1. 计算机组成及体系结构
2. 数据的表示：

原码：首位为符号位，不适合做加减，会产生偏差。

反码：正数和原码相同，负数则是在原码的基础上取反。

补码：正数和原码相同，负数在补码的基础上+1。

移动：在特定场合，做浮点运算当中的阶码，与补码符号位相反。

浮点数的精度由尾数的位数决定，表示范围的大小则主要由阶码的位数决定。

1. 计算机结构：
2. 流水线：

^t 为流水线周期。

①理论公式：(t1+t2+...+tk) + (n - 1) \* ^t；

②实践公式：(k + n - 1) \* ^t；k指流水线分成多少个步骤，比如取指，分析，执行k为

③吞吐率：

寻址方式：

1. 立即寻址：直接给出操作数，而非地址。
2. 直接寻址：直接给出操作数地址或所在寄存器号（寄存器寻址）
3. 间接寻址：给出的是指向操作数地址的地址，称之为间接寻址

（4）变址寻址：给出的地址需与特定的地址值累加从而得出操作数地址，称之为变址

1. 计算机存储结构：

Cache：可不比存在，但速度会大打折扣，同时可以按内容存储，也叫相连存储器。速度大于按地址存储。

存储体系：般计算机系统中主要有两种存储体系：Cache存储体系由Cache和主存储器构成，主要目的是提高存储器速度，对系统程序员以上均透明；虚拟存储体系由主存储器和在线磁盘存储器等辅存构成，主要目的是扩大存储器容量，对应用程序员透明。

1. 校验码：

模2除法，按位进行异或操作。

海明校验码：2^r >= x + r + 1; x 为信息为的长度，如；1011，x为4；

当进行异或操作时，两个二进制的数相比之后，若前面的树为零，则再与第三个数相比时，位为零的可以不再进行相比。

1. 程序语言基础
2. 递归下降分析法和预测分析法是常用的自顶向下分析法。算符优先分析法和LR分析法属于自底向上的语法分析方法。
3. 语法分析方法有多种，根据产生语法树的方向，可分为自底向上和自顶向下两类。递归下降分析法和预测分析法是常用的自顶向下分析法。算符优先分析法和LR分析法属于自底向上的语法分析方法。
4. 表达式：
5. 逆波兰式:
6. 同步和互斥：

互斥：当一个人使用时，其他人必须等待。

同步：当两件事情同时做时，若有一件事件快了，则需等待另一件事情，最后是两件事情同时完成。同步与互斥犹如缓冲区。

6. 后缀式(逆波兰式)是波兰逻辑学家卢卡西维奇发明的一种表示表达式的方法。这种表示方式把运算符写在运算对象的后面，例如，把a+b写成ab+，所以也称为后缀式。

1. 操作系统
2. 运行状态：有了cpu资源。

就绪状态：只缺cpu资源。

等待状态：没有cpu，还缺其他资源。

1.索引结构，一般有13个节点，从0~12

2.第\_\_字则是从1开始记，第\_\_位置则是从0开始算

3.页内地址为页面大小(多少k)化成2的多少次方，然后根据有多少位，再和题目中的逻辑地址比较(比较是页内地址的位数和逻辑地址从右往左数)，然后逻辑地址的最高位为页号

4.二进制的位数为

次方

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 1024 | 2048 | 4096 | 8192 | 16384 | 32768 |  |  |  |  |

5.虚实地址转换

( ( (x) + s ) + p ) \* ( 2 \*\* x ) + d

其中x为基号，s为段号，p为段内页号，d为页内偏移，n的值为d的总位数，（x）表示x里的内

容。

1. 进程

为了解决进程间的同步和互斥问题，通常采用一种称为信号量机制的方法

1. 磁道题目的延迟时间：
2. 不发生死锁的公式：k \* ( n - 1 ) + 1；k为进程数，n为所需要的资源；

死锁发生的必要条件

（1）互斥条件：即一个资源每次只能被一个进程使用，在操作系统中这是真实存在的情况。

（2）保持和等待条件：有一个进程已获得了一些资源，但因请求其他资源被阻塞时，对已获得

的资源保持不放。

（3）不可剥夺条件：有些系统资源是不可剥夺的，当某个进程已获得这种资源后，系统不能强

行收回，只能由进程使用完时自己释放。

1. 环路等待条件：若干个进程形成环形链，每个都占用对方要申请的下一个资源
2. 淘汰页面：状态位为1的进行考虑淘汰，访问位为0的进行淘汰，修改位为0的进行淘汰。
3. 指令只会产生一次缺页中断，而操作数则是两次。
4. 索引结点没有说明的时候，一般采用13个结点。

**第 4 章 系统开发基础**

软件生命周期：

1. 可行性分析与项目开发计划，2.需求分析，3.概要设计，4.详细设计，5.编码，6.测试，维护。
2. LOC软件规模代码行（LOC, Line of Code）是软件规模的一种量度，它表示源代码行数。

