

全国计算机技术与软件专业技术资格（水平）考试用书

系统架构设计师考试

考点突破、案例分析、试题实战
一本通

希赛教育软考学院 王勇 主编

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书由希赛教育软考学院组织编写,作为计算机技术与软件专业技术资格(水平)考试中的系统架构设计师级别的考试辅导指定教材。内容紧扣考试大纲,通过对历年试题进行科学分析、研究、总结、提炼而成。每章内容分为考点突破、典型试题分析、实战练习题、练习题解析四个部分。

基于历年试题,利用统计分析的方法,科学得出结论并预测以后的出题动向,是本书的一大特色。本书可以保证既不漏掉考试必需的知识点,又不加重考生备考负担,使考生轻松、愉快地掌握知识点并领悟系统架构设计师考试的真谛。

本书适合参加计算机技术与软件专业技术资格(水平)考试中的系统架构设计师级别的考生参考学习,也可作为相关培训班的教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

系统架构设计师考试考点突破、案例分析、试题实战一本通 / 王勇主编. —北京:电子工业出版社, 2012.9

全国计算机技术与软件专业技术资格(水平)考试用书

ISBN 978-7-121-17632-6

I. ①系… II. ①王… III. ①计算机系统—工程技术人员—资格考核—自学参考资料 IV. ①TP30

中国版本图书馆 CIP 数据核字(2012)第 158888 号

策划编辑:孙学瑛

责任编辑:葛娜

特约编辑:赵树刚

印刷:北京东光印刷厂

装订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开本:787×1092 1/16 印张:33.75 字数:864 千字

印次:2012 年 9 月第 1 次印刷

印数:4000 册 定价:69.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

全国计算机技术与软件专业技术资格（水平）考试（俗称“软考”）由人力资源和社会保障部、工业和信息化部主办，面向社会，用于考查计算机专业人员的水平与能力。考试客观、公正，得到了社会的广泛认可，并实现了中、日、韩三国互认。

本书紧扣考试大纲，采用了表格统计法、走势图分析法，科学地研究每个知识点的命题情况，准确把握每个出题点的深浅。同时基于每个章节知识点分布统计分析的结果，科学地编写实战练习题，完全紧扣大纲，结构科学、重点突出、针对性强。

内容超值，针对性强

本书每章的内容分为考点突破、典型试题分析、实战练习题、练习题解析四个部分。

第一部分为考点突破。考点突破部分首先对历年试题进行了统计分析，采用图表形式，形象而直观，使各考点“暴露无遗”。通过学习本部分内容，考生可以对考试的知识点分布、考试重点有一个整体上的认识和把握，然后重点对大纲中所规定的重要考试内容和考试必备的知识点进行了“画龙点睛”，章节中的知识点解析深浅程度根据该知识点在历年试题中的统计分析结果而定。

第二部分为典型试题分析。典型试题分析部分从历年考试真题中抽取具有代表性的、经常考到的试题进行详细的分析，通过阅读这一部分，考生可以熟悉考试方法、试题形式，试题的深度和广度，以及内容的分布、解决问题的方法和技巧。

第三部分为实战练习题。实战练习题部分给出了多道试题，根据考点突破部分的知识点统计、分析的结果而命题。这些试题与考试真题具有很大的相似性，用来检查考生学习前面两个部分内容的效果。

第四部分为练习题解析。练习题解析部分是实战练习题部分的补充，对实战练习的所有习题进行了较详细的分析，并给出了解答。考生需要掌握每个练习题及其解答，这一部分可以帮助考生温习和巩固前面所学的知识，这种辅导方式保证内容全面，突出重点，为考生打造一条通向考试终点的捷径。

作者权威，阵容强大

希赛教育（www.educity.cn）专业从事人才培养、教育产品开发、教育图书出版，在职业教育方面具有极高的权威性。特别是在在线教育方面，在国内名列前茅，希赛教育的远程教育模式得到了国家教育部门的认可和推广。

希赛教育软考学院(www.csairk.com)是全国计算机技术与软件专业技术资格(水平)考试的知名培训机构,拥有近20名资深软考辅导专家,参与了高级资格的考试大纲制订工作,以及软考辅导教材的编写工作,共组织编写和出版了60多本软考教材,内容涵盖了初级、中级和高级的各个专业,包括教程系列、辅导系列、考点分析系列、冲刺系列、串讲系列、试题精解系列、疑难解答系列、全程指导系列、案例分析系列、指定参考用书系列、一本通11个系列的书籍。希赛教育软考学院的专家录制了软考培训视频教程、串讲视频教程、试题讲解视频教程、专题讲解视频教程等4个系列的软考视频,希赛教育软考学院的软考教材、软考视频、软考辅导为考生助考、提高通过率做出了不可磨灭的贡献,在软考领域有口皆碑。特别是在高级资格领域,无论是考试教材,还是在线辅导和面授,希赛教育软考学院都独占鳌头。

本书由希赛教育软考学院王勇主编,参加编写工作的人员有施游、张友生、陈勇军、刘洋波、李雄、胡钊源、桂阳、何玉云和周玲。

在线测试,心中有数

上学吧(www.shangxueba.com)在线测试平台为考生准备了在线测试,其中有数十套全真模拟试题和考前密卷,考生可选择任何一套进行测试。测试完毕,系统自动判卷,立即给出分数。

对于考生做错的地方,系统会自动记忆,待考生第二次参加测试时,可选择“试题复习”。这样,系统就会自动把考生原来做错的试题显示出来,供考生重新测试,以加强记忆。

如此,读者可利用上学吧在线测试平台的在线测试系统检查自己的实际水平,加强考前训练,做到心中有数,考试不慌。

诸多帮助,诚挚致谢

在本书出版之际,要特别感谢全国软考办的命题专家们,编者在本书中引用了部分考试原题,使本书能够尽量方便读者的阅读。在本书的编写过程中,参考了许多相关的文献和书籍,编者在此对这些参考文献的作者表示感谢。

感谢电子工业出版社孙学瑛老师,她在本书的策划、选题的申报、写作大纲的确定,以及编辑、出版等方面,付出了辛勤的劳动和智慧,给予我们很多支持和帮助。

感谢参加希赛教育软考学院辅导和培训的学员,正是他们的想法汇成了本书的原动力,他们的意见使本书更加贴近读者。

由于编者水平有限,且本书涉及的内容很广,书中难免存在错漏和不妥之处,编者诚恳地期望各位专家和读者不吝指正和帮助,对此,我们将十分感激。

互动讨论，专家答疑

希赛教育软考学院（www.csairk.com）是中国知名的软考在线教育网站，该网站论坛是国内人气很旺的软考社区，在这里，读者可以和数十万考生进行在线交流，讨论有关学习和考试的问题。希赛教育软考学院拥有强大的师资队伍，为读者提供全程的答疑服务，在线回答读者的提问。

有关本书的意见反馈和咨询，读者可在希赛教育软考学院论坛“软考教材”板块中的“希赛教育软考学院”栏目上与作者进行交流。

希赛教育软考学院

目 录

第1章 操作系统

1.1 考点突破	1
1.1.1 历年考试情况分析	1
1.1.2 操作系统概论	2
1.1.3 进程管理	5
1.1.4 存储管理	10
1.1.5 文件管理	13
1.2 典型试题分析	15
1.3 实战练习题	23
1.4 练习题解析	28

第2章 数据库系统

2.1 考点突破	33
2.1.1 历年考试情况分析	33
2.1.2 数据库模式	34
2.1.3 E-R 模型	36
2.1.4 关系代数	38
2.1.5 完整性约束	39
2.1.6 规范化理论	41
2.1.7 SQL 语言	46
2.1.8 分布式数据库	50
2.1.9 数据仓库与数据挖掘	52
2.2 典型试题分析	58
2.3 实战练习题	69
2.4 练习题解析	74

第3章 计算机硬件基础及嵌入式系统设计

3.1 考点突破	80
----------	----

3.1.1 历年考试情况分析	80
3.1.2 嵌入式系统概述	81
3.1.3 嵌入式操作系统	82
3.1.4 嵌入式系统分析与设计	84
3.1.5 嵌入式低功耗设计	88
3.1.6 存储系统	89
3.2 典型试题分析	94
3.3 实战练习题	98
3.4 练习题解析	98

第4章 数据通信与计算机网络

4.1 考点突破	101
4.1.1 历年考试情况分析	101
4.1.2 开放系统互连参考模型	102
4.1.3 TCP/IP 协议族	103
4.1.4 网络规划与设计	105
4.1.5 网络存储技术	106
4.1.6 网络应用	109
4.2 典型试题分析	112
4.3 实战练习题	121
4.4 练习题解析	124

第5章 系统安全性与保密性设计

5.1 考点突破	129
5.1.1 历年考试情况分析	129
5.1.2 安全基础技术	130
5.1.3 网络安全	132
5.1.4 计算机病毒与木马	140
5.2 典型试题分析	142

5.3 实战练习题	150
5.4 练习题解析	151

第6章 信息化基础

6.1 考点突破	157
6.1.1 历年考试情况分析	157
6.1.2 信息与信息化基本概念	157
6.1.3 政府信息化与电子政务	158
6.1.4 企业信息化与电子商务	160
6.1.5 应用集成技术	164
6.1.6 企业门户	166
6.2 典型试题分析	167
6.3 实战练习题	176
6.4 练习题解析	178

第7章 系统开发基础

7.1 考点突破	185
7.1.1 历年考试情况分析	185
7.1.2 开发管理	186
7.1.3 需求管理	193
7.1.4 软件开发方法	197
7.1.5 软件设计方法	210
7.1.6 测试与评审	216
7.1.7 软件开发环境与工具	223
7.1.8 软件过程改进	225
7.2 典型试题分析	227
7.3 实战练习题	248
7.4 练习题解析	254

第8章 软件架构设计

8.1 考点突破	268
8.1.1 历年考试情况分析	268
8.1.2 软件架构的概念	270
8.1.3 软件架构的意义	271
8.1.4 软件架构的风格	272
8.1.5 分层架构	276

8.1.6 面向服务的架构 (SOA)	280
8.1.7 特定领域软件架构	290
8.1.8 软件产品线	292
8.1.9 基于架构的软件开发方法 (ABSD)	296
8.1.10 软件架构与质量属性	302
8.1.11 软件架构评估	305
8.1.12 设计模式	311
8.2 典型试题分析	325
8.3 实战练习题	340
8.4 练习题解析	344

第9章 应用数学

9.1 考点突破	351
9.1.1 历年考试情况分析	351
9.1.2 数学建模	352
9.1.3 线性规划	354
9.1.4 图论	357
9.1.5 决策论	359
9.2 典型试题分析	363
9.3 实战练习题	371
9.4 练习题解析	374

第10章 知识产权与标准化

10.1 考点突破	380
10.1.1 历年考试情况分析	380
10.1.2 保护期限	381
10.1.3 知识产权人确定	382
10.1.4 侵权判定	382
10.1.5 标准化	383
10.2 典型试题分析	384
10.3 实战练习题	392
10.4 练习题解析	394

第11章 系统配置与性能评价

11.1 考点突破	399
-----------------	-----

11.1.1	历年考试情况分析	399
11.1.2	系统性能评价	400
11.1.3	冗余技术	402
11.1.4	容错技术	404
11.1.5	系统可靠性计算	405
11.2	典型试题分析	406
11.3	实战练习题	414
11.4	练习题解析	416

第12章 案例分析

12.1	历年考试情况分析	422
12.2	典型试题分析	422
12.2.1	结构化软件系统建模	422
12.2.2	联合需求分析会议	425
12.2.3	电子政务	428
12.2.4	软件系统架构	431
12.2.5	SOA	434
12.2.6	软件质量属性	436
12.2.7	J2EE	439
12.2.8	数据库安全与数据库 备份	441
12.2.9	数据中心建设	444
12.2.10	分布式数据库系统	447
12.2.11	Web 2.0 技术	450
12.2.12	虚拟化技术	453
12.2.13	嵌入式软件体系架构	456
12.2.14	实时系统	460
12.2.15	信息系统安全	464
12.3	实战练习题	466
12.4	练习题参考答案	483

第13章 架构设计论文

13.1	历年考试情况分析	498
13.2	典型试题分析	499
13.2.1	论软件体系结构风格 及其应用	499

13.2.2	论开放系统应用的 互操作性技术	500
13.2.3	论多层分布式结构系统的 开发	501
13.2.4	论设计模式在软件 开发中的应用	502
13.2.5	论基于场景的软件体系 结构评估方法	503
13.2.6	论基于 DSSA 的软件架构 设计与应用	504
13.2.7	论信息系统建模方法	505
13.2.8	论高可靠性系统中软件 容错技术的应用	506
13.2.9	论数据仓库的设计与 实现	507
13.2.10	论计算机网络的 安全性	507
13.2.11	论 Web Service 技术的 应用与发展趋势	508
13.2.12	论系统的可靠性	509
13.2.13	论软件可靠性设计与 应用	510
13.2.14	论软件需求获取技术及 应用	510
13.3	实战练习题	512
13.4	练习题写作要点	516
13.4.1	试题 1 写作要点	516
13.4.2	试题 2 写作要点	516
13.4.3	试题 3 写作要点	518
13.4.4	试题 4 写作要点	519
13.4.5	试题 5 写作要点	520
13.4.6	试题 6 写作要点	523
13.4.7	试题 7 写作要点	523
13.4.8	试题 8 写作要点	529
13.4.9	试题 9 写作要点	531
13.4.10	试题 10 写作要点	531

1

第 1 章 操作系统

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

- 操作系统的类型和结构。
- 操作系统基本原理。
- 网络操作系统及网络管理。

1.1 考点突破

从历年的考试情况来看，本章主要考查操作系统的概念及特点、进程管理、存储管理、文件管理。

1.1.1 历年考试情况分析

在历年的考试试题中，有关操作系统知识的试题如表 1-1 所示。

表 1-1 操作系统知识试题分布表

题号	2009.11	2010.11	2011.11
1	系统结构分层	分布式操作系统特点	操作系统接口
2	前趋图	文件系统	前趋图与 PV 操作
3		页面淘汰算法 (LRU)	
4			

按照知识点进行总结和归类的试题分布情况如表 1-2 所示。

表 1-2 操作系统知识归类表

知识点	2009.11	2010.11	2011.11
操作系统概论	1	1	1
进程管理	3	0	3
存储管理	0	2	0
文件管理	0	1	0
合计	4	4	4

从表 1-2 中可以看出,操作系统知识方面的内容在历年的考试中分值非常稳定,一直稳定维持在 4 分。所占分数比例的趋势如图 1-1 所示。

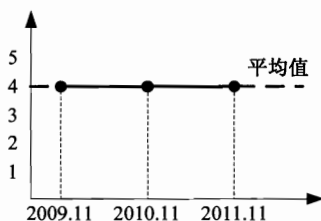


图 1-1 操作系统知识历年试题比例趋势图

1.1.2 操作系统概论

操作系统的定义、功能、类型和层次结构是理解操作系统的工作机制的基础,需要深入理解和掌握。

1. 操作系统的定义

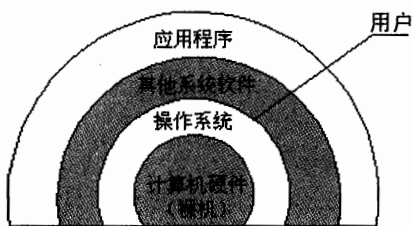


图 1-2 操作系统与硬件/软件的关系

任何一个计算机系统都是由两个部分组成的:计算机硬件系统和计算机软件系统。操作系统(Operating System, OS)是计算机系统核心系统软件,负责管理和控制计算机系统中硬件和软件资源,合理地组织计算机工作流程和有效利用资源,在计算机与用户之间起接口的作用,如图 1-2 所示。

其中,其他系统软件包含但不限于:语言处理程序、中间件。

在计算机系统中引入操作系统的目的可以从 4 个方面来理解。

(1) 用户观点

操作系统是用户与计算机之间的接口。一方面,用户可以透明地使用计算机软/硬件资源;另一方面,操作系统提供了一些功能强大的系统调用,用户软件可以使用这些系统调用请求操作系统服务。

(2) 资源管理观点

操作系统是计算机资源的管理者，它管理和分配计算机系统硬件和软件资源，合理地组织计算机的工作流程，使资源能为多个用户共享，当用户程序和其他程序争用这些资源时，提供有序的和可控的分配。

(3) 进程观点

操作系统由一个系统核心和若干并发运行的程序组成。这些运行的程序称为“进程”，进程可以分为用户进程和系统进程两大类。每个进程完成特定的任务，系统核心则控制和协调这些进程的运行。

(4) 分层观点

操作系统通常采用分层结构实现，各层次的程序按照一定的结构组织并协调工作。

2. 操作系统的分类

操作系统的基本类型有：批处理操作系统、分时操作系统、实时操作系统、网络操作系统、分布式操作系统、嵌入式操作系统、微内核操作系统。

(1) 批处理操作系统 (Batch Processing Operating System)

