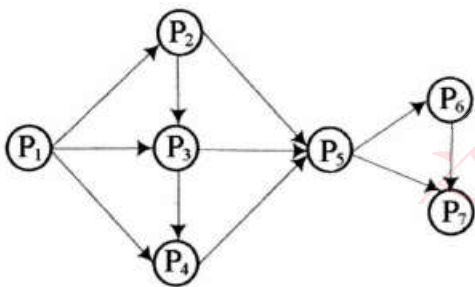


2020 年系统架构设计师真题及解析

上午基础知识

试题(1)前趋图(Precedence Graph)是一个有向无环图,记为: $\rightarrow = \{(P_i, P_j) | P_j \text{ must complete before } P_j \text{ may start}\}$ 。假设系统中进程 $P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$, 且进程的前趋图如下:



那么, 该前驱图可记为(1)。

- A. $\rightarrow = \{(P_1, P_2), (P_3, P_1), (P_4, P_1), (P_5, P_2), (P_5, P_3), (P_6, P_4), (P_7, P_5), (P_7, P_6), (P_5, P_6), (P_4, P_5), (P_6, P_7)\}$
- B. $\rightarrow = \{(P_1, P_2), (P_1, P_3), (P_1, P_4), (P_2, P_5), (P_2, P_3), (P_3, P_4), (P_3, P_5), (P_4, P_5), (P_5, P_6), (P_5, P_7), (P_6, P_7)\}$
- C. $\rightarrow = \{(P_1, P_2), (P_1, P_3), (P_1, P_4), (P_2, P_5), (P_2, P_3), (P_3, P_4), (P_5, P_3), (P_4, P_3), (P_5, P_6), (P_7, P_5), (P_6, P_7)\}$
- D. $\rightarrow = \{(P_1, P_2), (P_1, P_3), (P_2, P_3), (P_2, P_5), (P_3, P_6), (P_3, P_4), (P_4, P_7), (P_5, P_6), (P_6, P_7), (P_6, P_5), (P_7, P_5)\}$

试题(1)分析本题考查操作系统的基本概念。

前趋图是一个有向无环图, 记为 DAG (Directed Acyclic Graph), 用于描述进程之间执行的前后关系。图中的每个结点可用于描述一个程序段或进程, 乃至一条语句; 结点间的有向边则用于表示两个结点之间存在的偏序 (Partial Order, 亦称偏序关系) 或前趋关系 (Precedence Relation) “ \rightarrow ”

对于试题所示的前趋图, 存在前趋关系: $(P_1, P_2), (P_1, P_3), (P_1, P_4), (P_2, P_5), (P_2, P_3), (P_3, P_4), (P_3, P_5), (P_4, P_5), (P_5, P_6), (P_5, P_7), (P_6, P_7)$

可记为: $P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$

$\rightarrow = \{(P_1, P_2), (P_1, P_3), (P_1, P_4), (P_2, P_5), (P_2, P_3), (P_3, P_4), (P_3, P_5), (P_4, P_5), (P_5, P_6), (P_5, P_7), (P_6, P_7)\}$

注意：在前趋图中，没有前趋的结点称为初始结点(Initial Node)，没有后继的结点称为终止结点(Final Node)。

参考答案(1)B

试题(2)在支持多线程的操作系统中，假设进程P创建了线程T1、T2和T3，那么下列说法正确的是(2)。

- A. 该进程中已打开的文件是不能被T1、T2和T3共享的
- B. 该进程中T1的栈指针是不能被T2共享的，但可被T3共享
- C. 该进程中T1的栈指针是不能被T2和T3共享的
- D. 该进程中某线程的栈指针是可以被T1、T2和T3共享的

试题(2)分析在同一进程中的各个线程都可以共享该进程所拥有的资源，如访问进程地址空间中的每一个虚地址；访问进程所拥有的已打开文件、定时器、信号量等，但是不能共享进程中某线程的栈指针。

参考答案(2)C

试题(3)假设某计算机的字长为32位，该计算机文件管理系统磁盘空间管理采用位示图(bitmap)记录磁盘的使用情况。若磁盘的容量为300GB，物理块的大小为4MB，那么位示图的大小为(3)个字。

- (3)A. 2400 B. 3200 C. 6400 D. 9600

试题(3)分析本题考查操作系统文件管理方面的基础知识。

根据题意，若磁盘的容量为300GB，物理块的大小为4MB，则该磁盘的物理块数为 $300 \times 1024 / 4 = 76800$ 个，位示图的大小为 $76800 / 32 = 2400$ 个字。

参考答案(3)A

试题(4)实时操作系统主要用于有实时要求的过程控制等领域。因此，在实时操作系统中，对于来自外部的事件必须在(4)。

- A. 一个时间片内进行处理
- B. 一个周转时间内进行处理
- C. 一个机器周期内进行处理
- D. 被控对象允许的时间范围内进行处理

试题(4)分析本题考查操作系统的基础知识。

实时是指计算机对于外来信息能够以足够快的速度进行处理，并在被控对象允许的时间范围内做出快速响应。因此，实时操作系统与分时操作系统的第一点区别是交互性强弱不同，

分时系统交互性强，实时系统交互性弱但可靠性要求高；第二点区别是对响应时间的敏感性 强，对随机发生的外部事件必须在被控制对象规定的时间内做出及时响应并对其进行处理； 第三点区别是系统的设计目标不同，分时系统是设计成一个多用户的通用系统，交互能力强；而实时系统大都是专用系统。

参考答案 (4) D

试题 (5) 通常在设计关系模式时，派生属性不会作为关系中的属性来存储。按照这个原则，原设计的学生关系模式为 Students (学号, 姓名, 性别, 出生日期, 年龄, 家庭地址)，那么该关系模式正确的设计应为 (5)。

- A. Students (学号, 性别, 出生日期, 年龄, 家庭地址)
- B. Students (学号, 姓名, 性别, 出生日期, 年龄)
- C. Students (学号, 姓名, 性别, 出生日期, 家庭地址)
- D. Students (学号, 姓名, 出生日期, 年龄, 家庭地址)

试题 (5) 分析本题考查关系数据库方面的基本概念。

在概念设计中，需要概括应用系统中的实体及其联系，确定实体和联系的属性。派生属性是指可以由其他属性通过计算来获得，若在系统中存储派生属性，会引起数据冗余，增加额外存储和维护负担，还可能导致数据的不一致性，故派生属性不会作为关系中的属性来存储。

本题中“年龄”是派生属性，该属性可以由“系统当前时间-出生日期” 计算获得，故关系模式 Students 正确的设计是“年龄”不作为关系中的属性来存储。

参考答案 (5) C

试题 (6)、(7)

给出关系 $R(U, F)$, $U = \{A, B, C, D, E\}$, $F = \{A \rightarrow B, D \rightarrow C, BC \rightarrow E, AC \rightarrow B\}$, 求属性闭包的等式成立的是 (6)。 R 的候选关键字为 (7)。

- (6) A. $(A)_F^+ = U$ B. $(B)_F^+ = U$ C. $(AC)_F^+ = U$ D. $(AD)_F^+ = U$
- (7) A. AD B. AB C. AC D. BC

试题 (6)、(7) 分析本题考查关系数据库理论方面的基础知识。

设 F 为属性集 U 上的一组函数依赖, $X \subseteq U$, $X_F^+ = \{A | X \rightarrow A \text{ 能由 } F \text{ 根据 Armstrong 公理导出}\}$, 则称 X_F^+ 为属性集 X 关于函数依赖集 F 的闭包。

根据以上定义及求属性闭包算法, 分别求解属性集闭包 $(A)_F^+$ 、 $(B)_F^+$ 、 $(AC)_F^+$ 、 $(AD)_F^+$, 并判断等式是否成立。

求解 $(A)_F^+$ 。根据 F 中的 $A \rightarrow B$ 函数依赖, 可求得 $(A)_F^+ = AB \neq U$ 。

求解 $(B)_F^+$ 。由于 F 中不存在左部为 B 的函数依赖, 故 $(B)_F^+ = B \neq U$ 。

求解 $(AC)_F^+$ 。根据 F 中的 $A \rightarrow B$ 函数依赖，可求得 $(AC)_F^+ = ABC \neq U$ 。

求解 $(AD)_F^+$ 。根据 F 中的 $A \rightarrow B, D \rightarrow C, BC \rightarrow E$ 函数依赖，通过求属性闭包算法可以求得 $(AD)_F^+ = ABCDE = U$ 。

由于在属性集 AD 中不存在一个真子集能决定全属性，故 AD 为 R 的候选码。

参考答案(6)D (7) A

试题(8)在分布式数据库中有分片透明、复制透明、位置透明和逻辑透明等基本概念。其中，(8)是指用户无需知道数据存放的物理位置。

(8)A. 分片透明 B. 逻辑透明 C. 位置透明 D. 复制透明

试题(8)分析本题考查对分布式数据库基本概念的理解。

分片透明是指用户或应用程序不需要知道逻辑上访问的表具体是怎么分块存储的。复制透明是指采用复制技术的分布方法，用户不需要知道数据是复制到哪些节点，如何复制的。位置透明是指用户无需知道数据存放的物理位置。逻辑透明是指用户或应用程序无需知道局部场地使用的是哪种数据模型。

参考答案(8)C

试题(9)以下关于操作系统微内核架构特征的说法，不正确的是(9)。

- A. 微内核的系统结构清晰，利于协作开发
- B. 微内核代码量少，系统具有良好的可移植性
- C. 微内核有良好的伸缩性、扩展性
- D. 微内核的功能代码可以互相调用，性能很高

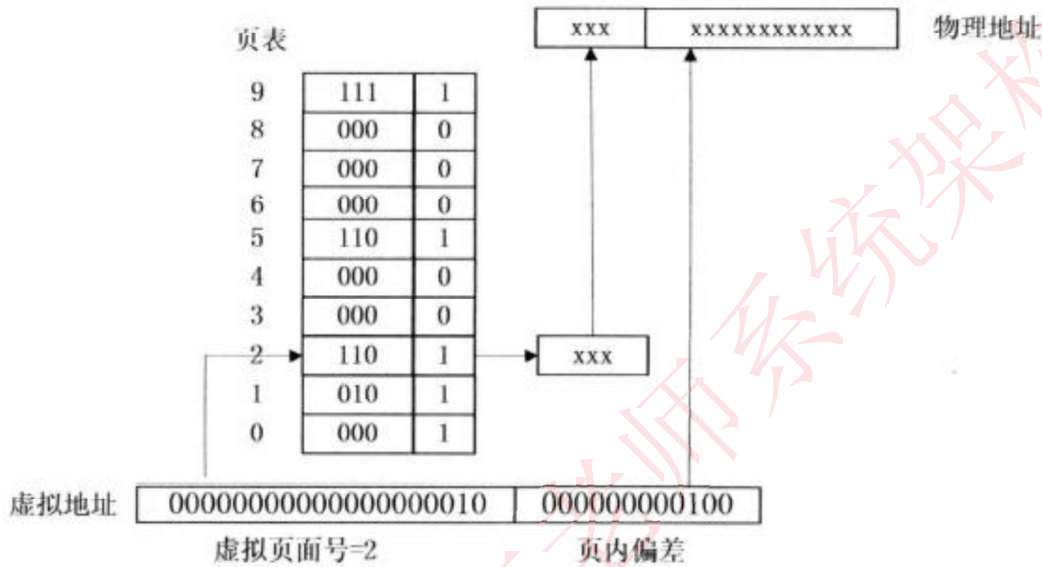
试题(9)分析本题考查操作系统的基础知识。

微内核(Micro Kernel)是现代操作系统普遍采用的架构形式。它是一种能够提供必要服务的操作系统内核，被设计成在很小的内存空间内增加移植性，提供模块设计，这些必要的服务包括任务、线程、交互进程通信以及内存管理等。而操作系统其他所有服务(含设备驱动)在用户模式下运行，可以使用户安装不同的服务接口(API)。

微内核的主要优点在于结构清晰、内核代码量少，安全性和可靠性高、可移植性强、可伸缩性、可扩展性高；其缺点是难以进行良好的整体优化、进程间互相通信的开销大、内核功能代码不能被直接调用而带来服务的效率低。

参考答案(9)D

试题(10)分页内存管理的核心是将虚拟内存空间和物理内存空间皆划分成大小相同的页面，并以页面作为内存空间的最小分配单位。下图给出了内存管理单元的虚拟地址到物理地址的翻译过程，假设页面大小为 4KB, 那么 CPU 发出虚拟地址 0010000000000100 后，其访问的物理地址是(10)。



- A. 1100000000000100
- B. 0100000000000100
- C. 1100000000000000
- D. 1100000000000010

试题(10)分析 本题考查计算机内存管理的基础知识。

虚拟内存管理是计算机体系结构设计中必须考虑的问题。计算机内存管理通过段页式管理算法，可以使计算机内存容量被无限延伸，以提升计算机处理能力。

分页式管理是将一个进程的逻辑地址空间分成若干个大小相等的片，称之为页面或页，并为各页加以编号，从 0 开始编码。相应地也把内存空间分成与页面相同大小的若干个存储块，称之为物理块或页框，也同为它们加以编号。在为进程分配内存时，以块为单位将进程中若干个页分别装入多个可以不相邻的物理块中，从而实现无存储碎片的管理。分页式管理中，通常进程使用的地址是一种虚拟存储地址，必须通过页表转换才能访问到实际物理地址，虚拟地址一般由页面号和页内偏移组成，页面号是指需要访问页表的序号，而页内偏移是指在某页内相对 0 地址的偏移值。

