

《系统架构设计师考试》易混淆知识点

【精简版】

第 1 章 企业信息化战略与实施

易混淆点 1：政府对公民与公民对政府

1、政府对公民（G2C，Government To Citizen）：政府对公民提供的服务。

社区公安和水、火、天灾等与公共安全有关的信息。户口、各种证件和牌照的管理。

2、公民对政府（C2G，Citizen To Government）：个人应向政府缴纳税费和罚款及公民反馈渠道。个人应向政府缴纳的税费款。了解民意，征求群众意见。报警服务（盗贼、医疗、急救、火警）。

3、政府对政府（G2G，Government To Government）：政府之间的互动及政府与公务员之间互动。基础信息的采集、处理和利用，如人口信息；各级政府决策支持。

G2G 原则上包含：政府对公务员（G2E，Government To Employee）：内部管理信息系统

易混淆点 2：数据仓库与数据湖

区别：数据仓库仅支持分析处理，数据湖既支持分析处理，也支持事务处理。

维度	数据仓库	数据湖
数据	清洗过的数据 结构化的数据	原始数据 结构化，半结构化数据
模式	数据存储之前定义数据模式 数据集成之前完成大量工作 数据的价值提前明确	数据存储之后定义数据模式 提供敏捷，简单的数据集成 数据的价值尚未明确
存取方法	标准 SQL 接口	应用程序，类 SQL 的程序
优势	多数据源集成 干净，安全的数据 转换一次，多次使用	无限扩展性 并行执行 支持编程框架 数据经济

易混淆点 3：业务流程管理 BPM 和业务流程重组 BPR

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。

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。

BPM 与 BPR 区别：

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。

易混淆点 4: 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。强调增强了互动。

企业知识门户 (EKP): 企业网站的基础上增加知识性内容。

更多备考资料, 尽在希赛网微信小程序, 扫码即可查看下载



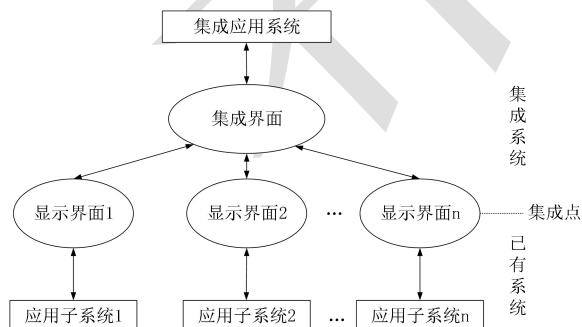
企业应用门户 (EAP): 实际上是对企业业务流程的集成。它以业务流程和企业应用为核心, 把业务流程中功能不同的应用模块通过门户技术集成在一起。

企业通用门户: 集以上四者于一身。

易混淆点 5: 企业应用集成

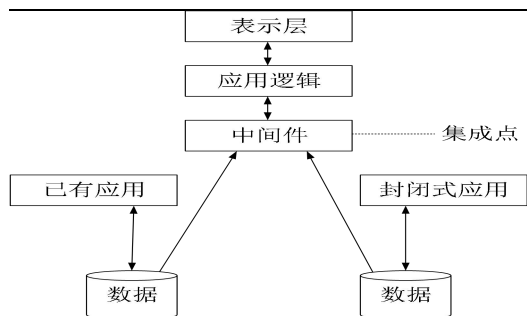
(1) 表示集成 (界面集成)