批处理操作系统也称为作业处理系统。在批处理操作系统中，作业成批地装入计算机中，由操作系统在计算机的输入并将其组织好，按一定的算法选择其中的一个或多个作业，将其调入内存使其运行。运行结束后，把结果放入磁盘输出井，由计算机统一输出后交给用户。

批处理操作系统中配置了一个监督程序，在该监督程序控制下，系统能够对一批作业自动进行处理。其基本特征是“批量”，把作业的吞吐量作为主要目标，同时兼顾作业的周转时间。批处理操作系统又分为单道批处理和多道批处理系统。

单道批处理系统在内存中只能存放一道作业，大大减少了人工操作的时间，提高了机器的利用率。但是，对于某些作业来说，当它发出 I/O 请求后，CPU (Central Processing Unit, 中央处理单元) 必须等待 I/O 的完成，而由于 I/O 设备的低速性，从而使 CPU 的利用率很低。为了改善 CPU 的利用率，引入了多道程序设计技术，就形成了多道批处理操作系统。

在多道批处理操作系统中，不仅在内存中可同时有多道作业在运行，而且作业可随时被调入系统，并存放在外存中形成作业队列。然后，由操作系统按一定的原则，从作业队列中调入一个或多个作业进入内存运行。多道批处理系统具有资源利用率高和系统吞吐量大的优点，但它将用户和计算机操作员分开，使用户无法直接与自己的作业进行交互。另外，作业要进行排队，依次处理，因此，作业的平均周转时间较长。

(2) 分时操作系统 (Time Share Operating System)

为了解决批处理系统无法进行人机交互的问题，并使多个用户能同时通过自己的终端以交互方式使用计算机，共享主机中的资源，为此，系统中采用了分时技术，即把 CPU 的时间划分成很短的时间片，轮流地分配给各个终端作业使用。这种操作系统称为分时操作系统，简称分时系统。

对于某个作业而言,若在分配给它的时间片内,作业没有执行完毕,也必须将 CPU 交给下一个作业使用,并等下一轮得到 CPU 时再继续执行。这样,系统便能及时地响应每个用户的请求,从而使每个用户都能及时地与自己的作业交互。分时系统具有多路性、独立性、及时性、交互性和同时性等特征。

(3) 实时操作系统 (Real Time Operating System)

实时操作系统是指当外界事件或数据产生时,能够接收并以足够快的速度予以处理,其处理的结果又能在规定的时间内来控制生产过程或对处理系统快速响应,并控制所有实时任务协调一致运行的操作系统。因而,提供及时响应和高可靠性是其主要特点。

实时操作系统有硬实时和软实时之分,硬实时要求在规定的时间内必须完成操作,这是在操作系统设计时保证的;软实时则只要按照任务的优先级,尽可能快地完成操作即可。我们通常使用的操作系统在经过一定改变之后就可以变成实时操作系统。

(4) 网络操作系统 (Network Operating System)

网络操作系统是指在计算机网络环境下,具有网络功能的操作系统。计算机网络是一个数据通信系统,它把地理上分散的计算机和终端设备连接起来,达到数据通信和资源共享的目的。网络操作系统最主要的特点是网络中各种资源的共享,以及各台计算机之间的通信。有关数据通信与计算机网络方面的详细知识,将在第 4 章中介绍。

(5) 分布式操作系统 (Distributed Operating System)

分布式系统是由多台计算机组成的系统,系统中若干台计算机可以相互合作,共同完成同一个任务。在分布式系统中,任意两台计算机之间都可以利用通信来交换信息,系统中的资源为所有用户共享。分布式系统的优点是各节点的自治性好、资源共享的透明性强、各节点具有协同性,其主要缺点是系统状态不精确、控制机构复杂、通信开销会引起性能的下降。

分布式操作系统是网络操作系统的更高级形式,它保持了网络操作系统所拥有的全部功能,与网络操作系统的主要区别在于任务的分布性,即把一个大任务分为若干个子任务,分派到不同的 CPU 上执行。

(6) 嵌入式操作系统 (Embedded Operation System)

嵌入式操作系统运行在嵌入式智能芯片环境中,对整个智能芯片和它所操作、控制的各种部件装置等资源进行统一协调、处理、指挥和控制。嵌入式操作系统具有微型化、可定制、实时性、可靠性、易移植性等特点。嵌入式实时操作系统是指系统能及时响应外部事件的请求,在规定的时间内完成对该事件的处理,并控制所有实时任务协调一致地运行。嵌入式实时操作系统的特点是及时性、支持多道程序设计、高可靠性和较强的过载防护能力。

(7) 微内核操作系统 (Microkernel Operating System)

微内核操作系统结构是 20 世纪 80 年代后期发展起来的,其基本思想是将操作系统中最基本的部分放入内核中,而把操作系统的绝大部分功能都放在微内核外面的一组服务器中实现。这样使得操作系统内核变得非常小,自然提高了系统的可扩展性,

增强了系统的可靠性和可移植性，同时微内核操作系统提供了对分布式系统的支持，融入了面向对象技术。虽然微内核操作系统具有诸多优点，但它非常完美无缺，在运行效率方面它就不如以前传统的操作系统。

当前比较流行的、能支持多处理机运行的操作系统，几乎全部都采用了微内核结构，如卡耐基梅隆大学研制的 Mach OS，便属于微内核结构操作系统；又如当前广泛使用的 Windows 操作系统，也采用了微内核结构。

3. 操作系统的功能

从资源管理的观点看，操作系统的功能分成五大部分，即进程管理（处理机管理）、存储管理、文件管理、作业管理和设备管理。这五大部分相互配合，协调工作，实现对计算机系统的资源管理和控制程序的执行，为用户提供方便的使用接口和良好的运行环境。

4. 操作系统的结构设计模式

（1）模块化结构

操作系统由许多标准的、可兼容的基本单位构成（称为模块），各模块相对独立，模块之间通过规定的接口相互调用。模块化设计方法的优点是缩短了系统的开发周期，缺点是模块之间调用关系复杂、相互依赖，从而使分析、移植和维护系统较易出错。

（2）层次化结构

层次化结构是指操作系统由若干模块按照某种逻辑关系进行分层组合而成，各层之间只能单向依赖。优点是大大地简化了接口的设计，整个系统的正确性由各层次的正确性来保证，易于保证可靠性，也便于维护和移植。

（3）客户/服务器结构

操作系统中只包括一个最小的核心，操作系统的其他功能（如文件服务、进程服务等）由用户进程来实现。优点是服务器以用户进程的形式运行而不是运行于核心态，它们不直接访问硬件，单个服务器的崩溃不会导致整个系统的崩溃，它适用于分布式系统。

（4）对象模式

面向对象的操作系统是按照面向对象思想设计的，具有数据隐藏及消息激活对象等特征。其中，对象是对操作系统管理的信息和资源的抽象，可以被视为受保护的信息或资源的总称。优点是适用于网络操作系统和分布式操作系统中，如 Windows NT、Windows Server 2003、Windows Server 2008 都被称为对象操作系统。

（5）对称多处理模式

如果一个操作系统在系统中的所有处理机运行且共享同一内存，这样的系统就是一个对称多处理系统。优点是适合共享存储器结构的多处理机系统，即紧耦合的多处理机系统。

1.1.3 进程管理

进程管理也称为处理机管理，该部分功能是操作系统最为重要的部分，所以也是

整个操作系统部分的考查重点，其主要知识点有：进程状态转换图、信号量与 PV 操作、死锁问题、银行家算法。

1. 进程状态转换图

进程状态转换图用于展现进程的状态，以及各种状态之间的转换。最为常见的有三态模型和五态模型，其后又提出了七态模型。在考试中，要求考生掌握三态模型与五态模型。五态模型是对三态模型的扩展（即五态模型已经包含了三态模型）。标准的五态模型如图 1-3 所示。

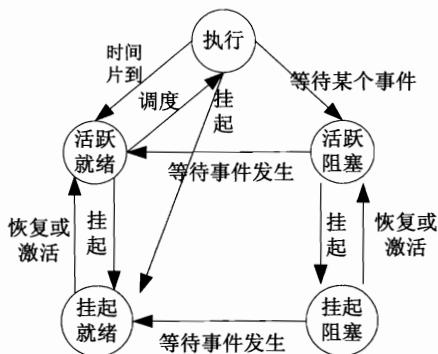


图 1-3 进程状态转换五态模型

从图 1-3 中可以看出，五态模型中的五态为：执行状态（运行状态）、活跃就绪状态、活跃阻塞状态、挂起就绪状态、挂起阻塞状态。其中，前三种状态组成了三态模型。

- 执行状态：指进程占有处理机正在 CPU 上执行的状态。在单 CPU 系统中，每一时刻只有一个进程处于执行状态。
- 活跃就绪状态：指进程分配到除处理机以外的必需的资源（已经具备了执行的条件）的状态。进程被创建后处于就绪状态，处于就绪状态的进程可以有多个。
- 活跃阻塞状态：指进程因等待某个事件的发生而放弃处理机进入等待状态。系统中处于这种状态的进程可以有多个。

在三态模型中，总是假设所有的进程都在内存中。事实上，可能出现这样一些情况，例如，由于进程的不断创建，系统的资源已经不能满足进程运行的要求，这个时候就必须把某些进程挂起，对换到磁盘镜像区中，暂时不参与进程调度，起到平滑系统操作负荷的目的。这就形成了挂起就绪状态和挂起阻塞状态。

- 挂起就绪状态：指进程被移至磁盘镜像区中，此时进程只缺处理机资源。
- 挂起阻塞状态：指进程被移至磁盘镜像区中，此时进程除了缺处理机资源，还缺其他资源。

2. 信号量与 PV 操作

在操作系统中，进程之间经常会存在互斥（都需要共享独占性资源时）和同步（完成异步的两个进程的协作）两种关系。为了有效地处理这两种情况，W. Dijkstra 在 1965

年提出信号量和PV操作。

信号量是一种特殊的变量，表现形式是一个整型S和一个队列。

- P操作：也称为down()、wait()操作，使 $S=S-1$ ，若 $S<0$ ，进程暂停执行，放入信号量的等待队列。
- V操作：也称为up()、signal()操作，使 $S=S+1$ ，若 $S\leq 0$ ，唤醒等待队列中的一个进程。

(1) 完成互斥控制

为了保护共享资源，不让多个进程同时访问这个共享资源，换句话说，就是阻止多个进程同时进入访问这些资源的代码段，这个代码段称为临界区（也称为管程），这种一次只允许一个进程访问的资源称为临界资源。为了实现进程互斥地进入自己的临界区，代码可以如下所示：

```
P(信号量)
临界区
V(信号量)
```

由于只允许一个进程进入，因此信号量中整型值的初始应该为1。该值表示可以允许多少个进程进入，当该值 <0 时，其绝对值就是等待使用的进程数，也就是等待队列中的进程数。而当一个进程从临界区出来时，就会将整型值加1，如果等待队列中还有进程，则调入一个新的进程进入（唤醒）。

(2) 完成同步控制

最简单的同步形式是：进程A在另一个进程B到达L2以前，不应前进到超过L1，这样就可以使用程序，如下所示：

```
进程A          进程B
...             ...
L1: P(信号量)    L2: V(信号量)
...             ...
```

因此，要确保进程B执行V操作之前，不让进程A的运行超过L1，因此信号量的初值就应该为0。这样，如果进程A先执行到L1，那么执行P操作后，信号量的整型值就会小于1，也就停止执行。直到进程B执行到L2时，将信号量的整型值加1，并唤醒它以继续执行。

在考试中，该知识点出题形式主要是给出一系列操作，让考生在适当位置填充P操作或V操作。

例如，某工厂仓库有一名保管员，该仓库可存放 n 箱零件。该工厂生产车间有 m 名工人，只要仓库空闲，工人将生产好的整箱零件放入仓库，并由保管员登记入库数量；该工厂销售部有 k 名销售员，只要仓库库存数能满足客户要求，便可提货，并由保管员登记出库数量。规定工人和销售员不能同时进入仓库，但是工人和工人，销售员和销售员可以同时进入仓库，其工作流程如图1-4所示。

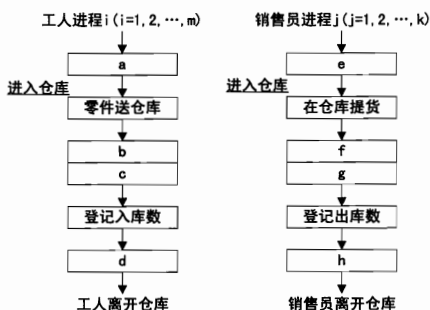


图 1-4 仓库管理系统流程

为了利用 PV 操作正确地协调工人和销售员进程之间的工作，设置了信号量 S1、S2 和 S3，它们的初值分别为 n、0 和 1。则图 1-4 中的 a~h 应分别填写什么操作呢？

根据问题给出的条件，我们可以判断出，信号量 S1 表示仓库空闲位置个数，初值为 n；S2 表示仓库中零件箱数，初值为 0；S3 用于实现对保管员的互斥访问，初值为 1。

对于工人进程，首先应执行 P(S1)，看仓库中是否有空闲位置，若有，则将零件送入仓库，然后执行 V(S2)，表明仓库中已有一箱零件，通知销售员可以提货。然后执行 P(S3)，看保管员是否空闲，若空闲，则登记入库数，然后执行 V(S3)，使保管员处于空闲状态。

对于销售员进程，首先执行 P(S2)，看仓库中是否有货物，若有，则可以提货，然后执行 V(S1)，表明已经提走一箱零件，空闲出一个位置，工人进程可以放置货物；然后执行 P(S3)，看保管员是否空闲，若空闲，则登记出库数，然后执行 V(S3)，使保管员处于空闲状态。

3. 前趋图

前趋图是一个由节点和有向边构成的有向无循环图。该图通常用于表现事务之间先后顺序的制约关系。图中的每个节点可以表示一个语句、一个程序段或是一个进程，节点间的有向边表示两个节点之间存在的前趋关系。

例：在计算机中，经常采用流水线方式执行指令，若每一条指令都可以分解为取指、分析和执行 3 步。取指操作为 A_i，分析操作为 B_i 和执行操作为 C_i (i=1,2,3)。如图 1-5 所示为 3 个任务各程序段并发执行的前驱图。

图中 A₁ 没有前趋节点，称为开始节点，它不受任何制约，可以直接执行；而 B₁ 与 A₂ 只能在 A₁ 执行完成之后才能开始，而 B₂ 必须在 B₁ 与 A₂ 完成之后才能开始；C₃ 没有后继节点，称为终止节点。

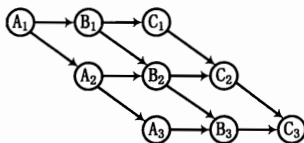


图 1-5 前趋图

在前趋图中，执行先后顺序的制约关系可分为两种：直接制约和间接制约。

直接制约通常是指一个操作中，多个步骤之间的制约关系，也可以说是“同步的进程之间的制约关系”。例如，在图 1-5 中， A_1 、 B_1 、 C_1 是一条指令的取指、分析、执行 3 个步骤，所以它们之间的关系是直接制约。

间接制约通常是指多个操作之间相同步骤的制约关系，也可以说是“互斥的进程之间的制约关系”。例如，在图 1-5 中， A_1 、 A_2 、 A_3 之间就存在间接制约的关系。

前趋图的应用广泛，在项目开发中，可用前趋图来分析哪些活动可以并行完成。同时，项目管理工具：Pert 图、单（双）代号网络图等都融入了前趋图的思想。

4. 死锁问题

死锁是指各并发进程彼此互相等待对方所拥有的资源，且这些并发进程在得到对方的资源之前不会释放自己所拥有的资源。从而造成大家都想得到资源而又都得不到资源，各并发进程不能继续向前推进的状态。

产生死锁的根本原因在于系统提供的资源个数少于并发进程所要求的该类资源数。产生死锁有 4 个必要条件。

- 互斥条件：即一个资源每次只能被一个进程使用，在操作系统中这是真实存在的情况。
- 保持和等待条件：有一个进程已获得了一些资源，但因请求其他资源被阻塞时，对已获得的资源保持不放。
- 不剥夺条件：有些系统资源是不可剥夺的，当某个进程已获得这种资源后，系统不能强行收回，只能由进程使用完时自己释放。
- 环路等待条件：若干个进程形成环形链，每个进程都占用对方要申请的下一个资源。

对待死锁的策略主要有以下 4 种。

- 死锁的预防。不让任一产生死锁的必要条件发生就可以预防死锁。
- 死锁的避免。这种策略不对用户进程的推进顺序加以限制，在进程申请资源时先判断这次分配安全否，只有安全才实施分配，典型的算法是银行家算法。
- 死锁的检测。这种策略采用资源请求分配图的化简方法来判断是否发生了不安全状态。资源请求分配图是一种有向图，表示进程与资源之间的关系。死锁的检测是在需要的时刻执行的，当发现系统处于不安全状态时，即执行死锁的解除策略。
- 死锁的解除。解除死锁的基本方法是剥夺。一种方法是把资源从一些进程处剥夺分给别的进程，被剥夺资源的进程则需回退到请求资源处重新等待执行；另一种方法是终止一个进程，剥夺其全部资源，以后再重新运行被终止的进程。

