引，引起效率的降低，可以考虑删除某些索引；

选择适当的索引类型－－比如经常使用范围查询，可以使用 B 树索引，比散列索引更高效；

将有利于大多数据查询和更新的索引设为聚簇索引。

4）设备的增强：高速的计算机、增加内存、高速网络设备、高速存储设备。

13. 网络与数据库

1、分布式数据库的目标：　一是，借助网络把分散在不同地域的数据管理起来；

另一是，各个地域的数据库系统仍能支持本地应用。

分布式数据的定义：　分布性；逻辑相关性；场地透明性；场地自治性。全部满足这些条件的数据库系统称为完全分布式数据库系统。

分布式数据库的特点：

1）数据的集中控制性；

2）数据独立性；

3）数据冗余可控性；

4）场地自治性；

5）存取的有效性－－分布式数据库系统中的查询优化有两个级别。一个是全局优化，决定在多个副本中选取合适的场地副本，使得场地间的数据传输量及次数最少，从而减少系统通信开销；另一个是局部优化，这与传统的集中式数据库中的优化是相同的。

上面的这些特点中，前 3 点都是数据库系统优于文件系统的方面，并且除第 4 点以外，传统数据库也应该具有这些特点。

2、分布式数据库的体系结构

（A）分布式数据库系统的模式结构：目前国际上还没有统一的标准。国内提出的 4 层模式结构如下：全局外层、全局概念层、局部概念层、局部内层。

1）全局外层　－－分布式数据库的全局视图是针对分布式数据库特定的全局用户，是对分布式数据库的最高层的抽象。

2）全局概念层－－是分布式数据库的整体抽象，但比集中式的概念层有更多的描述。全局概念层有 3 种模式描述信息，全局概念模式、分片模式、分配模式。

（1）全局概念模式：描述分布式数据库全局数据的逻辑结构；
（2）分片模式：描述全局数据逻辑划分的视图，每一个逻辑划分就是一个分片；
（3）分配模式：是分片后的物理分配视图。

因此分布式数据库的定义语言除了需要提供概念模式的定义语句外，还要提供分片模式和分配模式的定义语句。全局概念模式到分片模式的映射是一对多的；分片模式到分配模式的映射是一对多的或一对一的，这要由数据分布的冗余策略决定。作为 GDBA，将负责全局数据结构的定义、逻辑分布的定义和物理分布的定义。

3）局部概念层－－该层由局部概念模式描述，全局概念模式经逻辑划分后被分配在各局部场地上。

4）局部内层　－－相当于集中式数据库的内层。

4 层结构的全局数据库和局部数据库分离、数据独立性、透明性、数据冗余控制都体现了分布式数据库的特点。

（B）数据分布：数据的划分和放置是数据分布问题的两个重要方面。有几种处理策略，集中式、分割式、复制式

和混合式。

（C）数据分片：也称数据分割。对于关系数据库，数据分片有 3 种方法，水平分片（元组）、垂直分片（属性）、混合分片（水平和垂直）。水平划分元组为若干不相交的子集，可以通过合并操作恢复全局关系。垂直划分关系的属性为若干子集，要求所有属性都要被划分且每一垂直片都包含该全局关键字，可以通过连接操作恢复该全局关系。

数据分片要遵守的原则为：完备性条件；可重构条件；不相交条件（关键字除外）。

（D）分布透明性：也称为分布独立性，由高到低分成了 3 个级别，分片透明性－－分配透明性－－局部数据模型透明性。

局部数据模型透明性，也称为局部映像透明性，是透明性的最底层，在 4 层模式中处理分配模式和局部概念模式之间。全局数据模型与每个节点上局部数据库的数据模型的转换是由分配模式与局部概念模式之间的映像实现的。当某个节点上的数据库的数据模型改变时，只要改变分配模式到该节点局部概念模式之间的映像即可，应用程序不受影响，从而实现了局部数据模型透明性。

（E）分布式数据库管理系统 DDBMS：有两大类，综合型和联合型。前者是新建的一个分布式数据库，后者是整合已经存在的多个节点的数据而形成。联合型又可以分为同构型和异构型。

分布式数据库管理系统由 4 部分组成：LDBMS、GDBMS、GDD（全局数据字典）、CM（通信管理）。

一个完全的分布式管理系统要符合 12 条规则：场地自治性；非集中式管理；高可用性；位置独立性；数据分割独立性；数据复制独立性；分布式查询；分布式事务管理；硬件独立性；操作系统独立性；网络独立性；数据库管理系统独立性。