把各应用系统的界面集成起来, 统一入口, 产生“整体”感觉。



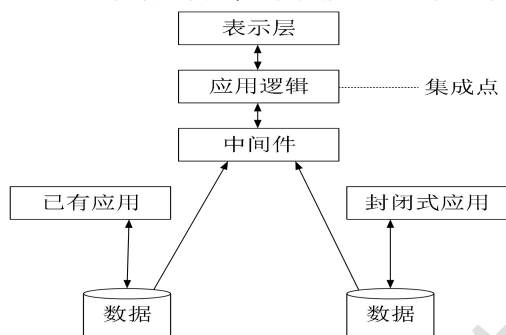
(2) 数据集成

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。



(3) 控制集成（功能集成、应用集成、API 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。



(4) 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。

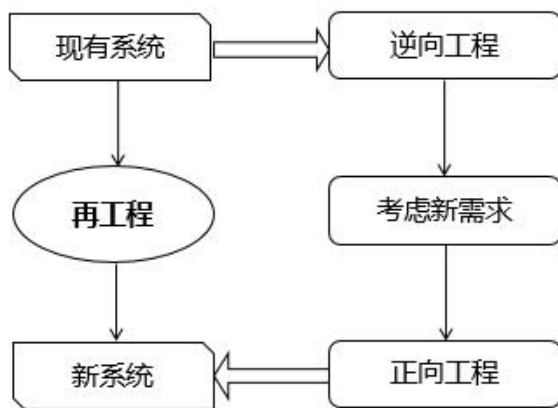
更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



第 2 章 软件工程

易混淆点 1：逆向工程，正向工程，再工程和设计恢复

(1) 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

(2) **再工程。**再工程是指在逆向工程所获得信息的基础上，修改或重构已有的系统，产生系统的一个新版本。再工程是对现有系统的重新开发过程，包括逆向工程、新需求的考虑过程和正向工程三个步骤。它不仅能从已存在的程序中重新获得设计信息，而且还能使用这些信息来重构现有系统，以改进它的综合质量。在利用再工程重构现有系统的同时，一般会增加新的需求，包括增加新的功能和改善系统的性能。

(3) **正向工程。**正向工程是指不仅从现有系统中恢复设计信息，而且使用该信息去改变或重构现有系统，以改善其整体质量。

(4) **设计恢复。**设计恢复是指借助工具从已有程序中抽象出有关数据设计、总体结构设计和过程设计等方面的信息。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



易混淆点 2：需求的分类

【需求的层次分类】

业务需求：是指反应企业或客户对系统高层次的目标要求，通常来自项目投资入、购买产品的客户、客户单位的管理人员、市场营销部门或产品策划部门等。通过业务需求可以确定项目视图和范围，为以后的开发工作奠定了基础。

用户需求：描述的是用户的具体目标，或用户要求系统必须能完成的任务。也就是说，用户需求描述了用户能使用系统来做些什么。

系统需求：是从系统的角度来说明软件的需求，包括功能需求、非功能需求和设计约束等。

功能需求也称为行为需求，它规定了开发人员必须在系统中实现的软件功能，用户利用这些功能来完成任务，满足业务要求。

性能需求（非功能需求）是指系统必须具备的属性或品质，又可细分为软件质量属性和其他非功能需求。

设计约束也称为限制条件或补充规约，通常是对系统的一些约束说明。

【需求的 QFD 分类】

质量功能部署 QFD 是一种将用户要求转化成软件需求的技术，其目的是最大限度地提升软件工程过程中用户的满意度。QFD 将软件需求分为三类：

常规需求（基本需求）：用户认为系统应该做到的功能或性能，实现越多用户会越满意。

期望需求：用户想当然认为系统应具备的功能或性能，但并不能正确描述自己想要得到的这些功能或性能需求。如果期望需求没有得到实现，会让用户感到不满意。

兴奋需求（意外需求）：是用户要求范围外的功能或性能，实现这些需求用户会更高兴，但不实现也不影响其购买的决策。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