5. 银行家算法

银行家算法是一种经典的死锁避免方法。银行家算法的基本思想是：当某个进程提出申请时，必须判断将资源分配给该进程后，会不会引起死锁。若不会，则进行分配；否则就不分配。这样做能保证在任何时刻至少有一个进程可以得到所需的全部资

源而执行结束，并将归还资源加入到系统的剩余资源中，这些资源又至少可以满足一个进程的最大需求，于是保证所有进程都能在有限的时间内得到需求的全部资源。

按照银行家算法的思想，当进程请求资源时，系统将按如下原则分配资源：

- 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。
- 当系统现有的资源能满足进程尚需资源数时，必须测试系统现存的资源能否满足该进程尚需的最大资源数，若能满足，则按当前的申请量分配资源，否则，推迟分配。

1.1.4 存储管理

存储管理是操作系统的重要职能之一，它的主要任务是对内存空间进行分配与回收。由于计算机内存容量有限，故在此基础上发展产生了虚拟存储系统。虚拟存储系统的基本思想是用外存来换取内存。它通过将运行进程访问的地址（逻辑地址、虚拟地址）与主存的物理地址（实地址）分开，从而使得提供大于物理地址的逻辑地址空间成为可能。建立虚拟地址和实地址之间的对应关系、实现转换的工作就称为“虚存管理”。在本节中，主要介绍虚存的组织结构及管理。

1. 页式存储组织

页式存储组织的基本原理是将各进程的虚拟空间划分为若干个长度相等的页，把内存空间以与页相等的大小划分为大小相等的片或页面，采用请求调页或预调页技术实现内、外存的统一管理。

页式存储组织的主要优点是利用率高，产生的内存碎片小，内存空间分配及管理简单。主要缺点是要有相应的硬件支持，增加了系统开销；请求调页的算法如选择不当，有可能产生“抖动”（又称 Belady）现象。

2. 段式存储组织

一个作业是由若干个具有逻辑意义的段（如主程序、子程序、数据段等）组成的。在分段系统中，允许程序（作业）占据内存中若干分离的分区。分段系统中的虚地址是一个有序对（段号，段内位移）。系统为每一个作业建立一个段表，其内容包括段号与内存起始地址的对应关系、段长和状态等。状态指出这个段是否已调入内存，若已调入内存，则指出这个段的起始地址位置，状态同时也指出这个段的访问权限。如果该段尚未调入内存，则产生缺段中断，以便装入所需要的段。

段式存储组织的主要优点有：便于多道程序共享内存、便于对存储器的保护、各段程序修改互不影响。其缺点是内存利用率低、内存碎片浪费大。

3. 段页式存储组织

段页式存储组织是分段式和分页式结合的存储组织方法，这样可充分利用分段管理和分页管理的优点。在段页式管理的存储器中，程序按逻辑单位分成基本独立的段，

再把每段分成固定大小的页。内存则等分成与上述页大小相等的页。程序对内存的调入或调出是按页进行的。但它又可按段实现共享和保护。

在多道程序环境中，每道程序都有一张段表和一个作为用户标志的基号。一个逻辑地址中，除了基号 x 、段号 s 和页号 p 外，还有一个页内地址 d 。每个逻辑地址变换成实地址的过程如下：

根据基号找到相应的基址寄存器，由该基址寄存器内容找到该程序对应的段表起始地址，再由段号找到该段表中相应行地址，该行地址中的内容为页表起始地址，再由页号找到物理页号的地址（已是内存中的某页），它与页内地址拼接后即得物理地址。可见段页式管理中需要多次查表才能最终获得物理地址。该过程可简单地用一个式子来示意，即

$$(((x) + s) + p) \times 2^n + d$$

其中， (x) 表示基寄存器中地址为 x 的单元内容， n 为页内地址的位数。

段页式管理将段式存储管理和页式存储管理两种方式相结合，互相取长补短，充分发挥了它们的优点。使段页式虚拟存储器管理方案具有空间浪费小、存储共享容易、存储保护容易、能动态连接的特点。但由于管理软件的增加，复杂性和开销也随之增加，需要的硬件及占用的内存也有所增加，使得执行速度大大下降。

4. 页面置换算法

由于实际主存是小于虚存的，因此可能会发生内存中已满，但需要使用的页不在主存中这一情况。这时就需要进行置换，即将一些主存中的页淘汰到外存，腾出空间给要使用的页，这个过程也称为 Swapping。其工作原理与 Cache 调入相类似。常见的页面置换算法有以下几种。

(1) 最优算法 (OPT)

淘汰不用的或最远的、将来才用的页。这是一种理想算法，不可能实现，只是用来作为衡量算法效率的参照物。

(2) 随机算法 (RAND)

随机淘汰，这种算法开销小，但性能不稳定。

(3) 先进先出算法 (FIFO)

淘汰最早调入（也是驻留时间最长）的页。实现 FIFO 算法需要把各个已分配页面按页面分配时间顺序链接起来，组成 FIFO 队列，并设置一个置换指针，指向 FIFO 队列的队首页面。FIFO 算法忽略了一种现象的存在，那就是在内存中停留时间最长的页往往也是经常要访问的页。将这些页淘汰，很可能刚置换出去，又请求调用该页，致使缺页中断太频繁，严重降低内存的利用率。

FIFO 的另一个缺点是它可能会产生一种异常现象。一般来说，对于任一作业或进程，如果给它分配的内存页面数越接近于它所要求的页面数，则发生缺页的次数会越少。但使用 FIFO 算法时，有时会出现分配的页面数增多，缺页次数反而增加的现象，称为“抖动”（又称 Belady）现象。

(4) 最近最久未使用算法 (LRU)

淘汰离当前时刻最近的一段时间内使用得最少的页。例如, 考虑一个仅 460 个字的程序的内存访问序列 (10, 11, 104, 170, 73, 309, 185, 245, 246, 434, 458, 364), 页面的大小为 100 个字节, 则 460 个字节应占 5 页, 编号为 0~4, 第 0 页字节为 0~99, 第 1 页为 100~199, 依此类推。得到页面的访问序列是 (0, 0, 1, 1, 0, 3, 1, 2, 2, 4, 4, 3), 可简化为 (0, 1, 0, 3, 1, 2, 4, 3)。如果内存中有 200 个字节可供程序使用, 则内存提供 2 个页帧供程序使用。按照 FIFO 算法, 共产生 6 次缺页中断, 如表 1-3 所示。

表 1-3 FIFO 算法缺页中断

0	1	0	3	1	2	4	3
0	0	0	3	3	3	4	4
	1	1	1	1	2	2	3
×	×		×		×	×	×

按照 LRU 算法, 共产生 7 次缺页中断, 如表 1-4 所示。

表 1-4 LRU 算法缺页中断

0	1	0	3	1	2	4	3
0	0	0	0	1	1	4	4
	1	1	3	3	2	2	3
×	×		×	×	×	×	×

(5) 最近没有使用页面置换算法 (NUR)

在需要置换某一页时, 从那些最近的一个时期内未被访问的页任选一页置换。只要在页表中增设一个访问位即可实现。当某页被访问时, 访问位置为 1, 否则访问位置为 0。系统周期性地对所有引用位清 0。当需淘汰一页时, 从那些访问位为零的页中选一页进行淘汰。

(6) 时钟页面替换算法 (Clock)

使用页表中的引用位, 将作业已调入内存的页面连成循环队列, 用一个指针指向循环队列中的下一个将被替换的页面。其实现方法如下: 一个页面首次装入内存时, 其引用位置 1; 在内存中的任何一个页面被访问时, 其引用位置 1; 淘汰页面时, 存储管理从指针当前指向的页面开始扫描循环队列, 把所遇到的引用位是 1 的页面的引用位清 0, 并跳过这个页面; 把所遇到的引用位是 0 的页面淘汰掉, 指针推进一步; 扫描循环队列时, 如果遇到的所有页面的引用位均为 1, 则指针就会绕整个循环队列一圈, 将碰到的所有页面的引用位清 0; 指针停在起始位置, 并淘汰掉这一页, 然后指针推进一步。

5. 局部性原理

存储管理策略的基础是局部性原理, 即进程往往会不均匀地、高度局部化地访问内存。局部性分为时间局部性和空间局部性。时间局部性是指最近访问存储位置, 很可能在不久的将来还要访问; 空间局部性是指存储访问有聚集的倾向, 当访问了某个位置后, 很可能也要访问其附近的位置。

根据局部性原理的特征性, Denning 阐述了程序性能的工作集理论。工作集是进程频繁访问的页面的集合。工作集理论指出, 为使进程有效地运行, 它的页面工作集应驻留内存中。否则, 由于进程频繁地从外存请求页面, 而出现称为“抖动”(又称 Belady) 的过度的页面调度活动。此时, 处理页面调度的时间超过了程序的执行时间。显然, 此时 CPU 的有效利用率会急速下降。

工作集的大小依赖于工作集窗口(进程在定时间间隔中涉及的页面的集合)的大小, 在进程执行时, 工作集会发生变化。有时, 当进程进入另一个完全不同的执行阶段时, 工作集会出现显著的变化。不过在一个进程的执行过程中, 工作集的大小处于稳定状态的时间基本上占绝大多数。

另一种控制“抖动”的技术是控制缺页率。操作系统规定缺页率的上下限, 当一个进程的缺页率高于上限时, 表明该进程需要更大的内存空间, 则分配较多的内存页面给它。当进程的缺页率低于下限时, 表明该进程占用的内存空间过大, 可以适当收回若干内存页面。

1.1.5 文件管理

对文件管理部分的考查主要集中于位示图、树形目录结构及索引文件结构。

1. 树形目录结构

在计算机的文件系统中, 一般采用树形目录结构。在树形目录结构中, 树的根节点为根目录, 数据文件作为树叶, 其他所有目录均作为树的节点。

根目录隐含于一个硬盘的一个分区中, 根目录在最顶层。它包含的子目录是一级子目录。每一个一级子目录又可以包含若干二级子目录, …… , 这样的组织结构就叫作目录树。

当前盘和当前目录是系统默认的操作对象。如果用户没有指明操作对象, 系统就将用户命令指向当前盘和当前目录。

路径是指从根目录或者当前目录开始到访问对象(目录或者文件), 在目录树中经过的所有目录的序列。例如, “c:\dos\mouse\mouse” 就是 Windows 系统中的一条路径。在树形目录结构中, 从根目录到任何数据文件之间, 只有一条唯一的通路, 从树根开始, 把全部目录文件名与数据文件名, 依次用 “/” (UNIX/Linux 系统) 或 “\” (Windows 系统) 连接起来, 构成该数据文件的路径名, 且每个数据文件的路径名是唯一的。这样, 可以解决文件重名问题。

从树根开始的路径为绝对路径, 如果文件系统有很多级时, 使用不是很方便, 所以引入相对路径, 即是从当前目录开始, 再逐级通过中间的目录文件, 最后到达所要访问的数据文件。

绝对路径给出文件或目录位置的完全的描述, 通常由层次结构的顶端开始(根目录), 通常第一个字符是 “/” (UNIX/Linux 系统) 或者是盘符 (Windows 系统)。相对路径通常由目录结构中的当前的位置开始, 一般都比绝对路径要短。

父目录是指当前路径的上一层目录。每个目录下都有代表当前目录的 “.” 文件和代表当前目录父目录的 “..” 文件, 相对路径名一般就是从 “..” 开始的。

2. 位示图

位示图法是为管理磁盘空闲存储空间而提出的一种方法。该方法是在外存上建立一张位示图 (Bitmap)，记录文件存储器的使用情况。每一位仅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。文件存储器上的物理块依次编号为：0、1、2、…。假如系统中字长为 32 位，有 4096 个物理块，那么在位示图中的第 1 个字对应文件存储器上的 0、1、2、…、31 号物理块；第 2 个字对应文件存储器上的 32、33、34、…、63 号物理块；第 32 字对应文件存储器上的 4064、4065、…、4095 号物理块。这样位示图的大小为 32 字。

位示图是利用二进制的一位来表示磁盘中一个盘块的使用情况，如图 1-6 所示。当其值为“0”时，表示对应的盘块空闲；为“1”时表示已分配。由所有盘块对应的位构成一个集合，称为位示图。位示图也可描述为一个二维数组 `map: Var map:array[1..m,1..n]of bit;`

	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	0	0	0	0
3																
⋮																
15																

图 1-6 位示图

3. 索引文件

索引文件是一种对文件存储不连续分配的方法。为每个文件建立一张索引表，索引表中的每一表项指出文件信息所在的逻辑块号 and 与之对应的物理块号。

索引文件既可以满足文件动态增长的要求，又可以方便而迅速地实现随机存取。对一些大的文件，当索引表的大小超过一个物理块时，会发生索引表的分配问题。一般采用多级（间接索引）技术，这时，在由索引表指出的物理块中存放的不是文件存放处而是存放文件信息的物理块地址。这样，如果一个物理块能存储 n 个地址，则一级间接索引将使可寻址的文件长度变成 n^2 块，对于更大的文件可以采用二级甚至三级间接索引（例如，UNIX 操作系统采用三级索引结构，如图 1-7 所示）。

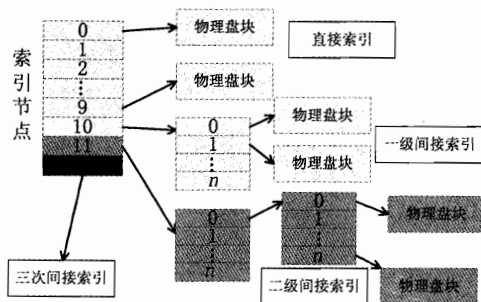


图 1-7 索引结构图

索引文件的优点是既适用于顺序存取，又适用于随机存取；缺点是索引表增加了存储空间开销。另外，在存取文件时需要访问两次磁盘，一次是访问索引表，另一次是根据索引表提供的物理块号访问文件信息。为了提高效率，一种改进的方法是，在对某个文件进行操作之前，预先将索引表调入内存。这样，文件的存取就能直接从内存的索引表中确定相应的物理块号，从而只需要访问一次磁盘。

1.2 典型试题分析

试题 1

(1) 不是操作系统关心的主要问题。

- (1) A. 管理计算机裸机 B. 设计、提供用户程序与计算机硬件系统的界面
C. 管理计算机系统资源 D. 高级程序设计语言的编译器

试题分析

操作系统有 2 个重要的功能，一是通过资源管理，提高计算机系统的效率；二是改善人机界面，向用户提供友好的工作环境。因此，在所给的选项中，显然只有 D 不是操作系统关心的主要问题。

试题答案

(1) D

试题 2

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做不同的进程，其工作流程如图 1-8 所示。利用 PV 操作实现该过程，设置信号量 S_1 、 S_2 和 S_n ，初值分别为 0、0、 n 。则图 1-8 中 a_1 和 a_2 应填入 (2)， b_1 和 b_2 应填入 (3)。

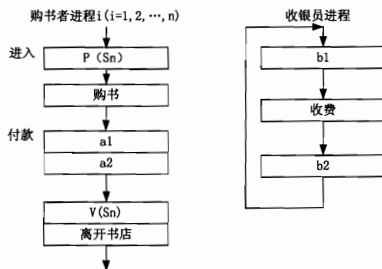


图 1-8 购书工作流程

- (2) A. $V(S_1)$ 、 $P(S_2)$ B. $V(S_n)$ 、 $P(S_n)$
C. $P(S_1)$ 、 $V(S_2)$ D. $P(S_2)$ 、 $V(S_1)$
(3) A. $P(S_n)$ 、 $V(S_2)$ B. $V(S_n)$ 、 $P(S_2)$
C. $P(S_1)$ 、 $V(S_2)$ D. $P(S_2)$ 、 $V(S_1)$

试题分析

根据试题描述,在本题中, S_n 显然是代表允许进入书店的购书者的人数,初值为 n ,表示可以进入 n 个购书者。 S_2 用于实现对收银员的互斥访问,初值为0,表示收银员空闲,可以付款。 S_1 代表有多少顾客等待付款,初值为0,表示没有顾客付款。

整个工作流程描述如下:首先是收银员上班,执行 $P(S_1)$,看是否有顾客付款,如果没有,则阻塞。否则,开始收费,收费完毕后,再执行 $V(S_2)$,表示收银员空闲。购书者进入书店后,执行 $P(S_n)$,然后购书,当要付款时,执行 $V(S_1)$,表示多了一个顾客在等待付款,然后执行 $P(S_2)$,看收银员是否忙,如果忙,则阻塞。否则,开始付款。顾客离开书店时,执行 $V(S_n)$ 。