2. 而数据处理领域的不太复杂的软件，适于用结构化方法进行开发。

1. 瀑布模型：结构化方法。适用于需求明确场合，二次开发。
2. 原型：先做一个简易的系统给用户体验，再让用户明确他想要的功能，针对需求不明确的情况。需求不清晰且规模不太大时采用原型化方法最合适
3. 演化模型：原型一步一步演化完善为演化模型
4. 增量模型：先做一点，再接着做，特点是可以更早的让用户和核心功能接触
5. 螺旋模型：风险分析是螺旋模型最为显著的特征之一，具有原型，增量模型等特点。
6. V模型：特点之一，测试分为较细步骤。开发与测试有着一一的对应关系。强调测试。
7. 喷泉模型：最大特点是面向对象的模型，迭代和无间隙，开发过程模型以用户需求为动力，以对象为驱动，适合于面向对象的开发方法。
8. RAD（快速开发模型）：业务建模：数据建模，过程建模，应用生成，测试与交付。结合瀑布模型和构建化开发。
9. 构建组装模型(CBSD)：构件组件特点，组成构件库，极大地提高了软件开发的复用性，可靠性，效率等。
10. 统一过程(up/rup)模型：用例驱动，以架构为中心，迭代和增量。
11. 敏捷开发方法：是一组模型，一般做小型项目，快速。
12. 信息系统开发方法：

结构化法：缺点，流程固化，不容易变化。

原型法： 应对需求不明确，做个简单的界面效果。

面向对象方法：

面向服务方法。

1. 内聚：模块的内部功能紧密程度，越高越好；

耦合：模块与模块之间的关联，越小越好；

1. 软件工程是一种层次化的技术，从底向上分别为质量、过程、方法和工具

关键路径是指设计中从输入到输出经过的延时最长的逻辑路径。关键路径上的活动均为关键活动。

敏捷方法

|  |  |
| --- | --- |
| 极限编程XP | 是激发开发人员创造性、使得管理负担最小的一组技术. |
| 水晶法Crystal | 认为每—个不同的项目都需要一套不同的策略、约定和方法论。 |
| 并列争球法(Scram) | 使用迭代的方法，其中把每30天一次的迭代称为个冲刺， 并按需求的优先级来实现产品多个自组织和自治小组并行地递增实现产品，协调是通过简短的日常情况会议进行。 |
| 自适应软件开发（ASD) | 六个基本的原则:  ①在自适应软件开发中，有一个使命作为指导，它设立了项目的目标，但不描述如何达到这个目标；  ②特征被视为客户键值的关键，因此，项目是围绕着构造的构件来组织并实现特征；  ③过程中的迭代是很重要的，因此重做与做同样重要，变化也包含其中；  ④变化不视为是一种更正，而是对软件开发实际情况的调整；  ⑤确定的交付时间迫使开发人员认真考虑每一个生产版本的关键需求；  ⑥风险也包含其中，它使开发人员首先跟踪最艰难的问题。 |

17.软件配置管理SCM用于整个软件工程过程，其主要目标是标识变更、控制变更、确保变更正确的实现，报告变更。其主要内容包括版本管理、配置支持、变更支持、过程支持、团队支持、变化报告和审计支持等。

