目 录

考试介绍 1
企业信息化战略与实施 9
软件工程 22
软件架构设计 62
系统安全分析与设计 118
系统可靠性分析与设计 127
项目管理 138
计算机组成与体系结构 152
系统配置与性能评价 170
操作系统 174
计算机网络 193
数据库系统 206
数学与经济管理 238
知识产权与标准化 249
系统架构设计案例分析 255
系统架构设计论文 278
练习题 289
论文范文 331
练习题参考答案 336
模拟题 339
模拟题参考答案 351

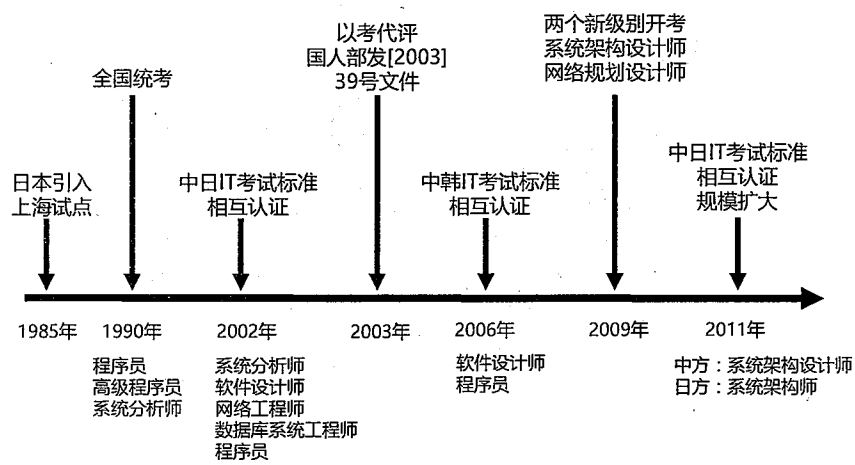
希赛

系统架构设计师
考试介绍

400-111-9811 关注希赛微信

希赛

软考历史



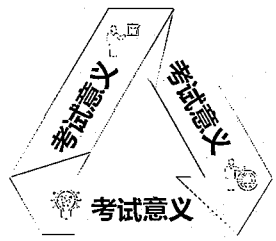
希赛

软考体系

	计算机软 件	计算机网 络	计算机应用技术	信息系统	信息服务
高级资格	信息系统项目管理师　系统分析师　**系统架构设计师**　网络规划设计师 系统规划与管理师				
中级资格	软件评测师 软件设计师 软件过程能力评估师	网络工程师	多媒体应用设计师 嵌入式系统设计师 计算机辅助设计师 电子商务设计师	系统集成项目管理工程师 信息系统监理师 信息安全工程师 数据库系统工程师 信息系统管理工程师	计算机硬件工程师 信息技术支持工程师
初级资格	程序员	网络管理员	多媒体应用制作技术员 电子商务技术员	信息系统运行管理员	网页制作员 信息处理技术员

希赛

考试介绍



01 获取职称：以考代评

02 国际意义：中日韩互认

03 升职加薪
拓展就业方向

希赛

考试介绍

报名
支持网上报名和现场报名

缴费
报名审核通过后，现场或网上缴费

打印准考证
考前一周左右

参加考试
带齐资料，准时参加考试

考试报名时间：根据地区有差异，请关注：

国家软考办官网：http://www.ceiaec.org/

希赛软考报名及查询成绩专题：

http://www.educity.cn/rk/zt/

希赛软考报名费汇总专题（全国各地收费不一）：

https://www.educity.cn/rk/1836633.html

希赛

考试介绍-时间、科目

考试时间

2019年11月9日

考核标准

划分数线方式：

一般情况所有科目达到**45分合格**

科目一：综合知识

客观题：75道　150分钟

科目二：案例分析

主观题：5道大题（只做3道）　90分钟

第1道必答，后4道选答2道

科目三：论文写作

论文题：4道题选做1道　120分钟

希赛

复习策略——总纲领

调整心态，加强训练，相互鼓励，共同提高！

注意事项：

1、以讲义为中心（注意有些内容讲义上有，但教程上没有，一样会考）

2、以《系统架构设计师视频教程》为预习材料

3、通过直播课反思巩固

4、通过习题训练了解细节

5、一般情况把《系统架构设计师教程（第4版）》当字典，这个教程是希赛写的。学完视频课程，可以浏览这个部分的教程，不必全书通读。

希赛

复习策略——综合知识

科目一：综合知识

客观题：75道　150分钟

特点：范围广，但考试时间多得让人犯困。

复习策略

1、各个击破

2、混合练习

3、分层做题，不听天由命

希赛

复习策略——案例分析

科目二：案例分析

主观题：5道大题（只做3道） 90分钟

第1道必答，后4道选答2道

复习策略

1、多练历年试题，熟悉答题规则，习惯“意外”

2、对于技术主题，要自己有意识组织语言进行总结

3、善于顺势而为的解题

4、从容的心态上考场

希赛

复习策略——论文写作

科目三：论文写作

论文题：4道题选做1道 120分钟

复习策略

1、早早的学好写作技巧

2、早早的选好项目

3、学习各项技术时，就有意识的想想这项技术能不能用在我的项目中，如何用

4、论文要准备的是素材，素材是组织成文的构件，构件全了，什么都不是问题

5、提早练练字

6、考场上，心态是自己最大的敌人

希赛

复习策略——论文写作

应用场景的合理性

技术方案的合理性

论文 = 心态 + 技巧 + 理论 + 实践

猜题

沟通的技巧
把问题表达清楚的技巧

希赛

专业介绍 – 综合知识考点分布

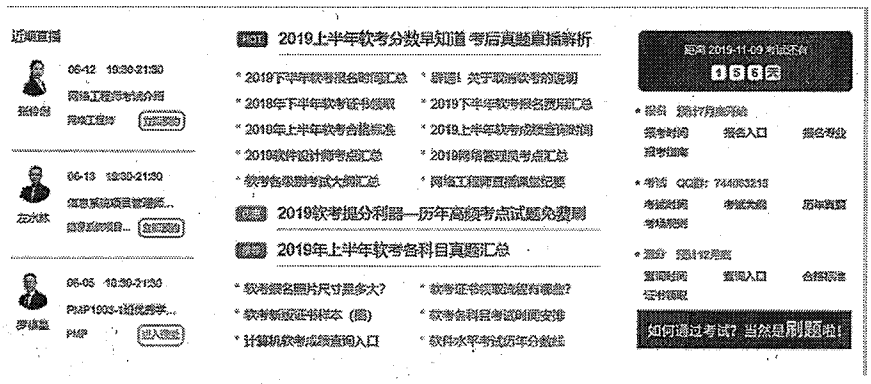
知识点	分数	说明	比例
计算机组成原理与体系结构	2-4	中断、存储器、并/串转换、寄存器、内存容量计算、磁盘文件读取、异步传输、CISC与RISC、Cache特点、虚拟存储器、总线、宿主机	2.7%-5.3%
系统配置与性能评价	0-2	基准程序测试、计算机性能优化/性能评价、负载均衡、数据备份、MIPS	0-2.7%
操作系统	6-8	概念、前趋图与PV操作、微内核、文件系统、页面置换算法、索引文件	8%-10.7%
计算机网络	3-6	网络层次化设计、逻辑/物理网络设计、网络存储、综合布线、DNS服务	4%-8%
数据库系统	4-9	数据库设计、ER模型、完整性约束、触发器、候选关键字、模式分解、分布式数据库、SQL语言、关系代数、数据仓库、范式、事务处理	5.3%-12%
企业信息化战略与实施	3-8	商业智能、企业应用集成、EDI、企业门户、CRM、企业信息化方法、信息化“三流“	4%-10.7%
软件工程	12-18	需求管理、敏捷方法、逆向工程、用例包含关系、UML图、界面设计、系统测试、黑盒测试、CMMI、螺旋模型、软件开发环境、软件重用、RUP、软件维护、RAD、面向对象设计原则	16%-24%
项目管理	0-4	用户文档、配置项、项目管理工具、需求变更	0-5.3%
软件架构设计	22-28	设计模式、基于构件的开发模型、构件技术、MVC模式、架构设计阶段、架构设计基本概念、软件架构需求、ABSD、软件架构文档、架构复审、软件架构风格选择、架构模式、架构评估方法、CORBA、“4+1”视图、架构与质量属性、敏感点与风险、ADL、中间件	29.3%-37.3%
系统安全性与保密性设计	2	安全策略、对称加密算法、网络安全漏洞、ARP、SNMP v3、PGP、PKI	2.7%
法律法规与标准化	3	著作权保护期限、著作权法、侵权判定、标准类型、商标法、商业秘密	4.00%
数学与经济管理	2	数学建模、转移矩阵、盈亏平衡点、线性规划、函数曲线	2.7%
专业英语	5	软件架构风格、软件架构、信息系统设计、DFD	6.67%

希赛

希赛学习平台优势及资源

★ 如何充分利用希赛网辅导平台？

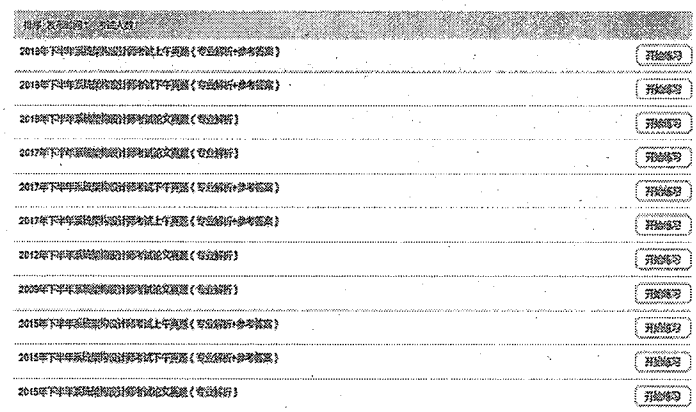
★ 1、通过希赛平台实时掌握考试动态



希赛

希赛学习平台优势及资源

★ 2、希赛平台部分真题免费开放（学员全开放）

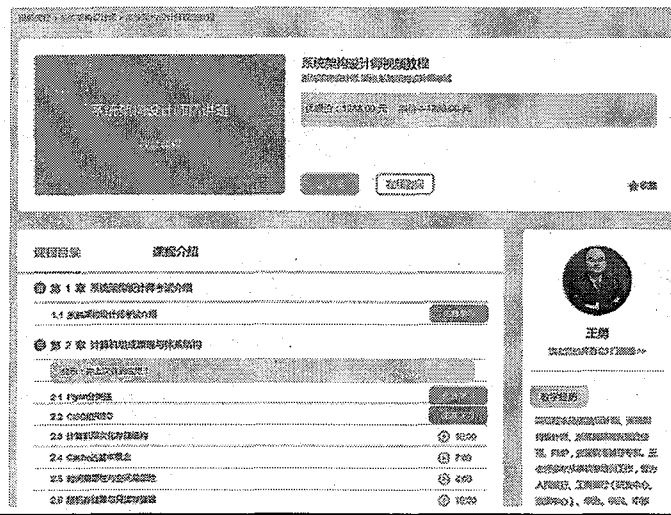


扫码做练习

希赛

希赛学习平台优势及资源

★ 3、希赛网录播短视频及直播课程视频学员均不限次观看

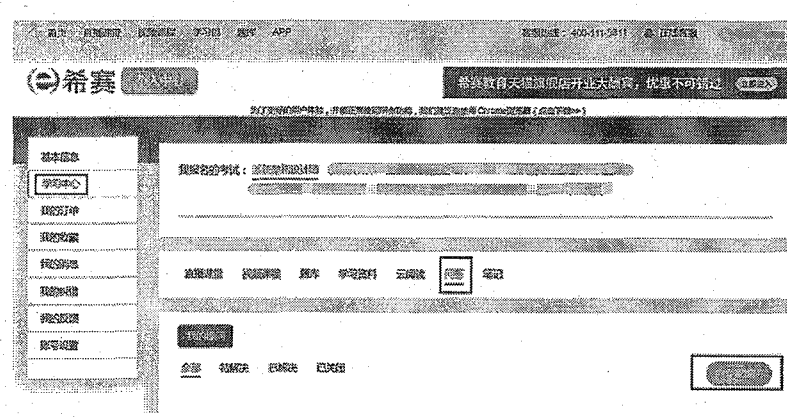


扫码看视频

希赛

希赛学习平台优势及资源

★ 4、希赛提供实时提问平台老师及时答疑



希赛

企业信息化战略与实施

希赛

课程内容提要

- 信息与信息化的概念（★）
- 信息系统生命周期（★）
- 信息系统战略规划（★★★）
- 政府信息化与电子政务（★）
- 企业信息化与电子商务（★★★）
- 企业门户（★★）
- 企业应用集成（★★★）

希赛

一 信息的概念及特点

信息的定义

- 香农：信息就是不确定性的减少。
- 维纳：信息就是信息，既不是物质，也不是能量。

信息的特点

- 客观性（真伪性）：也叫事实性，不符合事实的信息不仅没价值，还有副作用。
- 动态性：信息随着时间的变化而变化。
- 层次性：信息可分为战略级、管理级和操作级。
- 传递性：信息在时间上的传递即是存储；在空间上的传递即是转移或扩散。
- 滞后性：信息是数据加工后的产物，所以相对数据有滞后性。
- 扩压性：信息和实物不同，它可以扩散也可以压缩。
- 分享性：信息可以分享，这和物质不同，并且信息分享具有非零和性。

希赛

一 信息化及其意义

信息化的概念

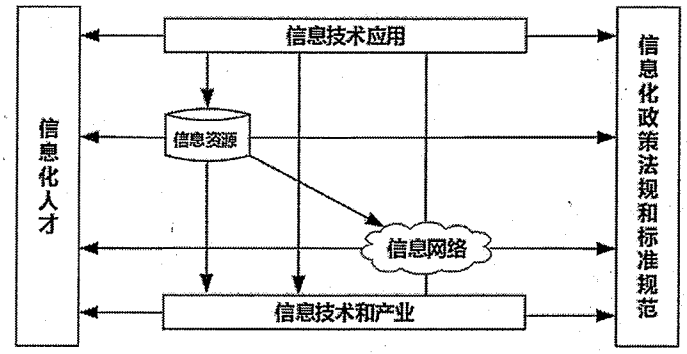
- 信息化是从工业社会到信息社会的演进与变革。
- 信息化的主体是全体社会成员（政府、企业、团体和个人），时域是一个长期过程，空域是经济和社会的一切领域，手段是先进社会生产工具。

信息化对组织的意义

- 组织的结构创新：如虚拟企业、虚拟社区。
- 组织的管理创新：实施电子政务、建立电子政府、建立电子社区以提升管理水平。
- 组织经营创新：利用信息化将经营和管理融为一体。
- 造就信息化的人才队伍：三类人才（IT专业人才，业务人才，专家型人才）。

希赛

一 信息化标准、法律和规定



希赛

信息系统战略规划 – 方法

第一阶段
以数据处理为核心
围绕职能部门需求

企业系统规划法（BSP）：自上而下识别系统目标，自下而上设计信息系统，对组织机构的变动具有适应性。

关键成功因素法（CSF）：找实现目标的关键信息集合，从而确定开发优先次序。

战略集合转化法（SST）：把战略目标看成“信息集合”，把战略目标转变成信息系统的战略目标。

第二阶段
以企业内部MIS为核心
围绕企业整体需求

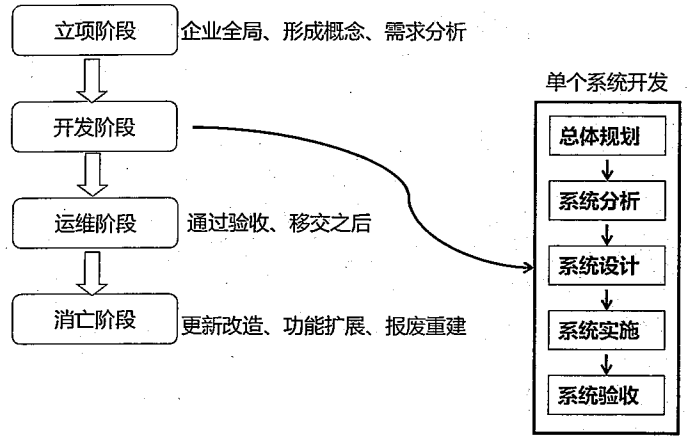
战略数据规划法（SDP）、信息工程法（IE）、战略栅格法（SG）

第三阶段
综合考虑企业内外环境
以集成为核心，围绕企业战略需求

价值链分析法（VCA）、战略一致性模型（SAM）

希赛

一 信息系统的生命周期



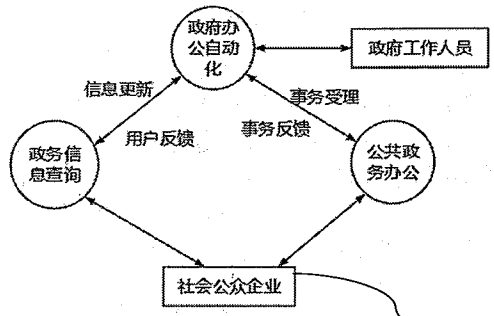
希赛

一 信息系统的生命周期

系统规划	初步调查、分析系统目标、子系统组成、拟实施方案、可行性研究、制定系统建设方案	系统设计任务书（系统建设方案、实施计划）
系统分析	业务流程分析、数据与数据流程分析、软件需求分析、网络需求分析	系统需求规格说明书、软件需求规格说明书、确认测试计划、系统测试计划、初步的用户手册
系统设计	软件架构设计、软件概要设计、详细设计、网络设计	架构设计文档、概要设计说明书、详细设计说明书、程序规格说明书、概要测试计划、详细测试计划、各类设计图
系统实施	软件编码、软件单元/集成/系统测试、综合布线	源码、单元测试、集成测试报告、操作手册
系统验收	确认测试、试运行	确认测试报告，项目验收报告

希赛

二 政府信息化与电子政务



模式

- 政府对政府（G2G）(Government To Government)
- 政府对企业（G2B）或 B2G（Government To Business）
- 政府对公民（G2C）或 C2G（Government To Citizen)
- 政府对公务员（G2E）（Government To Employee）

希赛

三 企业信息化与电子商务

- 企业资源计划
- 客户关系管理
- 供应链管理
- 产品数据管理
- 商业智能
- 决策支持系统
- 知识管理
- 业务流程重组

希赛

三 企业信息化与电子商务 – 企业资源计划（ERP）

物料需求计划
（Material Requirement Planning，MRP）
物料单系统

↓

制造资源计划
（MRPII，manufacturing resource planning II）
增加库存、分销等

↓

企业资源计划
（ERP，enterprise resource planning）
打通了供应链
把财务、人力资源、销售管理等 纳入

希赛

三 企业信息化与电子商务 – 企业资源计划（ERP）

- 管理思想：他是管理思想的变革。
- 软件产品：但不是直接买来就用，需要个性化的开发与部署。
- 管理系统：存在众多的子系统，这些子系统有统一的规划，是互联互通的，便于事前事中监控。

◆ 财会管理
- 会计核算
- 财务管理

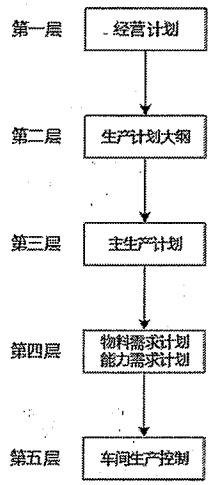
◆ 物流管理
- 分销管理
- 库存控制
- 采购管理

◆ 生产控制管理
- 主生产计划
- 物料需求计划
- 能力需求计划
- 车间控制
- 制造标准

➢ 人力资源管理
- 人力资源规划
- 招聘管理
- 工资核算
- 工时管理
- 差旅费核算

希赛

三 企业信息化与电子商务－企业资源计划（ERP）



经营计划：经营计划是企业总目标的具体体现，企业的高层决策者根据市场调查和需求分析、国家有关政策、企业资源能力和历史状况、同行竞争对手的情况等有关信息，制定经营计划。

生产计划大纲：是根据经营计划的生产目标制定的，是对企业经营计划的细化，用以描述企业在可用资源的条件下，在一定时期中的产量计划。生产计划大纲在企业决策层的三个计划中有承上启下的作用，一方面它是企业经营计划和战略规划的细化，另一方面它又用于指导企业编制主生产计划，指导企业有计划地进行生产。

主生产计划：是对企业生产计划大纲的细化，说明在一定时期内的如下计划：生产什么，生产多少和什么时候交货。

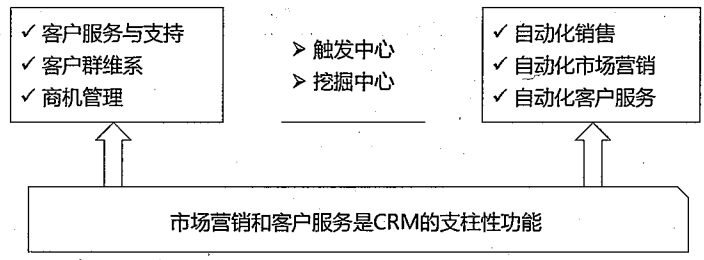
物料需求计划：是对主生产计划的各个项目所需的全部制造件和全部采购件的网络支持计划和时间进度计划。它根据主生产计划对最终产品的需求数量和交货期，推导出构成产品的零部件及材料的需求数量和需求时期，再导出自制零部件的制作订单下达日期和采购件的采购订单发送日期，并进行需求资源和可用能力之间的进一步平衡。

能力需求计划：是对物料需求计划所需能力进行核算的一种计划管理方法。旨在通过分析比较MRP的需求和企业现有生产能力，及早发现能力的瓶颈所在，为实现企业的生产任务而提供能力方面的保障。

车间作业计划：在MRP所产生的加工制造订单的基础上，按照交货期的前后和生产优先级选择原则以及车间的生产资源情况（如设备、人员、物料的可用性、加工能力的大小等），将零部件的生产计划以订单的形式下达给适当的车间。

希赛

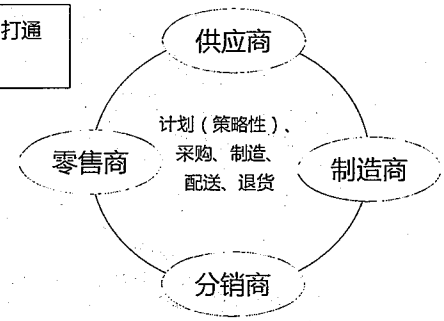
三 企业信息化与电子商务－客户关系管理（CRM）



希赛

三 企业信息化与电子商务－供应链管理（SCM）

SCM理念：强强联合，整合与优化“三流”，打通企业间“信息孤岛”，严格的数据交换标准。



★ **信息化的三流**

- **信息流**

  需求信息流：如客户订单、生产计划、采购合同等

  供应信息流：如入库单、完工报告单、库存记录、可供销售量、提货发运单等

- **资金流**
- **物流**

希赛

三 企业信息化与电子商务－商业智能（BI）

- 需求分析
- 数据仓库建模
- 数据抽取
- 建立BI分析报表
- 用户培训和数据模拟测试
- 系统改进和完善

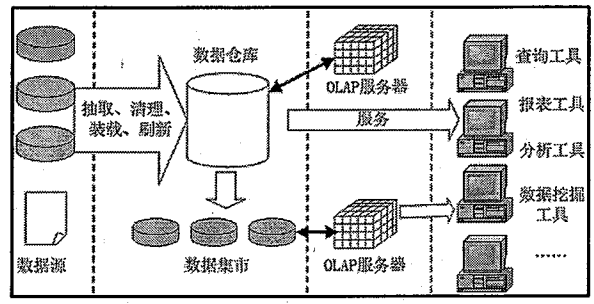
数据仓库、数据挖掘、OLAP

希赛

企业信息化与电子商务 – 数据仓库

数据仓库的特点：

- 面向主题：数据按主题组织。
- 集成的：消除了源数据中的不一致性，提供整个企业的一致性全局信息。
- 相对稳定的(非易失的)：主要进行查询操作，只有少量的修改和删除操作（或是不删除）。
- 反映历史变化(随着时间变化)：记录了企业从过去某一时刻到当前各个阶段的信息，可对发展历程和未来趋势做定量分析和预测。



希赛

企业信息化与电子商务 – 数据挖掘

方法

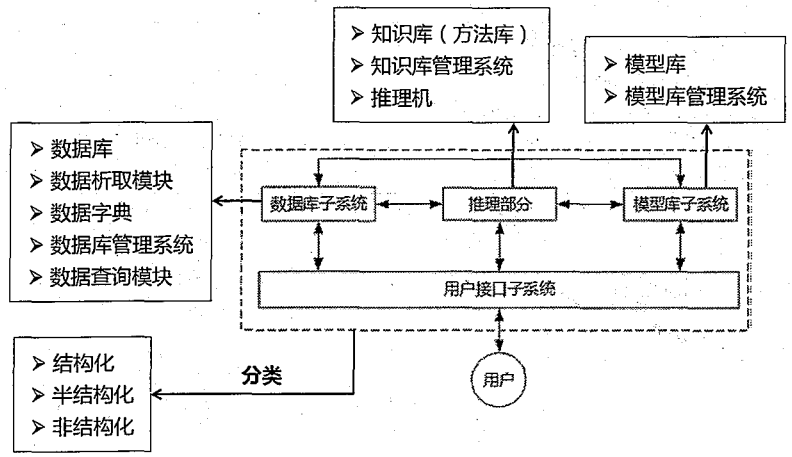
- 决策树（构建树结构进行分析）
- 神经网络（类似统计学中的判别、回归、聚类等功能）
- 遗传算法（三个基本过程：繁殖（选择）、交叉（重组）、变异（突变））
- 关联规则挖掘算法（关联规则是描述数据之间存在关系的规则）

分类

- 关联分析：挖掘出隐藏在数据间的相互关系。
- 序列模式分析：侧重点是分析数据间的前后关系（因果关系）。
- 分类分析：为每一个记录赋予一个标记再按标记分类。
- 聚类分析：分类分析法的逆过程。

希赛

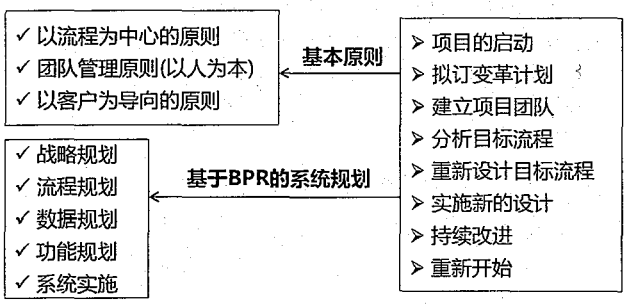
三 企业信息化与电子商务 – 决策支持系统（DSS）



希赛

三 企业信息化与电子商务 – 业务流程重组

- BPR是对企业的业务流程进行根本性的再思考和彻底性的再设计，从而获得可以用诸如成本、质量、服务和速度等方面的业绩来衡量的显著性的成就



希赛

三 企业信息化与电子商务 –业务流程管理

★ BPM是一种以规范化的构造端到端的卓越业务流程为中心，以持续的提高组织业务绩效为目的的系统化方法。

★ PDCA闭环的管理过程

- 明确业务流程所欲获取的成果
- 开发和计划系统的方法，实现以上成果
- 系统地部署方法，确保全面实施
- 根据对业务的检查和分析以及持续的学习活动，评估和审查所执行的方法。并进一步提出计划和实施改进措施

□ BPM与BPR管理思想最根本的不同就在于流程管理并不要求对所有的流程进行再造。构造卓越的业务流程并不是流程再造，而是根据现有流程的具体情况，对流程进行规范化的设计

□ 流程管理包含三个层面：规范流程、优化流程和再造流程

希赛

三 企业信息化与电子商务 – 企业门户

➢ 企业网站：注重单向信息传递，缺互动。

➢ 企业信息门户（EIP）：把各种应用系统、数据资源和互联网资源统一集成到企业门户之下。

➢ 企业知识门户（EKP）：企业网站的基础上增加知识性内容。

➢ 企业应用门户（EAP）：实际上是对企业业务流程的集成。它以业务流程和企业应用为核心，把业务流程中功能不同的应用模块通过门户技术集成在一起。

➢ 企业通用门户：集以上四者于一身。

希赛

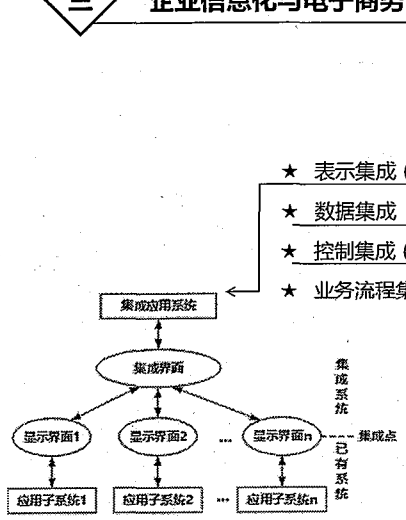
三 企业信息化与电子商务－企业应用集成

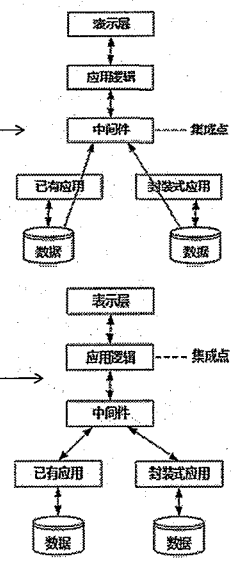
★ 表示集成（界面集成）

★ 数据集成

★ 控制集成（应用集成，API集成）

★ 业务流程集成（过程集成，B2B）





希赛

三 企业信息化与电子商务－企业应用集成

★ 界面集成：把各应用系统的界面集成起来，统一入口，产生“整体”感觉。

★ 数据集成：数据集成是应用集成和业务过程集成的基础。把不同来源、格式、特点性质的数据在逻辑上或物理上有机地集中，从而为企业提供全面的数据共享。ETL、数据仓库、联邦数据库都可视为数据集成。

★ 控制集成（功能集成、应用集成）：业务逻辑层次集成，可以借助于远程过程调用或远程方法调用、面向消息的中间件等技术。

★ 业务流程集成（过程集成）：进行业务流程集成时，企业必须对各种业务信息的交换进行定义、授权和管理，以便改进操作、减少成本、提高响应速度。

★ 消息集成：适用于数据量小、但要求频繁地、立即地、异步地数据交换场合。

★ 共享数据库：实时性强、可以频繁交互，数据的交换属于同步方式。

★ 文件传输：适用于数据量大、交换频度小、即时性要求低的情况。

希赛

三 企业信息化与电子商务 – 电子商务

★ **电子商务的形式**

- 企业对消费者(B2C)
- 企业对企业(B2B)
- 消费者对消费者(C2C)
- 线上对线下（O2O）

希赛

软件工程

希赛

课程内容提要

- 软件开发方法
- 需求工程
- 软件系统建模
- 系统设计
- 测试与评审
- 系统运行与软件维护

希赛

软件开发方法

- ✓ 软件开发方法（★★★）
- ✓ 软件开发模型（★★★★）
- ✓ 逆向工程（★）
- ✓ 净室软件工程（★）

希赛

软件开发方法

- ◆ 结构化法
  - 用户至上
  - 严格区分工作阶段，每阶段有任务与成果
  - 强调系统开发过程的整体性和全局性
  - 系统开发过程工程化，文档资料标准化
  - 自顶向下，逐步分解(求精)
- ◆ 原型法
  - 适用于需求不明确的开发
  - 包括抛弃型原型和进化型原型
- ◆ 面向对象方法
  - 更好的复用性
  - 关键在于建立一个全面、合理、统一的模型
  - 分析、设计、实现三个阶段，界限不明确
- ◆ 面向服务的方法
  - SO方法有三个主要的抽象级别：操作、服务、业务流程
  - SOAD分为三个层次：基础设计层（底层服务构件）、应用结构层（服务之间的接口和服务级协定）和业务组织层（业务流程建模和服务流程编排）
  - 服务建模：分为服务发现、服务规约和服务实现三个阶段

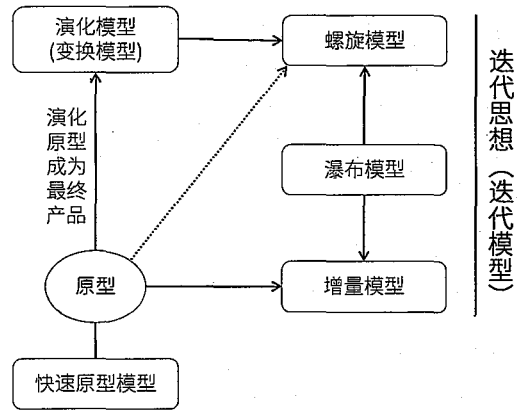
希赛

软件开发方法－软件开发模型

- ✓ 瀑布模型
- ✓ 演化模型
- ✓ 增量模型
- ✓ 螺旋模型
- ✓ 快速原型模型
- ✓ 喷泉模型
- ✓ V模型
- ✓ 迭代模型/迭代开发方法
- ✓ 快速应用开发
- ✓ 构件组装模型/基于构件的开发方法
- ✓ 统一过程/统一开发方法
- ✓ 敏捷开发方法
- ✓ 模型驱动的开发方法
- ✓ 基于架构的开发方法

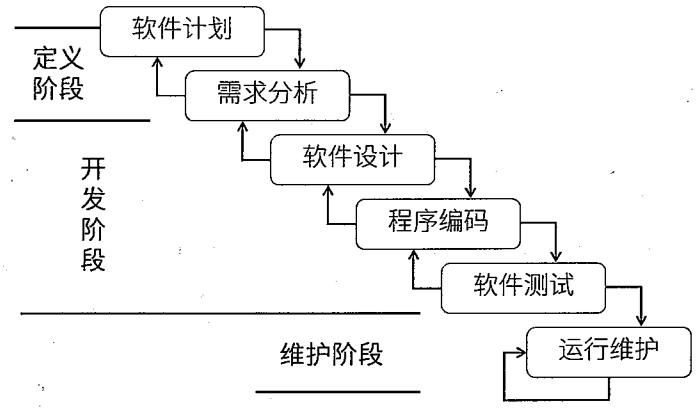
希赛

软件开发方法－软件开发模型－其他经典模型



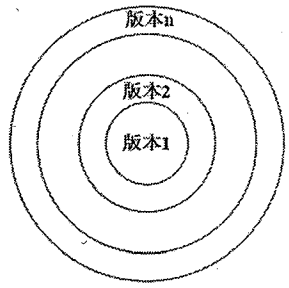
希赛

软件开发方法－软件开发模型－瀑布模型

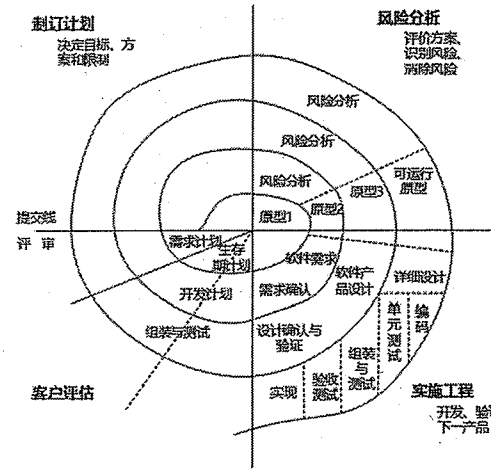


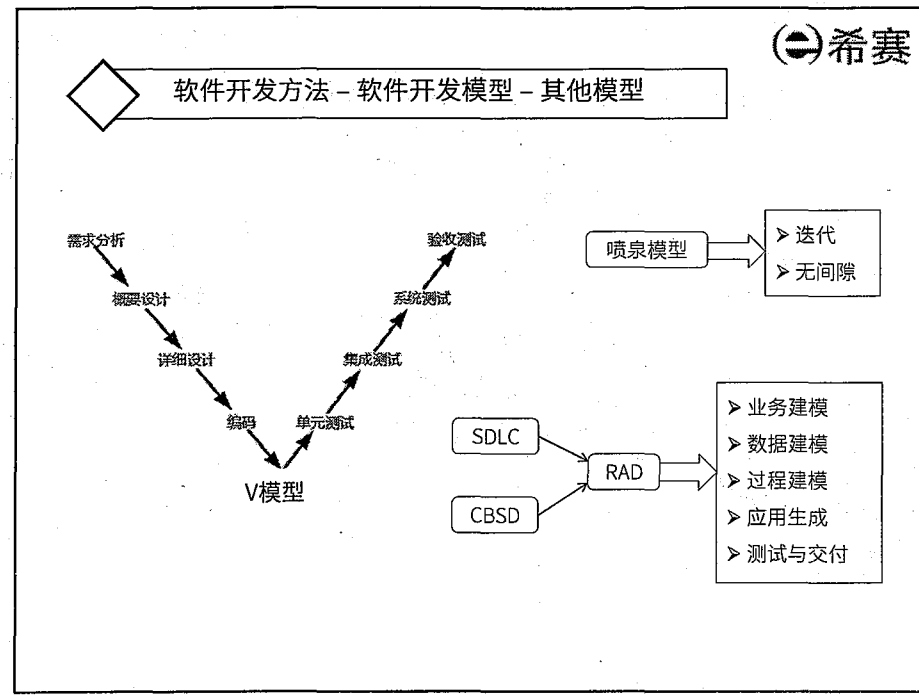
希赛

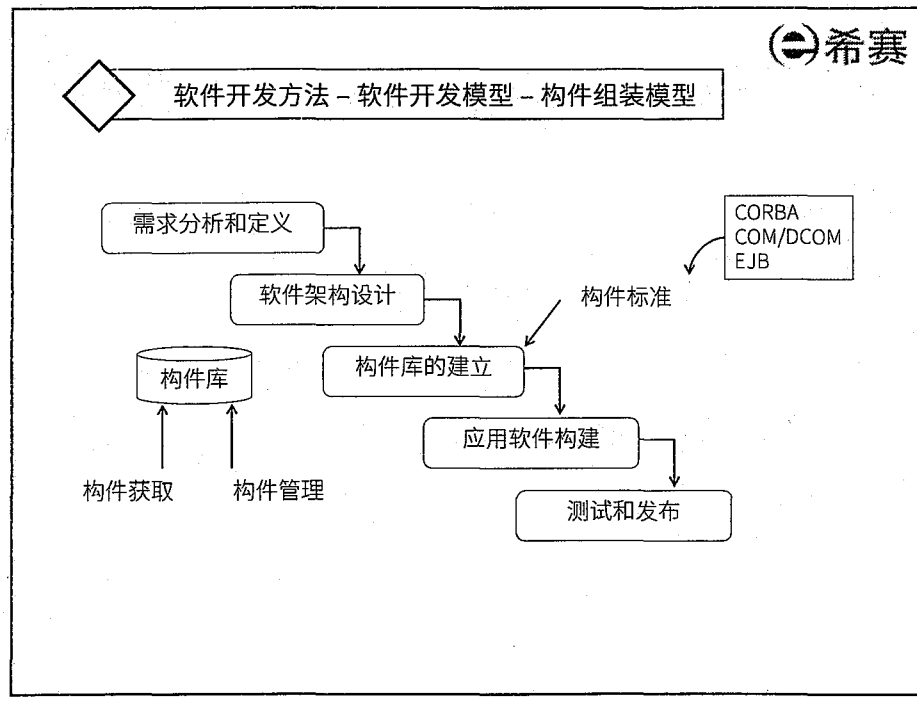
软件开发模型－增量模型与螺旋模型



每一个增量均发布一个可操作的产品

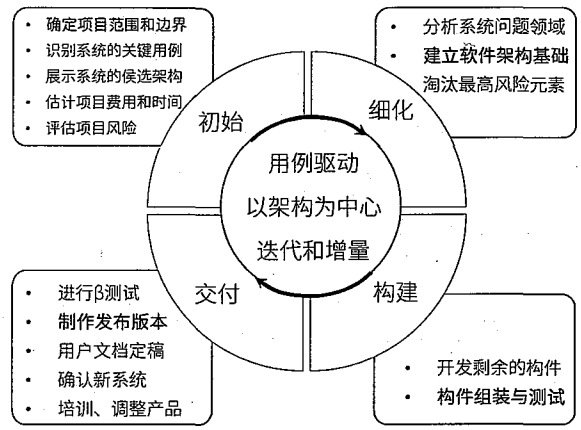



希赛
软件开发方法－软件开发模型－其他模型
需求分析
概要设计
详细设计
编码
单元测试
集成测试
系统测试
验收测试
V模型
喷泉模型
➢ 迭代
➢ 无间隙
SDLC
CBSD
RAD
➢ 业务建模
➢ 数据建模
➢ 过程建模
➢ 应用生成
➢ 测试与交付


希赛
软件开发方法－软件开发模型－构件组装模型
需求分析和定义
软件架构设计
构件库的建立
应用软件构建
测试和发布
构件库
构件获取
构件管理
构件标准
CORBA
COM/DCOM
EJB

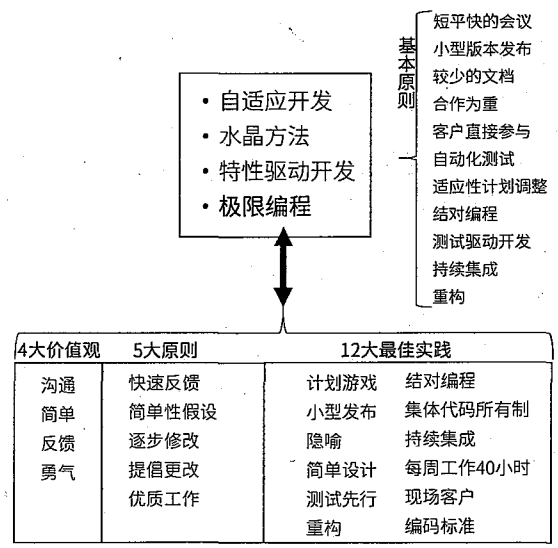
希赛

软件开发方法－软件开发模型－统一过程



希赛

软件开发方法－软件开发模型－敏捷方法



希赛

软件开发模型－敏捷开发方法

- XP (Extreme Programming，极限编程)在所有的敏捷型方法中，XP是最引人瞩目的。它源于Smalitalk圈子，特别是Kent Beck和Ward Cunningham在20世纪80年代末的密切合作。XP在一些对费用控制严格的公司中的使用，已经被证明是非常有效的。
- Cockburn的水晶系列方法，水晶系列方法是由Alistair Cockburn提出的。它与XP方法一样，都有以人为中心的理念，但在实践上有所不同。Alistair考虑到人们一般很难严格遵循一个纪律约束很强的过程，因此，与XP的高度纪律性不同，Alistair探索了用最少纪律约束而仍能成功的方法，从而在产出效率与易于运作上达到一种平衡。也就是说，虽然水晶系列不如XP那样的产出效率，但会有更多的人能够接受并遵循它。
- 开放式源码，这里提到的开放式源码指的是开放源码界所用的一种运作方式。开放式源码项目有一个特别之处，就是程序开发人员在地域上分布很广，这使得它和其他敏捷方法不同，因为一般的敏捷方法都强调项目组成员在同一地点工作。开放源码的一个突出特点就是查错排障(debug)的高度并行性，任何人发现了错误都可将改正源码的“补丁”文件发给维护者。然后由维护者将这些“补丁”或是新增的代码并入源码库。

希赛

软件开发模型－敏捷开发方法

- SCRUM。SCRUM已经出现很久了，像前面所论及的方法一样，该方法强调这样一个事实，即明确定义了的可重复的方法过程只限于在明确定义了的可重复的环境中，为明确定义了的可重复的人员所用，去解决明确定义了的可重复的问题。
- Coad的功用驱动开发方法(FDD-Feature Driven Development)

FDD是由Jeff De Luca和大师Peter Coad提出来的。像其他方法一样，它致力于短时的迭代阶段和可见可用的功能。在FDD中，一个迭代周期一般是两周。

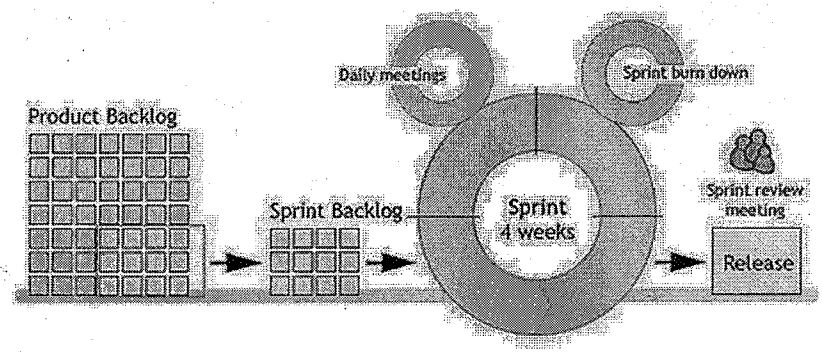
在FDD中，编程开发人员分成两类：首席程序员和“类”程序员(class owner)。首席程序员是最富有经验的开发人员，他们是项目的协调者、设计者和指导者，而“类”程序员则主要做源码编写。

- ASD方法，ASD (Adaptive Software Development)方法由Jim Highsmith提出，其核心是三个非线性的、重叠的开发阶段：猜测、合作与学习。

希赛

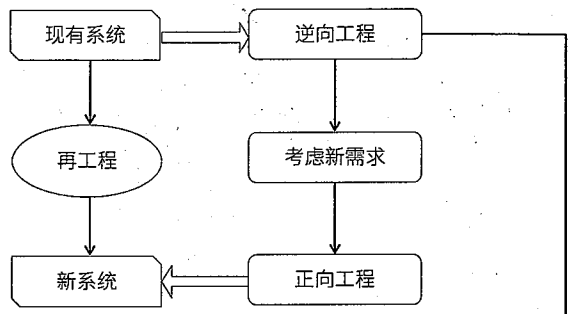
软件开发模型－敏捷开发方法

Scrum开发模型



希赛

软件开发方法－逆向工程



实现级：包括程序的抽象语法树、符号表、过程的设计表示

结构级：包括反映程序分量之间相互依赖关系的信息，例如调用图、结构图、程序和数据结构

功能级：包括反映程序段功能及程序段之间关系的信息，例如数据和控制流模型

领域级：包括反映程序分量或程序诸实体与应用领域概念之间对应关系的信息，例如实体关系模型

希赛

净室软件工程

- 净室即无尘室、洁净室。也就是一个受控污染级别的环境。
- 使用盒结构规约（或形式化方法）进行分析和设计建模，并且强调将正确性验证，而不是测试，作为发现和消除错误的主要机制。
- 使用统计的测试来获取认证被交付的软件的可靠性所必需的出错率信息。

希赛

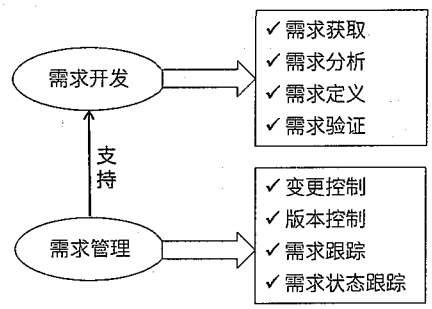
需求工程

- 需求获取（★★★）
- 需求分析（★★★）
- UML4+1视图（★★★★）
- UML图（★★★★★）
- UML关系（★★★★）

希赛

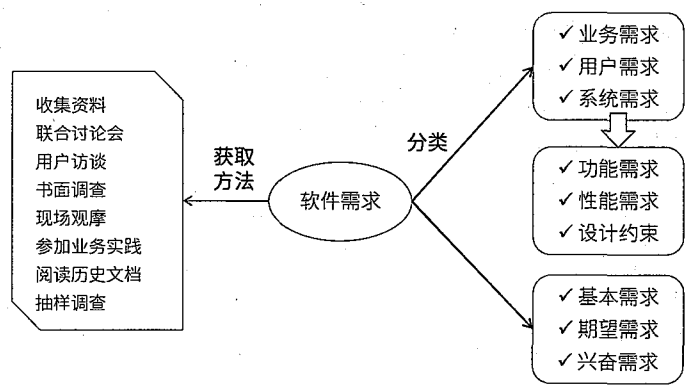
需求工程 – 概述

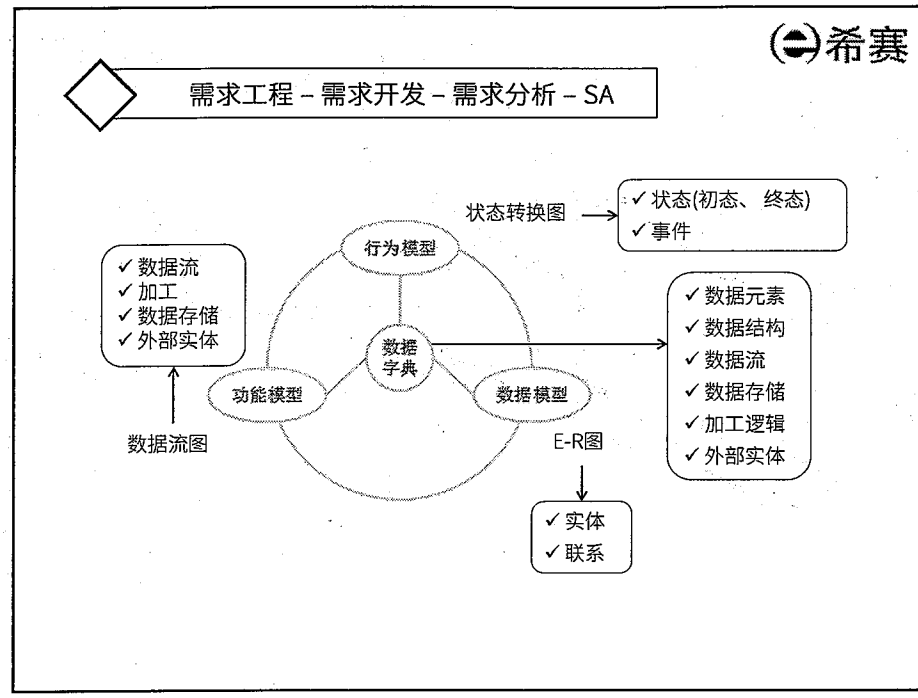
- 软件需求是指用户对系统在功能、行为、性能、设计约束等方面的期望。
- 软件需求是指用户解决问题或达到目标所需的条件或能力，是系统或系统部件要满足合同、标准、规范或其他正式规定文档所需具有的条件或能力，以及反映这些条件或能力的文档说明。

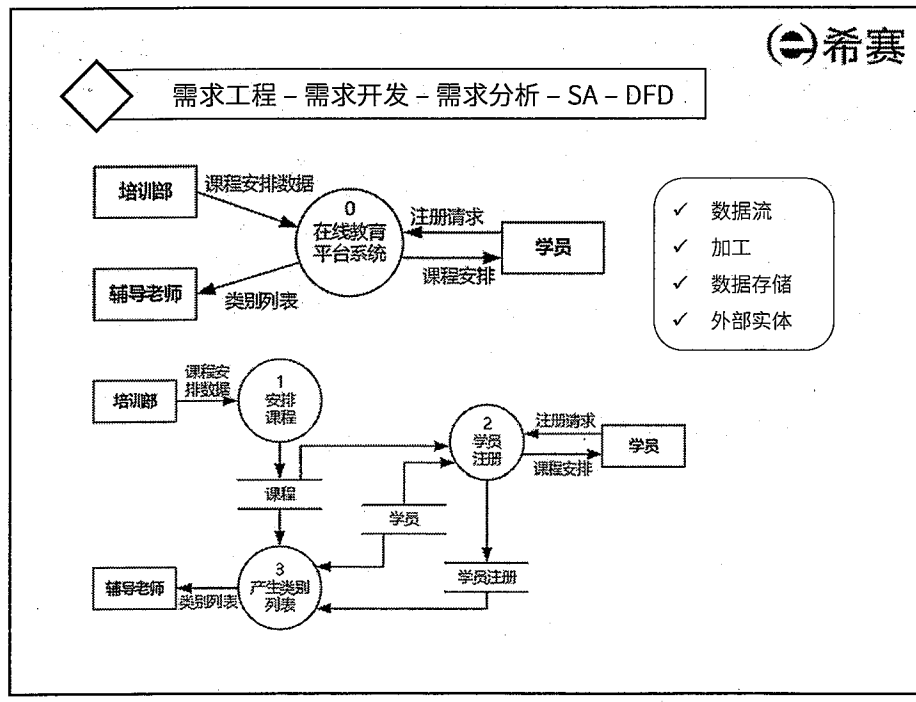


希赛

需求工程 – 需求开发 – 需求获取

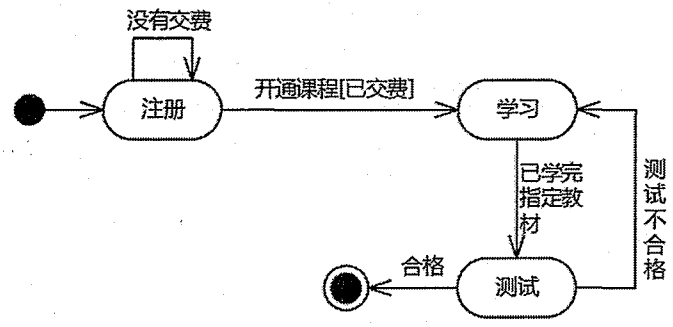



希赛
需求工程 – 需求开发 – 需求分析 – SA
状态转换图
✓ 状态(初态、 终态)
✓ 事件
行为模型
✓ 数据流
✓ 加工
✓ 数据存储
✓ 外部实体
数据字典
✓ 数据元素
✓ 数据结构
✓ 数据流
✓ 数据存储
✓ 加工逻辑
✓ 外部实体
功能模型
数据模型
数据流图
E-R图
✓ 实体
✓ 联系


希赛
需求工程 – 需求开发 – 需求分析 – SA – DFD
培训部
课程安排数据
0
在线教育
平台系统
注册请求
学员
课程安排
辅导老师
类别列表
✓ 数据流
✓ 加工
✓ 数据存储
✓ 外部实体
培训部
课程安
排数据
1
安排
课程
2
学员
注册
注册请求
学员
课程安排
课程
学员
学员注册
辅导老师
类别列表
3
产生类别
列表

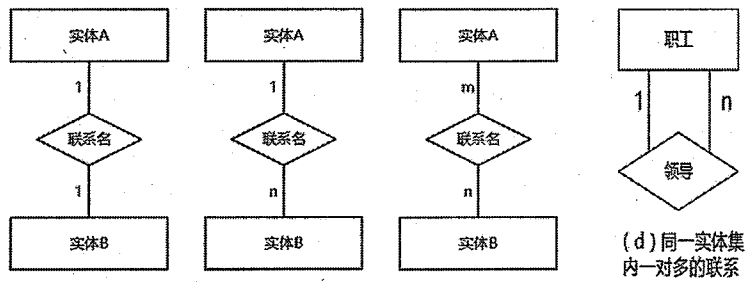
希赛

需求工程－需求开发－需求分析－SA－STD

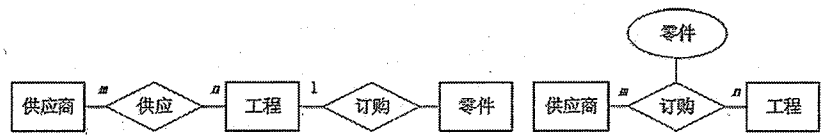


希赛

需求工程－需求开发－需求分析－SA－E-R图



(a)1：1的联系　(b)1：n的联系　(c)m：n的联系　(d)同一实体集内一对多的联系



(a)　(b)

希赛

需求开发－需求分析－OOA－相关概念

- ★ 对象
- ★ 类（实体类、边界类、控制类）
- ★ 抽象
- ★ 封装
- ★ 继承与泛化
- ★ 多态
- ★ 接口
- ★ 消息
- ★ 组件
- ★ 模式和复用

- ◻ 实体类映射需求中的每个实体，实体类保存需要存储在永久存储体中的信息，例如，在线教育平台系统可以提取出学员类和课程类，它们都属于实体类。
- ◻ 控制类是用于控制用例工作的类，一般是由动宾结构的短语（“动词+名词”或“名词+动词”）转化来的名词，例如，用例“身份验证”可以对应于一个控制类“身份验证器”，它提供了与身份验证相关的所有操作。
- ◻ 边界类用于封装在用例内、外流动的信息或数据流。边界类位于系统与外界的交接处，包括所有窗体、报表、打印机和扫描仪等硬件的接口，以及与其他系统的接口。

希赛

需求工程－需求开发－需求分析－OOA

- ✓ 构造块
- ✓ 规则
- ✓ 公共机制

- ➢ 事物
- ➢ 关系
- ➢ 图

➢ 结构事物：最静态的部分，包括：类、接口、协作、用例、活动类、构件和节点。
➢ 行为事物：代表时间和空间上的动作。包括：消息、动作次序、连接。
➢ 分组事物：看成是个盒子，如：包、构件。
➢ 注释事物：UML模型的解释部分。描述、说明和标注模型的元素。

➢ 范围：给一个名字以特定含义的语境
➢ 可见性：怎样使用或看见名字
➢ 完整性：事物如何正确、一致地相互联系
➢ 执行：运行或模拟动态模型的含义是什么

➢ 规格说明：事物语义的细节描述，它是模型真正的核心
➢ 修饰：通过修饰来表达更多的信息
➢ 公共分类：类与对象、接口与实现
➢ 扩展机制：允许添加新的规则

希赛

UML

UML图

静态图（结构图）

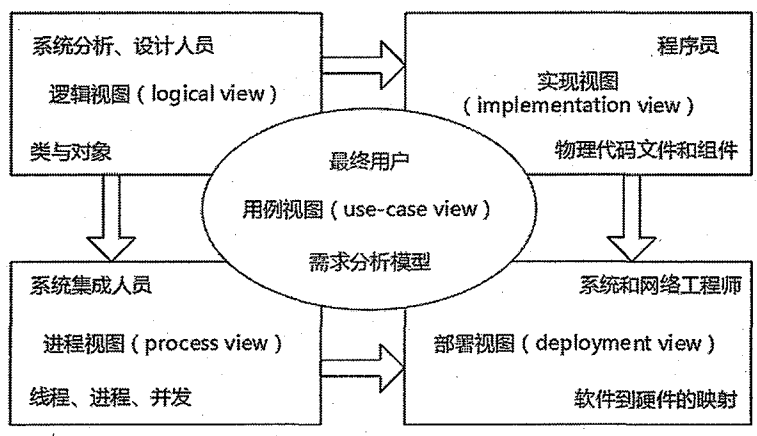
- 类图：一组类、接口、协作和它们之间的关系
- 对象图：一组对象及它们之间的关系
- 构件图：一个封装的类和它的接口
- 部署图：软硬件之间映射
- 制品图：系统的物理结构
- 包图：由模型本身分解而成的组织单元，以及他们之间的依赖关系
- 组合结构图

动态图（行为图）

- 用例图：系统与外部参与者的交互
- 顺序图：强调按时间顺序
- 通信图（协作图）
- 状态图：状态转换变迁
- 活动图：类似程序流程图，并行行为
- 定时图：强调实际时间
- 交互概览图

希赛

需求开发-需求分析-OOA-UML-4+1视图



希赛

需求工程－需求开发－需求分析－OOA

用例模型

- ✓ 识别参与者
- ✓ 合并需求获得用例
- ✓ 细化用例描述
- ✓ 调整用例模型

- 包含关系
- 扩展关系
- 泛化关系

- 用例名称
- 简要说明
- 事件流
- 非功能需求
- 前置条件
- 后置条件
- 扩展点
- 优先级

分析模型

- ✓ 定义概念类
- ✓ 识别类之间的关系
- ✓ 为类添加职责
- ✓ 建立交互图

- 依赖关系
- 关联关系
- 聚合关系
- 组合关系
- 泛化关系
- 实现关系

希赛

用例图

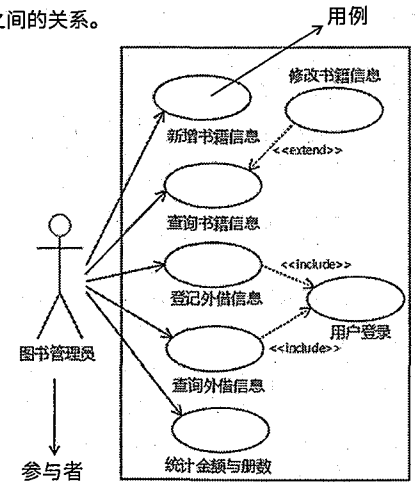
- ✓ 用例图描述一组用例、参与者及它们之间的关系。
- ✓ 用户角度描述系统功能；
- ✓ 参与者是外部触发因素；
  （包括用户、组织、外部系统，时间）
- ✓ 用例是功能单元。

关系包括：

包含关系、扩展关系、泛化关系

用例建模的流程：

- ★ 识别参与者（必须）
- ★ 合并需求获得用例（必须）
- ★ 细化用例描述（必须）
- ★ 调整用例模型（可选）



希赛

用例图

1.用例名称：
开通课程（UC02）
2.简要说明：
为用户开通学习课程的权限，将其标记为“学员”，同时修改所选修课程的选修人数。
3.事件流：
3.1 主事件流
客服人员向系统发出“开通课程”请求。
系统要求客服人员选择开通课程的类型（软考、PMP、通信课程）。
客服人员做出选择后，系统显示相应界面，让客服人员输入信息，并自动根据权限规则生成权限。
客服人员输入学员的相关信息，包括学员用户名、所交费用、交费时间、选修课程名称。
系统确认学员所交费用和所选修课程的规定费用一致。
系统将所输入的信息存储建档，开通学员课程权限。
3.2 备选事件流
5a）如果学员所交费用少于所选修课程的规定费用，则显示所选修课程的规定费用，并要求客服人员选择修改或取消输入。
5a1）客服人员选择取消输入，则结束用例，不做存储建档工作。
5a2）客服人员选择修改用户所交费用后，转到5）。
5b）如果学员所交费用多于所选修课程的规定费用，则显示多余的费用数量……

希赛

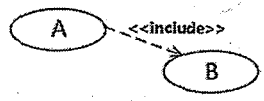
用例图

4.非功能需求
无特殊要求。
5.前置条件
客服人员登录在线教育平台系统。
6.后置条件
修改学员权限，修改课程选修人数。
7.扩展点
无。
8.优先级
最高（满意度 5，不满意度5）。

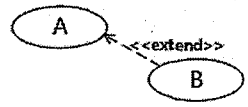
希赛

包含、扩展、泛化

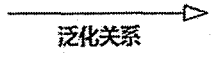
包含关系：其中这个提取出来的公共用例称为抽象用例，而把原始用例称为基本用例或基础用例系：当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。



扩展关系：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种分支，则可以将这个用例分为一个基本用例和一个或多个扩展用例，这样使描述可能更加清晰。



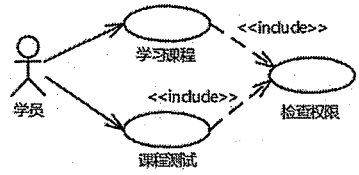
泛化关系：当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。



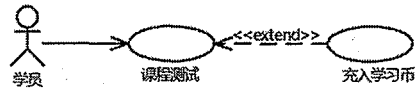
希赛

包含、扩展、泛化

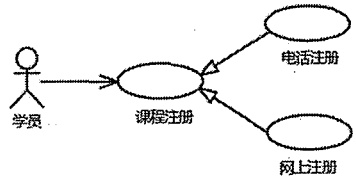
包含关系



扩展关系



泛化关系



希赛

例题讲解

● 在某银行业务的用例模型中，“取款”用例需要等到“存款”用例执行之后才能执行，两个用例之间的关系属于（ ）；“取款”和“存款”两个用例中都需要执行查询余额的功能，将查询余额提取成独立的用例，那么“取款”和“存款”用例与“查询余额”用例之间的关系属于（ ）。

A．关联关系　　B．扩展关系　　C．使用关系　　D．依赖关系

A．扩展关系　　B．使用关系　　C．依赖关系　　D．继承关系

希赛

类图与对象图

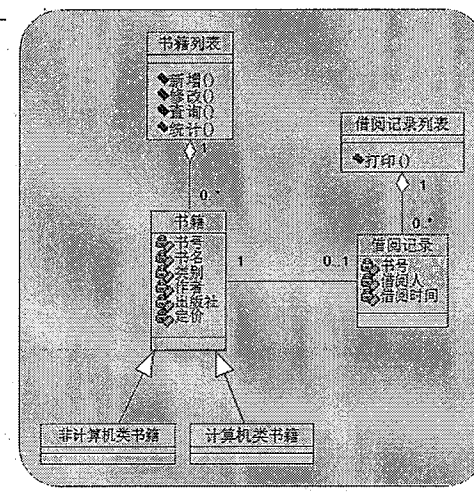
类图（class diagram）:类图描述一组类、接口、协作和它们之间的关系。

对象图（object diagram）:对象图描述一组对象及它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。

◆ 类名，方法名，属性名

◆ 多重度

◆ 关系



希赛

类图与对象图

1：表示一个集合中的一个对象对应另一个集合中1个对象。

0..*：表示一个集合中的一个对象对应另一个集合中的0个或多个对象。

（可以不对应）

1..*：表示一个集合中的一个对象对应另一个集合中的一个或多个对象。

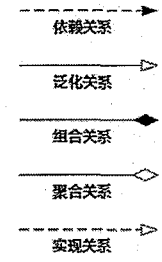
（至少对应一个）

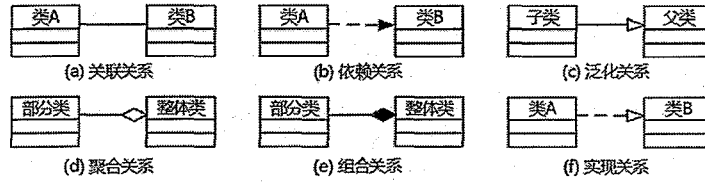
*：表示一个集合中的一个对象对应另一个集合中的多个的对象。

希赛

类图与对象图

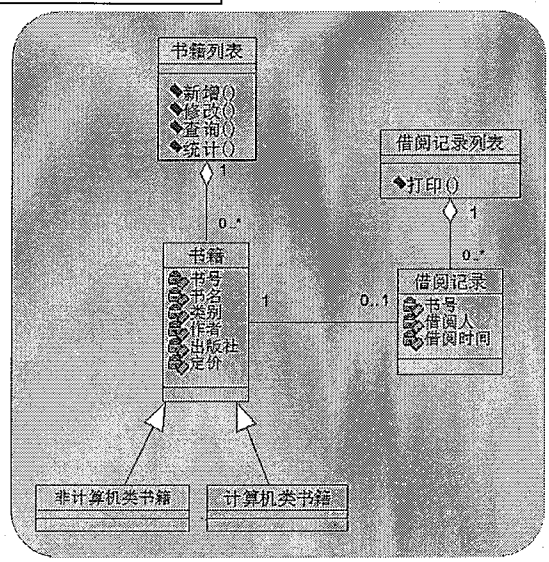
- ✓ 依赖关系：一个事物发生变化影响另一个事物。
- ✓ 泛化关系：特殊/一般关系
- ✓ 关联关系：描述了一组链，链是对象之间的连接。
  - 聚合关系：整体与部分生命周期不同。
  - 组合关系：整体与部分生命周期相同。
- ✓ 实现关系：接口与类之间的关系





希赛

类图与对象图



希赛

例题讲解

● UML 用关系把事物结合在一起，（ ）描述一个事物发生变化会影响另一个事物的语义；（ ）描述特殊元素的对象可替换一般元素的对象。

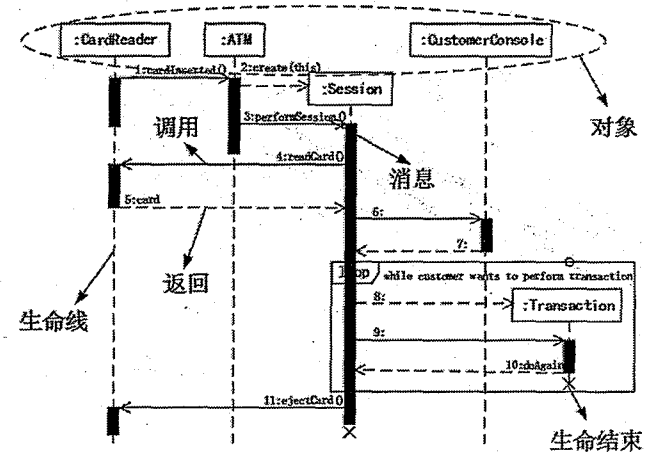
A. 聚合关系　B. 关联关系　C. 包含关系　D. 依赖关系

A. 实现关系　B. 聚合关系　C. 泛化关系　D. 关联关系

希赛

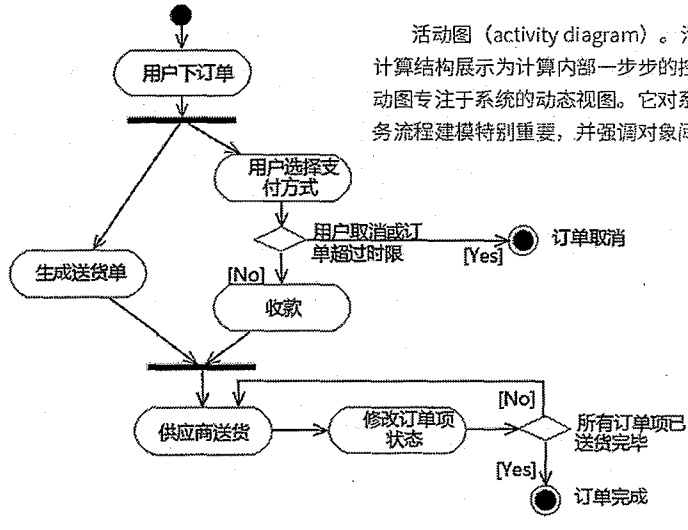
顺序图

顺序图（sequence diagram，序列图）。顺序图是一种交互图（interaction diagram），它强调对象之间消息发送的顺序，同时显示对象之间的交互。

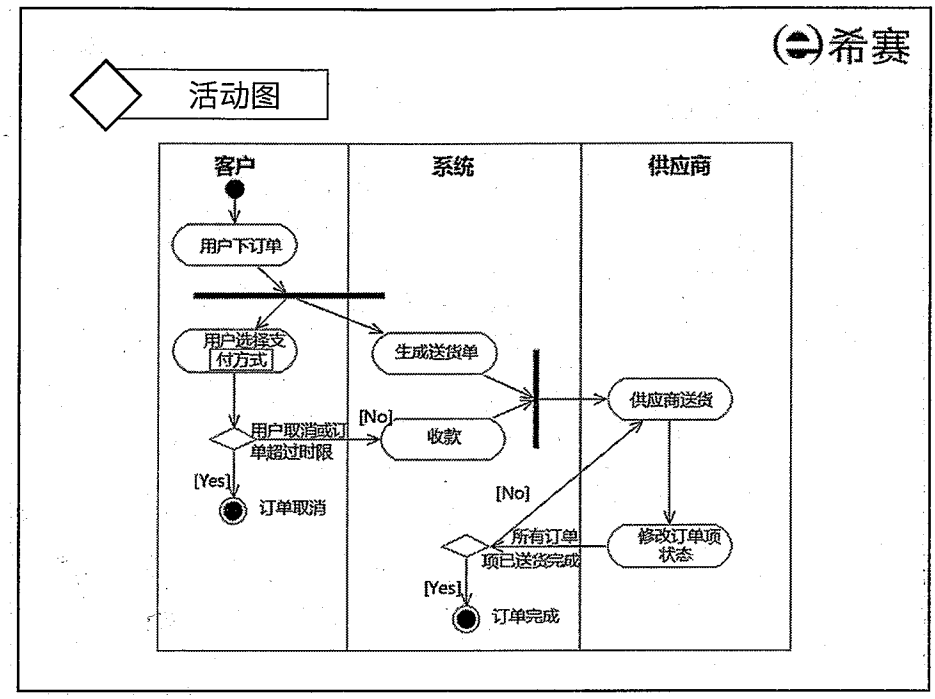


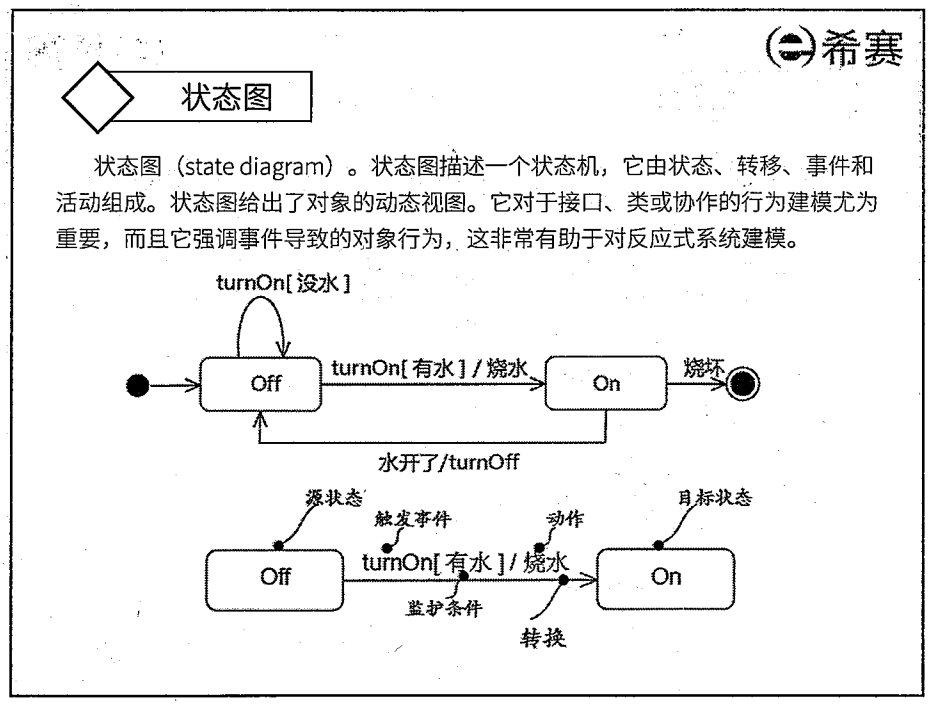
希赛

活动图



活动图（activity diagram）。活动图将进程或其他计算结构展示为计算内部一步步的控制流和数据流。活动图专注于系统的动态视图。它对系统的功能建模和业务流程建模特别重要，并强调对象间的控制流程。


希赛
活动图
客户
系统
供应商
用户下订单
用户选择支付方式
生成送货单
供应商送货
用户取消或订单超过时限
[No]
收款
[Yes]
订单取消
[No]
所有订单项已送货完成
修改订单项状态
[Yes]
订单完成


希赛
状态图
状态图（state diagram）。状态图描述一个状态机，它由状态、转移、事件和活动组成。状态图给出了对象的动态视图。它对于接口、类或协作的行为建模尤为重要，而且它强调事件导致的对象行为，这非常有助于对反应式系统建模。
turnOn[没水]
Off
turnOn[有水]/烧水
On
烧杯
水开了/turnOff
源状态
触发事件
动作
目标状态
Off
turnOn[有水]/烧水
On
监护条件
转换

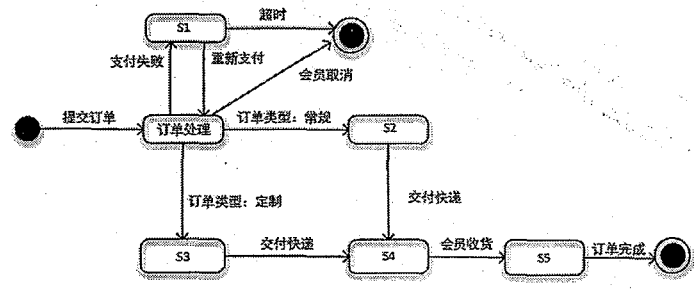
希赛

状态图

在订单处理的过程中，会员可以点击“取消订单"取消该订单。如果支付失败，该订单将被标记为挂起状态，可后续重新支付，如果挂起超时30分钟未支付，系统将自动取消该订单。订单支付成功后，系统判断订单类型：

(1)对于常规订单，标记为备货状态，订单信息发送到货运部，完成打包后交付快递发货；

(2)对于定制订单，会自动进入定制状态，定制完成后交付快递发货。会员在系统中点击”收货"按钮变为收货状态，结束整个订单的处理流程。



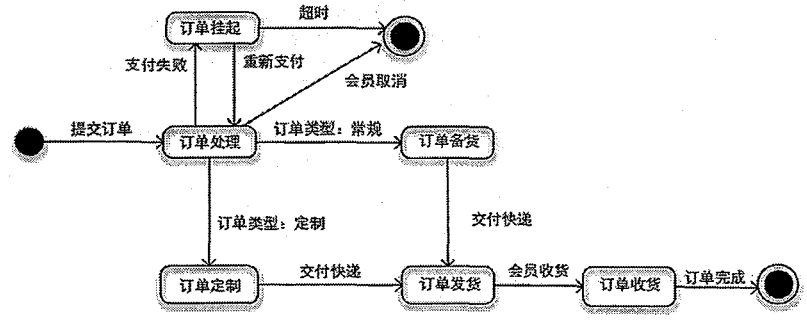
希赛

状态图

在订单处理的过程中，会员可以点击“取消订单"取消该订单。如果支付失败，该订单将被标记为挂起状态，可后续重新支付，如果挂起超时30分钟未支付，系统将自动取消该订单。订单支付成功后，系统判断订单类型：

(1)对于常规订单，标记为备货状态，订单信息发送到货运部，完成打包后交付快递发货；

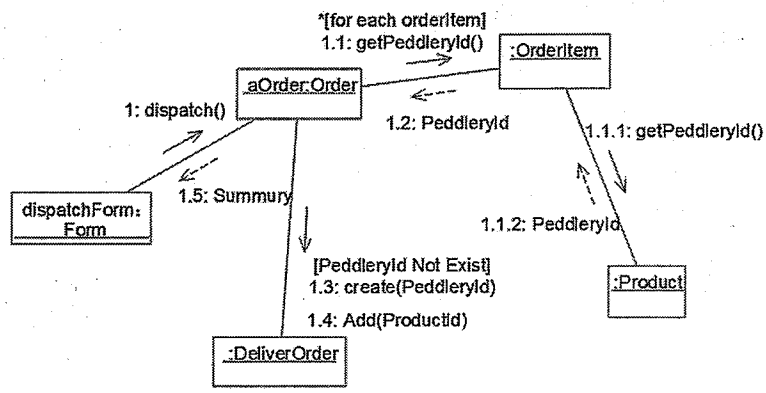
(2)对于定制订单，会自动进入定制状态，定制完成后交付快递发货。会员在系统中点击”收货"按钮变为收货状态，结束整个订单的处理流程。



希赛

通信图（协作图）

通信图（communication diagram）。通信图也是一种交互图，它强调收发消息的对象或参与者的结构组织。顺序图和通信图表达了类似的基本概念，但它们所强调的概念不同，顺序图强调的是时序，通信图强调的是对象之间的组织结构（关系）。



希赛

需求工程－需求开发－需求定义

严格定义法 ⇨
- ✓所有需求都能够被预先定义
- ✓开发人员与用户之间能够准确而清晰地交流
- ✓采用图形/文字可以充分体现最终系统

- 用结构化和自然语言编写文本型文档
- 建立图形化模型
- 编写形式化规格说明

原型法 ⇨
- ✓并非所有的需求都能在开发前被准确地说明
- ✓项目参加者之间通常都存在交流上的困难
- ✓需要实际的、可供用户参与的系统模型
- ✓有合适的系统开发环境
- ✓反复是完全需要和值得提倡的，需求一旦确定，就应遵从严格的方法

希赛

需求工程－需求开发－需求验证

需求验证

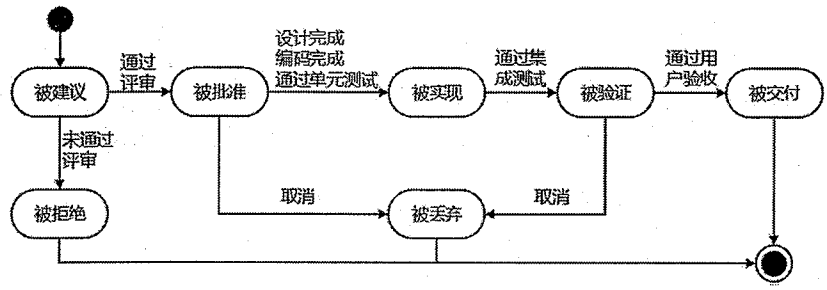
- 需求评审
- 需求测试

- 正式评审
- 非正式评审

- ✓ 用户签字确认
- ✓ 验收标准之一

希赛

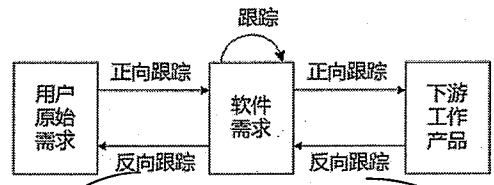
需求工程－需求管理－定义需求基线



需求的状态

希赛

需求工程－需求管理－需求跟踪

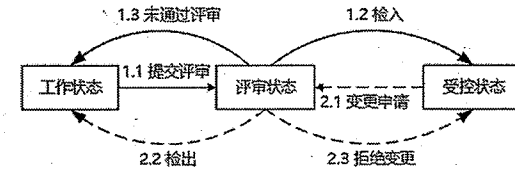


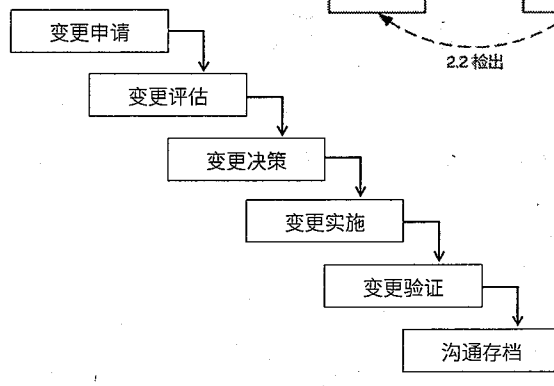
原始需求＼用例	UC-1	UC-2	UC-3	……	UC-n
FR-1					
FR-2					
……					
FR-m					

