2009 年系统架构师考试科目二：案例分析

1. 阅读以下软件架构设计的问题，在答题纸上回答问题 1 和问题 2。

【题目】

某软件开发公司欲为某电子商务企业开发一个在线交易平台，支持客户完成网上购物活动中的在线交易。在系统开发之初，企业对该平台提出了如下要求：

(1)在线交易平台必须在 1s 内完成客户的交易请求。

(2)该平台必须保证客户个人信息和交易信息的安全。

(3)当发生故障时，该平台的平均故障恢复时间必须小于 10s。

(4)由于企业业务发展较快，需要经常为该平台添加新功能或进行硬件升级。添加新功能或进行硬件升级必须在 6 小时内完成。

针对这些要求，该软件开发公司决定采用基于架构的软件开发方法，以架构为核心进行在线交易平台的设计与实现。

【问题 1】(9 分)

软件质量属性是影响软件架构设计的重要因素。请用 200 字以内的文字列举六种不同的软件质量属性名称，并解释其含义。

【问题 1 解析】

常见的软件质量属性有多种，例如性能(Performance)、可用性(Availability)、可靠性(Reliability)、健壮性(Robustness)、安全性(Security)、可修改性(Modification)、可变性(Changeability)、易用性(Usability)、可测试性(Testability)、功能性(Functionality)和互操作性(Inter-operation)等。

这些质量属性的具体含义是:

(1)性能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事件的个数。

(2)可用性是系统能够正常运行的时间比例。

(3)可靠性是指软件系统在应用或错误面前，在意外或错误使用的情况下维持软件系统功能特性的基本能力。

(4)健壮性是指在处理或环境中，系统能够承受压力或变更的能力。

(5)安全性是指系统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。

(6)可修改性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。

(7)可变性是指体系结构经扩充或变更成为新体系结构的能力。

(8)易用性是衡量用户使用一个软件产品完成指定任务的难易程度。

(9)可测试性是指软件发现故障并隔离、定位其故障的能力特性，以及在一定的时间和成本前提下，进行测试设计、测试执行的能力。

(10)功能性是系统所能完成所期望工作的能力。

(11)互操作性是指系统与外界或系统与系统之间的相互作用能力。

【问题 2】(16 分)

请对该在线交易平台的 4 个要求进行分析，用 300 字以内的文字指出每个要求对应何种软件质量属性；并针对每种软件质量属性，各给出 2 种实现该质量属性的架构设计策略。

【问题 2 解析】

(1) 该要求主要对应性能，可以采用的架构设计策略有增加计算资源、改善资源需求、资源管理和资源调度。

(2) 该要求主要对应安全性，可以采用的架构设计策略有抵御攻击、攻击检测、从攻击

中恢复和信息审计等。

(3) 该要求主要对应可用性，可以采用的架构设计策略有心跳、Ping/Echo、主动冗余、被动冗余、选举等。

(4) 该要求主要对应可修改性，接口-实现分离、抽象、信息隐藏等架构策略实现该属性。

2. 阅读以下关于软件系统数据架构建模的说明，在答题纸上回答问题 1 至问题 3。

【题目】

某公司拟开发一个商业情报处理系统，使公司能够及时针对市场环境的变化及时调整发展战略，以获取最大的商业利益。项目组经过讨论，决定采用结构化分析和设计方法。在系统分析阶段，为了更好地对情报数据处理流程及其与外部角色的关联进行建模，项目组成员分别给出了自己的设计思路：

(1)小张提出先构建系统流程图(System Flowcharts)，以便更精确地反映系统的业务处理过程及数据的输入和输出；

(2)小李提出先构建系统数据流图(Data Flow Diagrams)，来展现系统的处理过程和定义业务功能边界，并给出了情报分类子系统的 0 层和 1 层数据流图，后者如图 2-1 所示。