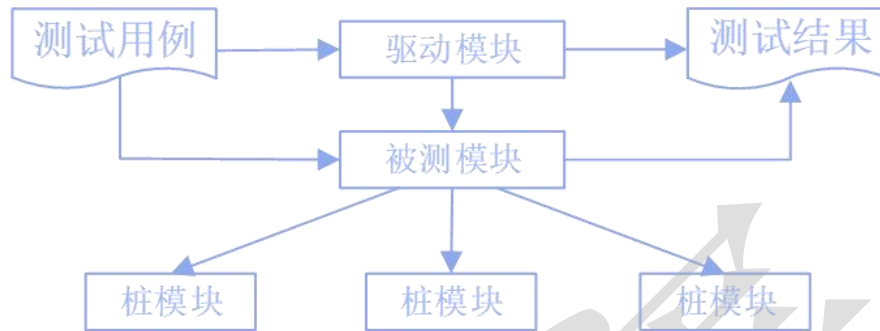
易混淆点 3：架构模式和设计模式和惯用法

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

易混淆点 4： 模块组装方式



一次性组装方式：其结果：发现有错误，却茫然找不到原因；查错和改错都会遇到困难。

自顶向下的增殖方式：优点：在测试过程中较早地验证了主要的控制和判断点；功能可行性较早地得到证实，还能增强开发者和用户成功的信心。缺点：导致过多的回归测试；增加建桩模块的复杂度，导致增加附加测试

自底向上的增殖方式：优点：可以把容易出问题的部分在早期解决；缺点：对主要的控制直到最后才接触到；可以实施多个模块的并行测试，提高测试效率。

混合增殖方式，完成的标志：成功地执行了测试计划中规定的所有集成测试；修正了所发现的错误（与用户就剩余错误的修改计划达成一致）；测试结果通过了专门小组的评审

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



易混淆点 5： 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。

易混淆点 6：系统维护分类

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不断变化的管理需求也使得各级管理人员不断提出新的信息需求。

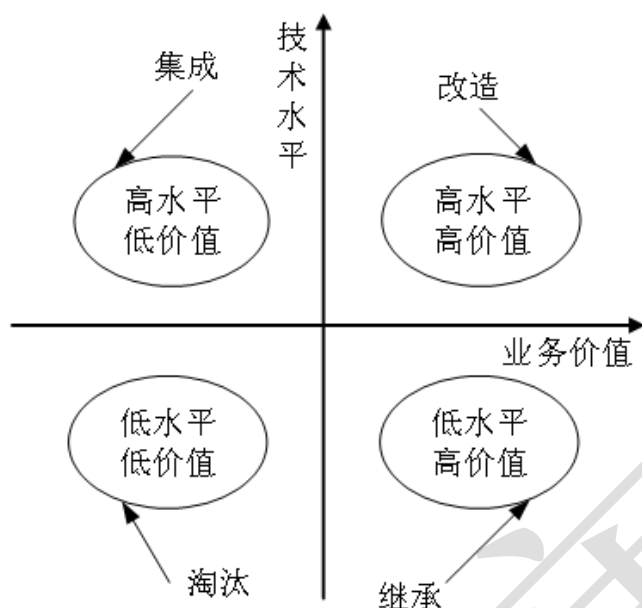
完善性维护：扩充功能和改善性能而进行的修改。对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。

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。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



易混淆点 7：遗留系统演化策略（注意区分继承策略和集成策略）



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。

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。

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。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



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，形成了一个信息孤岛，对这种遗留系统的演化策略为集成。

第 3 章 软件架构设计

易混淆点 1：微服务与 SOA 的对比

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	企业服务总线（ESB）充当了服务之间通信的角色

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

易混淆点 2：敏感点和权衡点

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：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：影响多个质量属性的特性，是多个质量属性的敏感点。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



易混淆点 3：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

易混淆点 4：MVP 与 MVC

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。

MVP 模式与 MVC 模式的主要区别：

在组件耦合度方面：在 MVP 模式中，视图并不直接使用模型，它们之间的通信通过 Presenter 进行，从而实现了视图与模型的分离，而在 MVC 模式中，视图直接与模型交互。

在组件分工方面：在 MVP 模式中，视图需要处理鼠标及键盘等触发的界面事件，而在 MVC 模式中这通常是由控制器完成的工作；在 MVP 模式中，系统核心业务逻辑组织集中在 Presenter 中，而在 MVC 模式中，相应的控制器通常只完成事件的分发。