用例＼元素	功能点	设计元素	代码模块	测试用例
UC-1				
UC-2				
……				
UC-n				

希赛

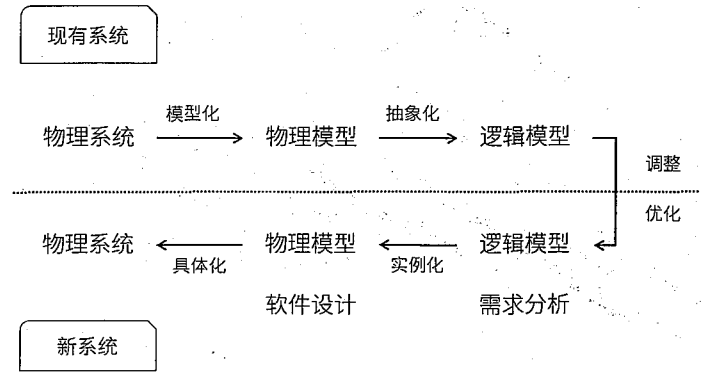
变更控制





希赛

软件系统建模



希赛

系统设计

★ 界面设计（★★）

★ 结构化设计（★★）

★ 面向对象设计（★★★★★）

希赛

系统设计 – 人机界面设计

★ 置于用户控制之下

- 以不强迫用户进入不必要的或不希望的动作的方式来定义交互方式
- 提供灵活的交互
- 允许用户交互可以被中断和撤消
- 当技能级别增加时可以使交互流水化并允许定制交互
- 使用户隔离内部技术细节
- 设计应允许用户和出现在屏幕上的对象直接交互

★ 减少用户的记忆负担

- 减少对短期记忆的要求
- 建立有意义的缺省
- 定义直觉性的捷径
- 界面的视觉布局应该基于真实世界的隐喻
- 以不断进展的方式揭示信息

★ 保持界面的一致性

- 允许用户将当前任务放入有意义的语境
- 在应用系列内保持一致性
- 如过去的交互模型已建立起了用户期望，除非有迫不得已的理由，不要改变它

希赛

系统设计 – 结构化设计

➢ 概要设计
➢ 详细设计

✓ 抽象化
✓ 自顶而下、逐步求精
✓ 信息隐蔽
✓ 模块独立（高内聚、低耦合）

- 保持模块的大小适中
- 尽可能减少调用的深度
- 多扇入，少扇出
- 单入口，单出口
- 模块的作用域应该在模块之内
- 功能应该是可预测的

希赛

系统设计－结构化设计

高内聚

内聚类型	描　述
功能内聚	完成一个单一功能，各个部分协同工作，缺一不可
顺序内聚	处理元素相关，而且必须顺序执行
通信内聚	所有处理元素集中在一个数据结构的区域上
过程内聚	处理元素相关，而且必须按特定的次序执行
瞬时内聚（时间内聚）	所包含的任务必须在同一时间间隔内执行
逻辑内聚	完成逻辑上相关的一组任务
偶然内聚（巧合内聚）	完成一组没有关系或松散关系的任务

低内聚

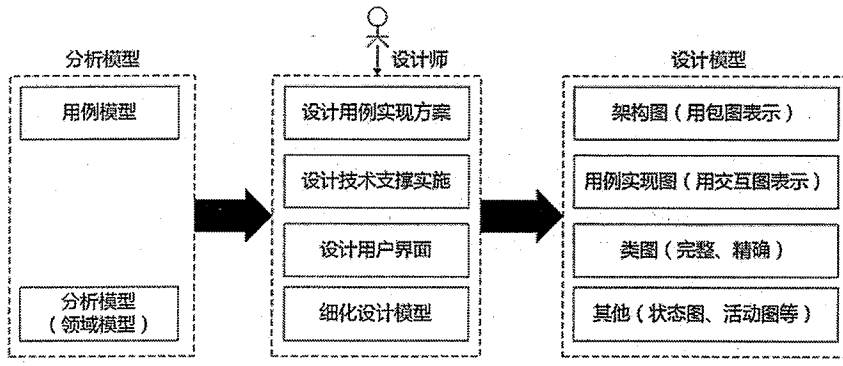
低耦合

耦合类型	描　述
非直接耦合	两个模块之间没有直接关系，它们之间的联系完全是通过主模块的控制和调用来实现的
数据耦合	一组模块借助参数表传递简单数据
标记耦合	一组模块通过参数表传递记录信息（数据结构）
控制耦合	模块之间传递的信息中包含用于控制模块内部逻辑的信息
外部耦合	一组模块都访问同一全局简单变量，而且不是通过参数表传递该全局变量的信息
公共耦合	多个模块都访问同一个公共数据环境
内部耦合	一个模块直接访问另一个模块的内部数据；一个模块不通过正常入口转到另一个模块的内部；两个模块有一部分程序代码重叠；一个模块有多个入口

高耦合

希赛

系统设计－面向对象设计－基本过程



希赛

系统设计 – 面向对象设计 – 设计原则

★ 单一职责原则：设计目的单一的类
★ 开放-封闭原则：对扩展开放，对修改封闭
★ 李氏(Liskov)替换原则：子类可以替换父类
★ 依赖倒置原则：要依赖于抽象，而不是具体实现；针对接口编程，不要针对实现编程
★ 接口隔离原则：使用多个专门的接口比使用单一的总接口要好
★ 组合重用原则：要尽量使用组合，而不是继承关系达到重用目的
★ 迪米特(Demeter)原则(最少知识法则)：一个对象应当对其他对象有尽可能少的了解

希赛

系统设计 – 面向对象设计 – 设计模式的概念

★ 架构模式：软件设计中的高层决策，例如C/S结构就属于架构模式，架构模式反映了开发软件系统过程中所作的基本设计决策
★ 设计模式：主要关注软件系统的设计，与具体的实现语言无关
★ 惯用法：是最低层的模式，关注软件系统的设计与实现，实现时通过某种特定的程序设计语言来描述构件与构件之间的关系。每种编程语言都有它自己特定的模式，即语言的惯用法。例如引用-计数就是C++语言中的一种惯用法

希赛

系统设计 – 面向对象设计 – 设计模式的分类

➢ 创建型模式：创建对象

- 工厂方法（Factory Method）模式
- 抽象工厂（Abstract Factory）模式
- 原型（Prototype）模式
- 单例（Singleton）模式
- 构建器（Builder）模式

➢ 结构型模式：更大的结构

- 适配器（Adapter）模式
- 桥接（Bridge）模式
- 组合（Composite）模式
- 装饰（Decorator）模式
- 外观（Facade）模式
- 享元（Flyweight）模式
- 代理（Proxy）模式

➢ 行为型模式：交互及职责分配

- 职责链（Chain of Responsibility）模式
- 命令（Command）模式
- 解释器（Interpreter）模式
- 迭代器（Iterator）模式
- 中介者（Mediator）模式
- 备忘录（Memento）模式
- 观察者（Observer）模式
- 状态（State）模式
- 策略（Strategy）模式
- 模板方法（Template Method）模式
- 访问者（Visitor）模式

有下划线的表示既可以是类模式，也可以是对象模式；
无下划线的表示只是对象模式

希赛

系统设计 – 面向对象设计 – 创建性模式

设计模式名称	简要说明	速记关键字
Factory Method 工厂方法模式	定义一个创建对象的接口，但由子类决定需要实例化哪一个类。工厂方法使得子类实例化的过程推迟	动态生产对象
Abstract Factory 抽象工厂模式	提供一个接口，可以创建一系列相关或相互依赖的对象，而无需指定它们具体的类	生产成系列对象
Builder 构建器模式	将一个复杂类的表示与其构造相分离，使得相同的构建过程能够得出不同的表示	复杂对象构造
Prototype 原型模式	用原型实例指定创建对象的类型，并且通过拷贝这个原型来创建新的对象	克隆对象
Singleton 单例模式	保证一个类只有一个实例，并提供一个访问它的全局访问点	单实例

希赛

系统设计 - 面向对象设计 - 结构性模式

设计模式名称	简要说明	速记关键字
Adapter 适配器模式	将一个类的接口转换成用户希望得到的另一种接口。它使原本不相容的接口得以协同工作	转换接口
Bridge 桥接模式	将类的抽象部分和它的实现部分分离开来，使它们可以独立地变化	继承树拆分
Composite 组合模式	将对象组合成树型结构以表示“整体-部分”的层次结构，使得用户对单个对象和组合对象的使用具有一致性	树形目录结构
Decorator 装饰模式	动态地给一个对象添加一些额外的职责。它提供了用子类扩展功能的一个灵活的替代，比派生一个子类更加灵活	动态附加职责
Facade 外观模式	定义一个高层接口，为子系统中的一组接口提供一个一致的外观，从而简化了该子系统的使用	对外统一接口
Flyweight 享元模式	提供支持大量细粒度对象共享的有效方法	汉字编码
Proxy 代理模式	为其他对象提供一种代理以控制这个对象的访问	快捷方式

希赛

系统设计 - 面向对象设计 - 行为性模式(1)

设计模式名称	简要说明	速记关键字
Chain of Responsibility 职责链模式	通过给多个对象处理请求的机会，减少请求的发送者与接收者之间的耦合。将接收对象链接起来，在链中传递请求，直到有一个对象处理这个请求	传递职责
Command 命令模式	将一个请求封装为一个对象，从而可用不同的请求对客户进行参数化，将请求排队或记录请求日志，支持可撤销的操作	日志记录，可撤销
Interpreter 解释器模式	给定一种语言，定义它的文法表示，并定义一个解释器，该解释器用来根据文法表示来解释语言中的句子	虚拟机的机制
Iterator 迭代器模式	提供一种方法来顺序访问一个聚合对象中的各个元素，而不需要暴露该对象的内部表示	数据集
Mediator 中介者模式	用一个中介对象来封装一系列的对象交互。它使各对象不需要显式地相互调用，从而达到低耦合，还可以独立地改变对象间的交互	不直接引用

希赛

系统设计 – 面向对象设计 – 行为性模式(2)

设计模式名称	简要说明	速记关键字
Memento 备忘录模式	在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态，从而可以在以后将该对象恢复到原先保存的状态	游戏存档
Observer 观察者模式	定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并自动更新	联动
State 状态模式	允许一个对象在其内部状态改变时改变它的行为	状态变成类
Strategy 策略模式	定义一系列算法，把它们一个个封装起来，并且使它们之间可互相替换，从而让算法可以独立于使用它的用户而变化	多方案切换
Template Method 模板方法模式	定义一个操作中的算法骨架，而将一些步骤延迟到子类中，使得子类可以不改变一个算法的结构即可重新定义算法的某些特定步骤	框架
Visitor 访问者模式	表示一个作用于某对象结构中的各元素的操作，使得在不改变各元素的类的前提下定义作用于这些元素的新操作	数据与操作分离

希赛

系统设计

● （ ）的选择是开发一个软件系统时的基本设计决策；（ ）是最低层的模式，关注软件系统的设计与实现，描述了如何实现构件及构件之间的关系。引用一计数是C++管理动态资源时常用的一种（ ）。

(1) A. 架构模式　　B. 惯用法
　　C. 设计模式　　D. 分析模式

(2) A. 架构模式　　B. 惯用法
　　C. 设计模式　　D. 分析模式

(3) A. 架构模式　　B. 惯用法
　　C. 设计模式　　D. 分析模式

希赛

软件测试

★ 软件测试方法

★ 软件调试

希赛

软件测试 – 测试类型

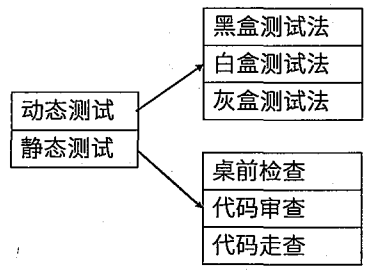
★ 尽早、不断的进行测试

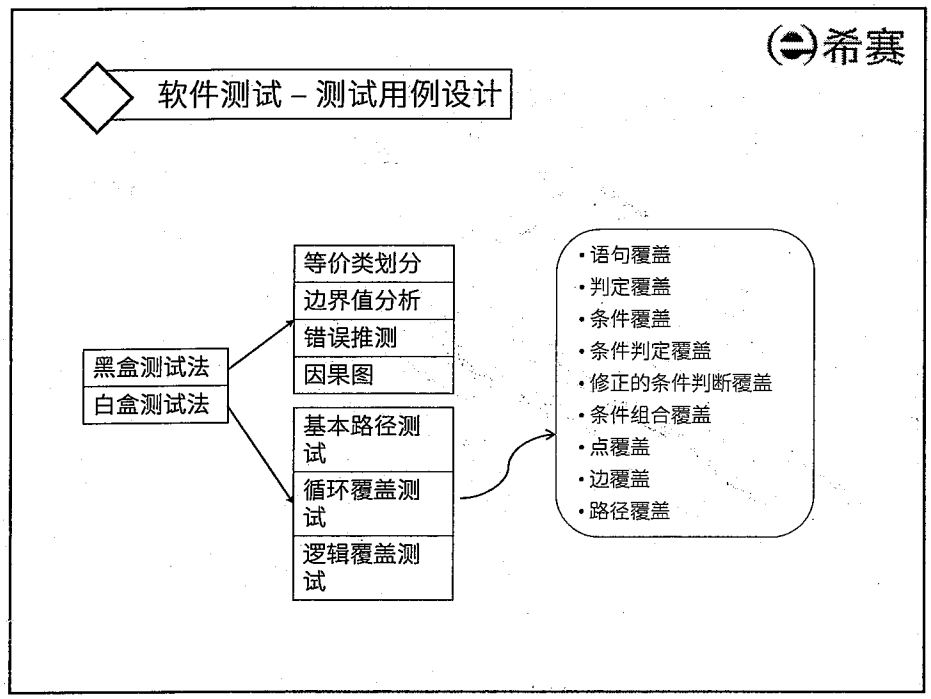
★ 程序员避免测试自己设计的程序

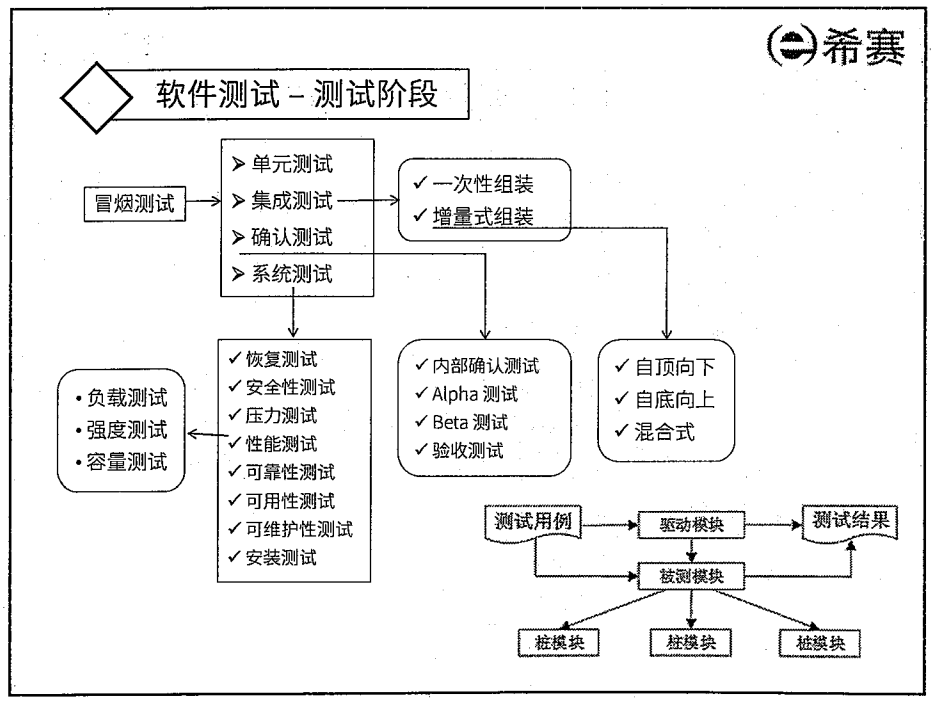
★ 既要选择有效、合理的数据，也要选择无效、不合理的数据

★ 修改后应进行回归测试

★ 尚未发现的错误数量与该程序已发现错误数成正比

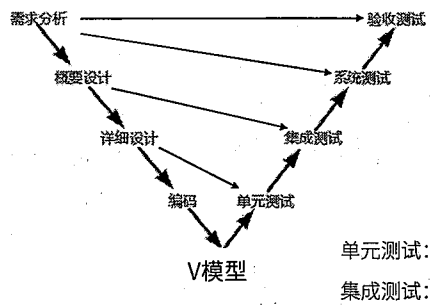



希赛
软件测试－测试用例设计
黑盒测试法
白盒测试法
等价类划分
边界值分析
错误推测
因果图
基本路径测试
循环覆盖测试
逻辑覆盖测试
• 语句覆盖
• 判定覆盖
• 条件覆盖
• 条件判定覆盖
• 修正的条件判断覆盖
• 条件组合覆盖
• 点覆盖
• 边覆盖
• 路径覆盖


希赛
软件测试－测试阶段
冒烟测试
➢ 单元测试
➢ 集成测试
➢ 确认测试
➢ 系统测试
✓ 一次性组装
✓ 增量式组装
• 负载测试
• 强度测试
• 容量测试
✓ 恢复测试
✓ 安全性测试
✓ 压力测试
✓ 性能测试
✓ 可靠性测试
✓ 可用性测试
✓ 可维护性测试
✓ 安装测试
✓ 内部确认测试
✓ Alpha 测试
✓ Beta 测试
✓ 验收测试
✓ 自顶向下
✓ 自底向上
✓ 混合式
测试用例
驱动模块
测试结果
被测模块
桩模块
桩模块
桩模块

希赛

软件测试－测试阶段



单元测试：模块测试，模块功能、性能、接口等

集成测试：模块间的接口

确认测试：验证软件与需求的一致性。内部确认测试、Alpha测试、Beta测试，验收测试

系统测试：真实环境下，验证完整的软件配置项能否和系统正确连接。

回归测试：测试软件变更之后，变更部分的正确性对变更需求的符合性。

希赛

软件测试－面向对象的测试

★ 算法层（单元测试）：包括等价类划分测试、组合功能测试（基于判定表的测试）、递归函数测试和多态消息测试

★ 类层（模块测试）:包括不变式边界测试、模态类测试和非模态类测试

★ 模板层/类树层（集成测试）：包括多态服务测试和展平测试

★ 系统层（系统测试）

软件调试

★ 软件调试方法

- 蛮力法：主要思想是“通过计算机找错”，低效，耗时
- 回溯法：从出错处人工沿控制流程往回追踪，直至发现出错的根源。复杂程序由于回溯路径多，难以实施
- 原因排除法：主要思想是演绎和归纳，用二分法实现

★ 软件调试与测试的区别

- 测试的目的是找出存在的错误，而调试的目的是定位错误并修改程序以修正错误
- 调试是测试之后的活动，测试和调试在目标、方法和思路上都有所不同
- 测试从一个已知的条件开始，使用预先定义的过程，有预知的结果；调试从一个未知的条件开始，结束的过程不可预计
- 测试过程可以事先设计，进度可以事先确定；调试不能描述过程或持续时间

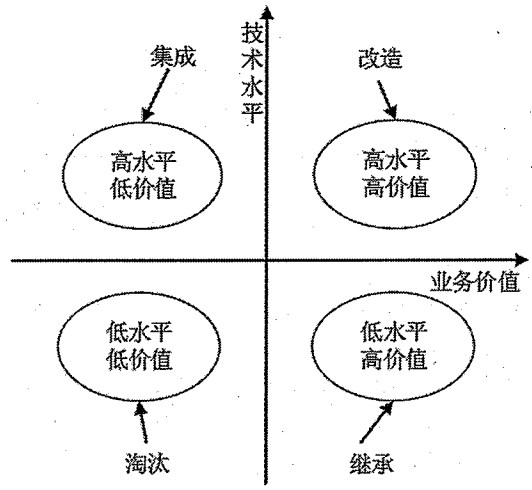
系统运行与软件维护

★ 系统转换计划

★ 软件维护

希赛

系统转换计划－遗留系统演化策略



希赛

系统转换计划－遗留系统演化策略

1. 淘汰策略

遗留系统的技术含量较低，且具有较低的业务价值。对遗留系统的完全淘汰是企业资源的根本浪费，系统分析师应该善于“变废为宝”，通过对遗留系统功能的理解和借鉴，可以帮助新系统的设计，降低新系统开发的风险。

2. 继承策略

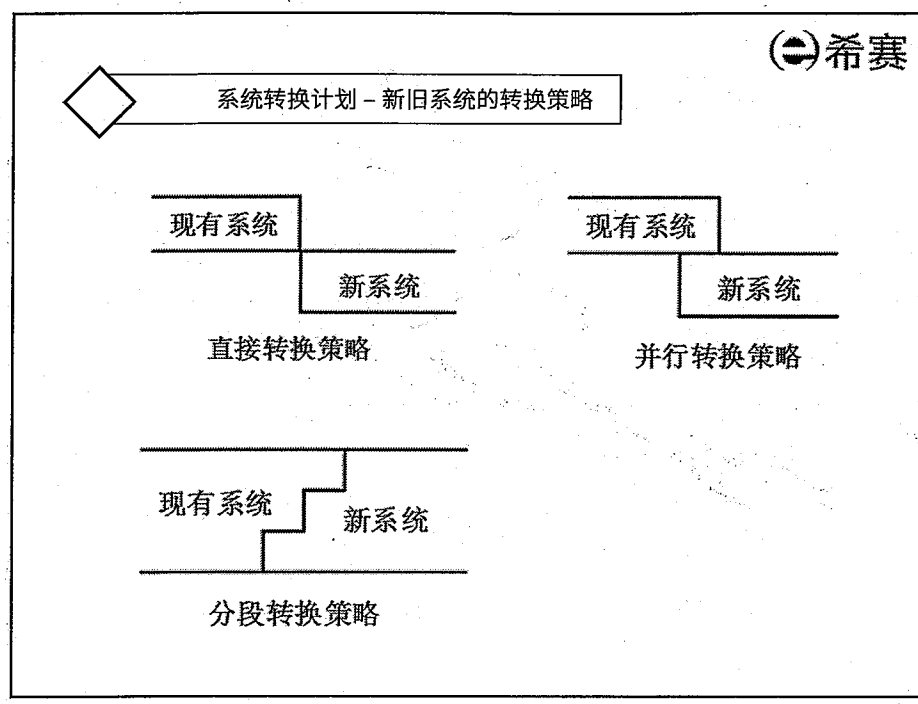
遗留系统的技术含量较低，已经满足企业运作的功能或性能要求，但具有较高的商业价值，目前企业的业务尚紧密依赖该系统。对这种遗留系统的演化策略为继承。在开发新系统时，需要完全兼容遗留系统的功能模型和数据模型。为了保证业务的连续性，新老系统必须并行运行一段时间，再逐渐切换到新系统上运行。

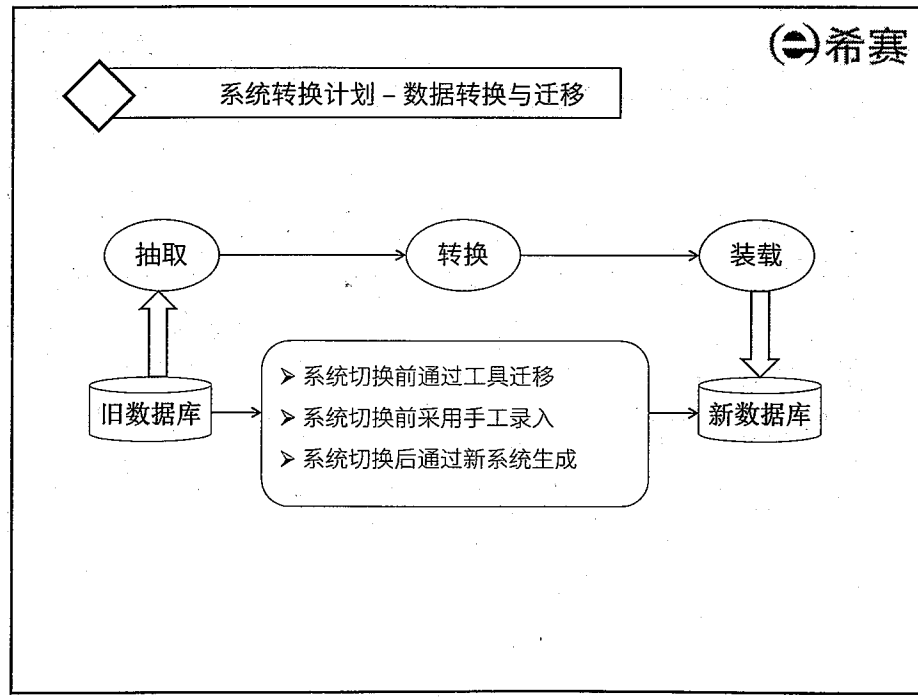
3. 改造策略

遗留系统具有较高的业务价值，基本上能够满足企业业务运作和决策支持的需要。这种系统可能建成的时间还很短，对这种遗留系统的演化策略为改造。改造包括系统功能的增强和数据模型的改造两个方面。系统功能的增强是指在原有系统的基础上增加新的应用要求，对遗留系统本身不做改变；数据模型的改造是指将遗留系统的旧的数据模型向新的数据模型的转化。

4. 集成策略

遗留系统的技术含量较高，但其业务价值较低，可能只完成某个部门（或子公司）的业务管理。这种系统在各自的局部领域里工作良好，但对于整个企业来说，存在多个这样的系统，不同的系统基于不同的平台、不同的数据模型，形成了一个个信息孤岛，对这种遗留系统的演化策略为集成。


希赛
系统转换计划－新旧系统的转换策略
现有系统
新系统
直接转换策略
现有系统
新系统
并行转换策略
现有系统
新系统
分段转换策略


希赛
系统转换计划－数据转换与迁移
抽取
转换
装载
旧数据库
➢ 系统切换前通过工具迁移
➢ 系统切换前采用手工录入
➢ 系统切换后通过新系统生成
新数据库

希赛

系统运行与维护

★ 正确性维护：指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。

★ 适应性维护：指使应用软件适应信息技术变化和管理需求变化而进行的修改。企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。

★ 完善性维护：扩充功能和改善性能而进行的修改。对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。

★ 预防性维护：为了改进应用软件的可靠性和可维护性，为了适应未来的软硬件环境的变化，应主动增加预防性的新的功能，以使用系统适应各类变化而不被淘汰。如将专用报表功能改成通用报表生成功能，以适应将来报表格式的变化。

希赛

例题讲解

● 软件的维护并不只是修正错误。为了满足用户提出的增加新功能、修改现有功能以及一般性的改进要求和建议，需要进行（ ），它是软件维护工作的主要部分；软件测试不可能揭露旧系统中所有潜在的错误，所以这些程序在使用过程中还可能发生错误，诊断和更正这些错误的过程称为（ ）；为了改进软件未来的可维护性或可靠性，或者为了给未来的改进提供更好的基础而对软件进行修改，这类活动称为（ ）。

A. 完善性维护　B. 适应性维护　C. 预防性维护　D. 改正性维护

A. 完善性维护　B. 适应性维护　C. 预防性维护　D. 改正性维护

A. 完善性维护　B. 适应性维护　C. 预防性维护　D. 改正性维护

希赛

软件架构设计

希赛

课程内容提要

- 软件架构的概念（★★★）
- 软件架构风格（★★★★★）
- 架构描述语言ADL（★★★）
- 特定领域软件架构（★★★）
- 基于架构的软件开发（★★★★）
- 软件质量属性（★★★★★）
- 软件架构评估（★★★★★）
- 软件产品线（★★★）
- 构件与中间件技术（★★★★）
- Web架构设计（★★★★★）

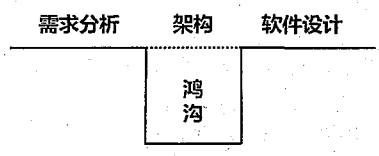
希赛

软件架构

概念 → 风格 → 设计 → 评估

希赛

软件架构的概念 – 软件架构的概念

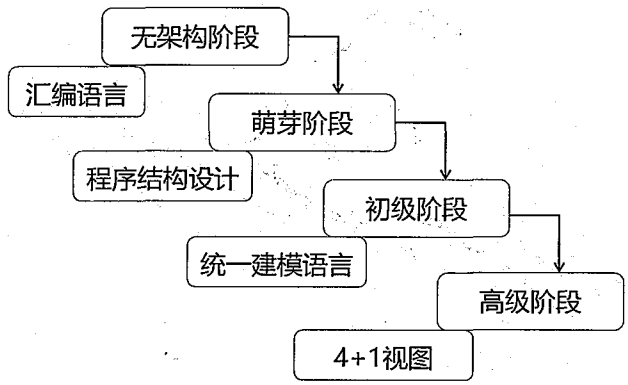


架构设计就是需求分配，即将满足需求的职责分配到组件上。

- ✓ 软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义一个系统家族，即一个体系结构定义一个词汇表和一组约束。词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。
- ✓ 软件架构为软件系统提供了一个结构、行为和属性的高级抽象，由构成系统的元素的描述、这些元素的相互作用、指导元素集成的模式以及这些模式的约束组成。
- ✓ 软件架构是项目干系人进行交流的手段，明确了对系统实现的约束条件，决定了开发和维护组织的组织结构，制约着系统的质量属性
- ✓ 软件架构使推理和控制的更改更加简单，有助于循序渐进的原型设计，可以作为培训的基础
- ✓ 软件架构是可传递和可复用的模型，通过研究软件架构可能预测软件的质量

希赛

软件架构的概念 – 软件架构的发展史

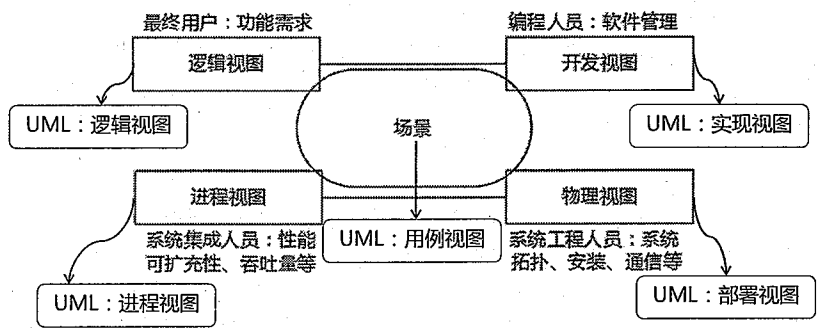


希赛

软件架构的概念 – 软件架构建模

- ✓ 结构模型：以架构的构件、连接件和其他概念来刻画结构
- ✓ 框架模型：不太侧重描述结构的细节而更侧重于整体的结构
- ✓ 动态模型：系统的“大颗粒”的行为性质
- ✓ 过程模型：构建系统的步骤和过程
- ✓ 功能模型：由一组功能构件按层次组成，下层向上层提供服务

软件架构的概念 – 软件架构建模



软件架构风格

- 架构设计的一个核心问题是能否达到架构级的软件复用
- 架构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个构件有效地组织成一个完整的系统
- 架构风格定义了用于描述系统的术语表和一组指导构建系统的规则

- 数据流风格：批处理序列、管道-过滤器
- 调用/返回风格：主程序/子程序、面向对象、层次结构
- 独立构件风格：进程通信、事件驱动系统（隐式调用）
- 虚拟机风格：解释器、基于规则的系统
- 仓库风格：数据库系统、超文本系统、黑板系统

希赛

软件架构风格－数据流风格

◆ **批处理序列**

- 构件为一系列固定顺序的计算单元，构件之间只通过数据传递交互。每个处理步骤是一个独立的程序，每一步必须在其前一步结束后才能开始，数据必须是完整的，以整体的方式传递。

◆ **管道-过滤器**

- 每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流。这个过程通常是通过对输入数据流的变换或计算来完成的，包括通过计算和增加信息以丰富数据、通过浓缩和删除以精简数据、通过改变记录方式以转化数据和递增地转化数据等。这里的构件称为过滤器，连接件就是数据流传输的管道，将一个过滤器的输出传到另一个过滤器的输入。
- 早期编译器就是采用的这种架构。要一步一步处理的，均可考虑采用此架构风格。

希赛

软件架构风格－调用/返回风格

◆ **主程序/子程序**

- 单线程控制，把问题划分为若干个处理步骤，构件即为主程序和子程序，子程序通常可合成为模块。过程调用作为交互机制，即充当连接件的角色。调用关系具有层次性，其语义逻辑表现为主程序的正确性取决于它调用的子程序的正确性

◆ **面向对象**

- 构件是对象，对象是抽象数据类型的实例。在抽象数据类型中，数据的表示和它们的相应操作被封装起来，对象的行为体现在其接受和请求的动作。连接件即是对象间交互的方式，对象是通过函数和过程的调用来交互的

◆ **层次结构**

- 构件组织成一个层次结构，连接件通过决定层间如何交互的协议来定义。每层为上一层提供服务，使用下一层的服务，只能见到与自己邻接的层。通过层次结构，可以将大的问题分解为若干个渐进的小问题逐步解决，可以隐藏问题的复杂度。修改某一层，最多影响其相邻的两层（通常只能影响上层）

希赛

软件架构风格－调用/返回风格（层次结构）

◆ **优点：**

1、这种风格支持基于可增加抽象层的设计，允许将一个复杂问题分解成一个增量步骤序列的实现。

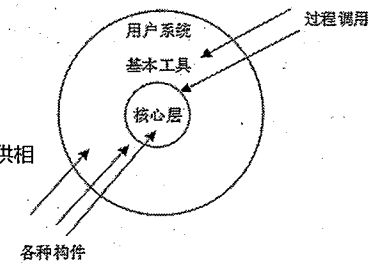
2、不同的层次处于不同的抽象级别：

越靠近底层，抽象级别越高；

越靠近顶层，抽象级别越低；

3、由于每一层最多只影响两层，同时只要给相邻层提供相同的接口，允许每层用不同的方法

实现，同样为软件复用提供了强大的支持。



◆ **缺点：**

1、并不是每个系统都可以很容易地划分为分层的模式。

2、很难找到一个合适的、正确的层次抽象方法。

希赛

软件架构风格－独立构件风格

◆ **进程通信**

- 构件是独立的过程，连接件是消息传递。构件通常是命名过程，消息传递的方式可以是点对点、异步或同步方式，以及远程过程（方法）调用等。

◆ **事件驱动系统（隐式调用）**

- 构件不直接调用一个过程，而是触发或广播一个或多个事件。构件中的过程在一个或多个事件中注册，当某个事件被触发时，系统自动调用在这个事件中注册的所有过程。一个事件的触发就导致了另一个模块中的过程调用。这种风格中的构件是匿名的过程，它们之间交互的连接件往往是以过程之间的隐式调用来实现的。主要优点是为软件复用提供了强大的支持，为构件的维护和演化带来了方便；其缺点是构件放弃了对系统计算的控制。

希赛

软件架构风格－虚拟机风格

◆ **解释器**

- 解释器通常包括一个完成解释工作的解释引擎、一个包含将被解释的代码的存储区、一个记录解释引擎当前工作状态的数据结构，以及一个记录源代码被解释执行的进度的数据结构。具有解释器风格的软件中含有一个虚拟机，可以仿真硬件的执行过程和一些关键应用，其缺点是执行效率比较低。

◆ **基于规则的系统**

- 基于规则的系统包括规则集、规则解释器、规则/数据选择器和工作内存，一般用在人工智能领域和DSS中。

希赛

软件架构风格－仓库风格（以数据为中心的风格）

◆ **数据库系统**

- 构件主要有两大类，一类是中央共享数据源，保存当前系统的数据状态；另一类是多个独立处理单元，处理单元对数据元素进行操作。

◆ **黑板系统**

- 包括知识源、黑板和控制三部分。知识源包括若干独立计算的不同单元，提供解决问题的知识。知识源响应黑板的变化，也只修改黑板；黑板是一个全局数据库，包含问题域解空间的全部状态，是知识源相互作用的唯一媒介；知识源响应是通过黑板状态的变化来控制的。黑板系统通常应用在对于解决问题没有确定性算法的软件中（信号处理、问题规划和编译器优化等）。

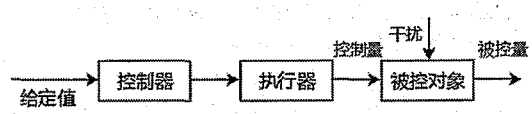
◆ **超文本系统**

- 构件以网状链接方式相互连接，用户可以在构件之间进行按照人类的联想思维方式任意跳转到相关构件。超文本是一种非线性的网状信息组织方法，它以结点为基本单位，链作为结点之间的联想式关联。超文本系统通常应用在互联网领域。

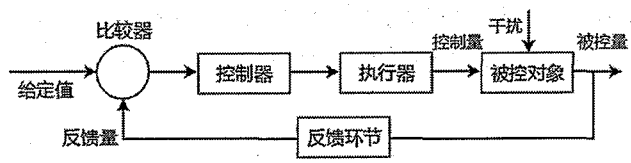
现代集成编译环境一般采用这种架构风格。

软件架构风格－闭环控制架构（过程控制）

◆ 当软件被用来操作一个物理系统时，软件与硬件之间可以粗略地表示为一个反馈循环，这个反馈循环通过接受一定的输入，确定一系列的输出，最终使环境达到一个新的状态。适合于嵌入式系统，涉及连续的动作与状态。



开环控制系统



闭环控制系统

软件架构风格－闭环控制架构（过程控制）

◆ Java程序可以做到“一次编写，到处运行”，从架构风格上看符合（　　）风格的特点。

◆ 在网络通信中，进行包的解析，一般先进行包头的分离，然后进行报文解析及后续处理，根据这一特点，选用（　　）风格最合适。

◆ 某公司欲开发一个基于图形用户界面的集成调试器。该调试器的编辑器和变量监视器可以设置调试断点。当调试器在断点处暂停运行时，编辑程序可以自动卷屏到断点，变量监视器刷新变量数值。针一对这样的功能描述，采用（　　）的架构风格最为合适。

◆ 某游戏公司欲开发一个大型多人即时战略游戏，游戏设计的目标之一是能够支持玩家自行创建战役地图，定义游戏对象的行为和之间的关系。针对该目标，公司应该采用（　　）架构风格最为合适。（四选一：管道-过滤器、隐式调用、主程序-子程序、解释器）

◆ 某公司承接了一个开发家用空调自动调温器的任务，调温器测量外部空气温度，根据设定的期望温度控制空调的开关。根据该需求，公司应采用（　　）架构风格最为合适。（四选一：解释器、过程控制、分层、管道-过滤器）

希赛

软件架构风格－架构风格判断

- 某公司欲开发一个语音识别系统，语音识别的主要过程包括分割原始语音信号、识别音素、产生候选词、判定语法片断、提供语义解释等。每个过程都需要进行基于先验知识的条件判断并进行相应的识别动作。针对该系统的特点，采用（　）架构风格最为合适。（四选一：解释器、面向对象、黑板、隐式调用）
- 某公司欲开发一个漫步者机器人，用来完成火星探测任务。机器人的控制者首先定义探测任务和任务之间的时序依赖性，机器人接受任务后，需要根据自身状态和外界环境进行动态调整，最终自动完成任务。针对这些需求，该机器人应该采用（ ）架构风格最为合适。（四选一：解释器、主程序-子程序、隐式调用、管道-过滤器）
- Windows操作系统在图形用户界面处理方面采用的核心架构风格是（ ）风格。

希赛

软件架构风格－练习题

“编译器”是一种非常重要的基础软件，其核心功能是对源代码形态的单个或一组源程序依次进行预处理、词法分析、语法分析、语义分析、代码生成、代码优化等处理，最终生成目标机器的可执行代码。考虑以下与编译器相关的软件架构设计场景：传统的编译器设计中，上述处理过程都以独立功能模块的形式存在，程序源代码作为一个整体，依次在不同模块中进行传递，最终完成编译过程。针对这种设计思路，传统的编译器采用（1）架构风格比较合适。

随着编译、链接、调试、执行等开发过程的一体化趋势发展，集成开发环境（IDE）随之出现。IDE集成了编译器、连接器、调试器等多种工具，支持代码的增量修改与处理，能够实现不同工具之间的信息交互，覆盖整个软件开发生命周期。针对这种需求，IDE采用（2）架构风格比较合适。IDE强调交互式编程，用户在修改程序代码后，会同时触发语法高亮显示、语法错误提示、程序结构更新等多种功能的调用与结果呈现，针对这种需求，通常采用（3）架构风格比较合适。

某公司已经开发了一款针对某种嵌入式操作系统专用编程语言的IDE，随着一种新的嵌入式操作系统上市并迅速占领市场，公司决定对IDE进行适应性改造，支持采用现有编程语言进行编程，生成符合新操作系统要求的运行代码，并能够在现有操作系统上模拟出新操作系统的运行环境，以支持代码调试工作。针对上述要求，为了使IDE能够生成符合新操作系统要求的运行代码，采用基于（4）的架构设计策略比较合适；为了模拟新操作系统的运行环境，通常采用（5）架构风格比较合适。

- （1）A．管道-过滤器　B．顺序批处理　C．过程控制　D．独立进程
- （2）A．规则引擎　B．解释器　C．数据共享　D．黑板
- （3）A．隐式调用　B．显式调用　C．主程序-子程序　D．层次结构
- （4）A．代理　B．适配　C．包装　D．模拟
- （5）A．隐式调用　B．仓库结构　C．基于规则　D．虚拟机

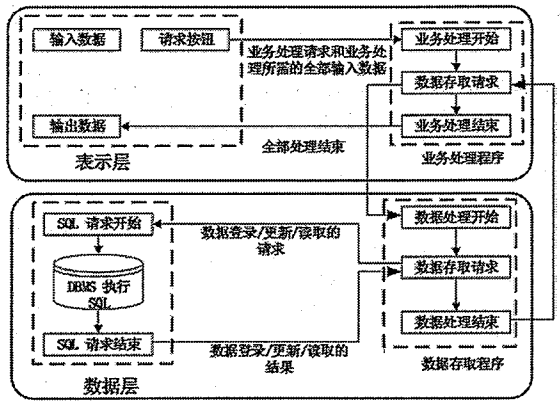
希赛

软件架构风格 – 层次架构风格

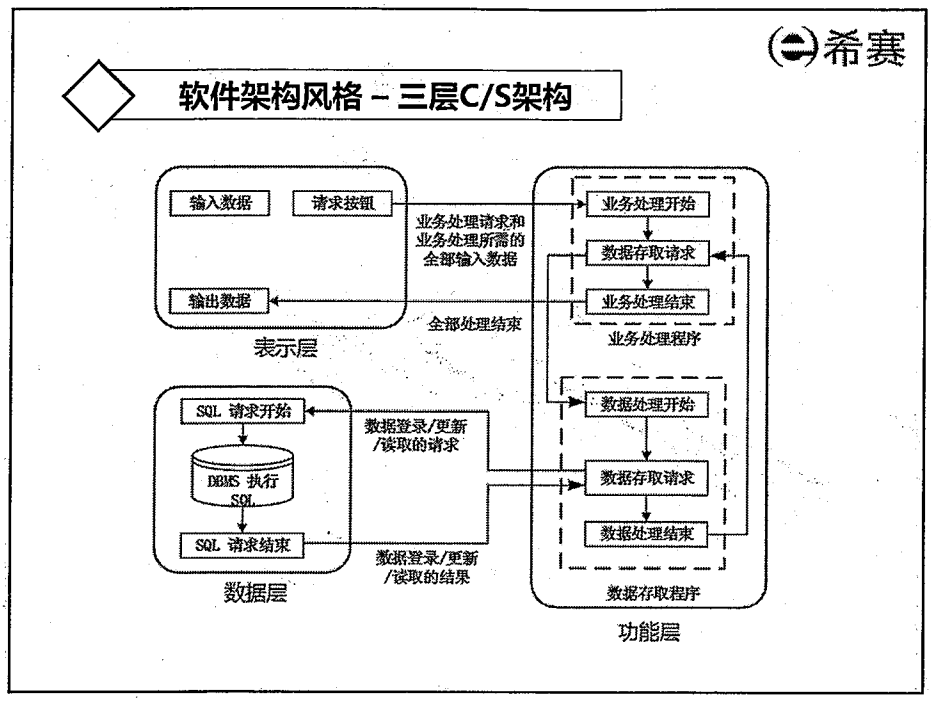
两层C/S → 三层C/S → 三层B/S → 混合架构

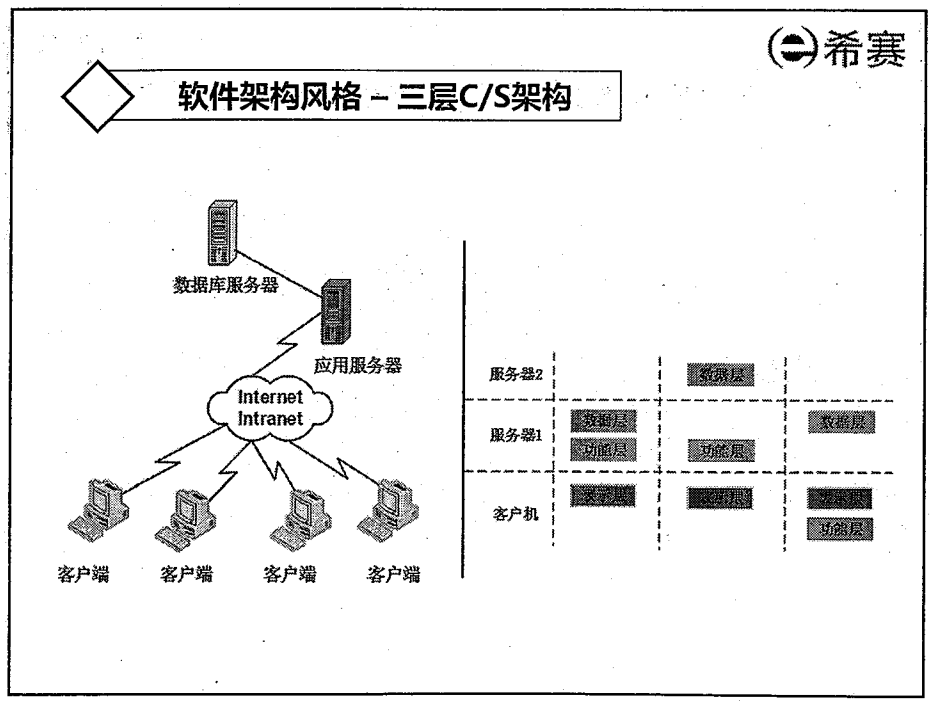
希赛

软件架构风格 – 两层C/S架构



- ✓ 开发成本较高
- ✓ 客户端程序设计复杂
- ✓ 信息内容和形式单一
- ✓ 用户界面风格不一
- ✓ 软件移植困难
- ✓ 软件维护和升级困难
- ✓ 新技术不能轻易应用

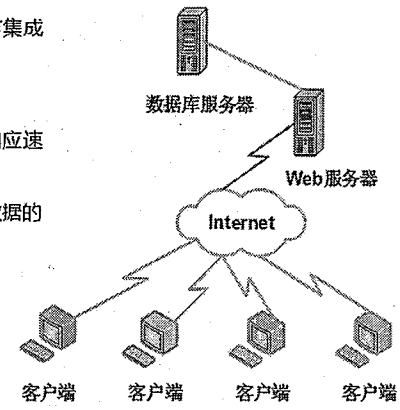

希赛
软件架构风格－三层C/S架构
输入数据
请求按钮
输出数据
表示层
业务处理请求和业务处理所需的全部输入数据
全部处理结束
业务处理开始
数据存取请求
业务处理结束
业务处理程序
SQL 请求开始
DBMS 执行 SQL
SQL 请求结束
数据登录/更新/读取的请求
数据登录/更新/读取的结果
数据层
数据处理开始
数据存取请求
数据处理结束
数据存取程序
功能层


希赛
软件架构风格－三层C/S架构
数据库服务器
应用服务器
Internet
Intranet
客户端
客户端
客户端
客户端
服务器2
服务器1
客户机
数据层
功能层

希赛

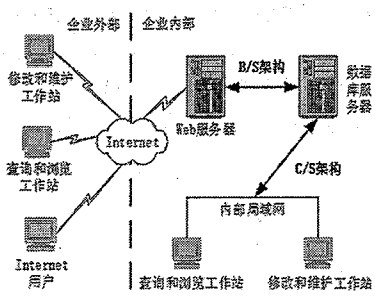
软件架构风格－三层B/S架构

- ✓ B/S架构缺乏对动态页面的支持能力，没有集成有效的数据库处理功能
- ✓ B/S架构的安全性难以控制
- ✓ 采用B/S架构的应用系统，在数据查询等响应速度上，要远远低于C/S架构
- ✓ B/S架构的数据提交一般以页面为单位，数据的动态交互性不强，不利于OLTP应用

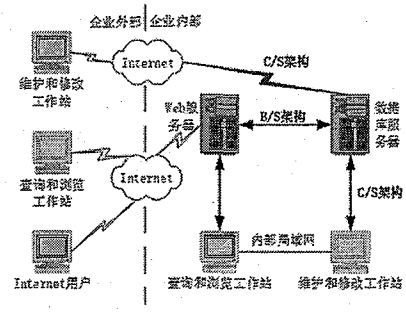


希赛

软件架构风格－混合架构风格



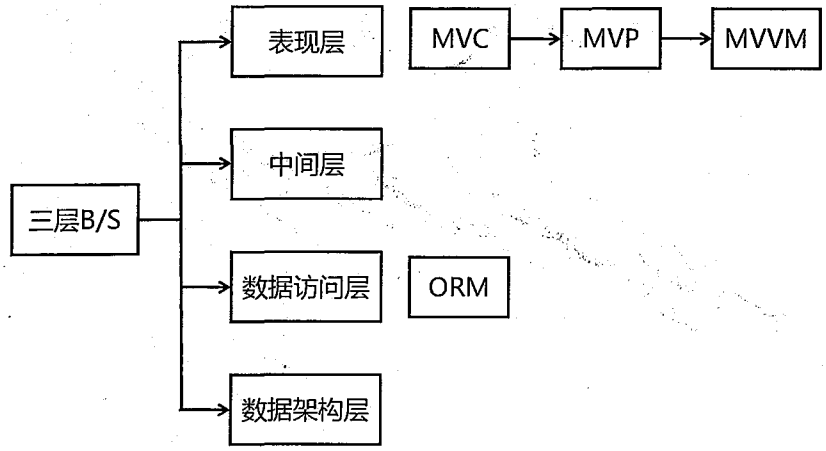
内外有别模型



查改有别模型

希赛

软件架构风格－层次架构风格



希赛

软件架构风格－MVC架构风格

- Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。
- View（视图）是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。
- Controller（控制器）是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

★ J2EE体系结构中：

★ 视图（View）：JSP

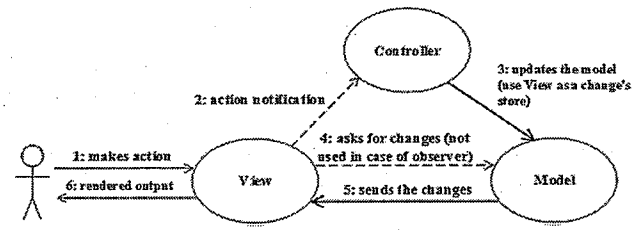
★ 控制（Controller）：Servlet

★ 模型（Model）：Entity Bean、Session Bean

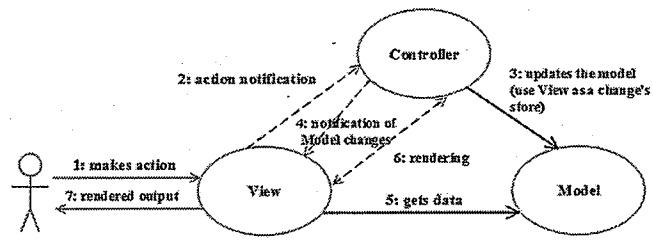
希赛

软件架构风格 – MVC架构风格

主动MVC



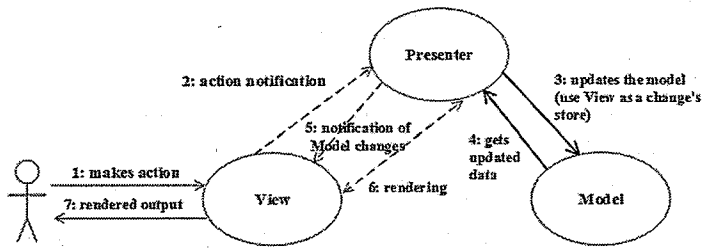
被动MVC



希赛

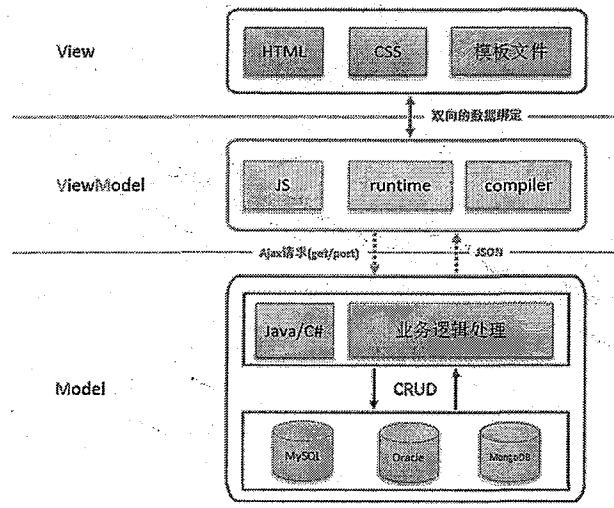
软件架构风格 – MVP架构风格

- MVP是MVC的变种
- MVP实现了V与M之间的解耦（V不直接使用M，修改V不会影响M）
- MVP更好的支持单元测试（业务逻辑在P中，可以脱离V来测试这些逻辑；可以将一个P用于多个V，而不需要改变P的逻辑）
- MVP中V要处理界面事件，业务逻辑在P中，MVC中界面事件由C处理



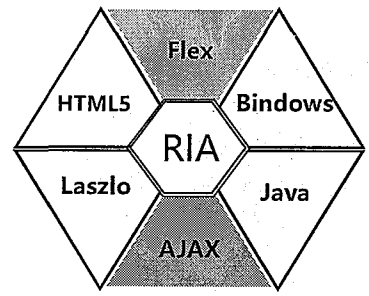
希赛

软件架构风格 – MVVM架构风格



希赛

软件架构风格 – 富互联网应用(RIA)

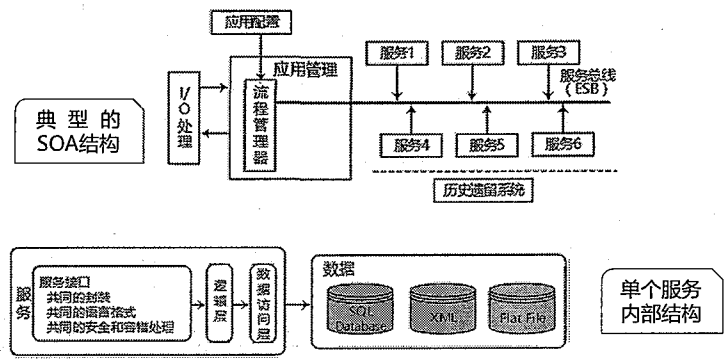


- ✓ RIA结合了C/S架构反应速度快、交互性强的优点，以及B/S架构传播范围广及容易传播的特性
- ✓ RIA简化并改进了B/S架构的用户交互
- ✓ 数据能够被缓存在客户端，从而可以实现一个比基于HTML的响应速度更快且数据往返于服务器的次数更少的用户界面

希赛

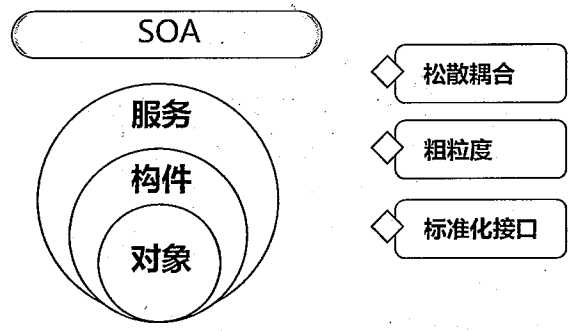
软件架构风格－基于服务的架构(SOA)

✓ 服务是一种为了满足某项业务需求的操作、规则等的逻辑组合，它包含一系列有序活动的交互，为实现用户目标提供支持



希赛

软件架构风格－基于服务的架构(SOA)



✓ 服务构件粗粒度，传统构件细粒度居多

✓ 服务构件的接口是标准的，主要是WSDL接口，传统构件常以具体API形式出现

✓ 服务构件的实现与语言无关，传统构件绑定某种特定语言

✓ 服务构件可以通过构件容器提供QoS的服务，传统构件完全由程序代码直接控制

希赛

软件架构风格 – SOA – 关键技术

功能	协议
发现服务	UDDI、DISCO
描述服务	WSDL、XML Schema
消息格式层	SOAP、REST
编码格式层	XML（DOM，SAX）
传输协议层	HTTP、TCP/IP 、SMTP等

希赛

软件架构风格 – SOA – 关键技术 – XML

- ✓ 实现不同数据的集成
- ✓ 使用于多种应用环境
- ✓ 客户端数据处理与计算
- ✓ 数据显示多样化
- ✓ 局部数据更新
- ✓ Web Service
- ✓ 数据集成
- ✓ 软件架构描述
- ✓ 软件架构演化
- ✓ RIA
- ✓ XML数据库

```
<?xml version="1.0"?>
<visit>
    <to>alluser</to>
    <from>educity.cn</from>
    <heading>welcome</heading>
    <body>Welcome to the best online education website!</body>
</visit>
```

希赛

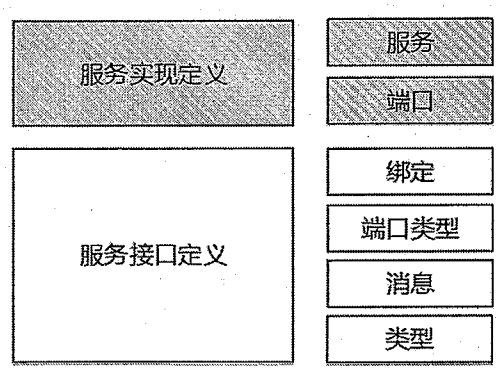
软件架构风格 – SOA – 关键技术 – REST

- ✓ HTTP + XML进行基于 Web 通信的技术
- ✓ 简单性，缺少严格配置文件
- ✓ 只支持几个操作(POST、GET、PUT、DELETE)
- ✓ 强调信息本身，称为资源

- 网络上的所有事物都被抽象为资源
- 每个资源对应一个唯一的资源标识
- 通过通用的连接器接口对资源进行操作
- 对资源的各种操作不会改变资源标识
- 所有的操作都是无状态的

希赛

软件架构风格 – SOA – 关键技术 – WSDL



希赛

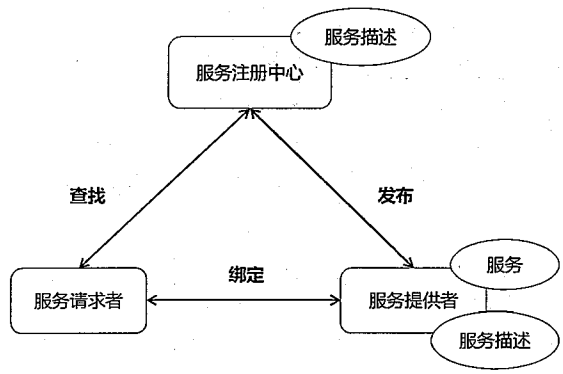
软件架构风格 – SOA – 关键技术 – UDDI&DISCO

- ✓ UDDI数据模型。一个用于描述企业和服务的XML Schema
- ✓ UDDI API。一组用于查找或发布UDDI数据的方法，基于SOAP
- ✓ UDDI注册服务。一种基础设施，对应着服务注册中心的角色

- ✓ DISCO 可以定义一个文档格式和询问算法，发现给定服务器上公开的 服务
- ✓ DISCO 能够发现每个 服务的功能(通过文档)，以及如何与它进行交互(通过 WSDL)
- ✓ 如果要使用 DISCO 发布一个已部署的 服务，只需创建一个 .disco 文件

希赛

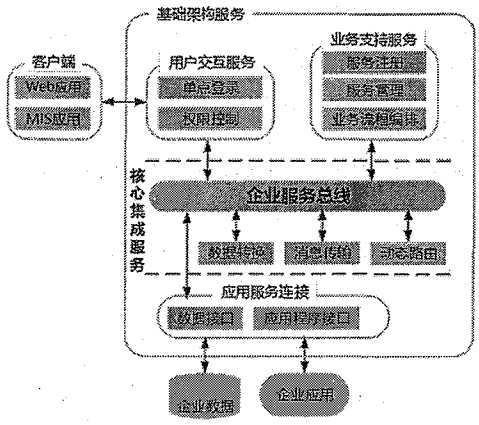
软件架构风格 – SOA的实现方式 – Web Service



- ✓ 底层传输层
- ✓ 服务通信协议层
- ✓ 服务描述层
- ✓ 服务层
- ✓ 业务流程层
- ✓ 服务注册层

希赛

软件架构风格 – SOA的实现方式 – ESB



- ✓ 提供位置透明性的消息路由和寻址服务
- ✓ 提供服务注册和命名的管理功能
- ✓ 支持多种的消息传递范型
- ✓ 支持多种可以广泛使用的传输协议
- ✓ 支持多种数据格式及其相互转换
- ✓ 提供日志和监控功能

希赛

软件架构风格 – SOA的实现方式 – 服务注册表

- ✓ 服务注册：应用开发者（服务提供者）向注册表公布服务的功能
- ✓ 服务位置：服务使用者（服务应用开发者），帮助他们查询注册服务，寻找符合自身要求的服务
- ✓ 服务绑定：服务使用者利用检索到的服务接口来编写代码，所编写的代码将与注册的服务绑定、调用注册的服务，以及与它们实现互动

希赛

软件架构风格－微服务

微服务顾名思义，就是很小的服务，所以它属于面向服务架构的一种。

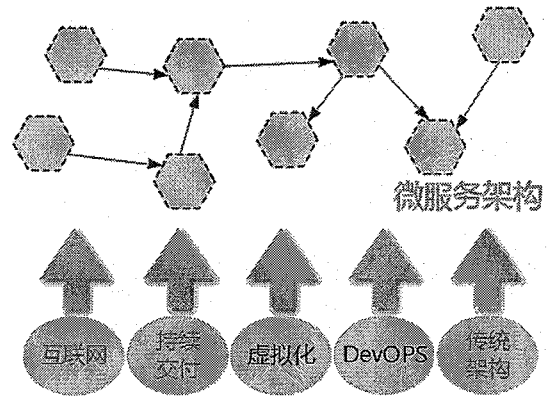
微服务架构是一种架构模式，它提倡将单一应用程序划分成一组小的服务，服务之间互相协调、互相配合，为用户提供最终价值。每个服务运行在其独立的进程中，服务与服务间采用轻量级的通信机制互相沟通（通常是基于HTTP协议的RESTful API）。每个服务都围绕着具体业务进行构建，并且能够被独立的部署到生产环境、类生产环境等。另外，应当尽量避免统一的、集中式的服务管理机制，对具体的一个服务而言，应根据业务上下文，选择合适的语言、工具对其进行构建。

◆ **特点**

- 小, 且专注于做一件事情
- 轻量级的通信机制
- 松耦合、独立部署

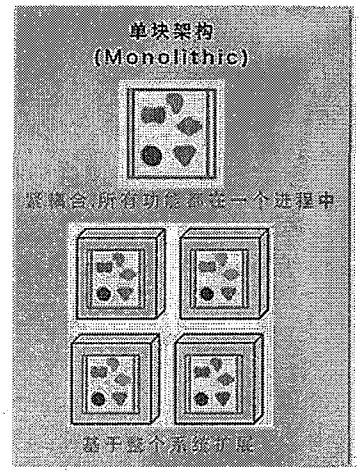
希赛

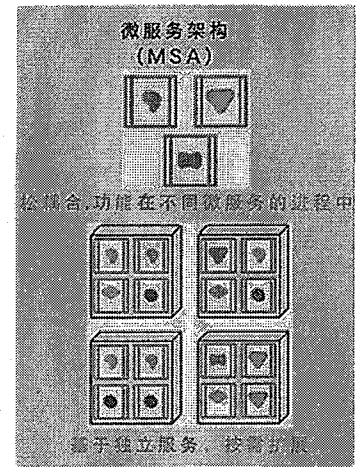
软件架构风格－微服务



希赛

软件架构风格 – 微服务





希赛

软件架构风格 – 微服务

◆ **微服务的优势**

(1) 技术异构性

(2) 弹性

(3) 扩展

(4) 简化部署

(5) 与组织结构相匹配

(6) 可组合性

(7) 对可替代性的优化

希赛

软件架构风格 – 微服务

◆ **微服务面临的挑战**

（1）分布式系统的复杂度

（2）运维成本

（3）部署自动化

（4）DevOps与组织结构

（5）服务间依赖测试

（6）服务间依赖管理

希赛

软件架构风格 – 微服务与SOA

微服务	SOA
能拆分的就拆分	是整体的，服务能放一起的都放一起
纵向业务划分	是水平分多层
由单一组织负责	按层级划分不同部门的组织负责
细粒度	粗粒度
两句话可以解释明白	几百字只相当于SOA的目录
独立的子公司	类似大公司里面划分了一些业务单元（BU）
组件小	存在较复杂的组件
业务逻辑存在于每一个服务中	业务逻辑横跨多个业务领域
使用轻量级的通信方式，如HTTP	企业服务产总线（ESB）充当了服务之间通信的角色

希赛

软件架构风格 – 微服务与SOA

微服务架构实现	SOA实现
团队级，自底向上开展实施	企业级，自顶向下开展实施
一个系统被拆分成多个服务，粒度细	服务由多个子系统组成，粒度大
无集中式总线，松散的服务架构	企业服务总线，集中式的服务架构
集成方式简单（HTTP/REST/JSON）	集成方式复杂（ESB/WS/SOAP）
服务能独立部署	单块架构系统，相互依赖，部署复杂

希赛

架构描述语言（ADL）

✓ ADL是这样一种形式化语言，它在底层语义模型的支持下，为软件系统的概念体系结构建模提供了具体语法和概念框架。基于底层语义的工具为体系结构的表示、分析、演化、细化、设计过程等提供支持。

✓ **ADL的三个基本元素**

- 构件：计算或数据存储单元；
- 连接件：用于构件之间交互建模的体系结构构造块及其支配这些交互的规则；
- 架构配置：描述体系结构的构件与连接件的连接图。

希赛

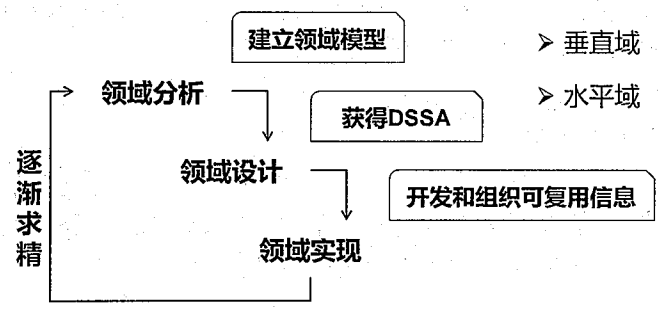
架构描述语言（ADL）

✓ **主要的架构描述语言**

- Aesop：支持体系结构风格的应用；
- MetaH：为设计者提供了关于实时电子控制软件系统的设计指导；
- C2：支持基于消息传递风格的用户界面系统的描述；
- Rapide：支持体系结构设计的模拟并提供了分析模拟结果的工具；
- SADL：提供了关于体系结构加细的形式化基础；
- Unicon：支持异构的构件和连接类型并提供了关于体系结构的高层编译器；
- Wright：支持体系结构构件之间交互的说明和分析。

希赛

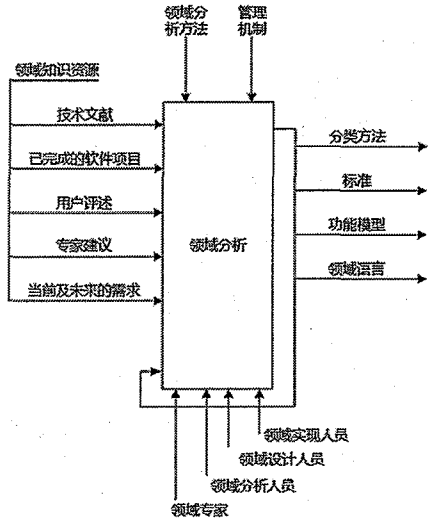
特定领域软件架构(DSSA) – 基本活动



- 垂直域
- 水平域

希赛

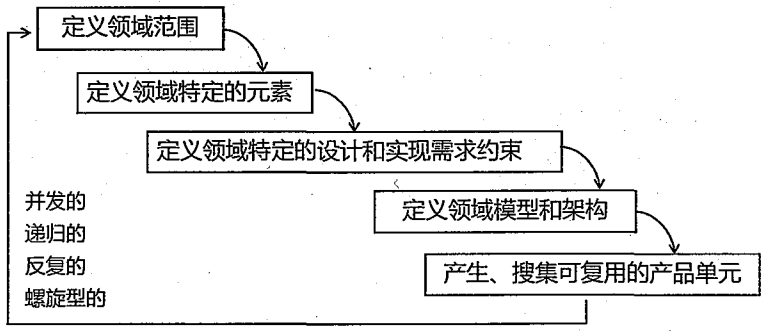
特定领域软件架构(DSSA) – 领域分析机制



- ✓ 1、领域专家：有经验的用户、从事该领域中系统的需求分析、设计、实现以及项目管理的有经验的软件工程师等。
- ✓ 领域专家的主要任务包括提供关于领域中系统的需求规约和实现的知识。
- ✓ 2、领域分析人员：领域分析人员应由具有知识工程背景的有经验的系统分析员来担任。
- ✓ 3、领域设计人员：领域设计人员应由有经验的软件设计人员来担任。
- ✓ 4、领域实现人员：领域实现人员应由有经验的程序设计人员来担任。

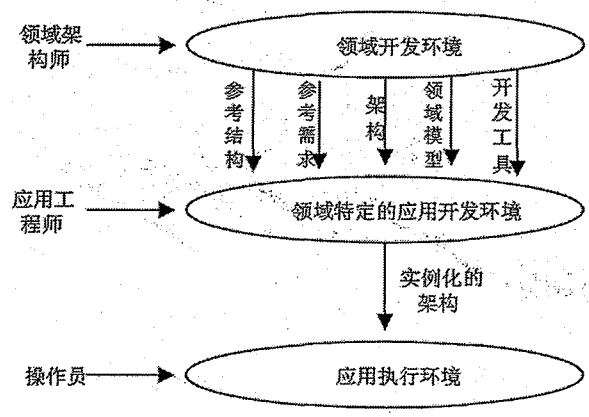
希赛

特定领域软件架构(DSSA) – 建立过程



希赛

特定领域软件架构(DSSA) – 三层次模型



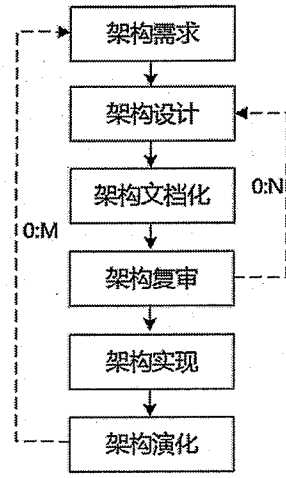
希赛

基于架构的软件开发方法 – 基于架构的软件设计(ABSD)

- ✓ ABSD方法是架构驱动，即强调由业务、质量和功能需求的组合驱动架构设计。
- ✓ 使用ABSD方法，设计活动可以从项目总体功能框架明确就开始，这意味着需求获取和分析还没有完成（甚至远远没有完成），就开始了软件设计。
- ✓ ABSD方法有三个基础。第一个基础是功能的分解。在功能分解中，ABSD方法使用已有的基于模块的内聚和耦合技术；第二个基础是通过选择架构风格来实现质量和业务需求；第三个基础是软件模板的使用。软件模板利用了一些软件系统的结构。
- ✓ ABSD方法是递归的，且迭代的每一个步骤都是清晰地定义的。因此，不管设计是否完成，架构总是清晰的，这有助于降低架构设计的随意性。

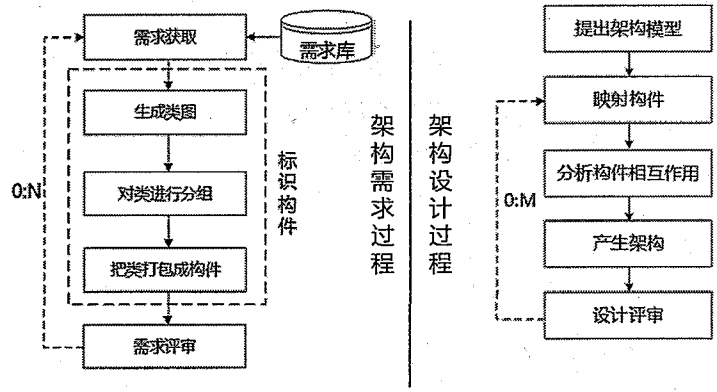
希赛

基于架构的软件开发方法 – 开发过程



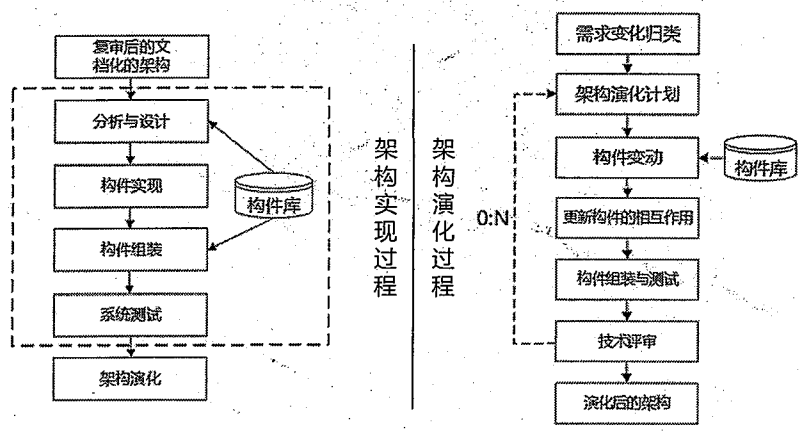
希赛

基于架构的软件开发方法 – 开发过程



希赛

基于架构的软件开发方法－开发过程



希赛

软件架构评估

- ✓ 1、为什么要进行架构评估？
- ✓ 2、架构评估到底评什么？
- ✓ 3、架构评估怎么评？

希赛

软件架构评估－质量属性

1、性能

性能（performance）是指系统的响应能力，即要经过多长时间才能对某个事件做出响应，或者在某段时间内系统所能处理的事件的个数。

代表参数：响应时间、吞吐量　　设计策略：优先级队列、资源调度

2、可靠性

可靠性（reliability）是软件系统在应用或系统错误面前，在意外或错误使用的情况下维持软件系统的功能特性的基本能力。

代表参数：MTTF、MTBF　　设计策略：冗余、心跳线

3、可用性

可用性（availability）是系统能够正常运行的时间比例。经常用两次故障之间的时间长度或在出现故障时系统能够恢复正常的速度来表示。

代表参数：故障间隔时间　　设计策略：冗余、心跳线

4、安全性

安全性（security）是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。安全性又可划分为机密性、完整性、不可否认性及可控性等特性。

设计策略：追踪审计

希赛

软件架构评估－质量属性

5、可修改性

可修改性（modifiability）是指能够快速地以较高的性能价格比对系统进行变更的能力。通常以某些具体的变更为基准，通过考察这些变更的代价衡量可修改性。

主要策略：信息隐藏

6、功能性

功能性（functionality）是系统所能完成所期望的工作的能力。一项任务的完成需要系统中许多或大多数构件的相互协作。

7、可变性

可变性（changeability）是指体系结构经扩充或变更而成为新体系结构的能力。这种新体系结构应该符合预先定义的规则，在某些具体方面不同于原有的体系结构。当要将某个体系结构作为一系列相关产品（例如，软件产品线）的基础时，可变性是很重要的。

8、互操作性

作为系统组成部分的软件不是独立存在的，经常与其他系统或自身环境相互作用。为了支持互操作性（interoperation），软件体系结构必须为外部可视的功能特性和数据结构提供精心设计的软件入口。程序和用其他编程语言编写的软件系统的交互作用就是互操作性的问题，这种互操作性也影响应用的软件体系结构。

希赛

软件架构评估－质量属性

（1）用户提交搜索请求后，系统必须在1秒内显示结果；

（2）用户信息数据库授权必须保证99.9%可用；

（3）系统由MySQL数据库升级为Oracle数据库，必须在1人月内完成；

（4）主服务器出现严重问题无法提供服务时，备用系统10分钟内能接替其工作；

（5）需要在3人周内为系统添加一种新的支付方式——支付宝；

（6）视频点播时，超清模式必须保证画面具有1280*720的分辨率；

（7）主站点断电后，需要在3秒内将访问请求重定向到备用站点。

希赛

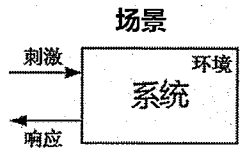
软件架构评估

风险点：系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。

敏感点是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点是影响多个质量属性的特性，是多个质量属性的敏感点。

- ✓ 基于调查问卷（检查表）的方式
- ✓ 基于度量的方式
- ✓ 基于场景的方式



评估方式	调查问卷或检查表		场景	度量
	调查问卷	检查表		
通用性	通用	特定领域	特定系统	通用或特定领域
评估者对架构的了解程度	粗略了解	无限制	中等了解	精确了解
实施阶段	早	中	中	中
客观性	主观	主观	较主观	较客观

希赛

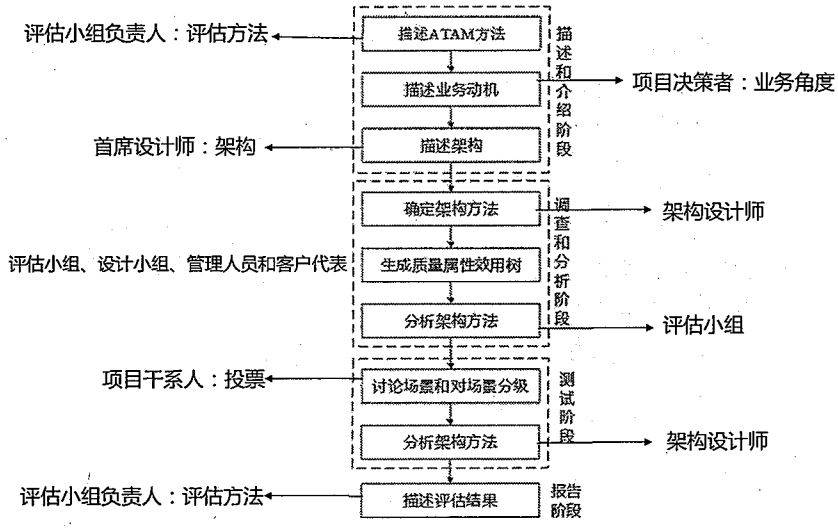
软件架构评估 – 基于场景的方式

- ✓ 确定应用领域的功能和软件架构的结构之间的映射
- ✓ 设计用于体现待评估质量属性的场景
- ✓ 分析软件架构对场景的支持程度

- ✓ 架构权衡分析法（ATAM）
- ✓ 软件架构分析法（SAAM）
- ✓ 成本效益分析法（CBAM）

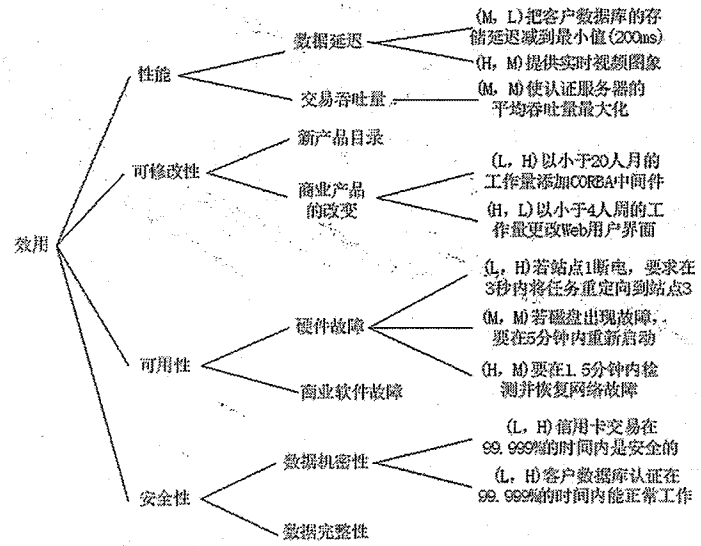
希赛

软件架构评估 – ATAM



希赛

软件架构评估 – 质量效用树



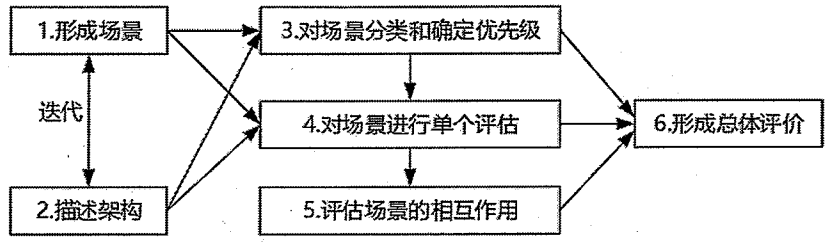
希赛

软件架构评估 – CBAM

- ✓ 整理场景
- ✓ 对场景进行求精
- ✓ 确定场景的优先级
- ✓ 分配效用
- ✓ 形成“策略-场景-响应级别”的对应关系
- ✓ 确定期望的质量属性响应级别的效用
- ✓ 计算各架构策略的总收益
- ✓ 根据受成本限制影响的投资报酬率选择架构策略