18. 软件评审的内容包括设计质量评审、程序质量评审和与运行环境接口的评审。评审的主要目标是为了发现软件中的错误。

其中决策树和决策表适于用来表示加工中涉及多个逻辑条件的情况。

结构化设计：

概要设计：

详细设计：

软件测试

测试原则：

测试类型：

动态测试：

黑盒测试：等价类划分，边界值分析，错误推测，因果图。

白盒测试：

灰盒测试：

静态测试：

桌前检查：

代码走查：

代码审查：

测试用例设计：

13. 环路复杂度公式：V(G) = m - n + 2; m代表弧数，线条。n代表节点。

14. 由于数据流图只反映系统必须完成的逻辑功能，所以它是一种功能模

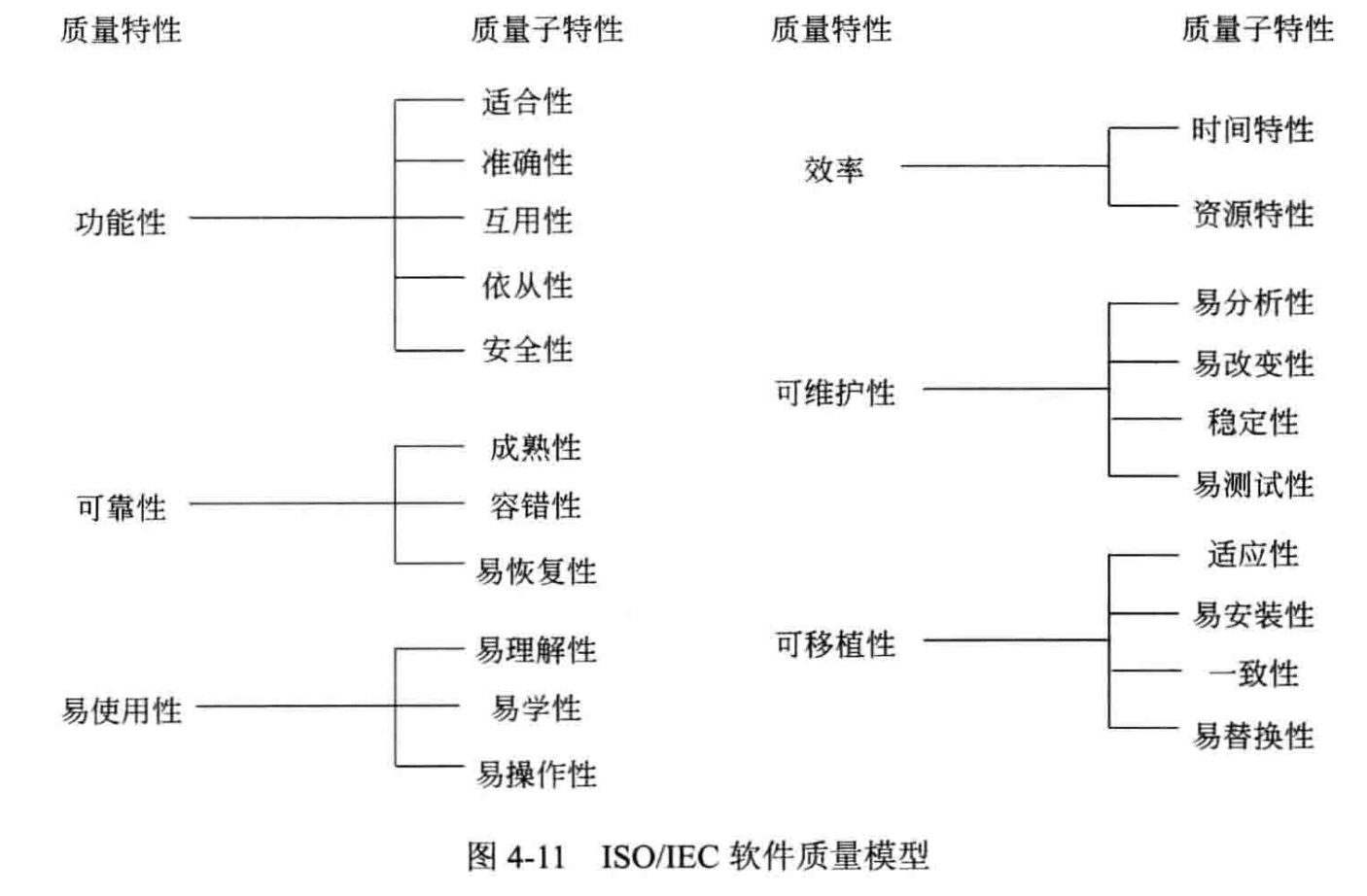
型

1. CVS 是一种版本控制软件工具。

CMM -> CMMI：能力成熟度模型，

17. pert图计算：选择最长路径为关键路径，最晚开始时间用关键路径逆推得出最晚开始时间。

18. TCP三次握手：1-SYN\_RECV，2-SYN\_RECV，3-ESTABLISHED



**第 6 章 网络信息安全与多媒体知识**

1..ICMP：ping命令属于此

ARP：ip换mac

DNS：域名解析

DHCP：动态地址分配,169.254.X.X 和 0.0.0.0 是假地址。

SNMP：简单网络管理协议。

TFTP：跟FTP区别，FTP可靠的协议，TFTP不可靠的协议。

IP转MAC是ARP协议，MAC转IP是RARP。

中间件技术：

A.Java RMI B.CORBA C.DCOM

物理地址：物理地址通常是指主机的网卡地址（MAC地址），每一网卡都有唯一的地址。

路由汇聚算法：

设有2条路由21.1.193.0/24 和 21.1.194.0.24

先吧两个网络地址转换成二进制，然后取完全相同的部分作为聚合路由的ip  
如193：00010101 00000001 11000001 00000000  
而194：00010101 00000001 11000010 00000000  
完全相同的是00010101 00000001 110000XX XXXXXXXX  
相同部分长度就是掩码长度  
所以聚合后是21.1.192.0/22

