

## 1 已经完成的题目

2018, 上半年, 上午。2020, 下半年, 上午。

程序计数器 PC 用于存放计算机将要执行的指令所在的存储单元地址指令寄存器 IR 保存从存储器中取出的指令（正在执行的指令）地址寄存器存放当前 CPU 所访问的内存单元地址指令译码器对指令寄存器 IR 中的指令进行译码分析

## 2 Computer Basic

计算机体系结构分类-Flynn, 根据指令流与数据流 SISD, 单处理器系统 SIMD, 并行处理机, 阵列处理机 MISD, 流水线计算机, 理论模型 MIMD, 多计算机

CISC 复杂, 指令: 多, 使用频率差别大, 可变长; 支持多种寻址方式; 微程序控制技术实现; 研制周期长。RISC 精简, 指令: 少, 使用频率差别小, 定长, 大部分为单周期指令, 只有 Load 和 Store 操作内存; 支持寻址方式少; 通用寄存器 (速度快), 硬布线逻辑控制为主, 适合用流水线; 优化编译, 高级语言支持性好。

流水线: 取址 2ns, 分析 2ns, 执行 1ns; 流水线周期: 执行时间最长的一段, 2ns。100 条指令用时: 理论公式  $2+2+1 + 99 * 2 = 203\text{ns}$ , 第 1 条指令完整 + 剩余的 \* 流水线周期 100 条指令用时: 实践公式  $2+2+2 + 99 * 2 = 204\text{ns}$ , 第 1 条指令每一步按照流水线周期算 + 剩余的 \* 流水线周期吞吐率: Through Put rate, TP, 指令条数/流水线执行时间,  $100/203$ ; 吞吐率极限:  $1/\text{流水线周期}$ 流水线加速比: 不用流水线的时间/用流水线的时间

### 2.1 数据表示

10 进制转换成二进制, 短除法, 最后余 1 或 0 停止, 结果倒着写。原码, 最高位的符号, 数的绝对值; 反码, 正数同原码, 负数是符号位为 1, 数的绝对值的相反数补码, 正数同原码, 负数是符号位为 1, 数的绝对值的相反数加上 1 移码, 在补码基础上, 将符号位取反, 使得正数表示起来比负数大。

浮点数表示  $N=M \cdot R^K$ ;

$1.0\text{e}3 + 1.2\text{e}2$

$= 1.0\text{e}3 + 0.12\text{e}3$  //阶数对齐,

$= 1.02\text{e}3$  // 尾数计算;

$= 1.02\text{e}3$  // 格式化

两个浮点数相加减, 首先需要比较阶码大小, 使小阶向大阶看齐(称为“对阶”)。即小阶的尾数向右移位(相当于小数点左移), 每右移一位, 其阶码加1, 直到阶码相等, 右移的位数等于阶差。

## 2.2 IO

### 2.2.1 控制方式

程序控制：需要主动问程序是否完成。

程序中断：程序运行完发送中断让系统处理。

DMA，DMA 不需要 CPU 执行程序指令。专门的控制器。主存与外设直接成块传送。

无条件传送、程序查询、

## 2.3 校验码

码距：编码系统中任意两个码字的最小距离。码组内检测  $k$  个错误码，最小码距  $d \geq k+1$ ；码组内纠正  $k$  个错误码，最小码距  $d \geq 2k+1$ 。

奇偶校验 Parity Code,

### 2.3.1 CRC

循环冗余校验码，Cyclic Redundancy Check, CRC。可检测错误。思路是传输数据末尾增加一些位，使得计算指定的运算后余数为 0。如模 2 除法，写还是按照除法竖式计算，算的时候按位异或不计进位。CRC 举例：1) 例如对于多项式  $x^4 + x^3 + x + 1$ , 各次的系数为：11011，作为模 2 除法的除数；2) 多项式有 5 位，原始信息末尾补 5 个 0；3) 计算模 2 除法，得到余数；4) 则最终得到：原始信息 + 余数

### 2.3.2 Hamming Code

海明码 Hamming Code,  $n$  位数据,  $k$  位校验, 满足  $2^k - 1 \geq n + k$ 。

如求 1011 的海明码:

1) 求得校验位需要 3 位, 分别放在 1, 2, 4 位 (2 的  $i$  次方上), 分别记为  $a, b, c$ 。

2) 从右往左写: 1 0 1 c 1 b a

7 6 5 4 3 2 1

7=111; 6 = 110; 5 = 101; 3 = 011

所以  $c$  由 7, 6, 5 位异或,  $c = 1 \oplus 0 \oplus 1 = 0$

$b$ , 7, 6, 3,  $b = 1 \oplus 0 \oplus 1 = 0$

$a$ , 7, 5, 3  $a = 1 \oplus 1 \oplus 1 = 1$

## 2.4 存储

CPU 中的寄存器, cache (按内容存取), 内存 (主存), 外存, 容量越大, 速度越慢。

### 2.4.1 Cache

提高性能依据的是程序的局部性原理。读取时系统的平均周期 = 命中率 \* cache 周期时间 + (1-命中率) \* 主存周期时间

直接映像，全相联映像，组相联映像

## 2.4.2 内存

随机存取存储器 RAM，只读存储器 ROM

AC000H 到 C7FFFH 共多少 K 个地址单元？ $(C7FFFH + 1H - AC000H) / 2^{10} = (1C000H) / 4 / 16 / 16 = 28 * 4 = 112$ ；内存地址按字（16bit）编址，28 片存储芯片组成，每片 16k 个存储单元，则每个存储单元存多少位？ $112K * 16bit = 28 * 16 * x$ ； $x = 4bit$