因此，本题中给出虚拟地址 0010000000000100 中的页表序号是 02 (10), 图中页表 2 序列中内容是 110, 因此物理地址应该是 110 加偏移地址，即 1100000000000100 是正确答案。

参考答案(10) A

试题(11) 以下关于计算机内存管理的描述中，(11)属于段页式内存管理的描述。

- A. 一个程序就是一段，使用基址极限对来进行管理
- B. 一个程序分为许多固定大小的页面，使用页表进行管理
- C. 程序按逻辑分为多段，每一段内又进行分页，使用段页表来进行管理

D. 程序按逻辑分成多段，用一组基址极限对来进行管理。基址极限对存放在段表里

试题(11)分析本题考查计算机内存管理的基础知识。

计算机内存管理有多种管理算法，从发展历史看，内存管理经历了固定分区、非固定分区、页式、段式和段页式等方法，当前较流行的是段页式内存管理。

页式内存管理：其核心是将虚拟内存空间和物理内存空间皆划分成大小相同的页面，并以页面作为内存空间的最小分配单位。一个程序的一个页面可以放在任意一个物理页面里。

段式内存管理：其核心是将一个程序按照逻辑单元分成多个程序段，每一个段使用自己单独的虚拟地址空间。采用段页表来进行管理。比如编译器可以将一个程序分成5个虚拟空间，即符号表、代码段、常数段、数据段和调用栈。

因此，选项A的管理方法属于分区式管理；选项B的管理方法属于页式管理；选项D的管理方法属于段式管理；只有选项C的管理方法属于段页式管理。

参考答案(11)C

试题(12)软件脆弱性是软件中存在的弱点(或缺陷)，利用它可以危害系统安全策略，导致信息丢失、系统价值和可用性降低。嵌入式系统软件架构通常采用分层架构，它可以将问题分解为一系列相对独立的子问题，局部化在每一层中，从而有效地降低单个问题的规模和复杂性，实现复杂系统的分解。但是，分层架构仍然存在脆弱性。常见的分层架构的脆弱性包括(12)等两个方面。

- A. 底层发生错误会导致整个系统无法正常运行、层与层之间功能引用可能导致功能失效
- B. 底层发生错误会导致整个系统无法正常运行、层与层之间引入通信机制势必造成性能下降
- C. 上层发生错误会导致整个系统无法正常运行、层与层之间引入通信机制势必造成性能下降
- D. 上层发生错误会导致整个系统无法正常运行、层与层之间功能引用可能导致功能失效

试题(12)分析本题考查软件架构脆弱性方面的基础知识。

脆弱性表示人、事物、组织机构等面对波动性、随机性变化或者压力时表现出来的变化趋势，软件脆弱性是指软件中存在的弱点(或缺陷)，利用它可以危害系统安全策略，导致信息丢失、系统价值和可用性降低等。通常在软件设计时，分层架构由于其良好的可扩展性和可维护性被广泛采纳，但是，分层架构也存在众多脆弱性问题，主要表现在以下两个方面：

① 一旦某个底层发生错误，那么整个程序将会无法正常运行，如产生一些数据溢出、空指针、空对象的安全问题，也有可能得出错误的结果；

② 将系统隔离为多个相对独立的层，这就要求在层与层之间引入通信机制，这种本来“直来直去”的操作现在要层层传递，势必造成性能的下降。

参考答案(12)B

试题（13）以下关于区块链应用系统中“挖矿”行为的描述中，错误的是（13）。

- A. 矿工“挖矿”取得区块链的记账权，同时获得代币奖励
- B. “挖矿”本质上是在尝试计算一个 Hash 碰撞
- C. “挖矿”是一种工作量证明机制
- D. 可以防止比特币的双花攻击

试题（13）分析本题考查区块链的基础知识。

以区块链技术最成功的应用比特币为例，矿工的“挖矿”行为，其动机是为了获得代币奖励；其技术本质是尝试计算一个 Hash 碰撞，从而完成工作量证明；对社区而言，成功挖矿的矿工获得记账权和代币奖励是区块链应用系统的激励机制，是社区自我维持的关键。然而，挖矿行为自身并不能防止双花攻击（即一笔钱可以花出去两次）。

参考答案（13）D

试题（14）在 Linux 系统中，DNS 的配置文件是（14），它包含了主机的域名搜索顺序和 DNS 服务器的地址。

- A. /etc/hostname
- B. /dev/host.conf
- C. /etc/resolv.conf
- D. /dev/name.conf

试题（14）分析本题考查 Linux 中 DNS 的配置知识。

在 Linux 中，DNS 的配置文件保存在/etc/resolv.conf。/etc/resolv.conf 是 DNS 客户机的配置文件，用于设置 DNS 服务器的 IP 地址及 DNS 域名，还包含了主机的域名搜索顺序。该文件是由域名解析器（一个根据主机名解析 IP 地址的库）使用的配置文件。它的格式比较简单，每行以一个关键字开头，后接一个或多个由空格隔开的参数。

参考答案（14）C

试题（15）下面关于网络延迟的说法中，正确的是（15）。

- A. 在对等网络中，网络的延迟大小与网络中的终端数量无关
- B. 使用路由器进行数据转发所带来的延迟小于交换机
- C. 使用 Internet 服务能够最大限度地减小网络延迟
- D. 服务器延迟的主要影响因素是队列延迟和磁盘 I/O 延迟

试题（15）分析本题考查网络延迟的基础知识。

网络中的延迟产生与以下几个方面有关：运算、读取和写入、数据传输以及数据传输过程中的拥塞所带来的延迟。在网络中，数据读写的速率较之于数据计算和传输的速率要小得多，因此数据读写的延迟是影响网络延迟的最大的因素。

在对等网络中，由于采用总线式的连接，因此网络中的终端数量越多，终端所能够分配到的转发时隙就越小，所带来的延迟也就越大。

路由器一般采用存储转发方式，需要对待转发的数据包进行重新拆包，分析其源地址和目的地址，再根据路由表对其进行路由和转发，而交换机采取的是直接转发方式，不对数据包的三层地址进行分析，因此路由器转发所带来的延迟要小于交换机。

数据在 Internet 中传输时，由于互联网中的转发数据量大且所需经过的节点多，势必会带来更大的延迟。

参考答案(15)D

试题(16)、(17)进行系统监视通常有三种方式：一是通过(16)，如 UNIX/Linux 系统中的 ps、last 等；二是通过系统记录文件查阅系统在特定时间内的运行状态；三是集成命令、文件记录和可视化技术的监控工具，如(17)。

(16)A. 系统命令 B. 系统调用 C. 系统接口 D. 系统功能

(17)A. Windows 的 netstat B. Linux 的 iptables

C. Windows 的 Perfmon D. Linux 的 top

试题(16)、(17)分析本题考查系统安全知识。

Windows 的 netstat 命令用来查看某个端口号是否被占用以及由哪个进程占用。

Perfmon (Performance Monitor)是 Windows 自带的性能监控工具，提供了图表化的系统性能实时监视器、性能日志和警报管理。通过添加性能计数器(Performance Counter)可以实现对 CPU、内存、网络、磁盘、进程等多类对象的上百个指标的监控。

iptables 是在 Linux2.4 内核之后普遍使用的基于包过滤的防火墙工具，可以对流入和流出服务器的数据包进行很精细的控制。

top 命令是 Linux 下常用的性能分析工具，能够实时显示系统中各个进程的资源占用状况。

参考答案(16)A (17) C

试题(18)~(21)与电子政务相关的行为主体主要有三类，即政府、企(事)业单位及居民。因此，政府的业务活动也主要围绕着这三类行为主体展开。政府与政府、政府与企(事)业单位以及政府与居民之间的互动构成了 5 种不同的、却又相互关联的领域。其中人口信息采集、处理和利用业务属于(18)领域：营业执照的颁发业务属于(19)领域：户籍管理业务属于(20)领域：参加政府工程投标活动属于(21)领域。

(18)A. 政府对企(事)业单位(G2B) B. 政府与政府(G2G)

C. 企业对政府(B2G) D. 政府对居民(G2C)

-
- (19) A. 政府对企（事）业单位（G2B） B. 政府与政府（G2G）
C. 企业对政府（B2G） D. 政府对居民（G2C）
- (20) A. 政府对企（事）业单位（G2B） B. 政府与政府（G2G）
C. 企业对政府（B2G） D. 政府对居民（G2C）
- (21) A. 政府对企（事）业单位（G2B） B. 政府与政府（G2G）
C. 企业对政府（B2G） D. 政府对居民（G2C）

试题（18）～（21）分析与电子政务相关的行为主体主要有三个，即政府、企（事）业单位及居民。

因此，政府的业务活动也主要围绕着这三个行为主体展开。政府与政府，政府与企（事）业单位，以及政府与居民之间的互动构成了下面5个不同的、却又相互关联的领域。

1) 政府与政府（G2G）

政府与政府之间的互动包括首脑机关与中央和地方政府组成部门之间的互动，中央政府与各级地方政府之间，政府的各个部门之间、政府与公务员和其他政府工作人员之间的互动。这个领域涉及的主要是政府内部的政务活动，包括国家和地方基础信息的采集、处理和利用，如人口信息；政府之间各种业务流所需要采集和处理的信息，如计划管理；政府之间的通信系统，如网络系统；政府内部的各种管理信息系统，如财务管理；以及各级政府的决策支持系统和执行信息系统，等等。

2) 政府对企（事）业单位（G2B）

政府面向企业的活动主要包括政府向企（事）业单位发布的各种方针、政策、法规、行政规定，即企（事）业单位从事合法业务活动的环境；政府向企（事）业单位颁发的各种营业执照、许可证、合格证和质量认证等。

3) 政府对居民（G2C）

政府对居民的活动实际上是政府面向居民所提供的服务。政府对居民的服务首先是信息服务，让居民知道政府的规定是什么，办事程序是什么，主管部门在哪里，以及各种关于社区公安和水、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等政府面向居民提供的各种服务。政府对居民提供的服务还包括各公共部门，如学校、医院、图书馆和公园等。

4) 企业对政府（B2G）

企业面向政府的活动包括企业应向政府缴纳的各种税款，按政府要求应该填报的各种统计信息和报表，参加政府各项工程的竞、投标，向政府供应各种商品和服务，以及就政府如何创造良好的投资和经营环境，如何帮助企业发展等提出企业的意见和希望，反映企业在经营活动中遇到的困难，提出可供政府采纳的建议，向政府申请可能提供的援助等等。

5) 居民对政府（C2G）

居民对政府的活动除了包括个人应向政府缴纳的各种税款和费用,按政府要求应该填报的各种信息和表格,以及缴纳各种罚款等外,更重要的是开辟居民参政、议政的渠道,使政府的各项工作不断得以改进和完善。政府需要利用这个渠道来了解民意,征求群众意见,以便更好地为人民服务。此外,报警服务(盗贼、医疗、急救、火警等)即在紧急情况下居民需要向政府报告并要求政府提供的服务,也属于这个范围。

参考答案(18) B (19) A (20) D (21) C

试题(22)、(23)软件文档是影响软件可维护性的决定因素。软件的文档可以分为用户文档和(22)两类。其中,用户文档主要描述(23)和使用方法,并不关心这些功能是怎样实现的。

(22)A. 系统文档 B. 需求文档 C. 标准文档 D. 实现文档

(23)A. 系统实现 B. 系统设计 C. 系统功能 D. 系统测试

试题(22)、(23)分析本题考查软件文档的相关知识。

软件文档是影响软件可维护性的决定因素。根据文档内容,软件文档又可分为用户文档和系统文档两类。其中,用户文档主要描述系统功能和使用方法,并不关心这些功能是怎样实现的。

参考答案(22)A (23) C

试题(24)、(25)软件需求开发的最终文档经过评审批准后,就定义了开发工作的(24),它在客户和开发者之间构筑了产品功能需求和非功能需求的一个(25),是需求开发和需求管理之间的桥梁。

(24)A. 需求基线 B. 需求标准 C. 需求用例 D. 需求分析

(25)A. 需求用例 B. 需求管理标准 C. 需求约定 D. 需求变更

试题(24)、(25)分析本题考查软件需求工程的相关知识。

需求基线指已经通过正式评审和批准的规格说明或产品,可作为进一步开发的基础,而且只有通过正式的变更控制过程才能修改它。建立需求基线的目的是防止需求的变化给程序架构造成重大影响。因此,它是团队成员已经承诺将在某一特定产品版本中实现的功能性和非功能性需求的一组集合,它在客户和开发者之间构筑了一个需求约定,是需求开发和需求管理之间的桥梁。