地址解析：就是主机在发送帧前将目标IP地址转换成目标MAC地址的过程

路由器功能：

路由器工作于网络层，路由器的主要功能有：

① 选择最佳的转发数据的路径，建立非常灵活的连接，均衡网络负载；

② 利用通信协议本身的流量控制功能来控制数据传输，有效地解决拥挤问题；

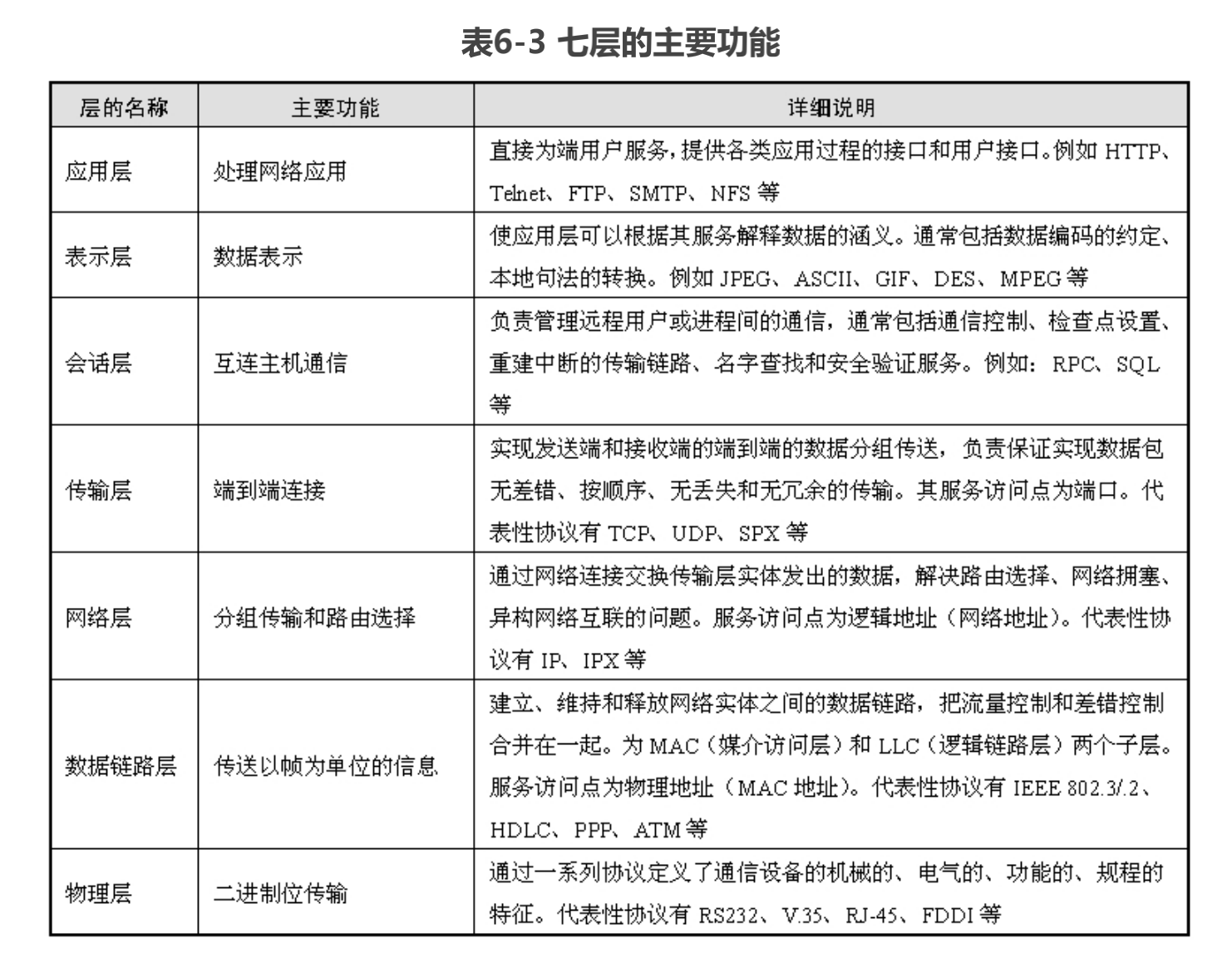
③ 具有判断需要转发的数据分组的功能，不仅可根据LAN网络地址和协议类型，而且可根据网

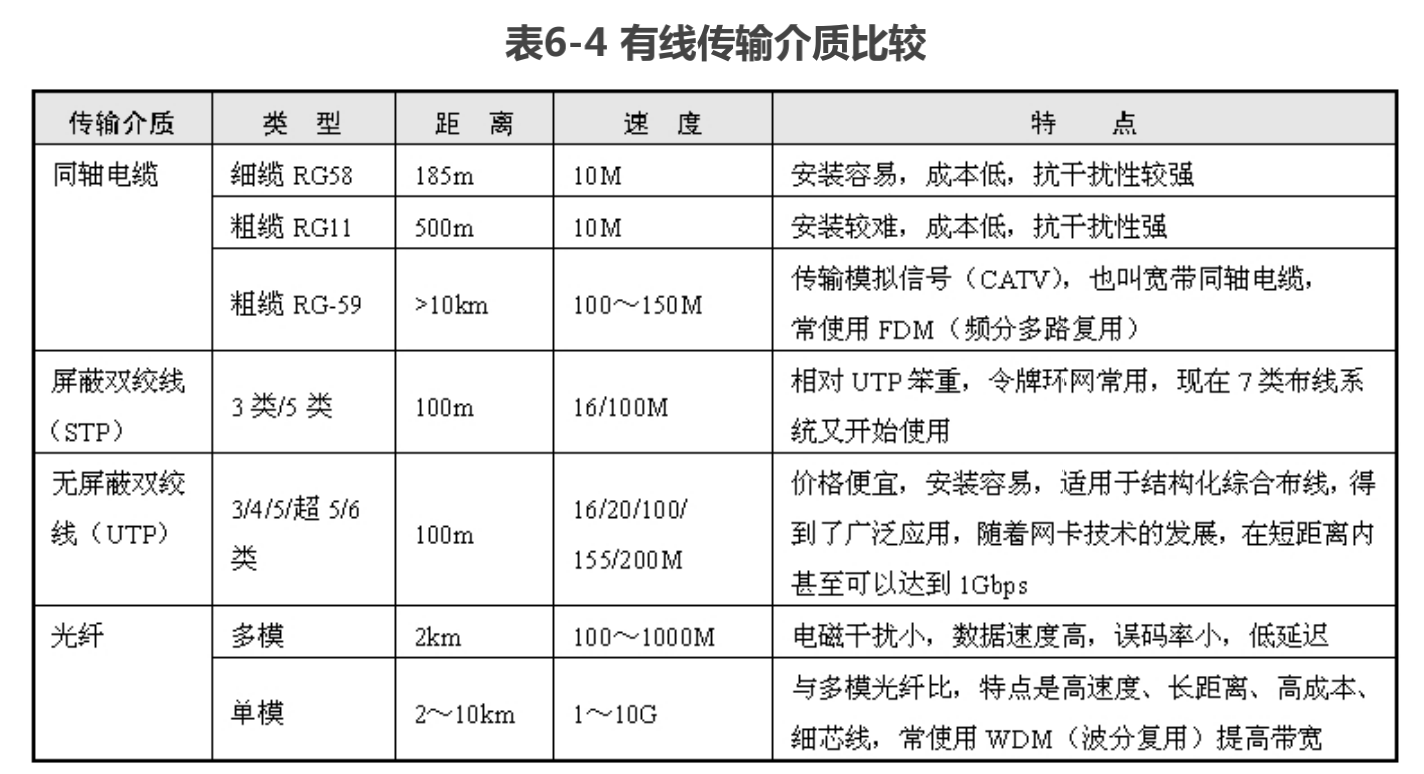
间地址、主机地址、数据类型（如文件传输、远程登陆或电子邮件）等，判断分组是否应该转发。