磁盘存取时间 = 寻道时间 + 等待时间（平均定位时间 + 转动延迟时间）

总线内部系统：数据，地址，控制外部

系统可靠性：串联： $\lambda = \sum \lambda_i$  并联： $\mu = \frac{\lambda}{\sum_{j=1}^n \frac{1}{j}}$  模冗余系统，m 个子系统同时计算，结果利用表决器屏蔽错误。

## 2.5 加密和认证

HTTPS 用 ssl 协议对报文进行封装

对称加密，如 DES, 3DES 或 TDEA, RC5, IDEA, AES，适合大量明文传输；非对称加密，如 RSA；DES 是共享密钥加密。

信息摘要算法：SHA-1, MD5 MD5，信息以 512 位分组，结果 128 位。

CA 的公钥是验证 CA 签名的依据，所以不同 CA 互换公钥是用户互信的必要条件

采样频率大于等于工作频率的 2 倍，能够恢复实际波形。

## 3 编译原理

【词法分析】根据词法规则，对源程序逐个字符扫描，识别出一个个“单词”符号，词汇检查。【语法分析】根据语法规则将单词符号序列分解成各类语法单位，如“表达式”“语句”和“程序”等。语法规则就是各类语法单位的构成规则，主要针对结构的检查。【语义分析】分析各语法结构的含义，检查源程序是否包含语义错误，主要针对句子含义的检查。

## 4 操作系统

### 4.1 进程管理

前驱图

进程的同步与异步；互斥与共享。单缓冲区，多缓冲区，同步与互斥问题。

#### 4.1.1 PV 操作

临界资源：进程间需要互斥方式进行共享的资源。临界区：访问临界资源的那段代码。信号量：一种变量，如 P(s) 中的 s P 操作， $S=S-1$ ； $S<0$ ，阻塞；V 操作， $S=S+1$ ； $s\leq 0$ ，唤醒。

不死锁：资源数 = 并发进程数 \* (进程资源数-1) + 1

死锁的条件：互斥保持、等待不剥夺环路等待

死锁避免：有序资源分配；银行家算法：银行放贷的思想，评估无能力偿还就不分配。需求量不超过资源数时接纳该进程；进程可分期请求资源；资源不满足时，推迟分配。

## 4.2 存储管理

申请内存时的分配方法：首次适应法，选择地址最小的足够大的块；最佳适应法，选择块最小的满足的块。（缺点是导致碎片小）最差适应法，选择块最大的满足的块，增大碎片。循环首次适应，内存块首地址每次查找时移动一个。

#### 4.2.1 页式存储

页表：用户程序的所在页号，与内存中的块号（物理地址）之间关系表。逻辑地址前 k 位为页号，根据页表查找块号，块号和逻辑地址剩余的位组成物理地址。

#### 4.2.2 段式存储

有效地址 = 段号 + 位移量。根据段号查找段长和基址，基址 + 位移量得到物理地址。

各段长度可变，内存碎片浪费大，多程序共享内存。

#### 4.2.3 页面置换算法

最优 (OPT) 算法随机算法

先进先出 FIFO 算法，可能产生抖动，抖动意思是增加资源反而效率更低。

最近最少使用 LRU 算法，不会抖动

## 4.3 文件管理

#### 4.3.1 索引文件结构

索引节点，直接盘块与一级、二级、三级盘块地址复合组成。

#### 4.3.2 文件和树形目录结构

绝对路径，从根开始的路径，盘符开始；相对路径。例如固定电话的区号。

### 4.3.3 空闲存储空间管理

空闲区表法（空闲文件目录）空闲链表法位示图法【重点】成组链接法

## 4.4 设备管理

虚拟设备与 Spooling 技术：例如 4 个人要打印，打印设备先将打印内容存放到缓冲区，顺次打印缓冲区的内容。

## 4.5 微内核操作系统

只实现客户进程、进程服务器、终端服务器、文件服务器等基本功能；图形系统、文件系统、设备驱动、通讯功能放在内核之外。

分为用户态、核心态，核心态响应客户进程的请求，回答文件服务器。

嵌入式操作系统的特点：（1）微型化，从性能和成本角度考虑，希望占用的资源和系统代码量少；（2）可定制，从减少成本和缩短研发周期考虑，要求嵌入式操作系统能运行在不同的微处理器平台上，能针对硬件变化进行结构与功能上的配置，以满足不同应用的需求；（3）实时性，嵌入式操作系统主要应用于过程控制、数据采集、传输通信、多媒体信息及关键要害领域需要迅速响应的场合，所以对实时性要求较高；（4）可靠性，系统构件、模块和体系结构必须达到应有的可靠性，对关键要害应用还要提供容错和防故障措施；（5）易移植性，为了提高系统的易移植性，通常采用硬件抽象层和板级支撑包的底层设计技术。