参考答案(24) A (25) C

试题(26)~(28)软件过程是制作软件产品的一组活动及其结果。这些活动主要由软件人员来完成,软件活动主要包括软件描述、(26)、软件有效性验证和(27)。其中,(28)定义了软件功能以及使用的限制。

(26)A. 软件模型 B. 软件需求 C. 软件分析 D. 软件开发

(27)A. 软件分析 B. 软件测试 C. 软件演化 D. 软件开发

(28)A. 软件分析 B. 软件测试 C. 软件描述 D. 软件开发

试题(26)～(28)分析本题考查软件过程的相关知识。

软件过程(SoftwareProcedure)是指软件生存周期所涉及的一系列相关过程。过程是活动的集合；活动是任务的集合；任务起着把输入进行加工然后输出的作用。活动的执行可以是顺序的、重复的、并行的、嵌套的或者是有条件地引发的。软件过程是指软件整个生命周期，包括需求获取、需求分析、设计、实现、测试、发布和维护的一个过程模型。一个软件过程定义了软件开发中采用的方法，但软件过程还包含该过程中应用的技术方法和自动化工具。过程定义一个框架，为有效交付软件，这个框架必须创建。软件过程构成了软件项目管理控制的基础，并且创建了一个环境以便于技术方法的采用、工作产品(模型、文档、报告、表格等)的产生、里程碑的创建、质量的保证、正常变更的正确管理。

软件过程中的活动主要由软件人员来完成，软件活动主要包括软件描述、软件开发、软件有效性验证和软件演化。其中，软件描述定义了软件功能以及使用的限制。

参考答案(26) D (27) C (28) C

试题(29)、(30)对应软件开发过程的各种活动，软件开发工具有需求分析工具、(29)、编码与排错工具、测试工具等。按描述需求定义的方法可将需求分析工具分为基于自然语言或图形描述的工具和基于(30)的工具。

(29)A. 设计工具 B. 分析工具 C. 耦合工具 D. 监控工具

(30)A. 用例 B. 形式化需求定义语言 C. UML D. 需求描述

试题(29)、(30)分析本题考查软件系统工具的相关知识。

软件系统工具的种类繁多，很难有统一的分类方法。通常可以按软件过程活动将软件工具分为软件开发工具、软件维护工具、软件管理和软件支持工具。其中，对应软件开发过程的各种活动，软件开发工具有需求分析工具、设计工具、编码与排错工具、测试工具等。

需求分析工具用以辅助软件需求分析活动，辅助系统分析员从需求定义出发，生成完整的、清晰的、一致的功能规范。功能规范是软件所要完成的功能精确而完整的陈述，描述该软件要做什么及只做什么，是软件开发者和用户间的契约，同时也是软件设计者和实现者的依据。功能规范应正确、完整地反映用户对软件的功能要求，其表达是清晰的、无歧义的。需求分析工具的目标就是帮助分析员形成这样的功能规范。按描述需求定义的方法可将需求分析工具分为基于自然语言或图形描述的工具和基于形式化需求定义语言的工具。

参考答案(29) A (30) B

试题(31)、(32)软件设计包括四个既独立又相互联系的活动：(31)、软件结构设计、人机界面设计和(32)。

(31) A. 用例设计 B. 数据设计 C. 程序设计 D. 模块设计

(32) A. 接口设计 B. 操作设计 C. 输入输出设计 D. 过程设计

试题(31)、(32)分析 本题考查软件设计的基础知识。

软件设计包括四个既独立又相互联系的活动,即数据设计、软件结构设计、人机界面设计和过程设计,这四个活动完成以后就得到了全面的软件设计模型。

参考答案(31) B (32) D

试题(33)、(34)信息隐蔽是开发整体程序结构时使用的法则,通过信息隐蔽可以提高软件的(33)、可测试性和(34)。

(33) A. 可修改性 B. 可扩充性 C. 可靠性 D. 耦合性

(34) A. 封装性 B. 安全性 C. 可移植性 D. 可交互性

试题(33)、(34)分析 本题考查软件结构化设计的基础知识。

信息隐蔽是开发整体程序结构时使用的法则,即将每个程序的成分隐蔽或封装在一个单一的设计模块中,并且尽可能少地暴露其内部的处理过程。通过信息隐蔽可以提高软件的可修改性、可测试性和可移植性,它也是现代软件设计的一个关键性原则。

参考答案(33) A (34) C

试题(35)

按照外部形态,构成一个软件系统的构件可以分为五类,其中,(35)是指可以进行版本替换并增加构件新功能。

(35) A. 装配的构件 B. 可修改的构件 C. 有限制的构件 D. 适应性构件

试题(35)分析 本题考查软件构件的基础知识。

如果把软件系统看成是构件的集合,那么从构件的外部形态来看,构成一个系统的构件可分为五类:独立而成熟的构件得到了实际运行环境的多次检验;有限制的构件提供了接口,指出了使用的条件和前提;适应性构件进行了包装或使用了接口技术,把不兼容性、资源冲突等进行了处理,可以直接使用;装配的构件在安装时,已经装配在操作系统、数据库管理系统或信息系统不同层次上,可以连续使用;可修改的构件可以进行版本替换,如果对原构件修改错误、增加新功能,可以利用重新“包装”或写接口来实现构件的替换。

参考答案(35) B

试题(36)~(38)中间件是提供平台和应用之间的通用服务，这些服务具有标准的程序接口和协议。中间件的基本功能包括：为客户端和服务端之间提供(36)；提供(37)保证交易的一致性；提供应用的(38)。

- (36) A. 连接和通信 B. 应用程序接口 C. 通信协议支持 D. 数据交换标准
(37) A. 安全控制机制 B. 交易管理机制 C. 标准消息格式 D. 数据映射机制
(38) A. 基础硬件平台 B. 操作系统服务 C. 网络和数据库 D. 负载均衡和高可用性

试题(36)~(38)分析 本题考查中间件的基础知识。

中间件提供平台和应用之间的通用服务，这些服务具有标准的程序接口和协议。中间件的基本功能包括：为客户端和服务端之间提供连接和通信；提供交易管理机制保证交易的一致性；提供应用的负载均衡和高可用性等。

参考答案(36) A (37) B (38) D

试题(39)、(40)应用系统开发中可以采用不同的开发模型，其中，(39)将整个开发流程分为目标设定、风险分析、开发和有效性验证、评审四个部分；(40)则通过重用来提高软件的可靠性和易维护性，程序在进行修改时产生较少的副作用。

- (39) A. 瀑布模型 B. 螺旋模型 C. 构件模型 D. 对象模型
(40) A. 瀑布模型 B. 螺旋模型 C. 构件模型 D. 对象模型

试题(39)、(40)分析 本题考查软件开发模型的基础知识。

应用系统开发中可以采用不同的开发模型，包括瀑布模型、演化模型、原型模型、螺旋模型、喷泉模型和基于可重用构件的模型等。其中，螺旋模型将整个开发流程分为目标设定、风险分析、开发和有效性验证、评审四个部分；构件则通过重用来提高软件的可靠性和易维护性，程序在进行修改时产生较少的副作用。

参考答案(39) B (40) C

试题(41)关于敏捷开发方法的特点，不正确的是(41)。

- A. 敏捷开发方法是适应性而非预设性
B. 敏捷开发方法是面向过程的而非面向人的
C. 采用迭代增量式的开发过程，发行版本小型化
D. 敏捷开发中强调开发过程中相关人员之间的信息交流

试题(41)分析 本题考查敏捷开发方法的基础知识。

敏捷开发方法主要有两个特点：敏捷开发方法是适应性而非预设性的；敏捷开发方法是面向人而非面向过程的。敏捷开发方法以原型化开发方法为基础，采用迭代增量式开发，发行版本小型化。敏捷开发方法特别强调开发中相关人员之间的信息交流。

参考答案(41) B

试题(42)、(43)自动化测试工具主要使用脚本技术来生成测试用例，其中，(42)是录制手工测试的测试用例时得到的脚本；(43)是将测试输入存储在独立的数据文件中，而不是在脚本中。

(42) A. 线性脚本 B. 结构化脚本 C. 数据驱动脚本 D. 共享脚本

(43) A. 线性脚本 B. 结构化脚本 C. 数据驱动脚本 D. 共享脚本

试题(42)、(43)分析本题考查软件测试的基础知识。

自动化测试工具主要使用脚本技术来生成测试用例，脚本是一组测试工具执行的指令集合。脚本的基本结构主要有五种：线性脚本是录制手工测试的测试用例时得到的脚本；结构化脚本具有各种逻辑结构和函数调用功能；共享脚本是指一个脚本可以被多个测试用例使用；数据驱动脚本是指将测试输入存储在独立的数据文件中，而不是脚本中；关键字驱动脚本是数据驱动脚本的逻辑扩展，用测试文件描述测试用例。

参考答案(42) A (43) C

试题(44)~(47)考虑软件架构时，重要的是从不同的视角(perspective)来检查，这促使软件设计师考虑架构的不同属性。例如，展示功能组织的(44)能判断质量特性，展示并发行为的(45)能判断系统行为特性。选择的特定视角或视图也就是逻辑视图、进程视图、实现视图和(46)。使用(47)来记录设计元素的功能和概念接口，它本身在系统中的角色，这些角色包括功能、性能等。

(44) A. 静态视角 B. 动态视角 C. 多维视角

(45) A. 开发视角 B. 动态视角 C. 部署视角

(46) A. 开发视图 B. 配置视图 C. 部署视图 D. 物理视图

(47) A. 逻辑视图 B. 物理视图 C. 部署视图 D. 用例视图

试题(44)~(47)分析本题考查软件架构的相关知识。

在软件架构中，从不同的视角描述特定系统的体系结构，从而得到多个视图，并将这些视图组织起来以描述整体的软件架构模型。因此，在考虑体系结构时，可以从不同的视角来检查，这促使软件设计师考虑体系结构的不同属性。例如，展示功能组织的静态视角能判断质量特性，展示并发行为的动态视角能判断系统行为特性。选择的特定视角或视图也就是逻辑视图、进程视图、实现视图和配置视图。使用逻辑视图来记录设计元素的功能和概念接口，设计元素的功能定义了它本身在系统中的角色，这些角色包括功能、性能等。

参考答案(44) A (45) B (46) B (47) A

试题(48)~(50)在软件架构评估中,(48)是影响多个质量属性的特性,是多个质量属性的(49)。例如,提高加密级别可以提高安全性,但可能要耗费更多的处理时间,影响系统性能。如果某个机密消息的处理有严格的时间延迟要求,则加密级别可能就会成为一个(50)。

(48) A. 敏感点 B. 权衡点 C. 风险决策 D. 无风险决策

(49) A. 敏感点 B. 权衡点 C. 风险决策 D. 无风险决策

(50) A. 敏感点 B. 权衡点 C. 风险决策 D. 无风险决策

试题(48)~(50)分析 本题考查体系结构评估的相关知识。

敏感点(sensitivitypoint)和权衡点(tradeoffpoint)是关键的体系结构决策。敏感点是一个或多个构件(和/或构件之间的关系)的特性。研究敏感点可使设计人员或分析员明确在搞清楚如何实现质量目标时应注意什么。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。因此,改变加密级别可能会对安全性和性能产生非常重要的影响。提高加密级别可以提高安全性,但可能要耗费更多的处理时间,影响系统性能。如果某个机密消息的处理有严格的时间延迟要求,则加密级别可能就会成为一个权衡点。

参考答案(48) B (49) A (50) B

试题(51)~(53)针对二层C/S软件架构的缺点,三层C/S架构应运而生。在三层C/S架构中,增加了一个(51)。三层C/S架构是将应用功能分成表示层、功能层和(52)三部分。其中(53)是应用的用户接口部分,担负与应用逻辑间的对话功能。

(51) A. 应用服务器 B. 分布式数据库 C. 内容分发 D. 镜像

(52) A. 硬件层 B. 数据层 C. 设备层 D. 通信层

(53) A. 表示层 B. 数据层 C. 应用层 D. 功能层

试题(51)~(53)分析 本题考查软件架构中三层C/S架构的相关知识。

传统的二层C/S结构存在以下几个局限:是单一服务器且以局域网为中心的,所以难以扩展至大型企业广域网或Internet;受限于供应商;软硬件的组合及集成能力有限;难以管理大量的客户机。因此,三层C/S结构应运而生。

三层C/S结构是将应用功能分成表示层、功能层和数据层三部分,其解决方案是对这三层进行明确分割,并在逻辑上使其独立。原来的数据层作为DBMS已经独立出来,将表示层和功能层分离成各自独立的程序,使这两层间的接口简洁明了。三层C/S结构中,表示层是应用的用户接口部分,它担负着用户与应

用间的对话功能。它用于检查用户从键盘等输入的数据，显示应用输出的数据。功能层相当于应用的本体，它是将具体的业务处理逻辑编入程序中。数据层就是 DBMS，负责管理对数据库数据的读写。

参考答案(51) A (52) B (53) A

试题(54)、(55)经典的设计模式共有 23 个，这些模式可以按两个准则来分类：一是按设计模式的目的划分，可分为(54)型、结构型和行为型三种模式；二是按设计模式的范围划分，可以把设计模式分为类设计模式和(55)设计模式。