注意理解分布式数据库系统结构模式图，和分布式数据库管理系统结构图。

3、分布式查询处理和查询优化

与集中式数据库环境中的查询相比，分布式数据库环境中的查询还要考虑到：数据、信息传输的延迟问题；网络中存在多处理器时的并行数据处理机会。这两点都会影响到查询的速度。

查询优化器，在分布式数据库系统中它的任务是控制和加快查询的执行与数据传输的过程。在分布式查询处理技术中，查询优化的基本类型通常包括 2 类：针对查询执行代价的优化，和针对查询响应时间的优化。

查询执行代价优化的目标是，使查询执行所使用的系统资源尽量少（最便宜）；

查询响应时间优化的目标是，尽量减少响应时间而不计较系统资源的消耗（最快）。

4、分布事务管理

1）分布式事务：在分布式数据库系统中，一个分布式事务是由若干个不同站点上的子事务组成的。

2）分布式事务与集中式事务的相同特性：与集中式事务的特性相同为 ACID，原子性、一致性、隔离性和持久性。

分布式事务与集中式事务的不同特性：执行特性、操作特性和控制报文。执行过程中，分布式事务要必须创建一个控制进程，协调子事务、数据及控制报文；而集中式事务仅由并行调度算法进行调度即可。操作过程中，分布式事务还要加入大量的通信原语和控制原语。集中式事务没有使用控制报文。

3）分布式数据库故障：分为节点故障（事务故障、系统故障、介质故障），通信故障（报文故障、网络分割故障）。其中报文故障包括报文错、报文丢失、报文延迟。

4）分布式数据库的恢复原则：保证事务原子性的措施称为事务故障恢复，有几个原则是，

孤立和逐步退出事务的原则 UNDO；

成功结束事务原则 REDO；

夭折事务的原则。（撤消全部事务，恢复到初态）

在分布式事务恢复中，本地事务的恢复和集中式事务的恢复相同，由本地事务管理器 LTM 执行。整个的分布式事务的恢复由 LTM 与 DTM（分布式事务管理器）协同完成。

5）两阶段提交协议 2PC（准备提交、建议提交/撤销、全局提交/撤销、确认）

在 2PC 中，将分布式事务的某一个代理指定为协调者，其它代理都是参与者。参与者可以进行单方面撤销。2PC 可以分为两个步骤：先是表决阶段，然后是执行阶段。表决中实行一票否决。