## 5 软件工程与结构化开发

不同模块中相同程序块提取出来，块内语句没有关系，这属于巧合内聚，影响模块之间的耦合关系。

支持共同代码拥有和共同对系统负责是 XP 中的代码共享（代码集体所有权）

### 5.1 设计各阶段

接口设计主要基于需求分析阶段的数据流图

架构设计，定义软件的主要结构元素及之间的关系

设计软件模块结构时，减少高扇出结构可以改进设计质量，将相似功能模块合并不能。

结构化分析输出包括：数据流图，数据字典，加工逻辑。不包括结构图。

测试时，路径覆盖数独立的路径数量，语句覆盖在路径覆盖基础上减掉没有语句的路径。

## 5.2 极限编程（XP）的 12 个最佳实践

1. 现场客户（On-site Customer） XP：要求至少有一名实际的客户代表在整个项目开发周期在现场负责确定需求、回答团队问题以及编写功能验收测试。 评述：现场用户可以从一定程度上解决项目团队与客户沟通不畅的问题，但是对于国内用户来讲，目前阶段还不能保证有一定技术层次的客户常驻开发现场。解决问题的方法有两种：一是可以采用在客户那里现场开发的方式；二是采用有效的沟通方式。 项目：首先，我们在项目合同签署前，向客户进行项目开发方法论的介绍，使得客户清楚项目开发的阶段、各个阶段要发布的成果以及需要客户提供的支持等；其次，由项目经理每周向客户汇报项目的进展情况，提供目前发布版本的位置，并提示客户系统相应的反馈与支持。

2. 代码规范（Code Standards） XP：强调通过指定严格的代码规范来进行沟通，尽可能减少不必要的文档。 评述：XP 对于代码规范的实践，具有双重含义：一是希望通过建立统一的代码规范，来加强开发人员之间的沟通，同时为代码走查提供了一定的标准；二是希望减少项目开发过程中的文档，XP 认为代码是最好的文档。对于目前国内的大多数项目团队来说，建立有效的代码规范，加强团队内代码的统一性，是理所当然的；但是，认为代码可以代替文档却是不可取的，因为代码的可读性与规范的文档相比合适由一定的差距。同时，如果没有统一的代码规范，代码全体拥有就无从谈起。 项目：在项目实施初期，就由项目的技术经理建立代码规范，并将其作为代码审查的标准。

3. 每周 40 小时工作制（40-hour Week） XP：要求项目团队人员每周工作时间不能超过 40 小时，加班不得连续超过两周，否则反而会影响生产率。 评述：该实践充分体现了 XP 的“以人为本”的原则。但是，如果要真正的实施下去，对于项目进度和工作量合理安排的要求就比较高。 项目：由于项目的工期比较充裕，因此，很幸运的是我们并没有违反该实践。

4. 计划博弈（Planning Game） XP：要求结合项目进展和技术情况，确定下一阶段要开发与发布的系统范围。 评述：项目的计划在建立起来以后，需要根据项目的进展来进行调整，一成不变的计划是不存在。因此，项目团队需要控制风险、预见变化，从而制定有效、可行的项目计划。 项目：在系统实现前，我们首先按照需求的优先级做了迭代周期的划分，将高风险的需求优先实现；同时，项目团队每天早晨参加一个 15 分钟的项目会议，确定当天以及目前迭代周期中每个成员要完成的任务。

5. 系统隐喻（System Metaphor） XP：通过隐喻来描述系统如何运作、新的功能以何种方式加入到系统。它通常包含了一些可以参照和比较的类和设计模式。XP 不需要事先进行详细的架构设计。 评述：XP 在系统实现初期不需要进行详细的架构设计，而是在迭代周期中不断的细化架构。对于小型的系统或者架构设计的分析会推迟整个项目的计划的情况下，逐步细化系统架构倒是可以的；但是，对于大型系统或者是希望采用新架构的系统，就需要在项目初期进行相信的系统架构设计，并在第一个迭代周期中进行验证，同时在后续迭代周期中逐步进行细化。

项目：开发团队在设计初期，决定参照 STRUTS 框架，结合项目的情况，构建了针对工作流程处理的项目框架。首先，团队决定在第一个迭代周期实现配件申请的工作流程，在实际项目开发中验证了基本的程序框架；而后，又在其它迭代周期中，对框架逐渐精化。