(54)A. 创建 B. 实例 C. 代理 D. 协同

(55)A. 包 B. 模板 C. 对象 D. 架构

试题(54)、(55)分析软件模式主要可分为设计模式、分析模式、组织和过程模式等，每一类又可细分为若干个子类。在此着重介绍设计模式，目前它的使用最为广泛。设计模式主要用于得到简洁灵活的系统设计，GoF 的书中共有 23 个设计模式，这些模式可以按两个准则来分类：一是按设计模式的目的划分，可分为创建型、结构型和行为型三种模式；二是按设计模式的范围划分，即根据设计模式是作用于类还是作用于对象来划分，可以把设计模式分为类设计模式和对象设计模式。

参考答案(54)A (55) C

试题(56)~(58)创建型模式支持对象的创建，该模式允许在系统中创建对象，而不需要在代码中标识特定类的类型，这样用户就不需要编写大量、复杂的代码来初始化对象。在不指定具体类的情况下，模式为创建一系列相关或相互依赖的对象提供了一个接口。(57)模式将复杂对象的构建与其表示相分离，这样相同的构造过程可以创建不同的对象。(58)模式允许对象在不了解要创建对象的确切类以及如何创建等细节的情况下创建自定义对象。

(56) A. Prototype B. Abstract Factory C. Builder D. Singleton

(57) A. Prototype B. Abstract Factory C. Builder D. Singleton

(58) A. Prototype B. Abstract Factory C. Builder D. Singleton

试题(56)~(58)分析在系统中，创建性模式支持对象的创建。该模式允许在系统中创建对象，而不需要在代码中标识特定类的类型，这样用户就不需要编写大量、复杂的代码来初始化对象。它是通过该类的子类来创建对象的。

在不指定具体类的情况下，Abstract Factory 模式为创建一系列相关或相互依赖的对象提供了一个接口。根据给定的相关抽象类，Abstract Factory 模式提供了从一个相匹配的具体子类集创建这些抽象类的实例的方法。Abstract Factory 模式提供了一个可以确定合适的具体类的抽象类，这个抽象类可以用来创建实现标准接口的具体产品的集合。客户端只与产品接口和 Abstract Factory 类进行交互。使用

这种模式，客户端不用知道具体的构造类。Abstract Factory 模式类似于 Factory Method 模式，但是 Abstract Factory 模式可以创建一系列的相 关对象。

Builder 模式将复杂对象的构建与其表示相分离，这样相同的构造过程可以创建不同的对 象。通过只指定对象的类型和内容，Builder 模式允许客户端对象构建一个复杂对象。客户端 可以不受该对象构造的细节的影响。这样通过定义一个能够构建其他类实例的类，就可以简 化复杂对象的创建过程。Builder 模式生产一个主要产品，而该产品中可能有多个类，但是通 常只有一个主类。

Prototype 模式允许对象在不了解要创建对象的确切类以及如何创建等细节的情况下创 建自定义对象。使用 Prototype 实例，便指定了要创建的对象类型，而通过复制这个 Prototype， 就可以创建新的对象。Prototype 模式是通过先给出一个对象的 Prototype 对象，然后再初始 化对象的创建。创建初始化后的对象再通过 Prototype 对象对其自身进行复制来创建其他对 象。Prototype 模式使得动态创建对象更加简单，只要将对象类定义成能够复制自身就可以 实现。

参考答案(56) B (57) C (58) A

试题(59)~(63)某公司欲开发一个在线教育平台。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“网站在并发用户数量 10 万的负载情况下，用户请求的平均响应时间应小于 3 秒”这一场景主要与(59)质量属性相关，通常可采用(60)架构策略实现该属性；“主站宕机后，系统能够在 10 秒 内自动切换至备用站点并恢复正常运行”主要与(61)质量属性相关，通常可采用(62)架构策略实现该属 性；“系统完成上线后，少量的外围业务功能和界面的调整与修改不超过 10 人·月”主要与(63)质量属性相 关。

(59) A. 性能 B. 可用性 C. 易用性 D. 可修改性

(60) A. 抽象接口 B. 信息隐藏 C. 主动冗余 D. 资源调度

(61) A. 性能 B. 可用性 C. 易用性 D. 可修改性

(62) A. 记录/回放 B. 操作串行化 C. 心跳 D. 增加计算资源

(63) A. 性能 B. 可用性 C. 易用性 D. 可修改性

试题(59)~(63)分析本题考查质量属性的基础知识与应用。

架构的基本需求主要是在满足功能属性的前提下，关注软件质量属性，架构设计则是为 满足架构需求 (质量属性)寻找适当的“战术”(即架构策略)。

软件属性包括功能属性和质量属性，但是，软件架构(及软件架构设计师)重点关注的 是质量属性。因为在大量的可能结构中，可以使用不同的结构来实现同样的功能性，即功能 性在很大程度上是独立于结构的，架构设计师面临着决策(对结构的选择)，而功能性所关 心的是它如何与其他质量属性进行交互，以及它如何限制其他质量属性。

常见的 6 个质量属性为可用性、可修改性、性能、安全性、可测试性、易用性。质量属性场景是一种面向特定的质量属性的需求，由以下 6 部分组成：刺激源、刺激、环境、制品、响应、响应度量。

题目中描述的人员管理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“网站在并发用户数量 10 万的负载情况下，用户请求的平均响应时间应小于 3 秒”这一场景主要与性能质量属性相关，通常可采用提高计算效率、减少计算开销、控制资源使用、资源调度、负载均衡等架构策略实现该属性；“主站宕机后，系统能够在 10 秒内自动切换至备用站点并恢复正常运行”主要与可用性质量属性相关，通常可采用 Ping/Echo、心跳、异常检测、主动冗余、被动冗余、检查点等架构策略实现该属性；“系统完成上线后，少量的外围业务功能和界面的调整与修改不超过 10 人·月”主要与可修改性质量属性相关。

参考答案(59) A (60) D (61) B (62) C (63) D

试题(64) SYN Flooding 攻击的原理是(64)。

- A. 利用 TCP 三次握手，恶意造成大量 TCP 半连接，耗尽服务器资源，导致系统拒绝服务
- B. 操作系统在实现 TCP/IP 协议栈时，不能很好地处理 TCP 报文的序列号紊乱问题，导致系统崩溃
- C. 操作系统在实现 TCP/IP 协议栈时，不能很好地处理 IP 分片包的重叠情况，导致系统崩溃
- D. 操作系统协议栈在处理 IP 分片时，对于重组后超大的 IP 数据包不能很好地处理，导致缓存溢出而系统崩溃

试题(64)分析本题考查网络安全知识。

SYN Flooding 是一种常见的 DOS (denial of service, 拒绝服务) 和 DDoS (distributed denial of service, 分布式拒绝服务) 攻击方式。它使用 TCP 协议缺陷，发送大量的伪造的 TCP 连接请求，使得被攻击方 CPU 或内存资源耗尽，最终导致被攻击方无法提供正常的服务。

参考答案(64) A

试题(65)下面关于 Kerberos 认证的说法中，错误的是(65)。

- A. Kerberos 是在开放的网络中为用户提供身份认证的一种方式
- B. 系统中的用户要相互访问必须首先向 CA 申请票据
- C. KDC 中保存着所有用户的账号和密码
- D. Kerberos 使用时间戳来防止重放攻击

试题(65)分析本题目考查 Kerberos 认证系统的认证流程知识。

Kerberos 提供了一种单点登录(SSO)的方法。考虑这样一个场景，在一个网络中有不同的服务器，比如，打印服务器、邮件服务器和文件服务器。这些服务器都有认证的需求。很自然的，让每个服务器自己

实现一套认证系统是不合理的，而是提供一个中心认证服务器（AS-Authentication Server）供这些服务器使用。这样任何客户端就只需维护一个密码就能登录所有服务器。

因此，在 Kerberos 系统中至少有三个角色：认证服务器(AS)，客户端(Client)和普通服务器(Server)。客户端和服务器将在 AS 的帮助下完成相互认证。

在 Kerberos 系统中，客户端和服务器都有一个唯一的名字。同时，客户端和服务器都有自己的密码，并且它们的密码只有自己和认证服务器 AS 知道。

客户端在进行认证时，需首先向密钥分发中心来申请初始票据。

参考答案(65)B

试题(66)、(67)某软件公司根据客户需求，组织研发出一套应用软件，并与本公司的职工签订了保密协议，但是本公司某研发人员将该软件中的算法和部分程序代码公开发表。该软件研发人员(66)，该软件公司丧失了这套应用软件的(67)。

- (66)A. 与公司共同享有该软件的著作权，是正常行使发表权
- B. 与公司共同享有该软件的著作权，是正常行使信息网络传播权
- C. 不享有该软件的著作权，其行为涉嫌侵犯公司的专利权
- D. 不享有该软件的著作权，其行为涉嫌侵犯公司的软件著作权

(67) A. 计算机软件著作权 B. 发表权 C. 专利权 D. 商业秘密

试题(66)、(67)分析本题考查知识产权的基础知识。

根据题目描述，该软件公司的研发人员参与开发的该软件是职务作品，因此该软件著作权属于公司。软件著作权的客体是指计算机软件，即计算机程序及其有关文档。

软件著作权包括人身权、财产权等，人身权包括署名权、修改权、保护作品完整权等权力，财产权包括复制权、发行权、展览权、改编权、信息网络传播权等权利。发表权指决定软件是否公之于众的权利；发行权是指以出售或者赠与方式向公众提供软件的原件或者复制件的权利；信息网络传播权是指以有线或者无线方式向公众提供软件，使公众可以在其个人选定的时间和地点获得软件的权利。

研发人员将该软件中的算法和部分程序代码公开发表，使该公司丧失了商业秘密。

参考答案(66) D (67) D

试题(68)按照《中华人民共和国著作权法》的权利保护期，(68)受到永久保护。

(68)A. 发表权 B. 修改权 C. 复制权 D. 发行权

试题(68)分析本题考查知识产权的基础知识。

发表权指决定软件是否公之于众的权利；修改权是指对软件进行增补、删节，或者改变指令、语句顺序的权利；复制权是将软件制作一份或者多份的权利；发行权是指以出售或者赠与方式向公众提供软件的原件或者复制件的权利。

修改权属于软件著作权中的人身权，得到永久保护。

参考答案(68)B

试题(69)

为近似计算XYZ三维空间内由三个圆柱 $x^2+y^2 \leq 1, y^2+z^2 \leq 1, x^2+z^2 \leq 1$ 相交部分V的体积，以下四种方案中，(69)最容易理解，最容易编程实现。

- (69) A. 在 $z=0$ 平面中的圆 $x^2+y^2 \leq 1$ 上，近似计算二重积分
B. 画出V的形状，将其分解成多个简单形状，分别计算体积后，再求和
C. 将V看作多个区域的交集，利用有关并集、差集的体积计算交集体积
D. V位于某正立方体M内，利用M内均匀分布的随机点落在V中的比例进行计算

试题(69)分析

本题考查应用数学-随机模拟的基础知识。

由于三个圆柱相交部分很难画图，很难想象其形状，也很难确定其边界参数，因此，方案A、B、C的计算都有相当难度。方案D的计算非常容易，在计算机上利用伪随机数，很容易取得正立方体 $\{-1 \leq x, y, z \leq 1\}$ 内均匀分布的随机点，也很容易判断该点是否位于V内。对大量的随机点，很容易统计在该正立方体中的随机点位于V中的比例。该比例值的8倍就近似地等于V的体积。

参考答案(69)D

试题(70)某厂生产的某种电视机，销售价为每台2500元，去年的总销售量为25000台，固定成本总额为250万元，可变成本总额为4000万元，税率为16%，则该产品年销售量的盈亏平衡点为(70)台(只有在年销售量超过它时才能盈利)。

- (70)A. 5000 B. 10000 C. 15000 D. 20000

试题(70)分析

本题考查应用数学-管理经济学的基础知识。

可变成本总额与销售的电视机台数有关。去年销售了25000台，可变成本总额为4000万元，因此，每台电视机的可变成本为 $4000/25=1600$ 元。

如果年销售量为N台，则总成本=固定成本+N×每台的可变成本=250+0.16N(万元)。
总收益=0.25N(1-16%)=0.21N(万元)。

对于盈亏平衡点的年销售量N， $250+0.16N=0.21N$ ，所以 $N=5000$ (台)。

参考答案(70)A

试题(71) ~ (75)

The purpose of systems design is to specify a(n) (71) , which defines the technologies to be used to build the proposed information systems. This task is accomplished by analyzing the data models and process models that were initially created during (72) . The (73) is used to establish physical processes and data stores across a network. To complete this activity, the analyst may involve a number of system designers and (74) , which may be involved in this activity to help address business data, process, and location issues. The key inputs to this task are the facts, recommendations, and opinions that are solicited from various sources and the approved (75) from the decision analysis phase.