试题答案

(2) A

(3) C

试题 3

影响文件系统可靠性因素之一是文件系统的一致性问题,如果读取(4)的某磁盘块,修改后在写回磁盘前系统崩溃,则对系统的影响相对较大。通常的解决方案是采用文件系统的一致性检查,一致性检查包括块的一致性和文件的一致性检查。在块的一致性检查时,检测程序构造一张表,表中为每个块设立两个计数器,一个跟踪该块在文件中出现的次数,一个跟踪该块在空闲表中出现的次数。若系统有16个块,检测程序通过检测发现表(5)状态下的文件系统是一致的。

(4) A. 用户文件的某磁盘块

B. 空闲表磁盘块

C. 用户目录文件

D. 系统目录文件

(5)

A.	块号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	计数器 1:使用中的块	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0	1
	计数器 2:空闲块	0	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0

B.	块号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	计数器 1:使用中的块	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0	1
	计数器 2:空闲块	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1	0

C.	块号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	计数器 1:使用中的块	1	0	1	0	1	1	3	1	0	0	1	1	1	0	0	1
	计数器 2:空闲块	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1	0

D.	块号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	计数器 1:使用中的块	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0	1
	计数器 2:空闲块	0	1	0	1	0	0	0	0	2	1	0	0	0	1	1	0

试题分析

影响文件系统可靠性因素之一是文件系统的一致性问题,如果读取系统目录文件的某磁盘块,修改后在写回磁盘前系统崩溃,则对系统的影响相对较大。因为很多文

件系统是先读取磁盘块到主存，在主存进行修改，然后写回磁盘。但如果读取某磁盘块，修改后再将信息写回磁盘前系统崩溃，则文件系统就可能会出现不一致性状态。如果这些未被写回的磁盘块是系统目录文件，如索引节点块、目录块或空闲管理块等，那么后果是很严重的。

通常的解决方案是采用文件系统的一致性检查，一致性检查包括块的一致性检查和文件的一致性检查。在块的一致性检查时，检测程序构造一张表，表中为每个块设立两个计数器，一个跟踪该块在文件中出现的次数，一个跟踪该块在空闲表中出现的次数。

若系统有 16 个块，当进行文件的一致性检查时发现，选项 A 的第 3 块在计数器 1 中为 0，这意味着没有文件使用这个块，但在计数器 2 中也为 0，这意味着这个块不空闲。因此，文件系统进行一致性检查时发现出了问题。选项 C 的第 6 块在计数器 1 中为 3，说明这个块被重复分配了 3 次，因此文件系统进行一致性检查时发现出了问题。选项 D 的第 8 块在计数器 2 中为 2，说明这个块被重复释放了 2 次，因此文件系统进行一致性检查时发现出了问题。选项 B 的某个块在计数器 1 中为 1，在计数器 2 为 0；而某个块在计数器 1 中为 0，在计数器 2 中为 1。这说明文件系统的一致性检查是正确的。

试题答案

(4) D

(5) B

试题 4

如图 1-9 (a) 所示是某一个时刻 J1、J2、J3、J4 四个作业在内存中的分配情况，若此时操作系统先为 J5 分配 5KB 空间，接着又为 J6 分配 10KB 空间，那么操作系统采用分区管理中的 (6) 算法，使得分配内存后的情况如图 1-9 (b) 所示。

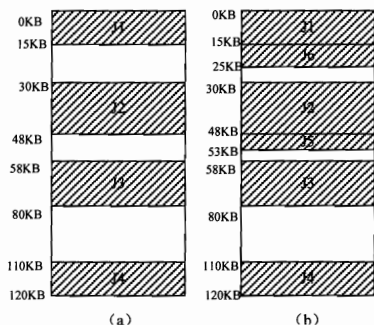


图 1-9 作业在内存中的分配情况

(6) A. 最先适应

B. 最佳适应

C. 最后适应

D. 最差适应

试题分析

根据图 1-9 (a)，为 J1、J2、J3、J4 分配了内存后，内存剩余空间为 J1 与 J2 之间的 15KB、J2 和 J3 之间的 10KB、J3 和 J4 之间的 30KB。当为任务 J5 分配 5KB 空间时，系统选择了分配在 J2 和 J3 之间的 10KB，然后为任务 J6 分配 10KB 空间时，系统选择了分配在 J1 和 J2 之间的 15KB，因此采用的是最佳适应算法。

试题答案

(6) B

试题 5

系统中有 R 类资源 m 个, 现有 n 个进程互斥使用。若每个进程对 R 资源的最大需求为 w , 那么当 m 、 n 、 w 取如表 1-5 所示的值时, 对于表 1-5 中的 a~e 这 5 种情况, (7) 可能会发生死锁。

表 1-5 进程和资源表

	a	b	c	d	e
m	2	2	2	4	4
n	1	2	2	3	3
w	2	1	2	2	3

(7) A. a 和 b

B. b 和 c

C. c 和 d

D. c 和 e

试题分析

本题考查死锁的基本知识, 给出系统资源情况, 判断死锁是否可能发生是一种典型的出题方式。

当给定资源情况, 如何进行死锁判断呢?

在此给出一个简单的公式: 进程数 \times (每个进程所需资源数-1) +1 得到当前条件下, 保证不产生死锁所需要的最少资源数。这个公式的原理为: 最坏的情况下, 系统的资源是平均分配, 这种情况下, 若每个进程分到了 (所需资源数-1) 个资源, 则再来一个资源, 即可完成进程操作。在这种情况下若系统还有 1 个资源, 则无论这个资源分配给谁, 都能保障其顺序执行完。当一个进程执行完以后, 将释放其占用的所有资源, 所以就不会产生死锁。利用该方法可以计算出 c 与 e 两种情况都有可能产生死锁。

试题答案

(7) D

试题 6

在一个单 CPU 的计算机系统中, 有 3 台不同的外部设备 R1、R2、R3 和 3 个进程 P1、P2、P3。系统 CPU 调度采用可剥夺式优先级的进程调度方案, 3 个进程的优先级、使用设备的先后顺序和占用设备时间如表 1-6 所示。

表 1-6 进程占用设备情况

进程	优先级	使用设备、CPU 的先后顺序和占用时间
P1	高	R1(20ms)→CPU(20ms)→R3(20ms)
P2	中	R3(40ms)→CPU(30ms)→R2(20ms)
P3	低	CPU(30ms)→R2(20ms)→CPU(20ms)

假设操作系统的开销忽略不计, 从 3 个进程同时投入运行到全部完成, CPU 的利用率约为 (8) %; R3 的利用率约为 (9) % (设备的利用率指该设备的使用时间与进程组全部完成所占用时间的比率)。

- (8) A.66.7 B.75 C.83.3 D.91.7
 (9) A.66 B.50 C.33 D.17

试题分析

根据表 1-6 画出系统的时空图, 如图 1-10 所示。

从图 1-10 中可以看出, P1 从投入运行到完成需要 60ms, P2 运行时间为 90ms, P3 由于等待资源, 运行时间延长为 120ms。CPU 在 80ms~100ms 共 20ms 时间内没有利用, 所以利用率为 $100/120=83.3\%$, 同样计算得 R3 的利用率为 $60/120=50\%$ 。

试题答案

- (8) C (9) B

试题 7

在如图 1-11 所示的树形文件系统中, 方框表示目录, 圆圈表示文件, “/” 表示路径中的分隔符, “/” 在路径之首时表示根目录。

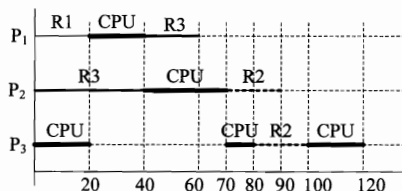


图 1-10 系统时空图

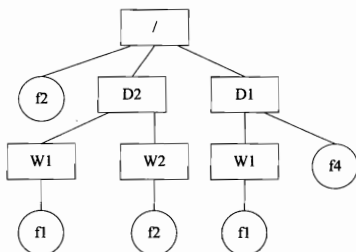


图 1-11 树形文件系统

假设当前目录是 D2, 进程 A 以如下两种方式打开文件 f2:

- 方式① `fd1=open("_(10)_/f2 " ,O_RDONLY);`
 方式② `fd1=open(" /D2/W2/f2 " , O_RDONLY);`

其中, 方式①的工作效率比方式②的工作效率高, 因为采用方式①, 文件系统是从_(11)_____。

- (10) A. /D2/W2 B. D2/W2 C. W2 D. /W2
 (11) A. 根目录开始查找文件 f2, 系统查找时间少, 读取 f2 文件次数不变
 B. 当前路径开始查找文件 f2, 系统查找时间少, 读取 f2 文件次数少
 C. 根目录开始查找文件 f2, 系统查找时间少, 读取 f2 文件次数少
 D. 当前路径开始查找文件 f2, 系统查找时间少, 读取 f2 文件次数不变

试题分析

在树形目录结构中, 树的根节点为根目录, 数据文件作为树叶, 其他所有目录均作为树的节点。在树形目录结构中, 从根目录到任何数据文件之间, 只有一条唯一的通路。从树根开始, 把全部目录文件名与数据文件名依次用 “/” 连接起来, 构成该数据文件的路径名, 且每个数据文件的路径名是唯一的。这样可以解决文件重名问题,

所以,对于第(10)空,虽然数据文件名均为 f2,但不一定是相同的文件。从树根开始的路径名为绝对路径名,如果文件系统没有很多级时,使用不是很方便,所以引入相对路径名,即从当前目录开始,逐级通过中间的目录文件,最后到达所要访问的数据文件。同样,从当前目录开始,采用相对路径名,较之采用绝对路径名,可以减少系统访问目录文件的次数,但是访问文件 f2 的次数是不变的。

试题答案

(10) C

(11) D

试题 8

虚拟内存是基于程序的局部性原理而设计的。下面关于局部性原理的描述正确的是____(12)____。

(12) A. 程序代码顺序执行

B. 程序按照非一致性方式访问内存

C. 程序连续地访问许多变量

D. 程序在一段时间内访问相对小的一段地址空间

试题分析

程序的局部性原理,即程序的地址访问流很强的时序相关性,未来的访问模式与最近已发生的访问模式相似。虚拟内存操作是基于程序执行的局部性原理,程序的局部性包括时间局部性和空间局部性,其中空间局部性是指程序在一段时间内访问相对小的一段地址空间。

试题答案

(12) D

试题 9

____(13)____操作需要特权指令执行。

(13) A. 读取当前时钟

B. 清除一块内存

C. 关闭中断

D. 从用户态切换到管态

试题分析

多任务的计算机系统中,特权指令主要用于系统资源的分配和管理,包括改变系统的工作方式、修改虚拟存储器管理的段表和页表、I/O 指令、设置时钟、设置控制寄存器和关闭中断等。简言之,不允许用用户程序中直接使用的指令称为“特权指令”。这里,A、B、D 选项可以由用户程序直接使用,而 C 是不可以直接使用的。

试题答案

(13) C

试题 10

关于分页式虚拟存储器的论述,正确的是____(14)____。

- (14) A. 根据程序的模块性, 确定页面大小
B. 可以将程序放置在页面内的任意位置
C. 可以从逻辑上极大地扩充内存容量, 并且使内存分配方便、利用率高
D. 将正在运行的程序全部装入内存

试题分析

由于内存的大小总是有限的, 如果都采用“实存管理”, 那么大于总物理内存的作业就无法运行。为了解决这一问题, 可行的方法就是用外存来换取内存, 这也就是虚拟存储系统。它通过将运行进程访问的地址(逻辑地址、虚拟地址)与主存的物理地址(实地址)分开, 从而就使得提供大于物理地址的逻辑地址空间成为可能。而建立虚拟地址和实地址之间的对应关系、实现转换的工作就称为“虚存管理”。设置虚拟存储器的目的就是要从逻辑上极大地扩充内存容量, 并且使内存分配方便、利用率高。

虚拟存储可以分为分区式、分页式、分段式、段页式等方法。分页的基本思想是把程序的逻辑空间和内存的物理空间按照同样的大小划分成若干页面, 以页面为单位进行分配。在页式存储管理中, 系统中虚地址是一个有序对(页号, 位移)。系统为每一个进程建立一个页表, 其内容包括进程的逻辑页号与物理页号的对应关系、状态等。

试题答案

(14) C

试题 11

通常将“C:\Windows\myprogram.exe”文件设置成只读和隐藏属性, 以便控制用户对该文件的访问, 这一级安全管理称为____(15)____安全管理。

- (15) A. 文件级 B. 目录级 C. 用户级 D. 系统级

试题分析

系统级安全管理的主要任务是不允许未经核准的用户进入系统。

用户级安全管理是为了给用户分配文件访问权而设计的, 包括对所有用户进行分类、为指定用户分配文件访问权。

目录级安全管理是为保护系统中的各种目录而设计的, 它与用户权限无关。为保证目录的安全, 规定只有系统核心才具有写目录的权利。通常, 系统是分别为用户和目录独立地指定权限的。

文件级安全管理是通过系统管理员或文件主对文件属性的设置, 来控制用户对文件访问。

试题答案

(15) A

试题 12

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 (16) 个字中描述；系统应该将 (17)。

- (16) A.128 B.129 C.130 D.131
- (17) 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”

试题分析

因为物理块编号是从 0 开始的，所以 4195 号物理块其实就是第 4196 块。因为字长为 32 位，也就是说，每个字可以记录 32 个物理块的使用情况。4196/32=131.125，所以，4195 号物理块应该在第 131 个字中（字的编号也是从 0 开始计数）。那么，具体在第 131 个字的哪一位呢？到第 130 个字为止，共保存了 131×32=4192 个物理块（0~4191），所以，第 4195 块应该在第 131 个字的第 3 位记录（要注意：0 是最开始的位）。因为系统已经将 4195 号物理块分配给某文件，所以其对应的位要置 1。

试题答案

- (16) D (17) B

试题 13

某系统进程的状态包括运行状态、活跃就绪状态、静止就绪状态、活跃阻塞状态和静止阻塞状态。针对图 1-12 所示的进程状态模型，为了确保进程调度的正常工作，(a)、(b) 和 (c) 的状态分别为 (18)。

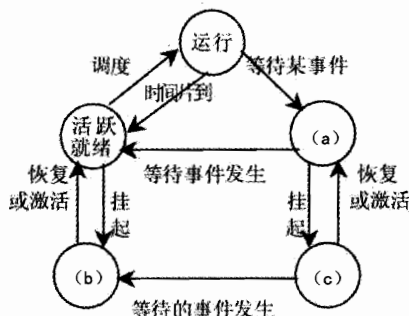


图 1-12 进程状态模型

- (18) A. 静止就绪、静止阻塞和活跃阻塞 B. 静止就绪、活跃阻塞和静止阻塞
- C. 活跃阻塞、静止就绪和静止阻塞 D. 活跃阻塞、静止阻塞和静止就绪

试题分析

请参看“1.1.3 进程管理”的图 1-3。

试题答案

(18) D

试题 14

如果一个索引式文件的索引节点有 10 个直接块, 1 个一级间接块, 1 个二级间接块, 1 个三级间接块。假设每个数据块的大小是 512B, 一个索引指针占用 4B。假设索引节点已经在内存中, 那么访问该文件偏移地址在 6000B 的数据需要再访问 (19) 次磁盘。

(19) A.1 B.2 C.3 D.4

试题分析

因为每个数据块的大小是 512B, 且前 10 块可以直接寻址, 得出 1~5120B 范围内可以直接寻址。对于间接索引块, 一个索引指针占 4B, 则一个索引块可以映射 $512/4=128$ 个数据块, 因为每个数据块的大小是 512B, 合计 64KB。6000B-5120B=880B<64KB, 所以只需一次映射就够了。因此, 第 1 次取索引指针, 第 2 次读数据, 一共需要两次访问。

试题答案

(19) B

试题 15

在操作系统的虚拟内存管理中, 内存地址由页目录号、页号和页内偏移 3 个部分组成。如果页目录号占 10 位、页号占 10 位、页内偏移占 12 位, 那么 (20)。

- (20) A. 页大小是 1K, 一个页目录最多 4K 页
B. 页大小是 2K, 一个页目录最多 2K 页
C. 页大小是 2K, 一个页目录最多 1K 页
D. 页大小是 4K, 一个页目录最多 1K 页

试题分析

已知内存地址由页目录号、页号和页内偏移 3 个部分组成, 页目录号占 10 位, 所以一个页目录最多 $2^{10}=1\text{K}$ 页。因为页内偏移占 12 位, 即页的大小为 $2^{12}=4\text{K}$ 。

试题答案

(20) D

1.3 实战练习题

- 计算机系统中硬件层之上的软件通常按照 3 层来划分, 如图 1-13 所示, 图中①

②③分别表示 (1)。



图 1-13 计算机软硬件结构图

- (1) A. 操作系统、应用软件和其他系统软件
 B. 操作系统、其他系统软件和应用软件
 C. 其他系统软件、操作系统和应用软件
 D. 应用软件、其他系统软件和操作系统