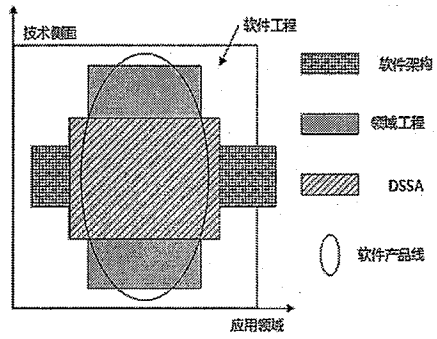
希赛

软件架构评估 – SAAM



希赛

软件产品线 – 基本概念

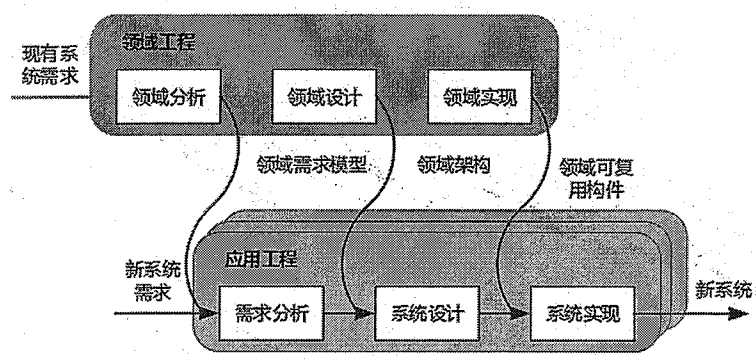


- ✓ 核心资源
- ✓ 产品集合

- ✓ 过程驱动
- ✓ 特定领域
- ✓ 技术支持
- ✓ 以架构为中心

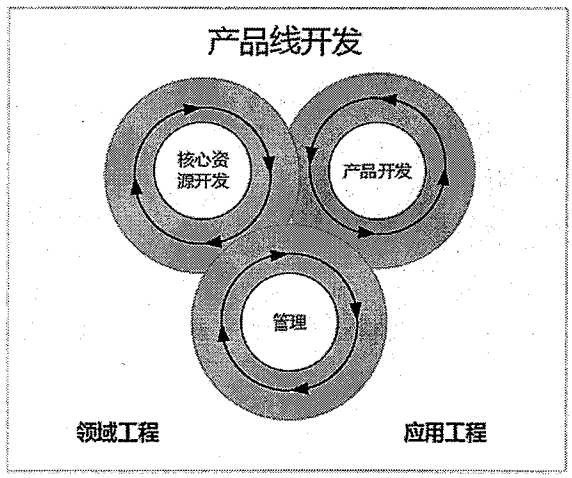
希赛

软件产品线－过程模型－双生命周期模型



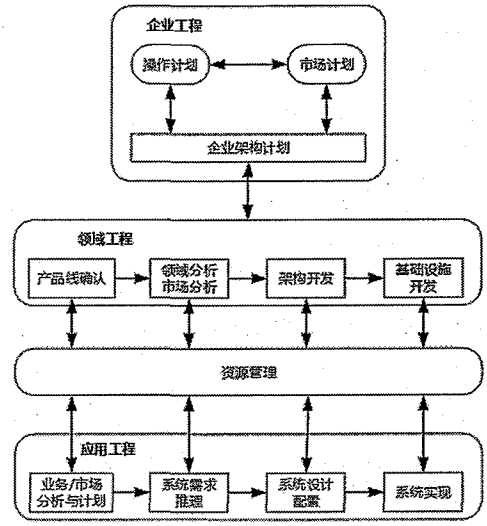
希赛

软件产品线－过程模型－SEI模型



希赛

软件产品线 – 过程模型 – 三生命周期模型



希赛

软件产品线 – 建立方式

	演化方式	革命方式
基于现有产品	基于现有产品架构设计产品线的架构，经演化现有构件，开发产品线构件	核心资源的开发基于现有产品集的需求和可预测的、将来需求的超集
全新产品线	产品线核心资源随产品新成员的需求而演化	开发满足所有预期产品线成员的需求的核心资源

★ 将现有产品演化为产品线

★ 用软件产品线替代现有产品集

★ 全新软件产品线的演化

★ 全新软件产品线的开发

希赛

软件产品线 – 组织结构

组织结构类型

★ 设立独立的核心资源小组

★ 不设立独立的核心资源小组

★ 动态的组织结构

要成功实施产品线，主要取决于以下因素

★ 对该领域具备长期和深厚的经验

★ 一个用于构建产品的好的核心资源库

★ 好的产品线架构

★ 好的管理（软件资源、人员组织、过程）支持

希赛

构件与中间件技术 – 概念

构件的定义

定义1：软件构件是一种组装单元，它具有规范的接口规约和显式的语境依赖。软件构件可以被独立地部署并由第三方任意地组装。

定义2：构件是某系统中有价值的、几乎独立的并可替换的一个部分，它在良好定义的体系结构语境内满足某清晰的功能。

定义3：构件是一个独立发布的功能部分，可以通过其接口访问它的服务。

构件的特性	对象的特性	模块的特性
1、独立部署单元； 2、作为第三方的组装单元； 3、没有（外部的）可见状态。	1、一个实例单元，具有唯一的标志。 2、可能具有状态，此状态外部可见。 3、封装了自己的状态和行为。	结构化开发的产物

希赛

构件与中间件技术 – 概念

构件系统架构特性

构件系统体系结构由一组平台决策、一组构件框架和构件框架之间的互操作设计组成。

构件框架是一种专用的体系结构（通常围绕一些关键的机制），同时，也是一组固定地作用于构件层次机制的策略。

概念框架的互操作设计包括系统体系结构连接的所有框架间的互操作的规则。

构件是一组通常需要同时部署的原子构件。 构件和原子构件之间的区别在于，大多数原子构件永远都不会被单独部署，尽管它们可以被单独部署。

一个原子构件是一个模块和一组资源。

模块是一组类和可能的非面向对象的结构体，比如过程或者函数。

资源是一个类型化的项的固定集合。

资源这个概念可以包含代码资源，进而包含模块。问题在于除了编译器编译一个模块或包生成的资源外，还可能存在其他的资源。在“纯对象”的方法中，资源是外部化的不可改变的对象——不可改变是因为构件没有持久化的标志，而且复制不能被区分。

希赛

构件与中间件技术 – 构件的复用

检索与提取构件 ⟹ 理解与评价构件 ⟹ 修改构件 ⟹ 组装构件

1、检索与提取构件

（1）基于关键字的检索。系统在图形用户界面上将构件库的关键字树形结构直观地展示给用户，复用者通过对树形结构的逐级浏览，寻找需要的关键字并提取相应的构件。

（2）刻面检索法。该方法基于刻面分类法，由三步构成，分别是构造查询、检索构件和对构件进行排序。这种方法的优点是它易于实现相似构件的查找，但复用者在构造查询时比较麻烦。

（3）超文本检索法。复用者首先给出一个或数个关键字，系统在构件的说明文档中进行精确或模糊的语法匹配，匹配成功后，向复用者列出相应的构件说明。这种方法的优点是用户界面友好，但在某些情况下复用者难以在超文本浏览过程中正确选取构件。

希赛

构件与中间件技术 – 构件的复用

检索与提取构件 ⟹ **理解与评价构件** ⟹ 修改构件 ⟹ 组装构件

2、理解与评价构件

（1）要复用构件，准确地理解构件至关重要。特别是对构件修改使用时。

（2）为达到目的，必须要求构件的开发过程遵循公共标准。

（3）一般构件库的文档中全面而准确地说明以下内容：构件的功能与行为、相关的领域知识、可适应性约束条件与例外情形、可以预见的修改部分及修改方法。

希赛

构件与中间件技术 – 构件的复用

检索与提取构件 ⟹ 理解与评价构件 ⟹ **修改构件** ⟹ 组装构件

3、修改构件

（1）理想状态是直接复用构件库中现成的构件，但大多数情况下，必须对构件进行或多或少的修改，以应对新需求。

（2）为了减少构件修改的工作量，要求开发人员尽量使构件的功能、行为和接口设计更为抽象化、通用化和参数化。这样，复用者即可通过对实参的选取来调整构件的功能或行为。如果这种调整仍不足以使构件适用于新系统，复用者就必须借助设计信息和文档来修改构件

（3）构件库中若无可修改使用的构件，则按新需求开发构件，并存入构件库。

希赛

构件与中间件技术 – 构件的复用

检索与提取构件 ⟹ 理解与评价构件 ⟹ 修改构件 ⟹ **组装构件**

4、组装构件

（1）基于功能的组装技术。基于功能的组装技术采用子程序调用和参数传递的方式将构件组装起来。它要求库中的构件以子程序/过程/函数的形式出现，并且接口说明必须清晰。当使用这种组装技术进行软件开发时，开发人员首先要对新系统进行功能分解，将系统分解为强内聚、松耦合的功能模块；然后根据各模块的功能需求提取构件，进行适应性修改后，再挂接在上述功能分解框架中。

（2）基于数据的组装技术。基于数据的组装技术首先根据当前软件问题的核心数据结构设计出一个框架，然后根据框架中各结点的需求提取构件并进行适应性修改，再将构件逐个分配至框架中的适当位置。此后，构件的组装方式仍然是传统的子程序调用与参数传递。这种组装技术也要求库中构件以子程序形式出现，但它所依赖的软件设计方法不再是功能分解，而是面向数据的设计方法，例如，Jackson系统开发方法。

（3）面向对象的组装技术。由于封装和继承特征，面向对象方法比其他软件开发方法更适合支持软件复用。在面向对象的软件开发方法中，如果从类库中检索出来的基类能够完全满足新系统的需求，则可以直接应用。否则，必须以基类为父类，生成相应的子类，以满足新系统的需求。

希赛

构件与中间件技术 – 构件的复用

检索与提取构件 ⟹ 理解与评价构件 ⟹ 修改构件 ⟹ **组装构件**

在构件组装阶段失配问题主要包括：

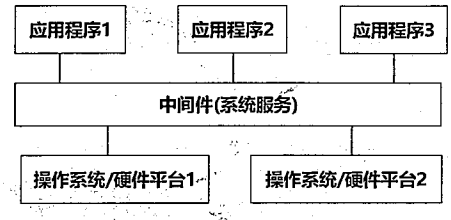
（1）由构件引起的失配，包括由于系统对构件基础设施、构件控制模型和构件数据模型的假设存在冲突引起的失配；

（2）由连接子引起的失配，包括由于系统对构件交互协议、连接子数据模型的假设存在冲突引起的失配；

（3）由于系统成分对全局体系结构的假设存在冲突引起的失配等。要解决失配问题，首先需要检测出失配问题，并在此基础上通过适当的手段消除检测出的失配问题。

希赛

构件与中间件技术 – 概念



- 中间件是一种独立的系统软件或服务程序，可以帮助分布式应用软件在不同的技术之间共享资源

★ 负责客户机与服务器之间的连接和通信，以及客户机与应用层之间的高效率通信机制

★ 提供应用层不同服务之间的互操作机制，以及应用层与数据库之间的连接和控制机制

★ 提供多层架构的应用开发和运行的平台，以及应用开发框架，支持模块化的应用开发

★ 屏蔽硬件、操作系统、网络和数据库的差异

★ 提供应用的负载均衡和高可用性、安全机制与管理功能，以及交易管理机制，保证交易的一致性

★ 提供一组通用的服务去执行不同的功能，避免重复的工作和使应用之间可以协作

希赛

构件与中间件技术 – 优点

采用中间件技术的优点

（1）面向需求。即设计师集中精力于业务逻辑本身。

（2）业务的分隔和包容性。应用开发人员可以按照不同的业务进行功能的划分，体现为不同的接口或交互模式。

（3）设计与实现隔离。构件对外发生作用或构件间的交互，都是通过接口进行的，构件使用者只需要知道构件的接口，而不必关心其内部实现，这是设计与实现分离的关键。

（4）隔离复杂的系统资源。架构很重要的一个功能就是将系统资源与应用构件隔离，这是保证构件可复用甚至“即插即用”的基础，与中间件的意图也是一致的。

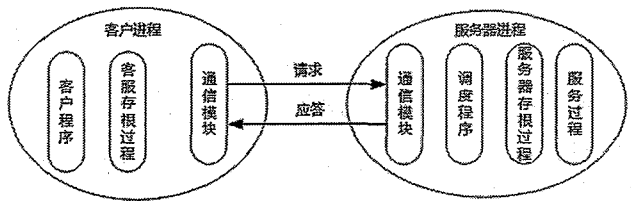
（5）符合标准的交互模型。中间件则实现了架构的模型，实现了标准的协议。

（7）软件复用。中间件提供了构件封装、交互规则、与环境的隔离等机制，这些都为软件复用提供了方便的解决方案。

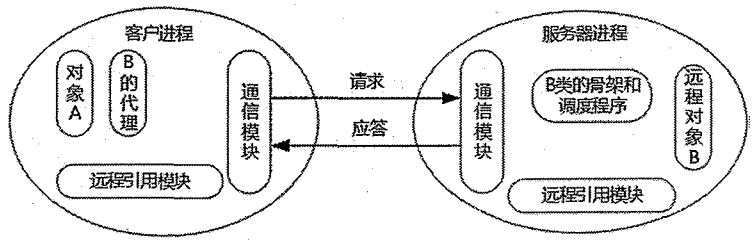
（8）提供对应用构件的管理。基于中间件的软件可以方便地进行管理，因为构件总可以通过标识机制进行划分。

希赛

中间件技术 – 主要的中间件

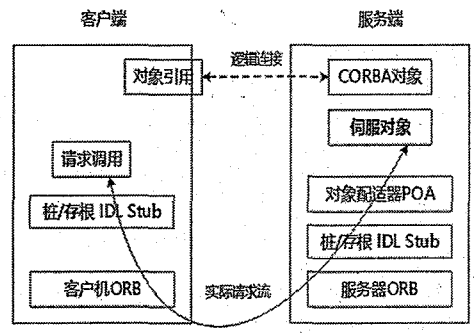


★ 远程过程调用
★ 对象请求代理
★ 远程方法调用
★ 面向消息的中间件
★ 事务处理监控器



希赛

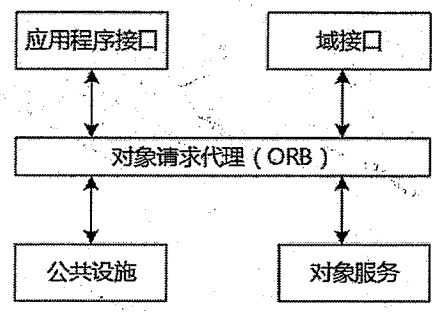
中间件技术 – Corba(公共对象请求代理体系结构)



伺服对象（Servant）：CORBA对象的真正实现，负责完成客户端请求。
对象适配器（Object Adapter）：用于屏蔽ORB内核的实现细节，为服务器对象的实现者提供抽象接口，以便他们使用ORB内部的某些功能。
对象请求代理（Object Request Broker）：解释调用并负责查找实现该请求的对象，将参数传给找到的对象，并调用方法返回结果。客户方不需要了解服务对象的位置、通信方式、实现、激活或存储机制。

希赛

中间件技术 – Corba(公共对象请求代理体系结构)



希赛

中间件技术 – Corba(公共对象请求代理体系结构)

CORBA体系的主要内容包括以下几部分：

（1）对象请求代理（Object Request Broker，ORB）。负责对象在分布环境中透明地收发请求和响应，它是构建分布对象应用、在异构或同构环境下实现应用间互操作的基础。

（2）对象服务（Object Services）。为使用和实现对象而提供的基本对象集合，这些服务应独立于应用领域。

（3）公共设施（Common Facilitites）。向终端用户提供一组共享服务接口，例如系统管理、组合文档和电子邮件等。

（4）应用接口（Application Interfaces）。由销售商提供的可控制其接口的产品，相应于传统的应用层表示，处于参考模型的最高层。

（5）领域接口（Domain Interfaces）。为应用领域服务而提供的接口，如OMG组织为PDM系统制定的规范。

希赛

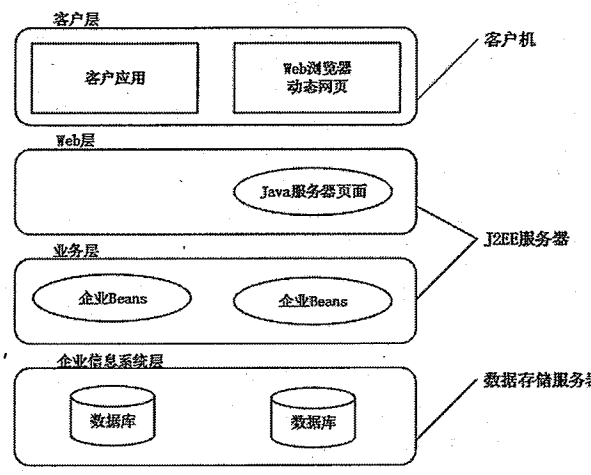
典型应用架构 – J2EE – 分布式多层应用程序

Bean运行于EJB容器之中，共分三类：

★ 会话Bean：描述了与客户端的一个短暂的会话。

★ 实体Bean：持久化数据，O/R映射。

★ 消息驱动Bean：会话Bean+JMS，客户把消息发送给JMS目的地，然后，JMS提供者和EJB容器协作，把消息发送给消息驱动Bean。支持异步消息。



希赛

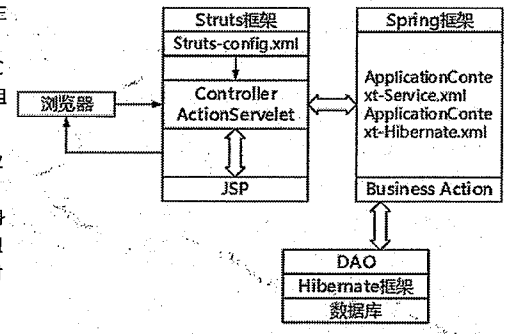
典型应用架构 – J2EE 核心组成

容器：
Applet Container、Application Container、Web Container、EJB Container
组件：
Applet、Application、JSP/Servlet、EJB
服务：
HTTP(Hypertext Transfer Protocol)超文本传输协议
RMI-IIOP(Remote Method Invocation ober the Internet Inter-ORB Protocol):远程方法调用，融合了Java RMI 和CORBA(Common Object Rrquest Broker Architecture 公共对象请求代理体系结构) 在使用Application 或Web 端访问EJB 端组件是使用
Java IDL(Java Interface Definition Language):Java 接口定义语言，主要用于访问外部的CORBA 服务
JTA(Java Transaction API):用于进行事务处理操作的 API
JDBC(Java Database Connectivity):为数据库操作提供的一组API
JMS(Java Massage Service):用于发送点对点消息的服务
JavaMail: 用于发送邮件
JAF(Java Activation Framework):用于封装传递的邮件数据
JNDI(Java Naming and Directory Interface)
JAXP(Java API for XML Parsing):专门用于XML解析操作的API
JCA(J2EE Connector Architecture):Java 连接器构架
JAAS (Java Authenticati on and Authorization Service)
JSF (Java Server Faces)
JSTL (JSP Standard Tag Library)
SAAJ (SOAP with Attachments API for JAVA)
JAXR (Java Apl for XML Registries)

希赛

典型应用架构 – J2EE – Java企业应用框架

- Struts是一个基于J2EE平台的MVC框架，主要采用Servlet和JSP技术来实现。在Struts中，M由实现业务逻辑的JavaBean构成，C由ActionServlet和Action来实现，V由一组JSP文件构成
- Spring通过RMI或Web Service远程访问业务逻辑，允许自由选择和组装各部分功能，还提供和其他软件集成的接口。Spring本身是个容器，管理构件的生命周期、构件的组态、依赖注入等，并可以控制构件在创建时是以原型或单例模式来创建



- Hibernate是一个对象关系映射框架，提供了Java对象到数据库表之间的直接映射，它对JDBC进行了非常轻量级的对象封装，使得Java程序员可以使用对象编程思维来操作数据库。在Hibernate 中，ORM机制的核心是一个XML文件，该文件描述了数据库模式是怎么与一组Java类绑定在一起的

希赛

典型应用架构 – J2EE – 重量级与轻量级之争

- 占用资源过多，在开发的过程中效率很低
- 大部分时间花在配置、运行的过程上，修改复杂
- 单元测试也比较麻烦

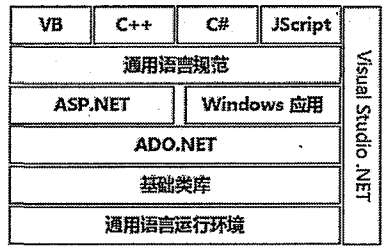
重量级

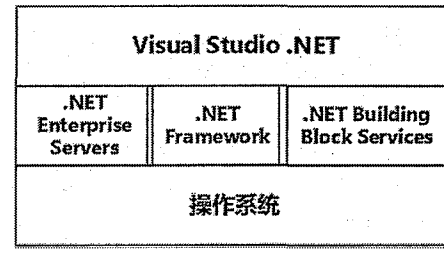
轻量级

- 提高了开发的速度
- 立即可以看到结果
- 做单元测试也非常简单
- 大量现成可供参考的开源代码

希赛

典型应用架构 – .NET





希赛

典型应用架构 – .NET与J2EE之争

- ★ JVM与CLR
- ★ 对多层分布式应用的支持
- ★ 安全性
- ★ 应用程序的部署
- ★ 可移植性
- ★ 外部支持

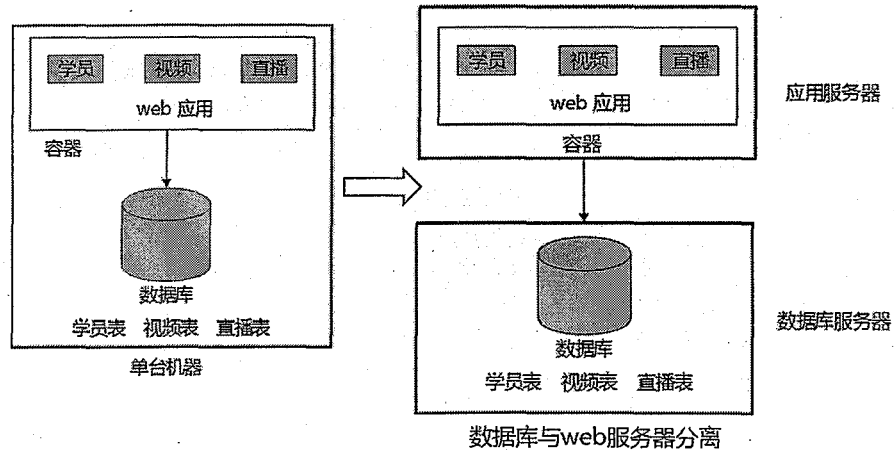
希赛

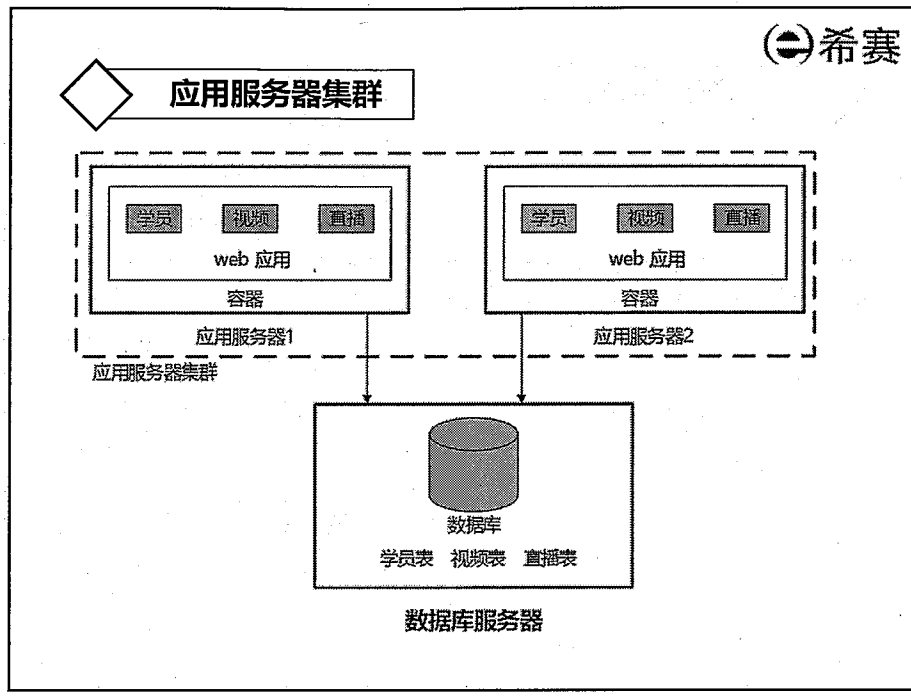
Web架构设计

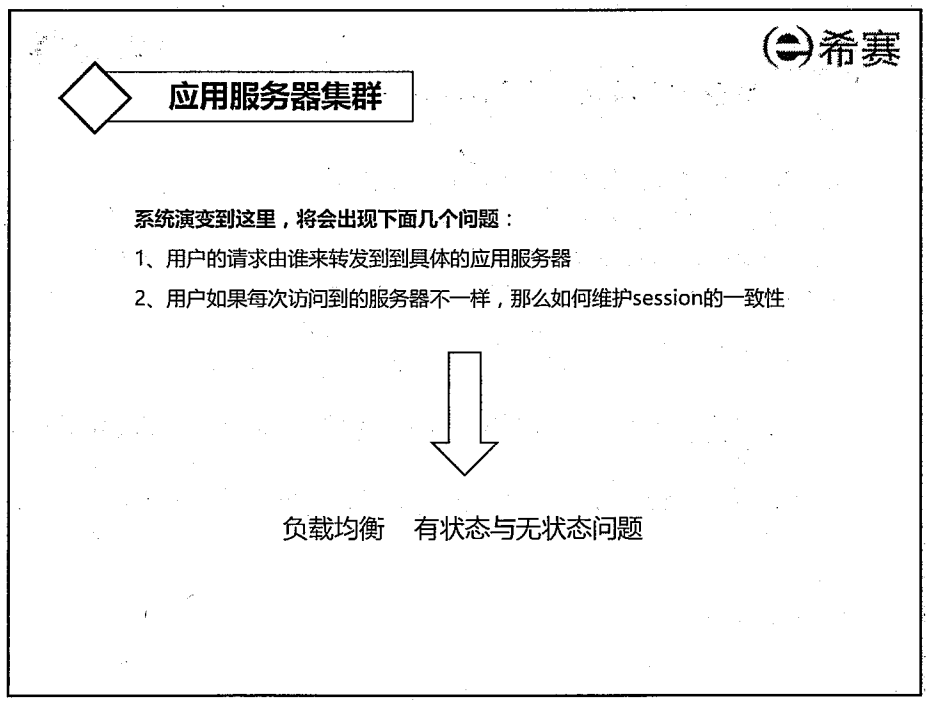
从架构来看：MVC，MVP，MVVM，REST，Webservice，微服务。
从缓存来看：MemCache，Redis，Squid。
从并发分流来看：集群（负载均衡）、CDN。
从数据库来看：主从库（主从复制），内存数据库，反规范化技术，NoSQL，分区（分表）技术，视图与物化视图。
从持久化来看：Hibernate，Mybatis。
从分布存储来看：Hadoop，FastDFS，区块链。
从数据编码看： XML，JSON。
从Web应用服务器来看： Apache，WebSphere，WebLogic，Tomcat，JBOSS，IIS。
其它：静态化，有状态与无状态，响应式Web设计。

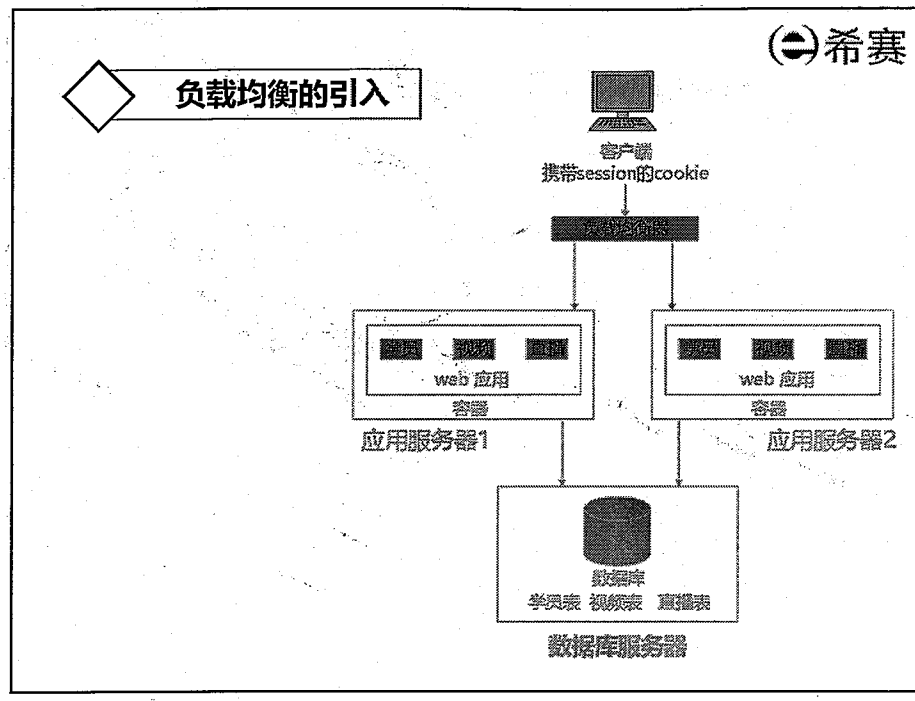
希赛

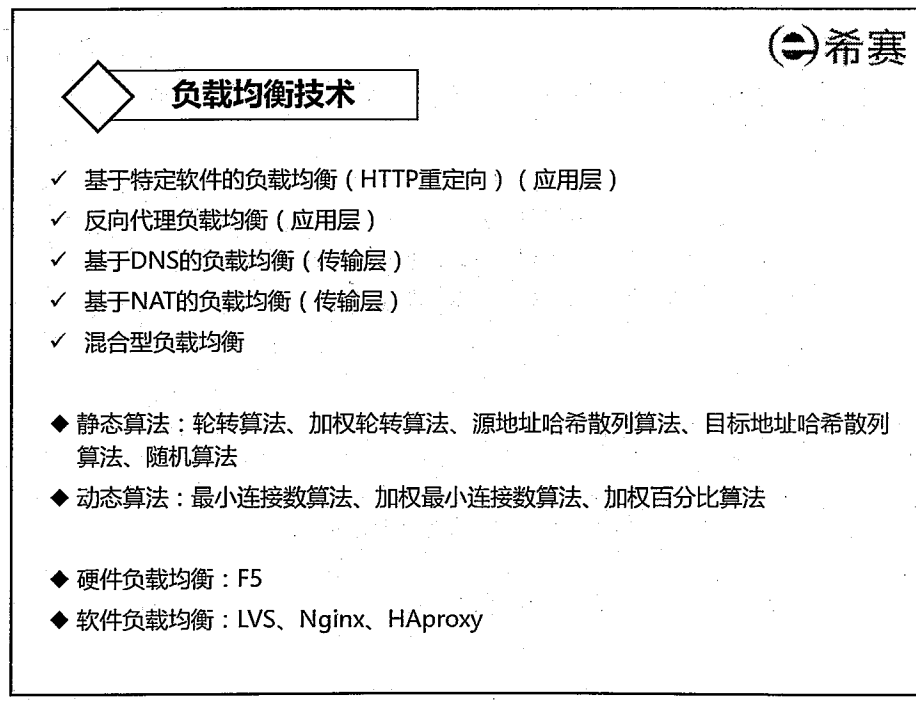
单台机器 到 数据库与Web服务器分离




希赛
应用服务器集群
学员
视频
直播
web 应用
容器
应用服务器1
学员
视频
直播
web 应用
容器
应用服务器2
应用服务器集群
数据库
学员表 视频表 直播表
数据库服务器


希赛
应用服务器集群
系统演变到这里，将会出现下面几个问题：
1、用户的请求由谁来转发到到具体的应用服务器
2、用户如果每次访问到的服务器不一样，那么如何维护session的一致性
负载均衡 有状态与无状态问题


希赛
负载均衡的引入
客户端
携带session的cookie
web 应用
容器
应用服务器1
web 应用
容器
应用服务器2
数据库
学员表 视频表 直播表
数据库服务器


希赛
负载均衡技术
✓ 基于特定软件的负载均衡（HTTP重定向）（应用层）
✓ 反向代理负载均衡（应用层）
✓ 基于DNS的负载均衡（传输层）
✓ 基于NAT的负载均衡（传输层）
✓ 混合型负载均衡
◆ 静态算法：轮转算法、加权轮转算法、源地址哈希散列算法、目标地址哈希散列算法、随机算法
◆ 动态算法：最小连接数算法、加权最小连接数算法、加权百分比算法
◆ 硬件负载均衡：F5
◆ 软件负载均衡：LVS、Nginx、HAproxy

希赛

负载均衡技术

应用层负载均衡

1、http重定向。HTTP重定向就是应用层的请求转发。用户的请求其实已经到了HTTP重定向负载均衡服务器，服务器根据算法要求用户重定向，用户收到重定向请求后，再次请求真正的集群。

特点：实现简单，但性能较差。

2、**反向代理服务器。**在用户的请求到达反向代理服务器时（已经到达网站机房），由反向代理服务器根据算法转发到具体的服务器。常用的apache，nginx都可以充当反向代理服务器。

特点：部署简单，但代理服务器可能成为性能的瓶颈。

希赛

负载均衡技术

传输层负载均衡

1、DNS域名解析负载均衡。DNS域名解析负载均衡就是在用户请求DNS服务器，获取域名对应的IP地址时，DNS服务器直接给出负载均衡后的服务器IP。

特点：效率比HTTP重定向高，减少维护负载均衡服务器成本。但一个应用服务器故障，不能及时通知DNS，而且DNS负载均衡的控制权在域名服务商那里，网站无法做更多的改善和更强大的管理。

2、基于NAT的负载均衡。基于NAT的负载均衡将一个外部IP地址映射为多个IP地址,对每次连接请求动态地转换为一个内部节点的地址。

特点：技术较为成熟，一般在网关位置，可以通过硬件实现。像四层交换机一般就采用了这种技术。

希赛

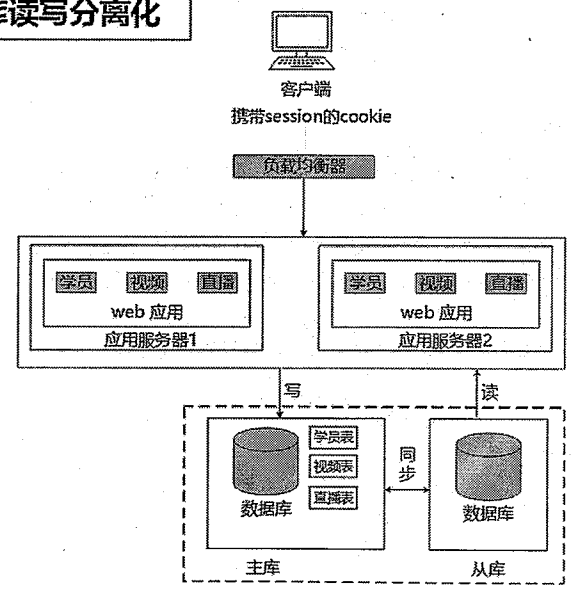
有状态与无状态

无状态服务（stateless service）对单次请求的处理，不依赖其他请求，也就是说，处理一次请求所需的全部信息，要么都包含在这个请求里，要么可以从外部获取到（比如说数据库），服务器本身不存储任何信息。

有状态服务（stateful service）则相反，它会在自身保存一些数据，先后的请求是有关联的。

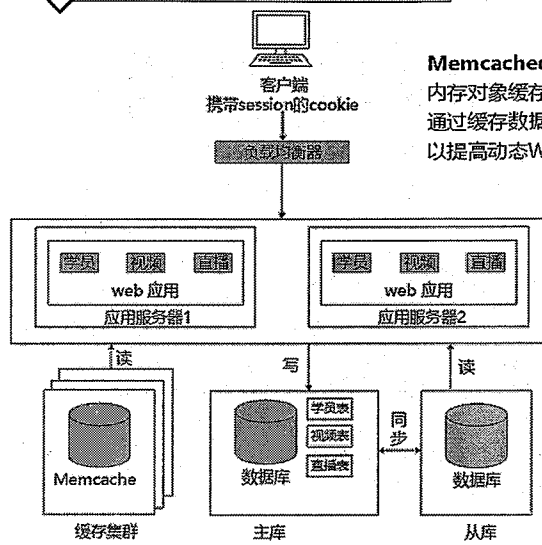
希赛

数据库读写分离化



希赛

用缓存缓解读库的压力



Memcached是一个自由开源的，高性能，分布式内存对象缓存系统。简洁的key-value存储系统。通过缓存数据库查询结果，减少数据库访问次数，以提高动态Web应用的速度、提高可扩展性。

希赛

CDN（内容分发网络）

CDN的全称是Content Delivery Network，即内容分发网络。其基本思路是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳定。

希赛

XML与JSON

扩展标记语言（Extensible Markup Language, XML），用于标记电子文件使其具有结构性的标记语言，可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。

XML的优点

A. 格式统一，符合标准；

B. 容易与其他系统进行远程交互，数据共享比较方便。

XML的缺点

A. XML文件庞大，文件格式复杂，传输占带宽；

B. 服务器端和客户端都需要花费大量代码来解析XML，导致服务器端和客户端代码变得异常复杂且不易维护；

C. 客户端不同浏览器之间解析XML的方式不一致，需要重复编写很多代码；

D. 服务器端和客户端解析XML花费较多的资源和时间。

希赛

XML与JSON

JSON(JavaScript Object Notation)一种轻量级的数据交换格式，具有良好的可读和便于快速编写的特性。可在不同平台之间进行数据交换。

JSON的优点

1、数据格式比较简单，易于读写，格式都是压缩的，占用带宽小；

2、易于解析，客户端JavaScript可以简单的通过eval()进行JSON数据的读取；

3、支持多种语言，包括ActionScript, C, C#, ColdFusion, Java, JavaScript, Perl, PHP, Python, Ruby等服务器端语言，便于服务器端的解析；

4、因为JSON格式能直接为服务器端代码使用，大大简化了服务器端和客户端的代码开发量，且完成任务不变，并且易于维护。

JSON的缺点

没有XML格式这么推广的深入人心和喜用广泛，没有XML那么通用性。

希赛

Web应用服务器

WEB应用服务器可以理解为两层意思：

（1）WEB服务器：其职能较为单一，就是把浏览器发过来的Request请求，返回Html页面。

（2）应用服务器：进行业务逻辑的处理。

Apache：Web服务器，市场占有率达60%左右。它可以运行在几乎所有的Unix、Windows、Linux系统平台上。

IIS：早期Web服务器，目前小规模站点仍有应用。

Tomcat：开源、运行servlet和JSP Web应用软件的基于Java的Web应用软件容器。

JBOSS：JBOSS是基于J2EE的开放源代码的应用服务器。一般与Tomcat或Jetty绑定使用。

WebSphere：一种功能完善、开放的Web应用程序服务器，它是基于Java 的应用环境，用于建立、部署和管理Internet 和Intranet Web 应用程序。

WebLogic：BEA WebLogic Server 是一种多功能、基于标准的web应用服务器，为企业构建自己的应用提供了坚实的基础。

Jetty：Jetty 是一个开源的servlet容器，它为基于Java的web容器。

希赛

缓存技术

MemCache： Memcache是一个高性能的分布式的内存对象缓存系统，用于动态Web应用以减轻数据库负载。Memcache通过在内存里维护一个统一的巨大的hash表，它能够用来存储各种格式的数据，包括图像、视频、文件以及数据库检索的结果等。

Redis：Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。

Squid： Squid是一个高性能的代理缓存服务器，Squid支持FTP、gopher、HTTPS和HTTP协议。和一般的代理缓存软件不同，Squid用一个单独的、非模块化的、I/O驱动的进程来处理所有的客户端请求。

希赛

缓存技术

Redis 与 Memcache 的差异

1、Redis和Memcache都是将数据存放在内存中，都是内存数据库。他们都支持key-value数据类型。同时Memcache还可用于缓存其他东西，例如图片、视频等等，Redis还支持list、set、hash等数据结构的存储。

2、Redis支持数据的持久化，可以将内存中的数据保持在磁盘中，重启的时候可以再次加载进行使用。Memcache挂掉之后，数据就没了。

3、灾难恢复-Memcache挂掉后，数据不可恢复; Redis数据丢失后可以恢复。

4、在Redis中，并不是所有的数据都一直存储在内存中的。这是和Memcache相比一个最大的区别。当物理内存用完时，Redis可以将一些很久没用到的value交换到磁盘。

5、Redis在很多方面支持数据库的特性，可以这样说他就是一个数据库系统，而Memcache只是简单地K/V缓存。

所以在选择方面如果有持久方面的需求或对数据类型和处理有要求的应该选择Redis。

如果简单的key/value存储应该选择Memcache。

希赛

REST（表述性状态传递）

REST（Representational State Transfer，表述性状态转移）是一种只使用HTTP和XML进行基于Web通信的技术，可以降低开发的复杂性，提高系统的可伸缩性。

REST的5个原则

（1）网络上的所有事物都被抽象为资源。

（2）每个资源对应一个唯一的资源标识。

（3）通过通用的连接件接口对资源进行操作。

（4）对资源的各种操作不会改变资源标识。

（5）所有的操作都是无状态的。

希赛

响应式Web设计

响应式WEB设计是一种网络页面设计布局，其理念是：集中创建页面的图片排版大小，可以智能地根据用户行为以及使用的设备环境进行相对应的布局。

方法与策略

（1）采用流式布局和弹性化设计：使用相对单位，设定百分比而非具体值的方式设置页面元素的大小。

（2）响应式图片：不仅要同比的缩放图片，还要在小设备上降低图片自身的分辨率。

希赛

系统安全分析与设计

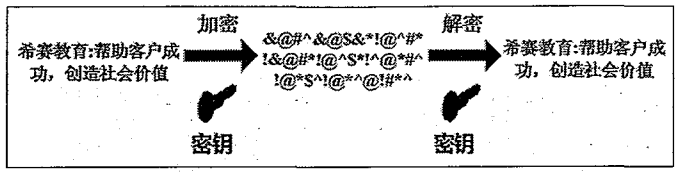
希赛

课程内容提要

- **安全基础技术**
  - 对称与非对称加密（★★★）
  - 数字签名（★★★）
  - 信息摘要（★★★）
- **网络安全**
  - 安全协议（★★★）
  - 网络攻击（★★）
  - 等级保护标准（★★）

希赛

对称加密技术



缺陷
- 1、加密强度不高，但效率高。
- 2、密钥分发困难。

常见对称密钥加密算法：

✓ DES：替换+移位、56位密钥、64位数据块、速度快、密钥易产生

3DES(三重DES)：两个56位的密钥K1、K2

加密：K1加密->K2解密->K1加密

解密：K1解密->K2加密->K1解密

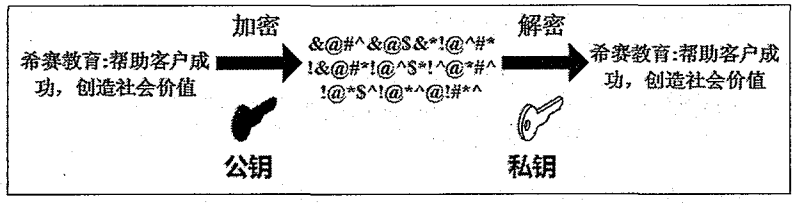
✓ RC-5：RSA数据安全公司的很多产品都使用了RC-5。

✓ IDEA算法：128位密钥、64位数据块、比DES的加密性好、对计算机功能要求相对低，PGP。

✓ AES算法：高级加密标准，又称Rijndael加密法，是美国政府采用的一种区块加密标准。

希赛

非对称加密技术



常见非对称密钥加密算法：RSA：2048位（或1024位）密钥、计算量极大、难破解

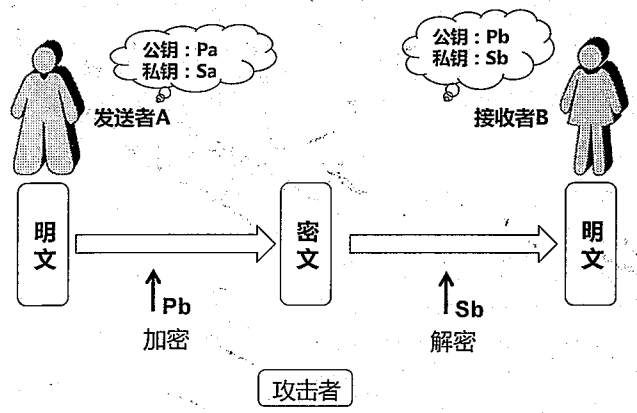
Elgamal：安全性依赖于计算有限域上离散对数这一难题。

ECC：椭圆曲线算法

缺陷：加密速度慢

希赛

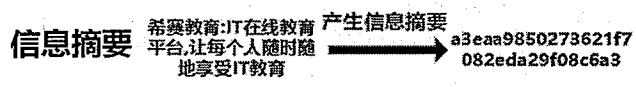
非对称加密技术



希赛

信息摘要

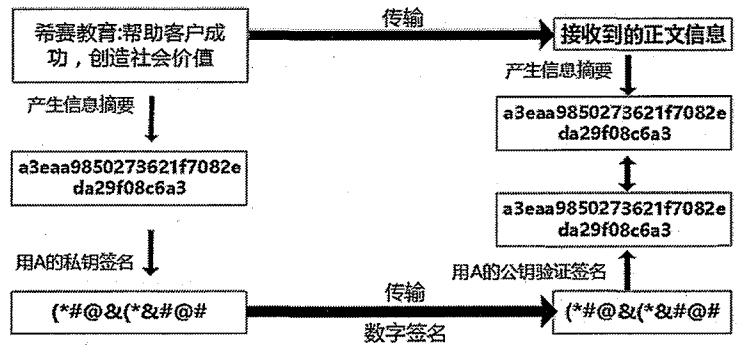
单向散列函数、固定长度的散列值。



常用的消息摘要算法有MD5，SHA等，市场上广泛使用的MD5，SHA算法的散列值分别为128和160位，由于SHA通常采用的密钥长度较长，因此安全性高于MD5。

希赛

数字签名



希赛

练习题

请依据已学习的加密解密技术，以及信息摘要，数字签名技术解决以下问题：

请设计一个安全邮件传输系统，要求：

该邮件以加密方式传输，邮件最大附件内容可达2GB，发送者不可抵赖，若邮件被第三方截获，第三方无法篡改。

希赛

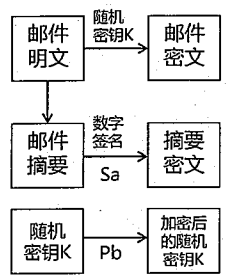
练习题

加密解密技术　　对称加密　　数字签名

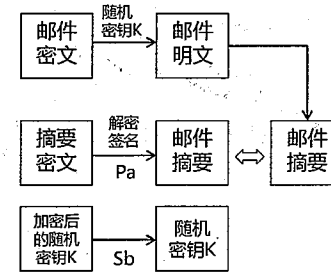
该邮件以加密方式传输，邮件最大附件内容可达2GB，发送者不可抵赖，若邮件被第三方截获，第三方无法篡改。

信息摘要技术

发送方A(公钥：Pa，私钥：Sa)：



接收方B(公钥：Pb，私钥：Sb)：

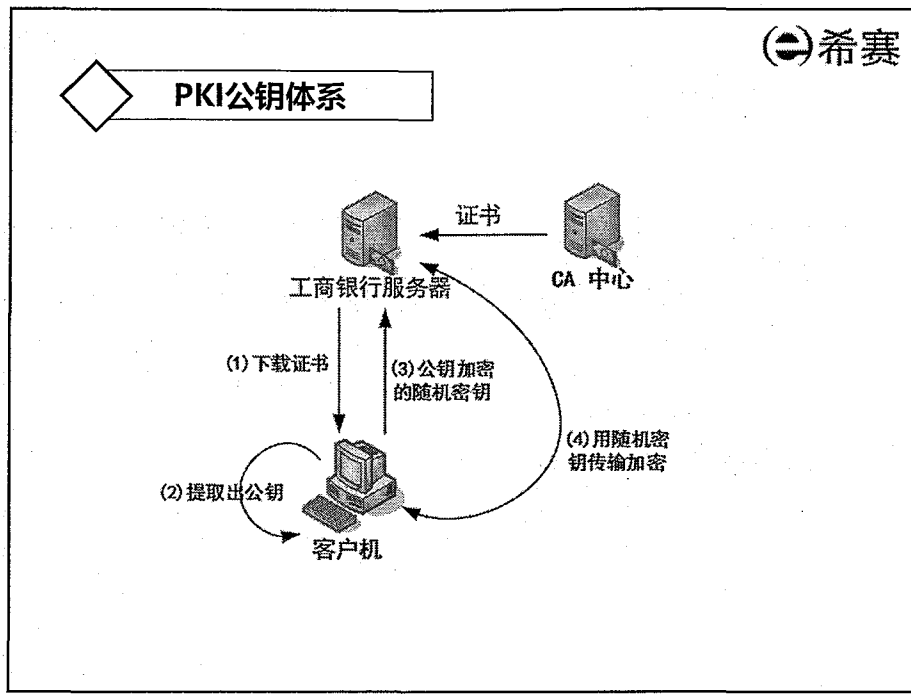


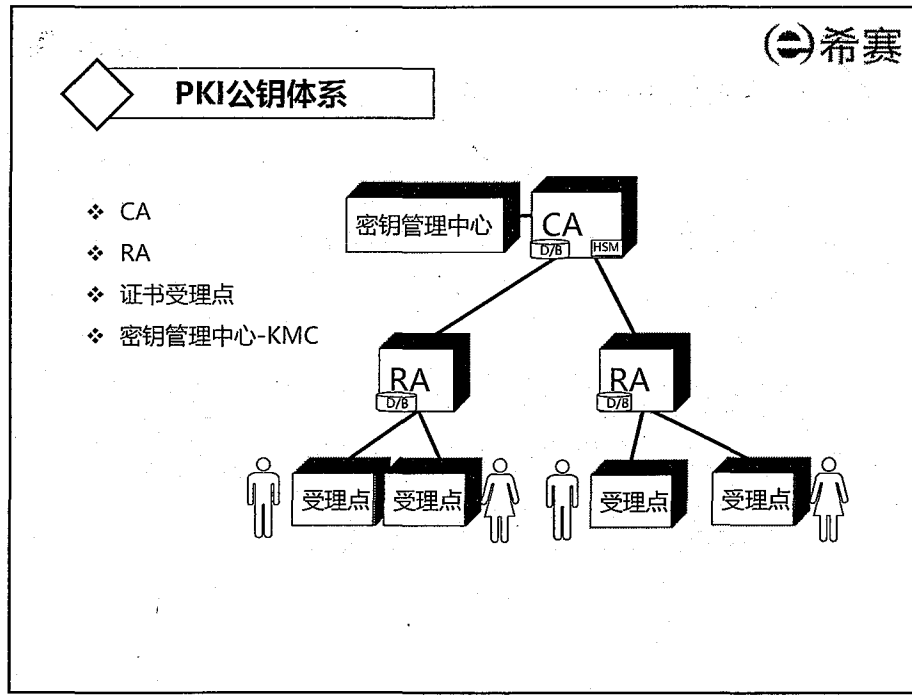
希赛

数字证书内容

◆ **数字证书内容**

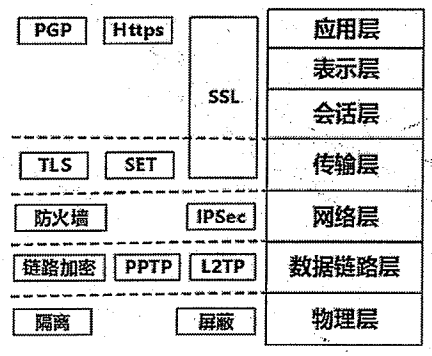
- 证书的版本信息；
- 证书的序列号，每个证书都有一个唯一的证书序列号；
- 证书所使用的签名算法；
- 证书的发行机构名称，命名规则一般采用X.500格式；
- 证书的有效期，现在通用的证书一般采用UTC时间格式，它的计时范围为1950-2049;
- 证书所有人的名称，命名规则一般采用X.500格式；
- 证书所有人的公开密钥；
- 证书发行者对证书的签名。


希赛
PKI公钥体系
证书
工商银行服务器
CA 中心
(1) 下载证书
(3) 公钥加密的随机密钥
(4) 用随机密钥传输加密
(2) 提取出公钥
客户机


希赛
PKI公钥体系
CA
RA
证书受理点
密钥管理中心-KMC
密钥管理中心
CA
D/B
HSM
RA
D/B
RA
D/B
受理点
受理点
受理点
受理点

希赛

网络安全－各个网络层次的安全保障



PGP：针对邮件的混合加密系统。
SSL：工作在传输层至应用层。
TLS：传输层安全协议。
SET：电子商务。
IPSEC：对IP包加密。

希赛

网络安全－网络威胁与攻击

被动攻击：收集信息为主，破坏保密性。
主动攻击：主动攻击的类别主要有：中断（破坏可用性），篡改（破坏完整性），伪造（破坏真实性）。

攻击类型	攻击名称	描述
被动攻击	窃听（网络监听）	用各种可能的合法或非法的手段窃取系统中的信息资源和敏感信息。
	业务流分析	通过对系统进行长期监听，利用统计分析方法对诸如通信频度、通信的信息流向、通信总量的变化等参数进行研究，从而发现有价值的信息和规律。
	非法登录	有些资料将这种方式归为被动攻击方式。

希赛

网络安全－网络威胁与攻击

攻击类型	攻击名称	描述
主动攻击	假冒身份	通过欺骗通信系统（或用户）达到非法用户冒充成为合法用户，或者特权小的用户冒充成为特权大的用户的目的。黑客大多是采用假冒进行攻击。
	抵赖	这是一种来自用户的攻击，比如：否认自己曾经发布过的某条消息、伪造一份对方来信等。
	旁路控制	攻击者利用系统的安全缺陷或安全性上的脆弱之处获得非授权的权利或特权。
	重放攻击	所截获的某次合法的通信数据拷贝，出于非法的目的而被重新发送。加时间戳能识别并应对重放攻击。
	拒绝服务（DOS）	对信息或其它资源的合法访问被无条件地阻止。

希赛

安全保护等级

计算机信息系统安全保护等级划分准则(GB17859-1999)

- ✓ **用户自主保护级**：适用于普通内联网用户

  系统被破坏后，对公民、法人和其它组织权益有损害，但不损害国家安全社会秩序和公共利益。

- ✓ **系统审计保护级**：适用于通过内联网或国际网进行商务活动，需要保密的非重要单位。

  系统被破坏后，对公民、法人和其它组织权益有严重损害，或损害社会秩序和公共利益，但不损害国家安全。

- ✓ **安全标记保护级**：适用于地方各级国家机关、金融机构、邮电通信、能源与水源供给部门、交通运输、大型工商与信息技术企业、重点工程建设等单位

  系统被破坏后，对社会秩序和公共利益造成严重损害，或对国家安全造成损害。

- ✓ **结构化保护级**：适用于中央级国家机关、广播电视部门、重要物资储备单位、社会应急服务部门、尖端科技企业集团、国家重点科研机构和国防建设等部门

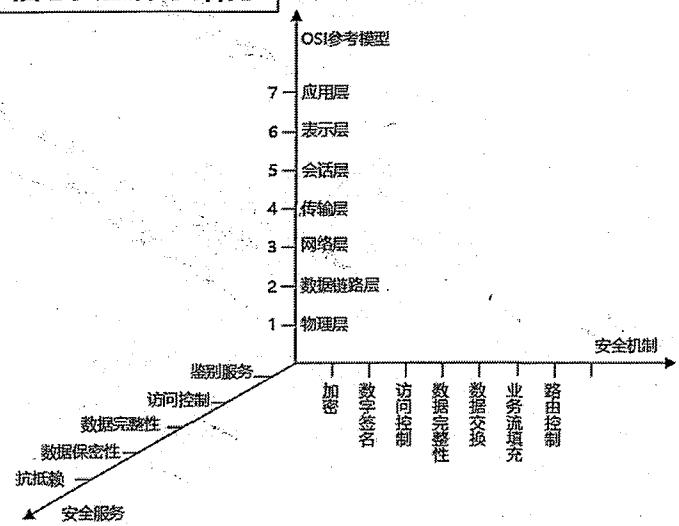
  系统被破坏后，对社会秩序和公共利益造成特别严重损害，或对国家安全造成严重损害。

- ✓ **访问验证保护级**：适用于国防关键部门和依法需要对计算机信息系统实施特殊隔离的单位

  系统被破坏后，对国家安全造成特别严重损害。

希赛

信息安全体系结构



希赛

信息安全体系结构

鉴别服务 →
- ✓ 用户名+口令
- ✓ 数字证书
- ✓ 生物特征识别

访问控制 →
- ✓ 自主访问控制(DAC)
- ✓ 访问控制列表(ACL)
- ✓ 强制访问控制(MAC)
- ✓ 基于角色的访问控制(RBAC)
- ✓ 基于任务的访问控制(TBAC)

数据完整性 →
- ✓ 阻止对媒体访问的机制：隔离，访问控制，路由控制
- ✓ 探测非授权修改的机制：数字签名，数据重复，数字指纹，消息序列号

数据保密性 →
- ✓ 通过禁止访问提供机密性
- ✓ 通过加密提供机密性

抗抵赖 →
- ✓ 数字签名

希赛

系统可靠性分析与设计

希赛

课程内容提要

- 可靠性相关基本概念（★★）
- 系统可靠性分析（★★★★）
- 软件可靠性设计（★★★★）

希赛

可靠性相关基本概念

- 系统可靠性是系统在规定的时间内及规定的环境条件下，完成规定功能的能力，也就是系统无故障运行的概率。
- 系统可用性是指在某个给定时间点上系统能够按照需求执行的概率。

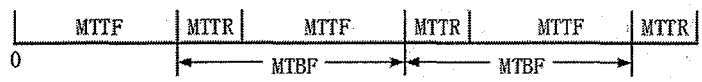
软件可靠性 ≠ 硬件可靠性

- 复杂性：软件复杂性比硬件高，大部分失效来自于软件失效。
- 物理退化：硬件失效主要是物理退化所致，软件不存在物理退化。
- 唯一性：软件是唯一的，每个COPY版本都一样，而两个硬件不可能完全一样。
- 版本更新周期：硬件较慢，软件较快。

希赛

系统可靠性分析 – 可靠性指标

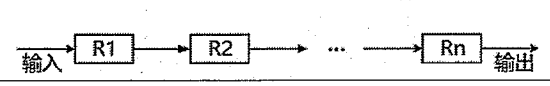
- 平均无故障时间 → (MTTF) MTTF=1/λ，λ为失效率
- 平均故障修复时间 → (MTTR) MTTR=1/μ，μ为修复率
- 平均故障间隔时间 → (MTBF) MTBF = MTTR + MTTF
- 系统可用性 → MTTF/(MTTR+MTTF)×100%



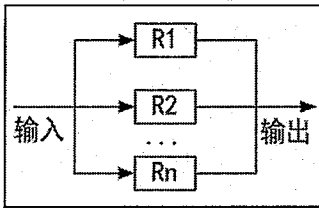
- 在实际应用中，一般MTTR很小，所以通常认为MTBF≈MTTF。

希赛

系统可靠性分析 – 串联系统与并联系统



可靠性 $R = R_1 \times R_2 \times \ldots \times R_n$ 失效率近似公式 $\lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_n$

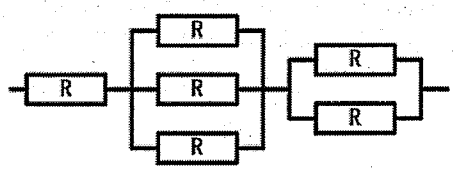


可靠性

$$R = 1-(1-R_1)\times(1-R_2)\times\ldots\times(1-R_n)$$

希赛

系统可靠性分析 – 混合系统



$$R\times(1-(1-R)^3)\times(1-(1-R)^2)$$

希赛

可靠性设计

✓ 结构冗余(硬件冗余、软件冗余)
✓ 信息冗余(校验码)
✓ 时间冗余(重复多次进行相同的计算)

冗余

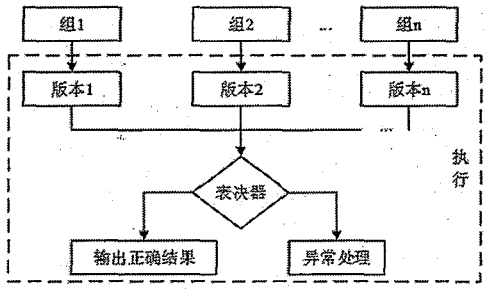
✓ 避错技术
✓ 容错技术
✓ 检错技术
✓ 降低复杂度设计

✓ N版本程序设计（静态冗余）
✓ 恢复块设计（动态冗余）
✓ 防卫式程序设计

✓ 出错后报警，人工处理，成本较低。

希赛

可靠性设计 – N版本程序设计

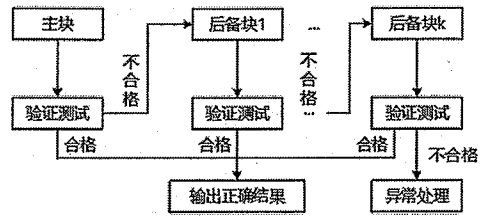


◆ 与通常软件开发过程不同的是，N版本程序设计增加了三个新的阶段：相异成份规范评审、相异性确认、背对背测试

◆ N版本程序的同步、N版本程序之间的通信、表决算法（全等表决、非精确表决、Cosmetie表决）、一致比较问题、数据相异性

希赛

可靠性设计 – 恢复块方法



◆ 设计时应保证实现主块和后备块之间的独立性，避免相关错误的产生，使主块和备份块之间的共性错误降到最低程度。

◆ 必须保证验证测试程序的正确性。

	恢复块方法	N版本程序设计
硬件运行环境	单机	多机
错误检测方法	验证测试程序	表决
恢复策略	后向恢复	前向恢复
实时性	差	好

➢ 前向恢复：使当前的计算继续下去，把系统恢复成连贯的正确状态，弥补当前状态的不连贯情况

➢ 后向恢复：系统恢复到前一个正确状态，继续执行

希赛

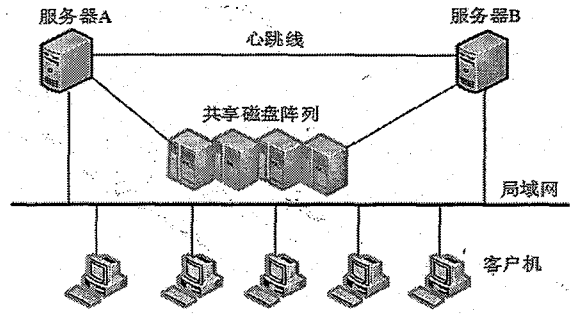
可靠性设计 – 防卫式程序设计

◆ 对于程序中存在的错误和不一致性，通过在程序中包含错误检查代码和错误恢复代码，使得一旦错误发生，程序能撤消错误状态，恢复到一个已知的正确状态中去

◆ 实现策略：错误检测、破坏估计、错误恢复

希赛

可靠性设计 -双机容错



- 双机热备模式（主系统、备用系统）
- 双机互备模式（同时提供不同的服务，心不跳则接管）
- 双机双工模式（同时提供相同的服务，集群的一种）

希赛

可靠性设计 - 集群技术

高性能主机系统 **PK** 集群系统

- 可伸缩性
- 高可用性
- 可管理性
- 高性价比
- 高透明性

案例分析

阅读以下信息系统可靠性问题的说明，在答题纸上回答问题1至问题3。

【说明】

某软件公司开发一项基于数据流的软件，其系统的主要功能是对输入数据进行多次分析、处理和加工，生成需要的输出数据。需求方对该系统的软件可靠性要求很高，要求系统能够长时间无故障运行。该公司将该系统设计交给王工负责。王工给出该系统的模块示意图如图1所示。王工解释：只要各个模块的可靠度足够高，失效率足够低，则整个软件系统的可靠性是有保证的。

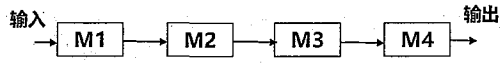


图1 王工建议的软件系统模块示意图

李工对王工的方案提出了异议。李工认为王工的说法有两个问题：第一，即使每个模块的可靠度足够高，但是整个软件系统模块之间全部采用串联，则整个软件系统的可靠度明显下降。假设各个模块的可靠度均为0.99，则整个软件系统的可靠度为$0.99^4 \approx 0.96$；第二，软件系统模块全部采用串联结构时，一旦某个模块失效，则意味着整个软件系统失效。

案例分析

李工认为，应该在软件系统中采用冗余技术中的动态冗余或者软件容错的N版本程序设计技术，对容易失效或者非常重要的模块进行冗余设计，将模块之间的串联结构部分变为并联结构，来提高整个软件系统的可靠性。同时，李工给出了采用动态冗余技术后的软件系统模块示意图，如图2所示。

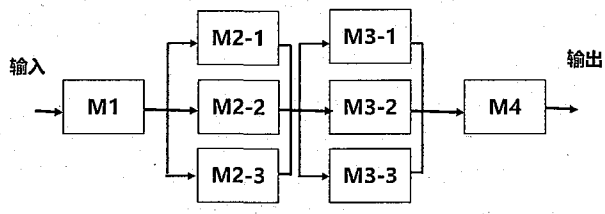


图2 李工建议的系统模块示意图

刘工建议，李工方案中M1和M4模块没有采用容错设计，但M1和M4发生故障有可能导致严重后果。因此，可以在M1和M4模块设计上采用检错技术，在软件出现故障后能及时发现并报警，提醒维护人员进行处理。

注：假设各个模块的可靠度均为0.99。

希赛

案例分析

【问题1】（4分）

在系统可靠性中，可靠度和失效率是两个非常关键的指标，请分别解释其含义。

【问题2】（13分）

请解释李工提出的动态冗余和N版本程序设计技术，给出图1中模块M2采用图2动态冗余技术后的可靠度。

请给出采用李工设计方案后整个系统可靠度的计算方法，并计算结果。

【问题3】（8分）

请给出检错技术的优缺点，并说明检测技术常见的实现方式和处理方式。

希赛

案例分析

【问题1】（4分）

可靠度就是系统在规定的条件下、规定的时间内不发生失效的概率。

失效率又称风险函数，也可以称为条件失效强度，是指运行至此刻系统未出现失效的情况下，单位时间系统出现失效的概率。

【问题2】（13分）

动态冗余又称为主动冗余，它是通过故障检测、故障定位及故障恢复等手段达到容错的目的。其主要方式是多重模块待机储备，当系统检测到某工作模块出现错误时，就用一个备用的模块来替代它并重新运行。各备用模块在其待机时，可与主模块一样工作，也可以不工作。前者叫冷备份系统（双工系统、双份系统），后者叫热备份系统（双重系统）。

N版本程序设计是一种静态的故障屏蔽技术，其设计思想是用N个具有相同功能的程序同时执行一项计算，结果通过多数表决来选择。其中N个版本的程序必须由不同的人独立设计，使用不同的方法、设计语言、开发环境和工具来实现，目的是减少N个版本的程序在表决点上相关错误的概率。

希赛

案例分析

M2采用动态冗余后的可靠度为：

$R = 1 - (1-0.99)^3 \approx 0.999999$

李工的方案同时采用了串联和并联方式，其计算方法为首先计算出中间M2和M3两个并联系统的可靠度，再按照串联系统的计算方法计算出整个系统的可靠度。

$R = 0.99 * 0.999999 * 0.999999 * 0.99 \approx 0.98$

【问题3】（8分）

检错技术实现的代价一般低于容错技术和冗余技术，但有一个明显的缺点，就是不能自动解决故障，出现故障后如果不进行人工干预，将最终导致软件系统不能正常运行。

检错技术常见的实现方式：最直接的一种实现方式是判断返回结果，如果返回结果超出正常范围，则进行异常处理；计算运行时间也是一种常用技术，如果某个模块或函数运行时间超过预期时间，可以判断出现故障；还有置状态标志位等多种方法，自检的实现方式需要根据实际情况来选用。

检错技术的处理方式，大多数都采用“查处故障-停止软件运行-报警”的处理方式。但根据故障的不同情况，也有采用不停止或部分停止软件系统运行的情况，这一般由故障是否需要实时处理来决定。

希赛

论文写作-2014年下半年考题

论软件的可靠性设计

现代军事和商用系统中，随着系统中软件成分的不断增加，系统对软件的依赖性越来越强。软件可靠性已成为软件设计过程中不可或缺的重要组成部分。实践证明，保障软件可靠性最有效、最经济、最重要的手段是在软件设计阶段采取措施进行可靠性控制，由此提出了可靠性设计的概念。可靠性设计就是在常规的软件设计中，应用各种方法和技术，使程序设计在兼顾用户的功能和性能需求的同时，全面满足软件的可靠性要求。

请以“软件的可靠性设计”为题，依次从以下三个方面进行论述。

1．概要叙述你参与管理和开发的软件项目以及你在其中所担任的主要工作。

2．简要说明目前比较主流的软件可靠性设计技术，结合项目实际情况，阐述所选择的可靠性设计技术及其原因。

3．结合你具体参与管理和开发的实际项目，举例说明所选取的软件可靠性技术的具体实施过程，并详细分析实施效果。

希赛

论文写作-2013年下半年考题

论软件可靠性设计技术的应用

随着软件的日益普及，系统中软件成分不断增加，使得系统对软件的依赖越来越强。

软件的可靠性对系统可靠性的影响越来越大。而实践证明，保障软件可靠性最有效、最经济、最重要的手段是在软件设计阶段采取措施进行可靠性控制，为此提出了软件可靠性设计的概念。

软件可靠性设计就是在常规的软件设计中，应用各种方法和技术，使软件设计在兼顾用户功能和性能需求的同时，全面满足软件的可靠性要求。软件可靠性设计应和软件的常规设计紧密结合，贯穿于软件设计过程的始终。

请围绕“软件可靠性设计技术的应用”论题，依次从以下三个方面进行论述。

1．概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。

2．结合项目实际，论述你在项目开发过程中，进行软件可靠性设计时遵循的基本原则；论述你在该项目中所采用的具体可靠性设计技术。

3．阐述你在具体的可靠性设计工作中，为了分析影响软件可靠性的主要因素，所采用的可靠性分析方法。

希赛

论文写作-2010年下半年考题

论软件可靠性评价

软件可靠性评价是指选用和建立合适的可靠性数学模型，运用统计技术和其他手段，对软件可靠性测试和系统运行期间的软件失效数据（也可能包含软件生命周期内其他可靠性数据）进行处理，并评估和预测软件可靠性的过程。

软件可靠性评价是软件可靠性活动的重要组成部分，既可在软件开发过程实施，也可针对最终软件系统实施。软件可靠性评价的难点在于软件可靠性模型的选择和软件可靠性数据的收集与处理。

请围绕“软件可靠性评价”论题，依次从以下三个方面进行论述。

1. 简要概述你参与实施的软件开发项目以及你承担的主要工作。

2. 说明你在项目实施过程中所选择的软件可靠性模型，并论述在软件可靠性模型选择时应该考虑的主要因素。

3. 收集软件可靠性数据时经常遇到的问题有哪些？简述你收集软件可靠性数据时所遇到的具体问题及解决的方法。

希赛

论文写作-2009年下半年考题

论软件可靠性设计与应用

目前在企业中，以软件为核心的产品得到了广泛的应用。随着系统中软件部分比例的不断增加，使得系统对软件的依赖性越来越强，对软件的可靠性要求也越来越高。软件可靠性与其它质量属性一样，是衡量软件架构的重要指标。

软件工程中已有很多比较成熟的设计技术，如结构化设计、模块化设计、自顶向下设计等，这些技术为保障软件的整体质量发挥了重要作用。在此基础上，为了进一步提高软件的可靠性，通常会采用一些特殊的设计技术，即软件可靠性设计技术。

在软件可靠性工程体系中，包含有可靠性模型与预测、可靠性设计和可靠性测试方法等。实践证明，保障软件可靠性最有效、最经济、最重要的手段是在软件设计阶段采取措施进行可靠性控制。

请围绕“软件可靠性设计与应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与实施的软件开发项目以及你所承担的主要工作。

2.简要叙述影响软件可靠性的因素有哪些。

3.阐述常用的软件可靠性设计技术以及你如何应用到实际项目中，效果如何。

希赛

项目管理

希赛

课程内容提要

- 范围管理（★★）
- 时间管理（★★★★）
- 成本管理（★）
- 软件质量管理（★★）
- 软件配置管理（★★）
- 风险管理（★）

希赛

范围管理

◆ 范围管理：确定项目的边界，即哪些工作是项目应该做的，哪些工作不应该包括在项目中。

范围计划编制 → 范围定义 → 创建WBS → 范围确认 → 范围控制

✓ 产品范围
✓ 工作范围

✓ WBS
✓ WBS字典

层次名称	层次编号	层次描述	层次目的
决策层	1	总项目	工作授权和解除
管理层	2	项目	预算编制
	3	任务	进度计划编制
技术层	4	子任务	内部控制
	5	工作包	
操作层	6	努力水平	

希赛

时间管理

◆ 时间管理：也叫进度管理，就是采用科学的方法，确定进度目标，编制进度计划和资源供应计划，进行进度控制，在与质量、成本目标协调的基础上，实现工期目标。

活动定义 → 活动排序 → 活动资源估算 → 活动历时估算 → 制订进度计划 → 进度控制

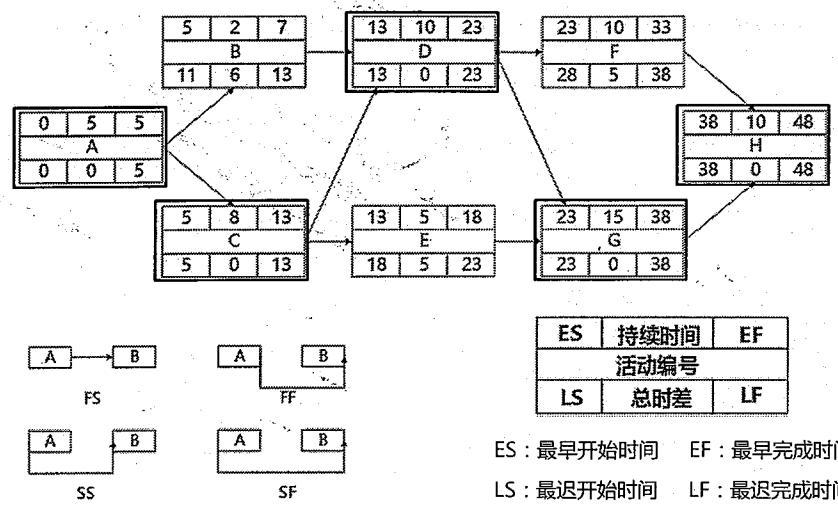
✓ 专家判断法
✓ 三点估算法
✓ 功能点估算法
✓ 自上而下的估算
✓ 自下而上的估算

✓ 是否为关键活动
✓ 偏差是否大于总时差
✓ 偏差是否大于自由时差

✓ 赶工：增加资源
✓ 快速跟进：活动并行执行

希赛

时间管理－前导图法(单代号网络图，PDM)



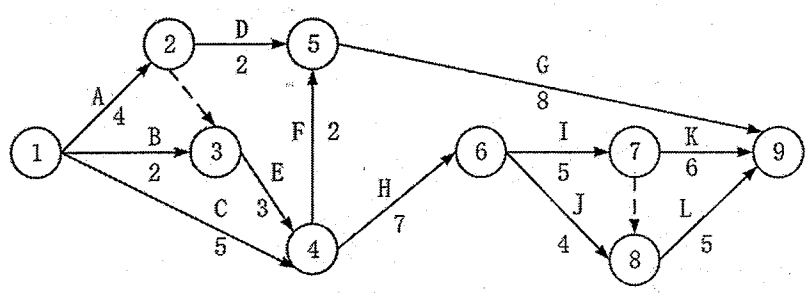
ES	持续时间	EF
活动编号		
LS	总时差	LF

ES：最早开始时间　EF：最早完成时间

LS：最迟开始时间　LF：最迟完成时间

希赛

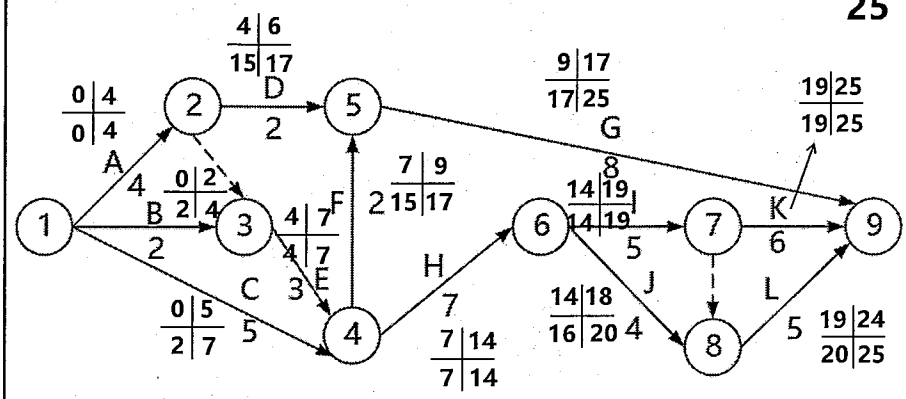
时间管理－关键路径法



希赛

时间管理 – 关键路径法

25



希赛

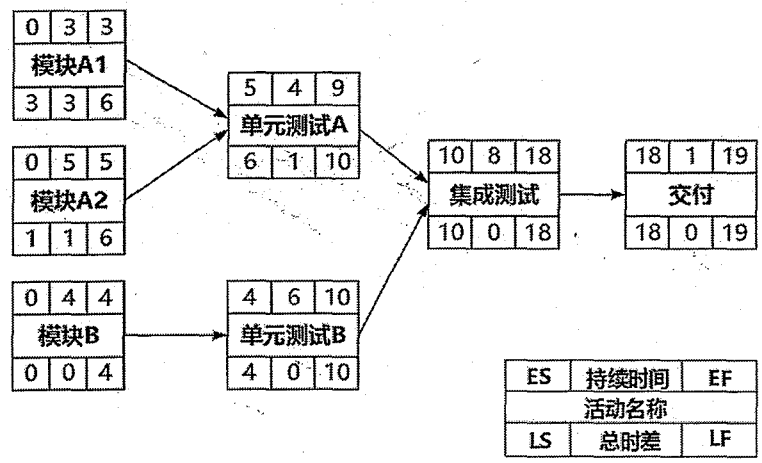
时间管理 – 关键路径法

- 关键路径法是在制订进度计划时使用的一种进度网络分析技术。关键路线法沿着项目进度网络路线进行正向与反向分析，从而计算出所有计划活动理论上的最早开始与完成日期、最迟开始与完成日期，不考虑任何资源限制

- 总时差（**松弛时间**）：在不延误总工期的前提下，该活动的机动时间。活动的总时差等于该活动最迟完成时间与最早完成时间之差，或该活动最迟开始时间与最早开始时间之差
- 自由时差：在不影响紧后活动的最早开始时间前提下，该活动的机动时间。
  - 对于有紧后活动的活动，其自由时差等于所有紧后活动最早开始时间减本活动最早完成时间所得之差的最小值
  - 对于没有紧后活动的活动，也就是以网络计划终点节点为完成节点的活动，其自由时差等于计划工期与本活动最早完成时间之差
- 对于网络计划中以终点节点为完成节点的活动，其自由时差与总时差相等。此外，由于活动的自由时差是其总时差的构成部分，所以，当活动的总时差为零时，其自由时差必然为零，可不必进行专门计算