- | | |
|-------------------------------------|-------------------------------|
| (71) A. physical model | B. prototype system |
| C. database schema | D. application architecture |
| (72) A. requirements analysis | B. problem analysis |
| C. cause-effect analysis | D. decision analysis |
| (73) A. entity-relationship diagram | B. physical data flow diagram |
| C. data flow diagram | D. physical database model |
| (74) A. system users | B. system analyst |
| C. system owner | D. project manager |
| (75) A. system architecture | B. system proposal |
| C. technical model | D. business procedure |

参考译文: 系统设计的目的是确定一种应用体系架构, 该架构定义了用于构建所建议信息系统的技术。通过分析最初在需求分析期间创建的数据模型和过程模型来完成该项任务。物理数据流程图用于在整个网络上建立物理过程和数据存储。为了完成此活动, 分析人员可能需要许多系统设计人员和系统用户参与到该活动中, 帮助处理业务数据、流程和位置问题。该任务的关键输入是从各种来源获取的事实、建议和意见, 以及在决策分析阶段获批的系统建议。

参考答案(71)D (72) A (73) B (74) A (75) B

案例分析

试题一

阅读以下关于软件架构设计与评估的叙述, 在答题纸上回答问题 1 和问题 2。

【说明】

某公司拟开发一套在线软件开发系统，支持用户通过浏览器在线进行软件开发活动。该系统的主要功能包括代码编辑、语法高亮显示、代码编译、系统调试、代码仓库管理等。在需求分析与架构设计阶段，公司提出的需求和质量属性描述如下：

- (a) 根据用户的付费情况对用户进行分类，并根据类别提供相应的开发功能；
- (b) 在正常负载情况下，系统应在 0.2 秒内对用户的界面操作请求进行响应；
- (c) 系统应该具备完善的安全防护措施，能够对黑客的攻击行为进行检测与防御；
- (d) 系统主站点断电后，应在 3 秒内将请求重定向到备用站点；
- (e) 系统支持中文昵称，但用户名必须以字母开头，长度不少于 8 个字符；
- (f) 系统宕机后，需要在 15 秒内发现错误并启用备用系统；
- (g) 在正常负载情况下，用户的代码提交请求应该在 0.5 秒内完成；
- (h) 系统支持硬件设备灵活扩容，应保证在 2 人·天内完成所有的部署与测试工作；
- (i) 系统需要为针对代码仓库的所有操作情况进行详细记录，便于后期查阅与审计；
- (j) 更改系统的 Web 界面风格需要在 4 人·天内完成；
- (k) 系统本身需要提供远程调试接口，支持开发团队进行远程排错。

在对系统需求、质量属性和架构特性进行分析的基础上，该公司的系统架构师给出了两种候选的架构设计方案，公司目前正在组织相关专家对候选系统架构进行评估。

【问题 1】(13 分)

针对该系统的功能，李工建议采用管道-过滤器(pipe and filter)的架构风格，而王工则建议采用仓库(repository)架构风格。请指出该系统更适合采用哪种架构风格，并针对系统的主要功能，从数据处理方式、系统的可扩展性和处理性能三个方面对这两种架构风格进行比较与分析，填写表 1·1 中的(1)~(4)空白处。

表 1-1 两种架构风格的比较与分析

架构风格名称	数据处理方式	系统可扩展性	处理性能
管道-过滤器	数据驱动机制，处理流程事先确定，交互性差	(2)	劣势：需要数据格式转换，性能降低 优势：支持过滤器并发调用，性能提高
仓库	(1)	数据与处理解耦合，可动态添加和删除处理组件	劣势：(3) 优势：(4)

【问题 2】(12 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。请将合适的质量属性名称填入图 1-1 中(1)、(2)空白处，并选择题干描述的(a)~(k)填入(3)~(6)空白处，完成该系统的效用树。

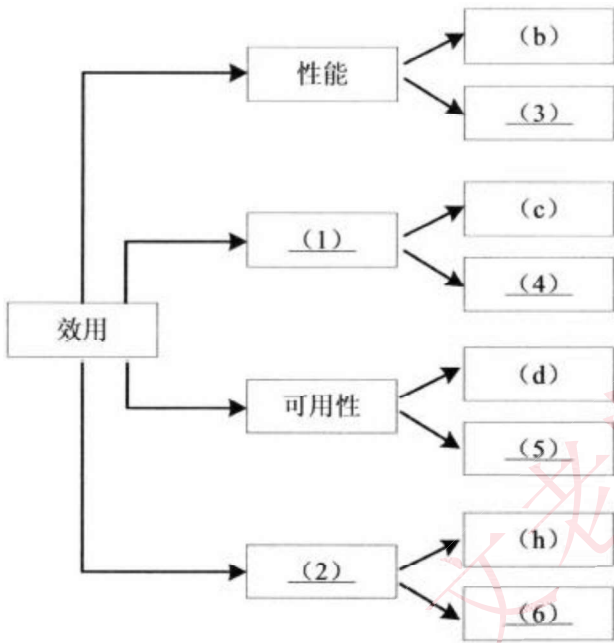


图 1-1 在线软件开发系统效用树

试题一分析

本题考查软件架构评估方面的知识与应用，主要包括质量属性效用树和架构分析两个部分。

此类题目要求考生认真阅读题目对系统需求的描述，经过分类、概括等方法，从中确定软件功能需求、软件质量属性、架构风险、架构敏感点、架构权衡点等内容，并采用效用树这一工具对架构进行评估。

【问题 1】

本问题考查考生对影响系统架构风格选型理解与掌握。根据系统要求，李工建议采用管道-过滤器(pipe and filter)的架构风格，而王工则建议采用仓库(repository)架构风格。考生需要从系统的主要功能和要求，从数据处理方式、系统的可扩展性和处理性能三个方面对这两种架构风格的优势和劣势进行比较与分析。具体如下表所示。

架构风格名称	数据处理方式	系统可扩展性	处理性能
管道-过滤器	数据驱动机制，处理流程事先确定，交互性差	数据与处理紧密关联，调整处理流程需要系统重新启动	劣势：需要数据格式转换，性能降低 优势：支持过滤器并发调用，性能提高
仓库	数据存储为中心仓库，处理流程独立，支持交互式处理	数据与处理解耦合，可动态添加和删除处理组件	劣势：数据与处理分离，需要加载数据，性能降低 优势：数据处理组件之间一般无依赖关系，可并发调用，提高性能

经过综合比较与分析，可以看出该系统更适合使用仓库风格。

【问题 2】

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。质量属性效用树主要关注性能、可用性、安全性和可修改性等四个用户最为关注的质量属性，考生需要对题干的需求进行分析，逐一找出这四个质量属性对应的描述，然后填入空白处即可。

经过对题干进行分析，可以看出：

- (a) 根据用户的付费情况对用户进行分类，并根据类别提供相应的开发功能(功能需求)；
- (b) 在正常负载情况下，系统应在 0.2 秒内对用户的界面操作请求进行响应(性能)；
- (c) 系统应该具备完善的安全防护措施，能够对黑客的攻击行为进行检测与防御(安全性)；
- (d) 系统主站点断电后，应在 3 秒内将请求重定向到备用站点(可用性)；
- (e) 系统支持中文昵称，但用户名必须以字母开头，长度不少于 8 个字符(功能需求)；
- (f) 系统宕机后，需要在 15 秒内发现错误并启用备用系统(可用性)；
- (g) 在正常负载情况下，用户的代码提交请求应该在 0.5 秒内完成(性能)；
- (h) 系统支持硬件设备灵活扩容，应保证在 2 人·天内完成所有的部署与测试工作(可修改性)；
- (i) 系统需要为针对代码仓库的所有操作情况进行详细记录，便于后期查阅与审计(安全性)；
- (j) 更改系统的 Web 界面风格需要在 4 人·天内完成(可修改性)；
- (k) 系统本身需要提供远程调试接口，支持开发团队进行远程排错(可测试性)。

参考答案

【问题 1】

该系统更适合采用仓库架构风格。


- (1) 数据存储在中心仓库，处理流程独立，支持交互式处理。
- (2) 数据与处理紧密关联，调整处理流程需要系统重新启动。
- (3) 数据与处理分离，需要加载数据，性能降低。
- (4) 数据处理组件之间一般无依赖关系，可并发调用，提高性能。

【问题 2】

- (1) 安全性
- (2) 可修改性
- (3) (g)
- (4) (i)
- (5) (f)
- (6) (j)

试题二

某企业委托软件公司开发一套包裹信息管理系统，以便于对该企业通过快递收发的包裹信息进行统一管理，在系统设计阶段，需要对不同快递信息的包裹单信息进行建模，其中，邮政包裹单如图 2-1 所示：

国内普通包裹详情单 (通知单联)		邮1106
		接收局号码:
 PA05837748444		收寄人名章:
收件人	内件品名及数量	
姓名: _____ 电话: _____	是否保价 是 <input type="checkbox"/> 否 <input type="checkbox"/>	重量: _____ 克
单位名称: _____		
详细地址: _____	寄件人声明: 同意并遵守背面的“使用须知”, 如包裹无法投递, 按如下选择处理: <input type="checkbox"/> 退还寄件人 <input type="checkbox"/> 抛弃处理	资费: _____ 元
姓名: _____ 电话: _____	签字: _____	挂号费: _____ 元
单位名称: _____	检查人员名章	保价费: _____ 元
详细地址: _____		回执费: _____ 元
用户代码: _____ 邮政编码: _____		总计: _____ 元

PA 0583-7748 4 44 填写本单前, 务请阅读背面的“使用须知”! 您的签名意味着您理解并接受“使用须知”内容。

图 2-1 包裹单示意图

[问题 1]

请说明关系型数据库开发中，逻辑数据模型设计过程包含哪些任务？根据图 2-1 包裹详情单应该涉及出哪些关系模式的名称，并指出每个关系模式的主键属性。

[问题 2]

请说明什么是超类实体？结合图中包裹单信息，试设计一种超类实体，给出完整的属性列表。

[问题 3]

请说明什么是派生属性？结合图中包裹单信息说明哪个属性是派生属性。

试题二分析

本题考查数据库设计与建模相关知识及应用。

数据库设计过程包括了逻辑数据建模和物理数据建模，逻辑数据建模阶段主要构造实体-联系图表达实体及其属性和实体间的联系，物理数据建模阶段主要根据所选数据库系统设计数据库模式。实体联系图 (Entity Relationship Diagram) 指以实体、联系、属性三个基本概念概括数据的基本结构，从而描述静态数据结构的概念模式。实体是具有公共性质的可相互区分的现实世界对象的集合，可以是具体的，也可以是抽象的概念或联系。属性是实体所具有的模拟特性，一个实体可由若干个属性来刻画。联系是数据对象彼此之间存在的相互关系。

此类题目要求考生认真阅读题目对问题的描述，准确理解数据库设计的主要任务和实体联系图中各个元素的含义，结合图中所给出的包裹单示意图中所描述的数据项，分析其关系，确定实体、属性和联系。

【问题 1】

在关系型数据库开发中，逻辑数据模型建设的主要任务是构建实体联系图。构建过程中，

首先通过上下文数据模型确定实体及其联系，为每个实体确定其标识性属性并添加完整属性，在此基础上利用规范化技术对所建立逻辑数据模型进行优化，一般需要满足第三范式 3NF 要求。对图 2-1 所示包裹单中所有数据项进行分析，主要涉及的实体包括收件人、寄件人及其之间的关联实体包裹单，其余数据项设计为上述三个实体的属性即可。

【问题 2】

数据库建模中可以对属性相似的实体进行进一步的抽象，通过将多个实体中相同的属性组合起来构造出新的抽象实体，即超类实体，原有多个实体称之为子类实体，通过两者之间的继承关系来表达抽象实体和具体实体的关系。图 2-1 中收件人和寄件人的属性都包括了姓名、电话、单位名称、详细地址和邮政编码等信息，可以设计出一个超类实体“用户”来实现通用属性的抽象表示。

【问题 3】

在数据库优化过程中，第三范式要求消除派生属性，即某个实体的非主键属性由该实体其他非主键属性决定，那么该属性可以称之为派生属性。图 2-1 所示属性中，包裹单的属性“费用总计”是由资费、挂号费、保价费、回执费等计算得出，所以是派生属性。

参考答案

【问题 1】

逻辑数据模型设计过程包含的任务：

- (1) 构建系统上下文数据模型，包含实体及实体之间的联系；
- (2) 绘制基于主键的数据模型，为每个实体添加主键属性；
- (3) 构建全属性数据模型，为每个实体添加非主键属性；
- (4) 利用规范化技术建立系统规范化数据模型。

包裹单的逻辑数据模型中包含的实体：

- (1) 收件人（主键：电话）；
- (2) 寄件人（主键：电话）；
- (3) 包裹单（主键：编号）。

【问题 2】

超类实体是将多个实体中相同的属性组合起来构造出的新实体。

用户（姓名、电话、单位名称、详细地址）

【问题 3】

派生属性是指某个实体的非主键属性由该实体其他非主键属性决定。

包裹单中的总计是由资费、挂号费、保价费、回执费计算得出，所以是派生属性。

试题三

阅读以下关于开放式嵌入式软件架构设计的相关描述，回答问题 1 至问题 3

【说明】

某公司一直从事宇航系统研制任务，随着宇航产品综合化、网络化技术发展的需要，公司的业务量急剧增加，研制新的软件架构已迫在眉睫。公司架构师王工广泛调研了多种现代架构的基础，建议采用基于 FACE（Future Airborne Capability Environment）的宇航系统开放式软件架构，以实现宇航系统的跨平台复用，实现宇航软件高质量、低成本的开发。公司领导肯定了王工的提案，并指出公司要全面实施基于 FACE 的开放式软件架构，应注意每个具体项目在实施中如何有效实现从需求到架构设计的关系，掌握基于软件需求的软件架构设计方法，并做好开放式软件架构中各段间的接口标准化设计工作。