6. 简单设计（Simple Design）

XP：认为代码的设计应该尽可能的简单，只要满足当前功能的要求，不多也不少。 评述：

传统的软件开发过程，对于设计是自顶而下的，强调设计先行，在代码开始编写之前，要有一个完美的设计模型。它的前提是需求不变化，或者很少变化；而 XP 认为需求是会经常变化的，因此设计不能一蹴而就，而应该是一项持续进行的过程。

Kent Beck 认为对于 XP 来说，简单设计应该满足以下几个原则：A. 成功执行所有的测试；B. 不包含重复的代码；C. 向所有的开发人员清晰地描述编码以及其内在关系；D. 尽可能包含最少的类与方法。

对于国内大部分的软件开发组织来说，应该首先确定一个灵活的系统架构，而后在每个迭代周期的设计阶段可以采用 XP 的简单设计原则，将设计进行到底。

项目：在项目的系统架构经过验证后的迭代周期内，我们始终坚持简单设计的原则，并按照 Kent Beck 的四项原则来进行有效的验证。对于新的迭代周期中出现需要修改既有设计与代码的情况，首先对原有系统进行“代码重构”，而后再增加新的功能。

7. 测试驱动 ( Test-driven ) XP：强调“测试先行”。在编码开始之前，首先将测试写好，而后再进行编码，直至所有的测试都得以通过。

评述：RUP 与 XP 对测试都是非常的重视，只是两者对于测试在整个项目开发周期内首先出现的位置处理不同。XP 是一项测试驱动的软件开发过程，它认为测试先行使得开发人员对自己的代码有足够的信心，同时也有勇气进行代码重构。测试应该实现一定的自动化，同时能够清晰的给出测试成功或者失败的结果。在这方面，xUnit 测试框架做了很多的工作，因此很多实施 XP 的团队，都采用它们进行测试工作。

项目：我们在项目初期就对 JUNIT 进行了一定的研究工作，在项目编码中，采用 JBUILD6 提供的测试框架进行测试类的编写。但是，不是对所有的方法与用例都编写，而只是针对关键方法类、重要业务逻辑处理类等进行。详细的关于 JUNIT 测试框架的使用，请参见我的同事撰写的另一篇文章 (<http://www-900.ibm.com/developerWorks/cn/java/l-junit/index.shtml>)

8. 代码重构 ( Refactoring ) XP：强调代码重构在其中的作用，认为开发人员应该经常进行重构，通常有两个关键点应该进行重构：对于一个功能实现和实现后。

评述：代码重构是指在不改变系统行为的前提下，重新调整、优化系统的内部结构以减少复杂性、消除冗余、增加灵活性和提高性能。重构不是 XP 所特有的行为，在任何的开发过程中都可能并且应该发生。在使用代码重构的时候要注意，不要过分的依赖重构，甚至轻视设计，否则，对于大中型的系统而言，将设计推迟或者干脆不作设计，会造成一场灾难。

项目：我们在项目中将 JREFACTORY 工具部署到 JBuilder 中进行代码的重构，重构的时间是在各个迭代周期的前后。代码重构在项目中的作用是改善既有设计，而不是代替设计。

9. 成对编程 ( Pair Programming ) XP：认为在项目中采用成对编程比独自编程更加有效。成对编程是由两个开发人员在同一台电脑上共同编写解决同一问题的代码，通常一个人负责写编码，而另一个负责保证代码的正确性与可读性。

评述：其实，成对编程是一种非正式的同级评审 ( Peer Review )。它要求成对编程的两个开发人员在性格和技能上应该相互匹配，目前在国内还不是十分适合推广。成对编程只是加强开发人员沟通与评审的一种方式，而非唯一的方式。具体的方式可以结合项目的情况进行。

项目：我们在项目中并没有采用成对编程的实践，而是在项目实施的各个阶段，加强了走查以及同级评审的力度。需求获取、设计与分析都有多人参与，在成果提交后，交叉进行走查；而在编码阶段，开发人员之间也要在每个迭代周期后进行同时评审。

10. 集体代码所有制 ( Collective Ownership )

XP: 认为开发小组的每个成员都有更改代码的权利, 所有的人对于全部代码负责。

评论: 代码全体拥有并不意味着开发人员可以互相推委责任, 而是强调所有的人都要负责。如果一个开发人员的代码有错误, 另外一个开发人员也可以进行 BUG 的修复。在目前, 国内的软件开发组织, 可以在一定程度上实施该实践, 但是同时需要注意一定要有严格的代码控制管理。

项目: 我们在项目开发初期, 首先向开发团队进行”代码全体拥有”的教育, 同时要求开发人员不仅要了解系统的架构、自己的代码, 同时也要了解其它开发人员的工作以及代码情况。这个实践与同级评审有一定的互补作用, 从而保证人员的变动不会对项目的进度造成很大的影响。

在项目执行中, 有一个开发人员由于参加培训, 缺席项目执行一周, 由于实行了”代码全体拥有”的实践, 其它的开发人员成功地分担了该成员的测试与开发任务, 从而保证项目的如期交付。