2PC 对故障的恢复：（1）场地故障　参与者在写入“建议提交”前发生故障　：　——协调者等到超时后将取消事务，该参与者自行可以　　终止事务。

参与者在写入“建议提交”后发生故障：　——而其它参与者可以正常结束事务，该参与者　　要访问协调者或其它参与者获得之前协调者作出　　的决定并执行相应的操作。

协调者在写入“准备提交”后，在写入“全局提交/撤销”前发生故障：

——协调者从头恢复提交协议

协调者在写入“全局提交/撤销”后，在写入“事务结束”记录前发生故障：

——协调者恢复时要给所有的参与者重发之前的全局决定。

（2）报文丢失　丢失参与者的回答报文（建议提交/撤销）：　——协调者将取消整个事务

丢失“准备提交”报文：　——协调者在超时后将取消整个事务

丢失“全局提交/撤销”报文：　——涉案参与者将请求协调者重发该报文

丢失“确认”报文：　——协调者将重发全局报文，参与者无论子事务 提交与否都要给予确认。

（3）网络分割故障，整个网络被分为 2 组，协调者组和参与者组。各自进行故障处理。

6）三阶段提交协议 3PC（在 2PC 基础上增加了，全局预提交和准备就绪，两个报文，可以确认所有参与者的状态）

第一阶段，协调者向所有的参与者发“准备提交”报文，只有所有参与者都投票“建议提交”，才会进入第二阶段；

第二阶段，协调者向所有的参与者发“全局预提交”报文，只有所有参与者都投票“准备就绪”，才能进入第三阶段；

第三阶段，协调者向所有的参与者发“全局提交”报文。

3PC 仅降低了阻塞发生的可能性，不是完全的非阻塞协议。

3PC 对故障的恢复：　协调者发出的“准备提交”报文延迟，参与者超时而撤销子事务；

协调者等待参与者投票时超时，协调者将撤销事务；

参与者处于“赞成提交”状态，而等待全局预提交时超时，参与者将进入恢复处理过程；

参与者处于“准备就绪”状态，而等待全局提交时超时，参与者将进入恢复处理过程。

在 3PC 协议中，恢复处理过程惟一可以做的是就近访问一个参与者，依照协调者之前作出的决定安排自己的操作。

5、WEB 数据库

WEB 数据库由数据处理和资源共享这两种技术结合而成，也称为网络数据库。

WWW 下的 WEB 数据库，RDB 增加了 DB 的面向对象成分、增加各种中间件（CGI、ISAPI、ODBC、JDBC、ASP），通过应用服务器解释执行各种 HTML 中嵌入的脚本，来解决 Internet 应用中数据库的显示、维护、输出及到 HTML 的格式转换。

WEB 服务器的连接方案，有 2 套：服务器端方案（CGI、SAPI、ASP、PHP、JSP）和客户端方案（JDBC、DHTML）。

连接数据库的常用方法有：ODBC、DAO、RDO、ADO。ADO 是 DAO、RDO 的简化合集。

6、应用开发平台 ASP、PHP、JSP

ASP，ASP 服务器被含在了 IIS 服务器中，ASP 中使用的是 VBscript，ASP 安装配置简单、易学易用。ASP 缺点是使用了组件带来了安全性问题，另外 ASP 不能跨平台使用。

PHP，简单易学，可以跨平台，具有良好的数据库交换能力，与 Apache 紧密结合，安全性好。缺点是安装配置复杂，并且缺少企业级的支持。

JSP，可移值性好，可以跨平台，伸缩性强大，多样化并有强大的工具支持。缺点是安装配置管理都较复杂，只适用于大型系统。

7、关于 CGI（Common Gateway Interface）

是最早出现的动态网页发布技术。通过 CGI 接口，服务器可以向 CGI 程序发送信息，CGI 程序也可以向服务器发送信息。可以使用任何可形成可执行程序的程序语言编写 CGI 程序，如 C Shell、Perl、C、C＋＋、FROTRAN 和数据库语言。CGI 支持 ODBC 方式。

8、关于 ASP（Active Server Page）

ASP 提供了一个在服务器端执行脚本指令的环境（包括 VBscript,html,JavaScript 等），脚本可以嵌入 HTML 中，还可以通过 ActiveX 控件实现更加强大的功能。

ASP 通过 ADO 对象模块来存取数据库，只要数据具有对应的 ODBC 或 OLE DB 驱动程序。ASP 提供的 ADO 对象模块俾昼作夜了下列 6 个对象和 3 个集合。 Connection 对象
Recordset 对象
Fields 集合
Field 对象
Command 对象－－返回值为 Recordset 对象
Parameter 集合
Parameter 对象
Error 集合
Error 对象

9、关于 Servlet 和 JSP

Servlet 是一个运行在 WEB 服务器中的 Java 程序。它从浏览器获取一个 HTTP 请求，动态生成内容，并把 HTTP 信息返回给浏览器。Servlet 优点是运行在服务器端且可以更直接地访问数据库。

JSP 技术通常与 Java Servlet 技术结合，可以在 HTML 或其它标记语言中嵌入 Java 代码段并且调用外部 Java 组件。JSP 是一个前端的处理工具，可以使用 JavaBean 实现更为复杂的业务逻辑和动态功能。

JSP 最大的优点是将网页的静态内容（HTML 代码）和动态内容（Java 代码）分隔处理，易于人员分工。

10、XML 的应用

采用文件存储 XML，会受到文件系统的一些限制（大小、迸发性、工具选择、版本、安全性、综合性）。

XML 与数据库的数据转换：

文档的逆反回归不一致性，将数据存储到数据库时会丢失大量文档相关的信息。同样地，从数据库中检索数据并生成的 XML 文档，除非预定义的实体一，不包含任何字符数据和实体引用，并且同层元素和属性的出现顺序常常就是从数据库中返回数据的顺序。