【问题 1】（9 分）

王工指出，软件开发中需求分析是根本，架构设计是核心，不考虑软件需求便进行软件架构设计很可能导致架构设计的失败，因此，如何把软件需求映射到软件架构至关重要。请从描述语言、非功能性需求描述、需求和架构的一致性等方面，用 300 字以内的文字说明软件需求到架构的映射存在哪些难点。

【问题 2】（10 分）

图 3-1 是王工给出的 FACE 架构布局，包括操作系统、I/O 服务、平台服务、传输服务和可移植组件等 5 个段；操作系统、I/O 和传输等 3 个标准接口。请分析图 3-1 给出的 FACE 架构的相关信息，用 300 字以内的文字简要说明 FACE 5 个段的含义。

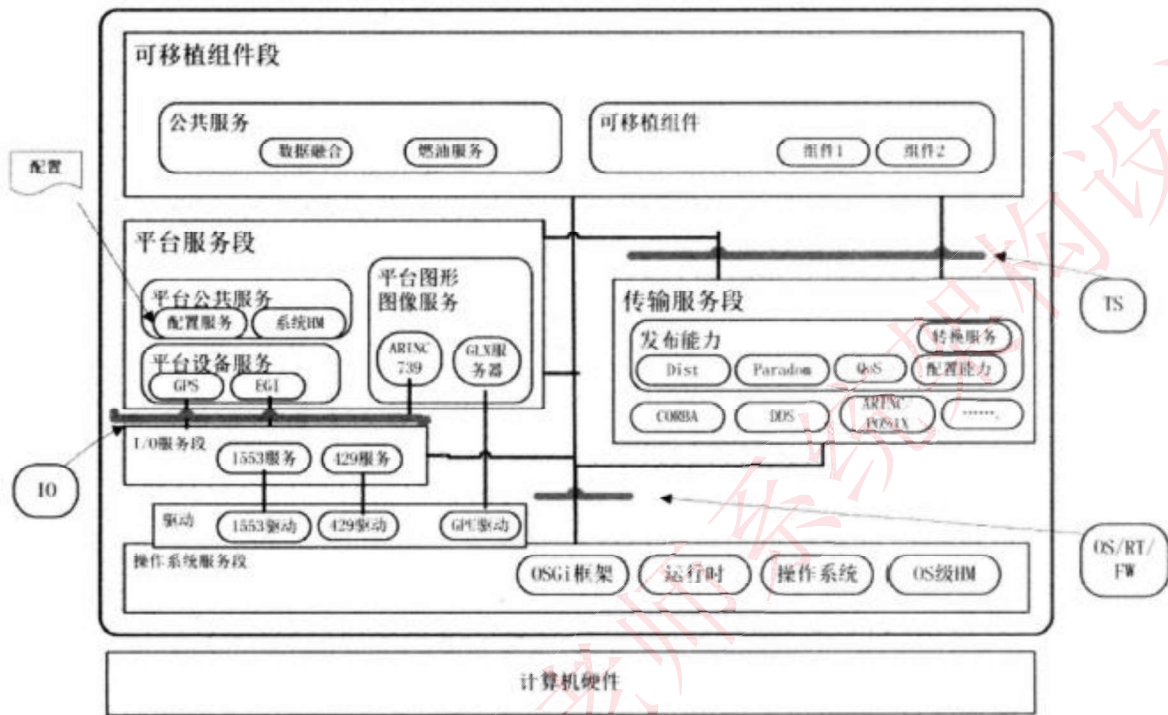


图 3-1 FACE 架构

【问题 3】(6 分)

FACE 架构的核心能力是可支持应用程序的跨平台执行和可移植性, 要达到可移植能力, 必须解决应用程序的紧耦合和封装的障碍。请用 200 字以内的文字简要说明在可移植性上, 应用程序的紧耦合和封装问题的主要表现分别是什么, 并给出解决方案。

试题三分析

FACE 是近年来宇航领域提出的一种面向服务的、安全可靠、可移植、可扩展的开放式嵌入式系统架构, 可实现宇航软件的跨平台复用以及高质量、低成本的开发工作。从图 3-1 可以看出, FACE 将宇航软件分为 5 个功能服务段, 各段之间通过标准的服务接口或传输服务实现功能间的相互调用。架构设计是软件系统开发中的重要环节, 其架构的优劣直接影响着软件系统的功能实现, 因此, 架构能否全面反映需求是架构设计的重中之重。

【问题 1】

通常在软件开发过程中, 需求会随着开发深入而有所变化, 而架构又不能完全地将需求全部反映出来, 因此, 如何把软件需求映射到软件架构是至关重要一个问题。在架构设计时, 架构设计师应密切关注需求到架构的映射存在以下 5 方面的难点:

- (1) 需求和架构描述语言存在差异: 软件需求是频繁获取的非正规的自然语言, 而软件架构常用某种正式语言。
- (2) 非功能属性难以在架构中描述: 系统属性中描述的非功能性需求通常很难在架构模型中形成规约。

(3) 需求和架构的一致性难以保障：从软件需求映射到软件架构的过程中，保持一致性和可追溯性很难，且复杂程度很高，因为单一的软件需求可能定位到多个软件架构的关注点。反之，架构元素也可能有多个软件需求。

(4) 用迭代和同步演化方法开发软件时，由于需求的不完整而带来的架构设计困难：架构设计必须基于一个准确的需求开展，而有些软件需求只能在建模后甚至是在架构实现时才能被准确理解。

(5) 难以确定和细化包含这些需求的架构相关信息：大规模系统必须满足数以千计的需求，会导致很难确定和细化包含这些需求的架构相关信息。

【问题 2】

从图 3-1 可知，FACE 架构由 5 个基本段组成，每段内又分为多个功能服务，从这些服务可以看出每段的基本能力。例如，操作系统段是 FACE 架构的基本功能，除基本操作系统外，还涵盖了运行库，操作系统的健康监控 (HM)，图中所给出的 OSGi 框架，实现功能组件“即插即用”能力。如果考生掌握了面向服务的架构风格，就不难给出各个段的具体含义。

(1) 操作系统服务段：为 FACE 架构其他段提供操作系统、运行时和操作系统级健康监控等服务。通过开放式 OSGi 框架为上层功能提供 OS 标准接口，并可实现上层组件的即插即用能力。本段是 FACE 架构的基本服务段。

(2) I/O 服务段：主要针对专用 I/O 设备进行抽象，屏蔽平台服务段软件与硬件设备的关系，形成种虚拟设备，这里隐含着对系统中的所有硬件 I/O 的虚拟化。由于图形服务软件和 GPU 处理器紧密相关，因此 I/O 服务段不对 GPU 驱动进行抽象。

(3) 平台服务段：主要是指平台/用户需要的共性服务软件，主要涵盖跨平台的系统管理、共享设备服务，以及健康管理等。如：系统级健康监控 (HM)、配置、日志和流媒体等服务。本段主要包括平台公共服务、平台设备服务和平台图像服务等三类。

(4) 传输服务段：通过使用传统跨平台中间件软件（如 CORBA、DDA 等），为平台上层可移植组件段提供平台性的数据交换服务，可移植组件将通过传输服务段提供的服务实现交换，禁止组件间直接调用。本段应具备 QoS 质量特征服务、配置能力服务以及分布式传输服务等。

(5) 可移植组件段：为用户软件段，提供了多组件使用能力和功能服务。主要包括公共服务和可移植组件两类。

【问题 3】

紧耦合和封装是软件模块化设计中最难以解决的两个问题，要使软件具备良好的可移植性、可复用性，就必须清楚其问题的表现形式。

紧耦合是应用程序移植的一个障碍，进一步说，就是计算平台的硬件设备和软件模块及其沟通之间的耦合代表了一个应用程序的可移植性方面的障碍。原因是便携性使得每个平台设备都有一个接口控制文件

(ICD), 描述了由硬件所支持的消息和协议, 应用程序对消息和协议的支持将紧密耦合于硬件。若要移植, 需要太多的工作来修改应用程序以支持不同的结构元素。

为了尽量减少支持新的硬件设备所需要的工作, 可采用分离原则, 通过隔离实现硬件特定信息和少数模块的代码, 来减少耦合性。

通常紧耦合问题主要表现在 I/O 问题、业务逻辑问题和表现问题。

传统的应用程序不可移植的另一个原因是这些应用程序被紧密耦合到一组固定的接口, 而这些数据的每个数据源或槽 (sinks) 都暴露出了设备的特殊接口, 这些特殊接口在每个平台中都是不同的。这样, 支持平台设备的接口控制文件 (ICD) 是被硬编码到应用程序中, 就导致应用程序不能成功在不同计算平台上执行。

为了解决这种接口控制文件 (ICD) 被硬编码而难以封装的问题, 可以通过提供数据源或槽的软件服务的方法, 从紧耦合组件分解出应用程序, 并将平台相关部分加入计算环境中, 在计算平台内提供数据源或槽的软件服务, 并实现接口标准化。

通常封装问题主要表现在: ICD 硬编码问题、组件的紧耦合问题、直接调用问题。

参考答案

【问题 1】

(1) 需求和架构描述语言存在差异: 软件需求是频繁获取的非正式的自然语言, 而软件架构常用的是一种正式语言。

(2) 非功能属性难于在架构中描述: 系统属性中描述的非功能性需求通常很难在架构模型中形成规约。

(3) 需求和架构的一致性难以保障: 从软件需求映射到软件架构的过程中, 保持一致性和可追溯性很难, 且复杂程度很高, 因为单一的软件需求可能定位到多个软件架构的关注点。反之, 架构元素也可能有多个软件需求。

【问题 2】

操作系统服务段: 为 FACE 架构其他段提供操作系统、运行时和操作系统级健康监控等服务。通过开放式 OSGi 框架为上层功能提供 OS 标准接口, 并可实现上层组件的即插即用能力。

I/O 服务段: 主要针对专用 I/O 设备进行抽象, 屏蔽平台服务段软件与硬件设备的关系。由于图形服务软件和 GPU 处理器紧密相关, 因此 I/O 服务段不对 GPU 驱动进行抽象。

平台服务段: 主要是指用户需要的共性软件, 如: 系统级健康监控 (HM)、配置、日志和流媒体等服务。本段可包括平台公共服务、平台设备服务和平台图像服务等三类。

传输服务段: 主要为上层可移植组件段提供平台性的数据交换服务。可移植组件将通过传输服务段提供的服务实现交换, 禁止组件间直接调用。

可移植组件段: 提供了多组件使用能力和功能服务。主要包括公共服务和可移植组件两类。

【问题 3】

紧耦合问题主要表现在：I/O 问题、业务逻辑问题和表现问题。

解决方案：可采用分离原则，通过隔离实现硬件特定信息和少数模块的代码，减少耦合性。

封装问题主要表现在：ICD 硬编码问题、组件的紧耦合问题、直接调用问题。

解决方案：可以通过提供数据源或槽的软件服务的方法，将紧耦合组件分解出应用程序，并将平台相关部分加入计算环境中，在计算平台内提供数据源或槽的软件服务，并实现接口标准化。

试题四

阅读以下关于数据库缓存的叙述，在答题纸上回答问题 1 至问题 3。

【说明】

某互联网文化发展公司因业务发展，需要建立网上社区平台，为用户提供一个对网络文化产品(如互联网小说、电影、漫画等)进行评论、交流的平台。该平台的部分功能如下：

- (a) 用户帖子的评论计数器；
- (b) 支持粉丝列表功能；
- (c) 支持标签管理；
- (d) 支持共同好友功能等；
- (e) 提供排名功能，如当天最热前 10 名帖子排名、热搜榜前 5 排名等；
- (f) 用户信息的结构化存储；
- (g) 提供好友信息的发布/订阅功能。

该系统在性能上需要考虑高性能、高并发，以支持大量用户的同时访问。开发团队经过综合考虑，在数据管理上决定采用 Redis+数据库(缓存+数据库)的解决方案。

【问题 1】(10 分)

Redis 支持丰富的数据类型，并能够提供一些常见功能需求的解决方案。请选择题干描述的(a)~(g)功能选项，填入表 4-1 中(1)~(5)的空白处。

表 4-1 Redis 数据类型与业务功能对照表

数据类型	存储的值	可实现的业务功能
STRING	字符串、整数或浮点数	(1)
LIST	列表	(2)
SET	无序集合	(3)
HASH	包括键值对的无序散列表	(4)
ZSET	有序集合	(5)

【问题 2】(7 分)