对于不该转发的信息（包括错误信息），都过滤掉，从而可避免广播风暴，比网桥外有更强的隔离

作用，提高安全保密性能；

④ 把一个大的网络划分为若干个子网。





TCP的流量控制采用了可变大小的滑动窗口协议，由接收方指明接收缓冲区的大小(字节数)，发送方发送了规定的字节数后等待接收方的下一次请求。

IP地址：

主机号全为0的地址为网络地址，不是主机地址。全1为广播地址。

A类：前8位为网络号，后面为主机号。

B类：前16位为网络号，后16为主机号。

C类：

D类；

ip地址/位数 表示前面多少位数转为网络号。

判断子网是否相同：把两个ip地址化成十进制，看前面的位数是否相同。

信息安全

非对称加密：甲(公钥),甲(私钥)，乙(公钥),乙(私钥)。

使用乙的公钥给甲加密，乙用私钥解密。

数字签名：用谁的公钥打开的这个包，就是用谁的私钥加密，识别身份的作用。

防火墙：网络级，只检查ip地址头。

应用级：开箱检查。

1. 主动攻击：重放攻击、IP地址欺骗、拒绝服务攻击

2. 被动攻击：流量分析攻击是通过持续检测现有网络中的流量变化或者变化趋势，而得到相应信息的一种被动攻击方式。

1. 终端设备与远程站点之间建立安全连接的协议是SSH
2. 机房安全属于物理安全，入侵检测属于网络安全，漏洞补丁管理属于系统安全，而数据库安全则是应用安全。

多媒体技术

1.颜色深度：如颜色深度为24位，则需要24个位来存储。

2.小写k代表1000，采样和传输用的小写k，存储用的大写K代表1000；

常见多媒体标准

|  |  |  |
| --- | --- | --- |
| JPEG |  |  |
| JPEG-2000 |  |  |
| MPEG-1 |  |  |
| MPEG-2 |  |  |
| MPEG-4 |  |  |
| MPEG-7 |  |  |
| MPEG-21 |  |  |

1. 采样率：

彩色空间：

RGB，颜色叠加

YUV：黑白电视和彩色电视兼容。

CMY：颜色相减

小写k和大写K：小写k 1000 ，大写K 1024。存储数据1024，传输数据1000；

压缩：前提是必须有冗余。

1. **数据库技术基础**

实体转换关系模式：

1：1联系：两边实体转成一个，把中间的联系可以转成单独的关系模式，也可以把联系记录到任意一个实体里面。至少转两个。

1：n联系： 两边实体分别转成一个，把联系放到任何一个实体记录下来，也可以单独记录下来。一般记录到多的那一方。

m：n联系：至少转三个，两个实体和联系都要转。

并：将两个集合合并在一起，重复的不显示。

交：集合相交，只显示共同的。

差：集合相减S1-S2，结果S1有的但S2没有的。

笛卡尔积：集合相乘的个数，S1每个集合都要和S2的集合连在一起。

投影：结果是遍历选个的字段的那几列。Select id,name from user；

选择：只选择符合条件的，如sql语句后面加上where id =1；

选择和投影都可以用数字1，2来代表列名。

连接：将两个表拥有的共同属性的行连接起来，如where user.id = student.uid。

**函数依赖：**由事件X可以确定结果Y的，称为X -> Y，X决定Y，Y依赖于X。

如(学号可以确定姓名，反过来，姓名也许会同名，不能确定唯一的学号)。

部分函数依赖：组键是两个属性的组合键，由组键中的部分属性可以确定由事件(A,B)->C，但是因为A可以单独决定C，称为部分函数依赖。

传递函数依赖：A -> B ，而 B -> C，则是函数传递依赖，前提是B不能确定A。

**非规范化的关系模式，**可能存在的问题包括：数据冗余，更新异常，插入异常，删除异常。

数据冗余： 表中的数据存在多余的数据或者重复的数据。

更新异常：当表中改名的时候，本想更改需要更改的信息，结果更改的时候只更改了一部分，其他的不能同步更改。

插入异常：

删除异常：

**键：**

**范式：**

**丢失更新：t1和t2运行，t2的结果会覆盖t1的结果。**

**不可重复度：当t1进行第二次读取时，结果和第一次的结果不相同。**

**读脏数据：不是在执行的真正数据，而是临时值，没有产生作用的值。**

**x锁是写锁，加上之后其他事务不能再加任何锁。**

**s是读锁，加上之后其他事务还可以加s锁，不能加x锁。**

反规范化主要是提升查询速度，牺牲空间等。

**第 8 9 章 数据结构和算法分析**

一维数组存储地址: a[i] a + i\*len ;