XML 与数据库之间传输数据，需要进行相互映射。有两种映射方式：模板驱动、模型驱动。

第十四章 数据库发展趋势与新技术

1、面向对象数据库系统应该具有以下特征：

表达和管理对象的能力；

具有任意复杂度的对象结构；

具有与面向对象编程语言交互的接口；

具有表达和管理数据库变化的能力。

2、面向对象数据模型

面向对象数据模型可以看作在一个更高层次上实现数据模型的新成员，并经常用作高层概念模型，尤其在软件工程领域中更是如此。

面向对象数据模型的基本概念有对象、类、继承、对象标识和对象嵌套。

对象结构：属性集合、方法集合、消息集合

对象类 ：在面向对象数据库中，类是一系列相似对象的集合，对应于 E－R 模型中的实体集概念。类是面向对象系统和数据库系统之间最重要的连接。首先，类直接说明了一个实例及其所属类之间的实例关系；其次，类提供了构成查询的基础；另外，类可以用来增加面向对象数据库的语义完整性；最后，类提出了所有对象的属性和方法的规格说明，便于生成对象。

类的概念相似于关系，类的属性相拟于关系的属性，对象相似于关系中的一个元组。

继承与多重继承：在面向对象的数据模型中，所有类形成了一个有限的层次结构或者是一个有根的无环有向图，称之

为层次。

对象标识：每个对象惟一的、由系统生成的对象标识 OID。几种常用的标识为，值、名称、内置名。对象标识要求必须具有永久持久性。根据实际应用的需求，大多面向对象数据库系统允许有对象和值两种标识表示方法共同使用。

对象嵌套：在面向对象数据模型中，对象的一个属性可以是一个单一值，也可以是一个来自值域的值集，即一个对象的属性可以是一个对象，形成了嵌套关系。

3、面向对象数据库语言：包括对象定义语言和对象操纵语言。对象查询语言是对象操纵语言的一个重要子集。

面向对象数据库语言应该具有这些功能：类的定义和操纵；

操作/方法的定义；

对象的操纵。

4、对象关系数据库系统

在今天的商业领域中，占统治地位的数据库管理系统产品有 2 个：关系数据库系统和面向对象数据库系统。另外还有两种主要的类型是层次数据库和网关数据库。

对象关系数据模型扩展关系数据模型的方式是提供一个包括复杂数据类型和面向对象的更丰富的类型系统。

（1）嵌套关系：元组在一个属性上取值可以是一个关系，于是关系可以存储在关系中，从而形成了关系的嵌套。这样，一个复杂的对象可以用嵌套关系的单个元组来表示。

（2）复杂类型：集合是集合体类型的一个实例，另外的集合体类型包括数组和多重集合。面向对象关系数据库允许属性是集合。

（3）继承、引用类型：SQL99 仅支持单继承

（4）与复杂类型有关的查询：　路径表达式；（使用右向箭头符号，例如 select head-->name,head-->address from departments）

以集合体为值的属性；

嵌套与解除嵌套。（将一个嵌套关系转换为 1NF 的过程称为解嵌套，反向过程就是嵌套。嵌套可以通过 SQL 的分组扩展实现，即不使用聚集函数而只是返回多重集合）

（5）函数与过程：对象关系数据库系统中允许用户定义函数与过程，既可以用 SQL 来定义，也可以用外部的程序设计语言来定义。

（6）面向对象与对象关系：几种数据库系统的能力如下，

关系系统　－－简单数据模型、功能强大的查询语言以及高保护性；

对象关系系统－－复杂数据模型、功能强大的查询语言以及高保护性；

以持久化程序设计语言为基础的面向对象系统－－复杂数据类型、与程序设计语言集成以及高性能。

5、ERP 与数据库

ERP 的形成经历了四个阶段：基本 MRP 阶段、闭环 MRP 阶段、MRP－2 阶段以及 ERP 阶段。

基本 MRP，在传统的库存理论中引入了时间分段和反映产品结构的物料清单，从而较好的解决了库存管理和生产控制中的难题。

闭环 MRP，在基本 MRP 的基础上，加入了对能力的约束的考虑，如供货能力或运输能力，从而形成了信息回路，称为闭环 MRP。

MRP－2，以生产计划为主线，统一计划和控制各种资源，使企业的物流、信息流、资金流都畅通，也实现了动态反馈。

ERP，它继承了 MRP－2 的基本思想，但是把管理重心转移到财务上来，在全过程中贯穿财务成本控制的概念。

6、ERP 设计的总体思路

一个中心，两类业务，三条干线。即以“财务”为中心，处理“计划”和“执行”两类业务，围绕“供应链管理、生产管理、财务管理”这 3 条干线。

在 ERP 系统的设计过程中怎么样应用数据库技术：

1）先要绘制出业务处理流程图；

2）绘制出数据流程图；

3）绘制 E－R 图；　　－－前 3 步都是在前一步的基础上发展深入，一步步为数据库设计打好基础。

4）绘制功能模块图。

7、决定支持系统

决策支持系统实质上是在管理信息系统的基础上发展起来的。它综合利用了各种数据、信息、知识，特别是模型技术，能够辅助各级决策者解决决策问题。决策支持系统的新特点就是增加了模型库和模型库管理系统。