11. 持续集成 ( Continuous Integration )      XP: 提倡在一天中集成系统多次, 而且随着需求的变化, 要不断的进行回归测试。因为, 这样可以使得团队保持一个较高的开发速度, 同时避免了一次系统集成的恶梦。      评述: 持续集成也不是 XP 专有的最佳实践, 著名的微软公司就有每日集成 ( Daily Build ) 的成功实践。但是, 要注意的是, 持续集成也需要良好的软件配置变更管理系统的有效支持。      项目: 使用 VSS 作为软件配置管理系统, 坚持每天进行一次的系统集成, 将已经完成的功能有效地结合起来, 进行测试。

12. 小型发布 ( Small Release )      XP: 强调在非常短的周期内以递增的方式发布新版本, 从而可以很容易地估计每个迭代周期的进度, 便于控制工作量和风险; 同时, 也可以及时处理用户的反馈。      评论: 小型发布突出体现了敏捷方法的优点。RUP 强调迭代式的开发, 对于系统的发布并没有作出过多的规定。用户在提交需求后, 只有在部署时才能看到真正的系统, 这样就不利于迅速获得用户的反馈。如果能够保证测试先行、代码重构、持续集成等最佳实践, 实现小型发布也不是一件困难的事情, 在有条件的组织可以考虑使用。      项目: 项目在筹备阶段就配置了一台测试与发布服务器, 在项目实施过程中, 平均每两周 (一个迭代周期结束后) 进行一个小型发布; 用户在发布后两个工作日内, 向项目小组提交”用户接收测试报告”, 由项目经理评估测试报告, 将有效的 BUG 提交至 Rational Clear Case, 并分配给相应的开发人员。项目小组应该在下一个迭代周期结束前修复所有用户提交的问题。

### 5.3 E-R 图

父图与子图平衡: 父某个加工的输入输出流, 与子的, 在数量和名称上相同; 父的一个流对于子图一个或多个流, 这些组合起来正好是父的。

## 6 面向对象

### 6.1 UML

通信图中, : 冒号前面是对象名, 后面是类名, 之间连线上是消息。

模块结构图的主要组成有: 模块、调用、数据、控制信息和转接符号



观察者模式：具体的观察者将自己注册到事件，具体的事件知道自己的观察者、体现类对扩展开放，对修改关闭。

## 7 算法设计与分析

### 7.1 数据结构

#### 7.1.1 数组

$a[i] = a + i * \text{len}$  ; //  $i$  from 0;  $a[i][j]$  按行存:  $a + (i * \text{len} + j) * \text{len}$  ;  $a[i][j]$  按列存:  $a + (j * \text{len} + i) * \text{len}$  ;

5\*5 的二维数组  $a$ , 各元素 2 字节,  $a[2][3]$  行有限存, 地址?  $2*5+3 = 13$ ,  $13*2 = 26$ ,  $a+26$ ;

#### 7.1.2 线性表

#### 7.1.3 树与二叉树

二叉树遍历：前序、后续、中序。反向构造二叉树：利用前 + 中，或后 + 中遍历结果，推出树的结构。只利用前 + 后不行。

树转二叉树：第一个孩子在左，兄弟都是右。

查找二叉树：左 < 根 < 右

最优二叉树、哈夫曼树：带权路径长度最小。1,2,8,4 构造哈夫曼树：step1: 1,2->3; 3,8,4; step2: 3,4->7;7,8 so: 15 7 8 3 4 1 2 权值： $1*3+2*3+4*2+8*1 = 25$

线索二叉树：前序、后续、中序，列举各元素后，叶子 LR 指针指向前后元素。

平衡二叉树：任意结点左子树与右子树深度差不大于 1。平衡度 = 左子树深度-右子树深度。

#### 7.1.4 图

有向图，无向图。完全图。

存储：邻接矩阵。 $n$  个点， $n*n$ 。 $i$  到  $j$  有邻接边， $R_{ij}=1$ ，否则为 0。邻接表， $V1 \rightarrow [2,6, \dots] \rightarrow [4,1, \dots] \rightarrow [6,50, \dots]$  //  $V1$  到 2 号结点距离 6，到 4 号结点距离 1，到 6 号结点距离 50

【遍历】深度优先，广度优先。

【拓扑排序，AOV 网络】有向边表示活动之间开始的先后关系。

【图的最小生成树，普里姆算法】留下的权值最小。树没有环路， $n$  个节点的树边最多  $n-1$  个。染色红，逐个收集最短的一个元素进来。注意过程中不能形成环。