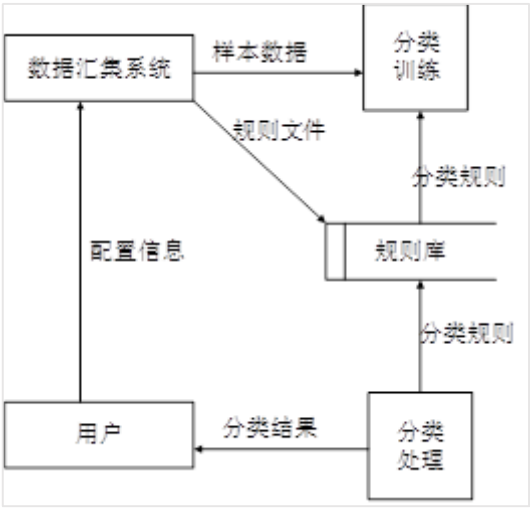


图 2-1 情报分类子系统的 1 层数据流图

项目组经过讨论确定以数据流图作为本阶段的建模手段。工程师老王详细说明了流程图和数据流图之间的区别与联系，并指出了图 2-1 的数据流图中存在的错误。

【问题 1】(11 分)

流程图和数据流图是软件系统分析设计中常用的两种手段，请用 300 字以内文字简要说明流程图与数据流图的含义及其区别，并说明项目组为何确定采用数据流图作为建模手段。

【问题 1 解析】

数据流图作为一种图形化工具，用来说明业务处理过程、系统边界内所包含的功能和系统中的**数据流**。

流程图以图形化的方式展示应用程序从数据输入开始到获得输出为止的逻辑过程，描述处理过程的**控制流**。

两者的区别主要包括：

(1)数据流图中的处理过程可并行；流程图在某个时间点只能处于一个处理过程。

(2)数据流图展现系统的数据流；流程图展现系统的控制流。

(3)数据流图展现全局的处理过程，过程之间遵循不同的计时标准；流程图中处理过程

遵循一致的计时标准。

(4)数据流图适用于系统分析中的逻辑建模阶段；流程图适用于系统设计中的物理模阶段。

【问题 2】(8 分)

请分析指出图 2-1 所示的数据流图中存在的错误及其原因，并针对图 2-1 的 1 层数据流图绘制出情报分类子系统的 0 层数据流图。

【问题 2 解析】

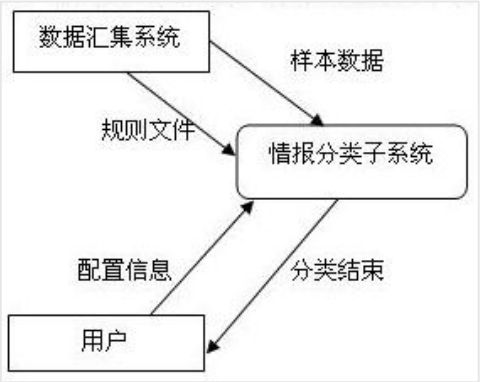
如图所示的数据流图中存在的错误有以下 4 种：

(1)“分类训练”加工：只有输入没有输出，产生数据黑洞；

(2)“分类处理”加工：有输出没有输入，无中生有；

(3)“规则文件”数据流：外部实体没有经过加工处理，直接到数据存储；

(4)“配置信息”数据流：外部实体之间没有加工处理，存在直接数据流。



【问题 3】(6 分)

高质量的数据流图是可读的、内部一致的并能够准确表示系统需求。请用 300 字以内文字说明在设计高质量的数据流图时应考虑的三个原则。

【问题 3 解析】

高质量数据流图设计时应考虑的三个原则：

(1)**复杂性最小化原则**。DFD 分层结构就是把信息划分为小的且相对独立的一大批子集例子，这样就可以单独考查每一个 DFD。如果要了解某个过程更加详的信息，可以跳转到该过程的下一层；如果要知道一个 DFD 如何与其他 DFD 相关联，可以跳转到上一层的 DFD 进行考查。