8、数据仓库

数据仓库已成为建立决策支持系统的重要技术手段，是建立决策支持系统的基础。

数据仓库的数据有 4 个特征：面向主题的、集成的、不可更新的和随时间不断变化的。

数据仓库的数据模型与传统的数据库相比有一些区别：它不包含纯操作型的数据；扩充了码的结构，增加了时间属性作为码的一部分；增加了一些导出数据。

数据仓库的设计过程与传统数据库相似，分为三级数据模型，即概念模型、逻辑模型、物理模型。概念模型使用 E－R 图描述，长方形表示主题而不再是实体，其它雷同。逻辑模型就是关系模型。

在进行数据仓库设计时，需要把数据分割和粒度划分结合起来考虑。粒度划分就是确定数据仓库中数据单元的详细程序级别，数据分割类似于数据分片概念。

元数据是数据仓库设计的一个重要组成部分。

9、数据转移技术

数据转移技术也称为数据转换或数据变换，把多种传统资源或外部资源信息中不完善的数据自动转换为准确可靠的数据。

数据转移技术有 4 种转移类型：　1）简单转移　（数据元的类型转换、日期/时间格式转换、字段解码）

2）清洗　　（检查字段中的有效值、重新格式化某些类型的数据，如地址）

3）集成

4）聚集和概括

10、OLTP，联机事务处理，以快速事务响应和频繁的数据修改为特征，方便用户使用数据库快速处理具体业务。

OLAP，联机分析处理，是以数据仓库进行分析决策的基础，是针对待定问题的联机数据访问和分析。

它们是两类不同的应用。

OLTP 与 OLAP 的对比：

OLTP	OLAP
数据库原始数据	数据库导出数据或数据仓库数据
细节性数据	综合性数据
当前数据	历史数据
经常更新	不可更新，但周期性刷新
一次性处理的数据量小	一次性处理的数据量大
响应时间要求高	响应时间合理
用户数量大	用户数量相对较少
面向操作人员，支持日常操作	面向决策人员，支持管理需要
面向应用，事务驱动	面向分析，分析驱动

11、企业决策支持解决方案，需要从决策支持系统的体系结构（C/S 较流行）、软件工具和开发（原型法、生命周期法）3 个方面加以分析。

第十五章 知识产权基础 第十六章　标准化

1、知识产权的概念，是指民事权利主体基于智力创造性的智力成果。可以分为工业产权和著作权（版权）。

知识产权的特点：无形性、双重性、确认性、独占性、地域性、时间性。

2、计算机软件受著作权保护的条件：按计算机软件保护条例规定，受保护的软件产品应“独立创作、可被感知、逻辑合理”。

计算机软件保护条例规定，合法受保护的对象为“程序和文档”，著作权法不保护软件开发使用的思想、概念、发现、原理、算法、处理过程和运算方法。

计算机软件著作权的权利有 2 类，人身权和财产权。人身权包括发表权、开发者身份权（署名权）。

财产权，计算机软件保护条例规定，软件著作权人享有：使用权、复制权、修改权、发行权、翻译权、注释权、信息网络传播权、出租权、使用许可权和获得报酬权、转让权。

计算机软件著作权的归属问题：

1）职务开发软件著作权的归属；

2）合作开发软件著作权的归属；

3）委托开发的软件著作权的归属；

4）接受任务开发的软件著作权归属；　　－－3,4 两条中都是先依据合同或协议来确认著作权归属，如未明确归定则著作权属于实　　际的开发者。

5）主体变更后软件著作权的归属；

6）权利转让后软件著作权的归属；

7）司法、判决引起的软件著作权归属问题；

8）保护期满权利丧失。

3、计算机软件的商业秘密权

商业秘密的定义：在＜中华人民共和国反不正当竞争法＞中将商业秘密定义为“不为公众所知的、能为权利人带来经济利益、具有实用性并经权利人采取保密措施的技术信息和经营信息”。经营秘密和技术秘密是商业秘密的基本内容。