【图的最小生成树，克鲁斯卡尔算法】从最短的边开始选边。

## 7.2 算法

又穷，确定，有效。

【复杂度】时间，空间

时间复杂度： $1, \log_2 n, n, n \log_2 n, n^2, n^3, \dots, 2^n$

### 7.2.1 查找

【顺序查找】平均查找长度： $\frac{n+1}{2}$  time,  $O(N)$

【二分查找】有序排列。比较次数最多  $\lfloor \log_2 n \rfloor + 1$  time,  $O(\log_2 n)$

【散列表】例如，存储空间 10， $p=5$ ，散列函数  $h = key \% p$ ，存储 3,8,12,17,9：线性探测：3,4,2, 5,6 冲突解决：线性探测，伪随机数。

### 7.2.2 排序

稳定、不稳定。【一样的数，保持原顺序，叫稳定】

【插入式：直接插入】新的一个与已经排好的比，插入到位置

【插入式：希尔】数据少时插入排序效率可以。

例如 10 个元素，先  $d=n/2=5$ ，隔 5 个一组，插入排序； $d = d/2 = 2$ ，取奇数是 3；隔 3 个一组，插入； $d = d/2 = 1$ ，全体插入排序。

【选择式：直接选择】每次选剩余最小的。

【选择式：堆排序】完全二叉树。。小顶堆： $k_i \leq k_{2i}, k_i \leq k_{2i+1}$  大顶堆： $k_i \geq k_{2i}, k_i \geq k_{2i+1}$  所有孩子都更小

从小到大排列：建小顶堆—》取顶—》建小顶堆—》。。。。

例如构造大顶堆：step1，数组顺序构造完全二叉树。step2，最后一个非叶子节点，与其 2 个孩子调整为大顶堆；倒数第 2 个非叶子节点，依次调整。如果有子树，要调整后继续调整子树。

【交换式：冒泡】

【交换式：快速】

【归并排序】

【基数排序】

## 8 数据库

B-S 体系中，系统安装、修改、维护都只在服务器端。

### 8.1 三级模式

物理数据库（对应一个文件）《-》内模式（数据的存储形式）《-》概念模式（数据划分为表）《-》外模式（各种用户视图）《-》用户

## 8.2 数据库设计

需求分析：数据处理要求、应用要求，输出数据流图、数据字典、数据说明书；

概念结构设计，输出 ER 模型【重点】，用户数据模型；

逻辑结构设计：转换规则、规范化理论，输出关系模式，视图、完整性约束、应用处理说明书；

物理设计：结合 DBMS 特性、硬件特性、OS 特性等。

### 8.2.1 ER 模型

实体，方框表示，如学生、课程；属性，椭圆表示，如学号、姓名；联系，菱形，如学生—M-选课-N—课程。

集成：画局部，合成整体的。集成会存在属性冲突、命名冲突、结构冲突等。

关系 O 对应的三个实体之间的联系数量分别为 A,B,C，则最少转换为 4 个关系模式（每个实体 1 个，关系至少一个）。

### 8.2.2 关系代数

关系代数表达式的等价转换；业务场景对应的关系代数表达式。

并 交 差 笛卡尔积 投影 选择 连接

## 8.3 规范化理论

解决数据冗余、更新异常、插入异常、删除异常等。

### 8.3.1 函数依赖

部分函数依赖：如当主键为（学号，课程）的复合键时，可以确定姓名；但是主键的一部分-学号，可以确定姓名。传递函数依赖：A->B->C，有 A->C

### 8.3.2 键

超键：唯一标识元组，可以存在冗余属性。候选键：唯一标识元组，不存在冗余属性。主键：任选一个候选键。外键：其他关系的主键。关联各表。

### 8.3.3 范式

1NF，属性值不可分，原子值；2NF，由 1NF 消除非主属性对候选键的部分依赖得到；3NF，由 2NF 消除非主属性对候选键的传递依赖得到；BCNF，由 3NF 消除主属性对候选键的传递依赖得到。越高级，数据表拆分越细。粒度小，性能下降。

#### 8.3.4 模式分解

表的拆分。保持函数依赖：冗余的依赖不需要保留。

无损分解：可以还原。

### 8.4 并发

事务机制：一系列操作作为一个整体，原子性、一致性、隔离性、持续性。

PV 操作可实现资源互斥使用。

#### 8.4.1 并发问题

丢失更新：T1 与 T2 都读取 A，分别将 A-5，写回。有可能某个进程后写回导致 A 只是-5

不可重读读取：T1 读取数据计算后，在 T1 校验计算结果时，T2 进程修改数据，导致 T1 校验失败。

错误数据读出：T1 修改了数据，T2 读到了修改的数据，T1 进行回滚恢复了数据，导致 T2 的数据无意义。

#### 8.4.2 封锁协议

S 封锁，X 封锁。一级：事务 T 修改数据 R 之间必须加 X 锁。防止丢失修改。二级：一级，且 T 读取 R 之间 S 锁，读完释放 S 锁，防止丢失修改、读脏数据。三级：一级，且 T 读取 R 之间 S 锁，事务完成再释放 S 锁，防止丢失修改、读脏数据、数据重复读。两段锁协议：可串行，可能死锁。