在开发工程化支持方面：MVP 模式可更好地支持单元测试，而在 MVC 模式中，由于模型与

视图绑定，因此难以实施相应的单元测试；在 MVP 模式中，Presenter 基于约定接口与视图和模型交互，可更好地支持组件的重用。

易混淆点 5：构件、对象、模块的对比

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

易混淆点 6：无状态服务和有状态服务

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。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



易混淆点 7：Redis 和 Memcache

Redis 和 Memcache 对比：

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。

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。

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

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。

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。

第 4 章 信息安全分析与设计

易混淆点 1：网络攻击分类

被动攻击：收集信息为主，破坏保密性。

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。

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
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 【旁路攻击】	密码学中是指绕过对加密算法的繁琐分析，利用密码算法的硬件实现的运算中泄露的信息。如执行时间、功耗、电磁辐射等，结合统计理论快速的破解密码系统。
	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。

易混淆点 2：安全保护等级

用户自主保护级：适用于普通内联网用户

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

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

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

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

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



第 5 章 系统可靠性分析与设计

易混淆点 1：恢复块方法与 N 版本程序设计

恢复块方法与 N 版本程序设计对比：

	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

易混淆点 2：可靠性和可用性

可靠性：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。

可用性：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。

易混淆点 3：可靠性指标



平均无故障时间 \rightarrow (MTTF) $MTTF=1/\lambda$, λ 为失效率

平均故障修复时间 \rightarrow (MTTR) $MTTR=1/\mu$, μ 为修复率

平均故障间隔时间 \rightarrow (MTBF) $MTBF = MTTR + MTTF$

系统可用性 $\rightarrow MTTF/(MTTR+MTTF) \times 100\%$

在实际应用中，一般 MTTR 很小，所以通常认为 $MTBF \approx MTTF$ 。

第 6 章 项目管理

易混淆点 1: PERT 图和 Gantt 图

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。

易混淆点 2: 质量保证与质量控制

(1) **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。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



(2) **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。

(3) 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。

第 7 章 计算机组成与体系结构

易混淆点 1: CISC 与 RISC (CISC: 复杂指令集; RISC: 精简指令集。)

CISC 与 RISC 对比:

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

易混淆点 2: Cache 映射方式

直接相联映像: 硬件电路较简单, 但冲突率很高。

全相联映像: 电路难于设计和实现, 只适用于小容量的 cache, 冲突率较低。

组相联映像: 直接相联与全相联的折中。

易混淆点 3: 流水线公式

流水线执行时间 (理论公式): $(t_1+t_2+...+t_k)+(n-1)*\Delta t$

流水线执行时间 (实践公式): $k*\Delta t+(n-1)*\Delta t$

流水线吞吐率:

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

流水线最大吞吐率 (即流水线周期的倒数)

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

流水线加速比: 顺序执行时间/流水线执行时间

第 8 章 嵌入式系统

易混淆点 1: 嵌入式微处理器分类

(1) 嵌入式微控制器 (MCU: Micro Controller Unit): 又称为单片机, 片上外设资源一般比较丰富, 适合于控制。

(2) 嵌入式微处理器 (EMPU: Embedded Micro Processing Unit): 又称为单板机, 由通用计算机中的 CPU 发展而来, 仅保留和嵌入式应用紧密相关的功能硬件。

(3) 嵌入式 DSP 处理器 (DSP: Digital Signal Processor): 专门用于信号处理方面的处理器。

(4) 嵌入式片上系统 (SOC): 追求产品系统最大包容的集成器件。

成功实现了软硬件的无缝结合, 直接在微处理器片内嵌入操作系统的代码模块。

减小了系统的体积和功耗、提高了可靠性和设计生产效率。

易混淆点 2: 哈佛结构和冯·诺依曼结构

(1) 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。(冯·诺依曼体系结构冯·诺依曼理论的要点是: 计算机的数制采用二进制; 计算机应该按照程序顺序执行。人们把冯·诺依曼的这个理论称为冯·诺依曼体系结构。) 特点:

一般用于 PC 处理器, 如 I3, I5, I7 处理器

指令与数据存储器合并在一起

指令与数据都通过相同的数据总线传输

更多备考资料, 尽在希赛网微信小程序, 扫码即可查看下载



(2) 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:

一般用于嵌入式系统处理器 (DSP)

指令与数据分开存储, 程序指令存储和数据存储分开, 可以使指令和数据有不同的数据宽度, 可以并行读取, 有较高数据的吞吐率

有 4 条总线: 指令和数据的数据总线与地址总线

易混淆点 3：数据总线、地址总线和控制总线

数据总线（Data Bus，DB）：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，AB）：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，CB）：将微处理器控制单元（Control Unit）的信号，传送到周边设备。

易混淆点 4：常见的嵌入式 RTOS（实时操作系统）VxWorks 和 RT-Linux

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

第 9 章 系统配置与性能评价

易混淆点 1：指令周期、总线周期和时钟周期

时钟频率（主频）：主频又称为时钟频率，**时钟周期**是时钟频率的倒数。如主频为 1GHz，则说明 1 秒钟有 1G 个时钟周期，每个时钟周期为 $1000 \times 1000 \times 1000 / 1G = 1ns$ 。（主频 = 外频 * 倍频）

指令周期、总线周期和时钟周期之间的关系：一个指令周期由若干个总线周期组成，而一个总线周期时间又包含有若干个时钟周期（也可说一个指令周期包含若干个时钟周期）。

指令周期（Instruction Cycle）：取出并执行一条指令的时间。

总线周期（BUS Cycle）：也就是一个访存储器或 I/O 端口操作所用的时间。

第 10 章 操作系统

易混淆点 1：绝对路径与相对路径

绝对路径：是从盘符开始的路径。绝对路径从根目录开始写起，并且该文件的全名即为绝对路径+文件名。

相对路径：是从当前路径开始的路径。相对路径从当前位置下一级目录开始写起。

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

易混淆点 2：I/O 传输控制方式：程序查询方式，中断方式，DMA

程序查询方式（CPU 一直处于询问、等待的过程，占用 CPU 时间最长，CPU 利用率最低）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。

中断方式（I/O 完成后向 CPU 发送中断请求信号，CPU 和 I/O 可以并行）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。

DMA（CPU 只做初始化，不参与具体数据传输过程）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。

易混淆点 3：页式存储和段式存储

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。

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。

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。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



易混淆点 4：互斥信号量和同步信号量的区分

信号量一般与 PV 操作结合在一起，对于同步模型描述中，涉及的就是同步信号量，一般同步是多个进程之间通过 PV 操作进行制约，此时 P 操作和成对的 V 操作一般是位于不同的进程中的，而互斥模型中 PV 操作控制的是对资源的访问限制，此时 P 操作和成对的 V 操作一般出现在资源使用前和适用后，一般位于同一个进程中。

第 11 章 计算机网络

易混淆点 1：递归查询和迭代查询

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。

查询过程：

主机向本地域名服务器的查询一般采用的都是递归查询：

如果主机所询问的本地域名服务器不知道被查询域名的 IP 地址，那么本地域名服务器就以 DNS 客户的身份，向其他根域名服务器继续发出查询请求报文。

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：

本地域名服务器向根域名服务器的查询通常是采用迭代查询。当根域名服务器收到本地域名服务器的迭代查询请求报文时，要么给出所要查询的 IP 地址，要么告诉本地域名服务器：