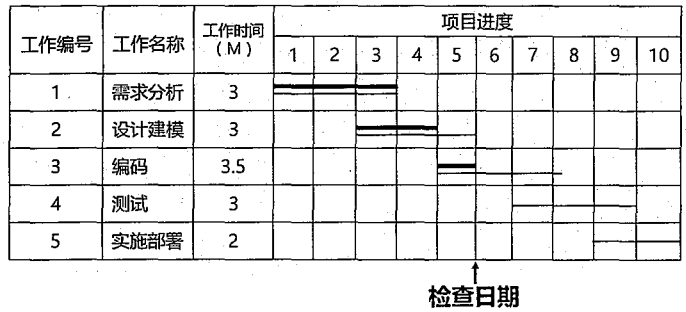
希赛

时间管理－自由时差



希赛

时间管理－甘特图（ Gantt ）

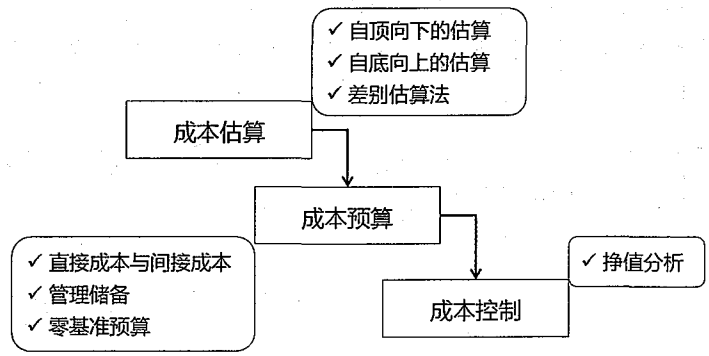


- 优点：甘特图直观、简单、容易制作，便于理解，能很清晰地标识出直到每一项任务的起始与结束时间，一般适用比较简单的小型项目，可用于WBS的任何层次、进度控制、资源优化、编制资源和费用计划。
- 缺点：不能系统地表达一个项目所包含的各项工作之间的复杂关系，难以进行定量的计算和分析，以及计划的优化等。

希赛

成本管理

◆ 成本管理：在整个项目的实施过程中，为确保项目在批准的预算条件下尽可能保质按期完成，而对所需的各个过程进行管理与控制。



希赛

成本管理 – 挣值管理

◆ 计划工作量的预算成本(PV)

PV = 计划工作量×预算定额

◆ 已完成工作量的实际成本(AC)

◆ 已完成工作量的预算成本(EV)

EV = 已完成工作量×预算定额

◆ 完工预算(BAC)

BAC=完工时的PV总和

✓ 进度偏差：SV = EV-PV

✓ 成本偏差：CV = EV-AC

✓ 进度绩效指数：SPI = EV/PV

✓ 成本绩效指数：CPI = EV/AC

✓ 剩余工作的成本(ETC)

ETC = BAC-EV

ETC = (BAC-EV)/CPI

✓ 完工估算(EAC)

EAC = AC + ETC

◆ 希赛教育在线测试项目涉及对10个函数代码的编写(假设每个函数代码的编写工作量相等)，项目由2个程序员进行结对编程，计划在10天内完成，总体预算是1000元，每个函数的平均成本是100元。项目进行到了第5天，实际消耗费用是400元，完成了3个函数代码的编写。

希赛

软件质量管理 – 质量保证与质量控制

◆ 质量保证一般是每隔一定时间（例如，每个阶段末）进行的，主要通过系统的质量审计和过程分析来保证项目的质量。独特工具包括：质量审计和过程分析。

◆ 质量控制是实时监控项目的具体结果，以判断它们是否符合相关质量标准，制订有效方案，以消除产生质量问题的原因。

◆ 一定时间内质量控制的结果也是质量保证的质量审计对象。质量保证的成果又可以指导下一阶段的质量工作，包括质量控制和质量改进

希赛

软件评审

★ 不应以测试代替评审
★ 评审人员应关注产品而不应评论开发人员
★ 评审人员应关注于实质性问题
★ 评审会议不应变为问题解决方案讨论会
★ 评审应被安排进入项目计划
★ 评审参与者应了解整个评审过程
★ 评审人员事先应对评审材料充分了解
★ 应重视评审的组织工作

- 技术评审
- 管理评审

希赛

软件过程改进 – CMMI

阶段式

组织能力成熟度

成熟度等级	过程域
已管理级	需求管理、项目计划、配置管理、项目监督与控制、供应商合同管理、度量和分析、过程和产品质量保证
已定义级	需求开发、技术解决方案、产品集成、验证、确认、组织级过程焦点、组织级过程定义、组织级培训、集成项目管理、风险管理、集成化的团队、决策分析和解决方案、组织级集成环境
定量管理级	组织级过程性能、定量项目管理
优化级	组织级改革与实施、因果分析和解决方案

连续式分组	过程域
过程管理	组织级过程焦点、组织级过程定义、组织级培训、组织级过程性能、组织级改革与实施
项目管理	项目计划、项目监督与控制、供应商合同管理、集成项目管理、风险管理、集成化的团队、定量项目管理
工程	需求管理、需求开发、技术解决方案、产品集成、验证、确认
支持	配置管理、度量和分析、过程和产品质量保证、决策分析和解决方案、组织级集成环境、因果分析和解决方案

软件过程能力

连续式

希赛

软件配置管理 – 配置项与配置库

关于配置项

- 配置项是构成产品配置的主要元素，配置项主要有以下两大类：

（1）属于产品组成部分的工作成果：如需求文档、设计文档、源代码和测试用例等；

（2）属于项目管理和机构支撑过程域产生的文档：如工作计划、项目质量报告和项 目跟踪报告等。 这些文档虽然不是产品的组成部分，但是值得保存。

- 设备清单、CASE工具操作手册等不属于配置项。
- 每个配置项的主要属性有：名称、标识符、文件状态、版本、作者和日期等。所有配置项都被保存在配置库里，确保不会混淆、丢失。配置项及其历史记录反映了软件的演化过程。

关于配置库

- 开发库（动态库、程序员库、工作库）：可以随意修改。
- 受控库（主库、系统库）：必须先申请，申请通过才有权限修改。
- 产品库（备份库、静态库）：不能修改。

希赛

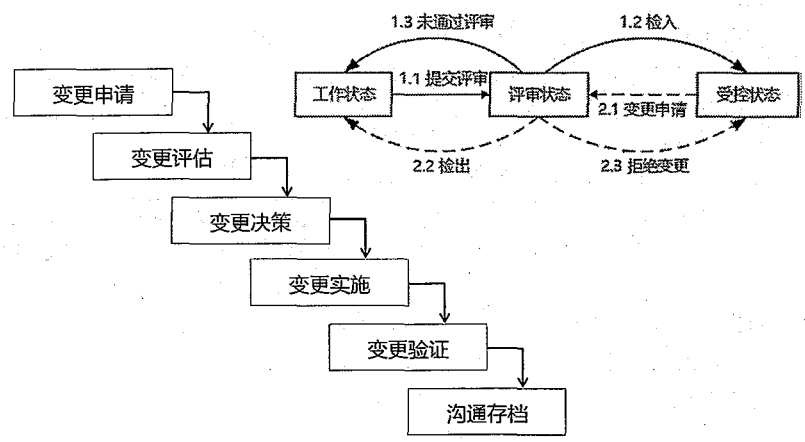
软件配置管理－软件工具

按软件过程活动将软件工具分为：

- 软件开发工具：需求分析工具、设计工具、编码与排错工具。
- 软件维护工具：版本控制工具（VSS、CVS、SCCS、SVN）、文档分析工具、开发信息库工具、逆向工程工具、再工程工具。
- 软件管理和软件支持工具：项目管理工具、配置管理工具、软件评价工具、软件开发工具的评价和选择。

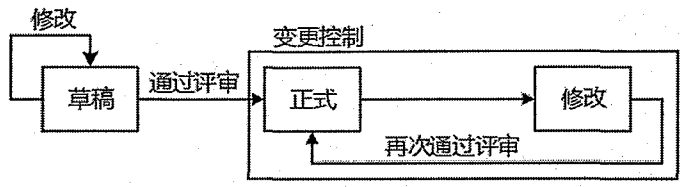
希赛

软件配置管理－变更控制



希赛

软件配置管理 – 版本控制



- 处于草稿状态的配置项的版本号格式为：0.YZ，其中YZ数字范围为01～99。随着草稿的不断完善，YZ的取值应递增。YZ的初值和增幅由开发者自己把握。
- 处于正式发布状态的配置项的版本号格式为：X.Y。其中X为主版本号，取值范围为1～9；Y为次版本号，取值范围为1 ～ 9。配置项第一次正式发布时，版本号为1.0。
- 如果配置项的版本升级幅度比较小，一般只增大Y值，X值保持不变。只有当配置项版本升级幅度比较大时，才允许增大X值。
- 处于正在修改状态的配置项的版本号格式为：X.YZ。在修改配置项时，一般只增大Z值，X.Y值保持不变。

希赛

风险管理 – 风险的概念

- 关心未来
- 关心变化
- 关心选择

- 风险存在的客观性和普遍性
- 某一具体风险发生的偶然性和大量风险发生的必然性
- 风险的可变性
- 风险的多样性和多层次性
- 基本属性：随机性和相对性

希赛

风险管理 – 风险的分类

- **项目风险**
  - 潜在的预算、进度、人员和组织、资源、用户和需求问题
  - 项目复杂性、规模和结构的不确定性
- **技术风险**
  - 潜在的设计、实现、接口、测试和维护方面的问题
  - 规格说明的多义性、技术上的不确定性、技术陈旧、最新技术（不成熟）
- **商业风险**
  - 市场风险：系统虽然很优秀但不是市场真正所想要的
  - 策略风险：系统不再符合企业的信息系统战略
  - 销售风险：开发了销售部门不清楚如何推销的系统
  - 管理风险：由于重点转移或人员变动而失去上级支持
  - 预算风险：开发过程没有得到预算或人员的保证

希赛

风险管理 – 风险曝光度

风险曝光度（Risk Exposure）：计算方法是风险出现的概率乘以风险可能造成的损失。

假设正在开发的软件项目可能存在一个未被发现的错误，而这个错误出现的概率是0.5%，给公司造成的损失将是1000000元，那么这个错误的风险曝光度就应为1000000×0.5%=5000元。

希赛

项目管理工具

能做什么（项目管理相关的工作辅助）：任务调度、成本估算、资源分配、预算跟踪、人时统计、配置控制，确定关键路径、松弛时间、超前时间和滞后时间，生成一定格式的报表和报告。

不能做什么（开发技术相关辅助工作）：不能指导软件设计人员按软件生存周期各个阶段的适用技术进行设计工作。

希赛

案例分析练习题

某软件企业为电信公司开发一套网上营业厅系统，以提升服务的质量和效率。项目组经过分析，列出了项目开发过程中的主要任务、持续时间和所依赖的前置任务，如下表所示。在此基础上，可分别绘制出管理该系统开发过程的PERT图和Gantt图。

【问题1】请用300字以内的文字分别解释说明PERT图和Gantt图的具体含义，并说明两种方法所描述开发过程的差异。

【问题2】根据表中所示活动及其各项活动之间的依赖关系，分别计算对应PERT图中活动C~H的松弛时间（Slack Time）填入（a）~（f）中。

【问题3】根据表中所示活动及其各项活动之间的依赖关系，计算对应PERT图中的关键路径及所需工期。

【问题4】如果将关键路径对应的工期作为期望工期；通过资源调配和任务调度，可以使得最优工期比期望工期少2周时间；根据项目组开发经验，解决项目开发过程中所有可能会遇到的问题最多需要8周时间，即最差工期比期望工期多8周时间。请计算项目最可能的开发工期。

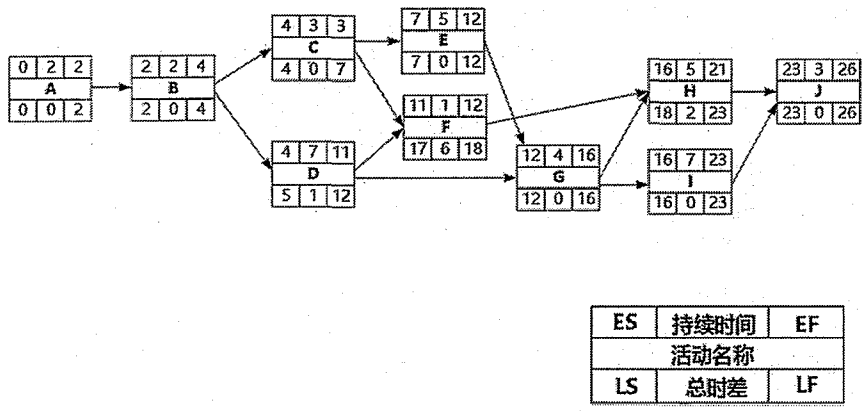
希赛

案例分析练习题

任务名称	持续时间（周）	前置任务	松弛时间
A. 需求获取	2	-	-
B. 业务流程建模	2	A	-
C. 数据建模	3	B	(a)
D. 过程设计	7	B	(b)
E. 数据库设计	5	C	(c)
F. 界面设计	1	C,D	(d)
G. 报表设计	4	D,E	(e)
H. 程序设计	5	F,G	(f)
I. 测试和文档	7	G	-
J. 安装	3	H,I	-

希赛

案例分析练习题



希赛

案例分析练习题

【问题1】

PERT（项目评估与评审技术）图是一种图形化的网络模型，描述一个项目中任务和任务之间的关系，每个节点表示一个任务，通常包括任务编号、名称、开始和结束时间、持续时间和松弛时间。

Gantt图是一种简单的水平条形图，它以一个日历为基准描述项目任务，横坐标表示时间，纵坐标表示任务，图中的水平线段表示对一个任务的进度安排，线段的起点和终点对应在横坐标上的时间分别表示该任务的开始时间和结束时间，线段的长度表示完成该任务所需的时间。

PERT图主要描述不同任务之间的依赖关系；Gantt图主要描述不同任务之间的重叠关系。

【问题2】

（a）0　（b）1　（c）0　（d）6　（e）0　（f）2

【问题3】

关键路径：A—B—C—E—G—I—J

所需工期：26周

【问题4】

最可能的开发工期=(最优工期+期望工期×4+最差工期)/6=(24+26×4+34)/6=27周

希赛

计算机组成与体系结构

希赛

课程内容提要

- Flynn分类法（★★）
- CISC与RISC（★★）
- 存储系统（★★★）
- 嵌入式系统（★★★★）

希赛

计算机体系结构分类 – Flynn

体系结构类型	结构	关键特性	代表
单指令流单数据流 SISD	控制部分：一个 处 理 器：一个 主存模块：一个		单处理器系统
单指令流多数据流 SIMD	控制部分：一个 处 理 器：多个 主存模块：多个	各处理器以异步的形式执行同一条指令	并行处理机 阵列处理机 超级向量处理机
多指令流单数据流 MISD	控制部分：多个 处 理 器：一个 主存模块：多个	被证明不可能，至少是不实际	目前没有，有文献称流水线计算机为此类
多指令流多数据流 MIMD	控制部分：多个 处 理 器：多个 主存模块：多个	能够实现作业、任务、指令等各级全面并行	多处理机系统 多计算机

希赛

CISC与RISC

指令系统类型	指令	寻址方式	实现方式	其它
CISC（复杂）	数量多，使用频率差别大，可变长格式	支持多种	微程序控制技术（微码）	研制周期长
RISC（精简）	数量少，使用频率接近，定长格式，大部分为单周期指令，操作寄存器，只有Load/Store操作内存	支持方式少	增加了通用寄存器；硬布线逻辑控制为主；适合采用流水线	优化编译，有效支持高级语言

希赛

冯·诺依曼结构与哈佛结构

冯·诺依曼结构

冯·诺依曼结构也称普林斯顿结构，是一种将程序指令存储器和数据存储器合并在一起的存储器结构。

特点：

（1）一般用于PC处理器，如I3，I5，I7处理器

（2）指令与数据存储器合并在一起

（3）指令与数据都通过相同的数据总线传输

哈佛结构

哈佛结构是一种将程序指令存储和数据存储分开的存储器结构。哈佛结构是一种并行体系结构，它的主要特点是将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个独立的存储器，每个存储器独立编址、独立访问。

特点：

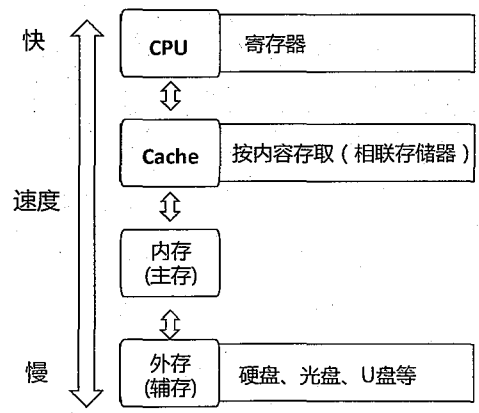
（1）一般用于嵌入式系统处理器（DSP）

（2）指令与数据分开存储，可以并行读取，有较高数据的吞吐率

（3）有4条总线：指令和数据的数据总线与地址总线

希赛

层次化存储结构



希赛

Cache – 概念

- ✓ Cache的功能:提高CPU数据输入输出的速率，突破冯·诺依曼瓶颈，即CPU与存储系统间数据传送带宽限制。
- ✓ 在计算机的存储系统体系中，Cache是访问速度最快的层次。
- ✓ 使用Cache改善系统性能的依据是程序的局部性原理。

如果以h代表对Cache的访问命中率，t_1表示Cache的周期时间，t_2表示主存储器周期时间，以读操作为例，使用"Cache+主存储器"的系统的平均周期为t_3，则：

$$t_3 = h \times t_1 + (1-h) \times t_2$$

其中，（1-h）又称为失效率（未命中率）。

希赛

局部性原理

- ✓ 时间局部性：指程序中的某条指令一旦执行，不久以后该指令可能再次执行，典型原因是由于程序中存在着大量的循环操作。
- ✓ 空间局部性：指一旦程序访问了某个存储单元，不久以后，其附近的存储单元也将被访问，即程序在一段时间内所访问的地址可能集中在一定的范围内，其典型情况是程序顺序执行。
- ✓ 工作集理论：工作集是进程运行时被频繁访问的页面集合。

例:

```
int i,j,s=0,n=10000;
for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
        s+=j;
printf("结果为:%d",s)
```

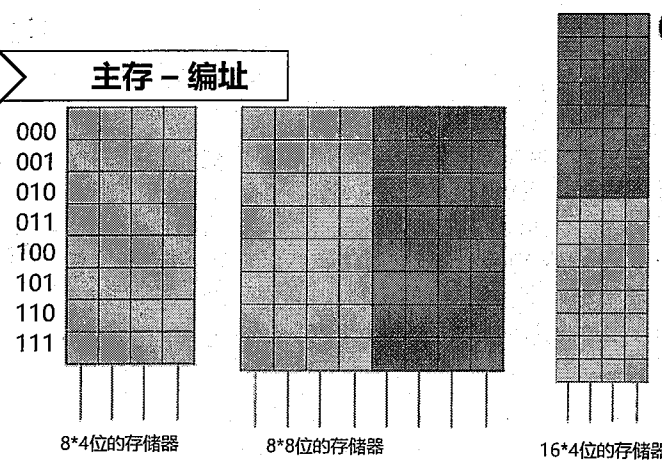
希赛

主存－分类

- **随机存取存储器**
  - DRAM（Dynamic RAM，动态RAM）-SDRAM
  - SRAM（Static RAM，静态）
- **只读存储器**
  - MROM（Mask ROM，掩模式ROM）
  - PROM（Programmable ROM，一次可编程 ROM）
  - EPROM（Erasable PROM，可擦除的 PROM）
  - 闪速存储器（flash memory，闪存）

希赛

主存－编址



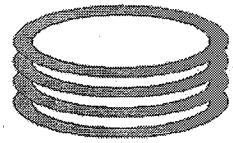
- 内存地址从AC000H到C7FFFH，共有（ ）K个地址单元，如果该内存地址按字（16bit）编址，由28片存储器芯片构成。已知构成此内存的芯片每片有16K个存储单元，则该芯片每个存储单元存储（ ）位。

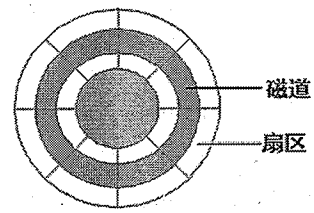
（1）A．96　　B．112　　C．132　　D．156

（2）A．4　　B．8　　C．16　　D．24

希赛

磁盘结构与参数





◆ 存取时间 = 寻道时间+等待时间(平均定位时间+转动延迟)

注意：寻道时间是指磁头移动到磁道所需的时间；等待时间为等待读写的扇区转到磁头下方所用的时间。

希赛

磁盘结构与参数

◆ **学员疑问：**

存取时间=寻道时间+等待时间（平均定位时间+转动延迟） 这个公式我认为有问题。

我认为应是 存取时间=寻道时间+转动延迟+存（或取）磁盘扇区内容的时间；

麻烦到时说一下讲义里如果是对的该怎么理解，我的如果是错的错在哪里？

◆ **平均存取时间(Average Access Time)，是指磁头找到指定数据的平均时间。**

平均存取时间是指磁头找到指定数据的平均时间，通常它是硬盘平均寻道时间和平均潜伏时间（等待时间）之和。平均存取时间最能代表硬盘找到某一数据所用的时间，数值越小越好。

◆ **平均访问时间=平均寻道时间+平均等待时间**

希赛

试题

● 假设某磁盘的每个磁道划分成11个物理块，每块存放1个逻辑记录。逻辑记录R_0，R_1，．．．，R_9，R_{10}存放在同一个磁道上，记录的存放顺序如下表所示：

物理块	1	2	3	4	5	6	7	8	9	10	11
逻辑记录	R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}

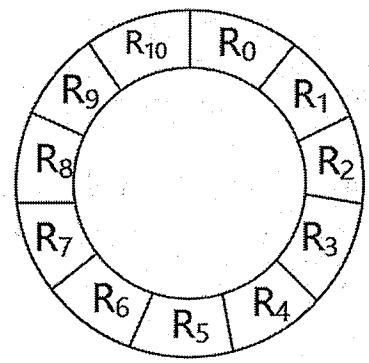
如果磁盘的旋转周期为33ms，磁头当前处在R_0的开始处。若系统使用单缓冲区顺序处理这些记录，每个记录处理时间为3ms，则处理这11个记录的最长时间为（48）；若对信息存储进行优化分布后，处理11个记录的最少时间为（49）。

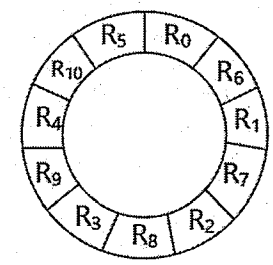
（48）A．33ms　　B．336ms　　C．366ms　　D．376ms

（49）A．33ms　　B．66ms　　C．86ms　　D．93ms

希赛

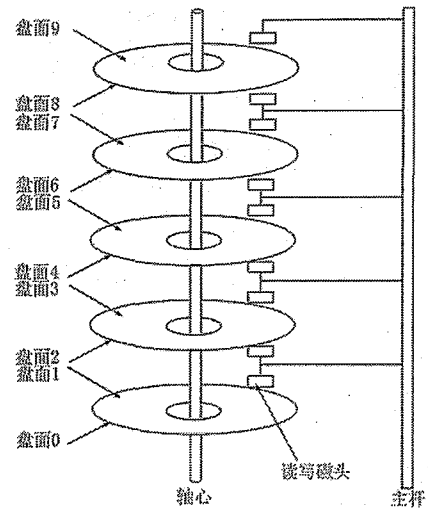
试题





希赛

磁盘结构

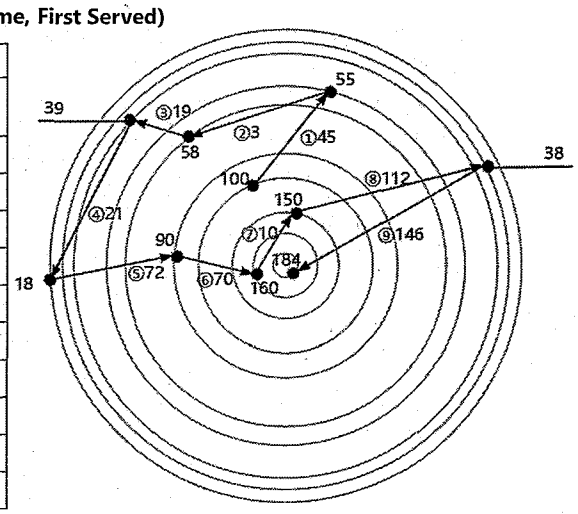


希赛

磁盘调度 - FCFS

先来先服务FCFS(First-Come, First Served)

（从100号磁道开始）		
序号	被访问的下一个磁道号	移动距离（磁道数）
①	55	45
②	58	3
③	39	19
④	18	21
⑤	90	72
⑥	160	70
⑦	150	10
⑧	38	112
⑨	184	146
平均寻道长度：55.3		

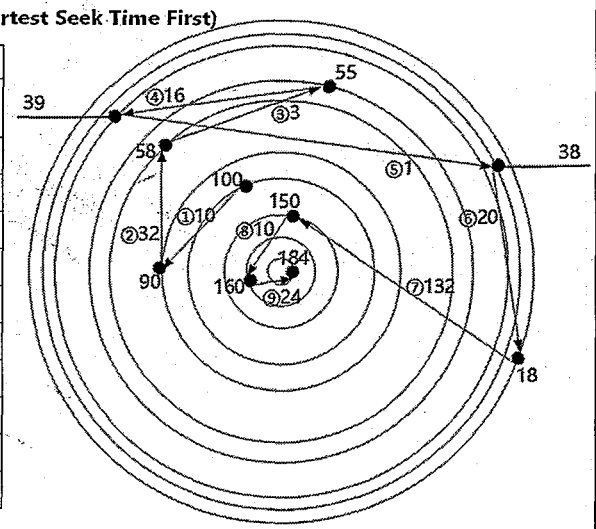


希赛

磁盘调度 – SSTF

最短寻道时间优先SSTF(Shortest Seek Time First)

（从100号磁道开始）		
序号	被访问的下一个磁道号	移动距离（磁道数）
①	90	10
②	58	32
③	55	3
④	39	16
⑤	38	1
⑥	18	20
⑦	150	132
⑧	160	10
⑨	184	24
平均寻道长度：27.5		



希赛

练习题

在磁盘调度管理中，应先进行移臂调度，再进行旋转调度。假设磁盘移动臂位于21号柱面上，进程的请求序列如下表所示。如果采用最短移臂调度算法，那么系统的响应序列应为（ ）。

A.②⑧③④⑤①⑦⑥⑨

B.②③⑧④⑥⑨①⑤⑦

C.①②③④⑤⑥⑦⑧⑨

D.②⑧③⑤⑦①④⑥⑨

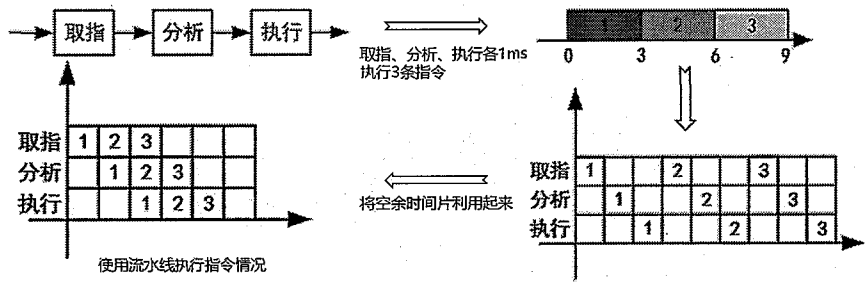
请求序列	柱面号	磁头号	扇区号
①	17	8	9
②	23	6	3
③	23	9	6
④	32	10	5
⑤	17	8	4
⑥	32	3	10
⑦	17	7	9
⑧	23	10	4
⑨	38	10	8

希赛

流水线－概念

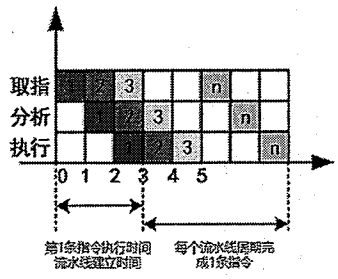
相关参数计算：流水线执行时间计算、流水线吞吐率、流水线加速比、流水线效率

流水线是指在程序执行时多条指令重叠进行操作的一种准并行处理实现技术。各种部件同时处理是针对不同指令而言的，它们可同时为多条指令的不同部分进行工作，以提高各部件的利用率和指令的平均执行速度



希赛

流水线－流水线计算



- ✓ 流水线周期为执行时间最长的一段
- ✓ 流水线计算公式为：

  1条指令执行时间 + (指令条数-1)*流水线周期

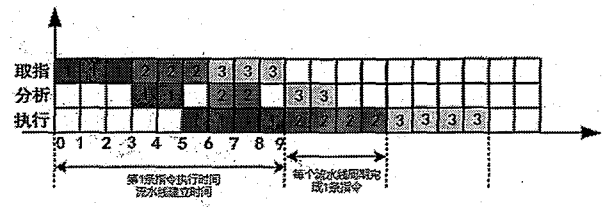
  ①理论公式：(t1+t2+..+tk)+(n-1)*Δt

  ②实践公式：k*Δt +(n-1)*Δt

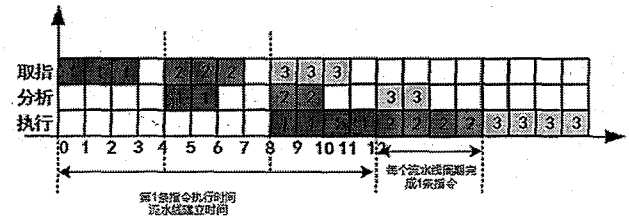
一条指令的执行过程可以分解为取指、分析和执行三步，在取指时间t取指=3ΔΔt、分析时间t分析=2ΔΔt、执行时间t执行=4ΔΔt的情况下，若按串行方式执行，则10条指令全部执行完需要（ ）ΔΔt；若按流水线的方式执行，流水线周期为（ ）ΔΔt，则10条指令全部执行完需要（ ）ΔΔt。

流水线 – 流水线计算

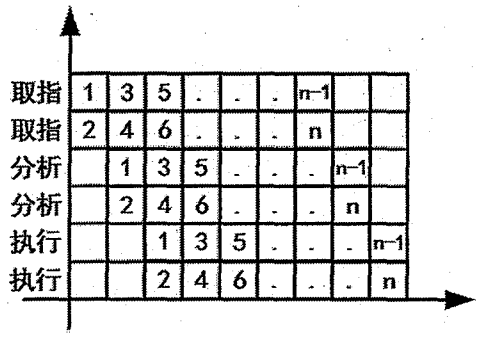
理论公式：(t1+t2+..+tk)+(n-1)*Δt



实践公式：k*Δt +(n-1)*Δt



流水线 – 超标量流水线



希赛

流水线－流水线吞吐率计算

流水线的吞吐率（Though Put rate，TP）是指在单位时间内流水线所完成的任务数量或输出的结果数量。计算流水线吞吐率的最基本的公式如下：

$$TP = \frac{\text{指令条数}}{\text{流水线执行时间}}$$

流水线最大吞吐率：

$$TP_{\max} = \lim_{n \to \infty} \frac{n}{(k+n-1)\Delta t} = \frac{1}{\Delta t}$$

希赛

流水线－流水线加速比计算

完成同样一批任务，不使用流水线所用的时间与使用流水线所用的时间之比称为流水线的加速比。计算流水线加速比的基本公式如下：

$$S = \frac{\text{不使用流水线执行时间}}{\text{使用流水线执行时间}}$$

希赛

总线

一条总线同一时刻仅允许一个设备发送，但允许多个设备接收。
所以总线是半双工模式。

总线的分类：

- 数据总线（Data Bus）：在CPU与RAM之间来回传送需要处理或是需要储存的数据。
- 地址总线（Address Bus）：用来指定在RAM（Random Access Memory）之中储存的数据的地址。
- 控制总线（Control Bus）：将微处理器控制单元（Control Unit）的信号，传送到周边设备，一般常见的为 USB Bus和1394 Bus。

希赛

嵌入式系统开发设计－芯片

1、DSP
DSP芯片，也称数字信号处理器，是一种特别适合于进行数字信号处理运算的微处理器，其主要应用是实时快速地实现各种数字信号处理算法。

2、SoC
System on Chip，简称Soc，也即片上系统。从狭义角度讲，它是信息系统核心的芯片集成，是将系统关键部件集成在一块芯片上；从广义角度讲， SoC是一个微小型系统，如果说中央处理器（CPU）是大脑，那么SoC就是包括大脑、心脏、眼睛和手的系统。

3、MPU
微机中的中央处理器（CPU）称为微处理器（MPU），是构成微机的核心部件，也可以说是微机的心脏。它起到控制整个微型计算机工作的作用，产生控制信号对相应的部件进行控制，并执行相应的操作。

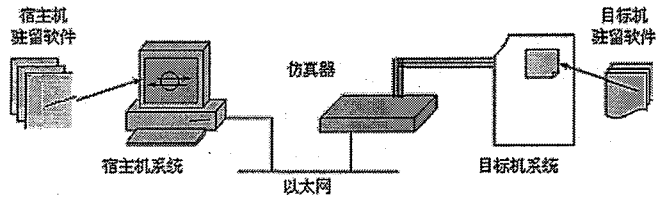
4、MCU
微控制单元(Microcontroller Unit；MCU) ，又称单片微型计算机(Single Chip Microcomputer)或者单片机，是把中央处理器(Central Process Unit；CPU)的频率与规格做适当缩减，并将内存(memory)、计数器(Timer)、USB、A/D转换、UART、PLC、DMA等周边接口，甚至LCD驱动电路都整合在单一芯片上，形成芯片级的计算机，为不同的应用场合做不同组合控制。

希赛

嵌入式系统开发设计 – 交叉开发环境

✓ 交叉编译器
✓ 交叉链接器
✓ 调试器

✓ 动态装载器
✓ 链接装载器
✓ 调试监视器
✓ 调试代理



目标代码（COFF、ELF）、调试信息

希赛

嵌入式系统初始化过程

片级初始化 → 板级初始化 → 系统初始化

◆ 片级初始化

完成嵌入式微处理器的初始化，包括设置嵌入式微处理器的核心寄存器和控制寄存器、嵌入式微处理器核心工作模式和嵌入式微处理器的局部总线模式等。片级初始化把嵌入式微处理器从上电时的默认状态逐步设置成系统所要求的工作状态。这是一个纯硬件的初始化过程。

◆ 板级初始化

完成嵌入式微处理器以外的其他硬件设备的初始化。另外，还需设置某些软件的数据结构和参数，为随后的系统级初始化和应用程序的运行建立硬件和软件环境。这是一个同时包含软硬件两部分在内的初始化过程。

◆ 系统初始化

该初始化过程以软件初始化为主，主要进行操作系统的初始化。BSP将对嵌入式微处理器的控制权转交给嵌入式操作系统，由操作系统完成余下的初始化操作，包含加载和初始化与硬件无关的设备驱动程序，建立系统内存区，加载并初始化其他系统软件模块，如网络系统、文件系统等。最后，操作系统创建应用程序环境，并将控制权交给应用程序的入口。

希赛

校验码-奇偶校验

- 奇偶校验码的编码方法是：由若干位有效信息（如一个字节），再加上一个二进制位（校验位）组成校验码。
- 奇校验：整个校验码（有效信息位和校验位）中“1”的个数为奇数。
  偶校验：整个校验码（有效信息位和校验位）中“1”的个数为偶数。

  奇偶校验，可检查1位的错误，不可纠错。

希赛

校验码-循环校验码CRC

CRC校验，可检错，不可纠错。

- CRC的编码方法是：在k位信息码之后拼接r位校验码。应用CRC码的关键是如何从k位信息位简便地得到r位校验位（编码），以及如何从k+r位信息码判断是否出错。
- 循环冗余校验码编码规律如下：

  ① 把待编码的N位有效信息表示为多项式M(X)；

  ② 把M(X)左移K位，得到$M(X)\times X^K$，这样空出了K位，以便拼装K位余数（即校验位）；

  ③ 选取一个K+1位的产生多项式G(X)，对$M(X)\times X^K$做模2除；

  ④ 把左移K位以后的有效信息与余数R(X)做模2加减，拼接为CRC码，此时的CRC码共有N+K位。
- 把接收到的CRC码用约定的生成多项式G(X)去除，如果正确，则余数为0；如果某一位出错，则余数不为0。不同的位数出错其余数不同，余数和出错位序号之间有惟一的对应关系。

校验码-循环校验码CRC

希赛

什么是模2除法，它和普通的除法有何区别？

模2除法是指在做除法运算的过程中不计其进位的除法。

例如，10111对110进行模2除法为：

```
       110
110)10111
    110
    0111
     110
     0011
```

普通除法运算过程：

```
       11
110)10111
    0110
     1011
     0110
     0101
```

校验码-循环校验码CRC

希赛

例：原始报文为“11001010101”，其生成多项式为：“x^4+x^3+x+1”。对其进行CRC编码后的结果为？

x^4+x^3+x+1

⇩

$x^4+x^3+0\times x^2+x+1$

1 1 0 1 1

```
11011)110010101010000
      11011
       10010
       11011
        10011
        11011
         10000
         11011
          10111
          11011
           11000
           11011
            11000
            11011
             0011
```

希赛

校验码-海明校验码

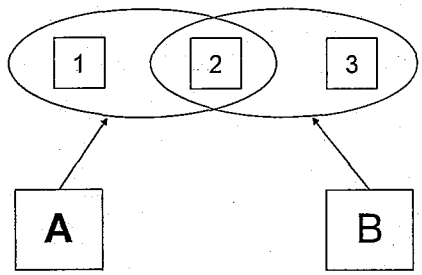
海明校验，可检错，也可纠错。

海明校验码的原理是：在有效信息位中加入几个校验位形成海明码，使码距比较均匀地拉大，并把海明码的每个二进制位分配到几个奇偶校验组中。当某一位出错后，就会引起有关的几个校验位的值发生变化，这不但可以发现错误，还能指出错误的位置，为自动纠错提供了依据

$$2^r \geq m+r+1$$

希赛

校验码-海明校验码



希赛

校验码-海明校验码

例：求信息1011的海明码。

（1）$2^r>=4+r+1$，确定校验码为3位：$2^3>=4+3+1$。分别放在$2^0=1$、$2^1=2$、$2^2=4$位。

（2）列出校验位公式。

$7=2^2+2^1+2^0$，$6=2^2+2^1$，$5=2^2+2^0$，$3=2^1+2^0$；

$r_2= I_4 \oplus I_3 \oplus I_2$；

$r_1= I_4 \oplus I_3 \oplus I_1$；

$r_0= I_4 \oplus I_2 \oplus I_1$；

（3）根据公式得$r_2=0$，$r_1=0$，$r_0=1$。

（4）将数据加入表格，如表所示。

7	6	5	4	3	2	1	位数
I_4	I_3	I_2		I_1			信息位
			r_2		r_1	r_0	校验位

若收到的信息为：1011101

则：

$r_2 \oplus I_4 \oplus I_3 \oplus I_2=1 \oplus 1 \oplus 0 \oplus 1=1$

$r_1 \oplus I_4 \oplus I_3 \oplus I_1=0 \oplus 1 \oplus 0 \oplus 1=0$

$r_0 \oplus I_4 \oplus I_2 \oplus I_1=1 \oplus 1 \oplus 1 \oplus 1=0$

7	6	5	4	3	2	1	位数
1	0	1		1			信息位
			0		0	1	校验位

希赛

练习题

- 已知数据信息为16位，最少应附加（　）位校验位，以实现海明码纠错。

A．4

B．5

C．6

D．7

希赛

系统配置与性能评价

希赛

课程内容提要

- 性能指标（★）
- 阿姆达尔解决方案（★★★）
- 性能评价方法（★★★）

希赛

系统性能设计 – 性能指标

- 字长和数据通路宽度
- 主存容量和存取速度
- 运算速度 → 主频与CPU时钟周期
  - CPI与IPC
  - MIPS与MFLOPS
- 吞吐量与吞吐率
- 响应时间（RT）
- 与完成时间（TAT）
- 兼容性

MIPS = 指令条数/(执行时间×10^6) = 主频/CPI = 主频×IPC

MFLOPS = 浮点操作次数/(执行时间×10^6)

希赛

系统性能设计 – 阿姆达尔（Amdahl）解决方案

✓ 对系统中某组件采用某种更快的执行方式，所获得的系统性能的改变程度，取决于该组件被使用的频率，或所占总执行时间的比例。加速比计算公式如下：

$$R=\frac{T_p}{T_i}=\frac{1}{(1-F_e)+F_e/S_e}$$

✓ 其中，T_p表示不使用改进组件时完成整个任务的时间，T_i表示使用改进组件时完成整个任务的时间。加速比主要取决于两个因素：

（1）在原有的系统上，能被改进的部分在总执行时间中所占的比例。这个值称为改进比例，记为F_e，它总是小于1。

（2）通过改进的执行方式所取得的性能提高，即如果整个系统使用了改进的执行方式，那么，系统的执行速度会有多少提高，这个值等于在原来的条件下系统的执行时间与使用改进组件后系统的执行时间之比，记为S_e，它总大于1。

希赛

系统性能设计 – 阿姆达尔（Amdahl）解决方案

✓ 阿姆达尔（Amdahl）定律量化定义了通过改进系统中某个组件的性能，使系统整体性能提高的程度。假设某一功能的处理时间为整个系统运行时间的60%，若使该功能的处理速度提高至原来的5倍，则根据阿姆达尔定律，整个系统的处理速度可提高至原来的（ ）倍。

A. 1.333　　B. 1.923　　C. 1.5　　D. 1.829

希赛

性能评价方法

✓ 时钟频率法：以时钟频率高低衡量速度。

✓ 指令执行速度法：表示机器运算速度的单位是MIPS。

✓ 等效指令速度法（ Gibson mix，吉普森混合法）：通过各类指令在程序中所占的比例（ W_i）进行计算得到的。特点：考虑指令比例不同的问题。

✓ 数据处理速率法（PDR）：PDR值的方法来衡量机器性能，PDR值越大，机器性能越好。PDR = L/R　特点：考虑CPU+存储

✓ 综合理论性能法（CTP）：CTP用MTOPS（Million Theoretical Operations Per Second，每秒百万次理论运算）表示。CTP的估算方法是，首先算出处理部件每个计算单元的有效计算率，再按不同字长加以调整，得出该计算单元的理论性能，所有组成该处理部件的计算单元的理论性能之和即为CTP。

✓ 基准程序法：把应用程序中用得最多、最频繁的那部分核心程序作为评估计算机系统性能的标准程序，称为基准测试程序（benchmark）。基准程序法是目前一致承认的测试系统性能的较好方法。

希赛

性能评价方法

真实的程序->核心程序->小型基准程序->合成基准程序

✓ Dhrystone基准程序：它是一个综合性的整数基准测试程序，是为了测试编译器和CPU处理整数指令和控制功能的有效性，人为地选择一些典型指令综合起来形成的测试程序。

✓ Linpack基准程序：它是国际上最流行的用于测试高性能计算机系统浮点性能的测试。

✓ Whetstone基准程序： 它是用Fortran语言编写的综合性测试程序，主要由执行浮点运算、功能调用、数组变址、条件转移和超越函数的程序组成。

✓ SPEC基准程序一种是测试计算机完成单项任务有多快，称为速度测试；另一种是测试计算机在一定时间内能完成多少项任务，称为吞吐率测试。

✓ TPC基准程序： TPC（Transaction Processing Council，事务处理委员会）基准程序用以评测计算机在事务处理、数据库处理、企业管理与决策支持系统等方面的性能。该基准程序的评测结果用每秒完成的事务处理数TPC来表示。

TPC-A基准程序规范用于评价在OLTP环境下的数据库和硬件的性能；

TPC-B测试的是不包括网络的纯事务处理量，用于模拟企业计算环境；

TPC-C测试的是联机订货系统；

TPC-D、TPC-H和TPC-R测试的都是决策支持系统，其中TPC-R允许有附加的优化选项；

TPC-E测试的是大型企业信息服务系统。

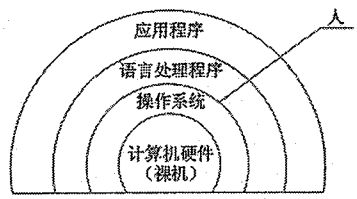
TPC-W是基于Web 应用的基准程序，用来测试一些通过Internet进行市场服务和销售的商业行为，所以TPC-W可以看作是一个服务器的测试标准。

希赛

操作系统

希赛

操作系统 – 概述



- 管理系统的硬件、软件、数据资源
- 控制程序运行
- 人机之间的接口
- 应用软件与硬件之间的接口

- 进程管理
- 存储管理
- 文件管理
- 作业管理
- 设备管理

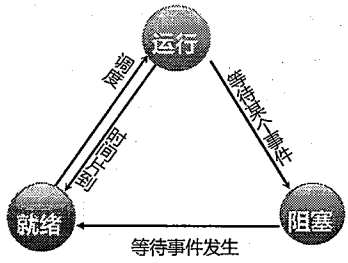
希赛

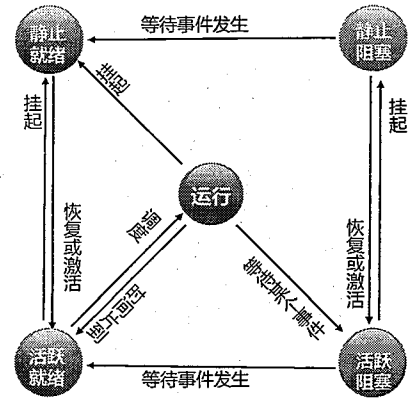
课程内容提要

- **进程管理**
  - 进程的状态（★）
  - 前趋图（★★★★）
  - 信号量与PV操作（★★★★★）
  - 死锁及银行家算法（★★）
- **存储管理**
  - 段页式存储（★★★★）
  - 页面置换算法（★）
- **文件管理**
  - 绝对路径与相对路径（★）
  - 索引文件（★★★）
  - 位示图（★★）
- **作业管理**
- **设备管理**
  - 虚设备与SPOOLING技术（★）
- **微内核操作系统**
- **嵌入式操作系统**

希赛

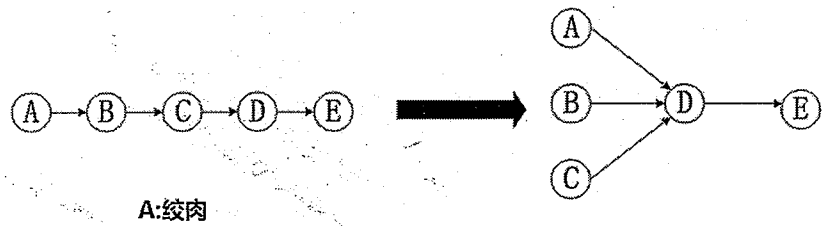
进程管理 - 进程的状态





希赛

进程管理－前趋图



A:绞肉

B:切葱末

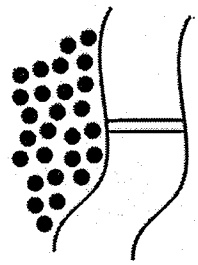
C:切姜末

D:搅拌

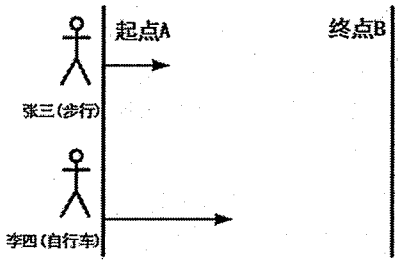
E:包饺子

希赛

进程管理－进程的同步与互斥



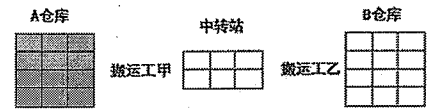
互斥:如千军万马过独木桥



同步：速度有差异，在一定情况停下等待。

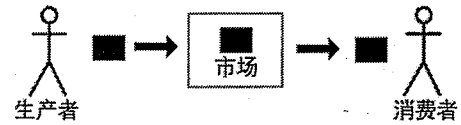
希赛

进程管理 – 进程的同步与互斥



生产者消费者问题

单缓冲区情况



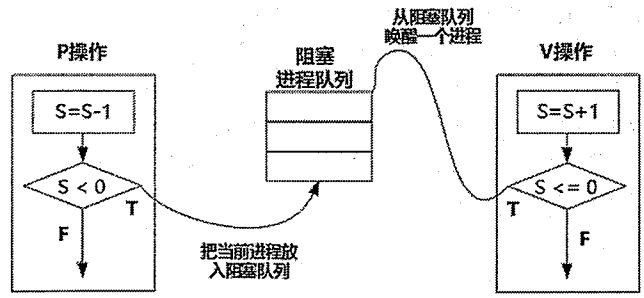
多缓冲区情况

希赛

进程管理 – PV操作

- 临界资源：诸进程间需要互斥方式对其进行共享的资源，如打印机、磁带机等
- 临界区：每个进程中访问临界资源的那段代码称为临界区
- 信号量：是一种特殊的变量

注：P是荷兰语的Passeren，V是荷兰语的Verhoog。



希赛

进程管理 – PV操作

未使用PV操作：

生产者：
生产一个产品;
送产品到缓冲区;

消费者：
从缓冲区取产品;
消费产品;

使用了PV操作：

生产者：
生产一个产品;
P(S1);
送产品到缓冲区;
V(S2);

消费者：
P(S2);
从缓冲区取产品;
V(S1);
消费产品;

S1初值为1，S2初值为0

希赛

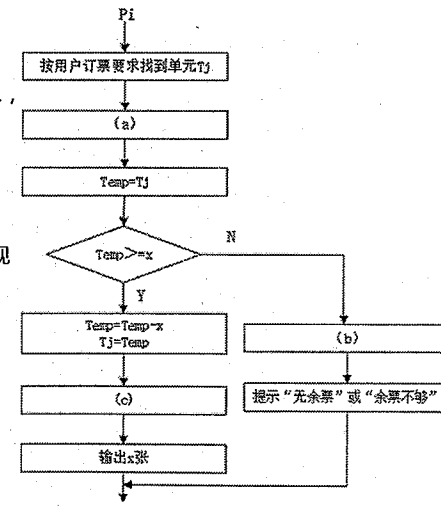
例题讲解

某航空公司机票销售系统有n个售票点，该系统为每个售票点创建一个进程Pi（i=1，2，…，n）管理机票销售。假设Tj（j=1，2，…，m）单元存放某日某航班的机票剩余票数，Temp为Pi进程的临时工作单元，x为某用户的订票张数。初始化时系统应将信号量S赋值为（ ）。Pi进程的工作流程如下图所示，若用P操作和V操作实现进程间的同步与互斥，则图中空(a)，空(b)和空(c)处应分别填入（ ）。

A．0　B．1　C．2　D．3

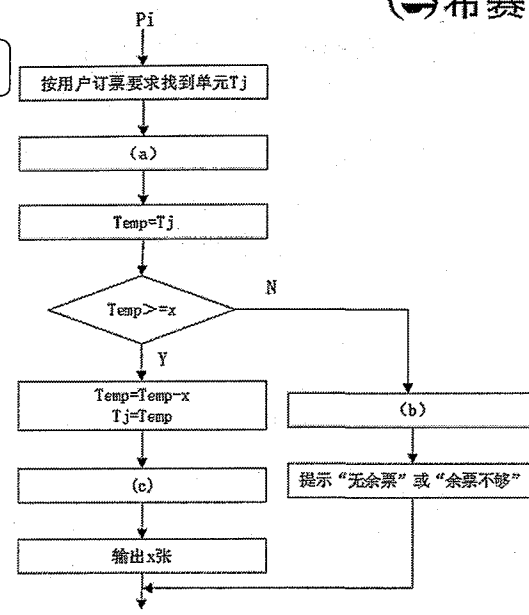
A．P(S),V(S)和V(S)　B．P(S),P(S)和V(S)

C．V(S),P(S)和P(S)　D．V(S),V(S)和P(S)



希赛

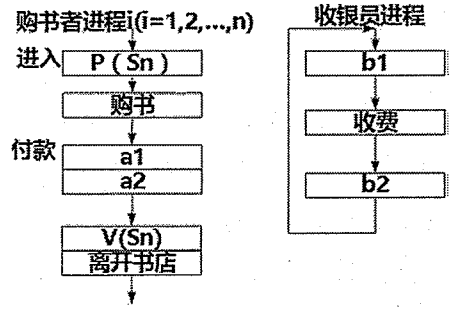
例题讲解



希赛

进程管理 – PV操作

某书店有一个收银员，该书店最多允许n个购书者进入。将收银员和购书者看作不同的进程，其工作流程如下图所示。利用PV操作实现该过程，设置信号量S1、S2和Sn,初值分别为0，0，n。则图中a1和a2应填入（ ），图中b1和b2应填入（ ）。

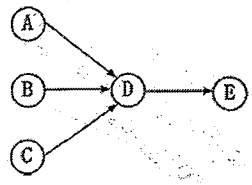


（1）A．V(S1)、P(S2)　B．V(Sn)、P(Sn)　C．P(S1)、V(S2)　D．P(S2)、V(S1)

（2）A．P(Sn)、V(S2)　B．V(Sn)、P(S2)　C．P(S1)、V(S2)　D．P(S2)、V(S1)

希赛

进程管理 – PV操作



A:绞肉

B:切葱末

C:切姜末

D:搅拌

E:包饺子

Sa=0；
Sb=0；
Sc=0；
Sd=0；

进程A:
绞肉；
V(Sa)；

进程B:
切葱末；
V(Sb)；

进程C:
切姜末；
V(Sc)；

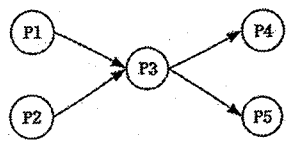
进程D:
P(Sa)；
P(Sb)；
P(Sc)；
搅拌；
V(Sd)；

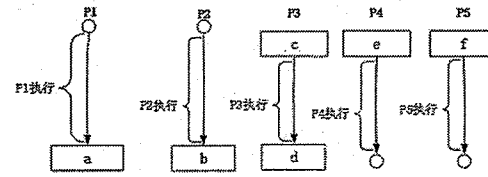
进程E:
P(Sd)；
包饺子；

希赛

进程管理 – PV操作

进程P1、P2、P3、P4、P5的前趋图如下图所示。若用PV操作控制进程并发执行的过程，则需要设置4个信号量S1、S2、S3和S4，且信号量初值都等于零。图中a和b应分别填写（ ），c和d应分别填写（ ），e和f应分别填写（ ）。





（1）A．P（S1）和P（S2）　　B．P（S1）和V（S2）
　　C．V（S1）和V（S2）　　D．V（S1）和P（S2）

（2）A．P（S1）、P（S2）和V（S3）、V（S4）
　　B．P（S1）、P（S2）和P（S3）、P（S4）
　　C．V（S1）、V（S2）和P（S3）、P（S4）
　　D．V（S1）、V（S2）和V（S3）、V（S4）

（3）A．P（S3）和P（S4）　　B．P（S3）和V（S4）
　　C．V（S3）和V（S4）　　D．V（S3）和P（S4）

希赛

进程管理 – PV操作

假设某系统采用非抢占式优先级调度算法，若该系统有两个优先级相同的进程P1和P2，各进程的程序段如下所示，若信号量S1和S2的初值都为0。进程P1和P2并发执行后a、b和c的结果分别为：a=（ ），b=（ ），c=（ ）。

```
P1程序段                    P2程序段
begin{                      begin{
   a:=1;                       b:=2;
   a:=a+1;                     b:=b+1;
   V(S1);                      P(S1);
   c:=a+5;                     b:=a+b;
   P(S2);                      V(S2);
   a:=a+c;                     c:=b+c;
  }                           }
end                         end
```

(4) A. 9　B. 12　C. 13　D. 14

(5) A. 5　B. 6　C. 9　D. 10

(6) A. 4　B. 6　C. 12　D. 13

希赛

进程管理 – PV操作

```
P1程序段                    P2程序段
begin{                      begin{
   a:=1 ;                      b:=2 ;
   a:=a+1 ; // a=2             b:=b+1 ; // b=3
   V ( S1 ) ; ──────────→ P ( S1 ) ; //唤醒
   c:=a+5 ; // c=7             b:=a+b ; // b=5
   P ( S2 ) ; //唤醒 ←────── V ( S2 ) ;
   a:=a+c ; // a=14            c:=b+c ; // c=12
  }                           }
end                         end
```

希赛

进程管理 – 死锁问题

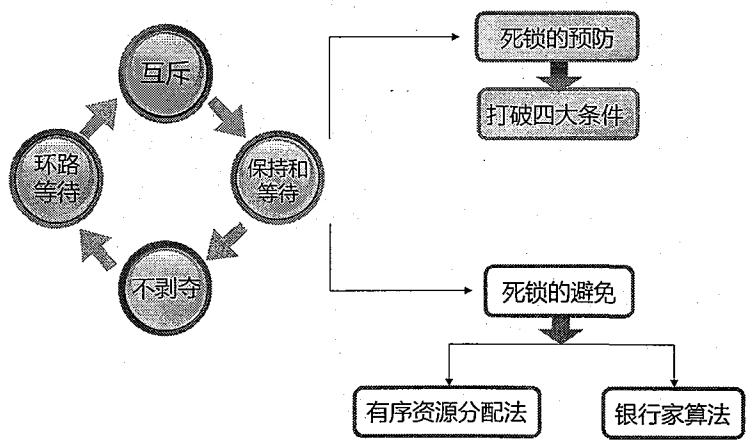
进程管理是操作系统的核心，但如果设计不当，就会出现死锁的问题。如果一个进程在等待一件不可能发生的事，则进程就死锁了。而如果一个或多个进程产生死锁，就会造成系统死锁。

例：系统有3个进程：A、B、C。这3个进程都需要5个系统资源。如果系统至少有多少个资源，则不可能发生死锁。

进程A	进程B	进程C

希赛

进程管理 – 死锁问题



希赛

进程管理 – 银行家算法

银行家算法：分配资源的原则

★ 当一个进程对资源的最大需求量不超过系统中的资源数时可以接纳该进程

★ 进程可以分期请求资源，但请求的总数不能超过最大需求量

★ 当系统现有的资源不能满足进程尚需资源数时，对进程的请求可以推迟分配，但总能使进程在有限的时间里得到资源

希赛

进程管理 – 银行家算法

银行家算法例子：

假设系统中有三类互斥资源R1、R2、R3,可用资源分别是9、8、5。在T0时刻系统中有P1、P2、P3、P4和P5五个进程，这些进程对资源的最大需求量和已分配资源数如下所示，如果进程按_____序列执行，那么系统状态是安全的。

进程资源表

资源 / 进程	最大需求量			已分配资源数		
	R 1	R 2	R 3	R 1	R 2	R 3
P 1	6	5	2	1	2	1
P 2	2	2	1	2	1	1
P 3	8	1	1	2	1	0
P 4	1	2	1	1	2	0
P 5	3	4	4	1	1	3

供选择的答案：

A．P1→P2→P4→P5→P3　　B．P2→P4→P5→P1→P3

C．P2→P1→P4→P5→P3　　D．P4→P2→P5→P1→P3

进程管理 – 银行家算法

首先求剩下的资源数：

R1 = 9-(1+2+2+1+1) = 2

R2 = 8-(2+1+1+2+1) = 1

R3 = 5-(1+1+3) = 0

还需资源数表

进程＼资源	最大需求量			已分配资源数			还需资源数		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P 1	6	5	2	1	2	1	5	3	1
P 2	2	2	1	2	1	1	0	1	0
P 3	8	1	1	2	1	0	6	0	1
P 4	1	2	1	1	2	0	0	0	1
P 5	3	4	4	1	1	3	2	3	1

求序列P 2→P 4→P 5→P 1→P3是否安全：

进程序列P 2→P 4→P 5→P 1→P3运行分析表

进程＼资源	现有资源			需要资源			已经分配			现有+已经分配			完成
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
P2	2	1	0	0	1	0	2	1	1	4	2	1	True
P4	4	2	1	0	0	1	1	2	0	5	4	1	True
P5	5	4	1	2	3	1	1	1	3	6	5	4	True
P1	6	5	4	5	3	1	1	2	1	7	7	5	True
P3	7	7	5	6	0	1	2	1	0	9	8	5	True

进程管理 – 银行家算法

求序列P2→P1→P4→P5→P3是否安全：

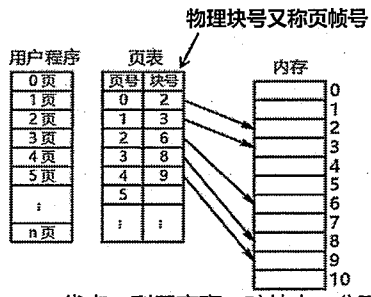
进程序列P2→P1→P4→P5→P3运行分析表

进程＼资源	现有资源			需要资源			已经分配			现有+已经分配			完成
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
P2	2	1	0	0	1	0	2	1	1	4	2	1	True
P1	4	2	1	5	3	1	1	2	1				

这时候，我们发现进程P1需要R1资源为5，我们能提供的R1资源为4，所以序列无法进行下去，为不安全序列。

存储管理 – 页式存储组织

页式存储：将程序与内存均划分为同样大小的块，以页为单位将程序调入内存。



高级程序语言使用逻辑地址；
运行状态，内存中使用物理地址。

逻辑地址=页号+页内地址
物理地址=页帧号+页内地址

例如，页式存储系统中，每个页的大小为4KB。
逻辑地址：
10 1100 1101 1110
对应的物理地址为：
110 1100 1101 1110

★ 优点：利用率高，碎片小，分配及管理简单

★ 缺点：增加了系统开销；可能产生抖动现象

存储管理 – 页式存储组织

高级程序语言中使用

1:在内存中
0:不在内存中

1:内容被修改过
0:内容未被修改

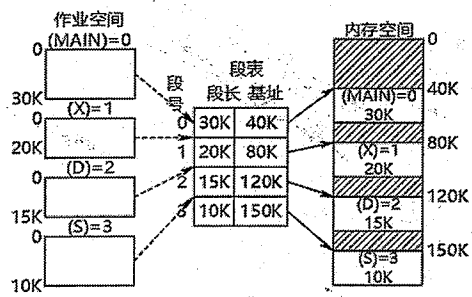
内存中使用

1:最近访问过
0:最近未被访问

页号（逻辑）	页帧号（物理）	状态位	访问位	修改位
0	2	1	1	0
1	3	1	0	1
2	5	1	1	0
3	—	0	0	0
4	—	0	0	0
5	6	1	1	1

存储管理 – 段式存储组织

段式存储：按用户作业中的自然段来划分逻辑空间，然后调入内存，段的长度可以不一样。



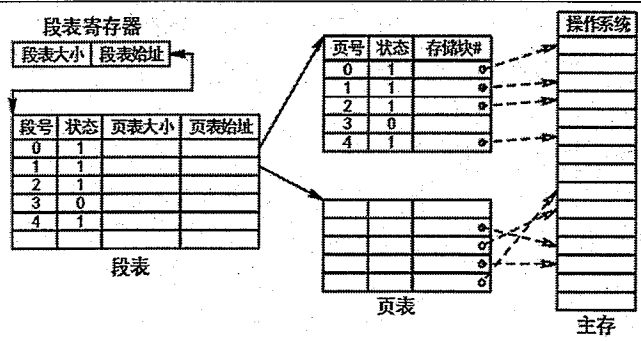
合法段地址：（0，25K）
非法段地址：（0，35K）

★ 优点：多道程序共享内存，各段程序修改互不影响

★ 缺点：内存利用率低，内存碎片浪费大

存储管理 – 段页式存储组织

段页式存储：段式与页式的综合体。先分段，再分页。1个程序有若干个段，每个段中可以有若干页，每个页的大小相同，但每个段的大小不同。



★ 优点：空间浪费小、存储共享容易、存储保护容易、能动态连接

★ 缺点：由于管理软件的增加，复杂性和开销也随之增加，需要的硬件以及占用的内容也有所增加，使得执行速度大大下降

希赛

存储管理－快表

★ 快表是一块小容量的相联存储器（Associative Memory），由高速缓存器组成，速度快，并且可以从硬件上保证按内容并行查找，一般用来存放当前访问最频繁的少数活动页面的页号。

快表：将页表存于Cache上；慢表：将页表存于内存上。

希赛

练习题

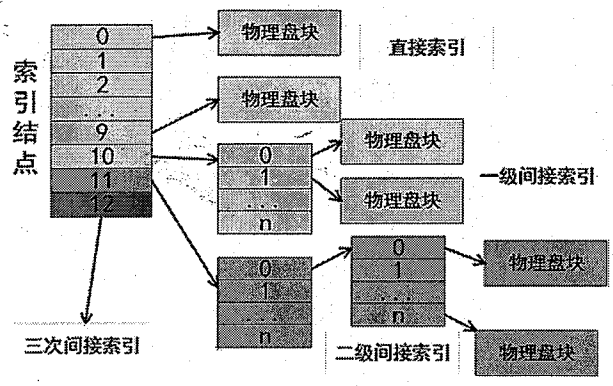
进程P有6个页面，页号分别为0～5，页面大小为4K，页面变换表如下所示。表中状态位等于1和0分别表示页面在内存和不在内存。假设系统给进程P分配了4个存储块，进程P要访问的逻辑地址为十六进制5A29H，那么该地址经过变换后，其物理地址应为十六进制（　）；如果进程P要访问的页面4不在内存，那么应该淘汰页号为（　）的页面。

页号	页帧号	状态位	访问位	修改位
0	2	1	1	0
1	3	1	0	1
2	5	1	1	0
3	—	0	0	0
4	—	0	0	0
5	6	1	1	1

（1）A．1A29H　　B．3A29H　　C．5A29H　　D．6A29H

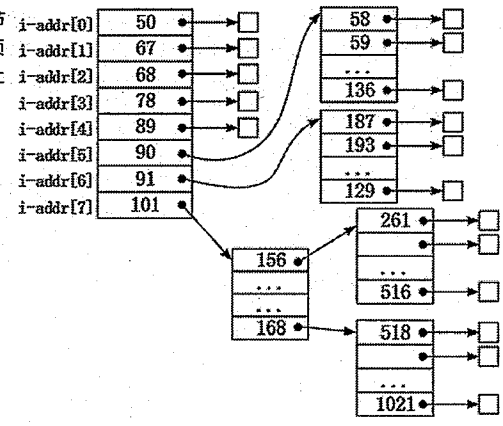
（2）A．0　　B．1　　C．2　　D．5

文件管理－索引文件结构



练习题

假设文件系统采用索引节点管理，且索引节点有8个地址项iaddr[0]～iaddr[7]，每个地址项大小为4字节，iaddr[0]～iaddr[4]采用直接地址索引，iaddr[5]和iaddr[6]采用一级间接地址索引，iaddr[7]采用二级间接地址索引。假设磁盘索引块和磁盘数据块大小均为1KB字节，文件File1的索引节点如下图所示。若用户访问文件File1中逻辑块号为5和261的信息，则对应的物理块号分别为__(1)__；101号物理块存放的是__(2)__。



（1）A．89和90　　B．89和136

C．58和187　　D．90和136

（2）A．File1的信息

B．直接地址索引表

C．一级地址索引表

D．二级地址索引表

希赛

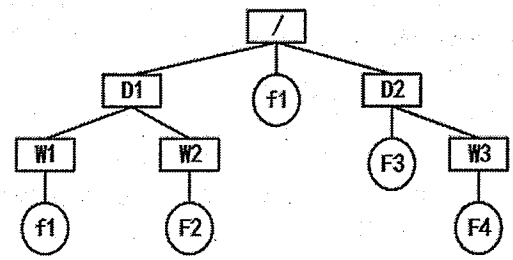
操作系统－文件和树型目录结构

◆**文件属性**

- R 只读文件属性
- A 存档属性
- S 系统文件
- H 隐藏文件

◆**文件名的组成**

- 驱动器号
- 路径
- 主文件名
- 扩展名



★绝对路径：是从盘符开始的路径。

★相对路径：是从当前目录开始的路径。

★若当前目录为：D1，要求F2路径，则：绝对路径：/D1/W2/F2，相对路径：W2/F2

希赛

文件管理－空闲存储空间的管理

- ✓ 空闲区表法（空闲文件目录）
- ✓ 空闲链表法
- ✓ 位示图法
- ✓ 成组链接法

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	0	0	1	1	1	0	0	1	0	1	1	1	0
1	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
2	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
3																
…																
15																

位示图

希赛

文件管理－试题讲解

某文件管理系统在磁盘上建立了位示图(bitmap)，记录磁盘的使用情况。若磁盘上的物理块依次编号为：0、1、2、…，系统中字长为32位，每一位对应文件存储器上的一个物理块，取值0和1分别表示空闲和占用，如下图所示。

31	30	…		3	2	1	0
0	1	…	1	0	0	0	1

假设将4195号物理块分配给某文件，那么该物理块的使用情况在位示图中的第__(1)__个字中描述；系统应该将__(2)__。

（1）A. 129　　B. 130　　C. 131　　D. 132

（2）A. 该字的第3位置“0”　　B. 该字的第3位置“1”

C. 该字的第4位置“0”　　D. 该字的第4位置“1”

希赛

文件管理－试题讲解

第1字	1	1	1	0	0	…	1	1
第2字	0	1	1		0	…	0	1
第3字	1	1	1	1	0	…	1	0
…	…							
第n字	0	0	0	1	1	…	0	0

字号=取整（（物理块号+1）/字长）+1
位号=求余（（物理块号+1）/字长）-1

位示图例

(4195+1)/32 = 131.125 -> 第132字。

131*32=4192 -> 0－4191。

第132字中：

第0位置 -> 4192　第1位置 -> 4193　第2位置 -> 4194　第3位置 -> 4195

希赛

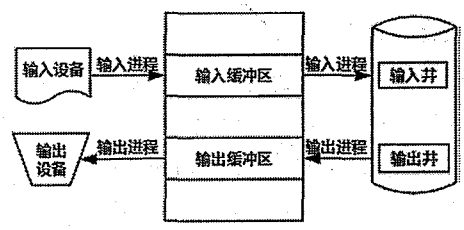
设备管理 – 数据传输控制方式

效率越来越高

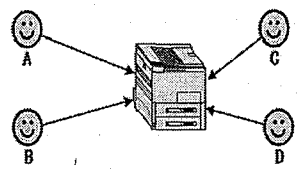
- ✓ **程序控制（查询）方式**：分为无条件传送和程序查询方式两种。方法简单，硬件开销小，但I/O能力不高，严重影响CPU的利用率。
- ✓ **程序中断方式**：与程序控制方式相比，中断方式因为CPU无需等待而提高了传输请求的响应速度。如：键盘
- ✓ **DMA方式**：DMA方式是为了在主存与外设之间实现高速、批量数据交换而设置的。DMA方式比程序控制方式与中断方式都高效。如：硬盘
- ✓ **通道方式**
- ✓ **I/O处理机**

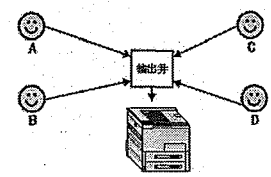
希赛

设备管理 – 虚设备与SPOOLING技术



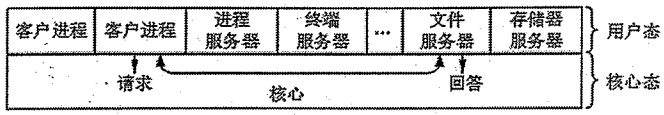
思考题：A，B，C，D共用一台打印机X，要进行资料打印时，很容易出现“打印机正在使用！”如何处理该问题？





希赛

微内核操作系统



	实质	优点	缺点
单体内核	将图形、设备驱动及文件系统等功能全部在内核中实现，运行在内核状态和同一地址空间。	减少进程间通信和状态切换的系统开销，获得较高的运行效率。	内核庞大，占用资源较多且不易剪裁。 系统的稳定性和安全性不好。
微内核	只实现基本功能，将图形系统、文件系统、设备驱动及通信功能放在内核之外。	内核精练，便于剪裁和移植。 系统服务程序运行在用户地址空间，系统的可靠性、稳定性和安全性较高。 可用于分布式系统	用户状态和内核状态需要频繁切换，从而导致系统效率不如单体内核。

希赛

嵌入式操作系统

1、嵌入式操作系统特点：

(1) 微型化、(2) 代码质量高、(3) 专业化、(4) 实时性强、(5) 可裁减、可配置。

2、实时嵌入式操作系统的内核服务有：异常和中断、计时器、I/O管理。

3、常见的嵌入式RTOS（实时操作系统）VxWorks、 RT-Linux、QNX、pSOS。

比较类型	VxWorks	RT-Linux
工作方式	操作系统与应用程序处于同一存储空间	操作系统与应用程序处于不同存储空间
多任务支持	支持多任务（线程）操作	支持多进程、多线程操作
实时性	实时系统	实时系统
安全性	任务间无隔离保护	支持进程间隔离保护
标准API	支持	支持

希赛

计算机网络

希赛

课程内容提要

- TCP/IP协议族
  - DHCP与DNS（★★★）
  - TCP与UDP（★★★）
- 网络规划与设计
  - 逻辑设计与物理设计（★★★★）
- 网络接入
  - 3G与4G标准（★★★）
- 网络存储
- 综合布线
- 物联网
- 云计算

希赛

DHCP协议

1、客户机/服务器模型

2、租约默认为8天

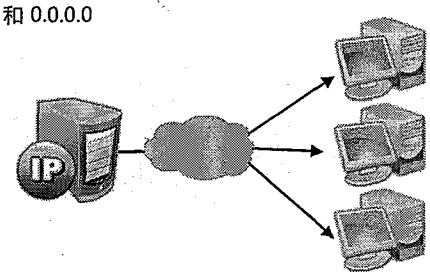
3、当租约过半时，客户机需要向DHCP服务器申请续租；

4、当租约超过87.5%时，如果仍然没有和当初提供IP的DHCP服务器联系上，则开始联系其他的DHCP服务器。

5、固定分配、动态分配和自动分配。

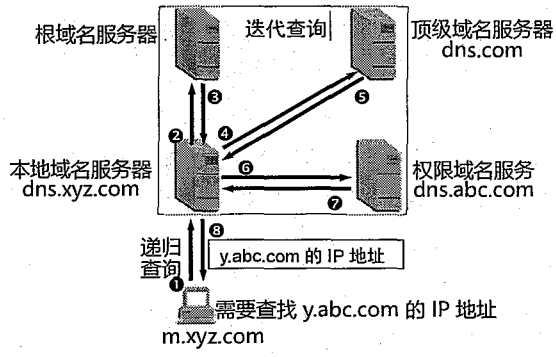
6、169.254.X.X 和 0.0.0.0

192.168.1.10
192.168.1.11
192.168.1.12
……



希赛

DNS协议

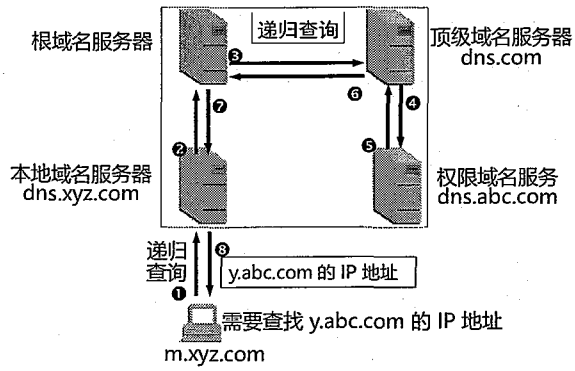


- 主机向本地域名服务器的查询采用递归查询。
- 本地域名服务器向根域名服务器的查询通常采用迭代查询。

- **递归查询**：服务器必需回答目标IP与域名的映射关系。
- **迭代查询**：服务器收到一次迭代查询回复一次结果，这个结果不一定是目标IP与域名的映射关系，也可以是其它DNS服务器的地址。

希赛

DNS协议

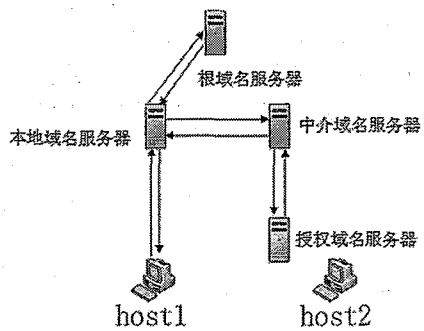


- 主机向本地域名服务器的查询采用递归查询。
- 本地域名服务器向根域名服务器的查询通常采用递代查询。
- 根域名服务器负担重，效率低，故较少采用。

希赛

DNS协议

主机host1对host2进行域名查询的过程如下图所示，下列说法中正确的是（ ）。



A．根域名服务器采用迭代查询，中介域名服务器采用递归查询

B．根域名服务器采用递归查询，中介域名服务器采用迭代查询

C．根域名服务器和中介域名服务器均采用迭代查询

D．根域名服务器和中介域名服务器均采用递归查询

希赛

网络规划与设计－需求分析

- ✓ 功能需求
- ✓ 通信需求
- ✓ 性能需求
- ✓ 可靠性需求
- ✓ 安全需求
- ✓ 运行与维护需求
- ✓ 管理需求

希赛

网络规划与设计－需求分析

◆ **网络规划原则**
- ✓ 实用性原则
- ✓ 开放性原则
- ✓ 先进性原则

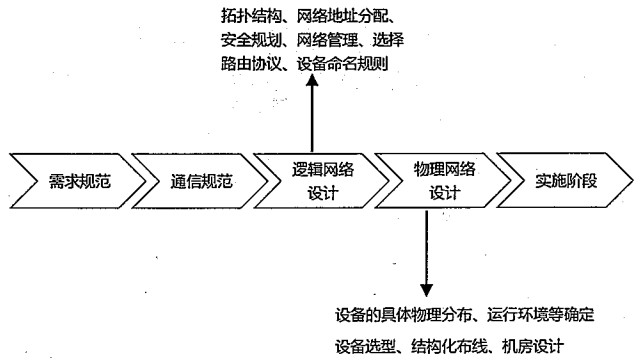
◆ **网络设计任务**
- ✓ 确定网络总体目标
- ✓ 确定总体设计原则
- ✓ 通信子网设计
- ✓ 资源子网设计
- ✓ 设备选型
- ✓ 网络操作系统与服务器资源设备
- ✓ 网络安全设计

◆ **网络实施原则**
- ✓ 可靠性原则
- ✓ 安全性原则
- ✓ 高效性原则
- ✓ 可扩展性原则

◆ **网络实施步骤**
- ✓ 工程实施计划
- ✓ 网络设备到货验收
- ✓ 设备安装
- ✓ 系统测试
- ✓ 系统试运行
- ✓ 用户培训
- ✓ 系统转换

希赛

网络规划与设计



希赛

网络规划与设计－逻辑网络设计

利用需求分析和现有网络体系分析的结果来设计逻辑网络结构，最后得到一份逻辑网络设计文档，输出内容包括以下几点：

- 逻辑网络设计图
- IP地址方案
- 安全方案
- 招聘和培训网络员工的具体说明
- 对软硬件、服务、员工和培训的费用初步估计

希赛

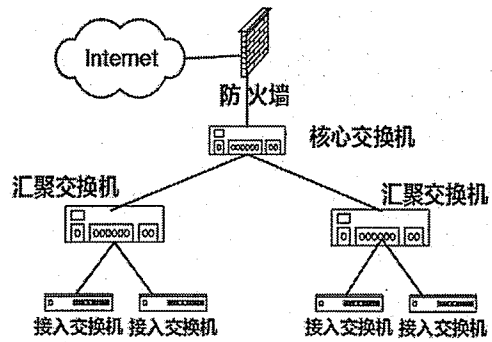
网络规划与设计 – 物理网络设计

物理网络设计是对逻辑网络设计的物理实现，通过对设备的具体物理分布、运行环境等确定，确保网络的物理连接符合逻辑连接的要求。输出如下内容：

- 网络物理结构图和布线方案
- 设备和部件的详细列表清单
- 软硬件和安装费用的估算
- 安装日程表，详细说明服务的时间以及期限
- 安装后的测试计划
- 用户的培训计划

希赛

网络规划与设计 – 分层设计



- 接入层 ：用户接入、计费管理、MAC地址认证、收集用户信息
- 汇聚层 ：网络访问策略控制、数据包处理、过滤、寻址
- 核心层 ：高速数据交换，常用冗余机制

希赛

网络接入技术

有线接入

- 公用交换电话网络（PSTN）
- 数字数据网（DDN）
- 综合业务数字网（ISDN）
- 非对称数字用户线路（ADSL）
- 同轴光纤技术（HFC）

无线接入

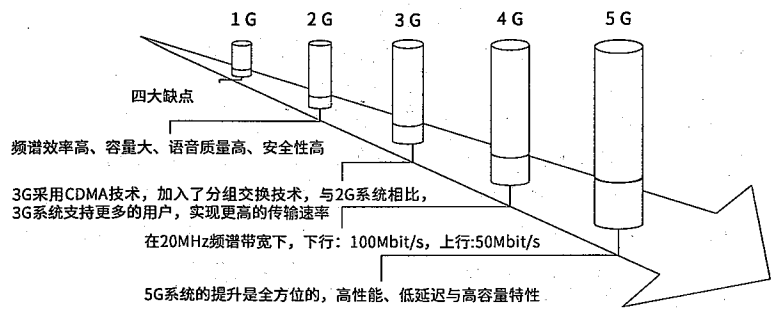
- IEEE 802.11(WiFi)
- IEEE 802.15(蓝牙Bluetooth)
- 红外(IrDA)
- WAPI