商业秘密的构成条件：未公开性、具有实用性、保密性。一项商业秘密要受到法律保护，必须具备这三个条件。

＜中华人民共和国反不正当竞争法＞保护计算机软件，是以计算机软件中是否包含着“商业秘密”为必要条件。即使软件尚未开发完成，在软件中已经形成的知识内容也可以作为商业秘密。

4、授予专利权的条件：新颖性、创造性、实用性。

专利法不适用的对象有：违法违德妨害公共利益的发明，客观世界已存在但未揭示的规律、性质和现象，智力活动的规则和方法，病的诊断及治疗方法，动物和植物的品种，原子核变换得到的物质。

发明专利的保护期为 20 年，实用新型专利权和外观设计专利权的保护期为 10 年。公民的作品发表权的保护期为公民终生及其死后 50 年。我国商标权的保护期为 10 年，到期时可以续注，每次 10 年。计算机软件著作权保护期为 50 年。

我国著作权法不保护“法律、法规、国家机关的决议等性质的文件，及其官方正式译文。

我国著作权法规定：作者的署名权、修改权、保护作品完整权的保护期不受限制。

著作权法规定：为介绍、评论某一作品或者说明某一问题，在作品中适当引用他人已经发表的作品，可以不经著作人许可，但应当指明作者姓名、作品名称且不能侵犯著作权人享有的其他权利。

著作权法规定：汇编若干作品、作品的片段或者不构成作品的数据或者其他材料，对其内容的选择或者编排体现独创性的作品，为汇编作品，其著作权由汇编人享有，但行使著作权时，不得侵犯原作品的著作权。

软件工程师接受单位的任务，独立完成了某应用软件的开发和设计，其软件著作权属于“单位的法人”。

我国专利法规定，申请专利的发明创造，在申请日以前 6 个月内，有下面情况发生的，不会丧失新颖性：

1）在中国政府主办或者承认的国际展览会上首次展出的；

2）在规定的学术会议或者技术会议上首次发表的；

3）他人未经申请人同意而泄露其内容的。

专利申请日：以专利局收到完整专利申请文件的日期为专利申请日。如果申请文件是邮寄来的，则以寄出的邮戳日期为申请日。

5、标准化的基本过程：标准产生子过程、标准实施子过程、标准更新子过程。

标准的编号：　国际标准编号形式为－－　标准代号＋专业类号＋顺序号＋年代号

我国标准编号形式为－－　标准代号＋标准发布顺序号＋年代号（年号即发布年份的后两位数字）

国家标准代号：GB（强制性的）　GB/T（推荐性的）

行业标准代号：由汉语拼音大写字母组成，加上斜线 T 后变为推荐标准。如：JR（金融）

地方标准代号：由 DB 加上两位地方代码（北京为 11）

企业标准代号：由 Q 加上斜线再加上企业代号组成。如 Q/**。

6、国际标准组织：　ISO－－国际标准化组织

IEC－－国际电工委员会

ITU－－国际电信联盟

国家标准组织：　ANSI－－美国国家标准

DIN－－德国国家标准

BS－－英国国家标准

JIS－－日本　SIS－－瑞典　SNV－－瑞士　NF－－法国

UNI－－意大利

GJB－－中华人民共和国国家军用标准

行业协会标准：　IEEE－－美国电气电子工程师学会

7、采用国际标准的程度分为：等同 IDT、等效 EQV、非等效 NEQ。

8、过程评估标准 ISO/IEC 15504，它提供了一个软件过程评估的框架。它可以被任何软件企业用于软件的设计、管理、监督、控制以及提高获得、供应、开发、操作、升级和支持能力。它涉及了过程评估的各个方面，其文档主要包括 9 个部分。

ISO 标准每 5 年复审一次。我国的国家标准有效期也是 5 年。国家标准的最后两位数字是年号。

条码中对应条码符号的一组阿拉伯数字称为条码代码，条码代码供人们直接识读，可以通过键盘向计算机输入数据。

ISO9000：2000＜质量管理体系　基础和术语＞标准提出的“8 项质量管理原则”，是在总结了质量管理经验的基础上，明确了一个组织在实施质量管理中必须遵循的原则，也是 ISO9000：2000 族标准制定的指导思想和理论基础。

ISO9000 系列标准是国际化标准化组织质量管理和质量保证技术委员会于 1987 年分布的质量管理和质量保证系列标准。

目前 ISO9000：2000 系列标准它包括 5 项具体标准。

ISO12207 是软件生命周期过程的国际标准。

ISO/IEC JTC1 是制定信息技术领域国际标准的机构。