(2)**接口最小化原则**。接口最小化是复杂性最小化的一种具体规则。在设计模式时，应使得模型中各个元素之间的接口数或连接数最小化。

(3)**数据流一致性原则**。一个过程和它的过程分解在数据流内容中是否有差别？是否存在有数据流出但没有相应的数据流入的加工？是否存在有数据流入但没有相应的数据流出的加工？

3. 阅读以下关于嵌入式软件体系架构的叙述，在答题纸上回答问题 1 至问题 3。

【题目】

某公司承担了一项宇航嵌入式设备的研制任务。本项目除对硬件设备环境有很高的要求外，还要求支持以下功能：

(1)设备由多个处理机模块组成，需要时外场可快速更换(即 LRM 结构)；

(2)应用软件应与硬件无关，便于软硬件的升级；

(3)由于宇航嵌入式设备中要支持不同功能，系统应支持完成不同功能任务间的数据隔离；

(4)宇航设备可靠性要求高，系统要有故障处理能力。

公司在接到此项任务后，进行了反复论证，提出三层栈(TLS)软件总体架构，如图 3-1 所示，并将软件设计工作交给了李工，要求其在三周内完成软件总体设计工作，给出总体设计方案。

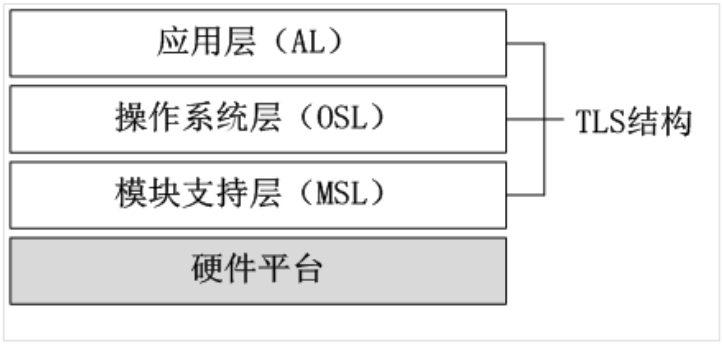


图 3-1 宇航嵌入式设备软件架构

【问题 1】(共 8 分)

用 150 字以内的文字，说明公司制定的 TLS 软件架构的层次特点，并针对上述功能需求(1)~(4)，说明架构中各层内涵。

【问题 1 解析】

TLS 结构框架的主要特点:

(1)应用软件仅与操作系统服务相关，不直接操作硬件。

(2)操作系统通过模块支持原软件访问硬件，可与具体硬件无关。

(3)模块支持层将硬件抽象成标准操作。

(4)通过三层栈的划分可实现硬件的快速更改与升级，应用软件的升级不会引起硬件的变更。

TLS 结构框架的各层内涵是:

(1)应用层主要完成宇航设备的具体工作，由多个功能任务组成，各功能任务间的隔离由操作系统层实现。

(2)操作系统层实现应用软件与硬件的隔离，为应用软件提供更加丰富的计算机资源服务。操作系统为应用软件提供标准的 API 接口(如 POSIX)，确保了应用软件的可升级性。

(3)模块支持层为操作系统管理硬件资源提供统一管理方法，用一种抽象的标准接口实现软件与硬件的无关性，达到硬件的升级要求，便于硬件的外场快速更换。

【问题 2】(共 10 分)

在 TLS 软件架构的基础上，关于选择哪种类型的嵌入式操作系统问题，李工与总工程师发生了严重分歧。李工认为，宇航系统是实时系统，操作系统的处理时间越快越好，隔离意味着以时间作代价，没有必要，建议选择类似于 VxWorks5.5 的操作系统；总工程师认为，应用软件间隔离是宇航系统安全性要求，宇航系统在选择操作系统时必须考虑这一点，建议选择类似于 Linux 的操作系统。