“你下一步应当向哪一个域名服务器进行查询”。然后让本地域名服务器进行后续的查询。

根服务器或者流量较大的域名服务器都不使用递归查询，其原因也很简单，大量的递归查询会导致服务器过载。

易混淆点 2：工作区子系统，水平子系统，管理子系统，垂直干线子系统

(1) **工作区子系统**由信息插座、插座盒、连接跳线和适配器组成。

(2) **水平子系统**由一个工作区的信息插座开始，经水平布置到管理区的内侧配线架的线缆所组成。

(3) **管理子系统**由交连、互连配线架组成。管理子系统为连接其它子系统提供连接手段。

(4) **垂直干线子系统**由建筑物内所有的垂直干线多对数电缆及相关支撑硬件组成，以提供设备间总配线架与干线接线间楼层配线架之间的干线路由

(5) **设备间子系统**是由设备间中的电缆、连接器和有关的支撑硬件组成，作用是将计算机、PBX、摄像头、监视器等弱电设备互连起来并连接到主配线架上。

(6) **建筑群子系统**将一个建筑物的电缆延伸到建筑群的另外一些建筑物中的通信设备和装置上，是结构化布线系统的一部分，支持提供楼群之间通信所需的硬件。它由电缆、光缆和入楼处的过流过压电气保护设备等相关硬件组成，常用介质是光缆。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



易混淆点 3：逻辑网络设计和物理网络设计的产出

(1) 逻辑网络设计

(拓扑结构、网络地址分配、安全规划、网络管理、选择路由协议、设备命名规则。)

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。

(2) 物理网络设计

(设备的具体物理分布、运行环境等确定，设备选型、结构化布线、机房设计)

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。输出内容如下：

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

第 12 章 数据库系统

易混淆点 1：外模式，模式和内模式

三级模式：外模式对应视图，模式（也称为概念模式）对应数据库表，内模式对应物理文件。

两层映像：外模式-模式映像，模式-内模式映像；两层映像可以保证数据库中的数据具有较高的逻辑独立性和物理独立性。

物理独立性：即数据库的内模式发生改变时，应用程序不需要改变。

逻辑独立性：即逻辑结构发生改变时，用户程序不需要改变。(逻辑独立性比物理独立性更难实现)

易混淆点 2：候选码，主码，外码，全码

候选键/候选码/码：如果在一个关系中，存在一个属性（或属性组）都能用来唯一标识该关

系的元组，并不含有多余的属性，这个属性（或属性组）称为该关系的候选码或候选键。候选键可以是单属性也可以是多属性集合，候选键可以是一个也可以有多个。

主键/主码：可以有多个不同的候选键，在其中任选一个作为主键。其值能唯一地标识元组的一个或多个属性，称为主码或关键字。组成候选码的属性就是主属性，其他为非主属性。

外键：关系中的某个属性（或属性组）不是该关系的主码或只是主码的一部分，但却是另一个关系的主码时，该属性（或属性组）称为这个关系的外码。

全码：关系中的所有属性组成这个关系模式的候选码。

易混淆点 3：丢失更新/丢失修改，不可重复读，读“脏”数据

◆ 丢失更新		◆ 不可重复读		◆ 读“脏”数据	
T1	T2	T1	T2	T1	T2
①读A=10 ② ③A=A-5写回 ④	读A=10 A=A-8 写回	①读A=20 读B=30 求和=50 ② ③读A=70 读B=30 求和=100 (验算不对)	读A=20 A←A+50 写A=70	①读A=20 A←A+50 写回70 ② ③ROLLBACK A恢复为20	读A=70

易混淆点 4：冷备份和热备份

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。

备份方式 \ 优缺点	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	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”

	是快速的	
--	------	--

易混淆点 5：分区与分表

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。**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。**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。

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载



第 13 章 知识产权与标准化

易混淆点 1：各法律法规的保护对象和范围以及保护期限

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。
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

易混淆点 2：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

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	

易混淆点 3：常见侵权与合理引用区分

不侵权	侵权
-----	----

<ul style="list-style-type: none">✓ 个人学习、研究或者欣赏：<ul style="list-style-type: none">✓ 适当引用；✓ 公开演讲内容✓ 用于教学或科学研究<ul style="list-style-type: none">✓ 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。	<ul style="list-style-type: none">✓ 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：<ul style="list-style-type: none">✓ 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。
---	---

更多备考资料，尽在希赛网微信小程序，扫码即可查看下载