3G/4G/5G

- WCDMA
- CDMA2000
- TD-SCDMA
- TD-LTE和FDD-LTE
- WirelessMAN-Advanced(802.16m)(WiMAX)

希赛

网络接入技术



四大缺点

频谱效率高、容量大、语音质量高、安全性高

3G采用CDMA技术，加入了分组交换技术，与2G系统相比，3G系统支持更多的用户，实现更高的传输速率

在20MHz频谱带宽下，下行：100Mbit/s，上行:50Mbit/s

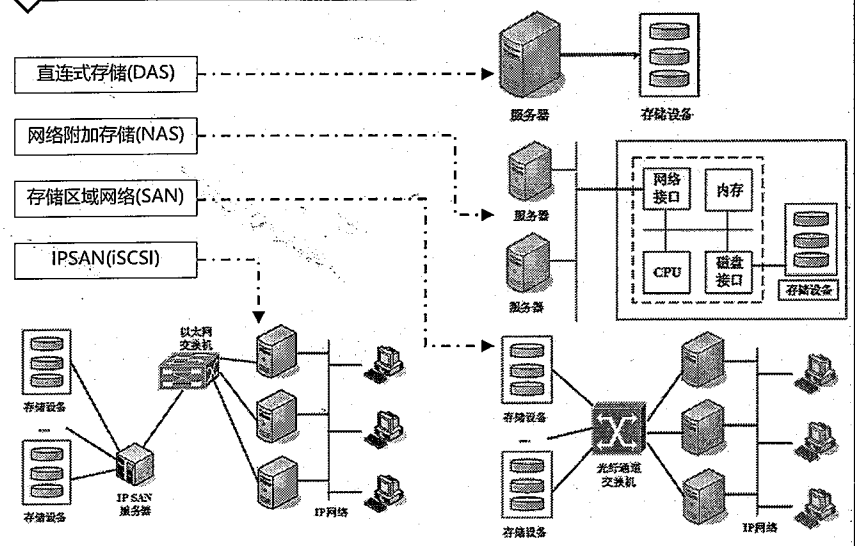
5G系统的提升是全方位的，高性能、低延迟与高容量特性

优点：体现在毫米波、小基站、MIMO，全双工及波束成型5大技术

目前试运行速率：1Gbit/s，最高速率：10Gbit/s

注：中兴曾试验速率高达50Gbit/s的5G网络

网络存储技术 – 分类

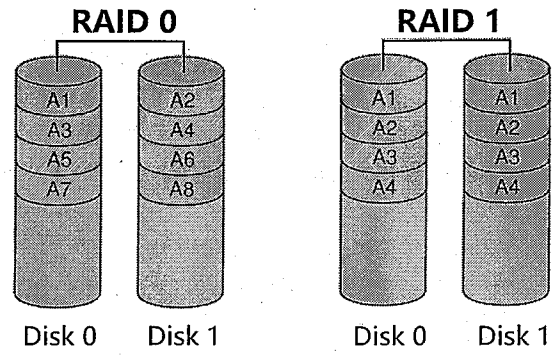


网络存储技术 – Raid

- Raid0(条块化)：性能最高，并行处理，无冗余，损坏无法恢复
- Raid1(镜像结构)：可用性，可修复性好，仅有50%利用率
- Raid0+1(Raid10)：Raid0与Raid1长处的结合，高效也可靠
- Raid3(奇偶校验并行传送) ：N+1模式，有固定的校验盘，坏一个盘可恢复
- Raid5(分布式奇偶校验的独立磁盘)：N+1模式，无固定的校验盘，坏一个盘可恢复
- Raid6(两种存储的奇偶校验)：N+2模式，无固定的校验盘，坏两个盘可恢复

希赛

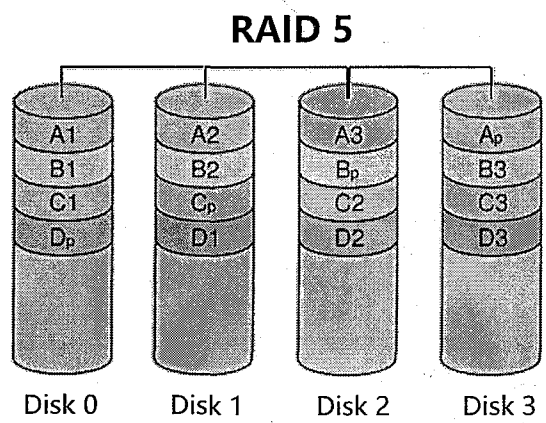
网络存储技术 – Raid



- RAID0磁盘利用率100%，访问速度最快。
- RAID1磁盘利用率为50%,具备纠错功能
- 现在企业采用RAID0与RAID1结合的方式比较多。

希赛

网络存储技术 – Raid



- RAID5磁盘利用率（n-1）/n，具备容错功能

希赛

IPv6

◆ **IPv6是设计用于替代现行版本IP协议（IPv4）的下一代IP协议。**

- （1）IPv6地址长度为128位，地址空间增大了2^{96}倍；
- （2）灵活的IP报文头部格式。使用一系列固定格式的扩展头部取代了IPv4中可变长度的选项字段。IPv6中选项部分的出现方式也有所变化，使路由器可以简单路过选项而不做任何处理，加快了报文处理速度；
- （3）IPv6简化了报文头部格式，字段只有8个，加快报文转发，提高了吞吐量；
- （4）提高安全性。身份认证和隐私权是IPv6的关键特性；
- （5）支持更多的服务类型；
- （6）允许协议继续演变，增加新的功能，使之适应未来技术的发展；

希赛

IPv6

- 单播地址(Unicast)：用于单个接口的标识符。
- 任播地址(Anycast)：泛播地址。一组接口的标识符，IPv4 广播地址。
- 组播地址(Multicast)：IPv6中的组播在功能上与IPv4中的组播类似。

多播：前缀为11111111

任播：前缀固定，其余位置为0

单播

- 可聚合全球单播地址：前缀001
- 本地单播地址：
  - 链路本地：前缀为1111111010
  - 站点本地：前缀为1111111011

希赛

IPv6

IPv6地址由8个16进制字段构成。例如：
2001:0db8:85a3:0000:1319:8a2e:0370:7344
IPv6地址的省写，上面的IP地址等价于：
2001:0db8:85a3::1319:8a2e:0370:7344。

遵守这些规则，如果因为省略而出现了两个以上的冒号，则可以压缩为一个，但这种零压缩在地址中只能出现一次。因此：
2001:0DB8:0000:0000:0000:0000:1428:57ab
2001:0DB8:0:0:0:0:1428:57ab
2001:0DB8::1428:57ab

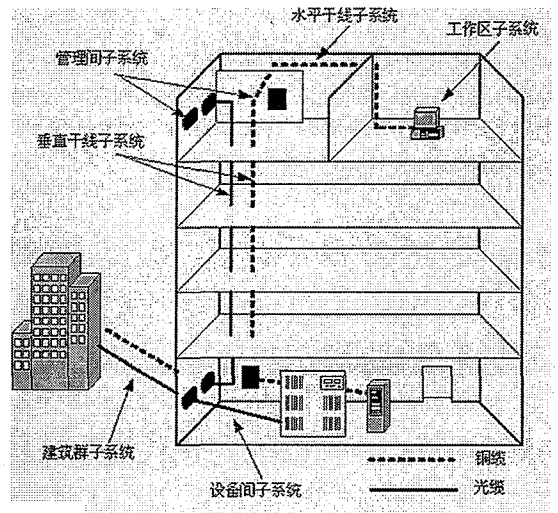
以上都是合法的地址，并且它们是等价的。同时前导的零可以省略，因此：2001:0DB8:02de::0e13等价于 2001:DB8:2de::e13。

◆ IPv4 / IPv6过渡技术有：

（1）双协议栈技术：双栈技术通过节点对IPv4和IPv6双协议栈的支持，从而支持两种业务的共存。
（2）隧道技术：隧道技术通过在IPv4网络中部署隧道，实现在IPv4网络上对IPv6业务的承载，保证业务的共存和过渡。隧道技术包括：6to4隧道；6over4隧道；ISATAP隧道。
（3）NAT-PT技术：NAT - PT使用网关设备连接IPv6和IPv4网络。当IPv4和IPv6节点互相访问时，NAT - PT网关实现两种协议的转换翻译和地址的映射。

希赛

综合布线系统

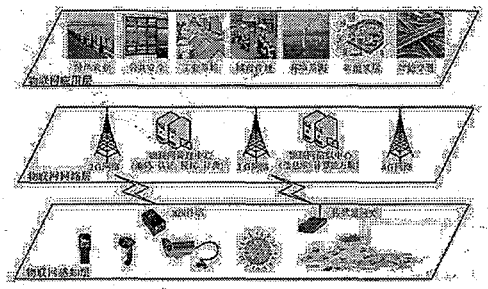


ANSI/EIA/TIA 568A 标准中综合布线系统的组成结构

希赛

物联网 – 物联网的概念与分层

物联网（The Internet of Things）是实现物物相连的互联网络，其内涵包含两个方面：第一，物联网的核心和基础仍然是互联网，是在互联网基础上延伸和扩展的网络；第二，其用户端延伸和扩展到了任何物体与物体之间，使其进行信息交换和通信



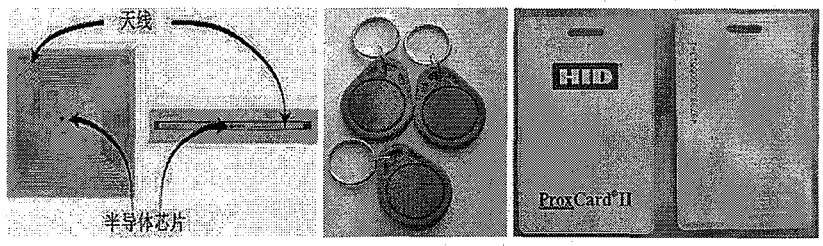
- 感知层：识别物体、采集信息。如：二维码、RFID、摄像头、传感器（温度、湿度）
- 网络层：传递信息和处理信息。通信网与互联网的融合网络、网络管理中心、信息中心和智能处理中心等
- 应用层：解决信息处理和人机交互的问题

希赛

物联网 – 物联网关键技术（RFID）

射频识别技术（Radio Frequency Identification，RFID），又称电子标签，是一种通信技术，可通过无线电讯号识别特定目标并读写相关数据，而无需识别系统与特定目标之间建立机械或光学接触。该技术是物联网的一项核心技术，很多物联网应用都离不开它

RFID的基本组成部分通常包括：标签、阅读器、天线。



希赛

物联网－物联网关键技术（二维码）

二维码 是用某种特定的几何图形按一定规律在平面（二维方向上）分布的黑白相间的图形记录数据符号信息的。在代码编制上巧妙地利用构成计算机内部逻辑基础的“0”、“1”比特流的概念，使用若干个与二进制相对应的几何形体来表示文字数值信息，通过图象输入设备或光电扫描设备自动识读以实现信息自动处理

二维条码中，常用的码制有：Data Matrix， Maxi Code， Aztec， QR Code， Vericode， PDF417， Ultracode， Code 49， Code 16K

- 若采用扩展的字母数字压缩格式，可容纳1850 个字符
- 若采用二进制/ ASCII格式，可容纳1108 个字节
- 若采用数字压缩格式，可容纳2710个数字

希赛

云计算的概念与特点

云计算是一种基于互联网的计算方式，通过这种方式，共享的软硬件资源和信息可以按需提供给计算机和其他设备。云其实是网络、互联网的一种比喻说法。云计算的核心思想，是将大量用网络连接的计算资源统一管理和调度，构成一个计算资源池向用户按需服务。提供资源的网络被称为“云”。狭义云计算指IT基础设施的交付和使用模式，指通过网络以按需、易扩展的方式获得所需资源；广义云计算指服务的交付和使用模式，指通过网络以按需、易扩展的方式获得所需服务。这种服务可以是IT和软件、互联网相关，也可是其他服务

特点:

- 集合了大量计算机，规模达到成千上万
- 多种软硬件技术相结合
- 对客户端设备的要求低
- 规模化效应

✓ 软件即服务（SaaS）
✓ 平台即服务（PaaS）
✓ 基础设施即服务（IaaS）

希赛

数据库系统

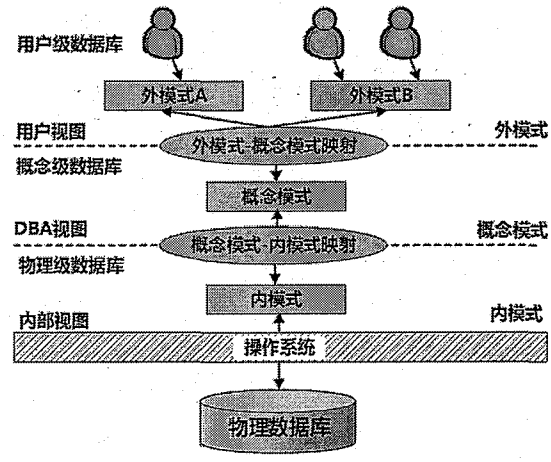
希赛

课程内容提要

- 数据库模式（★★★★）
- ER模型（★）
- 关系代数（★★★）
- 元组演算（★）
- 规范化理论（★★★★★）
- 并发控制（★）
- 数据库完整性约束（★★）
- 分布式数据库（★★★）
- 数据仓库与数据挖掘（★★★）

希赛

三级模式-两层映射



希赛

三级模式-视图

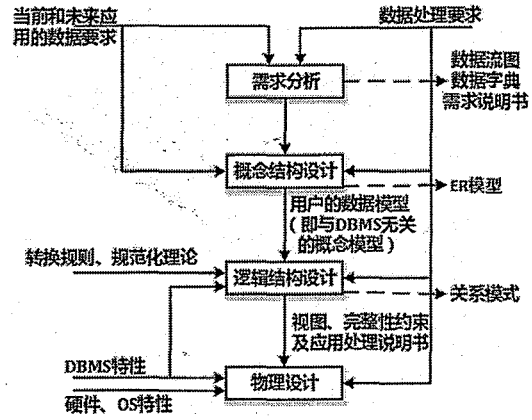
数据库视图：它一个虚拟表（逻辑上的表），其内容由查询定义（仅保存SQL查询语句）。同真实的表一样，视图包含一系列带有名称的列和行数据。但是，视图并没有真正存储这些数据，而是通过查询原始表动态生成所需要的数据。

视图的优点：
1、视图能简化用户操作
2、视图使用户能以多种角度看待同一数据
3、视图对重构数据库提供了一定程度的逻辑独立性
4、视图可以对机密数据提供安全保护

物化视图：它不是传统意义上虚拟视图，是实体化视图，其本身会存储数据。同时当原始表中的数据更新时，物化视图也会更新。

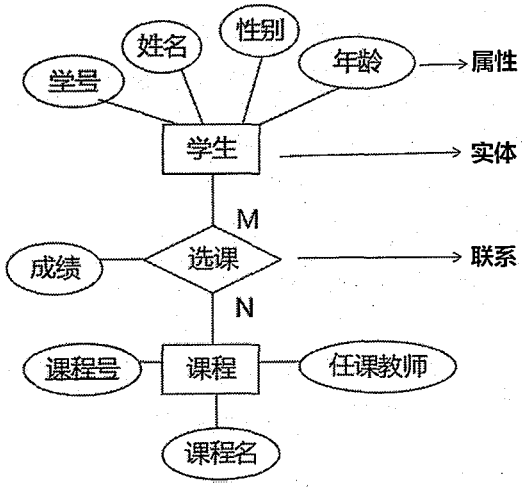
希赛

数据库设计过程



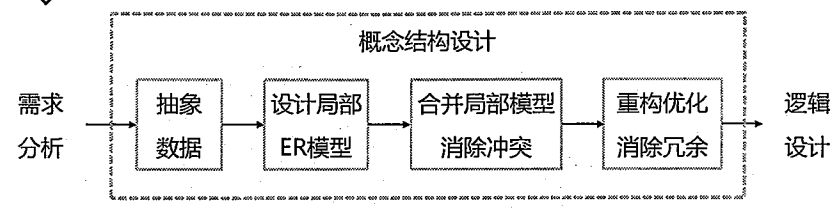
希赛

E-R模型



希赛

E-R模型



◆ **集成的方法：**

- 多个局部E-R图一次集成。
- 逐步集成，用累加的方式一次集成两个局部E-R。

◆ **集成产生的冲突及解决办法：**

- 属性冲突：包括属性域冲突和属性取值冲突。
- 命名冲突：包括同名异义和异名同义。
- 结构冲突：包括同一对象在不同应用中具有不同的抽象，以及同一实体在不同局部E-R图中所包含的属性个数和属性排列次序不完全相同。

希赛

E-R模型

★ 一个实体型转换为一个关系模式

★ 1：1联系可以转换为一个关系模式，也可以与任意一端实体的关系模式合并

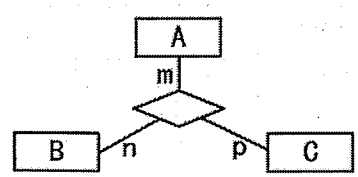
★ 1：n联系可以转换为一个关系模式，也可以与n端实体的关系模式合并

★ m：n联系必须单独转成一个关系模式

★ 三个以上实体间的一个多元联系

● 在数据库逻辑结构的设计中，将E-R模型转换为关系模型应遵循相关原则。对于三个不同实体集和它们之间的多对多联系$m:n:p$，最少可转换为___个关系模式。

A. 2　　B.3　　C. 4　　D. 5



希赛

关系代数

- ✓ 并
- ✓ 交
- ✓ 差
- ✓ 笛卡尔积
- ✓ 投影
- ✓ 选择
- ✓ 联接

关系S1

Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS

关系S2

Sno	Sname	Sdept
No0001	Mary	IS
No0008	Katter	IS
No0021	Tom	IS

S1 ∩ S2（交）

Sno	Sname	Sdept
No0001	Mary	IS

S1 U S2（并）

Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS
No0008	Katter	IS
No0021	Tom	IS

S1 - S2（差）

Sno	Sname	Sdept
No0003	Candy	IS
No0004	Jam	IS

希赛

关系代数

关系S1

Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS

关系S2

Sno	Sname	Sdept
No0001	Mary	IS
No0008	Katter	IS
No0021	Tom	IS

S1 × S2（笛卡尔积）

Sno	Sname	Sdept	Sno	Sname	Sdept
No0001	Mary	IS	No0001	Mary	IS
No0001	Mary	IS	No0008	Katter	IS
No0001	Mary	IS	No0021	Tom	IS
No0003	Candy	IS	No0001	Mary	IS
No0003	Candy	IS	No0008	Katter	IS
No0003	Candy	IS	No0021	Tom	IS
No0004	Jam	IS	No0001	Mary	IS
No0004	Jam	IS	No0008	Katter	IS
No0004	Jam	IS	No0021	Tom	IS

(投影)

Sno	Sname
No0001	Mary
No0003	Candy
No0004	Jam

(选择)

Sno	Sname	Sdept
No0003	Candy	IS

希赛

关系代数

关系S1		
Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS

关系S2	
Sno	Age
No0001	23
No0008	21
No0021	22

S1 ⋈ S2			
Sno	Sname	Sdept	Age
No0001	Mary	IS	23

希赛

规范化理论－价值与用途

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。

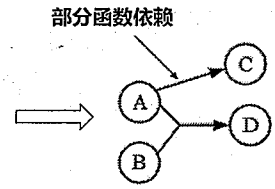
学号	姓名	系号	系名	系位置
S01	张三	D01	计算机系	1号楼
S02	李四	D01	计算机系	1号楼
S03	王五	D01	计算机系	1号楼
S04	赵六	D02	信息系	2号楼
...	...	...	...	...

规范化理论 – 函数依赖

设R(U)是属性U上的一个关系模式，X和Y是U的子集，r为R的任一关系，如果对于r中的任意两个元组u，v，只要有u[X]=v[X]，就有u[Y]=v[Y]，则称X函数决定Y，或称Y函数依赖于X，记为X→Y。

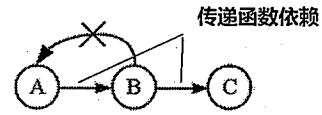
关系模式：R1（A，B，C，D）

依赖集：{AB -> D，A -> C}



关系模式：R2（A，B，C）

依赖集2：{A -> B，B -> C}



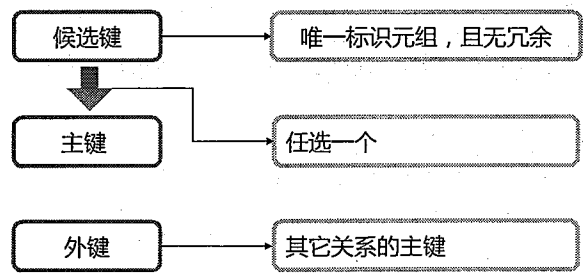
由：A -> B，B -> C，可得出A->C。
此为传递函数依赖

规范化理论 – 函数依赖

学号 -> 系号　　系号 -> 系名

希赛

规范化理论 - 键



希赛

规范化理论 - 求候选键

- 将关系模式的函数依赖关系用“有向图”的方式表示
- 找入度为0的属性，并以该属性集合为起点，尝试遍历有向图，若能正常遍历图中所有结点，则该属性集即为关系模式的候选键
- 若入度为0的属性集不能遍历图中所有结点，则需要尝试性的将一些中间结点（既有入度，也有出度的结点）并入入度为0的属性集中，直至该集合能遍历所有结点，集合为候选键

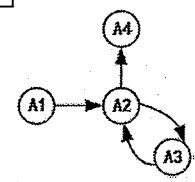
希赛

规范化理论 - 求候选键实例

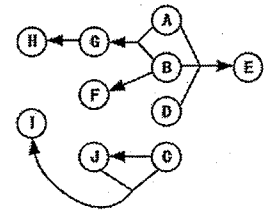
例1：给定关系R(A1，A2，A3，A4)上的函数依赖集F={A1→A2，A3→A2，A2→A3，A2→A4}，R的候选关键字为_____。

A．A1　　B．A1A3

C．A1A3A4　　D．A1A2A3



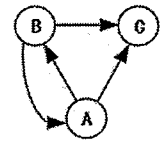
例2：关系模式P（A，B，C，D，E，F，G，H，I，J）满足下列函数依赖：FD={ABD→E，AB→G，B→F，C→J，CJ→I，G→H}，求候选码？



例3：关系R（A，B，C）满足下列函数依赖：F｛B→C，B→A，A→BC｝，关系R的候选关键字为 ___。

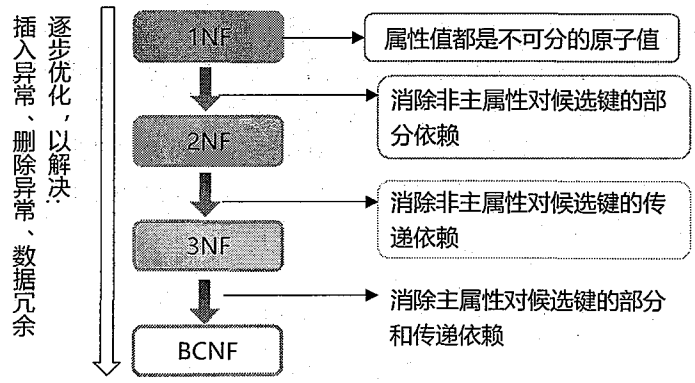
A．AB　　B．A和B

C．A和BC　　D．AC和AB



希赛

规范化理论 - 范式



希赛

规范化理论 - 范式

第一范式（1NF）：在关系模式R中，当且仅当所有域只包含原子值，即每个属性都是不可再分的数据项，则称关系模式R是第一范式。

例如：关系模式R（系名称，高级职称人数）是否满足1NF，如果不满足，应如何调整？

系名称	高级职称人数	
	教授	副教授
计算机系	6	10
电子系	3	5

希赛

规范化理论 – 第一范式 – 例题讲解

例：设有职工实体 Employee（职工号，姓名，性别，年龄，通信地址，家庭成员），其中通信地址记录了邮编、省、市、街道信息；家庭成员记录了职工的亲属的姓名。职工实体中的通信地址是一个（1）属性；为了将数据库模式设计的更合理，对于家庭成员属性（2）。

(1)A. 简单　B. 复合　C. 多值　D. 派生

(2)A. 可以不作任何处理直接记录亲属的姓名

B. 只允许记录一个亲属的姓名

C. 需要对职工实体设置若干个亲属姓名字段

D. 应该将职工的亲属的姓名加上职工号设计成为一个独立的实体

希赛

规范化理论－第二范式

第二范式（2NF）：当且仅当实体E是第一范式（1NF），且每一个非主属性完全依赖主键（不存在部分依赖）时，则称实体E是第二范式。

思考题：请思考该关系模式会存在哪些问题（从数据冗余、更新异常、插入异常、删除异常这几个方面来考虑），解决方案是什么？

学号	课程号	成绩	学分
S01	C01	75	4
S02	C01	92	4
S03	C01	87	4
S04	C01	55	4
S01	C02	87	2
S02	C02	95	2
S01	C03	94	5
…	…	…	…

希赛

规范化理论－第三范式

第三范式（3NF）：当且仅当实体E是第二范式（2NF），且E中没有非主属性传递依赖于码时，则称实体E是第三范式。

思考题：请思考该关系模式会存在哪些问题（从数据冗余、更新异常、插入异常、删除异常这几个方面来考虑），解决方案是什么？

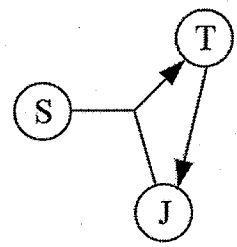
学号	姓名	系号	系名	系位置
S01	张三	D01	计算机系	1号楼
S02	李四	D01	计算机系	1号楼
S03	王五	D01	计算机系	1号楼
S04	赵六	D02	信息系	2号楼
…	…	…	…	…

希赛

规范化理论 – BC范式

BC范式（BCNF）：设R是一个关系模式，F是它的依赖集，R属于BCNF当且仅当其F中每个依赖的决定因素必定包含R的某个候选码。

例：关系模式STJ（S，T，J）中，S表示学生，T表示老师，J表示课程。每一老师只教一门课程。每门课程有若干老师，某一学生选定某门课，就对应一个固定老师。



希赛

规范化理论 – 第一范式 – 例题讲解

某公司的部门（部门号，部门名，负责人，电话）、商品（商品号，商品名称，单价，库存量）和职工（职工号, 姓名, 住址）三个实体之间的关系如表1、表2和表3所示。假设每个部门有一位负责人和一部电话，但有若干名员工；每种商品只能由一个部门负责销售。部门关系不属于第三范式的原因是__(1)__。如果用户要求得到表4所示的结果，需要__(2)__，并增加关系模式__(3)__。

A. 没有消除非主属性对码的部分函数依赖
如：部门名→负责人
B. 没有消除非主属性对码的部分函数依赖
如：负责人→电话
C. 只消除了非主属性对码的部分函数依赖而未消除传递函数依赖
D. 没有消除非主属性对码的部分函数依赖和传递函数依赖

A. 修改表1的结构，在表1中增加一个职工号
B. 修改表2的结构，在表2中增加一个职工号
C. 修改表2的结构，在表2中增加一个部门号
D. 修改表3的结构，在表3中增加一个部门号

A. 销售（职工号，商品号，日期，数量）
B. 销售（职工号，商品名称，商品号，数量）
C. 销售（职工号，部门号，日期，数量）
D. 销售（职工号，部门号，商品号，日期）

1

部门号	部门名	负责人	电话
001	家电部	E002	1001
002	百货部	E026	1002
003	食品部	E030	1003

2

商品号	商品名称	单价	库存量
30023	微机	4800	27
30024	打印机	1650	7
...	...	...	...
30101	毛巾	10	106
30102	牙刷	38	288
...	...	...	...

3

职工号	姓名	住址
E001	王军	南京路
E002	李晓斌	淮海路
E021	杨烨	江西路
E026	田波	西藏路
E028	李晓斌	西藏路
E029	刘丽华	淮海路
E030	李彬彬	唐山路
E031	胡慧芬	昆明路
...	...	...

4

职工号	姓名	部门名	月销售额
E001	王军	家电部	528900
E002	李晓斌	家电部	368000
E021	杨烨	百货部	12500
E028	李晓斌	百货部	82500
E031	胡慧芬	食品部	282608
...	...	...	...

希赛

规范化理论 – 无损分解

★ 保持函数依赖分解

设数据库模式ρ={R1，R2，…，Rk}是关系模式R的一个分解，F是R上的函数依赖集，ρ中每个模式Ri上的FD集是Fi。如果{F1，F2，…，Fk}与F是等价的（即相互逻辑蕴涵），那么称分解ρ保持FD。

★ 无损分解

什么是有损，什么又是无损？

有损：不能还原。　　无损：可以还原。

无损联接分解：指将一个关系模式分解成若干个关系模式后，通过自然联接和投影等运算仍能还原到原来的关系模式

希赛

规范化理论 – 无损分解

★ 保持函数依赖分解

例1：

有关系模式R（A，B，C），F={A→B，B→C}，将其拆分为：R1（A，B），R2（B，C），是否保持函数依赖。

例2：

有关系模式R（A，B，C），F={A→B，B→C，A→C}，将其拆分为：R1（A，B），R2（B，C），是否保持函数依赖。

希赛

规范化理论－示例

例1：

有关系模式R（A，B，C），F=｛A→B，B→C，A->C｝，将其拆分为：R1（A，B），R2（A，C），是否保持函数依赖。

例2：

有关系模式R（A，B，C，D，E），F=｛A→B，D→E｝，将其拆分为：R1（A，B，C），R2（D，E），是否保持函数依赖。

希赛

规范化理论－无损分解－例题讲解

有关系模式：成绩（学号，姓名，课程号，课程名，分数）

函数依赖：学号→姓名，课程号→课程名，（学号，课程号）→分数

若将其分解为：

成绩（学号，课程号，分数）

学生（学号，姓名）

课程（课程号，课程名）

请思考该分解是否为无损分解？

由于有：学号→姓名，所以：

成绩（学号，课程号，分数，姓名）

由于有：课程号→课程名，所以：

成绩（学号，课程号，分数，姓名，课程名）

希赛

规范化理论－无损分解－例题讲解

将一个具有函数依赖：学号→姓名，课程号→课程名，(学号，课程号)→分数的关系模式：成绩(学号，姓名，课程号，课程名，分数)，分解为：成绩(学号，课程号，分数)；学生(学号，姓名)；课程(课程号，课程名)。

初始表如下：

	学号	姓名	课程号	课程名	分数
成绩	√		√		√
学生	√	√			
课程			√	√	

希赛

规范化理论－无损分解－例题讲解

根据学号→姓名，对上表进行处理，将×改成符号√；然后考虑课程号→课程名，将×改为√，得下表：

	学号	姓名	课程号	课程名	分数
成绩	√	√	√	√	√
学生	√	√	×	×	×
课程	×	×	√	√	×

从上图中可以看出，第1行已全部为√，因此本次R分解是无损联接分解。

希赛

规范化理论－无损分解

定理：如果R的分解为$\rho=\{R_1, R_2\}$，F为R所满足的函数依赖集合，分解ρ具有无损联接性的充分必要条件是：

$R_1 \cap R_2 \rightarrow (R_1 - R_2)$

或 $R_1 \cap R_2 \rightarrow (R_2 - R_1)$

其中，$R_1 \cap R_2$表示模式的交，为R1与R2中公共属性组成，$R_1 - R_2$或$R_2 - R_1$表示模式的差集，$R_1 - R_2$表示R_1中去除R_1和R_2的公共属性所组成。当模式R分解成两个关系模式R_1和R_2时，如果R_1与R_2的公共属性能函数决定R_1中或R_2中的其它属性，这样的分解就具有无损联接性。

例：设R＝ABC，F＝{A→B}，则分解$\rho_1=\{R_1(AB), R_2(AC)\}$与分解$\rho_2=\{R_1(AB), R_3(BC)\}$是否都为无损分解？

$R_1 \cap R_2 = A$　　$R_1 - R_2 = B$　　$R_2 - R_1 = C$

A →B 或 A →C。

$R_1 \cap R_3 = B$　　$R_1 - R_3 = A$　　$R_3 - R_1 = C$

B →A 或 B →C。

希赛

规范化理论－无损分解

设关系模式R(U，F)，其中R上的属性集U={A，B，C，D，E}，R上的函数依赖集F={A→B，DE→B，CB→E，E→A，B→D}。（ ）为关系R的候选关键字。分解（ ）是无损连接，并保持函数依赖的。

A．AB　　B．DE　　C．CE　　D．DB

A．p={R1(AC)，R2(ED)，R3(B)}

B．p={R1(AC)，R2(E)，R3(DB)}

C．p={R1(AC)，R2(ED)，R3(AB)}

D．p={R1(ABC)，R2(ED)，R3(ACE)}

希赛

规范化理论－无损分解

关系模式R <U，F >来说有以下的推理规则：

A1.自反律（Reflexivity）：若Y⊆X⊆U，则X →Y成立。

A2.增广律（Augmentation）：若Z⊆U且X→Y，则XZ→YZ成立。

A3.传递律（Transitivity）：若X→Y且Y→Z，则X→Z成立。

根据A1，A2，A3这三条推理规则可以得到下面三条推理规则：

合并规则：由X→Y，X→Z，有X→YZ。（A2， A3）

伪传递规则：由X→Y，WY→Z，有XW→Z。（A2， A3）

分解规则：由X→Y及 Z ⊆ Y，有X→Z。（A1， A3）

希赛

规范化理论－无损分解

根据A1，A2，A3这三条推理规则可以得到下面三条推理规则：

合并规则：由X→Y，X→Z，有X→YZ。（A2， A3）

证明：$\left.\begin{array}{l} X\to Y \xrightarrow{A2} X\to XY \\ X\to Z \xrightarrow{A2} XY\to YZ \end{array}\right\} \xrightarrow{A3} X\to YZ$

伪传递规则：由X→Y，WY→Z，有XW→Z。（A2， A3）

证明：$\left.\begin{array}{l} X\to Y \xrightarrow{A2} WX\to WY \\ WY\to Z \end{array}\right\} \xrightarrow{A3} WX\to Z$

分解规则：由X→Y及 Z ⊆ Y，有X→Z。（A1， A3）

证明：$\left.\begin{array}{l} Z\subseteq Y \xrightarrow{A1} Y\to Z \\ X\to Y \end{array}\right\} \xrightarrow{A3} X\to Z$

希赛

反规范化

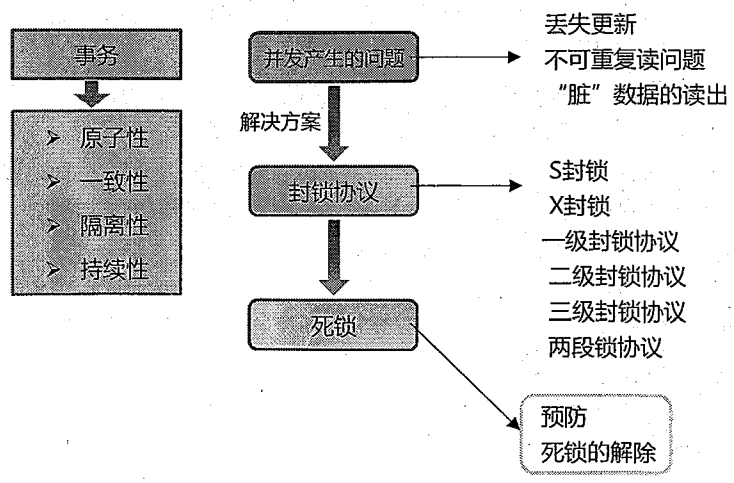
✓ 由于规范化会使表不断的拆分，从而导致数据表过多。这样虽然减少了数据冗余，提高了增、删、改的速度 但会增加查询的工作量。系统需要进行多次连接，才能进行查询操作，使得系统的效率大大的下降

✓ **技术手段**

- 增加派生性冗余列
- 增加冗余列
- 重新组表
- 分割表

希赛

并发控制 – 基本概念



希赛

并发控制 – 存在的问题示例

◆ 丢失更新

T1	T2
①读A=10	
②	读A=10
③A=A-5写回	
④	A=A-8 写回

◆ 不可重复读

T1	T2
①读A=20 读B=30 求和=50	
②	读A=20 A←A+50 写A=70
③读A=70 读B=30 求和=100 (验算不对)	

◆ 读“脏”数据

T1	T2
①读A=20 A←A+50 写回70	
②	读A=70
③ROLLBACK A恢复为20	

希赛

并发控制 – 封锁协议

- ✓ 一级封锁协议。事务T在修改数据R之前必须先对其加X锁，直到事务结束才释放。可防止丢失修改
- ✓ 二级封锁协议。一级封锁协议加上事务T在读取数据R之前先对其加S锁，读完后即可释放S锁。可防止丢失修改，还可防止读“脏”数据
- ✓ 三级封锁协议。一级封锁协议加上事务T在读取数据R之前先对其加S锁，直到事务结束才释放。可防止丢失修改、防止读“脏”数据与防止数据重复读
- ✓ 两段锁协议。可串行化的。可能发生死锁

希赛

并发控制 – 丢失更新加锁

T1	T2
① 对A加写锁	
②	对A加写锁
③ 读A=10	等待
④ A=A-5写回	等待
⑤ 释放对A的写锁	等待
⑥	读A=10
⑦	A=A-8 写回
⑧	释放对A的写锁

希赛

并发控制 –读“脏”数据加锁

T1	T2
① 对A加写锁	
② 读A=20	
③ A←A+50	
④ 写回70	
⑤	对A加读锁
⑥ ROLLBACK	等待
⑦ A恢复为20	等待
⑧	读A=20
	释放对A的读锁

希赛

并发控制－不可重复读加锁

T1	T2
① 对A与B加S锁（读锁） 读A=20 读B=30 求和=50 ②	对A加X锁（写锁） 注：由于A已加了读锁，所以等待 等待 等待 等待 等待
③读A=20 读B=30 求和=50 释放对A和B的读锁	读A=20 A←A+50 写A=70 释放对A的写锁

希赛

数据库完整性约束

- 实体完整性约束
- 参照完整性约束
- 用户自定义完整性约束

✓ 触发器

希赛

数据库安全性

措施	说明
用户标识和鉴定	最外层的安全保护措施，可以使用用户帐户、口令及随机数检验等方式
存取控制	对用户进行授权，包括操作类型（如查找、插入、删除、修改等动作）和数据对象（主要是数据范围）的权限。
密码存储和传输	对远程终端信息用密码传输
视图的保护	对视图进行授权
审计	使用一个专用文件或数据库，自动将用户对数据库的所有操作记录下来

希赛

数据备份

- 冷备份也称为静态备份，是将数据库正常关闭，在停止状态下，将数据库的文件全部备份（复制）下来。
- 热备份也称为动态备份，是利用备份软件，在数据库正常运行的状态下，将数据库中的数据文件备份出来。

优缺点 / 备份方式	优点	缺点
冷备份	非常快速的备份方法(只需复制文件)；容易归档(简单复制即可)；容易恢复到某个时间点上(只需将文件再复制回去)；能与归档方法相结合，做数据库“最佳状态”的恢复；低度维护，高度安全	单独使用时，只能提供到某一时间点上的恢复；在实施备份的全过程中，数据库必须要作备份而不能做其他工作；若磁盘空间有限只能复制到磁带等其他外部存储设备上，速度会很慢；不能按表或按用户恢复
热备份	可在表空间或数据库文件级备份，备份的时间短；备份时数据库仍可使用；可达到秒级恢复(恢复到某一时间点上)；可对几乎所有数据库实体做恢复；恢复是快速的	不能出错，否则后果严重；若热备份不成功所得结果不可用于时间点的恢复；因难于维护，所以要特别小心，不允许“以失败告终”

希赛

数据备份

- ✓ 完全备份：备份所有数据
- ✓ 差量备份：仅备份上一次完全备份之后变化的数据
- ✓ 增量备份：备份上一次备份之后变化的数据

➢ 日志文件 ：事务日志是针对数据库改变所做的记录，它可以记录针对数据库的任何操作，并将记录结果保存在独立的文件中

周日	周一	周二	周三	周四	周五	周六
全备	增备	增备	差备	增备	差备	增备

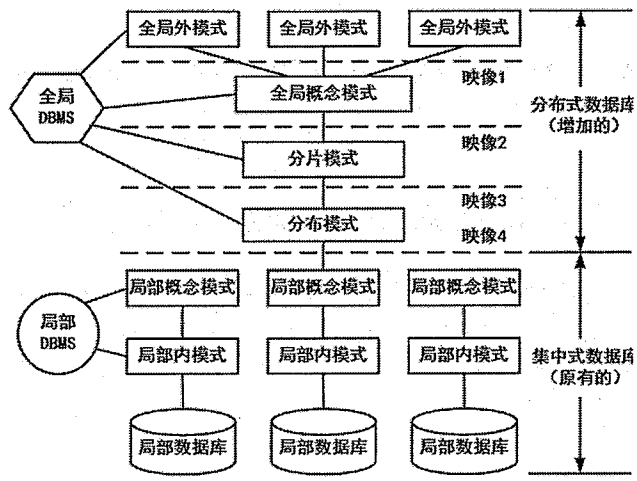
希赛

数据库故障与恢复

故障关系	故障原因	解决方法
事务本身的可预期故障	本身逻辑	在程序中预先设置Rollback语句
事务本身的不可预期故障	算术溢出、违反存储保护	由DBMS的恢复子系统通过日志，撤消事务对数据库的修改，回退到事务初始状态
系统故障	系统停止运转	通常使用检查点法
介质故障	外存被破坏	一般使用日志重做业务

希赛

分布式数据库－体系结构



希赛

分布式数据库－各种概念

➢ **分布透明性**

- 分片透明性 →
  - 水平分片
  - 垂直分片
  - 混合分片
- 位置透明性
- 局部数据模型透明性

➢ **分布式数据库管理系统–组成**

- LDBMS
- GDBMS
- 全局数据字典
- 通信管理（CM）

➢ **分布式数据库管理系统–结构**

- 全局控制集中的DDBMS
- 全局控制分散的DDBMS
- 全局控制部分分散的DDBMS

希赛

数据表分区

表分区：就是将一个数据量比较大的表，用某种方法把数据从物理上分成若干个小表来存储，从逻辑来看还是一个大表。

分区与分表的区别与联系

（1）两者都针对数据表，将数据做到分布式，提高数据检索的效率，降低数据库的频繁I/O压力值。

（2）分表是真正的生成数据表，是将一张大数据量的表分成多个小表实现数据均衡。

（3）分区并不是生成新的数据表，而是将表的数据均衡分摊到不同的硬盘，系统或是不同服务器存储介质中，实际上还是一张表。

希赛

数据表分区

分区的优点

1、相对于单个文件系统或是硬盘，分区可以存储更多的数据。

2、数据管理比较方便，比如要清理或废弃某年的数据，就可以直接删除该日期的分区数据即可。

3、精准定位分区查询数据，不需要全表扫描查询，大大提高数据检索效率。

4、可跨多个分区磁盘查询，来提高查询的吞吐量。

5、在涉及聚合函数查询时，可以很容易进行数据的合并。

希赛

数据表分区

分区的策略

范围分区（RANGE）：就是根据数据库表中某一字段的值的范围来划分分区。如：年份小于2016的分成一个区，其它分成另一个区。

散列分区（HASH）：散列分区是根据字段的hash值进行均匀分布，尽可能的实现各分区所散列的数据相等。

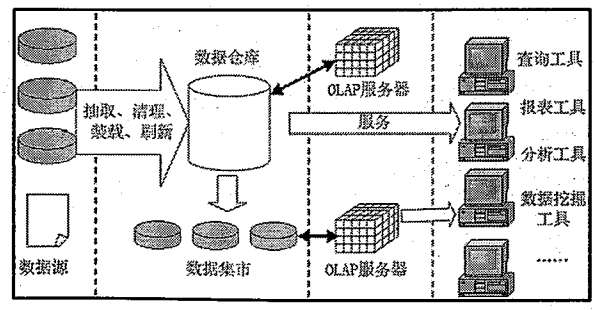
列表分区（LIST）：列表分区明确指定了根据某字段的某个具体值进行分区，而不是像范围分区那样根据字段的值范围来划分的。如：长沙、武汉分成一个区，北京一个区。

希赛

数据仓库与数据挖掘

数据仓库的特点：

- 面向主题：数据按主题组织。
- 集成的：消除了源数据中的不一致性，提供整个企业的一致性全局信息。
- 相对稳定的(非易失的)：主要进行查询操作，只有少量的修改和删除操作（或是不删除）。
- 反映历史变化(随着时间变化)：记录了企业从过去某一时刻到当前各个阶段的信息，可对发展历程和未来趋势做定量分析和预测。



希赛

联邦数据库

✓ 联邦数据库系统（FDBS）是一个彼此协作却又相互独立的成员数据库（CDBS）的集合，它将成员数据库系统按不同程度进行集成，对该系统整体提供控制和协同操作的软件叫做联邦数据库管理系统（FDBMS）

➢ **联邦数据库特征**

- 分布性
- 异构性
- 自治性
- 透明性

➢ **联邦数据库分类**

- 紧耦合
- 松耦合

希赛

NoSQL

✓ NoSQL（Not-only SQL）：泛指非关系型的数据库。

关系数据库的缺陷："高并发读写的性能低、支撑容量有限、数据库的可扩展性和可用性低、建设和运维成本高"。

	关系数据库模式	NoSQL模式
并发支持	支持并发、效率低	并发性能高
存储与查询	关系表方式存储、SQL查询	海量数据存储、查询效率高
扩展方式	向上扩展	向外扩展
索引方式	B树、哈希等	键值索引
应用领域	面向通用领域	特定应用领域

希赛

NoSQL

分类	典型应用场景	数据模型	优点	缺点	Examples举例
键值（key-value）	内容缓存，主要用于处理大量数据的高访问负载，也用于一些日志系统等等。	Key 指向 Value 的键值对，通常用hash table来实现	查找速度快	数据无结构化，通常只被当作字符串或者二进制数据	Redis, Tokyo Cabinet/ Tyrant, Voldemort, Oracle BDB
列存储数据库	分布式的文件系统	以列簇式存储，将同一列数据存在一起	查找速度快，可扩展性强，更容易进行分布式扩展	功能相对局限	HBase, Cassandra, Riak
文档型数据库	Web应用（与Key-Value类似，Value是结构化的，不同的是数据库能够了解Value的内容）	Key-Value对应的键值对，Value为结构化数据	数据结构要求不严格，表结构可变，不需要像关系型数据库一样需要预先定义表结构	查询性能不高，而且缺乏统一的查询语法。	CouchDB, MongoDb
图形数据库 (Graph)	社交网络，推荐系统等。专注于构建关系图谱	图结构	利用图结构相关算法。比如最短路径寻址，N度关系查找等	很多时候需要对整个图做计算才能得出需要的信息，而且这种结构不太好做分布式的集群方案。	Neo4J, InfoGrid, Infinite Graph

希赛

内存数据库

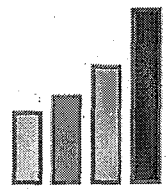
★ 内存数据库抛弃了磁盘数据管理的传统方式，基于全部数据都在内存中重新设计了体系结构，并且在数据缓存、快速算法、并行操作方面也进行了相应的改进，所以数据处理速度比传统数据库的数据处理速度要快很多，一般都在10倍以上。内存数据库的最大特点是其“主拷贝”或“工作版本”常驻内存，即活动事务只与实时内存数据库的内存拷贝打交道。

★ 常见的内存数据库包括：Redis、eXtremeDB、TT、FastDB、SQLite、Microsoft SQL Server Compact、MySQL的MEMORY存储引擎等。

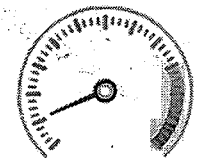
希赛

何谓大数据

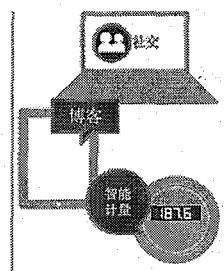
4V ?



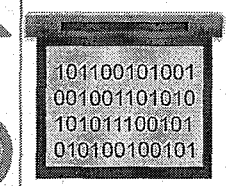
数据量
Volume



速度
Velocity



多样性
Variety



值
Value

希赛

大数据

比较维度	传统数据	大数据
数据量	GB或TB级	PB级或以上
数据分析需求	现有数据的分析与检测	深度分析（关联分析、回归分析）
硬件平台	高端服务器	集群平台

✓ **大数据处理系统应该具有的重要特征**

- 高度可扩展性
- 高性能
- 高度容错
- 支持异构环境
- 较短的分析延迟
- 易用且开放的接口
- 较低成本
- 向下兼容性

案例分析

阅读以下关于软件系统数据建模的说明，在答题纸上回答问题1至问题3。

【说明】

某软件公司成立项目组为某高校开发一套教职工信息管理系统。与教职工信息相关的数据需求和处理需求如下：

（1）数据需求：在教职工信息中能够存储学校所有在职的教工和职工信息，包括姓名、所属部门、出生年月、工资编号、工资额和缴税信息；部门信息中包括部门编号、部门名称、部门人数和办公地点信息。

（2）处理需求：能够根据编制内或外聘教职工的工资编号分别查询其相关信息；每个月的月底统一核发工资，要求系统能够以最快速度查询出教工或者职工所在部门名称、实发工资金额；由于学校人员相对稳定，所以数据变化及维护工作量很少。

项目组王工和李工针对上述应用需求分别给出了所设计的数据模型（如图2-1和图2-2所示）。王工遵循数据库设计过程，按照第三范式对数据进行优化和调整，所设计的数据模型简单且基本没有数据冗余；而李工设计的数据模型中存在大量数据冗余。

案例分析

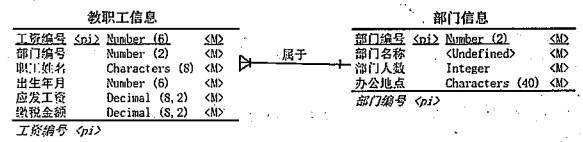


图2-1 王工设计的数据模型

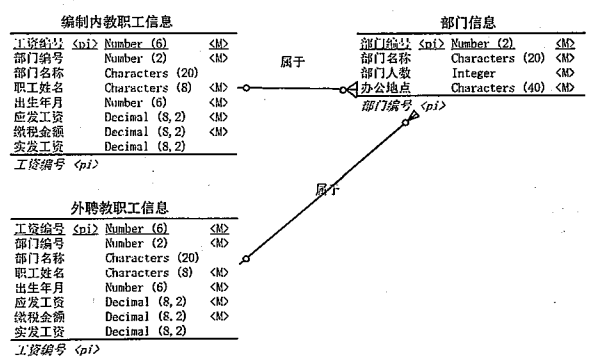


图2-2 李工设计的数据模型

希赛

案例分析

项目组经过分析和讨论，特别是针对数据处理中对数据访问效率的需求，最终选择了李工给出的数据模型设计方案。

【问题1】（9分）

请用300字以内的文字，说明什么是数据库建模中的反规范化技术，指出采用反规范化技术能获得哪些益处，可能带来哪些问题。

【问题2】（8分）

请简要叙述常见的反规范化技术有哪些。

【问题3】（8分）

请分析李工是如何应用反规范化技术来满足教职工信息管理需求的。

希赛

案例分析

【问题1】

规范化设计后，数据库设计者希望牺牲部分规范化来提高性能，这种从规范化设计的回退方法称为反规范化技术。

采用反规范化技术的益处：降低连接操作的需求、降低外码和索引的数目，还可能减少表的数目，能够提高查询效率。

可能带来的问题：数据的重复存储，浪费了磁盘空间；可能出现数据的完整性问题，为了保障数据的一致性，增加了数据维护的复杂性，会降低修改速度。

【问题2】

（1）增加冗余列：在多个表中保留相同的列，通过增加数据冗余减少或避免查询时的连接操作。

（2）增加派生列：在表中增加可以由本表或其它表中数据计算生成的列，减少查询时的连接操作并避免计算或使用集合函数。

（3）重新组表：如果许多用户需要查看两个表连接出来的结果数据，则把这两个表重新组成一个表来减少连接而提高性能。

（4）水平分割表：根据一列或多列数据的值，把数据放到多个独立的表中，主要用于表数据规模很大、表中数据相对独立或数据需要存放到多个介质上时使用。

（5）垂直分割表：对表进行分割，将主键与部分列放到一个表中，主键与其它列放到另一个表中，在查询时减少I/O次数。

希赛

案例分析

【问题3】

在教职工信息管理系统的需求中，能够根据编制内或外聘教职工的工资编号分别查询其相关信息，数据查询要求有很高的处理效率。李工所设计的数据模型中采用了三种反规范化技术：

（1）增加冗余列：增加“部门名称”列，消除了数据查询中“教职工信息”表和“部门信息”表之间的连接；

（2）增加派生列：增加“实发工资”列，消除了实发工资的计算过程；

（3）水平分割表：将教职工信息表分割为“编制内教职工信息”表和“外聘教职工信息”表，减少了数据查询的范围。

希赛

数学与经济管理

希赛

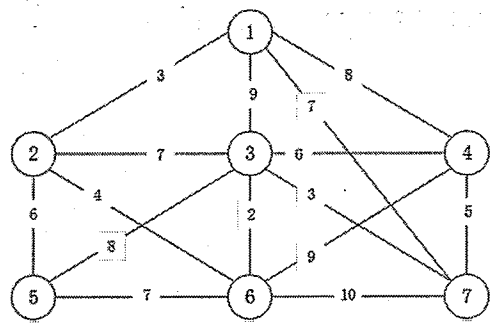
课程内容提要

- **图论应用**
  - 最小生成树（★）
  - 最短路径（★）
  - 网络与最大流量（★★）
- **运筹方法**
  - 线性规划（★）
  - 动态规划（★★★）
  - 预测与决策（★）
- **数学建模（★★）**

希赛

图论应用－最小生成树

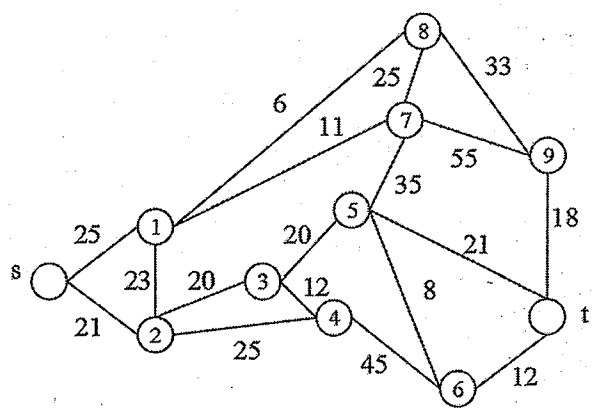
某小区有七栋楼房①~⑦（见下图），各楼房之间可修燃气管道路线的长度（单位：百米）已标记在连线旁。为修建连通各个楼房的燃气管道，该小区内部煤气管道的总长度至少为多少百米？



希赛

图论应用－最短路径

有一批货物要从城市s发送到城市t，线条上的数字代表通过这条路的费用（单位为万元）。那么，运送这批货物，至少需要花费多少元？



希赛

图论应用－最短路径

下表记录了六个结点A、B、C、D、E、F之间的路径方向和距离。从A到F的最短距离是（　）。

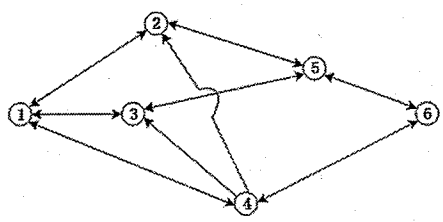
从＼到	B	C	D	E	F
A	11	16	24	36	54
B		13	16	21	29
C			14	17	22
D				14	17
E					15

A.38　　B.40　　C.44　　D.46

希赛

图论应用－网络与最大流量

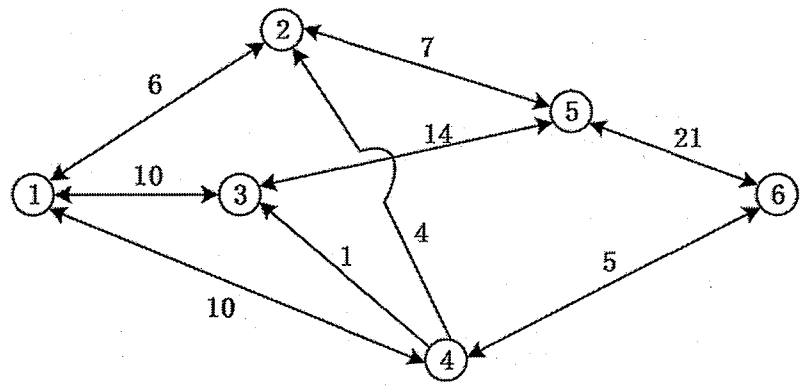
下图标出了某地区的运输网，各节点之间的运输能力如下表所示。那么，从节点①到节点⑥的最大运输能力（流量）可以达到多少万吨/小时？



	①	②	③	④	⑤	⑥
①		6	10	10		
②	6				7	
③	10				14	
④	10	4	1			5
⑤		7	14			21
⑥				5	21	

希赛

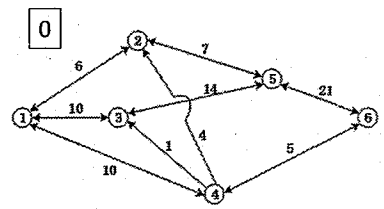
图论应用 – 网络与最大流量



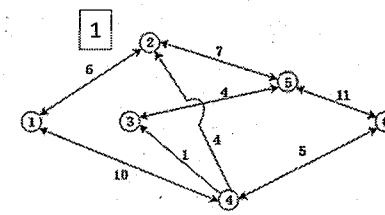
希赛

图论应用 – 网络与最大流量

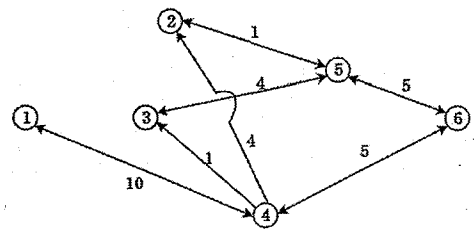
0



1

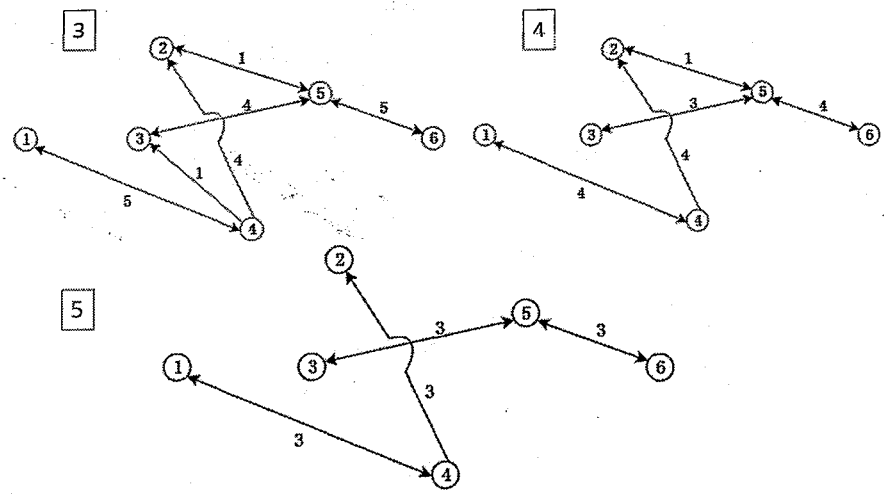


2



希赛

图论应用－网络与最大流量



希赛

运筹方法－线性规划

◆ 某企业需要采用甲、乙、丙三种原材料生产Ⅰ、Ⅱ两种产品。生产两种产品所需原材料数量、单位产品可获得利润以及企业现有原材料数如下表所示，则公司可以获得的最大利润是__(1)__万元。取得最大利润时，原材料__(2)__尚有剩余。

(1) A．21　　B．34　　C．39　　D．48

(2) A．甲　　B．乙　　C．丙　　D．乙和丙

		产品（吨）		现有原材料（吨）
		Ⅰ	Ⅱ	
所需资源	甲	1	1	4
	乙	4	3	12
	丙	1	3	6
单位利润(万元/吨)		9	12	

希赛

运筹方法－线性规划

- 设生产Ⅰ与Ⅱ产品的数量分别为：X和Y。则有：
- (1) X+Y <= 4
- (2) 4X+3Y <= 12　　9X+12Y=?
- (3) X+3Y <= 6

- (1) 与 (2) 求解得：X=0,Y=4。 X+3Y = 12
- (1) 与 (3) 求解得：X=3,Y=1。 4X+3Y = 15
- (2) 与 (3) 求解得：X=2,Y=4/3。 X+Y = 10/3

		产品（吨）		现有原材料（吨）	
		Ⅰ	Ⅱ		
所需资源	甲	1	1	4	2+4/3=10/3
	乙	4	3	12	2*4+3*(4/3)=12
	丙	1	3	6	2+3*(4/3)=6
单位利润(万元/吨)		9	12		

9*2+12*(4/3)=34

希赛

运筹方法－动态规划

某公司打算向它的三个营业区增设6个销售店，每个营业区至少增设1个。各营业区年增加的利润与增设的销售店个数有关，具体关系如表所示。可以调整各营业区增设的销售店的个数，使公司总利润增加额最大达__(　)__万元。

A. 520

B. 490

C. 470

D. 510

增设销售店个数	营业区A	营业区B	营业区C
1	100	120	150
2	160	150	165
3	190	170	175
4	200	180	190

营业区	A	B	C	利润值（万元）
增加销售店数	1	1	4	410
		2	3	425
		3	2	435
		4	1	430
	2	1	3	455
		2	2	475
		3	1	480
	3	1	2	475
		2	1	490
	4	1	1	470

希赛

运筹方法－转移矩阵

● 假设市场上某种商品有两种品牌A和B，当前的市场占有率各为50%。根据历史经验估计，这种商品当月与下月市场占有率的变化可用转移矩阵P来描述：

$$P=\begin{pmatrix} p(A\to A) & p(A\to B) \\ p(B\to A) & p(B\to B) \end{pmatrix}=\begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix}$$

其中，p（A→B）是A的市场占有份额中转移给B的概率，依次类推。这样，2个月后的这种商品的市场占有率变化为__（10）__。

（10）A．A的份额增加了10%，B的份额减少了10%

B．A的份额减少了10%，B的份额增加了10%

C．A的份额增加了14%，B的份额减少了14%

D．A的份额减少了14%，B的份额增加了14%

希赛

运筹方法－转移矩阵

第1个月后：

A：50%*0.8+50%*0.4=60%

B：1-60%=40%

第2个月后：

A： 60%*0.8+40%*0.4=64%

B：1-64%=36%

希赛

运筹方法－排队论

● 某博览会每天8:00开始让观众通过各入口处检票进场，8:00前已经有很多观众在排队等候。假设8:00后还有不少观众均匀地陆续到达，而每个入口处对每个人的检票速度都相同。根据以往经验，若开设8个入口，则需要60分钟才能让排队观众全部入场；若开设10个入口，则需要40分钟才能消除排队现象。为以尽量少的入口数确保20分钟后消除排队现象，博览会应在8:00和8:20开设的入口数分别为（ ）。

A．12,2　　B．14,4　　C．16,4　　D．18,6

希赛

运筹方法－排队论

设8点前已排队等候的人数为A，每分钟可以来Z人，每个入口每分钟能进Y人。

1式：8*60*Y=60*Z+A

2式：10*40*Y=40*Z+A

1式减2式得：

3式：80Y=20Z

把3式代入1式得：

A=240Y

所以要20分钟消除排队现象则有：

X*20*Y=20*(4Y)+240Y

求得X=16。

所以8:00应开入口16个，而8:20由于消除了排队，开口数量只需要4个就行了（依据：80Y=20Z）。

希赛

运筹方法－决策－不确定型决策

决策准则	说明
乐观主义准则	maxmax准则，其决策的原则是“大中取大”，总抱有乐观和冒险的态度，决不放弃任何获得最好结果的机会。在决策表中各个方案对各个状态的结果中选出最大者，记在表的最右列，再从该列中选出最大者
悲观主义准则	maxmin准则，其决策的原则是“小中取大”。抱有悲观和保守的态度，在各种最坏的可能结果中选择最好的。决策时从决策表中各方案对各个状态的结果选出最小者，记在表的最右列，再从该列中选出最大者
折中主义准则	Harwicz准则，既不乐观冒险，也不悲观保守，而是从中折中平衡，用一个系数α(称为折中系数)来表示，并规定$0\le\alpha\le1$，$cv_i=\alpha*\max\{a_{ij}\}+(1-\alpha)*\min\{a_{ij}\}$，然后比较cvi，从中选择最大者
等可能准则	Laplace准则，当决策者无法事先确定每个自然状态出现的概率时，可以将每个状态出现的概率定为1/n，然后按照EMV决策
后悔值准则	Savage准则，每个自然状态的最大收益值(损失矩阵取为最小值)作为该状态的理想目标，并将该状态的其它值与最大值的差作为未达到理想目标的后悔值。决策的原则是最大后悔值达到最小(minmax，大中取小，最小最大后悔值）

希赛

运筹方法－决策－不确定型决策

决策矩阵：

预计收益(单位万元人民币)		经济趋势预测		
		不景气	不变	景气
投资策略	积极	50	150	500
	稳健	100	200	300
	保守	300	250	200

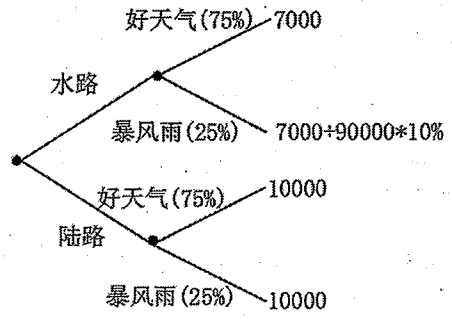
后悔值矩阵：

预计收益(单位万元人民币)		经济趋势预测		
		不景气	不变	景气
投资策略	积极	250	100	0
	稳健	200	50	200
	保守	0	0	300

希赛

运筹方法－决策树

某电子商务公司要从A地向B地的用户发送一批价值为90000元的货物。从A地到B地有水、陆两条路线。走陆路时比较安全，其运输成本为10000元；走水路时一般情况下的运输成本只要7000元，不过一旦遇到暴风雨天气，则会造成相当于这批货物总价值的10%的损失。根据历年情况，这期间出现暴风雨天气的概率为1/4，那么该电子商务公司该如何选择呢？



希赛

运筹方法－决策表

● 评估和选择最佳系统设计方案时，甲认为可以采用点值评估方法，即根据每一个价值因素的重要性，综合打分来选择最佳的方案。乙根据甲的提议，对如表所示的系统 A 和 B 进行评估，那么乙认为__( )__。

A．最佳方案是 A　　B．最佳方案是 B

C．条件不足，不能得出结论　　D．只能用成本/效益分析方法做出判断

		系统A	系统B
评估因素的重要性		评估值	评估值
硬件	35%	95	75
软件	40%	70	95
供应商支持	25%	85	90

希赛

数学建模

数学建模是一种数学的思考方法，是运用数学的语言和方法，通过抽象和简化，建立能近似刻画并解决实际问题的模型的一种强有力的数学手段。

数学建模过程

- 模型准备：了解问题的实际背景，明确其实际意义，掌握对象的各种信息。用数学语言来描述问题。
- 模型假设：根据实际对象的特征和建模的目的，对问题进行必要的简化，并用精确的语言提出一些恰当的假设。
- 模型建立：在假设的基础上，利用适当的数学工具来刻划各变量之间的数学关系，建立相应的数学结构。只要能够把问题描述清楚，尽量使用简单的数学工具。
- 模型求解：利用获取的数据资料，对模型的所有参数做出计算（估计）。
- 模型分析：对所得的结果进行数学上的分析。
- 模型检验：将模型分析结果与实际情形进行比较，以此来验证模型的准确性、合理性和适用性。如果模型与实际较吻合，则要对计算结果给出其实际含义，并进行解释。如果模型与实际吻合较差，则应该修改假设，再次重复建模过程。
- 模型应用：应用方式因问题的性质和建模的目的而异。

希赛

数学建模

数学建模方法

直接分析法：认识原理，直接构造出模型。
类比法：根据类似问题模型构造新模型。
数据分析法：大量数据统计分析之后建模。
构想法：对将来可能发生的情况给出设想从而建模。

希赛

知识产权与标准化

希赛

知识产权– 课程内容提要

- 保护范围与对象（★★★）
- 保护期限（★★）
- 知识产权人确定（★★★）
- 侵权判断（★★★★）
- 标准的分类（★）
- 标准代号的识别（★）

希赛

一 保护范围与对象

法律法规名称	保护对象及范围	注意事项
著作权法	著作权 文学、绘画、摄影等作品	1、不需要申请，作品完成即开始保护 2、绘画或摄影作品原件出售（赠予）著作权还归原作者，原件拥有者有：所有权、展览权。
软件著作权法 计算机软件保护条例	软件著作权 软件作品	1、不需要申请，作品完成即开始保护 2、登记制度便于举证
专利法	专利权	需要申请，专利权有效期是从申请日开始计算
商标法	商标权	需要申请，核准之日起商标受保护
反不正当竞争法	商业秘密权	1、商业秘密包括技术与经营两个方面 2、必须有保密措施才能认定商业秘密

希赛

二 知识产权－保护期限

客体类型	权力类型	保护期限
公民作品	署名权、修改权、保护作品完整权	没有限制
	发表权、使用权和获得报酬权	作者终生及其死亡后的50年（第50年的12月31日）
单位作品	发表权、使用权和获得报酬权	50年（首次发表后的第50年的12月31日），若其间未发表，不保护。
公民软件产品	署名权、修改权	没有限制
	发表权、复制权、发行权、出租权、信息网络传播权、翻译权、使用许可权、获得报酬权、转让权	作者终生及死后50年（第50年12月31日）。合作开发，以最后死亡作者为准。
单位软件产品	发表权、复制权、发行权、出租权、信息网络传播权、翻译权、使用许可权、获得报酬权、转让权	50年（首次发表后的第50年的12月31日），若其间未发表，不保护
注册商标		有效期10年（若注册人死亡或倒闭1年后，未转移则可注销，期满后6个月内必须续注）
发明专利权		保护期为20年（从申请日开始）
实用新型和外观设计专利权		保护期为10年（从申请日开始）
商业秘密		不确定，公开后公众可用

希赛

二 知识产权 –知识产权人确定

情况说明		判断说明	归属
作品	职务作品	利用单位的物质技术条件进行创作，并由单位承担责任的	除署名权外其他著作权归单位
		有合同约定，其著作权属于单位	除署名权外其他著作权归单位
		其他	作者拥有著作权，单位有权在业务范围内优先使用
软件	职务作品	属于本职工作中明确规定的开发目标	单位享有著作权
		属于从事本职工作活动的结果	单位享有著作权
		使用了单位资金、专用设备、未公开的信息等物质、技术条件，并由单位或组织承担责任的软件	单位享有著作权
专利权	职务作品	本职工作中作出的发明创造	单位享有专利
		履行本单位交付的本职工作之外的任务所作出的发明创造	单位享有专利
		离职、退休或调动工作后1年内，与原单位工作相关	单位享有专利

希赛

二 知识产权 –知识产权人确定

情况说明		判断说明	归属
作品软件	委托创作	有合同约定，著作权归委托方	委托方
		合同中未约定著作权归属	创作方
	合作开发	只进行组织、提供咨询意见、物质条件或者进行其他辅助工作	不享有著作权
		共同创作的	共同享有，按人头比例。 成果可分割的，可分开申请。
商标		谁先申请谁拥有（除知名商标的非法抢注） 同时申请，则根据谁先使用（需提供证据） 无法提供证据，协商归属，无效时使用抽签（但不可不确定）	
专利		谁先申请谁拥有 同时申请则协商归属，但不能够同时驳回双方的专利申请	

希赛

二 知识产权－侵权判定（1）

◆ 中国公民、法人或者其他组织的作品，不论是否发表，都享有著作权。

◆ 开发软件所用的思想、处理过程、操作方法或者数学概念不受保护

★ 著作权法不适用于下列情形：

- 法律、法规，国家机关的决议、决定、命令和其他具有立法、行政、司法性质的文件，及其官方正式译文；
- 时事新闻；
- 历法、通用数表、通用表格和公式。

希赛

二 知识产权－侵权判定（2）

不侵权	侵权
✓ 个人学习、研究或者欣赏； ✓ 适当引用； ✓ 公开演讲内容 ✓ 用于教学或科学研究 ✓ 复制馆藏作品； ✓ 免费表演他人作品； ✓ 室外公共场所艺术品临摹、绘画、摄影、录像； ✓ 将汉语作品译成少数民族语言作品或盲文出版。	✓ 未经许可，发表他人作品； ✓ 未经合作作者许可，将与他人合作创作的作品当作自己单独创作的作品发表的； ✓ 未参加创作，在他人作品署名； ✓ 歪曲、篡改他人作品的； ✓ 剽窃他人作品的； ✓ 使用他人作品，未付报酬； ✓ 未经出版者许可，使用其出版的图书、期刊的版式设计的。

希赛

二 知识产权－保护期限

◆ 以下著作权权利中，（ ）的保护期受时间限制。

A．署名权

B．发表权

C．修改权

D．保护作品完整权

◆ 以下作品中，不适用或不受著作权法保护的作品是（ ）。

A．国务院颁布的《计算机软件保护条例》

B．某作家的作品《绿化树》

C．最高人民法院组织编写的《行政诉讼案例选编》

D．某人在公共场所的即兴演说

希赛

三 标准化－课程内容提要

- **标准的分类**
- **标准的编号**

希赛

三 标准化－标准的分类

- 国际标准：ISO、IEC等国际标准化组织
- 国家标准：GB—中国、ANSI—美国、BS—英国、JIS—日本
- 区域标准：又称为地区标准，如PASC—太平洋地区标准会议、CEN—欧洲标准委员会、ASAC—亚洲标准咨询委员会、ARSO—非洲地区标准化组织
- 行业标准：GJB—中国军用标准、MIT-S—美国军用标准、IEEE—美国电气电子工程师协会
- 地方标准：国家的地方一级行政机构制订的标准
- 企业标准
- 项目规范

希赛

三 标准化－标准的编号

- 国际、国外标准代号：标准代号+专业类号+顺序号+年代号
- 我国国家标准代号：强制性标准代号为GB、推荐性标准代号为GB/T
  指导性标准代号为GB/Z、实物标准代号GSB
- 行业标准代号：由汉语拼音大写字母组成（如电子行业为SJ）
- 地方标准代号：由DB加上省级行政区划代码的前两位
- 企业标准代号：由Q加上企业代号组成

希赛

系统架构设计案例分析

希赛

调研：

大家觉得案例难，难在哪里？

希赛

案例 = 心态 + 技巧 + 理论

希赛

课程内容提要

- 考点分析
- 如何解答试题
- 试题解答实例

希赛

考点分析 – 考试大纲

- 系统规划
- 软件架构设计
- 设计模式
- 系统设计
- 系统建模
- 分布式系统设计
- 嵌入式系统设计
- 系统的可靠性分析与设计
- 系统的安全性和保密性设计

希赛

历年考试情况

时间	主要知识点	考查内容说明
09.11	软件质量特性	软件质量特性中子特性的归类，每个质量特性在开发中体现于哪些场景，或哪些需求对应着哪些特性
	系统流程图与数据流图	系统流程图与数据流图的区别、以及数据流图的基本应用
	嵌入式系统架构	架构风格
	系统软件架构	系统软件架构的选择
	系统安全	安全风险与身份认证方式
10.11	软件架构设计	常见架构风格的选用
	数据架构建模	分布式数据库的设计
	嵌入式系统	层次化架构
	ESB	ESB及设计模式
	系统可靠性	可靠性的计算、检错技术和容错技术

希赛

历年考试情况

时间	主要知识点	考查内容说明
11.11	软件架构评估	软件质量属性场景、架构风险、敏感点、权衡点
	软件系统数据建模	反规范化技术的优点、缺点、实施技术
	嵌入式系统设计	嵌入式系统架构及系统设计
	Web应用系统架构设计	数据采集与统计方案选择、REST
	信息系统安全性	对称加密策略与公钥加密策略、机密性与完整性机制、授权决策机制
12.11	软件架构风格	不同架构风格的优势、劣势、应用场合的比较分析
	分布式系统设计	ODP的五个视点、分布式中间件的应用、分布式系统通信问题
	嵌入式软件架构设计	GOA的相关概念与特点、GOA中的一系列接口
	设计模式	创建型、结构型、行为型模型的作用，设计模式的选择与应用
	数据架构设计	NoSQL的特点及优缺点

希赛

历年考试情况

时间	主要知识点	考查内容说明
13.11	架构设计	服务建模（服务的规划、服务的内容）、ESB
	项目管理	项目计划内容、项目绩效计算、项目延期解决方案
	嵌入式软件FMEA方法	故障模式分析及故障影响分析、FMEA步骤、软件风险优先数
	设计模式与XML	MVC设计模式、基于XML的界面管理技术
	信息安全	公钥体系、数据库敏感字段加密
14.11	架构设计	MVC、扩展接口模式
	系统过程建模	数据流图、CRUD矩阵
	嵌入式构件设计	遗留工程、构件技术
	架构评估	质量属性效用树、架构风险、敏感点和权衡点
	Web应用架构设计	负载均衡、磁盘I/O瓶颈问题、查询缓存(query cache)机制、Memcached分布式缓存机制

希赛

历年考试情况

时间	主要知识点	考查内容说明
15.11	软件架构评估	软件质量属性场景、架构风险、敏感点、权衡点
	系统设计建模	状态图与活动图的填充
	嵌入式系统可靠性分析	可靠性相关概念、恢复块方法、N版本程序设计
	应用系统数据架构	关系型数据库、文件系统、内存数据库、SQL设计策略
	Web应用系统架构	MVC模式、Hibernate框架、iBatis
16.11	软件架构设计	软件质量属性对应的需求、管道—过滤器风格和数据仓储风格对比，填充架构图
	软件系统建模	用例图与类图，用例关系与类之间的关系
	嵌入式实时系统设计	实时系统的特点、实时特性分类、可靠性
	Web系统应用	应用服务器、网站技术、大负荷和长时间运行下的稳定性以及可扩展性、J2EE的N层结构
	敏捷开发	Scrum敏捷开发过程、MVC、分层架构

希赛

历年考试情况

时间	主要知识点	考查内容说明
17.11	软件架构评估	软件质量属性场景、架构风险、敏感点、权衡点
	系统设计	MVC概念以及组成元素、EJB、有状态与无状态
	嵌入式系统	机器人操作系统ROS
	数据库设计	ORM优缺点、增加数据访问层的原因、工厂设计模式
	Web应用系统架构	响应式WEB设计，架构设计图填充，主从复制机制
18.11	系统设计	非功能性需求分类，瘦客户端C/S架构
	软件系统建模	结构化与面向对象分析，DFD，ERD，用例
	嵌入式系统	实时任务中简单任务和复杂任务的区分，“腰”型通信模式架构风格，
	分布式数据库	MemCache与Redis、数据可靠性和一致性、Redis集群切片的几种常见方式
	Web系统设计	什么是面向服务架构（SOA），ESB在SOA中的作用与特点，Web系统架构，信息系统安全保障的措施

希赛

如何解答试题 – 试题对考生的要求

- 具有一定的系统架构设计实践经验，有较好的分析问题和解决问题的能力
- 对于有关系统架构设计方面，有广博而坚实的知识或见解
- 对应用的背景、事实和因果关系等有较强的理解能力和归纳能力
- 对于一些可以简单定量分析的问题已有类似经验并能进行估算，对于只能定性分析的问题能用简练的语言抓住要点加以表达
- 善于从一段书面叙述中提取出最必要的信息，有时还需要舍弃一些无用的叙述或似是而非的内容

希赛

如何解答试题 – 试题解答步骤

- 标出问题要点，以此作为主要线索进行分析和思考
- 对照问题要点仔细阅读正文
- 通过定性分析或者定量估算，构思答案的要点
- 以最简练的语言写出答案

希赛

如何解答试题 －试题解答注意事项

- 遇到新的知识点，不要慌，稳住心态
- 列条目回答问题，把自己认为对的，都写上
- 分析题目问题的倾向性，顺势答题

希赛

案例分析1

某网上购物电子商务公司拟升级正在使用的在线交易系统，以提高用户网上购物在线支付环节的效率和安全性。在系统的需求分析与架构设计阶段，公司提出的需求和关键质量属性场景如下：

（a）正常负载情况下，系统必须在0.5秒内对用户的交易请求进行响应；

（b）信用卡支付必须保证99.999%的安全性；

（c）对交易请求处理时间的要求将影响系统的数据传输协议和处理过程的设计；

（d）网络失效后，系统需要在1.5分钟内发现错误并启用备用系统；

（e）需要在20人月内为系统添加一个新的CORBA中间件；

（f）交易过程中涉及到的产品介绍视频传输必须保证画面具有600*480的分辨率，20帧/秒的速率；

（g）更改加密的级别将对安全性和性能产生影响；

（h）主站点断电后，需要在3秒内将访问请求重定向到备用站点；

（i）假设每秒中用户交易请求的数量是10个，处理请求的时间为30毫秒，则“在1秒内完成用户的交易请求”这一要求是可以实现的；

（j）用户信息数据库授权必须保证99.999%可用；

（k）目前对系统信用卡支付业务逻辑的描述尚未达成共识，这可能导致部分业务功能模块的重复，影响系统的可修改性；

（l）更改Web界面接口必须在4人周内完成；

（m）系统需要提供远程调试接口，并支持系统的远程调试。

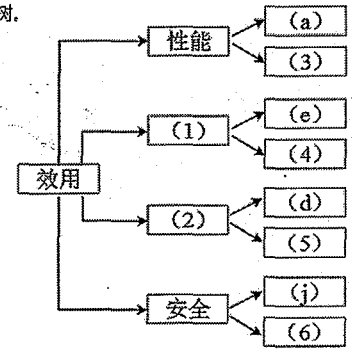
在对系统需求和质量属性场景进行分析的基础上，系统的架构师给出了三个候选的架构设计方案。公司目前正在组织系统开发的相关人员对系统架构进行评估。