请说明两种操作系统的主要差异，完成表 3-1 中的空白部分，并针对本任务要求，用 200 字以内的文字说明你选择操作系统的类型和理由。

表 3-1 两种操作系统的主要差异

比较类型	VxWorks5.5	Linux
工作方式	操作系统与应用程序处于同一存储空间	①
多任务支持	支持多任务（线程）操作	②
实时性	③	实时系统
安全性	④	⑤
标准API	支持	支持

【问题 2 解析】

两种操作系统的差异见下表。

比较类型	VxWorks5.5	Linux
工作方式		①操作系统与应用程序处于不同存储空间
多任务支持		②支持多进程、多线程操作
实时性	③硬实时系统	
安全性	④任务间无隔离保护	⑤支持进程间隔离保护
标准API		

选择类似于 Linux 的嵌入式操作系统。理由如下：

(1)Linux 操作系统是一种安全性较强的操作系统。内核工作在系统态，应用软件工作在用户态，可以有效防止应用软件对操作系统的破坏。

(2)Linux 操作系统调度的最小单位是线程，线程归属于进程，进程具有自己独立的资源。进程通过存储器管理部件(MMU)实现多功能应用间隔离。

(3)嵌入式 Linux 操作系统支持硬件抽象，可有效实现 TLS 结构，并将硬件抽象与操作系统分离，可方便实现硬件的外场快速更换。

【问题 3】(共 7 分)

故障处理是宇航系统软件设计中极为重要的组成部分。故障处理主要包括故障监视、故障定位、故障隔离和系统容错(重组)。用 150 字以内的文字说明嵌入式系统中故障主要分哪几类？并分别给出两种常用的故障滤波算法和容错算法。

【问题 3 解析】

(1)嵌入式系统中故障主要分为：

① 硬件故障：如 CPU、存储器和定时器等；

② 应用软件故障：如数值越界、异常和超时等；

③ 操作系统故障：如越权访问、死锁和资源枯竭等。

(2)滤波算法：

① 门限算法

② 递减算法

③ 递增算法

④ 周期滤波算法

(3)容错算法：

① N+1 备份

② 冷备

③ 温备

④ 热备

4. 阅读以下软件系统架构选择的问题，在答题纸上回答问题 1 至问题 3。

【题目】

某公司欲开发一个车辆定速巡航控制系统，以确保车辆在不断变化的地形中以固定的速度行驶。图 4-1 给出了该系统的简化示意图。表 4-1 描述了各种系统输入的含义。

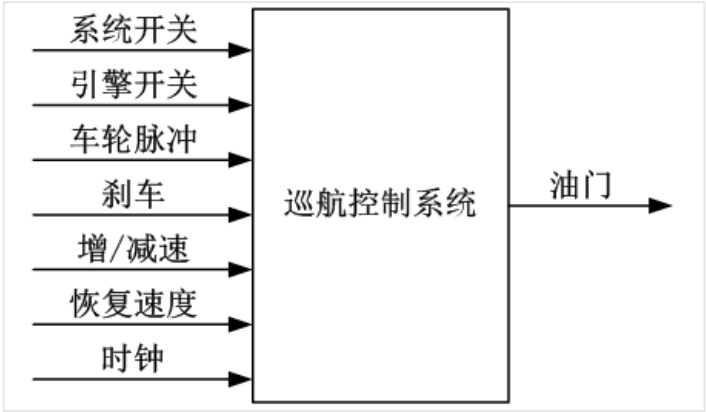


图 4-1 定速巡航控制系统的简化示意图

表 4-1 定速巡航控制系统输入说明

输入名称	作用
系统开关	开启/关闭巡航控制系统
引擎开关	开启/关闭洗车引擎（引擎开启时，巡航控制系统处于就绪状态）
车轮脉冲	车轮每转一次，相应地发出一次脉冲
刹车	当刹车被踩下时，定速巡航控制系统会临时恢复到人工控制
增/减速	增加或减慢当前车速（仅在定速巡航控制系统处于开启的状态下可用）
恢复速度	恢复原来保持的车速（仅在定速巡航控制系统处于开启的状态下可用）
时钟	每毫秒定时脉冲