二维数组按行存储: a[i][j] a + ( i\*n + j ) \* len;

二维数组按列存储: a[i][j] a + ( j\*m + i ) \* len;

广义表：长度，指元素个数。深度，指有多少层括号。表头，表的第一个元素。表尾，除表头以外的所有元素。

时间复杂度：关于所有常数集条数的命令都用O(1)来表示。

二分查找法：必须要是有序的数列才行。如(1,5,8,12,15)。查找出来的树取整。

散列表：按内容存储。

希尔排序：先将元素按元素距离和另一个元素变成一组进行对比，即一组元素可以分成多个组，对比完一轮之后，再将距离缩小比较，最后进行直接插入排序。

二叉树具有下列重要性质：

性质1 在二叉树的第i层上至多有2

i-1≥1）。

性质2 深度为k的二叉树至多有2

k

-1个结点（k≥1）。

性质3 对任何一棵二叉树，如果其叶子结点数为n0,度为2的结点数为n2,则n0=n2+1.

较高效率地解决背包问题一般用递归和贪心算法。

1. 关键路径：关键路径就是从源点到汇点权和最大的那条路径
2. 排序二叉树越平衡，查找效率越高。

**最优二叉树（哈弗曼树）**

树的路径长度：树的路径相加的长度，从根开始算起的单条路径。

权：某个叶子节点有一个数值，代表某个数值出现的频度。

带权路径长度：路径长度\*权值。

**算法基础**

算法定义及应用场景：

|  |  |
| --- | --- |
| 插入排序 | 若数据基本有序，插入排序应该是最佳选择，输入数据是否有序对归并和计数排序算法并没有影响。 |
| 计数排序 | 若关键字取值范围较小，则计数排序是最佳选择，因为在该情况下，该算法的时间复杂度为线性时间。 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**第十章 面向对象分析**

消息：对象之间进行交互的时候所采用的机制，走的是异步的方式传输的。

面向对象分析的步骤：

|  |
| --- |
| 第一步，确定问题域。 |
| 第二步，确定对象和类。这里所说的对象是对数据及其处理方式的抽象，它反映了系统保存和处理现实世界中某些事物的信息的能力。类是多个对象的共同属性和方法集合的描述，它包括如何在一个类中建立一个新对象的描述。 |
| 第三步，确定结构（structure）。结构是指问题域的复杂性和连接关系。类成员结构反映了泛化-特化关系，整体-部分结构反映整体和局部之间的关系。 |
| 第四步，确定主题（subject）。主题是指事物的总体概貌和总体分析模型。 |
| 第五步，确定属性（attribute）。属性就是数据元素，可用来描述对象或分类结构的实例，可在图中给出，并在对象的存储中指定。 |
| 第六步，确定方法（method）。方法是在收到消息后必须进行的一些处理方法，方法要在图中定义，并在对象的存储中指定。对于每个对象和结构来说，那些用来增加、修改、删除和选择一个方法本身都是隐含的，而有些则是显示的。 |