希赛

案例分析1

【问题1】

在架构评估过程中，质量属性效用树（utility tree）是对系统质量属性进行识别和优先级排序的重要工具。请给出合适的质量属性，填入下图中（1）、（2）空白处；并选择题干描述的（a）~（m），填入（3）~（6）空白处，完成该系统的效用树。



【问题2】

在架构评估过程中，需要正确识别系统的架构风险、敏感点和权衡点，并进行合理的架构决策。请用300字以内的文字给出系统架构风险、敏感点和权衡点的定义，并从题干（a）~（m）中各选出1个对系统架构风险、敏感点和权衡点最为恰当的描述。

希赛

案例分析1 - 参考答案

系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。

敏感点是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点是影响多个质量属性的特性，是多个质量属性的敏感点。

希赛

案例分析1 - 参考答案

（a）正常负载情况下，系统必须在0.5秒内对用户的交易请求进行响应；**（性能）**
（b）信用卡支付必须保证99.999%的安全性；**（安全性）**
（c）对交易请求处理时间的要求将影响系统的数据传输协议和处理过程的设计；**（敏感点）**
（d）网络失效后，系统需要在1.5分钟内发现错误并启用备用系统；**（可用性）**
（e）需要在20人月内为系统添加一个新的CORBA中间件；**（可修改性）**
（f）交易过程中涉及到的产品介绍视频传输必须保证画面具有600*480的分辨率，20帧/秒的速率；**（性能）**
（g）更改加密的级别将对安全性和性能产生影响；**（权衡点）**
（h）主站点断电后，需要在3秒内将访问请求重定向到备用站点；**（可用性）**
（i）假设每秒中用户交易请求的数量是10个，处理请求的时间为30毫秒，则"在1秒内完成用户的交易请求"这一要求是可以实现的；
（j）用户信息数据库授权必须保证99.999%可用；**（安全性）**
（k）目前对系统信用卡支付业务逻辑的描述尚未达成共识，这可能导致部分业务功能模块的重复，影响系统的可修改性；**（风险点）**
（l）更改Web界面接口必须在4人周内完成；**（可修改性）**
（m）系统需要提供远程调试接口，并支持系统的远程调试。**（可测试性）**

希赛

案例分析2

某软件企业为影音产品销售公司W开发一套在线销售系统，以提升服务的质量和效率。项目组经过讨论后决定采用面向对象方法开发该系统。在设计建模阶段需要满足以下设计要求。

（1）W公司经常进行促销活动。根据不同的条件（如订单总额、商品数量、产品种类等），公司可以提供百分比折扣或现金减免等多种促销方式供提交订单的用户选择。实现每种促销活动的代码量很大，且会随促销策略的不同经常修改。系统设计中需要考虑现有的促销和新的促销，而不用经常地重写控制器类代码。

（2）该在线销售系统需要计算每个订单的税率，不同商品的税率及计算方式会有所区别。所以W公司决定在系统中直接调用不同商品供应商提供的税率计算类，但每个供应商的类提供了不同的调用方法。系统设计中需要考虑如果公司更换了供应商，应该尽可能少地在系统中修改或创建新类。

项目组架构师决定采用设计模式来满足上述设计要求，并确定从当前已经熟练掌握的设计模式中进行选择，这些设计模式包括：适配器模式（Adapter）、构造器模式（Builder）、命令模式（Command）、外观模式（Facade）、中介模式（Mediator）、原型模式（Prototype）、代理模式（Proxy）、状态模式（State）和策略模式（Strategy）等。

希赛

案例分析2

【问题1】

设计模式按照其应用模式可以分为三类：创建型、结构型和行为型，请用200字以内文字说明三者的作用。

【问题2】

请将项目组已经掌握的设计模式按照其作用分别归类到创建型、结构型和行为型模式中。

【问题3】

针对题目中所提出的设计要求（1）和（2），项目组应该分别选择何种设计模式？请分别用200字以内文字说明具体的解决方案。

希赛

案例分析2- 参考答案

创建型模式主要用于创建对象，为设计类实例化新对象提供指南。

结构型模式主要用于处理类或对象的组合，对类如何设计以形成更大的结构提供指南。

行为型模式主要用于描述类或对象的交互以及职责的分配，对类之间交互以及分配责任的方式提供指南。

希赛

案例分析2 - 参考答案

- **创建性模式**
  - 工厂方法（factory method）模式
  - 抽象工厂（abstract factory）模式
  - 原型（prototype）模式
  - 单例（singleton）模式
  - 构建器（builder）模式
- **结构性模式**
  - 适配器（adapter）模式
  - 桥接（bridge）模式
  - 组合（composite）模式
  - 装饰（decorator）模式
  - 外观（facade）模式
  - 享元（flyweight）模式
  - 代理（proxy）模式
- **行为性模式**
  - 职责链（chain of responsibility）模式
  - 命令（command）模式
  - 解释器（interpreter）模式
  - 迭代器（iterator）模式
  - 中介者（mediator）模式
  - 备忘录（memento）模式
  - 观察者（observer）模式
  - 状态（state）模式
  - 策略（strategy）模式
  - 模板方法（template method）模式
  - 访问者（visitor）模式

希赛

案例分析2- 参考答案

（1）策略模式

解决方案：在具有公共接口的独立类中定义每个计算。可以利用该模式创建各种促销类，它们从同一个超类继承。每个类都有相同名称的标准接口方法，用于根据订单编号计算将要折扣的金额总数。计算每个促销的内部代码对促销类来说完全不同。

（2）适配器模式

解决方案：增加一个类作为适配器，转换类的接口到客户端类期望的另一个接口。实现一个适配器类，这个类为系统的其他部分提供了一个不变的方法供调用，为了集成不同商品供应商提供的税率计算类，编写一个适配器类的子类，包含调用购买类所需的代码。该子类将系统的调用映射到某个供应商的税率计算类。如果要更换供应商，那么只需要写一个新的适配器子类，其他保持不变。

希赛

案例分析3

某软件公司拟开发一套基于局域网的分布式系统，该系统由分布于某企业各部门的多个子系统构成。在该企业的日常运作中，各子系统之间要经常基于企业局域网进行交互。

实现各子系统之间的交互可以采用如下两种方式：

（1）基于某种成熟的分布式软件架构（如EJB，CORBA，DCOM/COM+）来构建整个系统。现在主流的分布式软件架构都融合了面向对象技术，对分布式对象提供了很好的支持，可以利用这些架构支持分布式对象访问的通信机制（如RMI/IIOP，ORB，ORPC等）来实现各子系统之间的交互，其优点是实现相对简单且比较可靠。

（2）先分别实现各个子系统，然后利用底层通信协议（如TCP/IP）实现各子系统之间的交互，其优点是通信效率高且可控性好。

【问题1】

虽然不同的分布式软件架构采用的具体实现方式不尽相同，但它们都支持客户端透明地访问分布式对象，即客户端可以像访问本地对象一样访问分布式对象。请用200字以内文字，说明实现这种透明性的基本原理。

【问题2】

由于应用的具体需求千差万别，再好的分布式软件架构也不可能适应所有的应用系统，有时不得不放弃现有的分布式软件架构的支持，自己利用底层的通信协议来实现各子系统之间的交互。请用200字以内文字，简要说明用底层通信协议实现各子系统之间的交互时要解决的主要问题。

希赛

案例分析3- 参考答案

【问题1】

客户端和服务端不是直接进行交互，而是利用客户端存根和服务端框架来间接进行通信，这样客户程序和服务程序就不需考虑底层的通信细节问题。

由于客户端存根和服务端框架一般由平台自动生成，不需程序员手工编写，所以这种通信模型的最大好处是可以省去程序员自己写程序来处理底层通信问题。

【问题2】

（1）异类系统间交互的兼容性：例如处理不同操作系统的字节顺序、数据长度、数据表示方式等。

（2）通信的可靠性问题：例如出错处理、重传机制等。

（3）通信的安全性问题。

（4）接口的可用性、易用性问题。

希赛

案例分析4

A公司是一家中等规模的计算机企业，专业从事网络安全防护软件系统的开发。从最初仅开发基于Windows的个人防火墙产品开始，现在，已经延伸到基于Linux、Windows系列、MAC操作系统的个人防火墙、企业防火墙、入侵检测系统、病毒扫描系统、安全扫描系统等多种产品。公司原来的产品都是一个一个地开发，为每个软件对应地组织一个项目组。

为了适应快速变化的市场，降低开发成本，公司想引入产品线方法。然而，由于软件产品线方法涉及了一个软件开发企业的多个产品，所以，公司的王总决定在弄清楚以下三个问题之后再做决定，首先就是本公司的业务范围是否适合使用产品线方法，其次是如何在原有产品的基础上建立产品线，最后是成功实施产品线的主要因素是什么？

【问题1】

请用100字以内文字，说明A公司是否适合采用产品线方法，并说明理由。

【问题2】

请用400字以内文字，说明在原有产品的基础上建立软件产品线的方式，并作简要评价。

【问题3】

请用150字以内文字，说明成功实施产品线的主要因素。

希赛

案例分析4- 参考答案

【问题1】

是否适合采用产品线开发方法关键在于企业是否存在拥有共性领域模型的产品集合。因为A公司的业务是在多平台下开发一系列相关网络安全防护软件，因此适合采用产品线方法。

【问题2】

在原有的产品的基础上建立软件产品线主要有两种方式。

（1）演化方式：即将现有产品演化为产品线，也就是将特定产品的构件逐渐转化到产品线的共用构件。该方式分解了投资回报周期，最大程度地减少了实施产品线方法的风险，但完成软件产品线核心资源的总周期和总投资都较大。

（2）革命方式：即用软件产品线替代现有产品集。这种方法将基本停止现有产品的开发，直接针对软件产品线的核心资源开发。虽然该方法完成软件产品线核心资源的总周期和总投资都会减少，但风险大大加大了。

【问题3】

（1）对该领域具备长期和深厚的经验。

（2）一个用于构建产品的好的核心资源库。

（3）好的产品线架构。

（4）好的管理（软件资源、人员组织、过程）支持。

希赛

案例分析5

某大型跨国企业的IT部门一年前基于SOA（Service-Oriented Architecture）对企业原有的多个信息系统进行了集成，实现了原有各系统之间的互连互通，搭建了支撑企业完整业务流程运作的统一信息系统平台。随着集成后系统的投入运行，IT部门发现在满足企业正常业务运作要求的同时，系统也暴露出明显的安全性缺陷，并在近期出现了企业敏感业务数据泄漏及系统核心业务功能非授权访问等严重安全事件。针对这一情况，企业决定由IT部门成立专门的项目组负责提高现有系统的安全性。

项目组在仔细调研和分析了系统现有安全性问题的基础上，决定首先为在网络中传输的数据提供机密性（Confidentiality）与完整性（Integrity）保障，同时为系统核心业务功能的访问提供访问控制机制，以保证只有授权用户才能使用特定功能。

经过分析和讨论，项目组决定采用加密技术为网络中传输的数据提供机密性与完整性保障。但在确定具体访问控制机制时，张工认为应该采用传统的强制访问控制（Mandatory Access Control）机制，而王工则建议采用基于角色的访问控制（Role-Based Access Control）与可扩展访问控制标记语言（eXtensible Access Control Markup Language，XACML）相结合的机制。项目组经过集体讨论，最终采用了王工的方案。

希赛

案例分析5

【问题1】

请用400字以内的文字，分别针对采用对称加密策略与公钥加密策略，说明如何利用加密技术为在网络中传输的数据提供机密性与完整性保障。

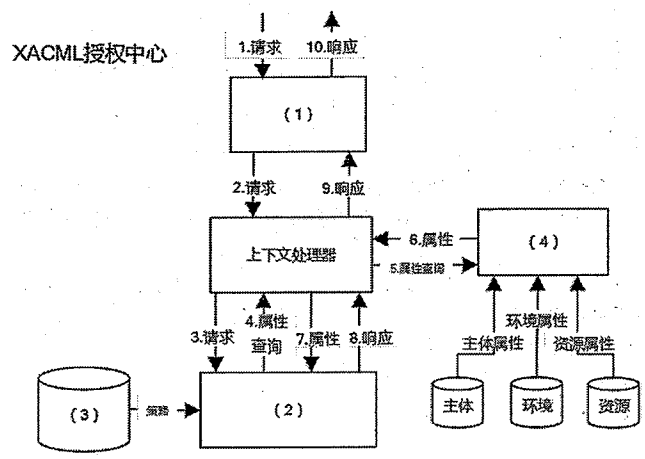
【问题2】

请用300字以内的文字，从授权的可管理性、细粒度访问控制的支持和对分布式环境的支持三个方面指出项目组采用王工方案的原因。

【问题3】

下图给出了基于XACML的授权决策中心的基本结构以及一次典型授权决策的执行过程，请分别将备选答案填入图中的（1）~（4）。

案例分析5



案例分析5 - 参考答案

【问题1】

1、对称加密策略

（1）机密性：发送者利用对称密钥对要发送的数据进行加密，只有拥有正确相同密钥的接收者才能将数据正确解密，从而提供机密性。

（2）完整性：发送者根据要发送的数据生成消息认证码（或消息摘要），利用对称密钥对消息认证码进行加密并附加到数据上发送；接收者使用相同密钥将对方发送的消息认证码解密，并根据接收到的数据重新生成消息认证码，比较两个认证码是否相同以验证数据的完整性。

2、公钥加密策略

（1）机密性：发送者利用接收者的公钥对要发送的数据进行加密，只有拥有对应私钥的接收者才能将数据正确解密，从而提供机密性。

（2）完整性：发送者根据要发送的数据生成消息认证码（或消息摘要），利用自己的私钥对消息认证码进行加密并附加到数据上发送；接收者利用对方的公钥将对方发送的消息认证码解密，并根据接收到的数据重新生成消息认证码，比较两个认证码是否相同以验证数据的完整性。

希赛

案例分析5 - 参考答案

【问题2】

（1）授权的可管理性：RBAC将用户与权限分离，与MAC相比，减小了授权管理的复杂性，更适合于大型企业级系统的安全管理。

（2）细粒度访问控制的支持：XACML提供了统一的访问控制策略描述语言，策略表达能力强，可用来描述各种复杂的和细粒度的访问控制安全需求，更适合企业复杂业务功能的访问控制要求。

（3）分布式环境的支持：XACML的标准性便于各子系统的协作交互，各子系统或企业业务部门可以分布管理访问控制权限，而MAC则通常需要对访问控制权限集中管理，不太适合企业基于SOA集成后的分布式系统。

【问题3】

（1）策略执行点（PEP）。

（2）策略决策点（PDP）。

（3）策略管理点（PAP）。

（4）策略信息点（PIP）。

希赛

案例分析6

某软件公司为其新推出的字处理软件设计了一种脚本语言，专门用于开发该字处理软件的附加功能插件。为了提高该语言的编程效率，公司组织软件工具开发部门为脚本语言研制一套集成开发环境。软件工具开发部门根据字处理软件的特点，对集成开发环境进行了需求分析，总结出以下3项核心需求：

（1）集成开发环境需要提供对脚本语言的编辑、语法检查、解释、执行和调试等功能的支持，并要实现各种功能的灵活组合、配置与替换。

（2）集成开发环境需要提供一组可视化的编程界面，用户通过对界面元素拖拽和代码填充的方式就可以完成功能插件核心业务流程的编写与组织。

（3）在代码调试功能方面，集成开发环境需要实现在脚本语言编辑界面中的代码自动定位功能。具体来说，在调试过程中，编辑界面需要响应调试断点命中事件，并自动跳转到当前断点处所对应的代码。

针对上述需求，软件工具开发部门对集成开发环境的架构进行分析与设计，王工认为该集成开发环境应该采用管道-过滤器的架构风格实现，李工则认为该集成开发环境应该采用以数据存储为中心的架构风格来实现。公司组织专家对王工和李工的方案进行了评审，最终采用了李工的方案。

希赛

案例分析6

【问题1】

请用200字以内的文字解释什么是软件架构风格，并从集成开发环境与用户的交互方式、集成开发环境的扩展性、集成开发环境的数据管理三个方面说明为什么最终采用了李工的设计方案。

【问题2】

在对软件系统架构进行设计时，要对架构需求进行分析，针对特定需求选择最为合适的架构风格，因此实际的软件系统通常会混合多种软件架构风格。请对核心需求进行分析，说明为了满足需求（2）和（3），分别应采用何种架构风格，并概要说明采用相应架构风格后的架构设计过程。

希赛

案例分析6

【问题1】

软件架构风格是指描述特定软件系统组织方式的惯用模式。组织方式描述了系统的组成构件和这些构件的组织方式，惯用模式则反映众多系统共有的结构和语义。

从集成开发环境与用户的交互方式看，用户通常采用交互式的方式对脚本语言进行编辑、解释执行与调试。在这种情况下，采用以数据存储为中心的架构风格能够很好地支持交互式数据处理，而管道-过滤器架构风格则对用户的交互式数据处理支持有限。

从集成开发环境的扩展性来看，系统核心需求要求实现各种编辑、语法检查、解释执行等多种功能的灵活组织、配置与替换。在这种情况下，采用以数据存储为中心的架构风格，以数据格式解耦各种功能之间的依赖关系，并可以灵活定义功能之间的逻辑顺序。管道-过滤器架构风格同样以数据格式解耦数据处理过程之间的依赖关系，但其在数据处理逻辑关系的灵活定义方面较差。

从集成开发环境的数据管理来看，集成开发环境需要支持脚本语言、语法树（用于检查语法错误）、可视化模型、调试信息等多种数据类型，并需要支持数据格式的转换。以数据存储为中心的架构将数据存储在统一的中心存储器中，中心存储器能够表示多种数据格式，并能够为数据格式转换提供各种支持。管道-过滤器架构风格通常只能支持有限度的数据格式，并且在数据格式转换方面的灵活性较差。

希赛

案例分析6

【问题2】

为了满足需求（2），应该采用解释器架构风格。具体来说，需要：① 为可视化编程元素及其拖拽关系定义某种语言，并描述其语法与语义；② 编写解释器对该语言进行解释；③ 生成对应的脚本语言程序。

为了满足需求（3），应该采用隐式调用架构风格。具体来说，首先需要定义“断点在调试过程中命中”这一事件，并实现当断点命中后的屏幕定位函数。集成开发环境维护一个事件注册表结构，将该事件与屏幕定位函数关联起来形成注册表中的一个记录项。在调试过程中，集成开发环境负责监听各种事件，当“断点在调试过程中命中”这一事件发生时，集成开发环境查找事件注册表，找到并调用屏幕定位函数，从而实现脚本语言编辑界面与调试代码的自动定位。

希赛

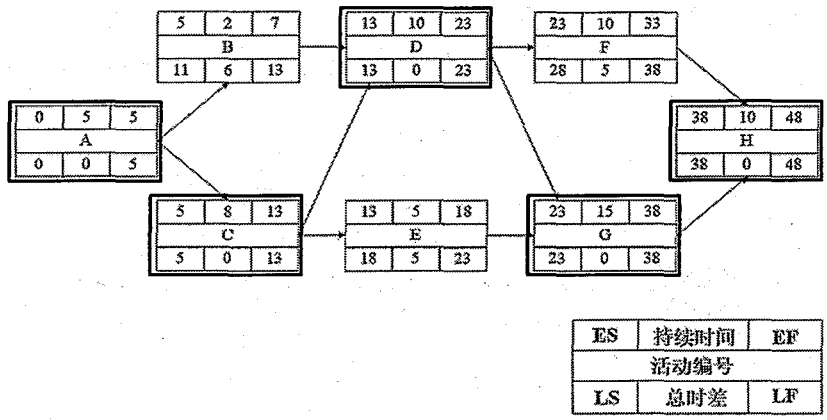
关键路径

- 关键路径法是在制订进度计划时使用的一种进度网络分析技术。关键路线法沿着项目进度网络路线进行正向与反向分析，从而计算出所有计划活动理论上的最早开始与完成日期、最迟开始与完成日期，不考虑任何资源限制

- 总时差：在不延误总工期的前提下，该活动的机动时间。活动的总时差等于该活动最迟完成时间与最早完成时间之差，或该活动最迟开始时间与最早开始时间之差
- 自由时差：在不影响紧后活动的最早开始时间前提下，该活动的机动时间
  - 对于有紧后活动的活动，其自由时差等于所有紧后活动最早开始时间减本活动最早完成时间所得之差的最小值
  - 对于没有紧后活动的活动，也就是以网络计划终点节点为完成节点的活动，其自由时差等于计划工期与本活动最早完成时间之差
- 对于网络计划中以终点节点为完成节点的活动，其自由时差与总时差相等。此外，由于活动的自由时差是其总时差的构成部分，所以，当活动的总时差为零时，其自由时差必然为零，可不必进行专门计算

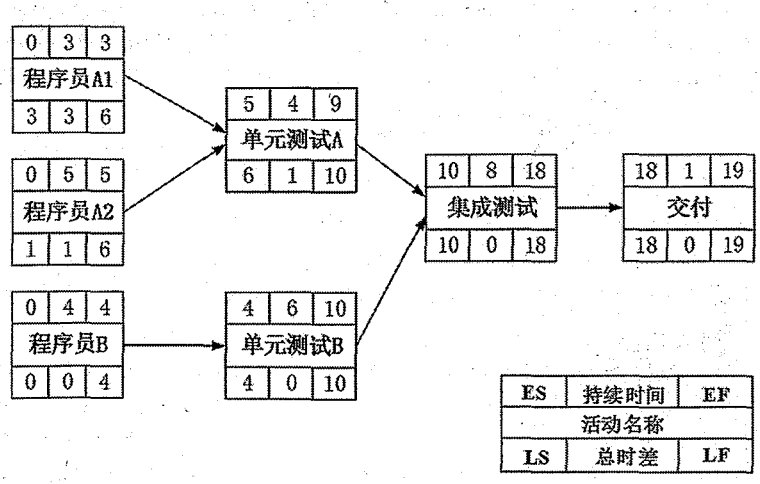
希赛

关键路径



希赛

制订进度计划－关键路径法－示例



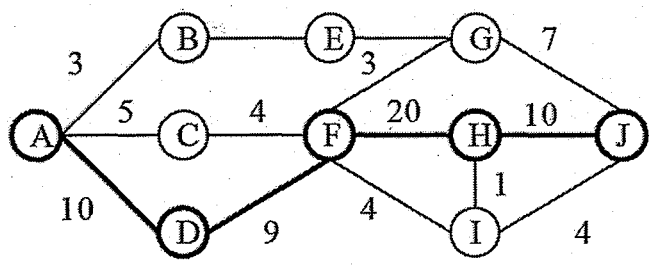
希赛

关键路径

● 在下面的活动图中，从A到J的关键路径是__（2）__，I和J之间的活动开始的最早时间是__（3）__。

（2）A．ABEGJ　　B．ADFHJ　　C．ACFGJ　　D．ADFIJ

（3）A．13　　B．23　　C．29　　D．40



希赛

案例分析7

某软件公司拟开发一套电子商务系统，王工作为项目组负责人负责编制项目计划。由于该企业业务发展需要，CEO急于启动电子商务系统，要求王工尽快准备一份拟开发系统的时间和成本估算报告。

项目组经过讨论后，确定出与项目相关的任务如表所示。其中，根据项目组开发经验，分别给出了正常工作及加班赶工两种情况下所需的时间和费用。

【问题1】

请用400字以内文字说明王工拟编制的项目计划中应包括哪些内容。

【问题2】

请根据表分别给出正常工作和最短工期两种情况下完成此项目所需的时间和费用。

【问题3】

如果项目在系统调研阶段用了7天时间才完成，公司要求尽量控制成本，王工可在后续任务中采取什么措施来保证项目能按照正常工作进度完成？

【问题4】

如果企业CEO想在34天后系统上线，王工应该采取什么措施来满足这一要求？这种情况下完成项目所需的费用是多少？

希赛

案例分析7

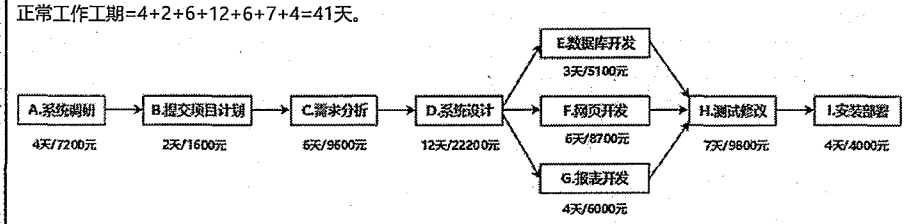
表 项目开发任务进度及费用

任务名称	正常工作	加班工作	前置任务
A.系统调研	4天/7200元	3天/8400元	-
B.提交项目计划	2天/1600元	1天/1900元	A
C.需求分析	6天/9600元	4天/14200元	B
D.系统设计	12天/22200元	8天/27600元	C
E.数据库开发	3天/5100元	2天/5700元	D
F.网页开发	6天/8700元	5天/10000元	D
G.报表开发	4天/6000元	任务外包无法赶工	D
H.测试修改	7天/9800元	4天/12800元	E,F,G
I.安装部署	4天/4000元	2天/5000元	H

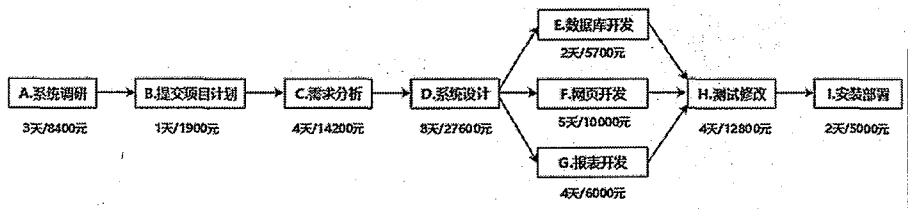
希赛

案例分析7

正常工作成本=7200元+1600元+9600元+22200元+5100元+8700元+6000元+9800元+4000元=74200元。
正常工作工期=4+2+6+12+6+7+4=41天。



最短工期成本=8400元+1900元+14200元+27600元+5700元+10000元+6000元+12800元+5000元=91600元
最短工期=3+1+4+8+5+4+2=27天。



希赛

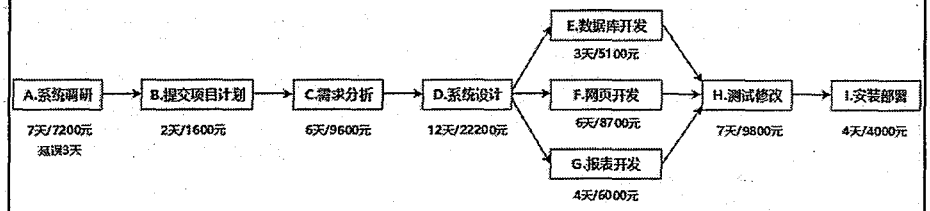
案例分析7

任务名称	正常工作	加班工作	可压缩天数	压缩1天增加费用
A.系统调研	4天/7200元	3天/8400元	1	1200元
B.提交项目计划	2天/1600元	1天/1900元	1	300元
C.需求分析	6天/9600元	4天/14200元	2	2300元
D.系统设计	12天/22200元	8天/27600元	4	1350元
E.数据库开发	3天/5100元	2天/5700元	1	600元
F.网页开发	6天/8700元	5天/10000元	1	1300元
G.报表开发	4天/6000元	任务外包无法赶工	0	
H.测试修改	7天/9800元	4天/12800元	3	1000元
I.安装部署	4天/4000元	2天/5000元	2	500元

希赛

案例分析7

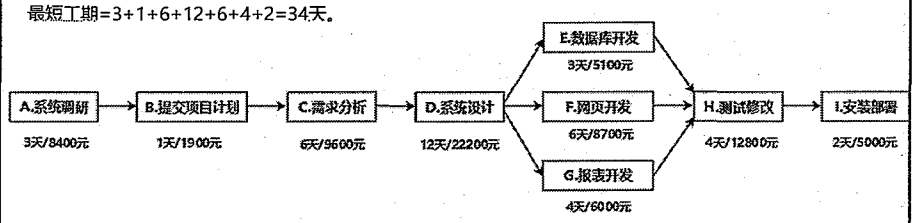
正常工作成本=7200元+1600元+9600元+22200元+5100元+8700元+6000元+9800元+4000元=74200元。
正常工作工期=4+3+2-1+6+12+6+7+4-2=41天。



希赛

案例分析7

最短工期成本=8400元+1900元+9600元+22200元+5100元+8700元+6000元+12800元+5000元=79700元
最短工期=3+1+6+12+6+4+2=34天。



希赛

案例分析7

【问题1】

（1）项目背景

（2）项目经理、项目经理的主管领导、客户方联系人、客户方的主管领导，项目领导小组（项目管理团队）和项目实施小组人员

（3）项目的总体技术解决方案

（4）所选择的项目管理过程及执行水平

（5）对这些过程的工具、技术和输入输出的描述

（6）选择的项目的生命周期和相关的项目阶段

（7）项目最终目标和阶段性目标

（8）进度计划

（9）项目预算

（10）变更流程和变更控制委员会

（11）对于内容、范围和时间的关键管理评审，以便于确定悬留问题和未决决策

【问题2】

正常工作成本=74200元。　　正常工作工期=41天。

最短工期成本=91600元。　　最短工期=27天。

【问题3】

在“B提交项目计划”和“I安装部署”任务中采用加班工作措施，以使得能够按照正常工作进度完成。

【问题4】

在“A系统调研”、“B提交项目计划”、“H测试修改”和“I安装部署”任务采用加班工作措施。项目所需费用79700元。

希赛

系统架构设计论文

希赛

一 考点分析 – 考试大纲

- 系统建模
- 软件架构设计
- 系统设计
- 分布式系统设计
- 系统可靠性分析与设计
- 系统安全性和保密性设计

希赛

二 历年考试情况

时间	考试题目	知识点说明
09年下半年	论基于DSSA的软件架构设计与应用	领域分析、领域设计、领域实现
	论信息系统建模方法	常见的信息系统建模方法及选择依据
	论基于REST服务的Web应用系统设计	REST的特点及其应用
	论软件可靠性设计与应用	可靠性的影响因素及可靠性设计技术
10年下半年	论软件的静态演化和动态演化及其应用	静态演化和动态演化的特点、优缺点及应用场合
	论数据挖掘技术的应用	数据挖掘的主要任务、数据挖掘的方法及实际应用效果
	论大规模分布式系统缓存设计策略	常见的缓存工作模式及应用场景
	论软件可靠性评价	可靠性模型及其选择依据

希赛

二 历年考试情况

时间	考试题目	知识点说明
11年下半年	论模型驱动架构在系统开发中的应用	模型驱动架构（MDA）的优缺点，及如何利用该方法进行相关开发工作
	论企业集成平台的架构设计	企业集成平台的基本功能和相关关键技术
	论企业架构管理与应用	企业架构管理所包含的内容及主要活动
	论软件需求获取技术及应用	需求获取技术以及每种技术的基本方法
12年下半年	论基于架构的软件设计方法及应用	要求了解ABSD的基本阶段
	论企业应用系统的数据持久层架构设计	每种数据访问模式的主要内容
	论决策支持系统的开发与应用	建立决策支持系统时需解决的一般关键问题
	论企业信息化规划的实施与应用	企业信息化规划目标及规划的具体内容

希赛

二 历年考试情况

时间	考试题目	知识点说明
13年下半年	论软件架构建模技术与应用	“4+1”视图模型的主要内容及其在项目中的应用
	论企业应用系统的分层架构风格	分层架构风格在项目中的应用
	论软件可靠性设计技术的应用	软件可靠性设计时遵循的基本原则、可靠性设计技术
	论分布式存储系统架构设计	分布式存储系统架构设计中所使用的分布式存储技术及其实现机制
14年下半年	论软件需求管理	需求管理过程中各个活动中的主要工作、如何进行软件需求管理
	论非功能性需求对企业应用架构设计的影响	企业应用架构设计中应该考虑哪些非功能性需求，这些非功能性需求是如何影响架构设计的
	论软件的可靠性设计	软件可靠性设计技术
	论网络安全体系设计	GB/T 9387.2-1995中的5大类安全服务，及每类安全服务的定义和主要实现手段

希赛

二 历年考试情况

时间	考试题目	知识点说明
15年下半年	论应用服务器基础软件	应用服务器在软件设计、开发、部署、运行和管理阶段，应该提供哪些核心功能
	论软件系统架构风格	常用的软件系统架构风格及每种风格的具体含义
	论面向服务的架构及其应用	SOA技术参考架构中都包含哪些服务类别及每类服务的定义和作用
	论企业集成平台的技术与应用	企业集成平台的基本功能及企业集成的关键技术
16年下半年	论软件系统架构评估	架构评估中关注的质量属性，采用了哪种评估方法实施过程与效果
	论软件设计模式及其应用	设计模式有哪几类，采用了哪些软件设计模式，具体实施效果
	论数据访问层设计技术及其应用	数据访问层设计技术包含的主要内容，采用了哪种数据访问层设计技术，实施过程与效果
	论微服务架构及其应用	微服务架构的特点，如何采用微服务架构模式的，遇到的实际问题和解决方案

希赛

二 历年考试情况

时间	考试题目	知识点说明
17年下半年	论软件系统建模方法及其应用	建模方法及特点与适用范围
	论软件架构风格	经典的架构风格及含义，架构选择的依据
	论无服务器架构及其应用	无服务器架构的特点
	论软件质量保证及其应用	质量保证的活动以活动的主要内容
18年下半年	论软件开发过程RUP及其应用	RUP在该项目中的具体实施内容
	论软件体系结构的演化	基于系统演化的6个步骤
	论面向服务架构设计及其应用	构建SOA架构时遇的问题，具体实施效果
	论NoSQL数据库技术及其应用	采用的NoSQL数据库技术的架构设计过程及其应用效果

希赛

注意事项

应用场景的合理性

技术方案的合理性

论文 = 心态 + 技巧 + 理论 + 实践

猜题

沟通的技巧
把问题表达清楚的技巧

希赛

注意事项

- 切记：不要猜题

  原则：要复用构件，不要整篇复用

- 不要抄范文，不要用网上的项目背景

  原则：宁可杜撰项目，也不用网上范文中的项目

- 练习时写文章前，一定做好功课

  原则：理论重于实践

- 做好准备工作之后，要勇于迈出第一步

  原则：万事开头难，迈出第一步，就不难了

- 换个视角看论文

  原则：以论文写作技巧视频为依据，对自己的论文进行自评

希赛

写作注意事项

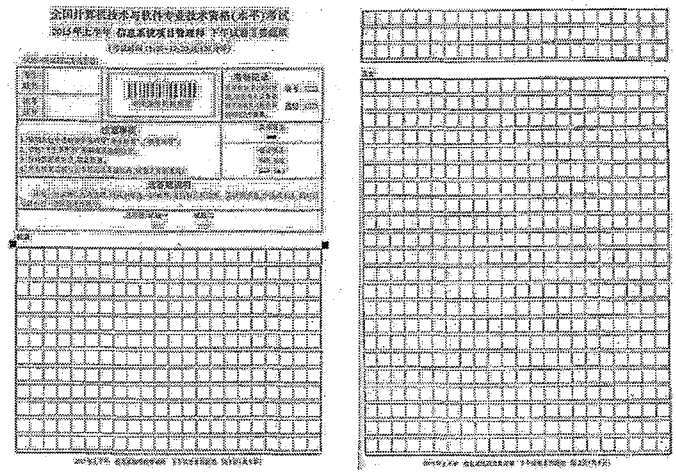
做好准备工作

- 加强学习
- 平时积累
- 共同提高
- 掌握写作技巧
- 提高写作速度
- 以不变应万变

希赛

写作注意事项

论文写作格式



希赛

如何解答试题

论文写作四部曲

- 找准核心论点（5分钟）
- 搭建论文框架（10分钟）
- 撰写摘要（15分钟）
- 正文写作（90分钟）

希赛

如何解答试题

论文解答步骤

试题：论系统设计中对用户需求的把握

对于系统工程师来说，在把某项工作系统化的时候，正确地理解该项工作的内容并设计出有效的系统，是一件最困难的事情。为了把用户的需求正确无误地反映到系统的规格说明中去，常规的作法是把系统的规格说明书和输出的报表交给用户征求意见。在某些情况下，还要做出系统的原型，请用户试用。

请围绕“系统设计中对用户需求的把握”论题，依次对以下三个问题进行论述。

1．叙述你参与的开发工程的概要，以及你所担任的工作。

2．就你所下过功夫的地方叙述，为了把用户需求反映到系统规格说明书中，采用过什么手段与用户进行通信？

3．对于你所采用的手段，各举出一点你认为有效果的方面和无效果的方面，简要叙述你对这方面的评价。

希赛

如何解答试题

论文解答步骤

找准核心论点

1、根据论文标题

2、分析题干问题

问题1要点：

✓ 软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

✓ 软件在系统分析方面的情况。

✓ “我”的角色和担任的主要工作。

问题2要点：

✓ 在系统分析过程中，把握用户需求的重要性。

✓ 为了把用户需求反映到系统规格说明书中，采用过什么手段与用户进行通信。

问题3要点：

✓ 采取的手段中有效果的手段，效果体现在什么地方。

✓ 采取的手段中无效果的手段，为什么没有效果。

✓ 还有哪些地方值得改进或提高。

希赛

如何解答试题

搭建论文框架

基本框架		内容	字数
摘要		项目相关背景及主要功能 你的岗位及主要职责 论文主体内容的总概 项目最终的实施效果或你的总结和感悟等	300字～320字
正文	（1）项目背景介绍	项目背景的详细介绍： 项目开发的原因 项目开始时间、实施周期 你的主要岗位职责等	400字左右
	（2）相关问题回应	非核心论点问题的回应 引出主体内容（核心论点）	300～400字
	（3）主体内容	采用总分式描述： “一总”加“三分” ”一对三”模式 可分4个段落	1000字～1500字
	（4）论文结论	结构上可分三步走： 先分析项目运行效果 再总结项目不足 最后提解决思路	400字～600字

希赛

如何解答试题

搭建论文框架

论文框架		内容	字数
摘要		摘要	300字～320字
正文	（1）项目背景介绍	系统概要： · 系统的概要 · 开发的体制和“我”担任的工作 · 在系统分析方面的情况	400字左右
	（2）相关问题回应	对需求的认识 把握用户需求的重要性	300～400字
	（3）主体内容	采用过的手段	1000字～1500字
	（4）论文结论	采取的手段中有效果的手段 效果体现在什么地方	200字左右
		采取的手段中无效果的手段 为什么没有效果	200字左右
		总结	100～200字

希赛

如何解答试题

撰写摘要

- 本文讨论……系统的……（论文主题）。该系统……（系统背景、简单功能介绍）。在本文中首先讨论了……（过程、方法、措施），最后……（不足之处/如何改进、特色之处）。在本系统的开发过程中，我担任了……（作者的工作角色）。

- 根据……需求（项目背景），我所在的……组织了……系统的开发。该系统……（系统背景、简单功能介绍）。在该系统的开发中，我担任了……（作者的工作角色）。我通过采取……（过程、方法、措施），使该系统开发工作圆满完成，得到了用户们的一致好评。但现在看来，……（不足之处/如何改进、特色之处）。

- ...年...月，我参加了……系统的开发，担任……（作者的工作角色）。该系统……（系统背景、简单功能介绍）。本文结合作者的实践，以……系统为例，讨论……（论文主题），包括……（过程、方法、措施）。

- ……是……（戴帽子，讲论文主题的重要性）。本文结合作者的实践，以……系统为例，讨论……（论文主题），包括……（过程、方法、措施）。 在本系统的开发过程中，我担任了……（作者的工作角色）。

希赛

论文写作方法

正文写作

- **以我为中心**
- **站在高级工程师的高度**
- **忠实于论点**
- **条理清晰，开门见山**
- **图文并茂，能收奇效**
- **标新立异，要有主见**
- **首尾一致**

希赛

论文写作方法

正文写作——常见问题及解决办法

- ✓ 走题
- ✓ 字数不够，字数偏多
- ✓ 摘要归纳欠妥
- ✓ 文章深度不够，缺少特色，泛泛而谈
- ✓ 文章口语化太重，文字表达能力太差
- ✓ 文章缺乏主题项目，项目年代久远
- ✓ 整篇文章从大一二三到小123，给人以压抑感
- ✓ 文章结构不够清晰，段落太长

希赛

论文评分标准

扣分与加分准则

下述情况的论文，需要适当扣5分到10分：

- ✓ 没有写论文摘要、摘要过于简略、或者摘要中没有实质性内容的论文
- ✓ 字迹比较潦草、其中有不少字难以辨认的论文
- ✓ 确实属于过分自我吹嘘或自我标榜、夸大其词的论文
- ✓ 内容有明显错误和漏洞的，按同一类错误每一类扣一次分
- ✓ 内容仅属于大学生或研究生实习性质的项目、并且其实际应用水平相对较低的论文

下述情况，可考虑适当加分（可考虑加5分到10分）：

- ✓ 有独特的见解或者有着很深入的体会、相对非常突出的论文
- ✓ 起点很高，确实符合当今信息系统发展的新趋势与新动向，并能加以应用的论文
- ✓ 内容详实、体会中肯、思路清晰、非常切合实际的很优秀的论文
- ✓ 项目难度很高，或者项目完成的质量优异，或者项目涉及国家重大信息系统工程且作者本人参加并发挥重要作用、并且能正确按照试题要求论述的论文

希赛

论文评分标准

雷区

下述情况之一的论文，不能给予及格分数：

× 虚构情节、文章中有较严重的不真实的或者不可信的内容出现的论文

× 没有项目开发的实际经验、通篇都是浅层次纯理论的论文

× 所讨论的内容与方法过于陈旧、或者项目的水准非常低下的论文

× 内容不切题意，或者内容相对很空洞、基本上是泛泛而谈且没有较深入体会的论文

× 正文与摘要的篇幅过于短小的论文（如正文少于1200字）

× 文理很不通顺、错别字很多、条理与思路不清晰、字迹过于潦草等情况相对严重的论文

企业信息化战略与实施

● 商业智能是指利用数据挖掘、知识发现等技术分析和挖掘结构化的、面向特定领域的存储与数据仓库的信息。它可以帮助用户认清发展趋势、获取决策支持并得出结论。

以下（1）活动，并不属于商业智能范畴。

（1）A．某大型企业通过对产品销售数据进行挖掘，分析客户购买偏好

B．某大型企业查询数据仓库中某种产品的总体销售数量

C．某大型购物网站通过分析用户的购买历史记录，为客户进行商品推荐

D．某银行通过分析大量股票交易的历史数据，做出投资决策

● 企业应用集成通过采用多种集成模式构建统一标准的基础平台，将具有不同功能和目的且独立运行的企业信息系统联合起来。其中，面向（2）的集成模式强调处理不同应用系统之间的交互逻辑，与核心业务逻辑相分离，并通过不同应用系统之间的协作共同完成某项业务功能。

（2）A．数据　　B．接口　　C．过程　　D．界面

● 某大型公司欲开发一个门户系统，该系统以商业流程和企业应用为核心，将商业流程中不同的功能模块通过门户集成在一起，以提高公司的集中贸易能力、协同能力和信息管理能力。根据这种需求，采用企业（4）门户解决方案最为合适。

（4）A．信息　　B．知识　　C．应用　　D．垂直

● 客户关系管理（CRM）系统将市场营销的科学管理理念通过信息技术的手段集成在软件上，能够帮助企业构建良好的客户关系。以下关于 CRM 系统的叙述中，错误的是（5）。

（5）A．销售自动化是 CRM 系统中最基本的模块

B．营销自动化作为销售自动化的补充，包括营销计划的编制和执行、计划结果分析等

C．CRM 系统能够与 ERP 系统在财务、制造、库存等环节进行连接，但两者关系相对松散，一般不会形成闭环结构

D．客户服务与支持是 CRM 系统的重要功能。目前，客户服务与支持的主要手段是通过呼叫中心和互联网来实现

● 共享数据库是一种重要的企业应用集成方式。以下关于共享数据库集成方式的叙述中，错误的是（6）。

（6）A．共享数据库集成方式通常将应用程序的数据存储在一个共享数据库中，通过制定统一的数据库模式来处理不同应用的集成需求

B．共享数据库为不同的应用程序提供了统一的数据存储与格式定义，能够解决不同应用程序中数据语义不一致的问题

C．多个应用程序可能通过共享数据库频繁地读取和修改相同的数据，这会使共享数据库成为一个性能瓶颈

D．共享数据库集成方式的一个重要限制来自外部的已封装应用，这些封装好的应用程序只能采用自己定义的数据库模式，调整和集成余地较小

● 某大型商业公司欲集成其内部的多个业务系统，这些业务系统的运行平台和开发语言差异较大，而且系统所使用的通信协议和数据格式各不相同，针对这种情况，采用基于（7）的集成框架较为合适。除此以

外，集成系统还需要根据公司的新业务需要，灵活、动态地定制系统之间的功能协作关系，针对这一需求，应该选择基于（8）技术的实现方式更为合适。

（7）A．数据库　　B．文件系统　　C．总线　　D．点对点

（8）A．分布式对象　　B．远程过程调用　　C．进程间通信　　D．工作流

● 企业战略数据模型可分为两种类型：（9）描述日常事务处理中的数据及其关系；（10）描述企业管理决策者所需信息及其关系。

（9）A．元数据模型　　B．数据库模型　　C．数据仓库模型　　D．组织架构模型

（10）A．元数据模型　　B．数据库模型　　C．数据仓库模型　　D．组织架构模型

● 运用信息技术进行知识的挖掘和（11）的管理是企业信息化建设的重要活动。

（11）A．业务流程　　B．IT 基础设施　　C．数据架构　　D．规章制度

● 以下关于企业信息化方法的叙述中，正确的是（12）。

（12）A．业务流程重构是对企业的组织结构和工作方法进行重新设计，SCM（供应链管理）是一种重要的实现手段

B．在业务数量浩繁且流程错综复杂的大型企业里，主题数据库方法往往形成许多“信息孤岛”，造成大量的无效或低效投资

C．人力资源管理把企业的部分优秀员工看作是一种资本，能够取得投资收益

D．围绕核心业务应用计算机和网络技术是企业信息化建设的有效途径

● 企业信息资源集成管理的前提是对企业（13）的集成，其核心是对企业（14）的集成。

（13）A．信息功能　　B．信息设施　　C．信息活动　　D．信息处理

（14）A．业务流　　B．内部信息流　　C．外部信息流　　D．内部和外部信息流

● 为了加强对企业信息资源的管理，企业应按照信息化和现代化企业管理要求设置信息管理机构，建立信息中心。信息中心的主要职能不包括（15）。

（15）A．处理信息，确定信息处理的方法

B．用先进的信息技术提高业务管理水平

C．组织招聘信息资源管理员

D．建立业务部门期望的信息系统和网络

● 企业信息化涉及到对企业管理理念的创新，按照市场发展的要求，对企业现有的管理流程重新整合，管理核心从对（16）的管理，转向对（17）的管理，并延伸到对企业技术创新、工艺设计、产品设计、生产制造过程的管理，进而还要扩展到对（18）的管理乃至发展到电子商务。

（16）A．人力资源和物资　　B．信息技术和知识

C．财务和物料　　D．业务流程和数据

（17）A．业务流程和数据　　B．企业信息系统和技术

C．业务流程、数据和接口　　D．技术、物资和人力资源

（18）A．客户关系和供应链　　B．信息技术和知识

C．生产技术和信息技术　　D．信息采集、存储和共享

● 企业信息集成按照组织范围分为企业内部的信息集成和外部的信息集成。在企业内部的信息集成中，（19）实现了不同系统之间的互操作，使得不同系统之间能够实现数据和方法的共享；（20）实现了不同应用系统之间的连接、协调运作和信息共享。

（19）A．技术平台集成　　B．数据集成
　　C．应用系统集成　　D．业务过程集成
（20）A．技术平台集成　　B．数据集成
　　C．应用系统集成　　D．业务过程集成

● 企业信息化一定要建立在企业战略规划基础之上，以企业战略规划为基础建立的企业管理模式是建立（21）的依据。
（21）A．企业战略数据模型　　B．企业业务运作模型
　　C．企业信息系统架构　　D．企业决策支持

● 集成平台是支持企业信息集成的支撑环境，包括硬件、软件、软件工具和系统。集成平台的基本功能中，（22）实现不同数据库系统之间的数据交换、互操作、分布数据管理和共享信息模型定义；（23）能够为应用提供数据交换和访问操作，使各种不同的系统能够相互协作。
（22）A．数据通信服务　　B．信息集成服务
　　C．应用集成服务　　D．操作集成服务
（23）A．数据通信服务　　B．信息集成服务
　　C．应用集成服务　　D．操作集成服务

● 供应链中的信息流覆盖了从供应商、制造商到分销商，再到零售商等供应链中的所有环节，其信息流分为需求信息流和供应信息流，（24）属于需求信息流，（25）属于供应信息流。
（24）A．库存记录　　B．生产计划　　C．商品入库单　　D．提货发运单
（25）A．客户订单　　B．采购合同　　C．完工报告单　　D．销售报告

● 商业智能系统的处理过程包括四个主要阶段：数据预处理通过（26）实现企业原始数据的初步整合；建立数据仓库是后续数据处理的基础；数据分析是体现系统智能的关键，主要采用（27）和（28）技术，前者能够实现数据的上卷、下钻和旋转分析；后者利用隐藏的知识，通过建立分析模型预测企业未来发展趋势；数据展现主要完成数据处理结果的可化。
（26）A．数据映射和关联　　B．数据集市和数据立方体
　　C．数据抽取、转换和装载　　D．数据清洗和数据集成
（27）A．知识库　　B．数据挖掘　　C．联机事务处理　　D．联机分析处理
（28）A．知识库　　B．数据挖掘　　C．联机事务处理　　D．联机分析处理

软件工程

● 用户文档主要描述所交付系统的功能和使用方法。下列文档中，（1）属于用户文档。

（1）A．需求说明书　　B．系统设计文档
　　C．安装文档　　D．系统测试计划

● 用例（use case）用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。在一个“订单输入子系统”中，创建新订单和更新订单都需要核查用户帐号是否正确。用例“创建新订单”、“更新订单”与用例“核查客户帐号”之间是（2）关系。

（2）A．包含（include）　　B．扩展（extend）
　　C．分类（classification）　　D．聚集（aggregation）

● 面向对象的设计模型包含以（3）表示的软件体系结构图，以（4）表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用以描述流程化处理的活动图等。

（3）A．部署图　　B．包图　　C．协同图　　D．交互图
（4）A．部署图　　B．包图　　C．协同图　　D．交互图

● 软件的横向重用是指重用不同应用领域中的软件元素。（5）是一种典型的、原始的横向重用机制。

（5）A．对象　　B．构件　　C．标准函数库　　D．设计模式

● 下列关于不同软件开发方法所使用的模型的描述中，正确的是（6）。

（6）A．在进行结构化分析时，必须使用数据流图和软件结构图这两种模型
　　B．采用面向对象开发方法时，可以使用状态图和活动图对系统的动态行为进行建模
　　C．实体联系图（E-R 图）是在数据库逻辑结构设计时才开始创建的模型
　　D．UML 的活动图与程序流程图的表达能力等价

● 下列关于软件需求管理或需求开发的叙述中，正确的是（7）。

（7）A．所谓需求管理是指对需求开发的管理
　　B．需求管理包括：需求获取、需求分析、需求定义和需求验证
　　C．需求开发是将用户需求转化为应用系统成果的过程
　　D．在需求管理中，要求维持对用户原始需求和所有产品构件需求的双向跟踪

● 在面向对象设计中，用于描述目标软件与外部环境之间交互的类被称为（8），它可以（9）。

（8）A．实体类　　B．边界类　　C．模型类　　D．控制类
（9）A．表示目标软件系统中具有持久意义的信息项及其操作
　　B．协调、控制其他类完成用例规定的功能或行为
　　C．实现目标软件系统与外部系统或外部设备之间的信息交流和互操作
　　D．分解任务并把子任务分派给适当的辅助类

● 最少知识原则（也称为迪米特法则）是面向对象设计原则之一，指一个软件实体应当尽可能少地与其他实体发生相互作用。这样，当一个实体被修改时，就会尽可能少地影响其他的实体。下列叙述中，“（10）”不符最少知识原则。

（10）A．在类的划分上，应当尽量创建松耦合的类
B．在类的设计上，只要有可能，一个类型应当设计成不变类
C．在类的结构设计上，每个类都应当尽可能提高对其属性和方法的访问权限
D．在对其他类的引用上，一个对象对其他对象的引用应当降到最低

• 快速迭代式的原型开发能够有效控制成本，（11）是指在开发过程中逐步改进和细化原型直至产生出目标系统。
（11）A．可视化原型开发　　B．抛弃式原型开发
C．演化式原型开发　　D．增量式原型开发

• 确认测试主要用于验证软件的功能、性能和其他特性是否与用户需求一致。下述各种测试中，（12）为确认测试。
（12）A．负载测试和压力测试　　B．α测试和β测试
C．随机测试和功能测试　　D．可靠性测试和性能测试

• 软件（13）是指改正产生于系统开发阶段而在系统测试阶段尚未发现的错误。
（13）A．完善性维护　B．适应性维护　C．正确性维护　D．预防性维护

• 某软件公司正在设计一个图像处理软件，该软件需要支持用户在图像处理过程中的撤销和重做等动作，为了实现该功能，采用（14）最为合适。
（14）A．单例模式　B．命令模式　C．访问者模式　D．适配器模式

• 螺旋模型将整个软件开发过程分为多个阶段，每个阶段都由目标设定、（15）、开发和有效性验证以及评审4个部分组成。
（15）A．需求分析　B．风险分析　C．系统设计　D．架构设计

• 快速应用开发（Rapid Application Development，RAD）通过使用基于（16）的开发方法获得快速开发。当（17）时，最适合于采用RAD方法。
（16）A．用例　B．数据结构　C．剧情　D．构件
（17）A．一个新系统要采用很多新技术
B．新系统与现有系统有较高的互操作性
C．系统模块化程度较高
D．用户不能很好地参与到需求分析中

• 以下关于软件开发方法的叙述，错误的是（18）。
（18）A．对于较为复杂的应用问题，适合采用形式化方法进行需求分析
B．形式化方法的优势在于能够精确地表述和研究应用问题及其软件实现
C．净室软件工程将正确性验证作为发现和排除错误的主要机制
D．净室软件工程强调统计质量控制技术，包括对客户软件使用预期的测试

● 对于违反里氏替换原则的两个类 A 和 B，可以采用的候选解决方案中，正确的是（19）。

（19）A．尽量将一些需要扩展的类或者存在变化的类设计为抽象类或者接口，并将其作为基类，在程序中尽量使用基类对象进行编程

B．创建一个新的抽象类 C，作为两个具体类的超类，将 A 和 B 共同的行为移动到 C 中，从而解决 A 和 B 行为不完全一致的问题

C．将 B 到 A 的继承关系改成组合关系

D．区分是"Is-a"还是"Has-a"。如果是 Is-a，可以使用继承关系，如果是 Has-a，应该改成组合或聚合关系

● 某软件公司欲设计一款图像处理软件，帮助用户对拍摄的照片进行后期处理。在软件需求分析阶段，公司的系统分析师识别出了如下 3 个关键需求：

图像处理软件需要记录用户在处理照片时所有动作，并能够支持用户动作的撤销与重做等行为。

图像处理软件需要根据当前正在处理的照片的不同特征选择合适的处理操作，处理操作与照片特征之间具有较为复杂的逻辑关系。

图像处理软件需要封装各种图像处理算法，用户能够根据需要灵活选择合适的处理算法；软件还要支持高级用户根据一定的规则添加自定义处理算法。

在系统设计阶段，公司的架构师决定采用设计模式满足上述关键需求中对系统灵活性与扩展性的要求。具体来说，为了支持灵活的撤销与重做等行为，采用（20）最为合适；为了封装图像操作与照片特征之间的复杂逻辑关系，采用（21）最为合适；为了实现图像处理算法的灵活选择与替换，采用（22）最为合适。

（20）A．工厂模式　　B．责任链模式　　C．中介者模式　　D．命令模式

（21）A．状态模式　　B．适配器模式　　C．组合模式　　D．单例模式

（22）A．模板方法模式　　B．访问者模式　　C．策略模式　　D．观察者模式

● 逆向工程导出的信息可以分为实现级、结构级、功能级和领域级四个抽象层次。程序的抽象语法树属于（23）；反映程序分量之间相互依赖关系的信息属于（24）。

（23）A．实现级　　B．结构级　　C．功能级　　D．领域级

（24）A．实现级　　B．结构级　　C．功能级　　D．领域级

● 某系统中的文本显示类（TextView）和图片显示类（PictureView）都继承了组件类（Component），分别显示文本和图片内容，现需要构造带有滚动条或者带有黑色边框，或者既有滚动条又有黑色边框的文本显示控件和图片显示控件，但希望最多只增加 3 个类。

那么采用设计模式（25）可实现该需求，其优点是（26）。

（25）A．外观　　B．单体　　C．装饰　　D．模板方法

（26）A．比静态继承具有更大的灵活性　　B．提高已有功能的重复使用性

C．可以将接口与实现相分离　　D．为复杂系统提供了简单接口

● 以下关于自顶向下开发方法的叙述中，正确的是（27）。

（27）A．自顶向下过程因为单元测试而比较耗费时间

B．自顶向下过程可以更快地发现系统性能方面的问题

C．相对于自底向上方法，自顶向下方法可以更快地得到系统的演示原型

D．在自顶向下的设计中，如发现了一个错误，通常是因为底层模块没有满足其规格说明（因为高层模块已经被测试过了）

● 以下关于面向对象软件测试的叙述中，正确的是（28）。

（28）A．在测试一个类时，只要对该类的每个成员方法都进行充分的测试就完成了对该类充分的测试
B．存在多态的情况下，为了达到较高的测试充分性，应对所有可能的绑定都进行测试
C．假设类 B 是类 A 的子类，如果类 A 已经进行了充分的测试，那么在测试类 B 时不必测试任何类 B 继承自类 A 的成员方法
D．对于一棵继承树上的多个类，只有处于叶子节点的类需要测试

• 系统建议方案中不应该包含的内容是（29）。
（29）A．问题陈述　　B．项目范围
C．候选方案及其可行性分析　　D．系统详细设计方案

• 下列关于联合需求计划(Joint Requirement Planning, JRP)的叙述中，不正确的是（30）。
（30）A．在 JRP 实施之前，应制定详细的议程，并严格遵照议程进行
B．在讨论期间尽量避免使用专业术语
C．JRP 是一种相对来说成本较高但十分有效的需求获取方法
D．JRP 的主要目的是对需求进行分析和验证

• 在结构化分析方法中，用（31）表示功能模型，用（32）表示行为模型。
（31）A．ER 图　　B．用例图　　C．DFD　　D．对象图
（32）A．通信图　　B．顺序图　　C．活动图　　D．状态转换图

• 下列关于用户界面设计的叙述中，错误的是（33）。
（33）A．界面交互模型应经常进行修改
B．界面的视觉布局应该尽量与真实世界保持一致
C．所有可视信息的组织需要按照统一的设计标准
D．确保用户界面操作和使用的一致性

• 一组对象以定义良好但是复杂的方式进行通信，产生的相互依赖关系结构混乱且难以理解。采用（34）模式，用一个特定对象来封装一系列的对象交互，从而使各对象不需要显式地相互引用，使其耦合松散，而且可以独立地改变它们之间的交互。
（34）A．解释器(Interpreter)　　B．策略(Strategy)
C．中介者(Mediator)　　D．迭代器(Iterator)

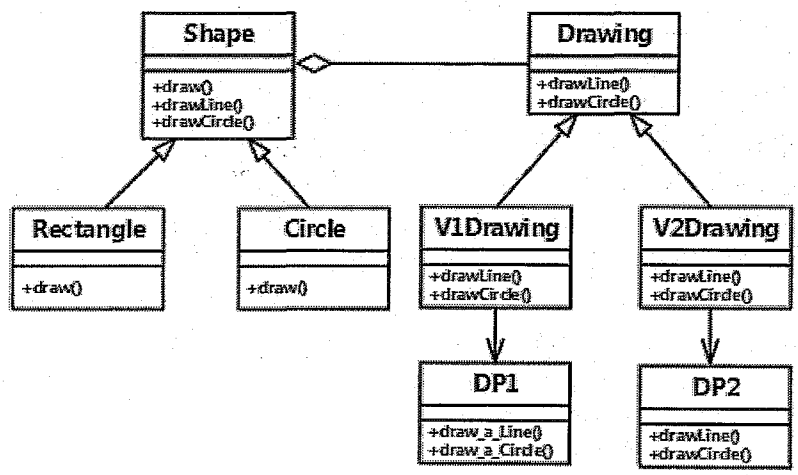
• 在 UML 提供的系统视图中，（35）是逻辑视图的一次执行实例，描述了并发与同步结构；（36）是最基本的需求分析模型。
（35）A．进程视图　　B．实现视图　　C．部署视图　　D．用例视图
（36）A．进程视图　　B．实现视图　　C．部署视图　　D．用例视图

• 下列关于软件调试与软件测试的叙述中，正确的是（37）。
（37）A．软件测试的目的是找出存在的错误，软件调试的目的是定位并修正错误
B．软件测试的结束过程不可预计，软件调试使用预先定义的过程
C．软件调试的过程可以实现设计
D．软件测试不能描述过程或持续时间

● 用例（use case）用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。在一个会员管理系统中，会员注册时可以采用电话和邮件两种方式。用例“会员注册”和“电话注册”、“邮件注册”之间是（38）关系。

（38）A．包含(include)　　B．扩展(extend)　　C．泛化(generalize)　　D．依赖(depends on)

● 某软件公司欲开发一个绘图软件，要求使用不同的绘图程序绘制不同的图形。在明确用户需求后，该公司的架构师决定采用 Bridge 模式实现该软件，并设计 UML 类图如下图所示。图中与 Bridge 模式中的“Abstraction”角色相对应的类是（39），与“Implementor”角色相对应的类是（40）。



（39）A．Shape　　B．Drawing　　C．Rectangle　　D．V2Drawing

（40）A．Shape　　B．Drawing　　C．Rectangle　　D．V2Drawing

● RUP 强调采用（41）的方式来开发软件，这样做的好处是（42）。

（41）A．原型和螺旋　　B．螺旋和增量　　C．迭代和增量　　D．快速和迭代

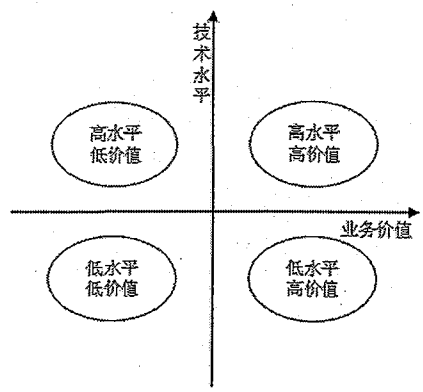
（42）A．在软件开发的早期就可以对关键的，影响大的风险进行处理

B．可以避免需求的变更

C．能够非常快速地实现系统的所有需求

D．能够更好地控制软件的质量

● 对于遗留系统的评价框架如下图所示，那么处于“高水平、低价值”区的遗留系统适合于采用的演化策略为（43）。



（43）A．淘汰　　B．继承　　C．改造　　D．集成

• （44）的目的是检查模块之间，以及模块和已集成的软件之间的接口关系，并验证已集成的软件是否符合设计要求。其测试的技术依据是（45）。

（44）A．单元测试　　B．集成测试　　C．系统测试　　D．回归测试

（45）A．软件详细设计说明书　　B．技术开发合同

C．软件概要设计文档　　D．软件配置文档

软件架构设计(第一部分)

● 在一个典型的基于 MVC（Model-View-Controller）的 J2EE 应用中，分发客户请求、有效组织其它构件为客户端提供服务的控制器由（1）实现。

（1）A．Entity Bean　　B．Session Bean　　C．Servlet　　D．JSP

● 架构复审是基于架构开发中一个重要的环节。以下关于架构复审的叙述中，错误的是（2）。

（2）A．架构复审的目标是标识潜在的风险，及早发现架构设计的缺陷和错误

B．架构复审过程中，通常会对一个可运行的最小化系统进行架构评估和测试

C．架构复审人员由系统设计与开发人员组成

D．架构设计、文档化和复审是一个迭代的过程

● 某公司欲开发一个基于图形用户界面的集成调试器。该调试器的编辑器和变量监视器可以设置调试断点。当调试器在断点处暂停运行时，编辑程序可以自动卷屏到断点，变量监视器刷新变量数值。针一对这样的功能描述，采用（3）的架构风格最为合适。

（3）A．数据共享　　B．虚拟机　　C．隐式调用　　D．显式调用

● 某公司欲开发一种工业机器人，用来进行汽车零件的装配。公司的架构师经过分析与讨论，给出了该机器人控制软件的两种候选架构方案：闭环控制和分层结构。以下对于这两种候选架构的选择理由，错误的是（4）。

（4）A．应该采用闭环控制架构，因为闭环结构给出了将软件分解成几个协作构件的方法，这对于复杂任务特别适合

B．应该采用闭环控制结构，因为闭环控制架构中机器人的主要构件监控器、传感器、发动机等）是彼此分开的，并能够独立替换

C．应该采用分层结构，因为分层结构很好地组织了用来协调机器人操作的构件，系统结构更加清晰

D．应该采用分层结构，因为抽象层的存在，满足了处理不确定性的需要：在较低层次不确定的实现细节在较高层次会变得确定

● Architecture Tradeoff Analysis Method (ATAM）是一种软件架构的评估方法，以下关于该方法的叙述中，正确的是（5）。

（5）A．ATAM 是一种代码评估方法　　B．ATAM 需要评估软件的需求是否准确

C．ATAM 需要对软件系统进行测试　　D．ATAM 不是一种精确的评估工具

● 在 RUP 中采用“4+1”视图模型来描述软件系统的体系结构。在该模型中，最终用户侧重于（6），系统工程师侧重于（7）。

（6）A．实现视图　　B．进程视图　　C．逻辑视图　　D．部署视图

（7）A．实现视图　　B．进程视图　　C．逻辑视图　　D．部署视图

● 软件架构是降低成本、改进质量、按时和按需交付产品的关键因素。以下关于软件架构的描述，错误的是（8）。

（8）A．根据用户需求，能够确定一个最佳的软件架构，指导整个软件的开发过程
B．软件架构设计需要满足系统的质量属性，如性能、安全性和可修改性等
C．软件架构设计需要确定组件之间的依赖关系，支持项目计划和管理活动
D．软件架构能够指导设计入员和实现人员的工作

● 基于软件架构的设计（Architecture Based Software Development，ABSD）强调由商业、质量和功能需求的组合驱动软件架构设计。它强调采用（9）来描述软件架构，采用（10）来描述需求。

（9）A．类图和序列图　B．视角与视图　C．构件和类图　D．构件与功能
（10）A．用例与类图　B．用例与视角　C．用例与质量场景　D．视角与质量场景

● （11）描述了一类软件架构的特征，它独立于实际问题，强调软件系统中通用的组织结构选择。垃圾回收机制是 Java 语言管理内存资源时常用的一种（12）。

（11）A．架构风格　B．开发方法　C．设计模式　D．分析模式
（12）A．架构风格　B．开发方法　C．设计模式　D．分析模式

● 某企业内部现有的主要业务功能已经封装为 Web 服务。为了拓展业务范围，需要将现有的业务功能进行多种组合，形成新的业务功能。针对业务灵活组合这一要求，采用（13）架构风格最为合适。

（13）A．管道-过滤器　B．解释器　C．显式调用　D．黑板

● 识别风险点、非风险点、敏感点和权衡点是软件架构评估过程中的关键步骤。针对某系统所作的架构设计中，“系统需要支持的最大并发用户数量直接影响传输协议和数据格式”描述了系统架构设计中的一个（14）；“由于系统的业务逻辑目前尚不清楚，因此现有系统三层架构中的第二层可能会出现功能重复，这会影响系统的可修改性”描述了系统架构设计中的一个（15）。

（14）A．敏感点　B．风险点　C．非风险点　D．权衡点
（15）A．敏感点　B．风险点　C．非风险点　D．权衡点

● 以下关于软件中间件的叙述，错误的是（16）。

（16）A．中间件通过标准接口实现与应用程序的关联，提供特定功能的服务
B．使用中间件可以提高应用软件可移植性
C．使用中间件将增加应用软件设计的复杂度
D．使用中间件有助于提高开发效率

● 以下叙述，（17）不是软件架构的主要作用。

（17）A．在设计变更相对容易的阶段，考虑系统结构的可选方案
B．便于技术人员与非技术人员就软件设计进行交互
C．展现软件的结构、属性与内部交互关系
D．表达系统是否满足用户的功能性需求

● 某公司欲开发一个网上商城系统，在架构设计阶段，公司的架构师识别出 3 个核心质量属性场景，其中“系统主站断电后，能够在 2 分钟内自动切换到备用站点，并恢复正常运行”主要与（18）质量属性相关，通常可采用（19）架构策略实现该属性；“在并发用户数不超过 1000 人时，用户的交易请求应该在 0.5s 内完

成”主要与（20）质量属性相关通常可采用（21）架构策略实现该属性；“系统应该能够抵挡恶意用户的入侵行为，并进行报警和记录”主要与（22）质量属性相关，通常可采用（23）架构策略实现该属性。

（18）A．性能　B．可用性　C．易用性　D．可修改性
（19）A．主动冗余　B．信息隐藏　C．抽象接口　D．记录/回放
（20）A．可测试性　B．易用性　C．性能　D．互操作性
（21）A．操作串行化　B．资源调度　C．心跳　D．内置监控器
（22）A．可用性　B．安全性　C．可测试性　D．可修改性
（23）A．内置监控器　B．记录/回放　C．追踪审计　D．维护现有接口

● 将系统需求模型转换为架构模型是软件系统需求分析阶段的一项重要工作，以下描述中，（24）是在转换过程中需要关注的问题。

（24）A．如何通过多视图模型描述软件系统的架构
B．如何确定架构模型中有哪些元素构成
C．如何采用表格或用例映射保证转换的可追踪性。
D．如何通过模型转换技术，将高层架构模型逐步细化为细粒度架构模型

● 某公司拟开发一个 VIP 管理系统，系统需要根据不同商场活动，不定期更新 VIP 会员的审核标准和 VIP 折扣系统。针对上述需求，采用（25）架构风格最为合适。

（25）A．规则系统　B．过程控制　C．分层　D．管道-过滤器

● 某公司拟开发一个地面清洁机器人。机器人的控制者首先定义清洁任务和任务之间的关系，机器人接受任务后，需要响应外界环境中触发的一些突发事件，根据自身状态进行动态调整，最终自动完成任务。针对上述需求，该机器人应该采用（26）架构风格最为合适。

（26）A．面向对象　B．主程序-子程序　C．规则系统　D．管道-过滤器

● 某公司拟开发一个语音识别系统，其语音识别的主要过程包括分割原始语音信号、识别音素、产生候选词、判定语法片断、提供语义解释等，每个过程都需要进行基于先验知识的条件判断并进行相应的识别动作。针对该系统的特点，采用（27）架构风格最为合适。

（27）A．解释器　B．面向对象　C．黑板　D．隐式调用

● 特定领域软件架构（Domain Specific Software Architecture, DSSA）以一个特定问题领域为对象，形成由领域参考模型，参考需求，（28）等组成的开发基础架构，支持一个特定领域中多个应用的生成。DSSA 的基本活动包括领域分析、领域设计和领域实现。其中领域分析的主要目的是获得（29），从而描述领域中系统之间共同的需求，即领域需求；领域设计的主要目标是获得（30），从而描述领域模型中表示需求的解决方案；领域实现的主要目标是开发和组织可重用信息，并实现基础软件架构。

（28）A．参考设计　B．参考规约　C．参考架构　D．参考实现
（29）A．领域边界　B．领域信息　C．领域对象　D．领域模型
（30）A．特点领域软件需求　B．特定领域软件架构
C．特定领域软件设计模型　D．特定领域软件重用模型

软件架构设计(第二部分)

● 软件架构文档是对软件架构的正式描述，能够帮助与系统有关的开发人员更好地理解软件架构。软件架构文档的写作应该遵循一定的原则。以下关于软件架构文档写作原则的叙述中，错误的是（1）。

（1）A．架构文档应该从架构设计者的角度进行编写
B．应该保持架构文档的即时更新，但更新不要过于频繁
C．架构文档中的描述应该尽量避免不必要的重复
D．每次架构文档修改，都应该记录修改的原则

● Windows 操作系统在图形用户界面处理方面采用的核心架构风格是（2）风格。Java 语言宣传的“一次编写，到处运行”的特性，从架构风格上看符合（3）风格的特点。

（2）A．虚拟机　B．管道-过滤器　C．事件驱动　D．微内核-扩展
（3）A．虚拟机　B．管道-过滤器　C．事件驱动　D．微内核-扩展

● （4）的选择是开发一个软件系统时的基本设计决策；（5）是最低层的模式，关注软件系统的设计与实现，描述了如何实现构件及构件之间的关系。引用一计数是 C++管理动态资源时常用的一种（6）。

（4）A．架构模式　B．惯用法　C．设计模式　D．分析模式
（5）A．架构模式　B．惯用法　C．设计模式　D．分析模式
（6）A．架构模式　B．惯用法　C．设计模式　D．分析模式

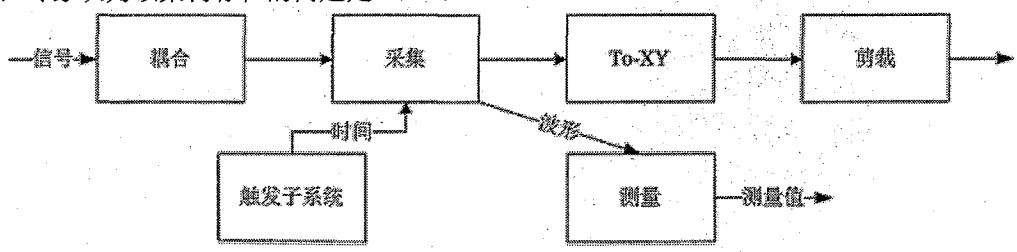
● 分布式系统开发中，通常需要将任务分配到不同的逻辑计算层。业务数据的综合计算分析任务属于（7）。

（7）A．表示逻辑层　B．应用逻辑层　C．数据处理层　D．数据层

● 软件架构设计包括提出架构模型、产生架构设计和进行设计评审等活动，是一个迭代的过程。以下关于软件架构设计活动的描述，错误的是（8）。

（8）A．在建立软件架构的初期，一般需要选择一个合适的架构风格
B．将架构分析阶段已标识的构件映射到架构中，并分析这些构件之间的关系
C．软件架构设计活动将已标识构件集成到软件架构中，设计并实现这些构件
D．一旦得到了详细的软件架构设计，需要邀请独立于系统开发的外部人员对系统进行评审

● 某公司欲为某种型号的示波器开发内置软件。该公司的架构师设计了如下图所示的软件架构。在软件架构评审时，专家认为该架构存在的问题是（9）。



（9）A．在功能划分上将各个模块独立起来
B．在硬件构件的混合和替换方面不是很灵活
C．没有清晰地说明用户怎样与其交互

D．没有明确的层次关系，没有强调功能之间的交互

• 1995 年 Kruchten 提出了著名的“4+1”视图，用来描述软件系统的架构。在“4+1”视图中，（10）用来描述设计的对象模型和对象之间的关系；（11）描述了软件模块的组织与管理；（12）描述设计的并发和同步特征。

（10）A．逻辑视图　B．用例视图　C．过程视图　D．开发视图
（11）A．逻辑视图　B．用例视图　C．过程视图　D．开发视图
（12）A．逻辑视图　B．用例视图　C．过程视图　D．开发视图

• 架构权衡分析方法（ATAM）是一种常用的软件架构评估方法，下列关于该方法的叙述中，正确的是（13）。

（13）A．ATAM 需要对代码的质量进行评估
B．ATAM 需要对软件系统需求的正确性进行评价
C．ATAM 需要对软件系统进行集成测试
D．ATAM 需要对软件质量属性进行优先级排序

• 某公司采用基于架构的软件设计（Architecture-Based Software Design，ABSD）方法进行软件设计与开发。ABSD 方法有三个基础，分别是对系统进行功能分解、采用（14）实现质量属性与商业需求、采用软件模板设计软件结构。

ABSD 方法主要包括架构需求等 6 个主要活动，其中（15）活动的目标是标识潜在的风险，及早发现架构设计中的缺陷和错误；（16）活动针对用户的需求变化，修改应用架构，满足新的需求。

小王是该公司的一位新任架构师，在某项目中主要负责架构文档化方面的工作。小王（17）的做法不符合架构文档化的原则。架构文档化的主要输出结果是架构规格说明书和（18）。

（14）A．架构风格　B．设计模式　C．架构策略　D．架构描述
（15）A．架构设计　B．架构实现　C．架构复审　D．架构演化
（16）A．架构设计　B．架构实现　C．架构复审　D．架构演化
（17）A．从使用者的角度书写文档　B．随时保证文档都是最新的
C．将文档分发给相关人员　D．针对不同背景的人员书写文档的方式不同
（18）A．架构需求说明书　B．架构实现说明书
C．架构质量说明书　D．架构评审说明书

• 以下关于软件架构设计重要性的描述，（19）是错误的。

（19）A．软件架构设计能够满足系统的性能、安全性、可维护性等品质
B．软件架构设计能够帮助项目干系人(Stakeholder)更好地理解软件结构
C．软件架构设计能够帮助架构师更好地捕获和细化系统需求
D．软件架构设计能够有效地管理系统的复杂性，并降低系统维护费用

• 特定领域软件架构(Domain Specific Software Architecture, DSSA)是在一个特定应用领域中，为一组应用提供组织结构参考的标准软件体系结构。参加 DSSA 的人员可以划分为多种角色，其中（20）的任务是控制整个领域分析过程，进行知识获取，将获取的知识组织到领域模型中；（21）的任务是根据领域模型和现有系统开发出 DSSA，并对 DSSA 的准确性和一致性进行验证。

（20）A．领域专家　B．领域分析者　C．领域设计者　D．领域实现者
（21）A．领域专家　B．领域分析者　C．领域设计者　D．领域实现者

• 软件架构风格描述某一特定领域中的系统组织方式和惯用模式，反映了领域中众多系统所共有的（22）特征。对于语音识别、知识推理等问题复杂、解空间很大、求解过程不确定的这一类软件系统。通常会采用（23）架构风格。

（22）A．语法和语义　　B．结构和语义　　C．静态和动态　　D．行为和约束

（23）A．管道-过滤器　　B．解释器　　C．黑板　　D．过程控制

• 某公司欲开发一个在线交易系统，在架构设计阶段，公司的架构师识别出 3 个核心质量属性场景。其中“当系统面临断电故障后，需要在 1 小时内切换至备份站点并恢复正常运行”主要与（24）质量属性相关，通常可采用（25）架构策略实现该属性；“在并发用户数量为 1000 人时，用户的交易请求需要在 0.5 秒内得到响应”主要与（26）质量属性相关，通常可采用（27）架构策略实现该属性；“对系统的消息中间件进行替换时，替换工作需要在 5 人/月内完成”主要与(28)质量属性相关，通常可采用(29)架构策略实现该属性。

（24）A．性能　　B．安全性　　C．可用性　　D．可修改性

（25）A．操作隔离　　B．资源调度　　C．心跳　　D．内置监控器

（26）A．性能　　B．易用性　　C．可用性　　D．互操作性

（27）A．主动冗余　　B．资源调度　　C．抽象接口　　D．记录/回放

（28）A．可用性　　B．安全性　　C．可测试性　　D．可修改性

（29）A．接口-实现分离　　B．记录/回放　　C．内置监控器　　D．追踪审计

• 识别风险、非风险、敏感点和权衡点是进行软件架构评估的重要过程。“改变业务数据编码方式会对系统的性能和安全性产生影响”是对（30）的描述，“假设用户请求的频率为每秒 1 个，业务处理时间小于 30 毫秒，则将请求响应时间设定为 1 秒钟是可以接受的”是对（31）的描述。

（30）A．风险点　　B．非风险　　C．敏感点　　D．权衡点

（31）A．风险点　　B．非风险　　C．敏感点　　D．权衡点

• 以下关于软件架构风格与系统性能的关系叙述中，错误的是（32）。

（32）A．对于采用层次化架构风格的系统，划分的层次越多，系统的性能越差

B．对于采用隐式调用架构风格的系统，可以通过处理函数的并发调用提高系统处理性能

C．采用面向对象架构风格的系统，可以通过引入对象管理层提高系统性能

D．对于采用解释器架构风格的系统，可以通过部分解释代码预先编译的方式提高系统性能

• 架构描述语言(Architecture Description Language，ADL）是一种为明确说明软件系统的概念架构和对这些概念架构建模提供功能的语言。ADL 主要包括以下组成部分：组件、组件接口、（33）和架构配置。