• 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, (2)。

- (2) 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

• 在虚拟存储器中, 辅存的编址方式是 (3)。

- (3) A. 按信息编址 B. 按字编址 C. 按字节编址 D. 按位编址

• 若某航空公司的飞机订票系统有 n 个订票终端, 系统为每个订票终端创建一个售票终端的进程。假设 P_i ($i=1,2,\dots,n$) 表示售票终端的进程, H_j ($j=1,2,\dots,m$) 表示公共数据区, 分别存放各个航班的现存票数, Temp 为工作单元。系统初始化时将信号量 S 赋值为 (4)。 P_i 进程的工作流程如图 1-14 所示, a、b 和 c 处将执行 P 操作和 V 操作, 则图 1-14 中 a、b 和 c 应填入 (5)。

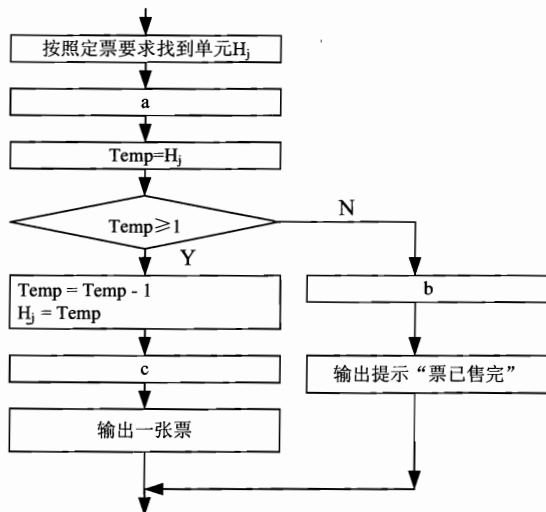


图 1-14 进程的工作流程

- (4) A. 0 B. 1 C. 2 D. 3
- (5) 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)

● 操作系统通常将 I/O 软件分成 4 个层次：用户应用层软件、中断处理程序、独立于设备的软件和设备驱动程序，分层的主要目的是 (6)。

- (6) A. 提高处理速度 B. 减少系统占用的空间
- C. 便于即插即用 D. 便于系统修改、扩充和移植

● 某计算机系统有一个 CPU、一台扫描仪和一台打印机。现有 3 个图像任务，每个任务有 3 个程序段：扫描 S_i ，图像处理 C_i 和打印 P_i ($i=1,2,3$)。如图 1-15 所示为 3 个任务各程序段并发执行的前驱图，其中，(7) 可并行执行，(8) 的直接制约，(9) 的间接制约。

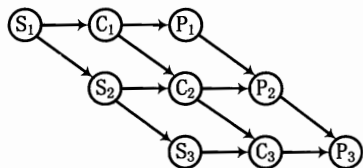


图 1-15 前趋图

- (7) 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. “ $S_1S_2S_3$ ”, “ $C_1C_2C_3$ ”, “ $P_1P_2P_3$ ”
- (8) 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
- (9) 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

● 若操作系统把一条命令的执行结果输出给下一条命令，作为它的输入并加以处理，这种机制称为 (10)。

- (10) A. 链接 B. 管道 C. 输入重定向 D. 输出重定向

● 假设磁盘上每个磁道划分成 9 个物理块，每块存放 1 个逻辑记录。逻辑记录 R_0, R_1, \dots, R_8 存放在同一个磁道上，记录的安排顺序如表 1-7 所示。

表 1-7 记录的安排顺序

物理块	0	1	2	3	4	5	6	7	8
逻辑记录	R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8

假定磁盘旋转一圈的时间为 27ms，磁头当前处在 R_0 的开始处。若系统顺序处理

这些记录,使用单缓冲区,每个记录处理时间为 3ms,则处理这 9 个记录的最长时间为 (11);若对信息存储进行优化分布后,处理 9 个记录的最少时间为 (12)。

(11) A. 243ms B. 246ms C. 254ms D. 280ms

(12) A. 30ms B. 36ms C. 54ms D. 60ms

● 某磁盘盘组共有 10 个盘面,每个盘面上有 100 个磁道,每个磁道有 32 个扇区,假定物理块的大小为 2 个扇区,分配以物理块为单位。若使用位示图(Bitmap)管理磁盘空间,则位图需要占用 (13) 字节空间。若采用空白文件管理磁盘空间,且空白文件目录的每个表项占用 5 个字节,则当空白文件数目大于 (14) 时,空白文件目录占用的字节数大于位图占用的字节数。

(13) A. 32000 B. 3200 C. 2000 D. 1600

(14) A. 400 B. 360 C. 320 D. 160

● 微内核的操作系统(OS)结构如图 1-16 所示,图中①和②分别工作在 (15) 方式下,与传统的 OS 结构模式相比,采用微内核的 OS 结构模式的优点是提高了系统的灵活性、可扩充性, (16)。

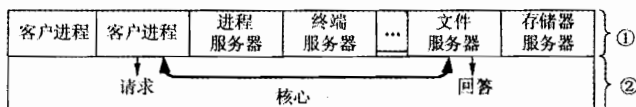


图 1-16 操作系统的结构

(15) A. 核心态和用户态 B. 用户态和核心态

C. 用户态和用户态 D. 核心态和核心态

(16) A. 并增强了可靠性,可运行于分布式系统中

B. 并增强了可靠性,但不适用于分布式系统

C. 但降低了可靠性,可运行于分布式系统中

D. 但降低了可靠性,不适用于分布式系统

● 进程 P1、P2、P3、P4、P5 的前趋图如图 1-17 所示。

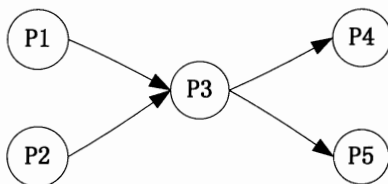


图 1-17 前趋图

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 1-18 中 a 和 b 应分别填写 (17),c 和 d 应分别填写 (18),e 和 f 应分别填写 (19)。

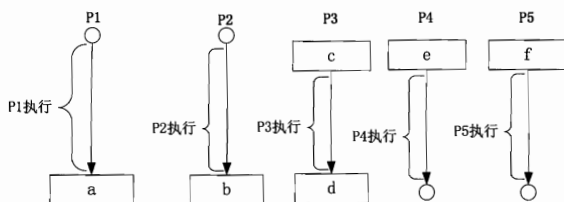


图 1-18 PV 操作控制过程

- (17) A. P (S1) 和 P (S2) B. P (S1) 和 V (S2)
 C. V (S1) 和 V (S2) D. V (S1) 和 P (S2)
- (18) 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)
- (19) A. P (S3) 和 P (S4) B. P (S3) 和 V (S4)
 C. V (S3) 和 V (S4) D. V (S3) 和 P (S4)

● 若操作系统文件管理程序正在将修改后的 (20) 文件写回磁盘时系统发生崩溃, 对系统的影响相对较大。

- (20) A. 用户数据 B. 用户程序 C. 系统目录 D. 空闲块管理

● 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 (21) 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 (22) 次缺页中断。

- (21) A. 50 B. 100 C. 5000 D. 10000
 (22) A. 50 B. 100 C. 5000 D. 10000

● 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 (23)。

- (23) A. 操作控制命令 B. 系统调用
 C. 菜单 D. 窗口

1.4 练习题解析

试题 1 分析

本题考查操作系统的基本概念，请参看“1.1.2 操作系统概论”的图 1-2。

试题 1 答案

(1) B

试题 2 分析

微内核操作系统结构是 20 世纪 80 年代后期发展起来的，其基本思想是将操作系统中最基本的部分放入内核中，而把操作系统的绝大部分功能都放在微内核外面的一组服务器中实现。这样使得操作系统内核变得非常小，自然提高了系统的可扩展性，增强了系统的可靠性和可移植性，同时微内核操作系统提供了对分布式系统的支持，融入了面向对象技术。虽然微内核操作系统具有诸多优点，但它并非完美无缺，在运行效率方面它就不如以前传统的操作系统。

从上述特点来看，本题应选 A。

试题 2 答案

(2) A

试题 3 分析

虚拟存储器必须建立在主存—辅存结构上，但一般的主存—辅存系统并不一定是虚拟存储器，虚拟存储器与一般的主存—辅存系统的本质区别如下：

- 虚拟存储器允许人们使用比主存容量大得多的地址空间来访问主存，非虚拟存储器最多只允许人们使用主存的整个空间，一般只允许使用操作系统分配的主存中的某一部分空间。
- 虚拟存储器每次访问主存时必须进行虚、实地址的变换，而非虚拟存储系统则不必变换。

虚拟存储技术实际上是将编写程序时所用的虚拟地址（逻辑地址）转换成较小的物理地址。在程序运行时，随时进行这种转换。为了便于主存与辅存之间信息的交换，虚拟存储器一般采用二维或三维的复合地址格式。采用二维地址格式时，将整个存储器划分为若干页（或段），每个页（或段）又包括若干存储单元。采用三维地址格式时，将整个存储空间分为若干段，每段分为若干页，每页又包括若干存储单元。根据地址格式不同，在虚拟存储系统中，基本信息传送单位可采用段、页或段页等几种不同的方式。

试题 3 答案

(3) A

试题 4~5 分析

因为该航空公司的飞机订票系统有 n 个订票终端，多个客户可能在不同的终端同

时订购某一航班的票, 导致售票出错。例如, 当某航班只剩下一张票时, 有 3 个客户在 3 个不同的终端订购, 则 3 个终端可能同时售出这张票, 而这是不允许的, 因此, 必须要设置一个信号量 S , 用来表示某航班当前是否有客户在订票或者待订票的人数。该信号量的初值为 1, 表示同时只能有一个客户订购某个航班的票。

当有客户在某个终端订购某航班的票时, 首先要执行 $P(S)$ 操作, 使其他客户等待。当完成订票或者票已售完时, 再执行 $V(S)$ 操作。

试题 4~5 答案

(4) B

(5) A

试题 6 分析

操作系统设备管理功能的内部结构设计一般是基于分层的思想, 因此, 通常将 I/O 软件分为用户应用层软件、中断处理程序、独立于设备的软件和设备驱动 4 个层次。采用分层思想的主要目的是便于系统修改、扩充和移植。

试题 6 答案

(6) D

试题 7~9 分析

本题考查前趋图的基础知识, 请参看“1.1.3 进程管理”中前趋图相关内容。

在图 1-15 中, 当 S_1 执行完毕后, 计算 C_1 与扫描 S_2 可并行执行; C_1 与 S_2 执行完后, 打印 P_1 、计算 C_2 与扫描 S_3 可并行执行; P_1 、 C_2 与 S_3 执行完毕后, 打印 P_2 与计算 C_3 可并行执行。

根据题意, 系统中有 3 个任务, 每个任务有 3 个程序段, 从前趋图中可以看出, 系统要先进行扫描 S_i , 然后再进行图像处理 C_i , 最后进行打印 P_i , 所以, C_1 和 P_1 受到 S_1 直接制约、 C_2 和 P_2 受到 S_2 的直接制约、 C_3 和 P_3 受到 S_3 的直接制约。

系统中有一台扫描仪, 因此 S_2 和 S_3 不能运行是受到了 S_1 的间接制约。如果系统中有 3 台扫描仪, 那么 S_2 和 S_1 能运行; 同理, C_2 和 C_3 受到 C_1 的直接制约、 P_2 和 P_3 受到 P_1 的间接制约。

试题 7~9 答案

(7) A

(8) C

(9) B

试题 10 分析

若操作系统把一条命令的执行结果输出给下一条命令, 作为它的输入并加以处理, 这种机制称为管道。

管道通信是一种共享文件模式, 它基于文件系统, 连接于两个通信进程之间, 以先进先出的方式实现消息的单向传送。管道是一个特殊文件, 在内核中通过文件描述符表示。一个管道总是连接两个命令, 将左边命令的标准输出与右边命令的标准输入相连, 于是左边命令的输出结果就直接成了右边命令的输入。

试题 10 答案

(10) B

试题 11~12 分析

因为系统使用的是单缓冲区，且顺序处理 9 个记录，每个记录处理时间为 3ms，加上读写时间，总的时间就超过 3ms 了。而磁盘旋转一圈的时间为 27ms，也就是说，当系统读取第 0 个记录后，正在处理的过程中，磁盘已经旋过了第 1 个记录。那么，要读取第 1 个记录，就需要磁盘再次旋转 to 第 1 个记录（即磁盘旋转 1 圈后， $27+3=30\text{ms}$ ）。同理，要读取第 2 个记录时，也需要等 30ms。这样，要读取后面 8 个记录，需要 $8 \times 30=240\text{ms}$ ，同时加上处理第 0 个记录的时间（3ms）和处理第 8 个记录的时间（3ms），共需 246ms。

要想节约时间，可以把记录错开存放，如表 1-8 所示。

表 1-8 错开存放的记录

物理块	0	1	2	3	4	5	6	7	8
逻辑记录	R0	R5	R1	R6	R2	R7	R3	R8	R4

这样，就可以在磁盘旋转 2 圈内完成所有记录的处理，时间为 54ms。要注意的是，最后处理的记录 R8 不是最后一个磁盘块，所以不需要旋转到最后 1 个物理块。也就是说，第 2 圈的旋转时间只需要 24ms 就到达 R8 了。但是，因为要加上 R8 的处理时间 3ms，所以，总时间仍然为 54ms。

试题 11~12 答案

(11) B

(12) C

试题 13~14 分析

已知磁盘盘组共有 10 个盘面，每个盘面上有 100 个磁道，每个磁道有 32 个扇区，则一共有 $10 \times 100 \times 32=32000$ 个扇区。试题又假定物理块的大小为 2 个扇区，分配以物理块为单位，即一共有 16000 个物理块。因此，位图所占的空间为 $16000/8=2000$ 字节。

若采用空白文件管理磁盘空间，且空白文件目录的每个表项占用 5 个字节， $2000/5=400$ ，因此，则当空白文件数目大于 400 时，空白文件目录占用的字节数大于位图占用的字节数。

试题 13~14 答案

(13) C

(14) A

试题 15~16 分析

现代操作系统大多拥有两种工作状态：核心态和用户态。我们使用的一般应用程序工作在用户态，而内核模块和最基本的操作系统核心工作在核心态。

微内核结构由一个非常简单的硬件抽象层和一组比较关键的原语或系统调用组成，这些原语仅仅包括了建立一个系统必需的几个部分，如线程管理，地址空间和进程间通信等。微内核的目标是将系统服务的实现和系统的基本操作规则分离开来。例如，进程的输入/输出锁定服务可以由运行在微内核之外的一个服务组件来提供。这些非常模块化的用户态服务用于完成操作系统中比较高级的操作，这样的设计使内核中

最核心的部分的设计更简单。一个服务组件的失效并不会导致整个系统的崩溃，内核需要做的，仅仅是重新启动这个组件，而不必影响其他的部分。

微内核技术的主要优点如下：

- 统一的接口，在用户态和核心态之间无须进程识别。
- 可伸缩性好，能适应硬件更新和应用变化。
- 可移植性好，所有与具体机器特征相关的代码，全部隔离在微内核中，如果操作系统要移植到不同的硬件平台上，只需修改微内核中极少代码即可。
- 实时性好，微内核可以方便地支持实时处理。
- 安全可靠性好，微内核将安全性作为系统内部特性来进行设计，对外仅使用少量应用编程接口。
- 支持分布式系统，支持多处理器的体系结构和高度并行的应用程序。
- 真正面向对象的操作系统。

由于操作系统核心常驻内存，而微内核结构精简了操作系统的核心功能，内核规模比较小，一些功能都移到了外存上，所以微内核结构十分适合嵌入式的专用系统，对于通用性较广的系统，将使 CPU 的通信开销增大，从而影响到计算机的运行速度。

试题 15~16 答案

(15) B

(16) A

试题 17~19 分析

根据题意，进程 P3 等待 P1、P2 的结果，因此，当 P1、P2 执行完毕需要使用 V 操作来通知 P3，即 a 处填 V(S1)，b 处填 V(S2)。进程 P3 的执行需要测试 P1、P2 有没有消息，应该在 c 处填 P(S1)、P(S2)；当 P3 执行完毕后需要使用 V 操作通知 P4 和 P5，即在 d 处填 V(S3)、V(S4)。进程 P4 和 P5 的执行需要测试 P3 有没有消息，故应该在 e 处填 P(S3)，在 f 处填 P(S4)。

试题 17~19 答案

(17) C

(18) A

(19) A

试题 20 分析

本题考查操作系统基本概念。操作系统为了实现“按名存取”，必须为每个文件设置用于描述和控制文件的数据结构，专门用于文件的检索，因此至少要包括文件名和存放文件的物理地址，该数据结构称为文件控制块（File Control Block, FCB），文件控制块的有序集合称为文件目录，或称系统目录文件。若操作系统正在将修改后的系统目录文件写回磁盘时系统发生崩溃，则对系统的影响相对较大。