更新 = 读 + 写。

### 8.5 其他

#### 8.5.1 完整性约束

提高数据可靠性。

实体完整性约束：主键不能为空、重复。参照完整性约束：外键，如员工的部门，应该是对应的部门表中的主键或还未分配部门时空。用户自定义完整性约束：如性别、年龄的限制

#### 8.5.2 安全

用户标识和鉴定，账户、校验存取控制，用户的操作权限密码存储和传输；视图和保护，视图授权；审计，记录用户对数据库的操作。

### 8.5.3 数据备份

静态备份，关闭数据库，复制。快速，易归档；动态备份，运行时复制。有选择性备份、恢复某个表，灵活，出错导致问题更大。

备份策略，如连续 7 天：完增增增差增增完全备份差量备份，与上次完全备份的差异增量备份，与上次备份的差异。

海量，转储，

增删改查，先写日志文件，然后再实际处理。

### 8.5.4 故障与恢复

可预期故障：程序中预先设置 rollback 语句；不可预期：如算术溢出、违反存储保护等，可以由 DBMS 的恢复子系统通过日志，撤销对数据库的修改；系统故障：检查点法；介质故障：使用日志。

### 8.5.5 数据仓库

数据仓库特点：面向主题，而一般数据库面向业务组织数据。集成的，相对稳定的，一般不修改与删除。反映历史变化

数据源进行抽取、清理、装载、刷新，得到数据仓库。OLAP 服务器，提供查询、报表、分析、数据挖掘等工具。部门级的数据仓库整合，得到企业级的数据仓库。

### 8.5.6 数据挖掘

方法：决策树、神经网络、遗传算法、关联规则挖掘算法

分类：关联分析，数据之间的隐藏关系分析；序列模式分析，分析数据之间的因果关系；分类分析，记录做标记，按标记分类；聚类分析

### 8.5.7 反规范化技术

规范化程度低，存在冗余；高了导致数据表多，查询效率低。

牺牲空间和规范化程度来提高查询效率。

技术：增加派生性冗余列；增加派生性冗余列；重新组织表；分割表

### 8.5.8 大数据

多种不同类型进行联合分析。

数据量大，PB 级或以上需要快速处理数据多样性数据有价值  
cookies 售卖；百度广告推送。

深度分析，关联分析，回归分析

集群平台。

大数据处理系统：高可扩展性；高性能；高度容错；支持异构环境；分析延迟小；开放接口易用；成本低；向下兼容。

## 9 网络信息与安全

局域网在最低两层。

### 9.1 OSI、RM 七层模型

二层以上通过协议保障安全。

物理层，二进制传输，中继器、集线器；【安全上，隔离，屏蔽】数据链路层，以帧为单位，网桥、交换机、网卡；【安全上，链路加密，PPTP，L2TP】【这两层：以太网，ATM，帧中继等】

网络层，分组传输与路由选择，三层交换机、路由器、IP【安全上，防火墙，IPSec】【IP，ICMP，IGMP，等】

传输层，端到端。【安全上，TLS,SET, SSL】【TCP,UDP】【UDP 没有验证】

会话层，建立、管理、终止会话；表示层，数据格式表示、加密、压缩；应用层，具体实现功能。【这 3 层，DHCP 等】【安全上，PGP，Https，SLL】

### 9.2 网络技术标准与协议

TCP-IP 协议族，Internet，可扩展，可靠；IPX-SPX 协议，局域网即时战略游戏等，路由；NETBEUI 协议，不支持路由，IBM，快速

#### 9.2.1 常见协议功能

RIP(Routing Information Protocol, 路由信息协议) 是使用最久的协议之一。RIP 是一种分布式的基于距离向量的路由选择协议，RIP 协议是施乐公司 20 世纪 80 年代推出的，主要适用于小规模的网络环境。RIP 协议主要用于一个 AS(自治系统) 内的路由信息的传递

OSPF 路由协议是用于网际协议 (IP) 网络的链路状态路由协议。该协议使用链路状态路由算法的内部网关协议 (IGP)，在单一自治系统 (AS) 内部工作。适用于 IPv4 的 OSPFv2 协议定义于 RFC 2328，RFC 5340 定义了适用于 IPv6 的 OSPFv3。

POP3: 邮件收取

SMTP: 邮件发送

FTP: 20 数据端口/21 控制端口，文件传输协议

HTTP: 超文本传输协议，网页传输

DHCP: IP 地址自动分配

SNMP: 简单网络管理协议

DNS: 域名解析协议, 记录域名与 IP 的映射关系

TCP: 可靠的传输层协议

UDP: 不可靠的传输层协议

ICMP: 因特网控制协议, PING 命令来自该协议

IGMP: 组播协议

ARP: 地址解析协议, IP 地址转换为 MAC 地址

RARP: 反向地址解析协议, MAC 地址转 IP 地址

BCP (边界网关协议) 是运行于 TCP 上的一种自治系统的编由协议。

### 9.2.2 TCP 三次握手

A 给 B, B 给 A, A 给 B。保证传输可靠

### 9.2.3 DHCP 协议

动态分配 IP 地址。169.254.0.0 和 0.0.0.0 是假的。租约 8 天。

### 9.2.4 DNS 协议

域名与 IP 转换。迭代查询: 可以丢给别人【根域名服务器—》顶级域名服务器—》权限域名服务器—》。。。】, 直接给出反馈, 不盘根究底。递归查询: 最终答案【本地域名服务器】