（33）A．架构风格　　B．架构实现　　C．连接件　　D．组件约束

• 某公司拟开发了个轿车巡航定速系统，系统需要持续测量车辆当前的实时速度，并根据设定的期望速度启动控制轿车的油门和刹车。针对上述需求，采用（34）架构风格最为合适。

（34）A．解释器　　B．过程控制　　C．分层　　D．管道-过滤器

• 某公司拟开发一套在线游戏系统，该系统的设计目标之一是支持用户自行定义游戏对象属性，行为和对象之间的交互关系。为了实现上述目标，公司应该采用（35）架构风格最为合适。

（35）A．管道-过滤器　　B．隐式调用　　C．主程序-子程序　　D．解释器

● 某公司为其研发的硬件产品设计实现了一种特定的编程语言，为了方便开发者进行软件开发，公司拟开发一套针对该编程语言的集成开发环境，包括代码编辑、语法高亮、代码编译、运行调试等功能。针对上述描述，该集成开发环境应采用（36）架构风格最为合适。

（36）A．管道-过滤器　　B．数据仓储　　C．主程序-子程序　　D．解释器

系统安全分析与设计

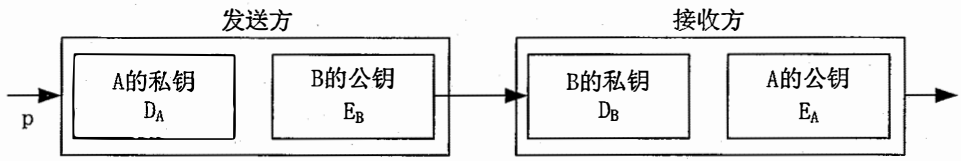
- 常用对称加密算法不包括（1）。

（1）A．DES　　B．RC-5　　C．IDEA　　D．RSA

- 数字签名的功能不包括（2）。

（2）A．防止发送方的抵赖行为　　B．发送方身份确认
　　C．接收方身份确认　　D．保证数据的完整性

- 某数字签名系统如下图所示。网上传送的报文是（3），如果 A 否认发送，作为证据的是（4）。



（3）A．P　　B．D_A（P）　　C．E_B（D_A（P））　　D．D_A

（4）A．P　　B．D_A（P）　　C．E_B（D_A（P））　　D．D_A

- 用户A从CA处获取了用户B的数字证书，用户A通过（5）可以确认该数字证书的有效性。

（5）A．用户B的公钥　　B．用户B的私钥　　C．CA的公钥　　D．用户A的私钥

- 信息安全的威胁有多种，其中（6）是指通过对系统进行长期监听，利用统计分析方法对诸如通信频度、通信的信息流向、通信总量的变化等参数进行研究，从中发现有价值的信息和规律。

（6）A．窃听　　B．信息泄露　　C．旁路控制　　D．业务流分析

- 信息安全策略应该全面地保护信息系统整体的安全，网络安全体系设计是网络逻辑设计工作的重要内容之一，可从物理线路安全、网络安全、系统安全、应用安全等方面来进行安全体系的设计与规划。其中，数据库的容灾属于（7）的内容。

（7）A．物理线路安全与网络安全　　B．网络安全与系统安全
　　C．物理线路安全与系统安全　　D．系统安全与应用安全

- 所谓网络安全漏洞是指（8）。

（8）A．用户的误操作引起的系统故障
　　B．网络节点的系统软件或应用软件在逻辑设计上的缺陷
　　C．网络硬件性能下降产生的缺陷
　　D．网络协议运行中出现的错误

- 下列安全协议中（9）是应用层安全协议。

（9）A．IPSec　　B．L2TP　　C．PAP　　D．HTTPS

- 在网络管理中要防止各种安全威胁。在SNMP v3中，无法预防的安全威胁是（10）。

（10）A．篡改管理信息：通过改变传输中的SNMP报文实施未经授权的管理操作

B. 通信分析：第三者分析管理实体之间的通信规律，从而获取管理信息
C. 假冒合法用户：未经授权的用户冒充授权用户，企图实施管理操作
D. 消息泄露：SNMP 引擎之间交换的信息被第三者偷听

- 下列安全协议中，（11）能保证交易双方无法抵赖。

（11）A. SET　　B. SHTTP　　C. PGP　　D. MOSS

- 下列技术中，不是传输层安全技术的是（12）。

（12）A. SSL　　B. SOCKS　　C. IPSEC　　D. 安全 RPC

- 安全审计系统是保障计算机系统安全的重要手段之一，其作用不包括（13）。

（13）A. 检测对系统的入侵
B. 发现计算机的滥用情况
C. 提供系统运行的日志，从而能发现系统入侵行为和潜在的漏洞
D. 保证可信网络内部信息不外泄

- 防火墙把网络划分为几个不同的区域，一般把对外提供网络服务的设备（如 WWW 服务器、FTP 服务器）放置于（14）区域。

（14）A. 信任网络　　B. 非信任网络　　C. 半信任网络　　D. DMZ（非军事化区）

- 如果杀毒软件报告一系列的Word文档被病毒感染，则可以推断病毒类型是（15）；如果用磁盘检测工具（CHKDSK、SCANDISK等）检测磁盘发现大量文件链接地址错误，表明磁盘可能被（16）病毒感染。

（15）A. 文件型　　B. 引导型　　C. 目录型　　D. 宏病毒
（16）A. 文件型　　B. 引导型　　C. 目录型　　D. 宏病毒

- 下图所示 PKI 系统结构中，负责生成和签署数字证书的是（17），负责验证用户身份的是（18）。

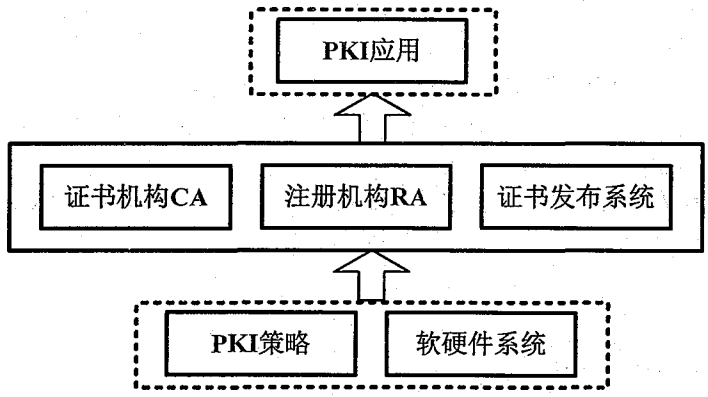


图 1　PKI 系统结构

（17）A. 证书机构 CA　　B. 注册机构 RA　　C. 证书发布系统　　D. PKI 策略
（18）A. 证书机构 CA　　B. 注册机构 RA　　C. 证书发布系统　　D. PKI 策略

项目管理

● 以下关于需求管理的叙述中，正确的是（1）。

（1）A．需求管理是一个对系统需求及其变更进行了解和控制的过程

B．为了获得项目，开发人员可以先向客户做出某些承诺

C．需求管理的重点在于收集和分析项目需求

D．软件开发过程是独立于需求管理的活动

● 项目管理工具用来辅助项目经理实施软件开发过程中的项目管理活动，它不能（2）。（3）就是一种典型的项目管理工具。

（2）A．覆盖整个软件生存周期

B．确定关键路径、松弛时间、超前时间和滞后时间

C．生成固定格式的报表和裁剪项目报告

D．指导软件设计人员按软件生存周期各个阶段的适用技术进行设计工作

（3）A．需求分析工具　B．成本估算工具　C．软件评价工具　D．文档分析工具

● 详细的项目范围说明书是项目成功的关键。（4）不应该属于范围定义的输入。

（4）A．项目章程　B．项目范围管理计划

C．批准的变更申请　D．项目文档管理方案

● 项目时间管理包括使项目按时完成所必需的管理过程，活动定义是其中的一个重要过程。通常可以使用（5）来进行活动定义。

（5）A．鱼骨图　B．工作分解结构（WBS）

C．层次分解结构　D．功能分解图

● 某软件公司开发某种软件产品时花费的固定成本为16万元，每套产品的可变成本为2元，设销售单价为10元，则需要销售（6）套才能达到盈亏平衡点。

（6）A．13000　B．16000　C．18000　D．20000

● 软件产品配置是指一个软件产品在生存周期各个阶段所产生的各种形式和各种版本的文档、计算机程序、部件及数据的集合。该集合的每一个元素称为该产品配置中的一个配置项。下列不应该属于配置项的是（7）。

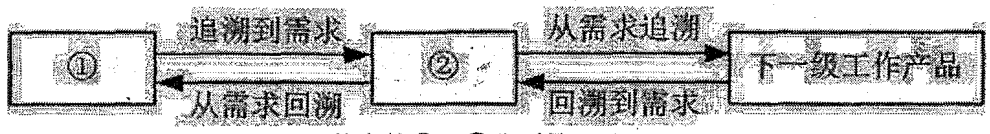
（7）A．源代码清单　B．设计规格说明书

C．软件项目实施计划　D．CASE工具操作手册

● 软件质量保证是软件项目控制的重要手段，（8）是软件质量保证的主要活动之一。

（8）A．风险评估　B．软件评审　C．需求分析　D．架构设计

● 利用需求跟踪能力链（traceability link）可以跟踪一个需求使用的全过程，也就是从初始需求到实现的前后生存期。需求跟踪能力链有4类，如下图所示：



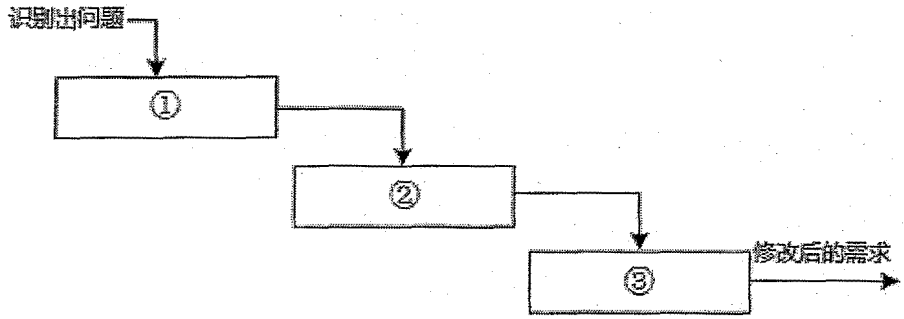
其中的①和②分别是（9）。

（9）A．客户需求、软件需求　　B．软件需求、客户需求
　　C．客户需求、当前工作产品　　D．软件需求、当前工作产品

● 项目配置管理中，配置项的状态通常包括（10）。

（10）A．草稿、正式发布和正在修改　　B．草稿、技术评审和正式发布
　　C．草稿、评审或审批、正式发布　　D．草稿、正式发布和版本变更

● 一个大型软件系统的需求总是有变化的。为了降低项目开发的风险，需要一个好的变更控制过程。如下图所示的需求变更管理过程中，①②③处对应的内容应是（11）；自动化工具能够帮助变更控制过程更有效地运作，（12）是这类工具应具有的特性之一。



（11）A．问题分析与变更描述，变更分析与成本计算，变更实现
　　B．变更描述与变更分析，成本计算，变更实现
　　C．问题分析与变更描述，变更分析，变更实现
　　D．变更描述，变更分析，变更实现

（12）A．自动维护系统的不同版本
　　B．支持系统文档的自动更新
　　C．自动判定变更是否能够实施
　　D．记录每一个状态变更的日期及变更者

计算机组成与体系结构

• 挂接在总线上的多个部件，（1）。

（1）A．只能分时向总线发送数据，并只能分时从总线接收数据

B．只能分时向总线发送数据，但可同时从总线接收数据

C．可同时向总线发送数据，并同时从总线接收数据

D．可同时向总线发送数据，但只能分时从总线接收数据

• RISC(精简指令系统计算机）的特点不包括：（2）。

（2）A．寻址方式尽量丰富，指令功能尽可能强

B．指令长度固定，指令种类尽量少

C．增加寄存器数目，以减少访存次数

D．用硬布线电路实现指令解码，以尽快完成指令译码

• 以下关于 Cache 的叙述中，正确的是（3）。

（3）A．在容量确定的情况下，替换算法的时间复杂度是影响 Cache 命中率的关键因素

B．Cache 的设计思想是在合理的成本下提高命中率

C．Cache 的设计目标是容量尽可能与主存容量相等

D．CPU 中的 Cache 容量应大于 CPU 之外的 Cache 容量

• 某 16 位 AD 芯片中标注电压范围是-5V～＋5V,请问该款 AD 芯片的分辨率是（ ）。

（4）A．10V　　B．0.0763mV　　C．0.1526mV　　D．0.3052mV

• 以下叙述中，正确的是（5）。

（5）A．宿主机与目标机之间只需要建立逻辑连接即可

B．在嵌入式系统中，调试器与被调试程序一般位于同一台机器上

C．在嵌入式系统开发中，通常采用的是交叉编译器

D．宿主机与目标机之间的通信方式只有串口和并口两种

• 以下关于嵌入式系统特点的叙述中，不正确的是（6）。

（6）A．软硬件基本分离　　B．系统实时性强

C．系统透明性　　D．系统资源受限

• 计算机系统中，在（7）的情况下一般应采用异步传输方式。

（7）A．CPU 访问内存　　B．CPU 与 I/O 接口交换信息

C．CPU 与 PCI 总线交换信息　　D．I/O 接口与打印机交换信息

• 某磁盘磁头从一个磁道移至另一个磁道需要 5ms。文件在磁盘上非连续存放，逻辑上相邻数据块的平均移动距离为 10 个磁道，每块的旋转延迟时间及传输时间分别为 50ms 和 2ms，则读取一个 100 块的文件需要（8）ms 的时间。

（8）A．10200　　B．11000　　C．11200　　D．20200

• 内存按字节编址，利用 8K×4bit 的存储器芯片构成 88000H 到 8FFFFH 的内存，共需（9）片。

（9）A．6　　B．8　　C．12　　D．24

• 某计算机系统采用 5 级流水线结构执行指令，设每条指令的执行由取指令(2 Δt)、分析指令(1Δt)、取操作数(3Δt)、运算(1Δt)和写回结果(2Δt) 组成，并分别用 5 个子部件完成，该流水线的最大吞吐率为（10）；若连续向流水线输入 10 条指令，则该流水线的加速比为（11）。

（10）A. $\frac{1}{9\Delta t}$　　B. $\frac{1}{3\Delta t}$　　C. $\frac{1}{2\Delta t}$　　D. $\frac{1}{1\Delta t}$

（11）A.1:10　　B. 2:1　　C. 5:2　　D. 3:1

• 雷达设计人员在设计数字信号处理单元时，其处理器普遍采用 DSP 芯片(比如：TI 公司的 TMS320C63xx)，通常 DSP 芯片采用哈佛(HarVard)体系结构，以下关于哈佛结构特征的描述，不正确的是（12）。

（12）A．程序和数据具有独立的存储空间，允许同时取指令和取操作数，并允许在程序空间或数据空间之间互传数据

B．处理器内部采用多总线结构，保证了在一个机器周期内可以多次访问程序空间和数据空间

C．哈佛体系结构强调的是多功能，适合多种不同的环境和任务，强调兼容性

D．处理器内部采用多处理单元，可以在一个指令周期内同时进行运算

系统配置与性能评价

● 在计算机的可靠性模型中，N 模冗余系统是由 N 个相同的模块和一个（1）组成。

（1）A．缓冲区　　B．控制器　　C．加速器　　D．表决器

● 假设单个 CPU 的性能为 1，则由 n 个这种 CPU 组成的多处理机系统的性能 P 为：

$$p=\frac{n}{1+(n-1)a}$$

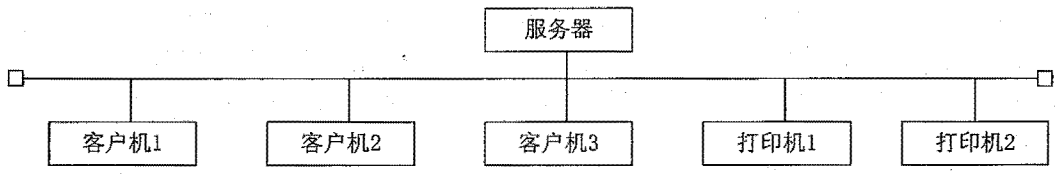
其中，a 是一个表示开销的常数。例如，a=0.1，n=4 时，P 约为 3。也就是说，由 4 个这种 CPU 组成的多机系统的性能约为 3。该公式表明，多机系统的性能有一个上限，不管 n 如何增加，P 都不会超过某个值。当 a=0.1 时，这个上限是（2）。

（2）A．5　　B．10　　C．15　　D．20

● 设系统的失效率为 λ，则系统的平均故障间隔时间 MTBF 为（3）。

（3）A．1/λ　　B．2λ　　C．λ^2　　D．1+λ

● 1 台服务器、3 台客户机和 2 台打印机构成了一个局域网（如下图所示）。在该系统中，服务器根据某台客户机的请求，数据在一台打印机上输出。设服务器、各客户机及各打印机的可用性分别为 a、b、c，则该系统的可用性为（4）。



（4）A．ab^3c^3　　B．$a(1-b^3)(1-c^2)$

C．$a(1-b)^3(1-c)^2$　　D．$a(1-(1-b)^3)(1-(1-c)^2)$

● 以下关于计算机性能改进的叙述中，正确的是（5）。

（5）A．如果某计算机系统的 CPU 利用率已经达到 100%，则该系统不可能再进行性能改进

B．使用虚存的计算机系统如果主存太小，则页面交换的频率将增加，CPU 的使用效率就会降低，因此应当增加更多的内存

C．如果磁盘存取速度低，引起排队，此时应安装更快的 CPU 以提高性能

D．多处理机的性能正比于 CPU 的数目，增加 CPU 是改进性能的主要途径

● 以下关于系统性能的叙述中，不正确的是（6）。

（6）A．常见的 Web 服务器性能评估方法有基准测试、压力测试和可靠性测试

B．评价 Web 服务器的主要性能指标有最大并发连接数、响应延迟和吞吐量

C．对运行系统进行性能评估的主要目的是以更好的性能/价格比更新系统

D．当系统性能降到基本水平时，需要查找影响性能的瓶颈并消除该瓶颈

● 事务处理系统运行时，系统的吞吐率指标（每秒处理的事务数）会随系统负荷（系统中待处理的事务数量）大小而变化。当系统的负荷从 0 开始逐步增大时，系统吞吐率的变化一般将先后经历如下三个阶段：（7）。

（7）A．快增长、慢增长、下降　　B．慢增长、快增长、下降
　　C．快增长、较慢增长、慢增长　　D．慢增长、较快增长、快增长

● 以下关于改进信息系统性能的叙述中，正确的是（8）。

（8）A．将 CPU 时钟周期加快一倍，能使系统吞吐率增加一倍
　　B．一般情况下，增加磁盘容量可以明显缩短作业的平均 CPU 处理时间
　　C．如果事务处理平均响应时间很长，首先应注意提高外围设备的性能
　　D．利用性能测试工具，可以找出程序中最花费运行时间的 20%代码，再对这些代码进行优化

● 提高计算机可靠性可以采用冗余技术，下面的例子中属于信息冗余的是（9）。

（9）A．软件备份　　B．CRC 校验　　C．程序卷回　　D．指令复执

● 下面关于计算机性能的各种评估方法的论述中，正确的是（10）。

（10）A．每秒百万次指令（MIPS）描述了计算机的浮点运算速度
　　B．等效指令速度法采用灵活的指令比例来评价计算机的性能
　　C．峰值 MFLOPS 以最慢的浮点指令来表示计算机的运算速度
　　D．CTP 以每秒百万次理论运算（MTOPS）来表示运算部件的综合性能

● 计算机性能评价有多种方法，在关于各种性能评价方法的描述中，不正确的是（11）。

（11）A．用 MIPS 来比较体系结构不同的计算机的运算速度可能会得出错误的结论
　　B．吉布森（Gibson）混合比例计算法会受到 Cache 命中率和流水线效率的影响而出现测量偏差
　　C．PDR 方法测量的是指令执行的平均时间，它可以全面反映计算机的性能指标
　　D．CTP 方法计算的是每秒百万次理论运算（MTOPS），因而叫做综合理论性能

● 在计算机系统中，某一功能的处理时间为整个系统运行时间的 50%，若使该功能的处理速度加快 10 倍，根据 Amdahl 定律，这样做可以使整个系统的性能提高（12）倍。若要使整个系统的性能提高 1.5 倍，则该功能的处理速度应加快（13）倍。

（12）A．1.6　　B．1.7　　C．1.8　　D．1.9
（13）A．3　　B．5　　C．7　　D．8

● 数据备份是信息系统运行管理时保护数据的重要措施。（14）可针对上次任何一种备份进行，将上次备份后所有发生变化的数据进行备份，并将备份后的数据进行标记。

（14）A．增量备份　　B．差异备份　　C．完全备份　　D．按需备份

● SPEC 计算机性能测试有不同的方法，吞吐率测试是指对（15）的测试。

（15）A．计算机完成单个任务要用多少时间　　B．计算机在一定时间内能完成多少任务
　　C．计算机在一定时间内能输出多少数据　　D．计算机在一段时间内能并行执行多少个程序

● 容错计算机中采用冗余技术来提高系统的可靠性和可用性。这些冗余技术不包括（16）。

（16）A．硬件冗余　　B．信息冗余　　C．时间冗余　　D．人员冗余

● 实现容错计算的主要手段是冗余。（17）中动态冗余的典型例子是热备系统和冷备系统。

（17）A．结构冗余　　B．信息冗余　　C．时间冗余　　D．冗余附加技术

● 常用的软件冗余方法有多种，在关于软件冗余的描述中，正确的是（18）。

（18）A．多版本程序设计可以检查软件故障，但不能检查硬件故障

B．用一组数据检查运算逻辑部件的功能属于能力检查

C．一致性检查时要把计算机的计算结果与手工计算结果进行比较

D．软件冗余是一种动态冗余技术

● 以下关于基准测试的叙述中，正确的是（19）。

（19）A．运行某些诊断程序，加大负载，检查哪个设备会发生故障

B．验证程序模块之间的接口是否正常起作用

C．运行一个标准程序对多种计算机系统进行检查，以比较和评价它们的性能

D．根据程序的内部结构和内部逻辑，评价程序是否正确

● 可用性质量属性主要关注软件系统的故障和它所带来的后果。（20）是能够提高系统可用性的措施。

（20）A．心跳检测　　B．模块的抽象化　　C．用户授权　　D．记录/重放

● 随着业务的增长，信息系统的访问量和数据流量快速增加，采用负载均衡（Load Balance）方法可避免由此导致的系统性能下降甚至崩溃。以下关于负载均衡的叙述中，错误的是（21）。

（21）A．负载均衡通常由服务器端安装的附加软件来实现

B．负载均衡并不会增加系统的吞吐量

C．负载均衡可在不同地理位置、不同网络结构的服务器群之间进行

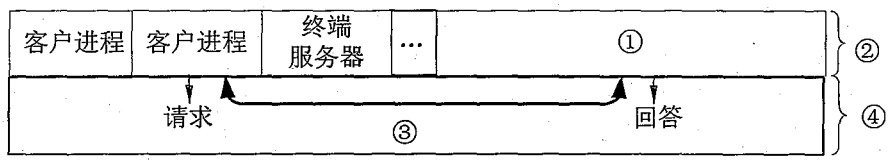
D．负载均衡可使用户只通过一个 IP 地址或域名就能访问相应的服务器

操作系统

● 操作系统为用户提供了两类接口：操作一级和程序控制一级的接口，以下不属于操作一级的接口是（1）。

（1）A．操作控制命令　　B．系统调用
　　C．菜单　　D．窗口

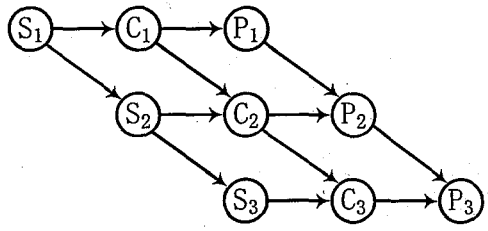
● 微内核体系结构的操作系统（OS）实现时的基本思想是（2），其结构图如下所示，图中①②③④应填写（3）。



（2）A．内核完成 OS 所有功能并在用户态下运行
　　B．内核完成 OS 所有功能并在核心态下运行
　　C．内核只完成 OS 最基本的功能并在核心态下运行，其他功能运行在用户态
　　D．内核只完成 OS 最基本的功能并在用户态下运行，其他功能运行在核心态

（3）A．①进程、文件和存储器服务器②核心态③进程调度、消息通讯等④用户态
　　B．①进程、文件和存储器服务器②用户态③进程调度、消息通讯等④核心态
　　C．①进程调度、消息通讯等②用户态③进程、文件和存储等服务器④核心态
　　D．①进程调度、消息通讯等②核心态③进程、文件和存储等服务器④用户态

● 某计算机系统中有一个 CPU、一台扫描仪和一台打印机。现有三个图像任务，每个任务有三个程序段：扫描 Si，图像处理 Ci 和打印 Pi （i=1,2,3）。下图为三个任务各程序段并发执行的前驱图，其中，（4）可并行执行，（5）的直接制约，（6）的间接制约。



（4）A．“C_1S_2”，“$P_1C_2S_3$”，“P_2C_3”　　B．“C_1S_1”，“$S_2C_2P_2$”，“C_3P_3”
　　C．“$S_1C_1P_1$”，“$S_2C_2P_2$”，“$S_3C_3P_3$”　　D．“$S_1S_2S_3$”，“$C_1C_2C_3$”，“$P_1P_2P_3$”

（5）A．S_1 受到 S_2 和 S_3、C_1 受到 C_2 和 C_3、P_1 受到 P_2 和 P_3
　　B．S_2 和 S_3 受到 S_1、C_2 和 C_3 受到 C_1、P_2 和 P_3 受到 P_1
　　C．C_1 和 P_1 受到 S_1、C_2 和 P_2 受到 S_2、C_3 和 P_3 受到 S_3
　　D．C_1 和 S_1 受到 P_1、C_2 和 S_2 受到 P_2、C_3 和 S_3 受到 P_3

（6）A．S_1 受到 S_2 和 S_3、C_1 受到 C_2 和 C_3、P_1 受到 P_2 和 P_3
　　B．S_2 和 S_3 受到 S_1、C_2 和 C_3 受到 C_1、P_2 和 P_3 受到 P_1
　　C．C_1 和 P_1 受到 S_1、C_2 和 P_2 受到 S_2、C_3 和 P_3 受到 S_3
　　D．C_1 和 S_1，受到 P_1、C_2 和 S_2 受到 P_2、C_3 和 S_3 受到 P_3

● 采用微内核结构的操作系统提高了系统的灵活性和可扩展性，（7）。

（7）A．并增强了系统的可靠性和可移植性，可运行于分布式系统中
B．并增强了系统的可靠性和可移植性，但不适用于分布式系统
C．但降低了系统的可靠性和可移植性，可运行于分布式系统中
D．但降低了系统的可靠性和可移植性，不适用于分布式系统

● 某虚拟存储系统采用最近最少使用（LRU）页面淘汰算法，假定系统为每个作业分配 4 个页面的主存空间，其中一个页面用来存放程序。现有某作业的程序如下：

```
Var A: Array[1. . 100,1. . 100] OF integer;
    i,j: integer;
    FOR i:=1 to 100 DO
        FOR j:=1 to 100 DO
            A[i,j]:=0;
```

设每个页面可存放 200 个整数变量，变量 i、j 存放在程序页中。初始时，程序及 i、j 均已在内存，其余 3 页为空。若矩阵 A 按行序存放，那么当程序执行完后共产生（8）次缺页中断；若矩阵 A 按列序存放，那么当程序执行完后共产生（9）次缺页中断。

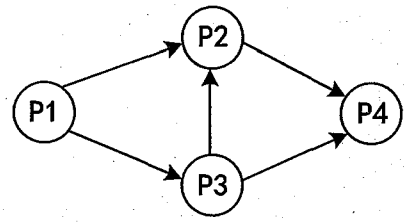
（8）A．50　　B．100　　C．5000　　D．10000
（9）A．50　　B．100　　C．5000　　D．10000

● 若读取（10）的某个磁盘块进行修改，将结果写回磁盘前系统崩溃，则对系统的影响相对较大。
（10）A．用户文件　　B．空闲表
C．系统目录文件　　D．用户目录文件

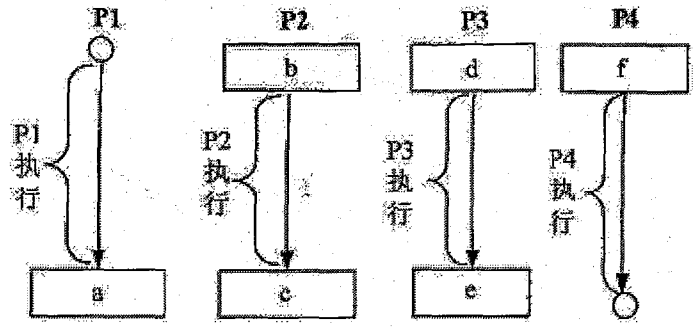
● 假设系统中有 n 个进程共享 3 台打印机，任一进程在任一时刻最多只能使用 1 台打印机。若用 PV 操作控制 n 个进程使用打印机，则相应信号量 s 的取值范围为（11）；若信号量 S 的值为-3，则系统中有（12）个进程等待使用打印机。
（11）A．0，-1，…，-(n-1)　　B．3，2，1，0，-1，…，-(n-3)
C．1，0，-1，…，-(n-1)　　D．2，1，0，-1，…，-(n-2)
（12）A．0　　B．1　　C．2　　D．3

● 在操作系统中，虚拟输入/输出设备通常采用（13）来实现。
（13）A．Spooling 技术，利用磁带　　B．Spooling 技术，利用磁盘
C．脱机批处理技术，利用磁盘　　D．通道技术，利用磁带

● 进程 P1、P2、P3 和 P4 的前趋图如下所示：



若用 PV 操作控制进程 P1~P4 并发执行的过程，则需要设置 5 个信号量 S1、S2、S3、S4 和 S5，且信号量 S4~S5 的初值都等于 0。下图中 a、b 和 c 处应分别填写（14）；d、e 和 f 处应分别填写（15）。



（14）A．V(S1)V(S2)、P(S1)V(S3)和 V(S4)　　B．P(S1)V(S2)、P(S1)P(S2)和 V(S1)
　　C．V(S1)V(S2)、P(S1)P(S3)和 V(S4)　　D．P(S1)P(S2)、V(S1)P(S3)和 V(S2)
（15）A．P(S2)、V(S3)V(S5)和 P(S4)P(S5)　　B．V(S2)、P(S3)V(S5)和 V(S4)P(S5)
　　C．P(S2)、V(S3)P(S5)和 P(S4)V(S5)　　D．V(S2)、V(S3)P(S5)和 P(S4)V(S5)

● 某进程有 4 个页面，页号为 0~3，页面变换表及状态位、访问位和修改位的含义如下图所示。若系统给该进程分配了 3 个存储块，当访问前页面 1 不在内存时，淘汰表中页号为（16）的页面代价最小。

页号	页帧号	状态位	访问位	修改位
0	6	1	1	1
1	—	0	0	0
2	3	1	1	1
3	2	1	1	0

状态位含义：=0 不在内存；=1 在内存
访问位含义：=0 未访问过；=1 访问过
修改位含义：=0 未修改过；=1 修改过

（16）A．0　　B．1　　C．2　　D．3

● 某文件管理系统在磁盘上建立了位示图，来记录磁盘的使用情况。若磁盘上的物理块依次编号为：0、1、2、…，系统中字长为 16 位，每一位对应文件存储器上的一个物理块，取值 0 和 1 分别表示空闲和占用，如下所示。

15	14	…		3	2	1	0
0	1	…	1	0	0	0	1

假设将 2057 号物理块分配给某文件，那么该物理块的使用情况在位示图中的第（17）个字中描述；系统应该将该字的（18）。

（17）A．128　　B．129　　C．130　　D．131
（18）A．编号为 9 的位置“0”　　B．编号为 9 的位置“1”
　　C．编号为 8 的位置“0”　　D．编号为 8 的位置“1”

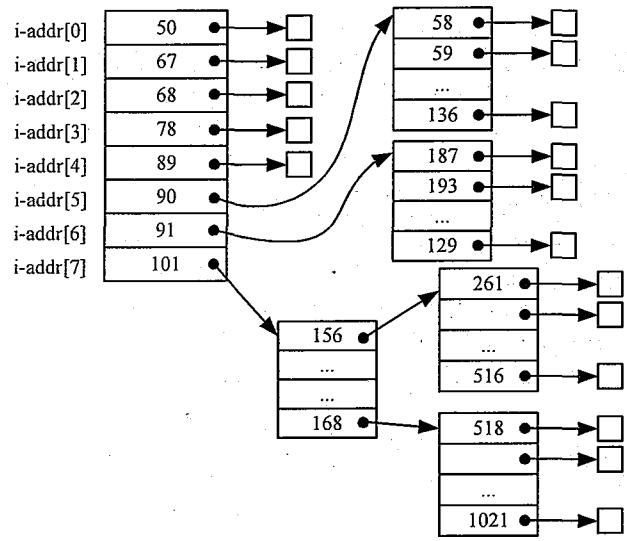
● 设某进程的段表如下所示，逻辑地址（19）可以转换为对应的物理地址。

段号	基地址	段长
0	1598	600
1	486	50

2	90	100
3	1327	2988
4	1952	960

（19）A. （0，1597）、（1，30）和（3，1390）
B. （0，128）、（1，30）和（3，1390）
C. （0，1597）、（2，98）和（3，1390）
D. （0，128）、（2，98）和（4，1066）

● 假设文件系统采用索引节点管理，且索引节点有 8 个地址项 iaddr[0]～iaddr[7]，每个地址项大小为 4 字节，iaddr[0]～iaddr[4]采用直接地址索引，iaddr[5]和 iaddr[6]采用一级间接地址索引，iaddr[7]采用二级间接地址索引。假设磁盘索引块和磁盘数据块大小均为 1KB 字节，文件 File1 的索引节点如图所示。若用户访问文件 File1 中逻辑块号为 5 和 261 的信息，则对应的物理块号分别为（20）；101 号物理块存放的是（21）。



索引文件示意图

（20）A. 89 和 90　　B. 89 和 136　　C. 58 和 187　　D. 90 和 136
（21）A. File1 的信息　　B. 直接地址索引表　　C. 一级地址索引表　　D. 二级地址索引表

计算机网络

● 网络开发过程中，物理网络设计阶段的任务是（1）。

（1）A．依据逻辑网络设计的功能要求，确定设备的具体物理分布和运行环境

B．分析现有网络和新网络的各类资源分布，掌握网络所处状态

C．根据需求规范和通信规范，实施资源分配和安全规划

D．理解网络应该具有的功能和性能，最终设计出符合用户需求的网络

● 希赛公司欲构建一个网络化的开放式数据存储系统，要求采用专用网络连接并管理存储设备和存储管理子系统。针对这种应用，采用（2）存储方式最为合适。

（2）A．内置式存储　　B．DAS　　C．SAN　　D．NAS

● 大型局域网通常划分为核心层、汇聚层和接入层，以下关于各个网络层次的描述中，不正确的是（3）。

（3）A．核心层进行访问控制列表检查　　B．汇聚层定义了网络的访问策略

C．接入层提供局域网络接入功能　　D．接入层可以使用集线器代替交换机

● 网络系统设计过程中，逻辑网络设计阶段的任务是（4）。

（4）A．依据逻辑网络设计的要求，确定设备的物理分布和运行环境

B．分析现有网络和新网络的资源分布，掌握网络的运行状态

C．根据需求规范和通信规范，实施资源分配和安全规划

D．理解网络应该具有的功能和性能，设计出符合用户需求的网络

● 网络系统生命周期可以划分为5个阶段，实施这5个阶段的合理顺序是（5）。

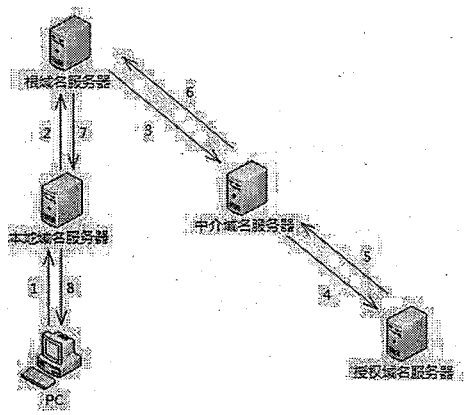
（5）A．需求规范、通信规范、逻辑网络设计、物理网络设计、实施阶段

B．需求规范、逻辑网络设计、通信规范、物理网络设计、实施阶段

C．通信规范、物理网络设计、需求规范、逻辑网络设计、实施阶段

D．通信规范、需求规范、逻辑网络设计、物理网络设计、实施阶段

● 主机 PC 对某个域名进行查询，最终由该域名的授权域名服务器解析并返回结果，查询过程如下图所示。这种查询方式中不合理的是（6）。



（6）A．根域名服务器采用递归查询，影响了性能

B．根域名服务器采用迭代查询，影响了性能
C．中介域名服务器采用迭代查询，加重了根域名服务器负担
D．中介域名服务器采用递归查询，加重了根域名服务器负担

● 网络设计过程包括逻辑网络设计和物理网络设计两个阶段，下面的选项中，（7）应该属于逻辑网络设计阶段的任务。
（7）A．选择路由协议　　B．设备选型　　C．结构化布线　　D．机房设计

● 以下关于网络存储的叙述，正确的是（8）。
（8）A．DAS 支持完全跨平台文件共享，支持所有的操作系统
B．NAS 通过 SCSI 连接至服务器，通过服务器网卡在网络上传输数据
C．FC SAN 的网络介质为光纤通道，而 IP SAN 使用标准的以太网
D．SAN 设备有自己的文件管理系统，NAS 中的存储设备没有文件管理系统

● （9）不是设备选型时应考虑的主要原则。
（9）A．技术指标　　B．成本因素　　C．原有设备的兼容性　　D．采用最新技术

● 确定网络的层次结构及各层采用的协议是网络设计中（10）阶段的主要任务。
（10）A．网络需求分析　　B．网络体系结构设计
C．网络设备选型　　D．网络安全性设计

● 以下关于层次化网络设计的叙述中，不正确的是（11）。
（11）A．在设计核心层时，应尽量避免使用数据包过滤、策略路由等降低数据包转发处理性能的特性
B．为了保证核心层连接运行不同协议的区域，各种协议的转换都应在汇聚层完成
C．接入层应负责一些用户管理功能，包括地址认证、用户认证和计费管理等内容
D．在设计中汇聚层应该采用冗余组件设计，使其具备高可靠性

● 网络设计方案中应重点体现安全性原则，但是不计成本的安全性设计也是不可取的，安全方案应该满足应用需求。下述选项中，（12）安全性需求相对较弱。
（12）A．政府网　　B．校园网　　C．企业网　　D．金融网

● 根据用户需求选择正确的网络技术是保证网络建设成功的关键，在选择网络技术时应考虑多种因素。以下叙述中，不正确的是（13）。
（13）A．选择的网络技术必须保证足够的带宽，使得用户能够快速地访问应用系统
B．选择网络技术时不仅要考虑当前的需求，而且要考虑未来的发展
C．对于大型网络工程，应该选择具有前瞻性的新的网络技术
D．选择网络技术要考虑投入产出比，通过投入产出分析确定使用何种技术

● 假设生产管理网络系统采用B/S工作方式，经常上网的用户数为200，每个用户平均每分钟产生8个事务处理，每个事务处理的规模平均为0.05MB，则这个系统需要的信息传输速率为（14）。
（14）A．4.3Mb/s　　B．7.8Mb/s　　C．10.7Mb/s　　D．17.8Mb/s

● 按照网络分层设计模型，通常把局域网设计为 3 层，即核心层、汇聚层和接入层，以下关于分层网络功

能的描述中，不正确的是（15）。

（15）A．核心层设备负责数据包过滤、策略路由等功能

B．汇聚层完成路由汇总和协议转换功能

C．接入层应提供一部分管理功能，例如 MAC 地址认证、计费管理等

D．接入层负责收集用户信息，例如用户 IP 地址、MAC 地址、访问日志等

- 以下关于域名服务器的叙述，错误的是（16）。

（16）A．本地缓存域名服务不需要域名数据库

B．顶级域名服务器是最高层次的域名服务器

C．本地域名服务器可以采用递归查询和迭代查询两种查询方式

D．权限服务器负责将其管辖区内的主机域名转换为该主机的 IP 地址

- 在网络设计阶段进行通信流量分析时可以采用简单的80/20规则，以下关于这种规则的叙述中，正确的是（17）。

（17）A．这种设计思路可以最大限度地满足用户的远程联网需求

B．这个规则可以随时控制网络的运行状态

C．这个规则适用于内部交流较多而外部访问较少的网络

D．这个规则适用的网络允许存在具有特殊应用的网段

- 假设需要把 25 盒磁带数据（每盒磁带数据量 40GB）从甲地传输到乙地，甲、乙相距 1km，可以采用的方法有汽车运输和 TCP/IP 网络传输，网络传输介质可选用双绞线、单模光纤、多模光纤等。通常情况下，采用（18）介质，所用时间最短。

（18）A．汽车　　B．双绞线　　C．多模光纤　　D．单模光纤

- 网络设计过程分为逻辑网络设计和物理网络设计两个阶段，各个阶段都要产生相应的文档，下面的选项中，（19）应该属于逻辑网络设计文档，（20）属于物理网络设计文档。

（19）A．网络IP地址分配方案　　B．设备列表清单

C．集中访谈的信息资料　　D．网络内部的通信流量分布

（20）A．网络IP地址分配方案　　B．设备列表清单

C．集中访谈的信息资料　　D．网络内部的通信流量分布

- 结构化布线系统分为六个子系统，其中水平子系统（21）。

（21）A．由各种交叉连接设备以及集线器和交换机等设备组成

B．连接了干线子系统和工作区子系统

C．由终端设备到信息插座的整个区域组成

D．实现各楼层设备间子系统之间的互连

- 使用者通过网络按需随时获取和使用资源或服务，并随时扩展，按使用付费，这种计算模式属于（22）。

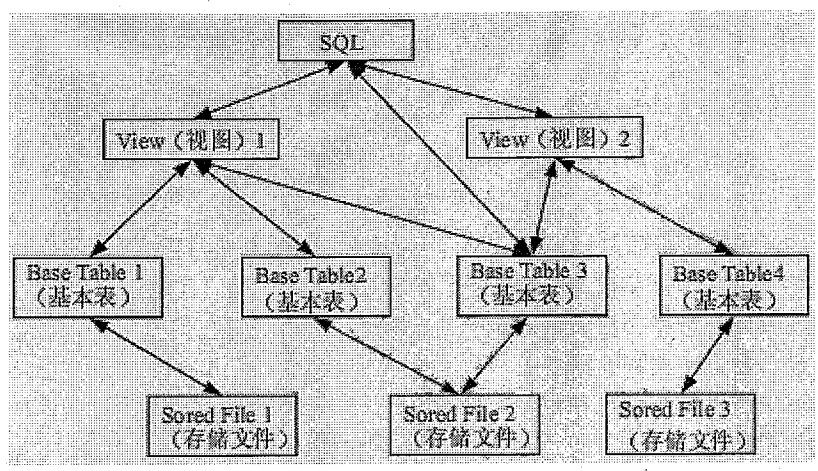
（22）A．远程计算　　B．即时计算　　C．云计算　　D．智能计算

- 在有关物体中安装信息传感设备，使其与互联网相连接，进行信息交换和通信，以实现对物体的智能化识别、定位、跟踪、监控和管理，这样的一种网络称为（23）。

（23）A．移动互联网　　B．全球定位系统　　C．智联网　　D．物联网

数据库系统

● SQL语言支持关系数据库的三级模式结构图如下所示，图中视图、基本表、存储文件分别对应（1）。



（1）A．模式、内模式、外模式　　B．外模式、模式、内模式
　　C．模式、外模式、内模式　　D．外模式、内模式、模式

● 在数据库设计的需求分析阶段应完成包括（2）在内的文档。

（2）A．E-R 图　　B．关系模式
　　C．数据字典和数据流图　　D．任务书和设计方案

● 设有职务工资关系 P（职务，最低工资，最高工资），员工关系 EMP（员工号，职务，工资），要求任何一名员工，其工资值必须在其职务对应的工资范围之内，实现该需求的方法是（3）。

（3）A．建立“EMP.职务”向“P.职务”的参照完整性约束
　　B．建立“P.职务”向“EMP.职务”的参照完整性约束
　　C．建立 EMP 上的触发器程序审定该需求
　　D．建立 P 上的触发器程序审定该需求

● 某数据库中有员工关系 E（员工号，姓名，部门，职称，月薪）；产品关系 P（产品号，产品名称，型号，尺寸，颜色）；仓库关系 W（仓库号，仓库名称，地址，负责人）；库存关系 I（仓库号，产品号，产品数量）。若数据库设计中要求：

① 仓库关系 W 中的“负责人”引用员工关系的员工号
② 库存关系 I 中的“仓库号，产品号”惟一标识 I 中的每一个记录
③ 员工关系 E 中的职称为“工程师”的月薪不能低于 3500 元

则①②③依次要满足的完整性约束是（4）。

（4）A．实体完整性、参照完整性、用户定义完整性
　　B．参照完整性、实体完整性、用户定义完整性
　　C．用户定义完整性、实体完整性、参照完整性
　　D．实体完整性、用户定义完整性、参照完整性

若需得到每种产品的名称和该产品的总库存量，则对应的查询语句为：

SELECT 产品名称, SUM（产品数量）

FROM P,I
WHERE P.产品号 =I.产品号 （5）;

（5）A．ORDER BY 产品名称　　B．ORDER BY 产品数量
C．GROUP BY 产品名称　　D．GROUP BY 产品数量

● 给定学生 S（学号，姓名，年龄，入学时间，联系方式）和选课 SC（学号，课程号，成绩）关系，若要查询选修了 1 号课程的学生学号、姓名和成绩，则该查询与关系代数表达式（6）等价。

（6）A．$\pi_{1,2,8}(\sigma_{1=6\wedge 7='1'}(S \bowtie SC))$　　B．$\pi_{1,2,7}(\sigma_{6='1'}(S \bowtie SC))$

C．$\pi_{1,2,7}(\sigma_{1=6}(S \bowtie SC))$　　D．$\pi_{1,2,8}(\sigma_{7='1'}(S \bowtie SC))$

● 在数据库设计的需求分析阶段应当形成（7），这些文档可以作为（8）阶段的设计依据。

（7）A．需求说明文档、数据字典和数据流图
B．需求说明文档、程序文档和数据流图
C．程序文档、数据字典和数据流图
D．需求说明文档、数据字典和程序文档

（8）A．逻辑结构设计　　B．物理结构设计
C．概念结构设计　　D．数据库运行和维护

● 若要使某用户只能查询表 EMP 中的部分记录，应采取的策略是（9）。

（9）A．将该用户级别设定为 DBA
B．将表 EMP 的查询权限赋予该用户
C．编写查询表 EMP 的存储过程
D．构建该部分记录的行级视图，并将该视图的查询权限赋予该用户

● 假设关系模式 R(U，F)，属性集 U={A，B，C}，函数依赖集 F={A→B，B→C}。若将其分解为 p={R1(U1，F1)，R2(U2，F2))，其中 U1={A，B}，U2={A，C}。那么，关系模式 R、R1、R2 分别达到了（10）；分解 ρ （11）。

（10）A．1NF、2NF、3NF　　B．1NF、3NF、3NF
C．2 NF、2NF、3NF　　D．2NF、3NF、3NF

（11）A．有损连接但保持函数依赖　　B．既无损连接又保持函数依赖
C．有损连接且不保持函数依赖　　D．无损连接但不保持函数依赖

● 在数据库系统中，数据的完整性是指数据的（12）。

（12）A．有效性、正确性和一致性　　B．有效性、正确性和可维护性
C．有效性、正确性和安全性　　D．正确性、一致性和安全性

● 某公司的商品（商品号，商品名称，生产商，单价）和仓库（仓库号，地址，电话，商品号，库存量）两个实体之间的关系如表 1 和表 2 所示。

表 1　商品表

商品号	商品名称	生产商	单 价
10023	笔记本	联想	4800
10024	激光打印机	联想	1650
10025	台式电脑	联想	3860

20003	激光打印机	HP	1280
20004	笔记本	HP	3900
20005	电冰箱	海尔	3860
…	…	…	…

表 2　仓库表

仓库号	地址	电话	商品号	库存量
01	高新路 1 号	8601	10024	26
01	高新路 1 号	8601	10025	89
01	高新路 1 号	8601	20003	10
02	友谊路 6 号	8602	10023	39
02	友谊路 6 号	8602	20004	26
03	高新路 1 号	8603	20005	18
…	…	…	…	…

商品关系的主键是（13）；仓库关系的主键是（14）；仓库关系（15），为了解决这一问题，需要将仓库关系分解为（16）。

（13）A．商品号　　B．商品名称　　C．生产商　　D．单价

（14）A．仓库号，地址　B．仓库号，电话　　C．仓库号，商品号　D．地址，电话

（15）A．无冗余、无插入异常，但存在删除异常　　B．无冗余，但存在插入异常和删除异常

C．存在冗余，但不存在修改操作的不一致

D．存在冗余、修改操作的不一致，以及插入异常和删除异常

（16）A．仓库 1（仓库号，地址）和仓库 2（仓库号，电话，商品号，库存量）

B．仓库 1（仓库号，地址，电话）和仓库 2（商品号，库存量）

C．仓库 1（仓库号，电话）和仓库 2（仓库号，地址，商品号，库存量）

D．仓库 1（仓库号，地址，电话）和仓库 2（仓库号，商品号，库存量）

- 数据仓库在收集数据过程中，会遇到一些略微不一致但可以纠正的数据，纠正的过程称为_（17）。

（17）A．数据转换　　B．数据抽取　　C．数据清洗　　D．数据装载

- 在数据库设计的需求分析阶段，业务流程一般采用（18）表示。

（18）A．数据流图　　B．E-R 图　　C．程序结构图　　D．功能模块图

- 以下关于 RDBMS 数据分布的叙述中，错误的是（19）。

（19）A．数据垂直分割是将不同表的数据存储到不同的服务器上

B．数据水平分割是将不同行的数据存储到不同的服务器上

C．数据复制是将数据的多个副本存储到不同的服务器上

D．数据复制中由 RDBMS 维护数据的一致性

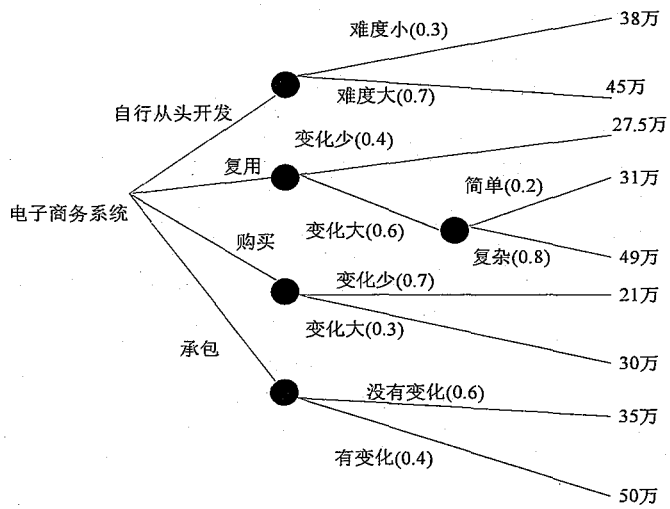
- 企业战略数据模型可分为两种类型：（20）描述日常事务处理中的数据及其关系；（21）描述企业管理决策者所需信息及其关系。

（20）A．元数据模型　　B．数据库模型　　C．数据仓库模型　　D．组织架构模型

（21）A．元数据模型　　B．数据库模型　　C．数据仓库模型　　D．组织架构模型

数学与经济管理

● 某企业拟进行电子商务系统的建设，有四种方式可以选择：①企业自行从头开发； ②复用已有的构件来构造；③购买现成的软件产品；④承包给专业公司开发。针对这几种 方式，项目经理提供了如下图所示的决策树，根据此图，管理者选择建设方式的最佳决策是（1）。



（1）A．企业自行从头开发　　B．复用已有的构件来构造
　　C．购买现成的软件产品　　D．承包给专业公司开发

● 某项目有 A~H 八个作业，各作业所需时间（单位：周）以及紧前作业如下表:

作业名称	A	B	C	D	E	F	G	H
紧前作业	–	A	A	A	B, C	C, D	D	E, F, G
所需时间	1	3	3	5	7	6	5	1

该项目的工期为（ ）周。如果作业 C 拖延 3 周完成，则该项目的工期（ ）。

（2）A．12　　B．13　　C．14　　D．15

（3）A．不变　　B．拖延 1 周　　C．拖延 3 周　　D．拖延 2 周

● 某公司希望举办一个展销会以扩大市场，选择北京、天津、上海、深圳作为候选会址。获利情况除了会址关系外，还与天气有关。天气可分为晴、多云、多雨三种。通过天气预报，估计三种天气情况可能发生的概率为 0．25、0．50、0．25，其收益（单位：人民币万元）情况下表。使用决策树进行决策的结果为（4）。

	晴（0．25）	多云（0．50）	多雨（0．25）
北京	4．5	4．4	1
天津	5	4	1．6
上海	6	3	1．3
深圳	5．5	3．9	0．9

（4）A．北京　　B．天津　　C．上海　　D．深圳

● 某公司新建一座 200 平米的厂房，现准备部署生产某产品的设备。该公司现空闲生产该产品的甲、乙、丙、丁四种型号的设备各 3 台，每种型号设备每天的生产能力由下表给出。在厂房大小限定的情况下，该

厂房每天最多能生产该产品（5）个。

	甲	乙	丙	丁
占地面积（平方米）	40	20	10	5
每天生产能力（个）	100	60	20	8

（5）A．500　　B．520　　C．524　　D．530

● 某公司需要将 4 吨贵金属材料分配给下属的甲、乙、丙三个子公司（单位：吨）。据测算，各子公司得到这些材料后所能获得的利润（单位：万元）如下表所示。

材料 / 子公司	1 吨	2 吨	3 吨	4 吨
甲	4	7	10	13
乙	5	9	11	13
丙	4	6	11	14

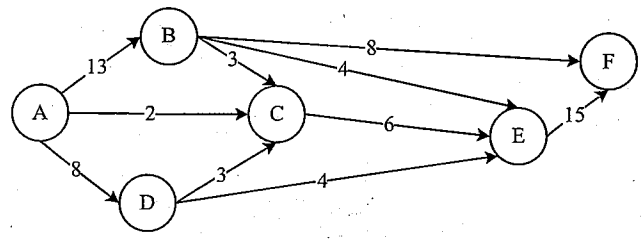
根据上表，只要材料分配适当，该公司最多可以获得利润（6）万元。

（6）A．14　　B．16　　C．17　　D．18

● 某软件公司开发某种软件产品时花费的固定成本为 16 万元，每套产品的可变成本为 2 元，设销售单价为 10 元，则需要销售（7）套才能达到盈亏平衡点。

（7）A．13000　　B．16000　　C．18000　　D．20000

● 在军事演习中，张司令希望将部队尽快从 A 地通过公路网（如图所示）运送到 F 地：



公路网示意图

图中标出了各路段上的最大运量（单位：千人/小时）。根据该图可以算出，从 A 地到 F 地的最大运量是（8）千人/小时。

（8）A．20　　B．21　　C．22　　D．23

● 对实际应用问题建立数学模型并求得结果后，还需要根据建模的目的和要求，利用相关知识，结合研究对象的特点，进行模型分析。模型分析工作一般不包括（9）。

（9）A．模型的合理性分析　　B．模型的误差分析

C．模型的先进性分析　　D．参数的灵敏性分析

● 假设市场上某种商品有两种品牌 A 和 B，当前的市场占有率各为 50%。根据历史经验估计，这种商品当月与下月市场占有率的变化可用转移矩阵 P 来描述：

$$P=\begin{pmatrix} p(A\to A) & p(A\to B) \\ p(B\to A) & p(B\to B) \end{pmatrix}=\begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix}$$

其中，p（A→B）是 A 的市场占有份额中转移给 B 的概率，依次类推。这样，2 个月后的这种商品的

市场占有率变化为（10）。

（10）A．A 的份额增加了 10%，B 的份额减少了 10%

B．A 的份额减少了 10%，B 的份额增加了 10%

C．A 的份额增加了 14%，B 的份额减少了 14%

D．A 的份额减少了 14%，B 的份额增加了 14%

● 求解许多定量的实际问题需要先建立数学模型，然后再对该数学模型进行求解。关于建立并求解数学模型的叙述，不正确的是（11）。

（11）A．建模过程中遇到的最大困难往往是对实际问题的分析、理解和正确描述

B．建模时往往要舍去次要因素，只考虑主要因素，因此模型往往是近似的

C．对复杂问题建立数学模型很难一次成功，往往要经过反复迭代，不断完善

D．连续模型中，模型参数的微小变化不会导致计算结果的很大变化

● 人们需要用观测或测量得到的原始数据建立数学模型来解决实际问题，这种方法称为数据建模法。在建模过程中，下面关于原始数据作用的叙述，不正确的是（12）。

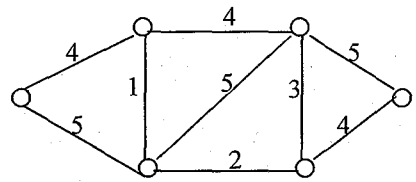
（12）A．原始数据能够对构建什么样的模型给予提示

B．原始数据可以帮助对模型的参数给出估计

C．模型的合理性取决于原始数据的精确性和完整性

D．原始数据可以帮助检验模型、优化模型

● 山区某乡的 6 个村之间有山路如下图所示，其中的数字标明了各条山路的长度（公里）。



乡政府决定沿山路架设电话线。为实现村村通电话，电话线总长至少为（13）公里。

（13）A．11　　B．14　　C．18　　D．33

● 生产某种产品有两个建厂方案：(1)建大厂，需要初期投资 500 万元。如果产品销路好，每年可以获利 200 万元；如果销路不好，每年会亏损 20 万元。(2)建小厂，需要初期投资 200 万元。如果产品销路好，每年可以获利 100 万元；如果销路不好，每年只能获利 20 万元。

市场调研表明，未来 2 年这种产品销路好的概率为 70%。如果这 2 年销路好，则后续 5 年销路好的概率上升为 80%；如果这 2 年销路不好，则后续 5 年销路好的概率仅为 10%。为取得 7 年最大总收益，决策者应（14）。

（14）A．建大厂，总收益超 500 万元　　B．建大厂，总收益略多于 300 万元

C．建小厂，总收益超 500 万元　　D．建小厂，总收益略多于 300 万元

●设三个煤场 A、B、C 分别能供应煤 12、14、10 万吨，三个工厂 X、Y、Z 分别需要煤 11、12、13 万吨，从各煤场到各工厂运煤的单价（百元/吨）见下表方框内的数字。只要选择最优的运输方案，总的运输成本就能降到（ ）百万元。

	工厂X	工厂Y	工厂z	供应量（万吨）
煤场A	5	1	6	12
煤场B	2	4	3	14
煤场C	3	6	7	10
需求量（万吨）	11	12	13	36

A. 153　　B. 113　　C. 91　　D. 83

● 九个项目 A11、A12、A13、A21、A22、A23、A31、A32、A33 的成本从 1 百万、2 百万、…，到 9 百万各不相同，但并不顺序对应。已知 A11 与 A21、A12 与 A22 的成本都有一倍关系，A11 与 A12、A21 与 A31、A22 与 A23、A23 与 A33 的成本都相差 1 百万。由此可以推断，项目 A22 的成本是（16）百万。

（16）A. 2　　B. 4　　C. 6　　D. 8

● 某企业拟生产甲、乙、丙、丁四个产品。每个产品必须依次由设计部门、制造部门和检验部门进行设计、制造和检验，每个部门生产产品的顺序是相同的。各产品各工序所需的时间如下表所示：

项目	设计（天）	制造（天）	检验（天）
甲	13	15	20
乙	10	20	18
丙	20	16	10
丁	8	10	15

只要适当安排好项目实施顺序，企业最快可以在（17）天全部完成这四个项目。

（17）A. 84　　B. 86　　C. 91　　D. 93

知识产权与标准化

● 我国《计算机软件保护条例》所称的“发表”的含义是指将软件作品（1）。

（1）A．出版发行　　B．公之于众

C．在版权局登记　　D．以某种物质形式固定下来

● 依据《计算机软件保护条例》，对软件的保护包括（2）。

（2）A．计算机程序，但不包括用户手册等文档

B．计算机程序及其设计方法

C．计算机程序及其文档，但不包括开发该软件的所用思想

D．计算机源程序，但不包括目标程序

● 以下关于计算机软件商业秘密的叙述，正确的是（3）。

（3）A．没有完成也没有发表的软件，其知识内容不构成商业秘密

B．计算机软件的商业秘密，就是技术秘密，不含经营秘密

C．构成技术秘密的知识内容，不包括程序设计的技巧和算法

D．开源软件也存在商业秘密问题

● 小张在某企业运作部工作，负责统计经营数据并制作成图表供企业高层参考决策。由于日常工作中统计数据量大，为便于及时、准确地完成报表，小张利用下班后的业余时间开发了一个统计软件，则（4）。

（4）A．小张在开发软件的过程中，没有利用企业的物质技术条件，该软件的著作权属于小张

B．该软件的著作权属于企业，因其是小张为完成本职工作而产生的结果

C．按照软件著作权归属的一般原则，该软件的著作权归属于小张

D．该软件没有用于商业目的，没有著作权的问题

● 利用（5）可以对软件的技术信息、经营信息提供保护。

（5）A．著作权　　B．专利权　　C．商业秘密权　　D．商标权

● 根据知识产权法规的有关规定，下列选项中正确的说法是（6）。

（6）A．企业名称权、商业秘密权、商标权均有法定保护期限

B．企业名称权、商标权有法定保护期限，商业秘密权无法定保护期限

C．企业名称权、商业秘密权和商标权均无法定保护期限

D．企业名称权、商业秘密权无法定保护期限，商标权有法定保护期限

● 甲公司的程序员在不影响本职工作的条件下，在乙公司兼职并利用其兼职单位的物质技术条件开发了一套应用程序，该应用程序的著作权属于（7）。

（7）A．甲公司　　B．乙公司　　C．程序员　　D．乙公司与程序员共有

● 希赛公司的用户购买了一台预先安装了操作系统的 PC，后经查实，该 PC 上的操作系统是盗版，而对此情况该用户并不知情，则（8）。

（8）A．该用户承担部分侵权责任　　B．应由其所在单位承担侵权责任

C．应由该 PC 的提供者承担侵权责任　　D．该用户承担全部侵权责任

● 甲公司开发的通信软件，使用“希赛”牌商标，商标没有注册。2011 年 4 月该地另一公司（乙公司）成立，主要开发通信软件，也拟使用“希赛”牌商标，并于 2011 年 5 月 10 日向商标局递交了商标注册申请书。甲公司得知这一消息后，于同年 5 月 25 日也向商标局递交了商标注册申请书。依据我国商标法，（9）能获准“希赛”牌商标注册。

（9）A．甲公司　　B．乙公司　　C．甲、乙公司都　　D．甲、乙公司都不

● 张某原是希赛软件的系统分析师，一直从事计算机网络端口优化处理的研发工作。2011 年 5 月张某退休。半年后，张某研发出网络端口优化处理程序，解决了提高计算机网络端口有效利用率这个技术问题，并以个人名义向专利局提出发明专利申请。该项发明创造应属于（10）所有。

（10）A．软件公司　　B．张某　　C．张某和软件公司共同　　D．社会公众

● GJB473—88（军用软件开发规范）是（11）。

（11）A．国家标准　　B．行业标准　　C．地方标准　　D．企业标准

● （12）为推荐性地方标准的代号。

（12）A．SJ/T　　B．Q/T11　　C．GB/T　　D．DB11/T

● 《GB 8567-88 计算机软件产品开发文件编制指南》是（13）标准，违反该标准而造成不良后果时，将依法根据情节轻重受到行政处罚或追究刑事责任。

（13）A．强制性国家　　B．推荐性国家
　　C．强制性软件行业　　D．推荐性软件行业

● 希赛公司开发了一套能够同硬件结合以提高设备性能的软件产品，向国家专利局申请方法发明专利，获得了专利权，并为该软件产品冠以“希赛”商品专用标识，但未进行商标注册上市销售。此情况下，该软件产品不可能得到我国（14）的保护。

（14）A．著作权法　　B．专利法　　C．商标法　　D．刑法

● 张某是 M 国际运输有限公司计算机系统管理员。任职期间，根据公司的业务要求开发了“空运出口业务系统”，并由公司使用。随后，张某向国家版权局申请了计算机软件著作权登记，并取得了《计算机软件著作权登记证书》，证书明确软件名称是“空运出口业务系统 V1.0”，著作权人为张某。以下说法中，正确的是（15）。

（15）A．空运出口业务系统 V1.0 的著作权属于张某
　　B．空运出口业务系统 V1.0 的著作权属于 M 公司
　　C．空运出口业务系统 V1.0 的著作权属于张某和 M 公司
　　D．张某获取的软件著作权登记证是不可以撤销的

● 甲公司的某个注册商标是乙画家创作的绘画作品，甲申请该商标注册时未经乙的许可，乙认为其著作权受到侵害。在乙可采取的以下做法中，错误的是（16）。

（16）A．向甲公司所在地人民法院提起著作权侵权诉讼
　　B．请求商标评审委员会裁定撤销甲的注册商标
　　C．首先提起诉讼，如对法院判决不服再请求商标评审委员会进行裁定
　　D．与甲交涉，采取许可方式让甲继续使用该注册商标

● 中国 M 公司与美国 L 公司分别在各自生产的平板电脑产品上使用 iPad 商标，且分别享有各自国家批准的

商标专用权。中国Y手电筒经销商，在其经销的手电筒高端产品上也使用iPad商标，并取得了注册商标。以下说法正确的是（17）。

（17）A．L公司未经M公司许可在中国市场销售其产品不属于侵权行为

B．L公司在中国市场销售其产品需要取得M公司和Y经销商的许可

C．L公司在中国市场销售其产品需要向M公司支付注册商标许可使用费

D．Y经销商在其经销的手电筒高端产品上使用iPad商标属于侵权行为

• M画家将自己创作的一幅美术作品原件赠予了L公司。L公司未经该画家的许可，擅自将这幅美术作品作为商标注册，且取得商标权，并大量复制用于该公司的产品上。L公司的行为侵犯了M画家的（18）。

（18）A．著作权　　B．发表权　　C．商标权　　D．展览权

• 以下我国的标准代号中，（19）表示行业标准。

（19）A．GB　　B．SJ　　C．DB11　　D．Q

论文范文

试题　论软件多层架构的设计

目前，三层架构或多层架构已经成为软件开发的主流，采用多层架构有很多好处，例如，能有效降低建设和维护成本，简化管理，适应大规模和复杂的应用需求，可适应不断的变化和新的业务需求等。在多层架构的开发中，中间件的设计占重要地位。

请围绕“**软件多层架构的设计**”论题，依次对以下三个方面进行论述。

1．概要叙述你参与设计和开发的软件项目，以及你所担任的主要工作。

2．具体讨论你是如何设计多层架构的，详细描述其设计过程，遇到过的问题，以及解决的办法。

3．分析你采用多层架构所带来的效果如何，以及有哪些还需要进一步改进的地方，如何改进？

【摘要】

我所在的单位是国内主要的商业银行之一，作为单位的主要技术骨干，2010 年 1 月，我主持了远期结售汇系统的开发，该系统是我行综合业务系统 XX2000 的一个子系统，由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层 C/S 架构作为该系统的软件架构，在详细的设计三层架构的过程中，我采用了字符终端为表示层，CICS TRANSATION SERVER 为中间层，DB2 UDB 7.1 为数据库层，并采用了 CICS SWITCH 组，并行批量的办法来解决设计中遇到的问题，保证了远期结售汇系统按计划完成并顺利投产，我设计的软件三层架构得到了同事和领导的一致认同和称赞。但是，我也看到在三层架构设计中存在一些不足之处：例如中间层的负载均衡算法过于简单，容易造成系统负荷不均衡，并行批量设计不够严谨，容易造成资源冲突等。

【正文】

我所在的单位是国内主要的商业银行之一。众所周知，银行的业务存在一个“二八定理”：即银行的百分之八十的利润是由百分之二十的客户所创造。为了更好地服务大客户，适应我国对外贸易的蓬勃发展态势，促进我国对外贸易的发展，2010 年 1 月，我行开展了远期结售汇业务。

所谓的远期结售汇就是企业在取得中国外汇管理局的批准后，根据对外贸易的合同等凭证与银行制定合约，银行根据制定合约当天的外汇汇率，通过远期汇率公式，计算出交割当天的外汇汇率，并在那天以该汇率进行成交的外汇买卖业务。远期结售汇系统是我行综合业务系统 XX2000 的一个子系统，它主要包括了联机部分、批量部分、清算部分和通兑部分，具有协议管理、合约管理、报价管理、外汇敞口管理、帐务管理、数据拆分管理、报表管理、业务缩微和事后监督等功能。

我作为单位的主要技术骨干之一，主持并参与了远期结售汇系统的项目计划、需求分析、设计、编码和测试阶段的工作。由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层 C/S 架构作为该系统的软件架构，下面，我将分层次详细介绍三层 C/S 软件架构的设计过程。

（1）表示层为字符终端。我行以前一直使用 IBM 的 VISUALGEN 2.0 附带的图形用户终端来开发终端程序，但在使用的过程中，分行的业务人员反映响应速度比较慢，特别是业务量比较大的时候，速度更是难以忍受。为此，我行最近自行开发了一套字符终端 CITE，它采用 VISUAL BASIC 作为开发语言，具有响应速度快，交互能力强，易学，编码快和功能强大的特点，在权衡了两者的优点和缺点之后，我决定选择字符终端 CITE 作为表示层。

（2）中间层为 CICS TRANSATION SERVER（CTS）。首先，我行与 IBM 公司一直保持着良好的合作关系，而我行的大部分技术和设备都采用了 IBM 公司的产品，其中包括了大型机，由于 CICS 在 IBM 的大型机上得到了广泛的应用，并在我行取得了很大的成功，为了保证与原来系统的兼容和互用性，我采用了 IBM 的 CTS 作为中间层，连接表示层和数据库层，简化系统的设计，使开发人员可以专注于表示逻辑和业务逻辑的开发工作，缩短了开发周期，减少开发费用和维护费用，提高了开发的成功率；其次，对于中间

层的业务逻辑，我采用了我行一直使用的 VISUALAGE FOR JAVA 作为开发平台，它具有简单易用的特点，特别适合开发业务逻辑，可以使开发人员快速而准确地开发出业务逻辑，确保了远期结售汇系统的顺利完成；最后，由于采用了 CTS，确保了系统的开放性和互操作性，保证了与我行原来的联机系统和其他系统的兼容，保护了我行的原有投资。

（3）数据层为 DB2 UDB7.1。由于 DB2 在大型事务处理系统中表现出色，我行一直使用 DB2 作为事务处理的数据库，并取得了很大的成功，在 DB2 数据库的使用方面积累了自己独到的经验和大量的人才，为了延续技术的连续性和保护原有投资，我选择了 DB2 UDB7.1 作为数据层。

但是，在设计的过程中也遇到了一些困难，我们主要采取了以下的办法来解决：

（1）CICS SWITCH 组。众所周知，银行系统对于安全性，可靠性，可用性和响应速度要求很高，特别是我行最近进行了数据集中，全国只设两个数据中心，分别在 XX 和 YY 两个地方，这样对以上的要求就更高了，为了保障我行的安全生产，我采用了 CTS SWITCH 组技术。为了简化系统的设计和缩短通讯时间，我采用了简单的负载均衡算法，例如这次分配给第 N 个 CTS，下次则分配给第 N+1 个 CTS，当到了最后一个，就从第一个开始；为了更好地实现容错，我采用了当第 N 个 CTS 失效的时候，把它正在处理的业务转到第 N+1 个上面继续处理，这样大大增加了系统的可用性，可以为客户提供更好的服务；此外，我还采用了数据库连接池的技术，大大缩短了数据库处理速度，提高了系统运行速度。

（2）并行批量。银行系统每天都要处理大量的数据，为了确保白天的业务能顺利进行，有一部分的帐务处理，例如一部分内部户帐务处理，或者代理收费业务和总帐与分户帐核对等功能就要到晚上批量地去处理，但是，这部分数据在数据集中之后就显得更加庞大，我行以前采用串行提交批量作业的办法，远远不能适应数据中心亿万级的数据处理要求，在与其他技术骨干讨论之后，并经过充分的论证和试验，我决定采用了并行批量的技术，所谓的并行批量，就是在利用 IBM 的 OPC（Tivoli Operations，Planning and Control）技术，把批量作业按时间和业务处理先后顺序由操作员统一提交的基础上，再利用 DB2 的 PARTITION 技术，把几个地区分到一个 PATITON 里面分别处理，大大提高了银行系统的数据处理速度，确保了远期结售汇系统三层架构的先进性。在并行批量的设计过程中，我考虑到批量作业有可能因为网络错误或者资源冲突等原因而中断，这样在编写批量程序和作业的时候必须支持断点重提，以确保生产的顺利进行。

由于软件三层架构设计得当，并采取了有效的措施去解决设计中遇到的问题，远期结售汇系统最后按照计划完成并顺利投产，不但保证了系统的开放性、可用性和互用性，取得了良好的社会效益和经济效益，而且我的软件三层架构设计得到了同事和领导的一致认同与称赞，为我行以后系统的开发打下了良好的基础。

在总结经验的同时，我也看到了我在软件三层架构设计中的不足之处：

首先，负载算法过于简单，容易造成系统的负荷不均衡：由于每个业务的处理时间不一样，有的可能差距很远，简单的顺序加一负载分配算法就容易造成负载不均衡，但是如果专门设置一个分配器，则增加了一次网络通讯，使得系统的速度变慢，这样对响应速度要求很高的银行系统来说也是不可行的，于是我决定采用基于统计的分配算法，即在收到请求的时候，根据预先设定的权值，按概率，直接分配给 CTS。

其次，由于批量作业顺序设计得不过够严谨等各种原因，容易造成资源冲突：在远期结售汇系统运行了一段时间之后，数据中心的维护人员发现了，系统有的时候会出现资源冲突现象，在经过仔细的分析之后，我发现，由于每天各个业务的业务量大小不一样，顺序的两个作业之间访问同一个表的时候便会产生资源冲突，另外，在 OPC 作业运行的过程中，操作员提交的其他作业与这个时间的 OPC 作业产生也有可能产生资源冲突。对于第一种情况，可以在不影响业务的情况下调整作业顺序或者对于查询作业运用 DB2 的共享锁的技术，而第二种情况则要制定规范，规定在某时间断内不允许提交某些作业来解决。为了更好地开展系统分析工作，我将在以后的工作实践中不断地学习，提高自身素质和能力，为我国的软件事业贡献自己的微薄力量。

试题　论软件架构风格的选择与应用

软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式（idiomatic paradigm）。对软件架构风格的研究和实践促进了对设计的重用，一些经过实践证实的解决方案也可以可靠地用于解决新的问题。架构风格的不变部分使不同的系统可以共享同一个实现代码。只要系统是使用常用的、规范的方法来组织，就可使别的设计者很容易地理解系统的架构。例如，如果某人把系统描述为 C/S 架构，则不必给出设计细节，我们立刻就会明白系统是如何组织和工作的。

请围绕"**软件架构风格的选择与应用**"论题，依次对以下三个方面进行论述。

1．概要叙述你参与设计和开发的软件项目以及你所担任的主要工作。

2．详细论述你是如何根据项目的实际需要设计软件架构的，特别是如何选择多种不同的架构来实现系统的。

3．分析你采用的架构所带来的效果，你的设计还存在哪些不足之处。

【摘要】

2009 年 5 月，我所在的公司承担了某省社会保险管理信息系统的开发工作，我在该项目中担任系统架构设计师职务，主要负责设计应用系统架构和网络安全体系架构。该系统以 IC 卡为信息载体，完成劳动和社会保险的主要业务管理，即"五保合一"管理，包括养老保险、医疗保险、劳动就业和失业保险、工伤保险、女工生育保险。整个业务流程十分复杂，牵涉面相当广泛。

本文以社会保险管理信息系统为例，讨论了软件架构的选择和应用。整个系统采用具有四层的层次式软件架构的设计思想，在业务管理层的设计中，采用了互连系统构成的系统的架构，把整个业务管理系统划分为八个从属系统。各从属系统的架构可以相同，也可以不同。整个系统的开发工作历时 19 个月，目前，该系统已经稳定运行 1 年多的时间。实践证明，这种架构设计有效地降低了维护成本，提高了系统的开放性、可扩充性、可重用性和可移植性。

【正文】

2009 年 5 月，我所在的公司承担了某省社会保险管理信息系统（以下简称为"SIMIS 系统"）的开发工作，我参加了该项目前期的一些工作，担任系统架构设计师职务，主要负责设计应用系统架构和网络安全体系架构。

1．项目概述

SIMIS 服从于国家劳动和社会保障部关于保险管理信息系统的总体规划，系统建设坚持一体化的设计思想，总体目标是建立比较完备、高效、与劳动和社会保障事业发展相适应、与国家经济信息系统相衔接的劳动和社会保险管理信息系统，实现劳动和社会保险管理体系的技术现代化、管理科学化。

SIMIS 系统以 IC 卡为信息载体，完成劳动和社会保险的主要业务管理，即"五保合一"管理，包括养老保险、医疗保险、劳动就业和失业保险、工伤保险、女工生育保险。整个业务流程十分复杂，牵涉面相当广泛。SIMIS 系统由省、地市、县三级组成，网络纵向覆盖全省各级劳动和社会保障机构，横向与财税、银行、卫生、邮政、企事业单位联网，是一个典型的广域网络系统；系统设计按照社会保险与个人账户相结合的模式，以养老保险为重点，并以此为全省劳动和社会保险管理信息网络主干网络，带动劳动力市场等其他社会保险业务管理信息系统建设。

2．架构设计

虽然国家劳动和社会保障部对整个业务有一套规定的指导性流程，但是，在调研的过程中，我们发现各市、县都或多或少地存在使用"土政策"的情况，正是这种"土政策"致使软件在设计阶段具有很多的不确定性需求。另外，我们还考虑到将来用户需求可能会发生变化，为了尽量降低维护成本，提高可重用性，我们引入了层次式软件架构的设计思想。

SIMIS 系统采用层次式软件架构的基本出发点在于，这种软件架构不但能够满足不同规模的用户（县级，地、市级）的需求，可以方便地在最小的完成基本功能的基本系统和最大的完成所有复杂功能的扩充系统之间进行选择安装，而且通过逐层功能扩展的方法来进行软件实现，有利于程序设计和构件开发。同时，一定级别的抽象层可以作为一种知识积累，对于同类软件的快速开发有着很大的作用。

根据调研的结果，我们把 SIMIS 系统设计成具有通用核心层、基本应用层、业务管理层和扩展应用层四个层次的层次式软件体系构，如图 1 所示。

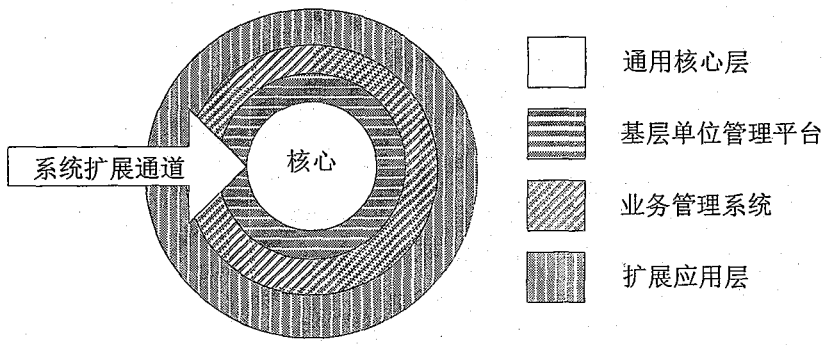


图 1　SIMIS 的层次式架构

通用核心层完成的是软件的一些通用的公共操作，这些操作能够尽量做到不与具体的数据库和表结构相关。基本应用层是 SIMIS 系统数据采集的主要来源；业务管理层是对基本应用层的进一步扩展，主要完成 SIMIS 系统的业务管理，管理内容涉及劳动者个人、企业和其它劳动组织的微观信息，能够实现数据的初步汇总。业务管理层与其内包含的两层一起构成了 SIMIS 的典型应用系统。扩展应用层是在典型应用系统的基础上扩充了一些更为复杂的功能，如对政策决策提供依据和支持，对政策执行状况进行监测、社会保险信息发布及个人账户电话语音查询系统等。

SIMIS 的设计和开发重点放在业务管理层上，在这一层的设计中，我们采用了互连系统构成的系统的架构，把整个业务管理系统划分为失业保险管理、养老保险管理、医疗保险管理、女工生育保险管理、工伤保险管理、工资收入管理、劳动关系管理、职业技能开发管理等八个从属系统，所有的从属系统共用同一个数据库管理系统，每个从属系统作为单独的系统，由不同的开发团队进行独立开发。