试题 20 答案

(20) C

试题 21~22 分析

本题考查页面置换算法——LRU。

从题干可知，作业共有 4 个页面的主存空间，其中一个已被程序本身占用，所以在读取变量时可用的页面数只有 3 个。每个页面可存放 200 个整数变量，程序中 A 数

组共有 $100 \times 100 = 10000$ 个变量。按行存放时，每个页面调入的 200 个变量刚好是程序处理的 200 个变量，所以缺页次数为 $10000/200 = 50$ 。而按列存放时，虽然每个页面调取数据时，同样也读入了 200 个变量，但这 200 个变量中，只有 2 个是近期需要访问的（如第 1 个页面调入的是 $A[*,1]$ 与 $A[*,2]$ ，但程序近期需要访问的变量只有 $A[1,1]$ 和 $A[1,2]$ ），所以缺页次数为 $10000/2 = 5000$ 。

试题 21~22 答案

(21) A

(22) C

试题 23 分析

本题考查操作系统的基本概念。

操作系统是管理计算机硬件与软件资源的程序，同时也是硬件与用户之间的接口。操作系统既提供了与用户交互的接口，也提供了与应用程序交互的接口。用户可以通过菜单、命令、窗口与操作系统进行交互，而应用程序可以通过系统调用（如调用系统 API）来与操作系统交互。

试题 23 答案

(23) B

2

第2章

数据库系统

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

- 数据库管理系统的类型、结构和性能评价。
- 常用的关系型数据库管理系统。
- 数据库模式。
- 数据库规范化。
- 分布式数据库系统、并行数据库系统。
- 数据仓库与数据挖掘技术。
- 数据库工程。
- 备份、恢复。

2.1 考点突破

从历年的考试情况来看，本章的考点比较分散，数据库的多个知识点都会涉及，但考查的难度都不大。

2.1.1 历年考试情况分析

在历年的考试试题中，有关数据库系统知识的试题如表 2-1 所示。

表 2-1 数据库系统知识试题分布表

题号	2009.11	2010.11	2011.11
5	数据库设计	数据库设计	实体联系、实体转关系模式
6	完整性约束、触发器	完整性约束	
7	候选关键字、模式分解	SQL 语言	
8		关系代数	关系代数
35			数据库与数据仓库

续表

题号	2009.11	2010.11	2011.11
36			
40	分布式数据库		

按照知识点进行总结和归类的试题分布情况如表 2-2 所示。

表 2-2 数据库系统知识归类表

知识点	2009.11	2010.11	2011.11
数据库模式	1	1	2
E-R 模型	0	0	1
完整性约束	1	1	0
关系代数	0	1	1
规范化理论	2	0	0
SQL 语言	0	1	0
分布式数据库	1	0	0
数据仓库与数据挖掘	0	0	2
合计	5	4	6

从表 2-2 中可以看出，数据库系统知识方面的内容在历年的考试中分值稳定，平均维持在 5 分。所占分数比例的趋势如图 2-1 所示。



图 2-1 数据库系统知识历年试题比例趋势图

2.1.2 数据库模式

数据库系统是指在计算机系统中引入数据库后的系统，一般由数据库、数据库管理系统（及其开发工具）、应用系统、数据库管理员和用户构成。

数据库系统的结构可以有多种不同的层次或不同的角度，其中最为著名的是美国 ANSI/ SPARC 数据库系统研究组 1975 年提出的三级划分法。下面将以该划分法为基础详细说明三级模式与二级映射。

1. 三级模式

美国 ANSI/ SPARC 数据库系统研究组提出来的三级模式如图 2-2 所示。从图中可以看出数据库系统由外模式、概念模式和内模式三级构成。

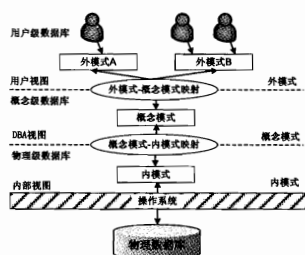


图 2-2 数据库系统结构层次图

(1) 外模式，对应于用户级数据库

外模式（子模式、用户模式）用以描述用户看到或使用的那部分数据的逻辑结构，用户根据外模式用数据操作语句或应用程序去操作数据库中的数据。外模式主要描述组成用户视图的各个记录的组成、相互关系、数据项的特征、数据的安全性和完整性约束条件。

外模式是数据库用户（包括程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。一个数据库可以有多个外模式。一个应用程序只能使用一个外模式。

(2) 概念模式，对应于概念级数据库

概念模式（模式、逻辑模式）用以描述整个数据库中数据库的逻辑结构，描述现实世界中的实体及其性质与联系，定义记录、数据项、数据的完整性约束条件及记录之间的联系，是数据项值的框架。

数据库系统概念模式通常还包含访问控制、保密定义、完整性检查等方面的内容，以及概念/物理之间的映射。

概念模式是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。一个数据库只有一个概念模式。

(3) 内模式，对应于物理级数据库

内模式是整个数据库的最低层表示，不同于物理层，它假设外存是一个无限的线性地址空间。内模式定义的是存储记录的类型、存储域的表示，以及存储记录的物理顺序，指引元、索引和存储路径等数据的存储组织。

内模式是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。一个数据库只有一个内模式。

三级模式的关系：

- 模式是数据库的中心与关键。
- 内模式依赖于模式，独立于外模式和存储设备。
- 外模式面向具体的应用，独立于内模式和存储设备。
- 应用程序依赖于外模式，独立于模式和内模式。

2. 两级独立性

数据库系统两级独立性是指物理独立性和逻辑独立性。三个抽象级间通过两级映

射（外模式/模式映射，模式/内模式映射）进行相互转换，使得数据库的三级形成一个统一的整体。

（1）物理独立性

物理独立性是指用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的。当数据的物理存储改变时，应用程序不需要改变。

物理独立性存在于概念模式和内模式之间的映射转换，说明物理组织发生变化时应用程序的独立程度。

（2）逻辑独立性

逻辑独立性是指用户的应用程序与数据库中的逻辑结构是相互独立的。当数据的逻辑结构改变时，应用程序不需要改变。

逻辑独立性存在于外模式和概念模式之间的映射转换，说明概念模式发生变化时应用程序的独立程度。逻辑独立性比物理独立性更难实现。

2.1.3 E-R 模型

数据库系统是对现实世界中数据的一种抽象，首先我们将通过概念模型将现实世界抽象成为信息世界，然后再抽象成为基本数据模型。最常使用的概念模型就是 E-R 模型。

1. E-R 模型相关基本概念

- 实体：客观存在并可相互区别的事物，可以是具体的人、事、物，也可以是抽象的概念或联系。
- 属性：实体所具有的某一特性称为属性，通常一个实体可以由多个属性来描述。
- 域：属性的取值范围。
- 实体型：具有相同属性的实体必然具有共同的特性和性质。用实体名及其属性名集合来抽象和描述同类实体，就称为实体型。
- 实体集：同型实体的集合称为实体集。
- 联系：实体内部的联系通常是指组成实体的各属性间的关系。

2. E-R 模型实体联系类型

（1）一对一联系（1:1）

对于实体集 A 中的每一个实体，实体集 B 中至多有一个实体与之联系。例如，一个班级只有一个班主任，一个班主任也只在在一个班级中任职。

（2）一对多联系（1:n）：对于实体集 A 中的每一个实体，实体集 B 中有 n ($n>0$) 个实体与之联系。反之，实体集 B 中的每一个实体，实体集 A 中至多只有一个实体与之联系。例如，一个班级中有许多学生，而每个学生只在在一个班级中学习。

（3）多对多联系（m:n）


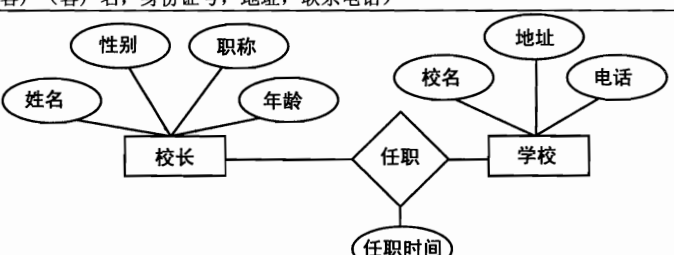
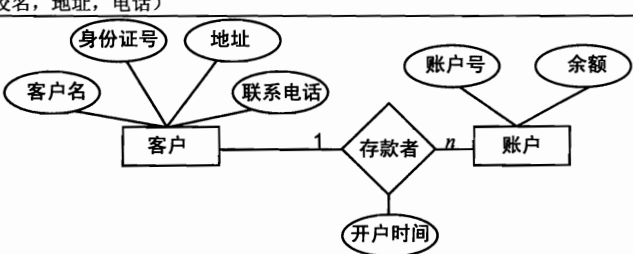
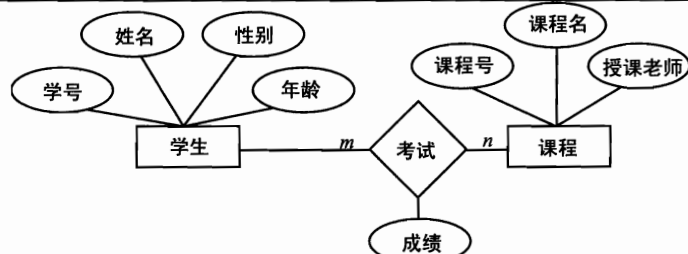
对于实体集 A 中的每一个实体，实体集 B 中有 n ($n>0$) 个实体与之联系。反之，

实体集 B 中的每一个实体, 实体集 A 中有 m ($m>0$) 个实体与之联系。一门课程同时有许多学生选修, 而一个学生也可以选修多门课程。

3. E-R 模型转关系模式

E-R 模型转关系模式有着严格的规则定义, 如表 2-3 所示。

表 2-3 E-R 模型与关系模型的转换

类 型	转换方法	示 例
实体类型	每个实体类型转换为一个关系模式	 <p>客户 (客户名, 身份证号, 地址, 联系电话)</p>
联系类型	1:1	 <p>校长 (姓名, 性别, 职称, 年龄, 校名, 任职时间) 学校 (校名, 地址, 电话)</p>
	1:n	 <p>客户 (客户名, 身份证号, 地址, 联系电话) 账户 (客户身份证号, 账户号, 余额, 开户时间)</p>
	m:n	 <p>学生 (学号, 姓名, 性别, 年龄) 课程 (课程号, 课程名, 授课老师) 考试 (课程号, 学号, 成绩)</p>

从表 2-3 中我们可以总结出来, 每一个实体类型都将生成一个关系模式, 而对于 1:1 和 1:n 联系类型而言无须生成新的关系模式, 只需并入到实体类型所对应的关系模式中即可。而对于 $m:n$ 的多对多联系类型则通常需要转换成一个关系模式。因此, 对于 3 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 4 个关系模式。

2.1.4 关系代数

从前面的历年试题分布结构分析我们可以得知, 关系代数是一个考点。本节将详细介绍相关内容。

关系代数的基本运算包括: 并、交、差、笛卡儿积、选择、投影、连接、除法运算。

1. 并 (Union, 用符号 \cup 表示)

计算两个表在集合理论上的联合。给出表 R 和 S (两者有相同元/列数), $R \cup S$ 的联合就是所有在 R 中有, 或 S 中有, 或在两个表中都有的记录集合。

2. 交 (Intersection, 用符号 \cap 表示)

计算两个表集合理论上的交集。给出表 R 和 S , $R \cap S$ 是同时在 R 和 S 中的记录的集合。我们同样要求 R 和 S 拥有相同的元/列数。

3. 差 (Difference, 用符号 $-$ 表示)

计算两个表的区别的集合。令 R 和 S 是拥有相同元/列的表。 $R-S$ 是在 R 中却不在 S 中的记录的集合。

4. 笛卡儿积 (PRODUCT, 用符号 \times 表示)

计算两个关系的笛卡儿乘积。令 R 为有 k_1 元的表, 令 S 为有 k_2 元的表。 $R \times S$ 是所有 k_1+k_2 元记录的集合, 其前 k_1 个元素来自 R 中的一条记录, 而后 k_2 个元素来自 S 中的一条记录。

5. 投影 (Project, 用符号 π 表示)

从一个关系里面抽取指明的属性(列)。令 R 为一个包含一个属性 x 的关系。 $\pi_{x(R)} = \{t(x) | t \in R\}$, 这里 $t(x)$ 表示记录 t 中的属性 x 的值。

6. 选择 (Select, 用符号 σ 表示)

从关系里面抽取出满足给定限制条件的记录。

7. 联接 (Join, 用符号 \bowtie 表示)

从两个关系的笛卡儿积中选取属性间满足一定条件的元组。

8. 除 (Division, 用符号 \div 表示)

设有关系 $R(X, Y)$ 与关系 $S(Z)$, 其中, X, Y, Z 为属性集合。假设 Y 和 Z 具有相同的属性个数, 且对应属性出自相同域。关系 $R(X, Y) \div S(Z)$ 所得的商关系是关系 R 在属性 X 上投影的一个子集, 该子集和 $S(Z)$ 的笛卡儿积必须包含在 $R(X, Y)$ 中, 记

为 $R \div S$ 。

例如，有关系 R 与关系 S，如图 2-3 所示。

关系 R	A	B	C	D
	a	b	c	d
	a	b	e	f
	c	a	c	d

关系 S	C	D
	c	d
	e	f

图 2-3 关系 R 与关系 S

则 $R \div S$ 的求解过程如下：

首先按除运算定义要求，确定 X、Y、Z 属性集合。Y 是关系 R 中的属性集合，Z 是 S 中全部属性的集合，所以很容易得出 $Z = \{C, D\}$ ，同时由于 $Y = Z$ ，所以 $Y = \{C, D\}$ ， $X = \{A, B\}$ 。因此 $R \div S$ 结果集包含属性：A 和 B。

然后将关系 R 的 A、B 元组（共有： $\langle a, b \rangle$ 与 $\langle c, a \rangle$ 两个元组）与关系 S 进行笛卡儿积运算，结果如图 2-4 所示。

A	B	C	D
a	b	c	d
a	b	e	f
c	a	c	d
c	a	e	f

图 2-4 $R \times S$

通过检查该表，可以发现元组 $\langle a, b \rangle$ 与 $S(Z)$ 的笛卡儿积被包含在 $R(X, Y)$ 中，而元组 $\langle c, a \rangle$ 与 $S(Z)$ 的笛卡儿积有一条记录未被包含在 $R(X, Y)$ 中，所以结果集中只有元组 $\langle a, b \rangle$ 。结果如图 2-5 所示。

R ÷ S	A	B
	a	b

图 2-5 $R \div S$

2.1.5 完整性约束

数据库的完整性是指数据的正确性和相容性。数据库是否具备完整性关系到数据库系统能否真实地反映现实世界，因此维护数据库的完整性是非常重要的。

1. 完整性约束条件

保证数据完整性的方法之一是设置完整性检查，即对数据库中的数据设置一些约束条件，这是数据的语义体现。完整性约束条件是指对数据库中数据本身的某些语法或语义限制、数据之间的逻辑约束，以及数据变化时应遵守的规则等。所有这些约束条件一般均以谓词逻辑形式表示，即以具有真假值的原子公式和命题连接词（并且、

或者、否定)所组成的逻辑公式表示。完整性约束条件的作用对象可以是关系、元组或属性3种。数据的完整性约束条件一般在关系模式中给出,并在运行时做检查,当不满足条件时立即向用户通报,以便采取措施。

数据库中数据的语法、语义限制与数据之间的逻辑约束称为静态约束,它反映了数据及其之间的固有逻辑特性,是最重要的一类完整性约束。静态约束包括静态属性级约束(对数据类型的约束、对数据格式的约束、对取值范围或取值集合的约束、对空值的约束、其他约束)、静态元组约束和静态关系约束(实体完整性约束、参照完整性约束、函数依赖约束、统计约束)。

数据库中的数据变化应遵守的规则称为数据动态约束,它反映了数据库状态变迁的约束。动态约束包括动态属性级约束(修改属性定义时的约束、修改属性值时的约束)、动态元组约束和动态关系约束。

完整性控制机制应该具有定义功能和检查功能,定义功能提供定义完整性约束条件的机制,检查功能检查用户发出的操作请求是否违背了完整性约束条件。如果发现用户的操作请求违背了约束条件,则采取一定的动作来保证数据的完整性。

2. 实体完整性

实体完整性要求主键中的任一属性不能为空,同时主键不能有重复值。之所以要保证实体完整性,主要是因为,在关系中,每个元组的区分是依据主键值的不同,若主键值取空值,则不能标明该元组的存在。例如,对于学生关系 $S(Sno, Sname, Ssex)$, 其主键为 Sno , 在插入某个元组时,就必须要求 Sno 不能为空。