该网上社区平台需要为用户提供 7*24 小时的不间断服务。同时在系统出现宕机等故障时，能在最短时间内通过重启等方式重新建立服务。为此，开发团队选择了 Redis 持久化支持。Redis 有两种持久化方式，分别是 RDB(Redis DataBase)持久化方式和 AOF(Append Only File)持久化方式。开发团队最终选择了 RDB 方式。

请用 200 字以内的文字，从磁盘更新频率、数据安全、数据一致性、重启性能和数据文件大小五个方面比较两种方式，并简要说明开发团队选择 RDB 的原因。

【问题 3】(8 分)

缓存中存储当前的热点数据，Redis 为每个 KEY 值都设置了过期时间，以提高缓存命中率。为了清除非热点数据，Redis 选择“定期删除+惰性删除”策略。如果该策略失效，Redis 内存使用率会越来越高，一般应采用内存淘汰机制来解决。

请用 100 字以内的文字简要描述该策略的失效场景，并给出三种内存淘汰机制。

试题四分析

本题考查数据库缓存的基本概念和具体应用。

【问题 1】

本问题考查 Redis 数据库缓存产品基本数据类型的常见应用。

- (1) STRING 类型：常规的 key/value 缓存应用，常规计数如粉丝数等；
- (2) LIST 类型：各类列表应用，如关注列表、好友列表、订阅列表等；
- (3) SET 类型：与 LIST 类似，但提供去重操作，也提供集合操作，可实现共同关注、共同喜好、共同好友等功能；
- (4) HASH 类型：存储部分变更数据，如用户数据等；
- (5) ZSET 类型：类似 SET 但提供自动排序，也可实现带权重的队列，如各类排行榜等。

【问题 2】

本问题考查 Redis 持久化存储的基本概念及应用。

Redis 提供了两种持久化存储的机制，分别是 RDB (Redis DataBase)持久化方式和 AOF (Append Only File)持久化方式。RDB 持久化方式是指在指定的时间间隔内将内存中的数据 集快照写入磁盘，是 Redis 默认的持久化方式。AOF 方式是指 redis 会将每一个收到的写命令都通过 write 函数追加到日志文件中。

两种方式各有优缺点，大致的比较如下：

- (1) 磁盘更新频率：AOF 比 RDB 文件更新频率高。
- (2) 数据安全：AOF 比 RDB 更安全。
- (3) 数据一致性：RDB 间隔一段时间存储，可能发生数据丢失和不一致；AOF 通过 append 模式写文件，即使发生服务器宕机，也可通过 redis-check-aof 工具解决数据一致性问题。
- (4) 重启性能：RDB 性能比 AOF 好。

(5)数据文件大小：AOF 文件比 RDB 文件大。

该项目的实际需求是：在系统出现宕机等故障时，需要在最短时间内通过重启等方式重新建立服务，因此重启性能是最需要考虑的因素，故该开发团队选择 RDB 方式。

【问题 3】

本问题考查 Rcdis 使用过程中数据清除相关的概念。

缓存中 般用来存储当前的热点数据，Redis 为每个 KEY 值都设置了过期时间，以提高缓存命中率。为了清除非热点数据，Redis 选择“定期删除+惰性删除”策略。

“定期删除+惰性删除”策略也会存在失效的可能。比如，如果“定期删除”没删除 KEY，也没即时去请求 KEY，也就是说“惰性删除”也没生效。这样，Redis 默认的“定期删除+ 惰性删除”策略就失效了。

如果该策略失效，Redis 内存使用率会越来越高，一般应采用内存淘汰机制来解决。常见的内存淘汰机制有：

- (1)从已设置过期时间的数据集最近最少使用的数据淘汰。
- (2)从已设置过期时间的数据集将要过期的数据淘。
- (3)从已设置过期时间的数据集任意选择数据淘汰。
- (4)从数据集最近最少使用的数据淘汰。
- (5)从数据集任意选择数据淘汰。

参考答案

【问题 1】

- (1) (a)
- (2) (b)、(g)
- (3) (c)、(d)
- (4) (f)
- (5) (e)

【问题 2】

磁盘更新频率：AOF 比 RDB 文件更新频率高。

数据安全：AOF 比 RDB 更安全。

数据一致性：RDB 间隔一段时间存储，可能发生数据丢失和不一致；AOF 通过 append 模式写文件，即使发生服务器宕机，也可通过 redis-check-aof 工具解决数据一致性问题。

重启性能：RDB 性能比 AOF 好。

数据文件大小：AOF 文件比 RDB 文件大。

综合上述五个方面的比较，考虑在系统出现宕机等故障时，需要在最短时间内通过重启等方式重新建立服务，因此开发团队最终选择了 RDB 方式。

【问题 3】

失效场景：如果“定期删除”没删除 KEY，也没即时去请求 KEY，也就是说“惰性删除”也没生效。这样，Redis 默认的“定期删除+惰性删除”策略就失效了。

对此，可采用内存淘汰机制解决：

- (1) 从已设置过期时间的数据集最近最少使用的数据淘汰。
- (2) 从已设置过期时间的数据集将要过期的数据淘汰。
- (3) 从已设置过期时间的数据集任意选择数据淘汰。
- (4) 从数据集最近最少使用的数据淘汰。
- (5) 从数据集任意选择数据淘汰。

试题五

阅读以下关于 Web 系统架构设计的叙述，在答题纸上回答问题 1 至问题 3。

【说明】

某公司拟开发一款基于 Web 的工业设备监测系统，以实现多种工业设备数据的分类采集、运行状态监测以及相关信息的管理。该系统应具备以下功能：

现场设备状态采集功能：根据数据类型对设备监测指标状态信号进行分类采集；

设备采集数据传输功能：利用可靠的传输技术，实现将设备数据从制造现场传输到系统后台；

设备监测显示功能：对设备的运行状态、工作状态以及报警状态进行监测并提供相应的图形化显示界面；

设备信息管理功能：支持设备运行历史状态、报警记录、参数信息的查询。

同时，该系统还需满足以下非功能性需求：

- (a) 系统应支持大于 100 个工业设备的并行监测；
- (b) 设备数据从制造现场传输到系统后台的传输时间小于 1s；
- (c) 系统应 7*24 小时工作；
- (d) 可抵御常见 XSS 攻击；
- (e) 系统在故障情况下，应在 0.5 小时内恢复；
- (f) 支持数据审计。

面对系统需求，公司召开项目组讨论会议，制定系统设计方案，最终决定采用三层拓扑结构，即现场设备数据采集层、Web 监测服务层和前端 Web 显示层。

【问题 1】(6 分)

请按照性能、安全性和可用性等三类非功能性需求分类，选择题干描述的(a) ~ (f) 填入(1) ~ (3)

表 5-1 非功能性需求归类表

非功能性需求类别	非功能性需求
性能	(1)
安全性	(2)
可用性	(3)

【问题 2】(14 分)

该系统的 Web 监测服务层拟采用 SSM (spring+spring MVC+Mybatis) 框架进行系统研发。SSM 框架的工作流程图如图 5.1 所示, 请从下面给出的 (a) ~ (k) 中进行选择, 补充完善图 5-1 中 (1) ~ (7) 处空白的内容。

- (a) Connection Pool
- (b) Struts2
- (c) Persistent Layer
- (d) Mybatis
- (e) HTTP
- (f) MVC
- (g) Kafka
- (h) View Layer
- (i) JSP
- (j) Controller Layer
- (k) Spring

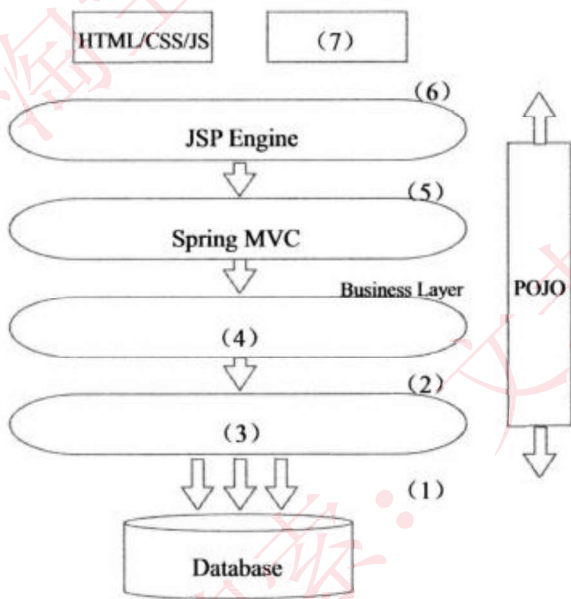


图 5-1 SSM 框架工作流程图

【问题 3】(5 分)

该工业设备检测系统拟采用工业控制领域中统一的数据访问机制，实现与多种不同设备的数据交互，请用 200 字以内的文字说明采用标准的数据访问机制的原因。

试题五分析

本题考查 Web 系统架构设计相关知识及如何在实际问题中综合应用。

此类题目要求考生认真阅读题目对现实系统需求的描述，结合 Web 系统设计相关知识、实现技术等完成 Web 系统分析设计。

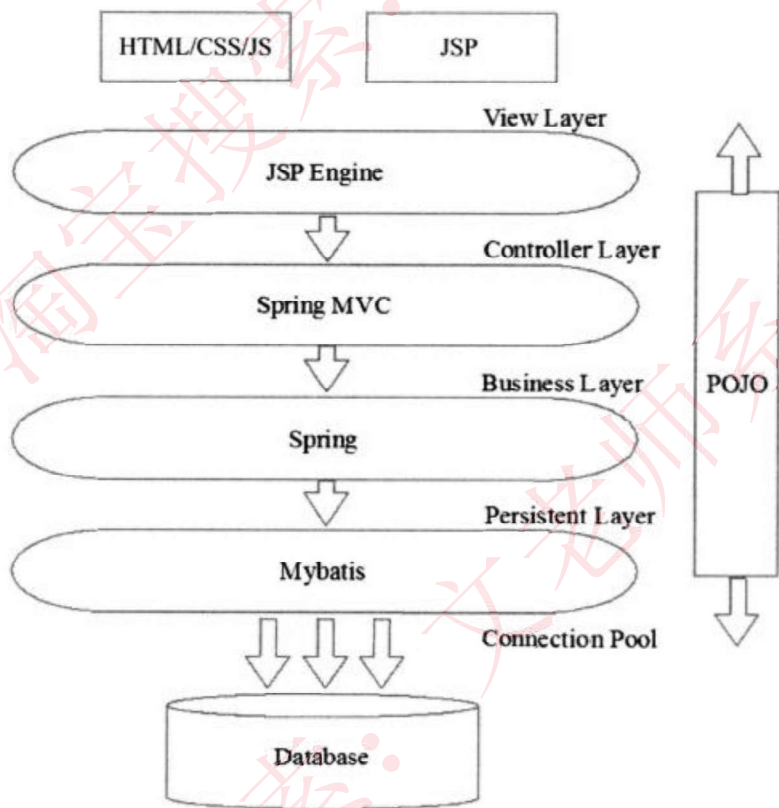
【问题 1】

软件质量属性有可用性、可修改性、性能、安全性、可测试性、易用性等。可用性关注的是系统产生故障的可能性和从故障中恢复的能力；性能关注的是系统对事件的响应时间；安全性关注的是系统保护合法用户正常使用系统、阻止非法用户攻击系统的能力；可测试性关注的是系统发现错误的能力；易用性关注的是对用户来说完成某个期望任务的容易程度和系统所提供的用户支持的种类。

【问题 2】

SSM 框架是 spring MVC，spring 和 Mybatis 框架的整合，是标准的 MVC 模式。其使用 spring MVC 负责请求的转发和视图管理；spring 实现业务对象管理；Mybatis 作为数据对象的持久化引擎。

因此，基于 SSM 的工业设备监测系统设计架构如下图所示。



【问题 3】

标准的数据访问机制可以在硬件供应商和软件开发商之间建立一套完整的规则。只要遵循这套规则，数据交互对两者来说都是透明的，硬件供应商只需考虑应用程序的多种需求和传输协议，软件开发商也不必了解硬件的实质和操作过程，实现对设备数据采集的统一管理。

例如，OPC (OLE for Process Control)即用于过程控制的OLE,是一个工业标准。OPC 是为了不同供应厂商的设备和应用程序之间的软件接口标准化，使其间的数据交换更加简单化的目的而提出的。作为结果，可以向用户提供不依赖于特定开发语言和开发环境且可以自由组合使用的过程控制软件组件产品。利用 OPC 的系统，是由按照应用程序(客户程序)的要求提供数据采集服务的 OPC 服务器，使用 OPC 服务器所必需的 OPC 接口，以及接受服务的 OPC 应用程序所构成。OPC 服务器是按照各个供应厂商的硬件所开发的，使之可以吸收各个供应厂商硬件和系统的差异，从而实现不依存于硬件的系统构成。同时利用一种叫作 Variant 的数据类型，可以不依存于硬件中固有数据类型。

参考答案

【问题 1】

- (1) (a) (b)
- (2) (d) (f)
- (3) (c) (e)

【问题 2】

- (1) (a)
- (2) (c)
- (3) (d)
- (4) (k)
- (5) (j)
- (6) (h)
- (7) (i)