公司的领域专家对需求进行深入分析后，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。

在对车辆定速巡航控制系统的架构进行设计时，公司的架构师王工提出采用面向对象的架构风格，而李工则主张采用控制环路的架构风格。在架构评估会议上，专家对这两种方案进行综合评价，最终采用了面向对象和控制环路相结合的混合架构风格。

【问题 1】(共 5 分)

在实际的软件项目开发中，采用成熟的架构风格是项目成功的保证。请用 200 字以内的文字说明：什么是软件架构风格；面向对象和控制环路两种架构风格各自的特点。

【问题 1 解析】

软件架构风格是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。

面向对象架构风格的特征是将数据表示和基本操作封装在对象中。这种模式的构件是对象，对象维护自身表示的完整性，对象之间通过消息机制进行通信，对象交互时需要知道彼此的标识，通过对象之间的协作完成计算过程。

控制环路架构风格是将过程输出的指定属性维护在一个特定的参考值(设定点)。控制环路风格包括过程变量、被控变量、输入变量、操纵变量和设定点等构件，通过收集实际和理想的过程状态信息，并能调整过程变量使得实际状态趋于理想状态。

【问题 2】(12 分)

用户需求没有明确给出该系统如何根据输入集合计算输出。请用 300 字以内的文字针对该系统的增减速功能，分别给出两种架构风格中的主要构件，并详细描述计算过程。

【问题 2 解析】

对于系统的增减速功能，采用面向对象风格的巡航控制系统首先会定义司机、油门、时钟、速度计和车轮等构件。

整个计算的主要过程是：

(1)司机进行增/减速操作设置期望速度，该期望速度以消息的形式传递给速度计；

(2)速度计通过向车轮和时钟发送消息获取车轮转速和时钟值，得到当前速度；

(3)速度计计算当前速度和期望速度的速度差值；

(4)该差值以消息的形式发送给油门，油门通过速度差值调节自身状态；

(5)整个过程在时钟的控制下定期向速度计发送消息，重复执行(2)～(4)。

控制环路的架构风格以控制器为核心，期望速度、车轮脉冲、时钟和油门等作为构件。具体的计算过程是：

(1)司机进行增/减速操作设置期望速度值；

(2)将设定值置为期望速度值；

(3)控制器采集车轮脉冲和时钟值，计算出当前速度；

(4)比较期望速度和当前速度，计算速度差值，控制油门动作；

(5)反复执行(3)和(4)。

【问题 3】(8 分)

实际的软件系统架构通常是多种架构风格的混合，不同的架构风格都有其适合的应用场景。以该系统为例，针对面向对象架构风格和控制环路架构风格，各给出两个适合的应用场景，并简要说明理由。

【问题 3 解析】

适合面向对象架构风格的应用场景：

(1)用户刹车，立即退出巡航控制系统。理由：这是一个典型的事件驱动的场景，适合于面向对象风格。

(2)系统对突发事件的处理，如某些部件失灵等。理由：当发生突发事件时，系统会同时产生数据和事件，这种清况用对象建模较为恰当。

适合面向控制环路架构风格的应用场景：

(1)在达到期望速度后，系统维持恒定速度行驶。理由：这是一个典型的闭环控制的情景，系统需要在外界情况不断发生变化的情况下进行调整，使得系统状态尽可能接近期望状态。

(2)用户改变期望速度后，系统不断进行调整，直至到达恒定速度。理由：这是一个闭环控制情景，当用户设定期望速度值后，系统需要在不断获取当前速度和外界条件的情况下对系统状态持续调整，使得系统状态尽可能接近这个新的期望状态。

5. 阅读以下关于信息系统安全性的叙述，在答题纸上回答问题 1 至问题 3。

【题目】

某企业根据业务扩张的要求，需要将原有的业务系统扩展到互联网上，建立自己的 B2C 业务系统，此时系统的安全性成为一个非常重要的设计需求。为此，该企业向软件开发商提出如下要求：