面向对象分析阶段包含5个活动：

认定对象、组织对象、描述对象间的相互作用、定义对象的操作、定义对象的内部信息。

UML的五个系统视图：

（1）逻辑视图：用来显示系统内部的功能是怎样设计的，它利用系统的静态结构和动态行为来

刻画系统功能。静态结构描述类、对象和它们之间的关系等。动态行为主要描述对象之间的动态协

作。

（2）进程视图（并发视图）：用来显示系统的并发工作状况。并发视图将系统划分为进程和处

理机方式，通过划分引入并发机制，利用并发高效地使用资源、并行执行和处理异步事件。它是逻

辑视图的一次执行实例。

（3）实现视图（组件视图）：用来显示代码组件的组织方式。它描述了实现模块和它们之间的

依赖关系。

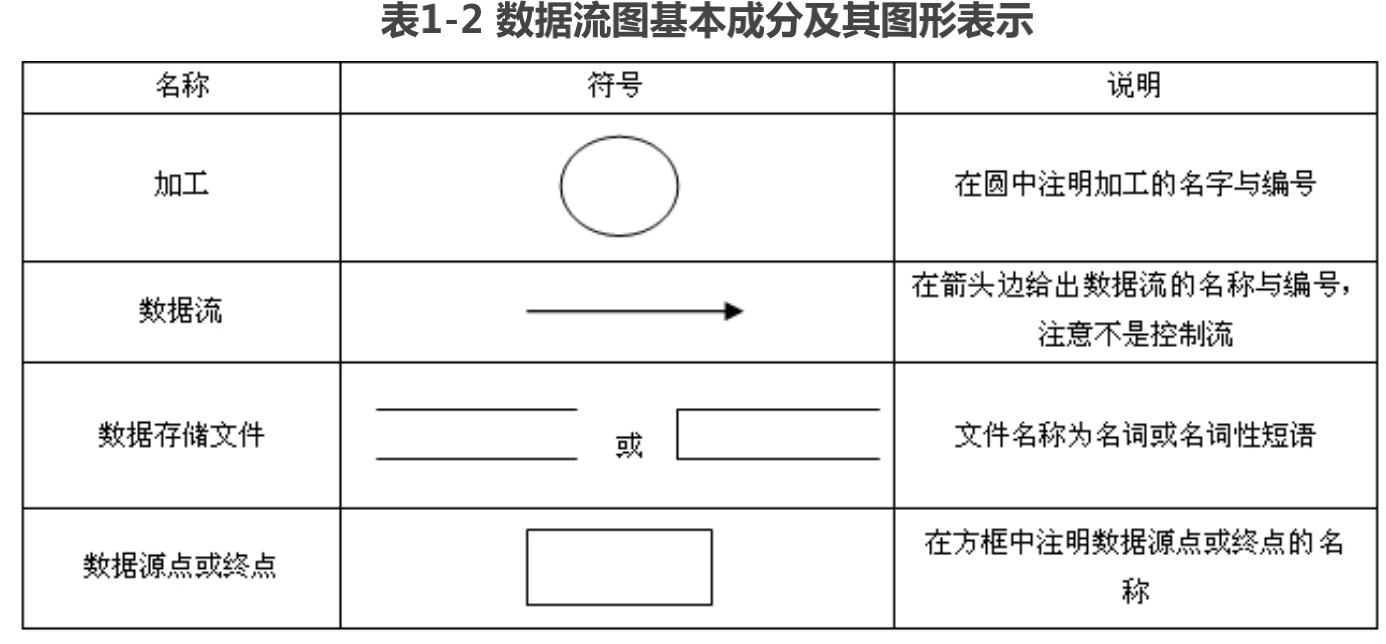
（4）部署视图：用来显示系统的物理架构，即系统的物理展开。比如，计算机和设备以及它们

之间的联接方式。其中计算机和设备称为结点。

（5）用例视图：最基本的需求分析模型，以外部参与者的角度来看待系统，他主要说明了谁要

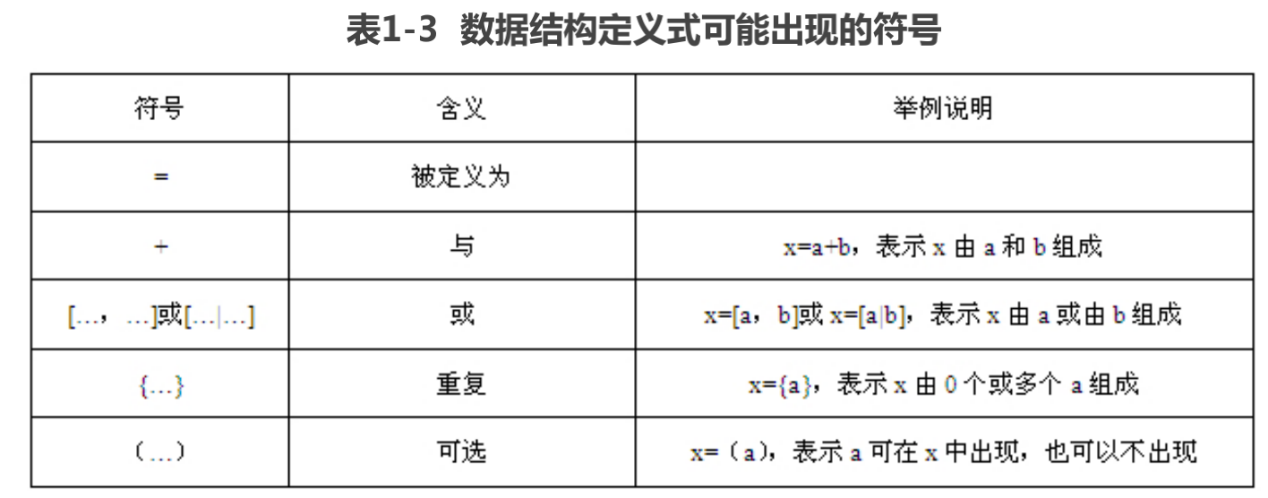
使用系统以及他们使用了该系统可以做些什么

**第 12 章 面向对象设计 数据流图**

****

采用数据流图的方式进行数据流程分析一般应遵循以下原则：

1. 明确系统边界。一张数据流图表示某个子系统或某个系统的逻辑模型。
2. 在总体上遵循自顶向下逐层分解的原则，即按照结构化方法的思想，采用分层的数据流图，把大问题、复杂的问题分解成若干个小问题，然后分别解决
3. 在局部上遵循由外向里的原则，即先确定每一层数据流图的边界或范围，
4. 分层的数据流图一般由顶层、中间层和底层组成。



解题技巧：

1. 找出外部实体，数据流，存储实体等关键字。

2. 0层图一般有数据存储关键字：XX表，XX文件。

3. 对于添加、删除、查找数据流的问题：

先仔细阅读分析每一条需求，接着在顶层数据流图或者0层数据流图一一对应数据流的数目和方向及内容是否一直。

分析顶层数据流图和0层数据流图的一致性。

题目：

1. 数据流图是在系统分析与总体设计阶段宏观地描述系统功能需求的重要图形化工具，程序流程图也是软件开发过程中比较常用的图形化工具。简要说明程序流程图的适用场合与作用。