【问题 3】

该工业设备检测系统需与不同设备进行数据交互，采用标准的数据访问机制可以在硬件供应商和软件开发商之间建立一套完整的规则。只要遵循这套规则，数据交互对两者来说都是透明的，硬件供应商只需考虑应用程序的多种需求和传输协议，软件开发商也不必了解硬件的实质和操作过程，实现对设备数据采集的统一管理。

论文写作

试题一 论企业集成架构设计及应用

企业集成架构(Enterprise Integration Architecture, EIA)是企业集成平台的核心,也是解决企业信息孤岛问题的关键。企业集成架构设计包括了企业信息、业务过程、应用系统集成架构的设计。实现企业集成的技术多种多样,早期的集成方式是通过在不同的应用之间开发一对一的专用接口来实现应用之间的数据集成,即采用点到点的集成方式;后来提出了利用集成平台的方式来实现企业集成,可以将分散的信息系统通过一个统一的接口,以可管理、可重复的方式实现单点集成。企业集成架构设计技术方案按照要解决的问题类型可以分为数据集成、应用集成和企业集成。

请围绕“论企业集成架构设计及应用”论题,依次从以下三个方面进行论述:

1. 概要叙述你参与的软件开发项目以及承担的主要工作。
2. 详细说明三类企业集成架构设计技术分别要解决的问题及其含义,并阐述每种技术具体包含了哪些集成模式。
3. 根据你所参与的项目,说明采用了哪些企业集成架构设计技术,其实施效果如何。

写作要点

- 一、简要描述所参与的软件系统开发项目,并明确指出在其中承担的主要任务和开展的主要工作。
- 二、详细说明三类企业集成架构设计技术分别要解决的问题及其含义,并阐述每种技术具体包含了哪些集成模式。

1. 数据集成。

数据集成是为了解决不同应用和系统间的数据共享和交换需求,具体包括共享信息管理、共享模型管理和数据操作管理三个部分。共享信息管理通过定义统一的集成服务模型和共享信息访问机制,完成对集成平台运行过程中产生数据信息的共享、分发和存储管理;共享模型管理则提供数据资源配置管理、集成资源关系管理、资源运行生命周期管理及相应的业务数据协同监控管理等功能;数据操作管理则为集成平台用户提供数据操作服务,包括多通道的异构模型之间的数据转换、数据映射、数据传递和数据操作等功能服务。

数据集成的模式包括:数据联邦、数据复制模式、基于结构的数据集成模式。

2. 应用集成。

应用集成是指两个或多个应用系统根据业务逻辑的需要而进行的功能之间的互相调用

和互操作。应用集成需要在数据集成的基础上完成。应用集成在底层的网络集成和数据集成的基础上实现异构应用系统之间应用层次上的互操作。它们共同构成了实现企业集成化运行最顶层集成所需要的技术层次上的支持。

应用集成的模式包括：集成适配器模式、集成信使模式、集成面板模式和集成代理模式。

3. 企业集成。

企业应用软件系统从功能逻辑上可以分为表示、业务逻辑和数据三个层次。其中表示层负责完成系统与用户交互的接口定义；业务逻辑层主要根据具体业务规则完成相应业务数据的处理；数据层负责存储由业务逻辑层处理所产生的业务数据，它是系统中相对稳定的一部分。支持企业间应用集成和交互的集成平台通常采用多层结构，其目的是在最大程度上提高系统的柔性。在集成平台的具体设计开发中，还需要按照功能的通用程度对系统实现模块进行分层。

企业集成的模式包括：前端集成模式、后端集成模式和混合集成模式。

三、针对考生本人所参与的项目中使用的企业集成架构设计技术，说明实施过程和具体实施效果。

试题二 论软件测试中缺陷管理及其应用

软件缺陷指的是计算机软件或程序中存在的某种破坏正常运行能力的问题、错误，或者隐藏的功能缺陷。缺陷的存在会导致软件产品在某种程度上不能满足用户的需要。在目前的软件开发过程中，缺陷是不可避免的。软件测试是发现缺陷的主要手段，其核心目标就是尽可能多地找出软件代码中存在的缺陷，进而保证软件质量。软件缺陷管理是软件质量管理的一个重要组成部分。

请围绕“论软件测试中缺陷管理及其应用”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承担的主要工作。
2. 详细论述常见的缺陷种类和级别，论述缺陷管理的基本流程。
3. 结合你具体参与管理和开发的实际项目，说明是如何进行缺陷管理的，请说明具体实施过程以及应用效果。

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、根据 IEEE 标准，软件测试中所发现的缺陷主要包括：输入/输出错误；逻辑错误；计算错误；接口错误；数据错误等；从软件测试角度还可以将缺陷分为五类：功能缺陷；系统缺陷；加工缺陷；数据缺陷；代码缺陷。不同企业的缺陷分类往往不同。

根据缺陷后果的严重程度，可以将缺陷分为多个不同的级别，例如 Beizer 将缺陷分为十级：轻微、中等、使人不悦、影响使用、严重、非常严重、极为严重、无法容忍、灾难性、传染性等。

缺陷管理是对软件测试环节中缺陷状态的完整跟踪和管理，确保每个被发现的缺陷都得到妥善处理。缺陷管理的目的是对各个阶段测试发现的缺陷进行跟踪管理，以保证各级缺陷的修复率达到标准，主要实现以下目标：保证信息的一致性；保证缺陷得到有效的跟踪；缩短沟通时间，解决问题更高效；收集缺陷数据并进行数据分析，作为缺陷度量的依据。

缺陷管理基本的流程如下：

- (1) 缺陷提交：测试人员发现缺陷后提交缺陷报告。
- (2) 缺陷审查：确定缺陷问题、种类和级别。
- (3) 修复流程：缺陷审查通过后进入修复流程，缺陷报告会转发给相应的软件开发人员进行修复。
- (4) 验证流程：开发人员提交修复后的代码，进入验证流程。通过回归测试等方法验证缺陷问题已经修复。
- (5) 缺陷关闭：在确认缺陷已完全解决后，关闭该缺陷。

部分缺陷管理流程中，还包括对缺陷状态的跟踪。

三、考生需结合自身参与项目的实际状况，指出其参与管理和开发的项目中所进行的缺陷管理活动，说明缺陷管理的具体实施过程，并对实际应用效果进行分析。

试题三 论云原生架构及其应用

近年来，随着数字化转型不断深入，科技创新与业务发展不断融合，各行各业正在从大工业时代的固化范式进化成面向创新型组织与灵活型业务的崭新模式。在这一背景下，以容器和微服务架构为代表的云原生技术作为云计算服务的新模式，已经逐渐成为企业持续发展的主流选择。云原生架构是基于云原生技术的一组架构原则和设计模式的集合，旨在将云应用中的非业务代码部分进行最大化剥离，从而让云设施接管应用中原有的大量非功能特性（如弹性、韧性、安全、可观测性、灰度等），使业务不再有非功能性业务中断困扰的同时，具备轻量、敏捷、高度自动化的特点。云原生架构有利于各组织在公有云、私有云和混合云等新型动态环境中，构建和运行可弹性扩展的应用，其代表技术包括容器、服务网格、微服务、不可变基础设施和声明式 API 等。

请围绕“论云原生架构及其应用”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承担的主要工作。
2. 服务化、弹性、可观测、韧性和自动化是云原生架构重要的设计原则。请简要对这些设计原则的内涵进行阐述。
3. 具体阐述你参与管理和开发的项目是如何采用云原生架构的，并围绕上述四类设计原则，详细论述在项目设计与实现过程中遇到了哪些实际问题，是如何解决的。

写作要点

一、简要叙述所参与管理和开发的软件项目，需要明确指出在其中承担的主要任务和开展的主要工作。

二、云原生架构的设计原则具体描述如下：

(1) 服务化原则。当代码规模超出小团队的合作范围时，就有必要进行服务化拆分，包括拆分为微服务架构、小服务 (mini service) 架构，通过服务化架构把不同生命周期的模块分离出来，分别进行业务迭代，避免迭代频繁模块被慢速模块拖慢，从而加快整体的进度和稳定性。同时服务化架构以面向接口编程，服务内部的功能高度内聚，模块间通过公共功能模块的提取增加软件的复用程度。

(2) 弹性原则。大部分系统部署上线需要根据业务量的估算，准备一定规模的机器，传统上线过程中需要经历采购申请、供应商洽谈、机器部署上电、软件部署、性能压测等阶段，周期很长，重新调整也非常困难。针对这种情况，弹性原则是指系统的部署规模可以随着业务量的变化自动伸缩，无须根据事先的容量规划准备固定的硬件和软件资源，从而提高资源利用率，降低成本。

(3) 可观测原则。可观测性原则是指主动通过日志、链路跟踪和度量等手段，每次业务请求背后的多次服务调用的耗时、返回值和参数都清晰可见，甚至可以下钻到三方软件调用、SQL 请求、节点拓扑、网络响应等。具备可观测能力可以使运维、开发和业务人员实时掌握软件运行情况，并结合多个维度的数据指标，获得前所未有的关联分析能力，不断对业务健康度和用户体验进行数字化衡量和持续优化。

(4) 韧性原则。韧性原则是指当软件所依赖的软硬件组件出现各种异常时，软件需要表现出抵御能力，这些异常通常包括硬件故障、硬件资源瓶颈、业务流量超出软件设计能力、故障和灾难、软件 bug、黑客攻击等对业务可用性带来致命影响的因素。韧性从多个维度诠释了软件持续提供业务服务的能力，核心目标是提升软件的 MTBF (Mean Time Between Failure, 平均无故障时间)。

(5) 自动化原则。自动化原则是指通过多种技术手段和自动化交付工具，一方面标准化企业内部的软件交付过程，另一方面在标准化的基础上进行自动化，通过配置数据自描述和面向终态的交付过程，让自动化工具理解交付目标和环境差异，实现整个软件交付和运维的自动化。

三、论文中需要结合项目实际工作，详细论述在项目中是如何采用云原生架构进行系统的设计与实现的，并围绕云原生架构的设计原则，论述遇到了哪些实际问题，是采用何种方法解决的。

试题四 论数据分片技术及其应用

数据分片就是按照一定的规则，将数据集划分成相互独立、正交的数据子集，然后将数据子集分布到不同的节点上。通过设计合理的数据分片规则，可将系统中的数据分布在不同的物理数据库中，达到提升应用系统数据处理速度的目的。

请围绕“论数据分片技术及其应用”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承担的主要工作。

2. Hash 分片、一致性 Hash (Consistent Hash) 分片和按照数据范围 (Range Based) 分片是三种常用的数据分片方式。请简要阐述三种分片方式的原理。

3. 具体阐述你参与管理和开发的项目采用了哪些分片方式，并具体说明其实现过程和应用效果。

写作要点

一、简要叙述所参与管理和开发的软件项目，需要明确指出在其中承担的主要任务和开展的主要工作。

二、三种分片方式的具体描述如下：

1. Hash 方式。

数据分片的 Hash 方式是基于哈希表的思想，即按照数据的某一特征 (key) 来计算哈希值，并将哈希值与系统中的节点建立映射关系，从而将哈希值不同的数据分布到不同的节点上。

按照 Hash 方式做数据分片，优点是映射关系非常简单，需要管理的元数据也非常之少，只需要记录节点的数目以及 Hash 方式即可。但是，Hash 方式的缺点也非常明显。首先，当加入或者删除一个节点的时候，大量的数据需要移动。其次，Hash 方式很难解决数据不均衡的问题，如原始数据的特征值分布不均匀，导致大量的数据集中到一个物理节点上；或者对于可修改的记录数据，单条记录的数据变大。

2. 一致性 Hash 方式。

一致性 hash 是将数据按照特征值映射到一个首尾相接的 Hash 环上，同时也将节点 (按照 IP 地址或者机器名 Hash) 映射到这个环上。对于数据，从数据在环上的位置开始，顺时针找到的第一个节点即为数据的存储节点。可以看到相比于 Hash 方式，一致性 Hash 方式需要维护的元数据额外包含了节点在环上的位置，但这个数据量是较小的。同时，一致性 Hash 在增加或者删除节点的时候，受到影响的数据是比较有限的，只会影响到 Hash 环上相应的节点，不会发生大规模的数据迁移。

3. 按照数据范围方式。

按照数据范围 (Range Based) 方式是按照关键值划分成不同的区间，每个物理节点负责一个或者多个区间。按照数据范围方式跟一致性 Hash 有相似之处，可以理解为物理节点在 Hash 环上的位置是动态变化的。在按照数据范围方式中，区间的大小不是固定的，每个数据区间的数量与区间的大小也是没有关系的。比如说，一部分数据非常集中，那么区间大小应该会比较小的，即以数据量的大小为片段标准。在实际工程中，一个节点往往负责多个区间，每个区间成为一个块，每个块有一个阈值，当达到这个阈值之后就会分裂成两个块。这样做的目的在于当有节点加入的时候快速达到均衡。

三、论文中需要结合项目实际工作，详细论述项目采用了哪些分片方式，并具体说明其实现过程和应用效果。