3. 参照完整性

若基本关系 R 中含有与另一基本关系 S 的主键 PK 相对应的属性组 FK (FK 称为 R 的外键), 则参照完整性要求,对 R 中的每个元组在 FK 上的值必须是 S 中某个元组的 PK 值,或者为空值。参照完整性的合理性在于, R 中的外键只能对 S 中的主键引用,不能是 S 中主键没有的值。例如,对于学生关系 $S(Sno, Sname, Ssex)$ 和选课关系 $C(Sno, Cno, Grade)$ 两个关系, C 中的 Sno 是外键,它是 S 的主键,若 C 中出现了某个 S 中没有的 Sno ,即某个学生还没有注册,却已有了选课记录,这显然是不合理的。

在实际应用中,对于参照完整性,需要明确外键能否接受空值的问题,以及在被参照关系中删除元组的问题。针对不同的应用,可以有不同的删除方式:

(1) 级联删除

将参照关系中所有外键值与被参照关系中要删除元组主键值相同的元组一起删除。如果参照关系同时又是另一个关系的被参照关系,则这种删除操作会继续级联下去。

(2) 受限删除

这是一般 DBMS 默认的删除方式。仅当参照关系中没有任何元组的外键值与被参照关系中要删除元组的主键值相同时,系统才可以执行删除操作,否则拒绝执行删除操作。

(3) 置空删除

删除被参照关系的元组,并将参照关系中相应元组的外键值置为空值。

同样,还需要考虑在参照关系中插入元组的问题,一般可以采用以下两种方式:

(1) 受限插入

仅当被参照关系中存在相应的元组时,其主键值与参照关系插入元组的外键值相同时,系统才执行插入操作,否则拒绝此操作。

(2) 递归插入

首先向被参照关系中插入相应的元组,其主键值等于参照关系插入元组的外键值,然后向参照关系插入元组。

4. 用户定义的完整性

实体完整性和参照完整性适用于任何关系型 DBMS。除此之外,不同的数据库系统根据其应用环境的不同,往往还需要一些特殊的约束条件。用户定义的完整性就是针对某一具体数据库的约束条件,反映某一具体应用所涉及的数据必须满足的语义要求。

如果在一条语句执行完后立即检查,则称立即执行约束;如果在整个事务执行结束后再进行检查,则称延迟执行约束。完整性规则的五元组表示为 (D, O, A, C, P) ,其中, D 表示约束作用的数据对象, O 表示触发完整性检查的数据库操作, A 表示数据对象必须满足的断言或语义约束, C 表示选择 A 作用的数据对象值的谓词, P 表示违反完整性规则时触发的过程。

5. 触发器

触发器 (Trigger) 是在关系数据库管理系统中应用得比较多的一种完整性保护措施。触发器的功能一般比完整性约束要强得多。一般而言,在完整性约束功能中,当系统检查出数据中有违反完整性约束条件时,则仅给出必要提示以通知用户,仅此而已。而触发器的功能则不仅仅起提示作用,它还会引起系统内自动进行某些操作以消除违反完整性约束条件所引起的负面影响。触发器除了有完整性保护功能外,还有安全性保护功能。

2.1.6 规范化理论

本节的主要考点包括判断关系模式的主键、属于哪种范式、最小函数依赖集、模式分解等。

1. 函数依赖

函数依赖是数据库的一种约束,决定了关系模式属于哪种范式。为了理解方面,下面我们先介绍函数依赖的有关概念。

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 $X \rightarrow Y$ 。

在 $R(U)$ 中, 如果 $X \rightarrow Y$ (Y 不是 X 的真子集), 且 $Y \rightarrow X$ 不成立, $Y \rightarrow Z$, 则称 Z 对 X 传递函数依赖。

例如, 记录职工信息的结构如下:

职工工号 (EMP_NO)

职工姓名 (EMP_NMAE)

所在部门 (DEPT)。

我们说 EMP_NO 函数决定 EMP_NMAE 和 DEPT, 或者说 EMP_NMAE、DEPT 函数依赖于 EMP_NO, 记为: $EMP_NO \rightarrow EMP_NMAE$, $EMP_NO \rightarrow DEPT$ 。

2. 键

- 超键: 在关系模式中, 能唯一标识元组的属性集称为超键。
- 候选键: 如果一个属性集能唯一标识元组, 且又不含有多余属性。
- 主键: 关系模式中用户正在使用的候选键称为主键。
- 外键: 如果公共关键字在一个关系中是主关键字, 那么这个公共关键字被称为另一个关系的外键。

例如, 记录职工信息的结构如下:

职工工号 (EMP_NO)

职工身份证号 (EMP_CARDID)

职工姓名 (EMP_NMAE)

职工性别 (EMP_SEX)

所在部门编号 (DEPT_NO)。

在此关系中:

(1) EMP_NO 和 EMP_NMAE 的组合是超键

理由: EMP_NO 和 EMP_NMAE 的组合能唯一标识元组的属性。

补充说明: 其实只需要 EMP_NO 就可以唯一标识元组的属性, 所以 EMP_NO 是本关系的候选键, 也可以是本关系的主键。

(2) EMP_NO 和 EMP_CARDID 是候选键

补充说明: 一个关系的候选键可以有多个。

(3) EMP_NO 或 EMP_CARDID 可选作本关系的主键

补充说明: 一个关系的候选键有多个, 但主键只能有一个。通常在候选键中选一个作为主键。

(4) DEPT_NO 为本关系相对于部门关系的外键

理由: DEPT_NO 在本关系中不是主键, 而在部门关系中是主键, 所以它是本关系的外键。

补充说明: 外键的作用是进行连接操作。例如, 现在要查找某个职工所在的部门名称, 我们就需要用到外键来与部门关系进行连接, 连接之后可以得到职工所在部门的名称。

3. 图示法求候选键

求关系模式的候选键是进行范式界定的基础，也是系统分析员应该掌握的基本技能。使用候选键的定义“如果一个属性集能唯一标识元组，且又不含有多余属性。”来求解一个简单关系模式的候选键尚能应对，但面对复杂一些的关系模式，这种方法就不管用了。在此，引入一种求候选键的快捷方法，即图示法。

使用图示法求候选键，主要有3个步骤：

- 1) 将关系的函数依赖关系，用“有向图”的方式表示。
- 2) 找出入度为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节点，则该属性集即为关系模式的候选键。
- 3) 若入度为0的属性集不能遍历图中所有节点，则需要尝试性地将一些中间节点（既有入度，也有出度的节点）并到入度为0的属性集中，直至该集合能遍历所有节点，集合为候选键。

例如，给定关系 $R(A1, A2, A3, A4)$ 上的函数依赖集 $F=\{A1 \rightarrow A2, A3 \rightarrow A2, A2 \rightarrow A3, A2 \rightarrow A4\}$ ，现在要求 R 的候选关键字。

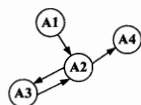


图 2-6 函数依赖集对应应有向图

第一步，需要针对函数依赖集画出有向图，如图 2-6 所示。

第二步，从该图中找出入度为0的节点，本图中入度为0的节点只有1个，即 $A1$ 。通过尝试，可以发现从 $A1$ 出发可以遍历有向图中的所有节点，所以本关系模式的候选键为 $A1$ 。

4. 范式

在数据库设计过程中，往往遇到数据冗余、修改异常、插入异常和删除异常等，为了设计一个好的数据库，人们定义了一些好的关系模式标准，称它们为规范的关系模式（简称范式，NF）。目前共定义了多个范式，分别为 1NF、2NF、3NF、BCNF、4NF 和 5NF。但实际应用中，一般只要达到 3NF。

如果有 $X \rightarrow U$ 在关系模式 $R(U)$ 上成立，并且不存在 X 的任一真子集 X' 使 $X' \rightarrow U$ 成立，那么称 X 是 R 的一个候选键。也就是 X 值唯一决定关系中的元组。

在 $R(U)$ 中，如果 $X \rightarrow Y$ ，并且对于 X 的任何一个真子集 X' ，都有 $X' \rightarrow Y$ 不成立，则称 Y 对 X 完全函数依赖。若 $X \rightarrow Y$ ，但 Y 不完全函数依赖于 X ，则称 Y 对 X 部分函数依赖。

在 $R(U)$ 中，如果 $X \rightarrow Y$ （ Y 不是 X 的真子集），且 $Y \rightarrow X$ 不成立， $Y \rightarrow Z$ ，则称 Z 对 X 传递函数依赖。

有了上述的函数依赖概念之后，我们再介绍范式的概念。

(1) 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，即每个分量都是不可再分的数据项，则称实体 E 是第一范式。本书中所引用的关系模式，一般都达到了第一范式的要求。

例如，如图 2-7 所示的教师职称情况关系表就不满足 1NF。

教师职称情况关系表		
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

图 2-7 教师职称情况关系表

原因在于该关系模式中的“高级职称人数”不是一个原子属性，若将其拆分为“教授”和“副教授”两个属性，则满足 1NF。满足 1NF 的关系模型会有许多重复值，并且增加了修改其数据时引起疏漏的可能性。为了消除这种数据冗余和避免更新数据的遗漏，我们需要更加规范的第二范式（2NF）。

(2) 第二范式（2NF）

当且仅当实体 E 是第一范式（1NF），且每一个非键属性完全依赖主键（没有不完全依赖）时，则称实体 E 是第二范式。

非键属性完全依赖又称部分函数依赖。

例如，有选课关系模式 SC(Sno, Cno, Grade, Credit)，其中，(Sno, Cno)→Grade、Cno→Credit。

由此可知 SC 的主键为(Sno, Cno)。这样(Sno, Cno)→Credit 就被称为 Credit 对主键(Sno, Cno)的部分函数依赖，即 Credit 属性只需要主键的一部分 Cno 即可确定。所以本关系模式不符合 2NF，只是 1NF。若要将该关系模式转化为 2NF，可以将关系模式拆分为 SC1(Sno, Cno, Grade)和 SC2(Cno, Credit)。

拆分前的 SC 与拆分后的 SC1 与 SC2 如图 2-8 所示，从该分解可以发现 Credit 在 SC 中有大量数据重复，而分解之后，解决了该问题。

SC 关系			
Sno	Cno	Grade	Credit
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
...

关系 SC1		
Sno	Cno	Grade
S01	C01	75
S02	C01	92
S03	C01	87
S04	C01	55
S01	C02	87
S02	C02	95
S01	C03	94
...

关系 SC2	
Cno	Credit
C01	4
C02	2
C03	5
...	...

图 2-8 关系 SC 及由此拆分的 SC1 与 SC2

(3) 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。

例如，学生关系 Student (Sno, Sname, Dno, Dname, Location) 各属性分别代表学号、姓名、所在系、系名称、系地址。其数据如表 2-9 所示，从关系模式中各属性之间的联系可以判断出本关系模式的函数依赖有： $Sno \rightarrow Sname$ ， $Sno \rightarrow Dno$ ， $Sno \rightarrow Dname$ ， $Sno \rightarrow Location$ ， $Dno \rightarrow Dname$ ， $Dno \rightarrow Location$ 。显然，Sno 为主键。在函数依赖中有： $Sno \rightarrow Dno \rightarrow Dname$ 与 $Sno \rightarrow Dno \rightarrow Location$ 。这便是传递函数依赖。由于 Dname 与 Location 为非主属性，同时传递依赖于码，所以本关系模式不满足 3NF。若要满足 3NF，需要将关系模式拆分为：Student (Sno, Sname, Dno) 和 Department (Dno, Dname, Location)。

关系 Student				
Sno	Sname	Dno	Dname	Location
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
...

图 2-9 关系 Student

(4) BCNF

如果关系模型 $R \in 1NF$ ，且 R 中每一个函数依赖关系中的决定因素都包含码，则 R 是满足 BC 范式的关系，记作 $R \in BCNF$ 。

按此定义，消除了任何属性对码的传递依赖与部分依赖即为 BCNF。但要判断任何属性是否对码存在传递依赖并不是一件容易的事情。

例如，有关系模式 P(C, S, T, R)，根据语义有函数依赖 $F = \{C \rightarrow T, S, T \rightarrow R, T, R \rightarrow C\}$ ，现在需要判断该关系模式是否满足 BCNF。

此时我们需要分析其是否存在传递依赖。根据之前的图示法求候选键的方法，我们可以先画出相应的函数依赖图，如图 2-10 所示。

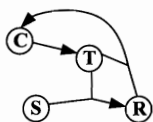


图 2-10 关系模式 P 函数依赖对应有关图

对图 2-10 进行分析，可以得知本关系模式的候选键有 (S, T) 和 (S, C)，所以主属性为 S、T、C，非主属性为 R。但此时由于属性之间的联系错综复杂，所以要界定关系模式是否存在传递函数依赖并不容易。为了准确地界定某关系模式是否为 BCNF，我们需要引入另外的一些判别方法。

一个 BCNF 的关系模式，必须满足以下条件：

- 所有非主属性对每一个码都是完全函数依赖。
- 所有的主属性对每一个不包含它的码，也是完全函数依赖。
- 没有任何属性完全函数依赖于非码的任何一组属性，即每一个函数依赖的左部都必须包含候选键。

这也就是说，若某关系模式不满足以上任意一条，则它不满足 BCNF。

由于有 $C \rightarrow T$ ，而 C 不包含候选键，所以本关系模式不满足 BCNF。

根据各范式的定义，各级范式之间存在如下关系：

$$1NF \supset 2NF \supset 3NF \supset BCNF$$

2.1.7 SQL 语言

结构化查询语言 (SQL) 是关系数据库的标准语言，它是集数据定义语言 (DDL)、数据操纵语言 (DML) 和数据控制功能 (授权、完整性规则和事务控制语句) 于一体的数据库语言。

1. 数据定义语言

SQL 的 DDL 主要包括 SQL 模式、基本表及视图的定义和撤销的操作。包括以下几种，如表 2-4 所示。

表 2-4 数据定义语言

操作对象	操作方式		
	创 建	删 除	修 改
表 (基本表)	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	CREATE INDEX	DROP INDEX	

- 创建基本表的语法格式如下：

```
CREATE TABLE <表名>
    (<列名><数据类型>[列级完整性约束条件]
    [, <列名><数据类型>[列级完整性约束条件]]...
    [, <表级完整性约束条件>]);
```

- 修改基本表的语法格式如下：

```
ALTER TABLE <表名>
    [ADD <新列名><数据类型>[列级完整性约束条件]]
    [DROP <表级完整性约束条件>]
    [MODIFY <列名><数据类型>];
```

2. 数据操纵语言

数据操纵语言可以分为数据查询和数据更新 (插入、删除、更新) 两大类操作。其中由 SELECT 语句实现的数据查询是核心操作，也是历年考试中最重要的考点。其语法如下：

```
SELECT [ALL|DISINCT] <目标列表表达式>[, <目标列表表达式>]...
FROM <表或视图名>[, <表或视图名>]...
[WHERE <条件表达式>]
[GROUP BY <列名 1> [HAVING <条件表达式>]]
[ORDER BY <列名 2> [ASC|DESC]];
```

下面,我们将针对以下几个示例关系模式进行阐述。

student (Sno,Sname,Ssex,Sage,Sdept): 学生(学号,姓名,性别,年龄、所在系)
course (Cno,Cname,Cpno,Ccredit): 课程(课程号,课程名,基础课号,学分)
SC (Sno,Cno,Grade): 学生选修表(学号,课程号,成绩)

(1) 单表查询

单表查询即只涉及一个表的查询,如表 2-5 所示。

表 2-5 单表查询

操作类别	操作子类	示 例	说明及等价代数关系
对列操作	查询指定列	SELECT Sno,Sname FROM student;	$\pi_{Sno,Sname}(student)$ 投影运算
	查询全部列	SELECT * FROM student;	就是表本身
对行操作	消除重复行	SELECT DISTINCT Sno FROM SC;	
	单条件	SELECT Sname FROM student WHERE Sage<20;	查询年龄<20 的学生 $\pi_{Sname}(\sigma_{Sage<'20'}(student))$
	确定集合	SELECT Sname FROM student WHERE Sdept IN('CS', 'MA'); 注: IN (属于), NOT IN (不属于)	查询属于计算机系(CS)及数学系(MA)的学生 $\pi_{Sname}(\sigma_{Sdept='CS'\vee Sdept='MA'}(student))$
	多条件	SELECT Sname FROM student WHERE Sdept='CS' Sage<20;	$\pi_{Sname}(\sigma_{Sdept='CS'\wedge Sage<'20'}(student))$

除此之外,还可以对查询结果进行以下处理,如表 2-6 所示。

表 2-6 对查询结果进行处理

处理类型	处理子类	示例/语法
对结果排序	降序	加上: ORDER BY 排序字段名 DESC;
	升序	默认值,关键字是 ASC
使用集函数(放置于 SELECT 子句中)	统计记录总数	COUNT([DISTINCT ALL] *)
	统计一列中值的个数	COUNT([DISTINCT ALL] <列名>)
	计算一列中值的总和	SUM([DISTINCT ALL] <列名>)
	计算一列值的平均值	AVG([DISTINCT ALL] <列名>)
	求一列值中的最大值	MAX([DISTINCT ALL] <列名>)
	求一列值中的最小值	MIN([DISTINCT ALL] <列名>)
对结果分组	将查询结果按某一列或多列值分组	GROUP BY <列名>,例如,按课程号统计学生数,即得到每个课程号的选修人数: SELECT Cno,Count(Sno)
对结果分组	将查询结果按某一列或多列值分组	FROM SC GROUP BY Cno; 还可以使用 HAVING <条件表达式>筛选, 如 HAVING COUNT(*)>5