从属系统可以自成一个软件系统，脱离上级系统而运行，有其自己的软件生命周期，在生命周期内的所有活动中都可以单独管理，可以使用不同的开发流程来开发各个从属系统。

各从属系统的架构可以相同，也可以不同。例如，在开发“养老保险管理系统”这个从属系统时，选择了正交软件架构。我们将整个系统设计为三级正交架构，第一级划分为八个线索，每个一级线索又可划分为若干个二级线索，每个二级线索又可划分为若干个三级线索。一条完整线索如图 2 所示。

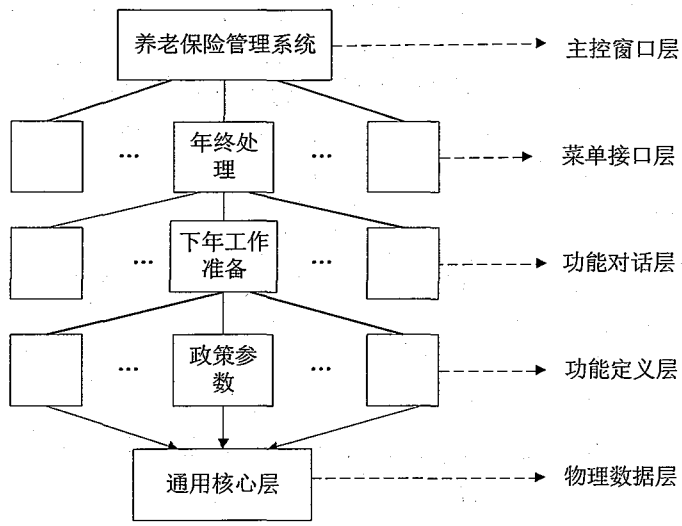


图 2　完整的一条线索架构

每一条线索完成整个系统中相对独立的一部分功能，所有线索是相互独立的，即不同线索中的构件之间没有相互调用。由于采用了正交架构的思想，在系统开发时，我们分成若干个小组并行开发，视开发难度情况，每个小组负责一条或数条线索，由一个小组来设计通用共享的数据存取构件。由于各条线索之间没有相互调用，所以各小组不会相互牵制，大大提高了编程的效率，缩短了开发周期，降低了工作量。

虽然各个从属系统相对独立，可以进行并行开发，但我们尽量注意了软件重用，以节约开发成本，加快开发进度。例如，“基金收缴”是失业保险、养老保险、医疗保险、女工生育保险和工伤保险五个从属系统中都要进行的操作，且其操作流程大致一样，只是基金收缴的参数有所区别。我们就只安排养老保险从属系统的开发团队开发一个通用构件，提供参数接口供其他从属系统使用。

3. 总结

在 SIMIS 系统的架构设计中，我们引入了层次式软件架构的设计思想。根据调研的结果，把 SIMIS 系统设计成具有通用核心层、基本应用层、业务管理层和扩展应用层四个层次。SIMIS 的设计和开发重点放在业务管理层上，在这一层的设计中，采用了互连系统构成的系统的架构，把整个业务管理系统划分为八个从属系统。每个从属系统作为单独的系统，由不同的开发团队进行独立开发。各从属系统的架构可以相同，也可以不同。

以上设计有效地降低了维护成本，提高了系统的开放性、可扩充性、可重用性和可移植性。但 SIMIS 的设计和开发也存在一些不足。例如，由于采用了互连系统构成的系统的软件架构，使资源管理开销增大，各从属系统的开发进度无法同步等。又如，由于采用了 B/S 和 C/S 架构混合的异构架构，使外部用户修改和维护数据时，速度较慢，较烦琐，数据的动态交互性不强等。

综合知识练习题参考答案

企业信息化战略与实施

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
B	C		C	C	B	C	D	B	C
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
A	D	A	D	C	C	D	A	C	D
(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)		
A	B	C	B	C	C	D	B		

软件工程

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
C	A	B	D	C	B	D	B	C	C
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
C	B	C	B	B	D	C	A	A	D
(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)
A	C	A	B	C	A	C	B	D	D
(31)	(32)	(33)	(34)	(35)	(36)	(37)	(38)	(39)	(40)
C	D	A	C	A	D	A	C	A	B
(41)	(42)	(43)	(44)	(45)					
C	A	D	B	C					

软件架构设计（第 1 部分）

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
C	C	C	A	D	C	D	A	B	C
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
A	C	B	A	B	C	D	B	A	C
(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)
B	B	C	C	A	C	C	C	D	B

软件架构设计（第 2 部分）

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
A	C	A	A	B	B	B	C	C	A
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
D	C	D	A	C	D	B	C	C	B
(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)
C	B	C	C	C	A	B	D	A	D

(31)	(32)	(33)	(34)	(35)	(36)				
B	C	C	B	D	B				

系统安全分析与设计

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
D	C	C	B	C	D	D	B	D	B
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)		
A	C	D	D	D	C	A	B		

项目管理

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
A	D	B	D	B	D	D	B	A	A
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
A	D								

计算机组成原理与体系结构

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
B	A	B	C	C	A	D	A	B	B
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
C	B								

系统配置与性能评价

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
D	B	A	D	B	C	A	D	B	D
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
C	C	A	A	B	D	A	B	C	A
(21)									
B									

操作系统

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
B	C	B	A	C	B	A	A	C	C
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
B	D	B	C	A	D	B	B	B	C
(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)
D									

计算机网络

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
A	C	A	C	A	A	A	C	D	B
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
D	B	C	C	A	B	C	A	A	B
(21)	(22)	(23)							
B	C	D							

数据库系统

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
B	C	C	B	C	B	A	C	D	D
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
D	A	A	C	D	D	C	A	A	B
(21)									
C									

数学与经济管理

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
C	B	D	B	B	C	D	C	C	C
(11)	(12)	(13)	(14)	(15)	(16)	(17)			
D	C	B	B	D	C	A			

知识产权与标准化

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
B	C	D	B	C	D	B	C	B	A
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
B	D	A	C	B	C	C	A	B	

2019年下半年系统架构设计师考试模拟试卷
综合知识

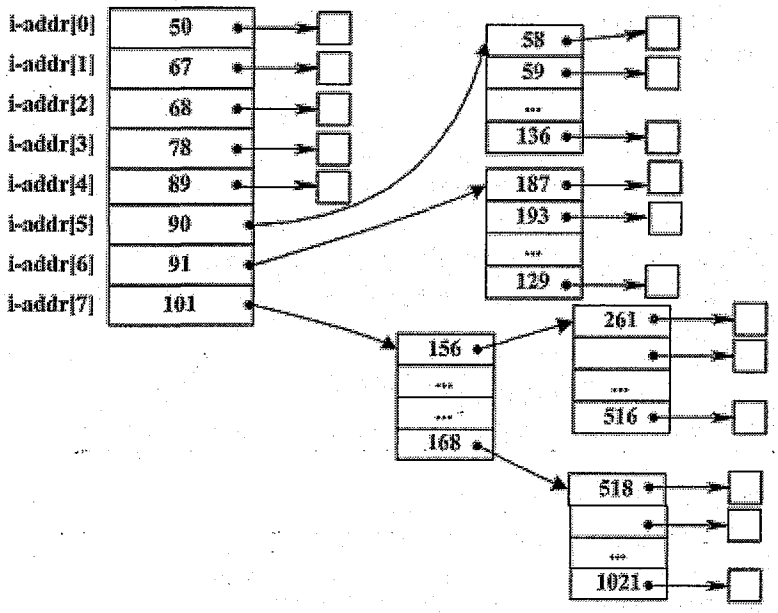
1、设系统仅有一类数量为M的独占型资源，系统中N个进程竞争该类资源，其中各进程对该类资源的最大需求量为W。当M、N、W分别取下列值时，（ ）可能会发生死琐。

A、M=5，N=3，W=2　　B、M=3，N=2，W=2　　C、M=2，N=2，W=1　　D、M=3，N=2，W=3

2、文件系统从单级目录调整为多级目录结构的目的是　（ ）。

A、减少系统开销　　B、节省存储空间　　C、解决命名冲突　　D、缩短查找时间

3-4、假设文件系统采用索引节点管理，且索引节点有8个地址项iaddr[0]～iaddr[7]，每个地址项大小为4字节，iaddr[0]～iaddr[4]采用直接地址索引，iaddr[5]和iaddr[6]采用一级间接地址索引，iaddr[7]采用二级间接地址索引。假设磁盘索引块和磁盘数据块大小均为1KB字节，文件File1的索引节点如下图所示。若用户访问文件File1中逻辑块号为516的信息，则对应的物理块号为（　）；90号物理块存放的是（　）。



A、136　　B、193　　C、129　　D、516

A、File1的信息　　B、直接地址索引表　　C、一级地址索引表　　D、二级地址索引表

5、以下是关于VxWorks嵌入式操作系统几个典型观点，选项（ ）代表的观点是正确的。

①操作系统与应用程序处于不同存储空间　　②支持进程间隔离保护

③支持多任务（线程）操作　　④硬实时系统

A、①、③　　B、②、④　　C、③、④　　D、①、②、③

6、设关系R是M元关系，关系S是N元关系，则关系R和S的笛卡儿积为（ ） 元关系。

A、M　　B、N　　C、M*N　　D、M+N

7、关系模式A(C，T，H，R，S)的候选键有（C，R，S）和（T，H），则说明该关系模式最高范式至少是（ ）。

A、1NF　　B、2NF　　C、3NF　　D、BCNF

8、采用三级结构/两级映像的数据库体系结构，如果对数据库的一张表创建聚簇索引，改变的是数据库的（　）。

A、用户模式　　B、外模式　　C、模式　　D、内模式

9、某水库现在的水位已超过安全线，上游河水还在匀速流入。为了防洪，可以利用其 10 个泄洪闸（每个闸的泄洪速度相同）来调节泄洪速度。经测算，若打开 1 个泄洪闸，再过 10 个小时就能将水位降到安全线；若同时打开 2 个泄洪闸，再过 4 个小时就能将水位降到安全线。现在抗洪指挥部要求再过 1 小时必须将水位降到安全线，为此，应立即同时打开（　）个泄洪闸。

A、6　　B、7　　C、8　　D、9

10、雷达设计人员在设计数字信号处理单元时，其处理器普遍采用 DSP 芯片(比如：TI 公司的 TMS320C63xx)，通常 DSP 芯片采用哈佛(HarVard)体系结构，以下关于哈佛结构特征的描述，不正确的是（ ）。

A、程序和数据具有独立的存储空间，允许同时取指令和取操作数，并允许在程序空间或数据空间之间互传数据

B、处理器内部采用多总线结构，保证了在一个机器周期内可以多次访问程序空间和数据空间

C、哈佛体系结构强调的是多功能，适合多种不同的环境和任务，强调兼容性

D、处理器内部采用多处理单元，可以在一个指令周期内同时进行运算

11、总线是计算机各种功能部件之间传送信息的公共通信干线。如果我们复用总线，可以（　）。

A、减少总线中信号线的数量　　B、提高总线的传输带宽

C、增加总线的功能　　D、提高 CPU 的利用率

12、嵌入式系统硬件初始化过程分为 3 个主要环节，按照自下而上、从硬件到软件的次序依次是（　）。

A、板级→系统级→片级　B、系统级→板级→片级　C、片级→板级→系统级　D、系统级→片级→板级

13、以下关于系统性能评估方法的描述，错误的是（　）。

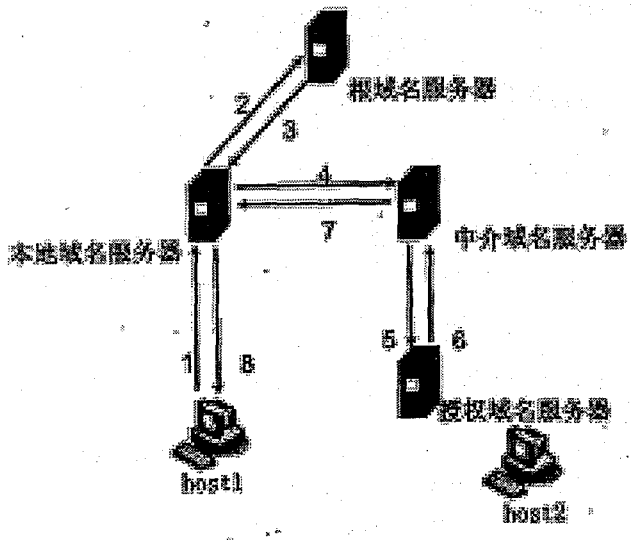
A、指令执行速度法常用每秒百万次指令运算（MIPS）评估系统性能

B、基准程序法主要针对 CPU（有时包括主存）的性能，但没有考虑 I/O 结构、操作系统、编译程序的效率等对系统性能的影响

C、等效指令速度法评估系统性能时需要计算各类指令在程序中所占的比例

D、综合理论性能法（CPT）采用每秒百万次理论运算（MTOPS）评估系统性能

14、主机 host1 对 host2 进行域名查询的过程如下图所示，下列说法中正确的是（　）。



A、本地域名服务器采用迭代算法　　B、中介域名服务器采用迭代算法

C、根域名服务器采用递归算法　　D、授权域名服务器采用何种算法不确定

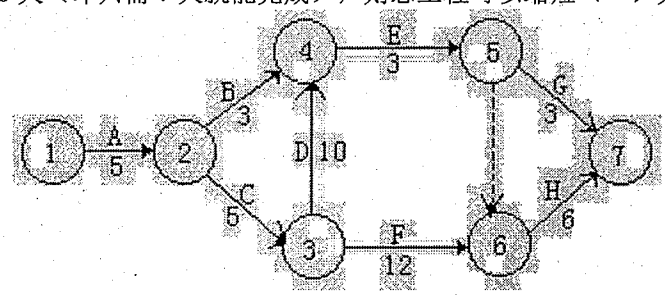
15、网络系统设计过程中，逻辑网络设计阶段的任务是（　）。
A、确定设备的具体物理分布和运行环境　　B、设计网络安全方案
C、进行软硬件和安装费用估算　　D、设计网络布线方案

16、下列有关网络设备选型原则中，不正确的是（　）。
A、所有网络设备尽可能选取同一厂家的产品，这样在设备可互连性、协议互操作性、技术支持、价格等方面都更有优势
B、在网络的层次结构中，主干设备选择可以不考虑扩展性需求
C、尽可能保留并延长用户对原有网络设备的投资，减少在资金投入上的浪费
D、选择性能价格比高、质量过硬的产品，使资金的投入产出达到最大值

17、结构化布线系统分为六个子系统，其中水平子系统（　）。
A、由各种交叉连接设备以及集线器和交换机等设备组成　　B、连接了干线子系统和工作区子系统
C、由终端设备到信息插座的整个区域组成　　D、实现各楼层设备间子系统之间的互连

18、某公司欲构建一个网络化的开放式数据存储系统，要求采用专用网络连接并管理存储设备和存储管理子系统。针对这种应用，采用（　）存储方式最为合适。
A、内置式存储　　B、DAS　　C、SAN　　D、NAS

19、某工程计划如下图所示，图中标注了完成任务 A～H 所需天数，其中，虚线表示虚任务。经评审后发现，任务 D 还可以缩短 3 天（即只需 7 天就能完成），则总工程可以缩短（　）天。



A、0　　B、3　　C、2　　D、1

20、配置管理系统中的静态系统，就是（　）。
A、开发库　　B、受控库　　C、产品库　　D、主库

21-22、《GB8567-1988 计算机软件产品开发文件编制指南》规定在一项软件开发过程中应该产生 14 种文件。其中开发进度月报主要（　）使用，软件需求说明书主要由（　）使用。
A、维护人员　　B、开发人员　　C、管理人员　　D、设计入员
A、维护人员　　B、开发人员　　C、管理人员　　D、设计入员

23-24、软件评审是保证软件质量的重要措施，它是对（　）　阶段成果的复审。下列　（　）是评审的必须内容。
A、软件准备　　B、软件开发　　C、软件实施　　D、软件维护
A、需求复审、总体设计复审、详细设计复审　　B、总体设计复审、详细设计复审、程序复审
C、需求复审、详细设计复审、程序复审　　D、需求复审、总体设计复审、详细设计复审、程序复审

25、软件需求说明书是软件需求分析阶段的重要文件，下述（　）是其应包含的内容。
①　数据描述　　②　功能描述　　③　模块描述　　④　性能描述
A、①②　　B、③④　　C、①②③　　D、①②④

26、基于 JavaEE 平台的基础功能服务构建应用系统时，（　　）可用来集成遗产系统。
A、JDBC、JCA 和 Java IDL　　B、JDBC、JCA 和 JMS
C、JDBC、JMS 和 Java IDL　　D、JCA、JMS 和 Java IDL

27-28、企业信息集成按照组织范围分为企业内部的信息集成和外部的信息集成。在企业内部的信息集成中，（　）实现了不同系统之间的互操作，使得不同系统之间能够实现数据和方法的共享：（　）实现了不同应用系统之间的连接、协调运作和信息共享。
A、技术平台集成　　B、数据集成　　C、应用系统集成　　D、业务过程集成
A、技术平台集成　　B、数据集成　　C、应用系统集成　　D、业务过程集成

29、企业中的信息系统一般可分为面向作业处理的系统、面向管理控制的系统与面向决策的系统这三类。下列选项中（　）不属于面向决策计划的系统。
A、知识工作支持系统（KWSS）　　B、战略信息系统（SIS）
C、管理专家系统（MES）　　D、决策支持系统（DSS）

30-31、基于 RUP 的软件过程是一个迭代过程。一个开发周期包括初始、细化、构建和移交四个阶段，每次通过这四个阶段就会产生一代软件，其中建立完善的架构是（　）阶段的任务。采用迭代式开发，（　）。
A、初始　　B、细化　　C、构建　　D、移交
A、在每一轮迭代中都要进行测试与集成　　B、每一轮迭代的重点是对特定的用例进行部分实现
C、在后续迭代中强调用户的主动参与　　D、通常以功能分解为基础

32、在结构型的瀑布模型中，（　）阶段定义的标准将成为软件测试中系统测试阶段的目标。
A、需求分析阶段　　B、概要设计阶段　　C、详细设计阶段　　D、可行性研究阶段

33、软件系统设计的主要目的是为系统制定蓝图，（　）并不是软件设计模型所关注的。
A、系统总体结构　　B、数据结构　　C、界面模型　　D、项目范围

34、某银行为了使其网上银行系统能够支持信用卡多币种付款功能而进行扩充升级，这需要对数据类型稍微进行一些改变，这一状况需要对网上银行系统进行（　）维护。
A、正确性　　B、适应性　　C、完善性　　D、预防性

35、以下关于软件测试的叙述中，正确的是（　）。
A、软件测试不仅能表明软件中存在错误，也能说明软件中不存在错误
B、软件测试活动应从编码阶段开始
C、一个成功的测试能发现至今未发现的错误
D、在一个被测程序段中，若已发现的错误越多，则残存的错误数越少

36-37、面向对象系统的单元测试包括方法层次的测试、类层次的测试和类树层次的测试。在常见的测试技术中，（　）属于方法层次的测试，（　）属于类层次的测试。
A、等价类划分测试和多态消息测试　　B、不变式边界测试和递归函数测试
C、组合功能测试和非模态类测试　　D、不变式边界测试和模态类测试
A、等价类划分测试和多态消息测试　　B、不变式边界测试和递归函数测试
C、组合功能测试和非模态类测试　　D、不变式边界测试和模态类测试

38、希赛教育计划研发一种语音识别软件系统，用于手机端的简单功能控制，语音识别软件需要对学员发出的语音指令进行音节分割、重音判断、语法分析和语义分析，最终对用户的意图进行推断。针对上述功能需求，该语音识别软件应该采用（　）架构风格最为合适。
A、解释器　　B、管道-过滤器　　C、以数据为中心　　D、隐式调用

39、系统管理员张工一直负责公司 Unix 服务器的维护工作，现在他要用 Unix shell 编写一个小的管理程序，实现目录文件的查看并分页显示的功能，（ ）架构风格最为合适。
A、隐式调用　　B、C2 风格　　C、黑板风格　　D、管道过滤器风格

40、软件的分层式体系结构把软件系统划分为 4 层，这 4 层结构自底向上分别是（ ）。
A、系统软件、中间件、应用软件、业务构件　　B、系统软件、应用软件、中间件、业务构件
C、应用软件、中间件、系统软件、业务构件　　D、系统软件、中间件、业务构件、应用软件

41、特定领域软件架构（DSSA）是一个特定的问题领域中由领域模型、参考需求和参考架构。以一下关于 DSSA 所具备的特征描述中，错误的是（ ）。
A、对整个领域能有合适程度的抽象　　B、具备该领域固定的、典型的在开发过程中的可重用元素
C、一个严格定义的问题域和 / 或解决域　　D、具有易用性，使其可以用于领域中某个特定应用的开发

42、希赛公司承接了 A 公司车辆定速巡航控制系统的研发任务。希赛公司的领域专家对需求深入分析后，将系统需求认定为：任何时刻，只要定速巡航控制系统处于工作状态，就要有确定的期望速度，并通过调整引擎油门的设定值来维持期望速度。若采用面向对象的架构分格对车辆定速巡航控制系统进行设计，则需要定义除（ ）之外的构件。
A、油门　　B、闭环控制器　　C、速度计　　D、时钟

43、注册表是 Windows 操作系统的核心文件，用于存储系统软硬件配置信息。在此之前，软硬件配置信息都是采用的.ini 文件单独存储的。注册表的引入，依赖于（ ）架构风格思想。
A、调用-返回　　B、虚拟机　　C、数据为中心　　D、独立构件

44、Architecture Tradeoff Analysis Method (ATAM)是一种软件架构的评估方法，以下关于该方法的叙述中，错误的是 （ ） 。
A、对于特定的软件体系结构，在系统开发之前，可以使用 ATAM 方法确定在多个质量属性之间进行评价和折中
B、在场景和需求收集有关的活动中，ATAM 方法需要所有系统相关人员的参与
C、ATAM 分析评估体系结构的过程包括场景开发、体系结构描述、单个场景评估、场景交互和总体评估等 5 个步骤
D、ATAM 方法可视为一个依赖于质量属性的框架，该框架可以使用不同的分析技术

45、架构设计、文档化和复审是一个迭代的过程。架构复审是基于架构开发中一个重要的环节。以下不属于架构复审的目标是（ ）。
A、评估和测试架构是否满足需求，是否存在可识别的技术和协作风险
B、质量需求是否在设计中得到体现，层次是否清晰
C、构件的划分是否合理，构件的设计是否满足功能与性能的要求
D、文档表达是否简练，能否使用系统演化步骤去修改应用以满足新的需求

46、以下质量属性属于商业质量属性的是（ ）。
A、性能　　B、安全　　C、成本　　D、可靠

47-52、希赛教育计划对在线辅导平台进行升级。在架构设计阶段，公司的架构师识别出一些核心的质量属性场景，其中“用户在点播教学视频时，若视频缓存 60 秒后仍不能正常播放，则认定为主站点故障，此时要求在 30 秒钟内自动切换到第三方合作机构视频站点，确保用户能正常点播”主要与（ ）质量属性相关，通常可采用（ ）架构策略实现该属性；“在并发学员数不超过 800 人时，用户的点播缓存时间要求在 5s 内完成”主要与（ ）质量属性相关通常可采用（ ）架构策略实现该属性；“在线点播系统应该能够抵挡一般恶意用户的入侵行为，并通过邮件与短信进行报警，及时通知管理员”主要与（ ）质量属性相关，通常可采用（ ）架构策略实现该属性。
A、性能　　B、安全性　　C、可用性　　D、可维护性
A、追踪审计　　B、信息隐藏　　C、资源调度　　D、主动冗余

A、可测试性　　B、可维护性　　C、性能　　D、易用性
A、资源调度　　B、操作串行化　　C、心跳　　D、内置监控器
A、可用性　　B、安全性　　C、可测试性　　D、可修改性
A、内置监控器　　B、记录/回放　　C、追踪审计　　D、维护现有接口

53-55、希赛公司决定采用基于架构的软件开发方法，以架构为核心对近期所承接的某电子商务企业的在线交易平台进行设计与实现。该电子商务企业要求在线交易平台必须保证客户个人信息和交易信息的安全；当发生故障时，该平台的平均故障恢复时间(MTB F)必须小于5分钟。该电子商务企业所提的两个要求分别对应于（ ）软件质量属性；实现并提高前一个软件质量属性的架构设计策略是（ ）；实现并提高后一个软件质量属性的架构设计策略不包括（ ）。
A、健壮性、可测试性　　B、安全性、可用性　　C、可靠性、可变性　　D、功能性、性能
A、设置检查点或回滚　　B、限制模块之间通信
C、采用身份认证和数字签名策略　　D、将接口与实现分离
A、添加适配器　　B、心跳　　C、备用件　　D、热重启

56、以下关于设计模式的描述，错误的是（ ）。
A、计模式是一些优雅的解决方案，是在大量实践经验的基础上提炼出来的。
B、设计模式通常依赖于某个特定的系统类型、程序设计语言或应用领域，它们是应用于某种特定场合的解决方案。
C、设计模式的建档方式使它们非常易于使用，因而可方便用于任何适宜的系统。
D、设计模式通常都非常简单，只涉及很少的一些类。为了构建更多更复杂的解决方案，可以把不同的设计模式与应用代码结合或混合使用。

57、“红灯停，绿灯行”，是日常生活中最简单的交通规则，交通信号灯装点着我们的城市，指挥着日益拥挤的城市交通。若要设计一个系统模拟交通信号灯与其指挥的汽车，采用（ ）模式最合适。
A、创建者　　B、中介者　　C、观察者　　D、解释器

58-59、已知一个类可以处理以英制标准（英寸、英里等）表示的数据，现在需要处理一公制单位表示的数据，则可以使用（ ）模式来解决该问题。当（ ）时，可以使用该设计模式。
A、Adapter　　B、Decorator　　C、Delegation　　D、Proxy
A、对一个抽象的实现部分的修改对用户不产生影响
B、想使用一个已经存在的类，而它的接口不符合用户需求
C、一个系统要独立于它的产品创建、组合和表示
D、一个对象的改变需要同时改变其他对象

60、以下模式中，（ ）属于创建型模式。
A、访问者模式　　B、命令模式　　C、单例模式　　D、代理模式

61-62、在UML提供的系统视图中，（ ）是逻辑视图的一次执行实例，描述了并发与同步结构；（ ）是最基本的需求分析模型。
A、进程视图　　B、实现视图　　C、部署视图　　D、用例视图
A、进程视图　　B、实现视图　　C、部署视图　　D、用例视图

63、安全管理是信息系统安全能动性的组成部分，它贯穿于信息系统规划、设计、运行和维护的各阶段。安全管理中的介质安全属于（ ）。
A、技术安全　　B、管理安全　　C、物理安全　　D、环境安全

64、网络安全体系设计可从物理线路安全、网络安全、系统安全、应用安全等方面来进行，其中，数据库容灾属于（ ）。
A、物理线路安全和网络安全　　B、应用安全和网络安全
C、系统安全和网络安全　　D、系统安全和应用安全

65、以下用于在网络应用层和传输层之间提供加密方案的协议是（　　）。
A、PGP　　B、SSL　　C、 IPSec　　D、 DES

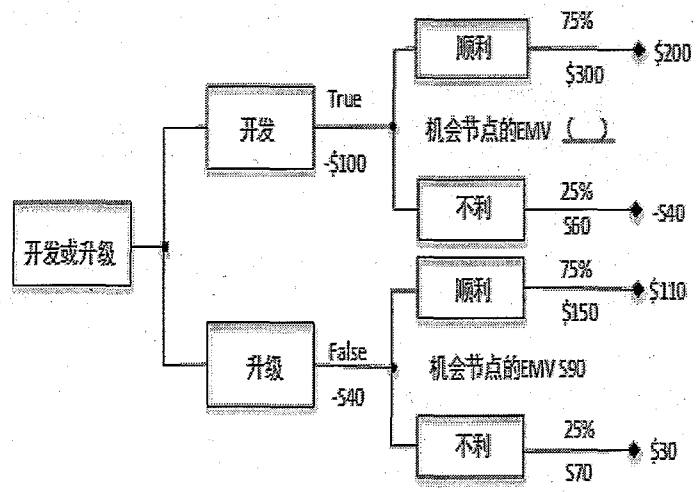
66、（　　）不是对称加密算法的优点。
A、加/解密速度快　　B、密钥管理简单
C、加密算法复杂，加密强度高　　D、适宜一对一的信息加密传输过程

67、在著作权法中，计算机软件著作权保护的对象包括（　　）。
A、硬件设备　　B、计算机软件开发文档
C、设备和操作系统软件　　D、源程序代码和底层环境

68、（　　）的保护期限是可以延长的。
A、专利权　　B、商标权　　C、著作权　　D、商业秘密权

69、刘某从画家孙某处购买了一幅画，刘某在对该画享有物权的同时，对该画还享有（　）。
A、复制权　　B、出租权　　C、展览权　　D、改编权

70、决策树分析法通常用决策树图表进行分析，根据下表的决策树分析法计算，图中机会节点的预期收益EMV 分别是$90 和$（　　）（单位：万元）。



A、160　　B、150　　C、140　　D、100

案例分析

试题一

阅读以下关于软件架构评估的说明，在答题纸上回答问题 1 和问题 2。

【说明】

希赛教育拟开发全新的在线教育平台，以改善学员学习体验。在系统需求分析与架构设计阶段，部分需求和关键质量属性场景如下：

（a）正常情况下，学员点击“开始考试”后的 0.3 秒内显示在线考试试题；

（b）对在线考试试题展示速度的要求将影响系统的数据存储和数据显示的设计；

（c）用户设置密码时，密码字符长度不低于 10 个字符，且必须含字母+数字；

（d）系统目前采用 MSSQL 数据库，将来可能变换为 Oracle 数据库，要求变更调整在 1 周内能完成；

（e）教学视频点播，画面具有 1024*768 以上的分辨率，点播时无明显卡顿；

（f）系统升级后，用户名必须达到 8 个字符；

（g）用户可用支付宝微信等多种支付方式进行学费支付，支付安全性达到 99.9999%；

（h）系统布署于云服务之上，以避免单台服务器故障带来的系统不能正常服务的现象产生；

（i）将来系统有集成第三方直播平台时，要求 1 个月内完成改造；

（j）目前管理层对于引入“翻转课程”教学模式尚未达成共识，这可能导致不少功能模块在开发中会有较大调整，引起频繁变更；

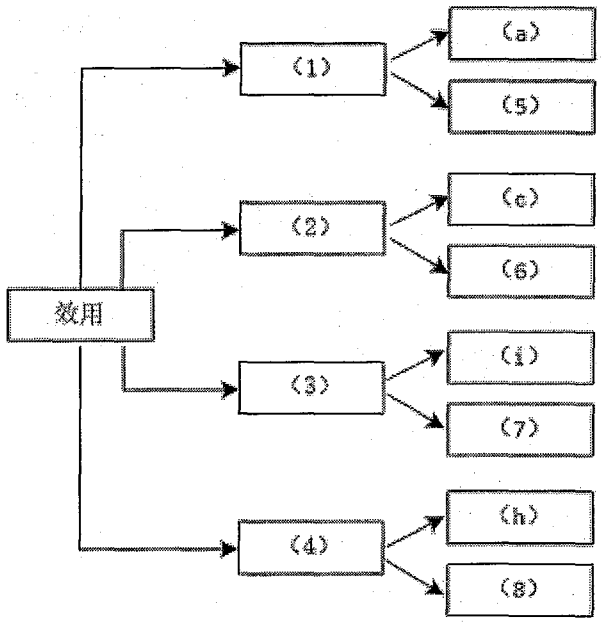
（k）系统需要提供远程调试接口，并支持系统的远程调试；

（l）教学视频点播系统采用双平台机制，A 点播平台出现故障不能点播时，自动切换至 B 点播平台；

（m）系统将引入 PKI/CA 体系，这会提高系统安全性，但同时会降低系统的性能。

【问题 1】（16 分）

在架构评估过程中，质量属性效用树（utility tree）是对系统质量属性进行识别和优先级排序的重要工具。请给出合适的质量属性（备选项：性能、可用性、安全性、可修改性），填入图中（1）、（2）、（3）、（4）空白处；并选择题干描述中的（a）～（m），将恰当的序号填入（5）～（8）空白处，完成该系统的效用树。



【问题 2】（9 分）

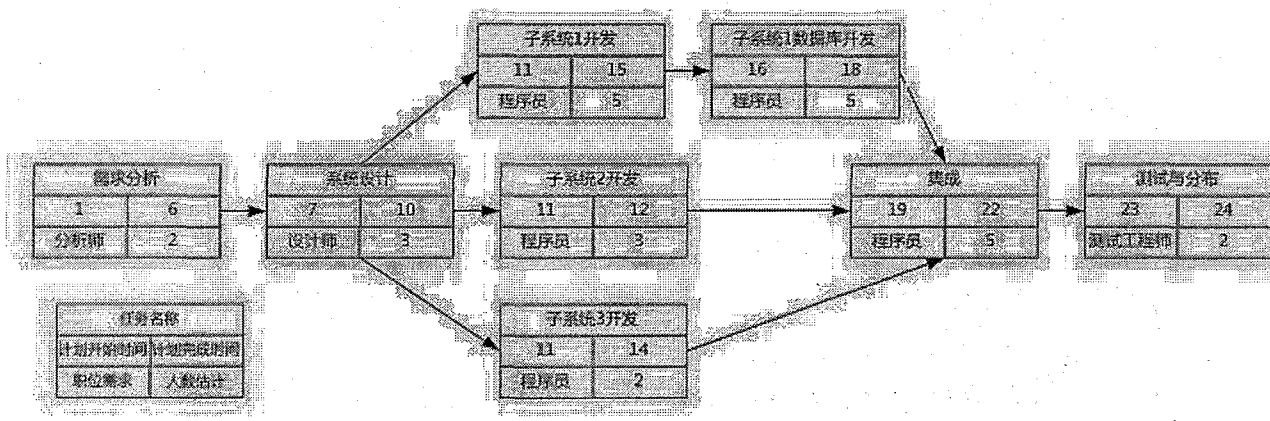
在架构评估过程中；需要正确识别系统的架构风险、敏感点和权衡点，并进行合理的架构决策。请用 300 字以内的文字给出系统架构风险、敏感点和权衡点的定义，并从题干描述中的（a）～（m）各选出 1 个属于系统架构风险、敏感点和权衡点的描述。

试题二

某公司计划完成一个工作量约为 100 人日的项目，公司技术人员人力成本如下表所示。

	分析师	设计师	程序员	测试工程师
日均成本（元）	500	300	200	200

项目经理根据项目的总体要求制定了某项目的网络资源计划图（图 1 所示，单位为日，为简化起见，不考虑节假日），并向公司申请了 2 名系统分析师负责需求分析，3 名设计师负责系统设计，10 名程序员负责子系统的开发和集成，2 名测试工程师负责系统测试和发布。项目经理估算总人力成本为 26700 元。



【问题 1】（6 分）
请指出项目经理在人力成本的估算中使用了哪些成本估算方法。
【问题 2】（3 分）
为了确保项目能按期完工，作为项目经理应该将主要精力集中于管理哪些项目活动。
【问题 3】（12 分）
第 12 日的工作结束时，项目组已完成需求分析、系统设计工作、子系统-1 的开发完成了五分之二、子系统-3 的开发完成了一半，其余工作尚未开展，此时人力部门统计应支付总人力成本 14000 元。请评价项目当前的进度绩效和成本绩效，并分析是否需要进行进度方面的调整，若需要，则给出具体调整措施。同时请预测原计划继续执行所需要的 ETC（完工尚需成本）。
【问题 4】（4 分）
假设每名项目组成员均可胜任分析、设计、开发、集成、测试和发布工作，在不影响工期的前提下，可重新安排有关活动的顺序以减少项目所需人数，此种情况下，该项目最少需要（1）人，子系统 3 的开发最晚应在第（2）日开始。

试题三
阅读以下关于嵌入式多核程序设计技术的描述，回答问题 1 至问题 3。
【说明】
近年来，多核技术已被广泛应用于众多安全关键领域(如:航空航天等)的电子设备中，面向多核技术的并行程序设计方法已成为软件人员急需掌握的主要技能之一。某宇航公司长期从事宇航电子设备的研制工作，随着宇航装备能力需求的提升，急需采用多核技术以增强设备的运算能力、降低功耗与体积，快速实现设备的升级与换代。针对面向多核开发，王工认为多核技术是对用户程序透明的，开发应把重点放在多核硬件架构和硬件模块设计上面，而软件方面，仅仅需要选择一款支持多核处理器的操作系统即可。而李工认为，多核架构能够使现有的软件更高效地运行，构建一个完善的软件架构是非常必要的。提高多核的利用率不能仅靠操作系统，还要求软件开发人员在程序设计 中考虑多进程或者多线程并行处理的编程问题。
【问题 1】(12 分)
请用 300 字以内文字说明什么是多核技术和多线程技术，并回答李工的意见是否正确，为什么?
【问题 2】(6 分)
在多核环境下，线程的活动有并行和并发两种方式，请用 300 字以内的文字说明这两种方式的含义及差别。

【问题 3】(7 分)
请根据自己所掌握的多核、多线程的知识，判别表 3-1 给出的说法是否正确，并将答案写在答题纸上对应空白处(填写正确或错误)。

表 3-1　关于多核和单核体系结构的说明

序号	说　明	是否正确
1	在面向多核体系结构开发应用程序时，只有有效地采用多线程技术并仔细分配各线程的工作负载才能够达到最高的性能	（1）
2	在面向多核平台设计多线程应用程序时，开发人员应当采用与面向单核平台时不同的设计思想	（2）
3	在多核平台上。多线程一般被当作是一种能够实现延迟隐藏的有效编程手段	（3）
4	多核平台为开发人员提供了一种优化应用程序的渠道，就是通过仔细分配加载到各线程（或各处理器核）上的工作负载就能够得到性能上的提升	（4）
5	在单核平台上，为了简化多线程应用程序的编写和调试，开发人员可能会做一些假设，这些假设也会适应于多核平台	（5）
6	在多核平台上，存储缓冲的 Cache 一致性问题是多核程序设计应当重点考虑的问题，但是，多核 Cache 的伪共享（False Sharing）问题在单核平台上也存在	（6）
7	在单核平台上，开发人员通常遵循优先级较高的线程不会受到优先级较低线程的干扰的思想对代码进行优化，这样的代码在多核平台上运行就会非常不稳定	（7）

试题四
请详细阅读有关数据架构方面的描述，回答问题 1 至问题 3。
【说明】
某软件公司欲开发一个基于 Web 2.0 的大型社交网络系统。就该系统的数据架构而言，李工决定采用公司熟悉的数据架构，使用通用的商用关系型数据库，系统内部数据采用中央集中方式存储。
该系统投入使用后，初期用户数量少，系统运行平稳。6 个月后，用户数出现了爆炸式增长，系统暴露出诸多问题，集中表现在：
（1）用户执行读写操作时，响应时间均变得很慢；
（2）随着系统功能的扩充，原有数据格式发生变化，又出现新的数据格式，维护困难；
（3）数据容量很快超过系统原有的设计上限，数据库扩容困难；
（4）软件系统不断出现宕机，整个系统可用性较差。
经过多次会议讨论，公司的王工建议采用 NoSQL 数据库来替代关系数据库，以解决上述问题。但李工指出 NoSQL 数据库出现时间不长，在使用上可能存在风险。公司技术人员对 NoSQL 数据库产品进行了认真测试，最终决定采用 NoSQL 数据库来替代现有的数据库系统。
【问题 1】（8 分）
分别解释产生问题（1）~（4）的原因。
【问题 2】（8 分）
请针对问题（1）~（4），分别指出 NoSQL 数据库的哪些特点促使公司最终采用了 NoSQL 数据库。
【问题 3】（9 分）

NoSQL 数据库具体来说，又分多种类型，Hbase 属于其中的列族数据库。请以 200 字以内的文字说明，列族数据库相对于传统的行式数据库，在进行大数据处理时有哪些优缺点。

试题五

【说明】

企业信息集成是解决“信息孤岛”问题的需要，由于“信息孤岛”的现象广泛存在，所以企业信息集成也为企业所重视。

企业集成的水平在很大程度上取决于企业内部各种系统、应用或服务的集成化运行水平，良好的软件支持工具可以帮助企业加快实现企业系统集成。作为支持企业集成化运行的使能工具，企业集成平台的主要功能是为企业中各种数据、系统、过程等多种对象的协同运行提供各种公共服务及运行时的支撑环境，从而降低实现企业内部的信息孤岛集成的复杂度，提高应用间集成的有效性，将信息系统实施规划中确定的企业中各种应用系统、服务、人员、信息资源及数字化设备的协同关系物化到集成化运行的可执行系统中去。

【问题 1】（10 分）

1、EAI 一般包括：（1）、数据集成、控制集成、（2）。其中（1）的主要作用是把各应用系统的界面集成起来，统一入口，使用户能够对集成系统产生一个“整体”的感觉。（3）为实现整体的业务目标，要定义、关联和管理不同的业务过程，并通过相应的业务信息系统中实现所需要的信息交换，从而降低成本，更高效地实现客户目标。并可以进行 B2B 集成。

2、数据集成主要有：数据联邦、数据复制和基于接口的数据集成三种模式。其中：（4）是指不同的应用共同访问一个全局虚拟数据库，通过全局虚拟数据库管理系统为不同的应用提供全局信息服务。（5）是指不同的应用系统之间利用适配器来实现相互调用以达到集成的目标。

【问题 2】（7 分）

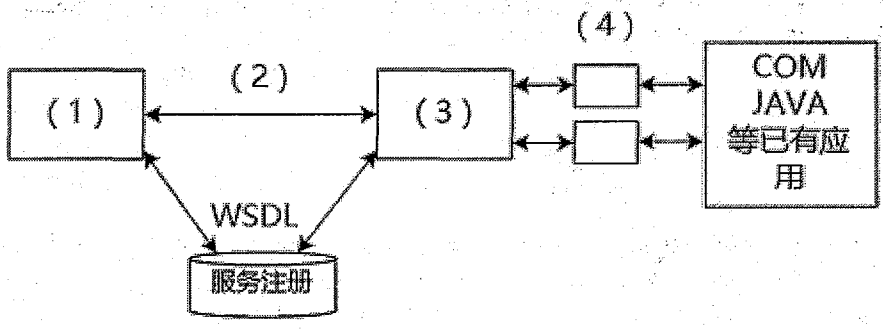
请简述应用之间开发一对一专用接口方式进行集成的优缺点。

【问题 3】（8 分）

面向服务的集成，经常会用到 WebService 技术对遗留系统进行集成。以下为面向服务的集成结构图，请使用以下词，根据自己的理解补充（1）~（4）。

A、ARP　　B、SOAP　　C、WSDL　　D、DCOM　　E、客户端

F、适配器　　G、服务提供者　　H、扩展开发接口



论文

试题一：论软件系统架构风格

系统架构风格（System Architecture Style）是描述某一特定应用领域中系统组织方式的惯用模式.架构风格定义了一个词汇表和一组约束，词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的口软件系统架构风格反映了领域中众多软件系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。软件系统架构风格的共有部分可以使得不同系统共享同一个实现代码，系统能够按照常用的、规范化的方式来组织，便于不同设计者很容易地理解系统架构。

请以“软件系统架构风格”论题，依次从以下三个方面进行论述：

1．概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。

2．分析软件系统开发中常用的软件系统架构风格有哪些？详细阐述每种风格的具体含义。

3．详细说明在你所参与的软件系统开发项目中，采用了哪种软件系统架构风格，具体实施效果如何。

论文二：论面向服务的架构及其应用

面向服务的架构（Service-Oriented Architecture,SOA）是一种组件模型，把应用程序中的不同功能单元（即服务）通过这些服务之间定义良好的接口和契约联系起来，使得这些系统中的服务能够以-种统一和通用的方式进行交互。从应用角度看，SOA 是一种应用框架，它关注企业日常的业务应用，将其划分为单独的业务功能和流程，并抽象为服务，用户和系统开发人员可以构建、部署和整合这些服务，无需依赖特定的应用程序及应用平台，从而提高企业业务流程的灵活性。SOA 有助于实现更多的信息资产重用、更轻松地管理和更快地应用开发与部署。

请以“面向服务的架构及其应用”为题，依次从以下三个方面进行论述：

1. 概要叙述你参与实施的、基于面向服务架构的软件开发项目以及所担任的主要工作。

2. 指出 SOA 技术参考架构中都包含哪些服务类别，并对每类服务的定义和作用进行简要说明。

3. 详细阐述你的项目是如何以面向服务的架构为指导进行实施的，在实施过程中遇到了哪些问题，是如何解决的。

论文三：论高可靠性系统中软件容错技术的应用

容错技术是当前计算机领域研究的热点之一，是提高整个系统可靠性的有效途径，许多重要行业（如航空、航天、电力、银行等）对计算机系统提出了高可靠、高可用、高安全的要求，用于保障系统的连续工作，当硬件或软件发生故障后，计算机系统能快速完成故障的定位与处理，确保系统正常工作。

对于可靠性要求高的系统，在系统设计中应充分考虑系统的容错能力，通常，在硬件配置上，采用了冗余备份的方法，以便在资源上保证系统的可靠性。在软件设计上，主要考虑对错误（故障）的过滤、定位和处理，软件的容错算法是软件系统需要解决的关键技术，也是充分发挥硬件资源效率，提高系统可靠性的关键。

请围绕“高可靠性系统中软件容错技术的应用”论题，依次从以下三个方面进行论述。

1．简述你参与设计和开发的、与容错相关的软件项目以及你所承担的主要工作。

2．具体论述你在设计软件时，如何考虑容错问题，采用了哪几种容错技术和方法。

3．分析你所采用的容错方法是否达到系统的可靠性和实时性要求。

论文四：论基于构件的软件开发

软件系统的复杂性不断增长、软件人员的频繁流动和软件行业的激烈竞争迫使软件企业提高软件质量、积累和固化知识财富，并尽可能地缩短软件产品的开发周期。

集软件复用、分布式对象计算、企业级应用开发等技术为一体的“基于构件的软件开发”应运而生，这种技术以软件架构为组装蓝图，以可复用软件构件为组装模块，支持组装式软件的复用，大大提高了软件生产效率和软件质量。

请围绕“基于构件的软件开发”论题，依次从以下三个方面进行论述。

1.简述你所参与开发的运用了构件技术的项目，以及你所担任的工作；

2.论述你在项目中如何运用构件技术来进行软件开发；

3.分析并讨论各种构件技术的优点、缺点，并展望构件技术的发展趋势。

2019年下半年系统架构设计师考试模拟试卷（答案+解析）

综合知识

1	2	3	4	5	6	7	8	9	10
D	C	C	C	C	D	C	D	B	B
11	12	13	14	15	16	17	18	19	20
A	C	B	D	B	B	B	C	D	C
21	22	23	24	25	26	27	28	29	30
C	B	B	D	D	A	C	D	A	B
31	32	33	34	35	36	37	38	39	40
A	A	D	C	C	A	D	C	D	D
41	42	43	44	45	46	47	48	49	50
D	B	C	C	D	C	C	D	C	A
51	52	53	54	55	56	57	58	59	60
B	C	B	C	A	B	C	A	B	C
61	62	63	64	65	66	67	68	69	70
A	D	C	D	B	C	B	B	C	C

案例分析

试题1答案：

【问题1】（每空2分）

（1）性能　　（2）安全性　　（3）可修改性　　（4）可用性

（5）（e）　　（6）（g）　　（7）（d）　　（8）（l）

【问题2】（定义1分1个，选择2分1个）

风险点：系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。

敏感点：是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点：是影响多个质量属性的特性，是多个质量属性的敏感点。

风险点：（j）；

敏感点：（b）；

权衡点：（m）。

试题解析：

（a）正常情况下，学员点击“开始考试”后的0.3秒内显示在线考试试题；属于性能。

（b）对在线考试试题展示速度的要求将影响系统的数据存储和数据显示的设计；属于敏感点。

（c）用户设置密码时，密码字符长度不低于10个字符，且必须含字母+数字；属于安全性。

（d）系统目前采用MSSQL数据库，将来可能变换为Oracle数据库，要求变更调整在1周内能完成；属于可修改性。

（e）教学视频点播，画面具有1024*768以上的分辨率，点播时无明显卡顿；属于性能。

（f）系统升级后，用户名必须达到8个字符；属于业务需求。

（g）用户可用支付宝微信等多种支付方式进行学费支付，支付安全性达到99.9999%；属于安全性。

（h）系统布署于云服务之上，以避免单台服务器故障带来的系统不能正常服务的现象产生；属于可用性。

（i）将来系统有集成第三方直播平台时，要求1个月内完成改造。属于可修改性。

（j）目前管理层对于引入“翻转课程”教学模式尚未达成共识，这可能导致不少功能模块在开发中会有较大调整，引起频繁变更；属于风险点。

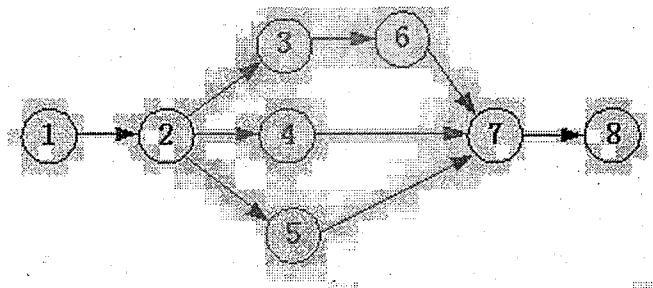
（k）系统需要提供远程调试接口，并支持系统的远程调试。属于可测试性。

（l）教学视频点播系统采用双平台机制，A 点播平台出现故障不能点播时，自动切换至 B 点播平台；属于可用性。
（m）系统将引入 PKI/CA 体系，这会提高系统安全性，但同时会降低系统的性能；属于权衡点。

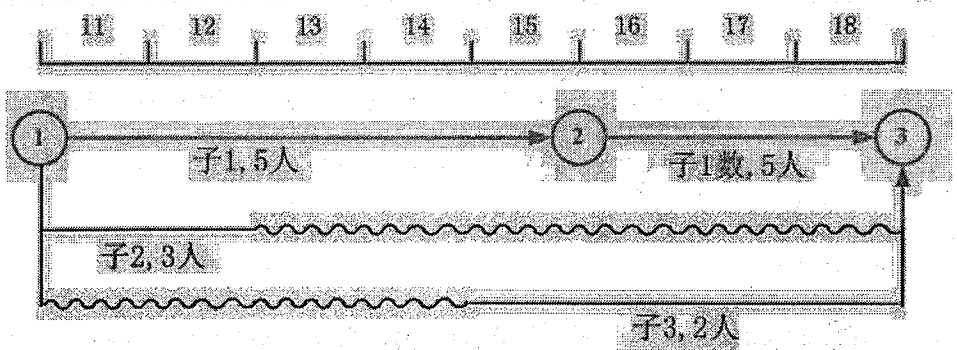
试题 2 答案：
【问题 1】（6 分）
确定资源费率、自下而上估算、参数估算 （答对 1 个给 3 分，最高 6 分）
【问题 2】（3 分）
关键活动即关键路径上的活动，所以包括：需求分析、系统设计、子系统 1 开发、子系统 1 数据库开发、集成、测试与发布。（评分规则，全部答对得 3 分，否则 0 分）
【问题 3】（12 分）
据题意，第 12 日的工作结束时：
应该完成的工作：需求分析、系统设计、子系统-1 的 2/5、子系统-3 的 1/2、子系统-2。
实际完成的工作：需求分析、系统设计、子系统-1 的 2/5、子系统-3 的 1/2
所以：
PV=需求分析+系统设计+子系统-1 的 2/5+子系统-3 的 1/2+ 子系统-2
=500(元/人*天)*2(人)*6(天)+300(元/人*天)*3(人)*4(天)+200(元/人*天)*5(人)*2(天)+200(元/人*天)*2(人)*2(天)+200(元/人*天)*3(人)*2(天)
=13600（元）（2 分）
EV=需求分析+系统设计+子系统-1 的 2/5+子系统-3 的 1/2
=500(元/人*天)*2(人)*6(天)+300(元/人*天)*3(人)*4(天)+200(元/人*天)*5(人)*2(天)+200(元/人*天)*2(人)*2(天)
=12400（元）（2 分）
AC=14000（元）（2 分）
CPI=EV/AC=12400/14000=0.89
SPI=EV/PV=12400/13600=0.91
当前，成本超支、进度落后。（2 分）
经过分析，我们可以发现被延误的活动“子系统-2”并非关键活动，且他的总时差大于 2 天，所以他的延误目前并不会影响项目的按时完工，所以可以暂不调整。（2 分）
根据题目可知：BAC=26700。
完工尚余成本：ETC=(BAC-EV)= 26700-12400=14300（元） （2 分）
【问题 4】（4 分）
（1）8 人 （2）15 天

试题解析：
【问题 1】
本题为纯概念题，项目涉及的估算方法包括：确定资源费率、自下而上估算、参数估算。
【问题 2】
在一个项目中，项目经理需要重点关注的活动是关键路径上的活动，因为这些活动一旦被耽误，则整个项目将被延误。
而关键路径是一个图中最长的路径，本题中，最长路径为：需求分析、系统设计、子系统 1 开发、子系统 1 数据库开发、集成、测试与发布组成的路径，所以这些活动是关键活动。
【问题 3】
据题意，第 12 日的工作结束时：
应该完成的工作：需求分析、系统设计、子系统-1 的 2/5、子系统-3 的 1/2、子系统-2。
实际完成的工作：需求分析、系统设计、子系统-1 的 2/5、子系统-3 的 1/2
所以：
PV=需求分析+系统设计+子系统-1 的 2/5+子系统-3 的 1/2+ 子系统-2
=500(元/人*天)*2(人)*6(天)+300(元/人*天)*3(人)*4(天)+200(元/人*天)*5(人)*2(天)+200(元/人*天)*2(人)*2(天)+200(元/人*天)*3(人)*2(天)
=13600（元）

EV=需求分析+系统设计+子系统-1 的 2/5+子系统-3 的 1/2
=500(元/人*天)*2(人)*6(天)+300(元/人*天)*3(人)*4(天)+200(元/人*天)*5(人)*2(天)+200(元/人*天)*2(人)*2(天)
=12400（元）
AC=14000（元）
CPI=EV/AC=12400/14000=0.89
SPI=EV/PV=12400/13600=0.91
当前，成本超支、进度落后。
经过分析，我们可以发现被延误的活动“子系统-2”并非关键活动，且他的总时差大于 2 天，所以他的延误目前并不会影响项目的按时完工，所以可以暂不调整。
根据题目可知：
BAC=26700。
完工尚余成本：
ETC=(BAC-EV)= 29700-12400=14300（元）
【问题 4】
本问考查资源平衡，要在最短工期下进行资源平衡，则关键路径上的资源是不能调整的，所以可以把关键路径下的资源固定下来，然后再考虑其它活动的资源调整。将题目原图简化后，可以产生以下简图，由于关键路径是 1，2，3，6，7，8。而 1，2，7，8 这些路径是单线，没有并行执行的任务，所以这些活动并无调整的可能。我们只需要考虑中间部分节点的调整。



由于，完成 2 活动，已是第 10 天，所以只需要分析从 11 天到 18 天的情况即可。由题意可绘制出以下资源调整图：



通过图形分析，可以得知，最少需要 8 人。而子系统 3，最迟应在第 15 天开始。

试题 3 答案：
【问题 1】
多核是多微处理器核的简称，是将两个或更多的独立处理器封装在一起，集成在一个电路中。多核处理器是单枚芯片（也称为硅核），能够直接插入单一的处理器插槽中，但操作系统会利用所有相关的资源，将它的每个执行内核作为分立的逻辑处理器。通过在多个执行内核之间划分任务，多核处理器可在特定的时钟周期内执行更多任务。
多线程是指从软件或者硬件上实现多个线程并发执行的技术。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程，进而提升整体处理性能。

李工的意见是正确的。多核处理器虽然带来了强大的计算能力，但如果无法实现程序的并行，那么，大量计算资源将被闲置，造成巨大的浪费。所以在开发中，要注重多核编程技术，从并行计算、共享资源分布式计算、任务分解与调度、Lock-Free 编程等方面充分考虑相关问题，以便将多核处理器的性能充分发挥出来。

【问题 2】

如果某个系统支持两个或者多个动作（Action）同时存在，那么这个系统就是一个并发系统，即系统中同时有多个线程，可以认为是并发的情况。

如果某个系统支持两个或者多个动作同时执行，那么这个系统就是一个并行系统。当系统中有多个线程，并在同一时刻有两个，或两个以上的线程在运行状态，可认为是并行系统。

【问题 3】

（1）√　（2）√　（3）×　（4）√　（5）×　（6）×　（7）√

试题解析：

多核是多微处理器核的简称，是将两个或更多的独立处理器封装在一起，集成在一个电路中。多核处理器是单枚芯片（也称为硅核），能够直接插入单一的处理器插槽中，但操作系统会利用所有相关的资源，将它的每个执行内核作为分立的逻辑处理器。通过在多个执行内核之间划分任务，多核处理器可在特定的时钟周期内执行更多任务。

多线程是指从软件或者硬件上实现多个线程并发执行的技术。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程，进而提升整体处理性能。具有这种能力的系统包括对称多处理机、多核心处理器以及芯片级多处理或同时多线程处理器。在一个程序中，这些独立运行的程序片段叫作“线程”，利用它编程的概念就叫作“多线程处理”。

如果某个系统支持两个或者多个动作（Action）同时存在，那么这个系统就是一个并发系统。如果某个系统支持两个或者多个动作同时执行，那么这个系统就是一个并行系统。

对此，有一种通俗的解读：

你吃饭吃到一半，电话来了，你一直到吃完了以后才去接，这就说明你不支持并发也不支持并行。

你吃饭吃到一半，电话来了，你停了下来接了电话，接完后继续吃饭，这说明你支持并发。

你吃饭吃到一半，电话来了，你一边打电话一边吃饭，这说明你支持并行。

多核编程技术主要包括并行计算、共享资源分布式计算、任务分解与调度、Lock-Free 编程等内容。其中共享资源分布式计算、任务分解与调度是最重要的内容，也是大多数程序员未接触过的内容，许多并行算法都可以通过它们来实现。多核编程模式主要是提供一种多核并行与分布式编程的普遍方法，有了这些编程模式后，程序员不再需要去学习各种复杂的并行算法，它可以复用现有的串行算法，很容易地实现并行和分布式计算。在多核编程技术中，最重要的一点是如何将计算均匀分摊到各个 CPU 核上。

多核时代的到来，给程序员的编程思维带来了巨大的冲击和挑战。为了能够充分利用多核性能，程序员必须学会以分块的思维设计程序，以多进程或多线程的形式来编写程序。到底应该使用多进程还是多线程的形式来编写程序，是最让程序员感到困惑的问题之一，这些需要根据具体的应用来决定。在通常情况下，使用多线程进行多核编程比使用多进程有更大的优势，因为：

（1）线程的创建和切换开销比进程更小。

（2）线程之间通信的方式比较多，而且简单也更有效率。

（3）多线程有很多的基础库支持。

（4）多线程的程序比多进程的程序更容易理解和修改。

除了编程形式，使用多线程编程的动机也发生了改变。过去，Windows 程序员使用多线程的主要原因之一是为了提高用户程序运行效率，例如，在长时间的计算中提高 GUI、I/O 或者网络的响应速度。而在多核时代编写应用程序为了充分利用多个计算核心，缩短计算时间，或者在相同的时间段内计算更多任务。例如，在进行游戏编程时，通过多线程的方式把碰撞检测的计算分散到多个 CPU 内核，就可以大大缩减计算时间，也可以利用多核做更细致的检测计算，从而能够模拟更加真实的碰撞。

处理器所能交换的最小存储单元就是一个 cache 行，或者一个 cache 块。两个独立的 cache 在需要读取同一 cache 行时，会共享该 cache 行。但如果在其中一个 cache 中，该 cache 行被写入，而在另一个 cache 中该 cache 行被读取，那么即使读写的地址不相交，也需要在这两个 cache 之间移动该 cache 行。就像两个人同时在写一本日志的两个不同部分，两人的写入动作相互独立，但是除非将日志撕成两半，否则这两个人必须来回地互相传递这本日志。同样地，两个硬件线程在写入一个 cache 的不同部分时，互相竞争 cache，就像在进行乒乓球比赛。

试题 4 答案：
【问题 1】
其原因主要是：
（1）用户响应时间慢。大型社交网络系统要根据用户个性化信息来实时生成动态页面和提供动态信息，所以基本上无法使用动态页面静态化技术，因此数据库并发负载非常高，往往要达到每秒上万次读写请求。关系数据库应付上万次 SQL 查询还勉强可以，但是应付上万次 SQL 写数据请求，硬盘 I/O 就已经无法承受了。特别是涉及到多表连接操作，会导致响应变慢。
（2）数据格式变化。大型社交网络系统随着用户的使用，会不断地增加新的功能，导致原有数据格式发生变化，甚至出现新的数据格式。但关系数据库中采用元组方式组织数据，难以使用新型数据格式，难以维护。
（3）数据容量超过设计上限。对于大型社交网络系统，往往会在很短时间内产生海量数据。关系数据库多采用中央数据存储，使得数据容量受限于前期设计的上限，很难实现数据容量的横向扩展。
（4）系统可用性差：关系数据库采用中央数据存储，容易成为系统的性能瓶颈，单点故障很容易导致系统崩溃，负载过高往往导致系统出现宕机现象。
【问题 2】
针对问题（1），NoSQL 数据库支持高并发数据访问，性能较高。
针对问题（2），NoSQL 数据库的数据存储结构松散，能够灵活支持多种类型的数据格式。
针对问题（3），NoSQL 数据库能够支持海量数据的存储，且易于横向扩展。
针对问题（4），NoSQL 数据库基于分布式数据存储，不存在单点故障和性能瓶颈，系统可用性高。
【问题 3】（评分准则，答对 1 条得 2 分，最多得 9 分）
优点：
1、极高的装载速度 （最高可以等于所有硬盘 IO 的总和，基本是极限了）
2、适合大量的数据而不是小数据
3、实时加载数据仅限于增加（删除和更新需要解压缩 Block 然后计算然后重新压缩储存）
4、高效的压缩率，不仅节省储存空间也节省计算内存和 CPU。
5、非常适合做聚合操作。
缺点：
1、不适合扫描小量数据
2、不适合随机的更新
3、批量更新情况各异，有的优化的比较好的列式数据库（比如 Vertica）表现比较好，有些没有针对更新的数据库表现比较差。
4、不适合做含有删除和更新的实时操作。

试题解析：
本题主要考查考生对 NoSQL 数据库的理解与掌握情况。
【问题 1】
本问主要考查对题干（1）～（4）出现问题的分析与理解情况，考生一定要紧紧围绕大型社交网络的特点，即海量用户、大量实时操作等特征进行论述。
【问题 2】
本问题承接上面的问题，主要考查如何结合实际场景，描述如何采用 NoSQL 数据库解决上述问题。注意需要有的放矢，不要发散。
【问题 3】
优点：
1、极高的装载速度 （最高可以等于所有硬盘 IO 的总和，基本是极限了）
原因分析：
不读取无效数据：降低 I/O 开销，同时提高每次 I/O 的效率，从而大大提高查询性能。查询语句只从磁盘上读取所需要的列，其他列的数据是不需要读取的。例如，有两张表，每张表 100GB 且有 100 列，大多数查询只关注几个列，采用列存储，不需要像行存数据库一样，将整行数据取出，只取出需要的列。磁盘 I/O 是行存储的 1/10 或更少，查询响应时间提高 10 倍以上。
2、适合大量的数据而不是小数据

3、实时加载数据仅限于增加（删除和更新需要解压缩 Block 然后计算然后重新压缩储存）
4、高效的压缩率，不仅节省储存空间也节省计算内存和 CPU。
由于按列存储，是把相同类型的数据放在一起，这样压缩起来能针对数据本身的特点来进行，有利于得到更高的压缩比。
高压缩比：压缩比可以达到 5 ~ 20 倍以上，数据占有空间降低到传统数据库的 1/10 ，节省了存储设备的开销。
当数据库的大小与数据库服务器内存大小之比达到或超过 2:1 （典型的大型系统配置值）时，列存的 I/O 优势就显得更加明显；
GBase 8a 分析型数据库的独特列存储格式，对每列数据再细分为“数据包”。这样可以达到很高的可扩展性：无论一个表有多大，数据库只操作相关的数据包，性能不会随着数据量的增加而下降。通过以数据包为单位进行 I/O 操作提升数据吞吐量，从而进一步提高 I/O 效率。
由于采用列存储技术，还可以实现高效的透明压缩。
由于数据按列包存储，每个数据包内都是同构数据，内容相关性很高，这使得 GBase 8a 更易于实现压缩，压缩比通常能够达到 1:10 甚至更优。这使得能够同时在磁盘 I/O 和 Cache I/O 上都提升数据库的性能，使 GBase 8a 在某些场景下的运算性能比传统数据库快 100 倍以上。
GBase 8a 允许用户根据需要设置配置文件，选择是否进行压缩。在启用压缩的情况下 GBase 8a 根据数据的不同特性以及不同的分布状况，自动采用相应的压缩算法，如：
行程编码（适用于大量连续重复的数据，特别是排序数据）；
基于数据的差值编码（适用于重复率低，但彼此差值较小的数据列）；
基于位置的差值编码（适用于重复率高，但分布比较随机的数据列）。
5、非常适合做聚合操作。
缺点：
1、不适合扫描小量数据
2、不适合随机的更新
3、批量更新情况各异，有的优化的比较好的列式数据库（比如 Vertica）表现比较好，有些没有针对更新的数据库表现比较差。
4、不适合做含有删除和更新的实时操作。

试题 5 答案：
【问题 1】（10 分）
（1）表示集成（界面集成）　　（2）业务流程集成（过程集成）
（3）业务流程集成（过程集成）　　（4）数据联邦
（5）基于接口的数据集成模式
【问题 2】（7 分，答对 1 点给 2 分，答对 4 条以上得全分）
优点：
直观，当企业应用数量少时容易实现。
缺点：
1、工作量大
2、集成系统的维护费用高，系统升级与扩展困难
3、不易于标准化，由于接口数量多，给系统管理造成比较大的困难
4、一般只能解决应用系统之间的数据集成问题，难以用来支持过程集成和应用之间的协调
【问题 3】（8 分）
（1）客户端　（2）SOAP　（3）服务提供　（4）适配器

试题解析：
EAI 一般包括：表示集成（界面集成）、数据集成、控制集成（应用集成）、 业务流程集成（过程集成）。
（1）界面集成：把各应用系统的界面集成起来，统一入口，使用户能够对集成系统产生一个“整体”的感觉。

（2）数据集成：数据集成是应用集成和业务过程集成的基础，可以提供企业之间的信息共享能力。在集成以前，要对数据进行统一标识、分类，并进行元数据建模。这三个步骤完成后，就可以实现企业范围的数据共享和数据分布了。

（3）应用集成：这一水平的集成目的是指将多个应用系统进行“绑定”，使之象一个实时运行的系统一样接受信息输入和产生数据输出，实现多个系统功能的“叠加”。应用集成广泛用于 B2B 集成、在后端服务应用基础上建立的客户关系管理系统、集成多个应用的 Web 门户等等。在 ERP 应用实施后，也要经常进行与新的应用系统的集成。

（4）过程集成（业务流程集成）：为实现整体的业务目标，要定义、关联和管理不同的业务过程，并通过相应的业务信息系统中实现所需要的信息交换，从而降低成本，更高效地实现客户目标。BPI 的要素包括过程管理，过程建模和工作流。

数据集成主要有以下三种模式：数据联邦、数据复制和基于接口的数据集成。如图所示，它们分别描述了对多个异构数据源透明、一致访问的三种实现方法。

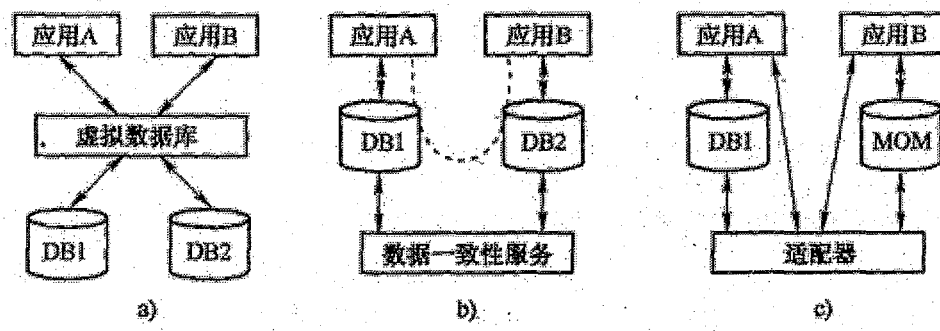


图　三种典型的数据集成模式

a）数据联邦　　b）数据复制　　c）基于接口的数据集成

（1）数据联邦

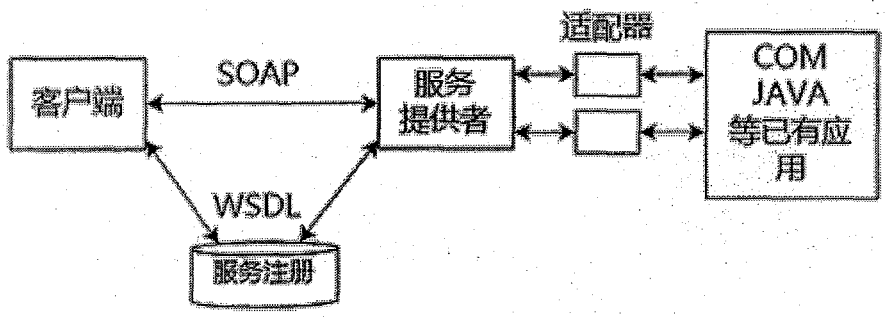
数据联邦是指不同的应用共同访问一个全局虚拟数据库，通过全局虚拟数据库管理系统为不同的应用提供全局信息服务，实现不同的应用和数据源之间的信息共享和数据交换，其具体实现由客户端应用、全局信息服务和若干个局部数据源三部分组成。

（2）数据复制模式

在数据复制模式中，通过底层应用数据源之间的一致性复制来实现（访问不同数据库的）不同应用之间的信息共享和互操作，其实现的关键是必须能够提供在两个或多个数据库系统之间实现数据转换和传输的基础结构（以屏蔽不同数据库间数据模型的差异）。

（3）基于接口的数据集成模式

在基于接口的数据集成模式中，不同的应用系统之间利用适配器（或接口代理）提供的应用编程接口来实现相互调用。应用适配器或接口代理通过其开放或私有接口将业务信息从其所封装的具体应用系统中提取出来，进而实现不同的应用系统之间业务数据的共享与互交换。接口调用的方式可以采用同步调用方法，也可以采用基于消息中间件的异步方法来实现。



论文

论文一

写作要点：

一、简要叙述所参与管理和开发的软件项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、常见的架构风格 5 大类，至少选 2-3 个类进行说明。

Garlan 和 Shaw 将软件架构风格分为五大类，数据流风格、调用/返回风格、独立构件风格、虚拟机风格和仓库风格。其中：

（1）数据流风格包括批处理序列架构风格和管道/过滤器架构风格；

（2）调用/返回风格包括主程序/子程序架构风格、数据抽象和面向对象架构风格和层次结构架构风格；

（3）独立构件风格包括进程通信架构风格和事件驱动的架构风格；

（4）虚拟机风格包括解释器架构风格和基于规则的系统；

（5）仓库风格包括数据库架构风格和黑板架构风格。

其他的还有特定领域软件架构、状态转移等以及分布式处理等。其中分布式架构风格中有客户机/服务器风格、浏览器/服务器风格、CORBA、DCOM、EJB。

每一种具体的软件结构风格的模型如下：

1．数据流风格包括批处理序列和管道/过滤器架构风格。

（1）批处理序列架构风格。组件为一系列固定顺序的计算单元，组件间只通过数据传递交互。每个处理步骤是一个独立的程序，每一步必须在前一步结束后才能开始，数据必须是完整的，以整体的方式传递。

（2）管道/过滤器架构风格。每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流，经过处理，产生输出数据流。这个过程通常通过对输入流的变换及增量计算来完成，包括通过计算和增加信息丰富数据，通过浓缩和删除精炼数据，通过改变记录方式转化数据，递增地转化数据等。在输入被完全消费之前，输出便产生了。这里构件被称为过滤器，连接件就是数据流传输的管道，将一个过滤器的输出传到另一过滤器的输入。

2．调用/返回风格包括主程序/子程序架构风格、数据抽象和面向对象架构风格以及层次结构架构风格

（1）主程序/子程序架构风格。单线程控制，把问题划分为若干处理步骤，构件即为主程序和子程序。子程序通常可合成为模块。过程调用作为交互机制，即充当连接件。调用关系具有层次性，其语义逻辑表现为子程序的正确性取决于它调用的子程序的正确性。

（2）数据抽象和面向对象架构风格。这种风格的构件是对象。对象是抽象数据类型的实例。在抽象数据类型中，数据的表示和它们的相应操作被封装起来。对象的行为体现在其接受和请求的动作。连接件即是对象间交互的方式，对象是通过函数和过程的调用来交互的。对象具有封装性，一个对象的改变不会影响其他对象。对象拥有状态和操作，也有责任维护状态。这种结构风格中包含有封装、交互、多态、集成、重用等特征。

（3）层次结构架构风格。层次系统组织成一个层次结构。构件在一些层实现了虚拟机。连接件通过决定层间如何交互的协议来定义，拓扑约束包括对相邻层间交互的约束。这个风格的特点是每层为上一层提供服务，使用下一层的服务，只能见到与自己邻接的层。大的问题分解为若干个渐进的小问题，逐步解决，隐藏了很多复杂度。修改一层，最多影响两层，而通常只能影响上层。上层必须知道下层的身份，不能调整层次之间的顺序。

3．独立构件风格包括进程通信架构风格和事件驱动的架构风格

（1）进程通信架构风格。构件是独立的过程，连接件是消息传递。这种风格的特点是构件通常是命名过程，消息传递的方式可以是点对点、异步和同步方式、以及远过程调用等

（2）事件驱动的架构风格。构件不直接调用一个过程，而是触发或广播一个或多个事件。系统中的其他构件中的过程在一个或多个事件中注册，当一个事件被触发，系统自动调用在这个事件中注册的所有过程。一个事件的触发就导致了另一个模块中的过程的调用。这种风格中的构件是非命名的过程，它们之间交互的连接件往往是以过程之间的隐式调用（Implicit Invocation）来实现的。基于事件的隐式调用风格的主要优点是为软件重用提供了强大的支持，为构件的维护和演化带来了方便，其缺点是构件放弃了对系统计算的控制。

4．虚拟机风格包括解释器架构风格和基于规则的系统

（1）解释器架构风格。一个解释器通常包括完成解释工作的解释引擎，一个包含将被解释的代码的存储区，一个记录解释引擎当前工作状态的数据结构，以及一个记录源代码被解释执行的进度的数据结构。具

有解释器风格的软件中含有一个虚拟机，可以仿真硬件的执行过程和一些关键应用。其缺点是执行效率较低。

（2）基于规则的系统。基于规则的系统包括规则集、规则解释器、规则/数据选择器以及工作内存。

5．仓库风格包括数据库架构风格和黑板架构风格

（1）数据库架构风格。数据库架构是库风格最常见的形式。构件主要有两大类，一个是中央共享数据源，保存当前系统的数据状态，另一个是多个独立处理元素，处理元素对数据元素进行操作。

（2）黑板架构风格。黑板架构包括知识源、黑板、控制三部分。知识源包括若干独立计算的不同单元，提供解决问题的知识，知识源响应黑板上的变化，也只修改黑板。黑板是一个全局数据库，包含解域的全部状态，是知识源互相作用的唯一媒介。知识源响应是通过黑板状态的变化来控制。黑板通常应用在对于解决问题没有确定性算法的系统中，例如信号处理、问题规划、编译器优化等软件系统的设计中。

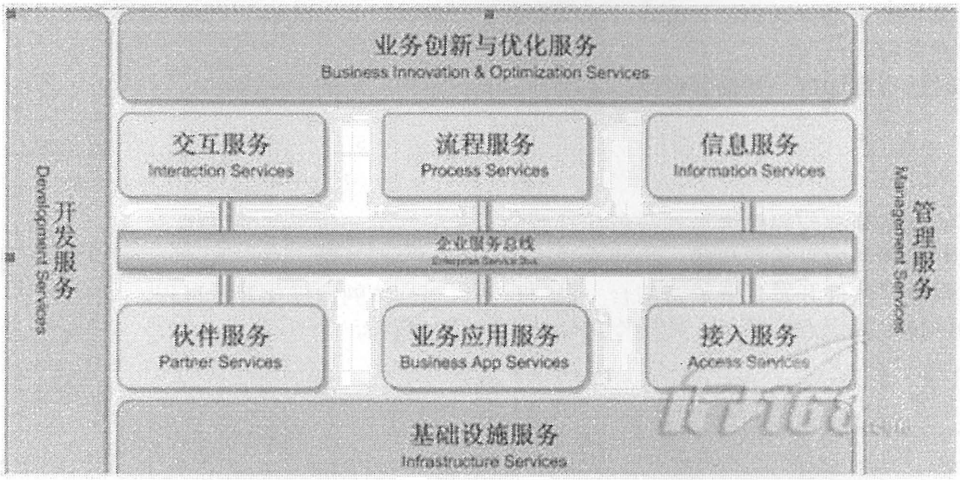
三、结合项目的实际状况，指出在架构设计时选择使用软件架构风格的情况，包括选择的依据、如何做的，要给出实际的效果及分析。

论文二：

写作要点：

一、按题目要求介绍作者参与的项目基本信息。

二、SOA 技术参考架构中包含的服务类别包括：



1、开发服务(Development Services)用于实现新开发的组件以及重用基础架构的能力。

2、业务创新优化服务(Business Innovation & Optimization Services)用于从 IT 和业务两个层面来监控和管理运行情况。

3、管理服务(Management Services)包括对服务、应用和资源的管理和保护能力，如通过负载均衡来有效的分配系统计算资源。

SOA 解决方案中的很多服务都是由已有应用系统提供的，接入服务(Access Services)提供访问已有应用或遗留系统的能力，同时提供已有应用、打包应用程序与 ESB 之间的桥接能力，将已有系统中的功能和信息转化为服务。

4、业务应用服务(Business App Services)指那些通过新的计算平台 JavaEE 来实现的新应用，它们所实现的功能和信息也都转化为服务提供出来。

在业务流程需要与外部的合作伙伴、供应商交互的情况下，伙伴服务(Partner Services)提供文档、协议以及伙伴管理的能力，比如说，可以提供企业边界处不同安全级别差异的转换。

5、信息服务(Information Services)是那些跟信息(而不是活动)有关系的服务，比如将多个系统中异构的数据，聚合、转换为业务需要的统一整齐的业务数据对象来访问。信息服务通过联合、复制和转换来解决基于不同实现方式的不同数据源之间的数据共享难题。

6、流程服务(Process Services)是指把多个服务聚合成为一个服务流程对应业务过程的服务，这种复合服务通常是长时间运行的过程。流程服务提供服务控制能力，将多个服务串起来实现一个业务流程。

7、交互服务(Interaction Service)一方面将人的活动，通过人机交互以服务的方式出现在整个业务过程中，作为流程服务)中的一部分;另一方面将 IT 的功能和数据传递给最终用户，并满足用户特定的使用习惯。

三、第 3 个问题是题目要重点描述的内容，要根据项目的实际情况来写自己是怎么做的，遇到什么样的问题，如何解决的。同时文章收尾要对效果进行评价。

论文三：
写作要点：
1．通过自己所从事的具有容错能力的软件课题，应说明以下几点：
- 项目的概述包括软件项目的功能与性能要求、软件整体结构、硬件环境描述；
- 在项目中对容错的具体要求；
- 所承担的工作内容。

2．作为容错技术，主要考虑四种冗余：硬件冗余、软件冗余、信息冗余和时间冗余。但作为一个具体软件系统，往往会采用多种技术去实现，这里应给出具体实现方法。主要说明以下几点容错方法（选中一种实现即可）：
- 软件冗余采用 N 版本技术（包括多种语言、多种编译器、背对背开发等）；
- 硬件冗余的基础上实现软件的冷备份、温备份、热备份；
- 信息冗余采用数据纠正码技术，给出几种纠正码（如 ECC、CRC 等）并说明实现原理；
- 时间冗余通常采用多次运行同一个程序（或 N 个版本的程序），对结果进行冗余比较。

3．主要让考生对自己采用的容错技术的结果进行系统的可靠性分析。主要说明以下几点：
- 通过容错技术实施，系统整个项目达到了几余度（如两余度、四余度、六余度）；
- 给出了本软件的可靠性指标；
- 分析出软件容错的实时性指标如何，如恢复时间、重组合（或称重构）时间。

论文四：
写作要点：
（1）首先用 400-500 字的篇幅简要叙述作者参与分析和开发的系统的概要和所担任的工作。
（2）简要说明构件技术开发的基本思想和步骤，为什么要选择构件技术来开发。
（3）如何运用构件技术来进行软件开发，要紧密结合主题项目，说明具体的构件化开发过程。
（4）分析并讨论各种构件技术的优点、缺点，并展望构件技术的发展趋势。