(1)合法用户可以安全地使用该系统完成业务；

(2)灵活的用户权限管理；

(3)保护系统数据的安全，不会发生信息泄漏和数据损坏；

(4)防止来自于互联网上各种恶意攻击；

(5)业务系统涉及到各种订单和资金的管理，需要防止授权侵犯；

(6)业务系统直接面向最终用户，需要在系统中保留用户使用痕迹，以应对可能的商业诉讼。

该软件开发商接受任务后，成立方案设计小组，提出的设计方案是：在原有业务系统的基础上，保留了原业务系统中的认证和访问控制模块；为了防止来自互联网的威胁，增加了防火墙和入侵检测系统。

企业和软件开发商共同组成方案评审会，对该方案进行了评审，各位专家对该方案提出了多点不同意见。李工认为，原业务系统只针对企业内部员工，采用了用户名/密码方式是一可以的，但扩展为基于互联网的 B2C 业务系统后，认证方式过于简单，很可能造成用户身份被盗取：王工认为，防止授权侵犯和保留用户痕迹的要求在方案中没有体现。而刘工则认为，即使是在原有业务系统上的扩展与改造，也必须全面考虑信息系统面临的各种威胁，设计完整的系统安全架构，而不是修修补补。

【问题 1】(9 分)

信息系统面临的安全威胁多种多样，来自多个方面。请指出信息系统面临哪些方面的安全威胁并分别予以简要描述。

【问题 1 解析】

信息系统面临的安全威胁来自于物理环境、通信链路、网络系统、操作系统、应用系统以及管理等多个方面。

物理安全威胁是指对系统所用设备的威胁，如自然灾害、电源故障、数据库故障和设备被盗等造成数据丢失或信息泄漏。

通信链路安全威胁是指在传输线路上安装窃听装置或对通信链路进行干扰。

网络安全威胁当前主要是指由于因特网的开放性、国际性与无安全管理性，对内部网络形成的严重安全威胁。

操作系统安全威胁指的是操作系统本身的后门或安全缺陷，如“木马”和“陷阱门”等。

应用系统安全威胁是指对于网络服务或用户业务系统安全的威胁，包括应用系统自身漏洞，也受到“木马”的威胁。

管理系统安全威胁指的是人员管理和各种安全管理制度。

【问题 2】(8 分)

认证是安全系统中不可缺少的环节，请简要描述主要的认证方式，并说明该企业应采用哪种认证方式。

【问题 2 解析】

目前主要的认证方式有三类:

(1)用户名和口令认证：主要是通过一个客户端与服务器共知的口令(或与口令相关的数据)进行验证。根据处理形式的不同，分为验证数据的明文传送、利用单向散列函数处理验证数据、利用单向散列函数和随机数处理验证数据。

(2)使用令牌认证：该方式中，进行验证的密钥存储于令牌中，目前的令牌包括安全证书和智能卡等方式。

(3)生物识别认证：主要是根据认证者的图像、指纹、气味和声音等作为认证数据。根据该企业的业务特征，采用令牌认证较为合适。

【问题 3】(8 分)

请解释授权侵犯的具体含义;针对王工的意见给出相应的解决方案,说明该解决方案的名称、内容和目标。

【问题 3 解析】

授权侵犯指的是被授权以某一目的使用某一系统或资源的某个人,将此权限用于其他非授权的目的，也称作“内部攻击”。

针对王工的建议，从系统安全架构设计的角度需要提供抗抵赖框架。

抗抵赖服务包括证据的生成、验证和记录，以及在解决纠纷时随即进行的证据恢复和再次验证。

框架中抗抵赖服务的目的是提供有关特定事件或行为的证据。例如，必须确认数据原发者和接收者的身份和数据完整性，在某些情况下，可能需要涉及上下文关系(如日期、时间、原发者/接收者的地点等)的证据，等等。