答：程序流程图通常在进行详细设计时使用，用来描述程序的逻辑结构。

2. 数据流图中可能出现的错误：

1. 黑洞：有输入没有输出。

2. 奇迹：没有输入也有输出。

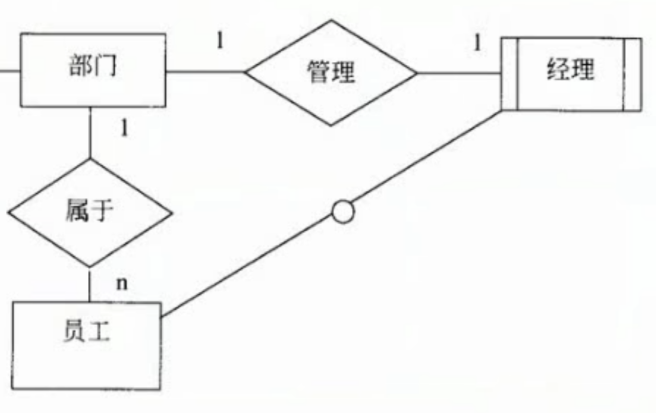
3. 命名冲突：输入流和输出流命名相同。

4. 结果不同：输入流根本不可能产生这种输出。

**第 12 章 面向对象程序设计 数据库设计**

1. 一对一联系：可以将联系单独做一个关系，也可以将联系记录到任意一方。
2. 一对多联系：只能将联系记录到多的一边。
3. 多对多联系：必须单独转成一个关系模式。

单出现像员工和经理这种关系的时候，可以用线和圆圈表示特殊关系。



1. 模型优化

改善数据库性能的考虑

①减少联接运算

②减小关系大小及数据量

③尽量使用快照

节省存储空间的一些

①缩小每个属性占用的空间

②采用假属性

1. 有问题是说将一张关系表转移到另一张关系表。

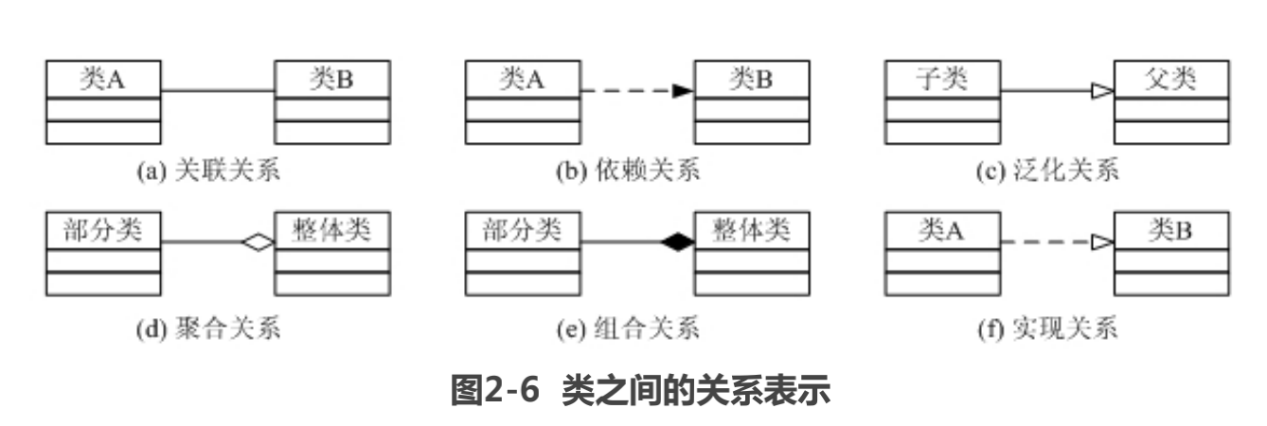
缺点：数据要大量重复存储，数据冗余。而且还可能更新异常等。

优点：提高存储速度。

6. 当给出重名现象时，则需要两个关系模式。

**第 12 章 面向对象程序设计 UML建模**

1. uml图考点：项目的详细描述，完整用例图当中的参与者和用例去掉，根据题干内容的哪个用例交叫什什么名称；两个用例之间是什么关系，例如，关联，泛化等。
2. 包含关系：做一件事情，必须做另一件事情，则是包含。关键字，是否。
3. 扩展关系：做一件事情，另一件事情有时候用到，有时候不用。关键字：必须；
4. 类图：泛化对类来讲，实现对接口。
5. 将顺序图和通信图统称为交互图。顺序图会强调时间顺序。



**第 12 章 面向对象程序设计 数据结构及算法**

1.分治法：

递归

二分法查找

回溯法

贪心法：每一步取最优，但结果不一定是最优。得到的不一定是最优解。

动态规划法：和分治法区别，基本都要用到查表得问题。

解题步骤：

先把问题中的元素排列出来，了解该算法的顺序流程。

赋初值的语句的时间复杂度是O(1);

循环语句看循环体的次数，n次的时间复杂度则为O(n);

循环嵌套语句和n相关，时间负责的则为O(n^2);

时间复杂度的总时间以最高为主。

**第 12 章 面向对象程序设计 面向对象程序设计**