## 9.3 子网划分

网络前缀 + 主机号。B 类 168.195.0.0 分 27 个子网, 子网掩码? 10101000, 因为  $27 < 2^5$ , 所以 255,255,11111000,0 所以 255,255,248,0

B 类 168.195.0.0 划分子网, 每个子网主机 700, 求掩码?  $2^k - 2 \geq 700$ , 所以 10 位地址, 所以 255,255,252,0

210.115.192.0/20, 可划分多少 c 类子网? 因为后面  $20-16=4$ , 中间还有  $2^4 = 16$  个

## 9.4 无线网

局域, WLAN, WIFI 城域, WMAN, WIMAX 广域, WWAN, 3G, 4G 个人, WPAN, bluetooth

## 9.5 网络接入

有线: 公用交换电话 PSTN 数字数据网 DDN ISDN, 可以打电话时上网; ADSL, 非对称数字用户, 电话线通信, 下行 8M, 上行 512K; 同轴光纤, HFC, 上行下行对称。

无线: WIFI, BLUETOOTH, 红外 IrDA, WAPI; 3G, 4G

## 9.6 系统安全分析

保密性，【最小授权，防暴露，信息加密，物理保密】完整性，【安全协议，校验码，密码校验，数字签名，公证】可用性，【IP 过滤，路由选择控制】不可抵赖，【数字签名】

安全的五个基本要素 机密性 (确保信息不暴露给未授权的实体或进程) 完整性 (只有得到允许的人才能修改数据，并能够判别数据是否已被篡改) 可用性 (得到授权的实体在需要时可访问数据) 可控性 (可以控制授权范围内的信息流向和行为方式) 可审查性 (对出现的安全问题提供调查的依据和手段)

### 9.6.1 加密

对称，【加密和解密密钥一样】【加密强度低，密钥分发困难】DES, 替换 + 移位，速度快；3DES, 56 位的 K1 和 K2, K1 加-K2 解-K1 加 AES, RC-5, IDEA,

非对称，【加密和解密密钥不一样】【加密速度慢】RSA, Elgamal, 基础是 Diffie-Hellman 密钥交换算法；ECC, 背包算法, Rabin, D-H 等

### 9.6.2 摘要

单向散列函数，单向 Hash 函数，定长的散列值。MD5, SHA, SHA 更长更安全。

### 9.6.3 数字签名

A: 我的名字-》信息摘要-》我的私钥加密得到签名; 对方: 1) 收到明文名字-》信息摘要; 【数字签名, 识别身份的作用】2) 收到的签名, 用 A 的公钥解密, 得到信息摘要; 【验证】3) 比较上述两个摘要是否相等。

### 9.6.4 数字信封与 PGP

A: 原文, 对称加密; 密钥用 B 的公钥加密后发送给 B。B: 收到电子信封, 用私钥解密信封, 去除密钥解密出原文。

密钥用加密时间长的复杂的非对称加密。

PGP 证书, 是电子邮件、文件存储加密。可以将文件用 PGP 加密后存到云盘, 更安全。

数字证书: 密钥与数字签名结合在一起。CA 机构颁发。验证数字证书上颁发机构的签名。

【试设计】邮件要加密传输, 最大附件 500M, 发送者不可抵赖, 第三方截获的话无法篡改。

发送端 A: 邮件正文-》随机密钥 K, 对称加密-》邮件密文; 邮件正文-》信息摘要-》数字签名 (私钥)-》摘要密文; 密钥 K-》数字信封技术, 非对称加密 (公钥)-》信封;

接收端 B: 信封-》非对称加密 (私钥)-》密钥 K 邮件密文-》随机密钥 K, 对称加密-》邮件正文; 邮件正文-》邮件摘要摘要密文-》解密签名 (公钥)-》邮件摘要, 与上一步的摘要验证;



#### 9.6.5 病毒

引导区：。主引导记录病毒感染硬盘的主引导区，如大麻病毒、2708 病毒、火炬病毒等；分区引导记录病毒感染硬盘的活动分区引导记录，如小球病毒、Girl 病毒等。宏：TaiwanNo.1, Nuclear 宏病毒木马：冰河，ICMP 类型的木马，灰鸽子和蜜蜂大盗，PassCopy 和暗黑蜘蛛侠蠕虫：震网（Stuxnet）

## 10 标准化和知识产权

商标权是可以续住来长期拥有的知识产权。

## 11 其他

MIME 它是一个互联网标准，扩展了电子邮件标准，使其能够支持，与安全无关。