(2) 联接查询

联接查询即同时涉及两个以上的表的查询。

它是关系数据库中最主要的查询，包括等值联接、自然联接、非等值联接、自身联接、外联接和复合条件联接查询等几大类，如表 2-7 所示。

表 2-7 联接查询

操作类别	示 例	说明等价代数关系
等值联接 (注：当联接运算符不是=就是非等值联接)	SELECT student.*,SC.* FROM student,SC WHERE student.Sno=SC.Sno;	查询每个学生及其选修课程情况 $\sigma_{\text{student.Sno}=\text{SC.Sno}}(\text{student} \times \text{SC})$ 使用广义笛卡儿积运算
	SELECT student.Sno,Sname,Cno FROM student,SC WHERE student.Sno=SC.Sno;	查询同上 $\pi_{\text{student.Sno,Sname,Cno}}(\text{student} \bowtie \text{SC})$ 使用自然联接运算
自身联接	SELECT A.Cno,B.Cpno FROM course A,course B WHERE A.Cpno=B.Cno; (注：course A, course B 是为 course 定义了个别名，分别为 A 和 B)	查询每一门课程的基础选修课的基础选修课。 $\pi_{\text{Cno,Poon}}(\text{course} \bowtie_{l=3} \text{course})$ 或 $\pi_{\text{student.Sno,Sname,Cno}}(\text{student} \bowtie \text{SC})$
外联接	SELECT student.Sno,Sname,Cno FROM student,SC WHERE student.Sno=SC.Sno(*);	查询每个学生及其选修课程情况，而在联接谓词中加上“*”号，就将使用外联接，即该查询会得出没有选课的学生信息。这个谓词在条件表达式的左边就是左外联接，在右边就是右外联接 $\pi_{\text{student.Sno,Sname,Cno}}(\text{student} \bowtie \text{SC})$
复合条件联接 (即在 WHERE 子句中有多个条件)	SELECT student.Sno,Sname FROM student,SC WHERE student.Sno=SC.Sno AND SC.Cno='2' AND SC.Grade>90;	查询选修 2 号课程且成绩在 90 分以上的学生。 $\pi_{\text{Sno,Sname,Grade}}(\sigma_{\text{SC.Cno}='2' \wedge \text{SC.Grade}>90}(\text{student} \bowtie \text{SC}))$
	SELECT student.Sno,Sname,Grade FROM student,SC,course WHERE student.Sno=SC.Sno AND SC.Cno=Course.Cno AND Cname='数据库';	查询选修课程“数据库”的学生信息及其成绩。 $\pi_{\text{Sno,Sname,Grade}}(\sigma_{\text{Cname}='数据库'}(\text{student} \bowtie \text{SC} \bowtie \text{course}))$

从表 2-7 中我们可以发现在 SQL 与关系代数表达式的转换中，有这样几个规律：SELECT 后面的<列名>一般体现为投影运算；FROM 后面列出了关系模式，WHERE 子句的等值部分通常是转为“自然联接”、“笛卡儿积”等运算；而非等值比较则通常是转为“选择”运算。

(3) 嵌套查询

即在 WHERE 子句中包含另一个 SELECT 查询语句。

在嵌套查询中，上层的查询块称为外层查询或父查询；下层查询块称为内层查询或子查询。SQL 语言允许多层嵌套查询。要注意的是，子查询的 SELECT 语句中不能

使用 ORDER BY 子句。

嵌套查询一般的求解方法是由里向外处理的。即每个子查询在上一级查询处理之前求解，子查询的结果用于建立其父查询的查找条件。嵌套查询通常包括以下几种形式，如表 2-8 所示。

表 2-8 嵌套查询

操作类别	示 例	说明等价代数关系
带 IN 谓词的子查询	SELECT Sno,Sname,Sdept FROM student A WHERE Sdept IN (SELECT Sdept FROM student B WHERE B.Sname='张三'); (注: IN 谓词用来表示属于, 如果要查询与“张三”不同系的学生, 只需将 IN 改为 NOT IN 即可)	查询和“张三”在同一个系里的学生。子查询找出张三所在系的名称。 等价于: SELECT A.Sno,A.Sname,A.Sdept FROM student A,student B WHERE A.Sdept=B.Sdept AND B.Sname='张三'; $\pi_{Sno,Sname,Sdept}(\sigma_{Sdept=B.Sdept \wedge B.Sname='张三'}(student))$
	SELECT Sno,Sname,Sdept FROM student A WHERE Sdept = (SELECT Sdept FROM student B WHERE B.Sname='张三');	当子查询返回的是单值时可以使用>、<、=、!= 等比较运算符。此示例的意义与上一示例相同。要注意的是, 子查询必须在比较运算符之后

注: 从上面的描述中, 可以发现 IN 与=ALL, NOT IN 与<>ALL 是等价的。

(4) 集合查询

SELECT 语句的查询结果是元组(记录)的集合, 所以多个 SELECT 语句的结果可以进行集合操作。SQL 语言提供的集合操作有并操作 UNION、交操作 INTERSECT 和差操作 MINUS。

(5) 数据更新操作(如表 2-9 所示)

表 2-9 数据更新操作

操作类别	语 法	示 例
插入数据	INSERT INTO<表名> [(<属性列 1> [<属性列 2>...])] VALUES(<常量 1> [<常量 2>]...); 注: VALUES 子句可用子查询替代	INSERT INTO student VALUES('95020', '李四', '男', 'CS', '18'); INSERT INTO SC(Sno,Cno) VALUES('95020', '1');
修改数据	UPDATE <表名> SET <列名>=<表达式> [<列名>=<表达式>]... [WHERE <条件>];	将学生 95001 的年龄改为 23 岁 UPDATE student SET Sage=23 WHERE Sno='95001';
删除数据	DELETE FROM <表名> [WHERE <条件>];	删除学号为 95019 的学生记录 DELETE FROM student WHERE Sno='95019';

3. 视图的基本概念

视图是关系数据库系统提供给用户以多种角度来观察数据库中数据的重要机制。它是从一个或多个基本表（及视图）中导出的表，它是一个虚表，只存储定义不存储数据，数据仍然存储在基本表中，基本表数据变化时，视图也就变化了。它和基本表一样，可以查询、删除、更新，但都有一些限制，不同的 DBMS 不尽相同。

视图能够简化用户的操作，能够以多种角度来看待同一数据，提供了一定程度上的逻辑独立性，能够对机密数据提供安全保护。

2.1.8 分布式数据库

分布式数据库系统是数据库技术与网络技术相结合的一个产物。其基本思想是将传统集中式数据库中的数据分布于网络上的多台计算机中。分布式数据库的产生，是信息化技术发展的必然。很多行业的信息化，见证了这一发展历程。如银行系统：最初的银行进行信息化时，是将银行的各个网点，用计算机管理起来。将原本用笔和纸记录的用户信息存入计算机；将以前账本记录的信息存入计算机；将以前的算盘计算，改为计算机计算。此时的各个银行营业点都是独立的，在 A 营业点存的钱，不能在 B 营业点取出。每个营业点的银行系统，都是用的集中式数据库。而随着社会的发展，以及 IT 技术的进步，银行的联网在技术上成为可能，同时社会也有这种需求，就形成了银行各个网点的联网。此时，用户在 A 营业点存钱，可以在 B 营业点取，这就用到了分布式数据库。

分布式数据库体系结构与集中式数据库的结构从宏观上来讲，是相似的。局部 DBMS 中的局部内模式与局部概念模式与集中数据库是完全一致的，不同之处在于新增的全局 DBMS，而整个全局 DBMS，可以看做是相对于局部概念模式的“外模式”。由于“外模式”部分有一系列的“分布模式、分片模式、全局概念模式、全局外模式”以及多级映射使得用户在使用分布式数据库时，可以用集中式数据库同样的方式来应用。

分布式数据库的体系系统结构如图 2-11 所示。

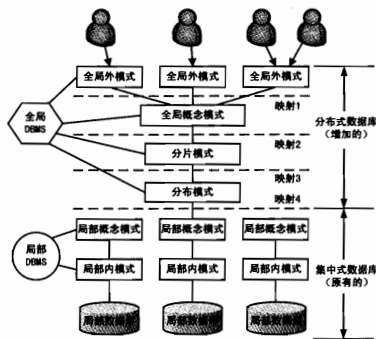


图 2-11 分布式数据库体系结构图

（1）全局外模式

它们是全局应用的用户视图，是全局概念模式的子集，该层直接与用户（或用户应用程序）交互。

(2) 全局概念模式

它定义分布式数据库中数据的整体逻辑结构,数据就如同根本没有分布一样,可用传统的集中式数据库中所采用的方法定义

全局概念模式中所用的数据模型应该易于向其他层次的模式映像,通常采用关系模型。这样,全局概念模式包括一组全局关系的定义。

(3) 分片模式

在某些情况下,我们需要将一个关系模式分解成为几个数据片,分片模式正是用于完成此工作。分片方式主要有水平分片、垂直分片和混合分片。

(4) 分布模式

分布式数据库的本质特性就是数据分布在不同的物理位置。分布模式的主要职责就是定义数据片段(即分片模式的处理结果)的存放节点。分布模式的映像类型确定了分布式数据库是冗余的还是非冗余的。若映像是一对多的,即一个片段分配到多个节点上存放,则是冗余的分布数据库,否则是不冗余的分布数据库。

根据分布模式提供的信息,一个全局查询可分解为若干子查询,每一子查询要访问的数据属于同一场地的局部数据库。由分布模式到各局部数据库的映像(映像4)把存储在局部场地的全局关系或全局关系的片段映像为各局部概念模式采用局部场地的DBMS所支持的数据模型。

(5) 局部概念模式

局部概念模式即局部数据库的“概念模式”。前文已对概念模式进行详细解释,在此不再复述。

(6) 局部内模式

局部内模式即局部数据库的“内模式”。前文已对内模式进行详细解释,在此不再复述。

理论上讲分布式数据库系统的模式结构有以上6个层次,但实际上并非所有分布式数据库系统都具有这种结构。在这种结构中各级模式的层次清晰,可以概括和说明任何分布式数据库系统的概念和结构。

数据分片将数据库整体逻辑结构分解为合适的逻辑单位(片段),然后由分布模式来定义片段及其副本在各场地的物理分布,其主要目的是提高访问的局部性,有利于按照用户的需求组织数据的分布和控制数据的冗余度。

分片的方式有多种,水平分片和垂直分片是两种基本的分片方式,混合分片和导出分片是较复杂的分片方式。

(1) 水平分片

水平分片把一个全局关系中的元组分裂成多个子集,每个子集为一个片段。分片条件由关系中的属性值表示。对于水平分片,重构全局关系可通过关系的并操作实现。

(2) 垂直分片

垂直分片把一个全局关系按列分裂成多个子集,应满足不相交性(关键字除外)。对于垂直分片,重构全局关系可通过连接运算实现。

(3) 混合分片

混合分片是在分片中采用水平分片、垂直分片两种形式的混合。

数据分片应遵循的准则如下：

- 完整性。全局关系的所有数据都必须分配到各个片段中，不允许某些数据属于全局关系但不属于任何片段。
- 重构性。分裂后各个片段可以重构原来的全局关系。
- 不相交性。全局关系中的每个元组仅属于一个片段，不能在多个片段中重复出现。此规则不是必须的，因为在有冗余 DDBS 中数据可有多个副本。但片段中的部分元组重复将会使数据的更新操作变得复杂，为简化操作控制，片段间一般是不相交的。

2.1.9 数据仓库与数据挖掘

数据仓库与数据挖掘技术应用于企业的决策分析，它们与之前描述的普通数据库存在着一定的差别。

1. 数据仓库体系结构

数据仓库是一个面向主题的、集成的、相对稳定的、反映历史变化的数据集合，用于支持管理决策。近年来，人们对数据仓库技术的关注程度越来越高，其原因是过去的几十年中，建设了无数的应用系统，积累了大量的数据，但这些数据没有得到很好的利用，有时反而成为企业的负担。数据仓库技术正是将这些数据变废为宝的有效手段，如图 2-12 所示为数据仓库体系结构图。

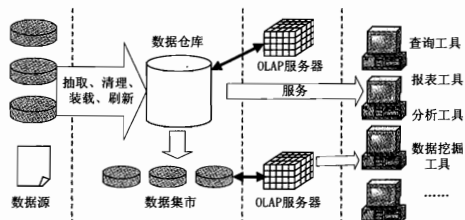


图 2-12 数据仓库体系结构图

(1) 数据源

数据源是数据仓库系统的基础，是整个系统的数据源泉，通常包括企业内部信息和外部信息。内部信息包括存放于 RDBMS 中的各种业务处理数据和各类文档数据。外部信息包括各类法律法规、市场信息和竞争对手的信息等。

(2) 数据的存储与管理

数据的存储与管理是整个数据仓库系统的核心。数据仓库的关键是数据的存储和管理。数据仓库的组织管理方式决定了它有别于传统数据库，同时也决定了其对外部数据的表现形式。要决定采用什么产品和技术来建立数据仓库的核心，则需要从数据仓库的技术特点着手分析；针对现有各业务系统的数据，进行抽取、清理，并有效集成，按照主题进行组织。按照数据的覆盖范围，数据仓库可以分为企业级数据仓库和

部门级数据仓库（通常称为数据集市）。

（3）OLAP 服务器

OLAP 服务器对分析需要的数据进行有效集成，按多维模型予以组织，以便进行多角度、多层次的分析，并发现趋势。其具体实现可以分为 ROLAP、MOLAP 和 HOLAP。ROLAP 基本数据和聚合数据均存放在 RDBMS 中；MOLAP 基本数据和聚合数据均存放于多维数据库中；HOLAP 基本数据存放于 RDBMS 中，聚合数据存放于多维数据库中。

（4）前端工具

前端工具主要包括各种报表工具、查询工具、数据分析工具、数据挖掘工具，以及各种基于数据仓库或数据集市的应用开发工具。其中，数据分析工具主要针对 OLAP 服务器，报表工具、数据挖掘工具主要针对数据仓库。

2. 数据仓库的开发与实施

根据软件工程思想及数据仓库系统实施方法学，可以将数据仓库的开发、实施分为以下几个步骤。

（1）业务需求分析

业务需求分析是数据仓库建设的基础，应该同用户进行充分沟通，了解用户的真实需求，避免理解的误差，同时，应该界定好项目开发范围。此阶段的主要工作包括：

- 设定可以达到的目标并明确所有需求。
- 确定系统体系结构。从实施的角度来看，设计数据仓库系统体系结构有多种方式：构造部门级的数据集市（DataMart）；直接构造企业级的数据仓库（DataWarehouse）系统；先建立部门级数据集市，然后发展成企业级数据仓库系统。
- 确定数据源。列出向数据仓库提供数据的数据源清单。数据源的复杂性、规模、完整性对建立数据仓库的影响比其他因素要大。要格外注意哪些数据源的数据类型、粒度和内容是兼容的。
- 容量规划。除了体系结构之外，硬件和软件资源对数据仓库也至关重要。作为需求定义的一部分，估计数据仓库将要存储的数据量，以及将对数据进行的处理很重要。
- 技术评价。在选择软件和硬件平台时，最好听取专家的建议，尤其是对与你的环境相似的有经验的专家。

（2）逻辑模型设计

逻辑模型设计主要是指数据仓库数据的逻辑表现形式。从最终应用的功能和性能的角度来看，数据仓库的数据模型也许是整个项目最重要的方面。为数据仓库和数据集市定义数据模型是一项复杂的工作，需要领域专家的参与。

（3）物理模型设计

在进行物理模型设计时，主要是将数据仓库的逻辑模型转换为在数据库中的物理表结构。

(4) 数据抽取、清洗、集成、装载等

数据抽取是数据仓库建立过程中的一个非常重要的步骤。它负责将分布在用户业务系统中的数据进行抽取、清洗、集成，并将数据存储到数据仓库中。在此阶段主要涉及以下两方面的工作：

- 定义数据载入和维护策略。
- 数据抽取/清洗/转换/装载。

(5) 数据仓库的管理

数据仓库元数据的管理也是极为重要的环节。

(6) 数据的分析、报表、查询等数据的表现

用户分析、报表、查询工具是用户进行分析决策使用的工具。因此，其所有操作要非常简单，但提供的功能却要十分强大。此外，数据挖掘技术也是数据仓库系统中一个重要部分。主流的数据仓库产品都提供相应的数据挖掘工具；当然也可以选用第三方数据挖掘工具。

(7) 数据仓库性能优化及发布

数据仓库性能的好坏直接影响系统查询、分析响应速度。

3. 数据仓库的实现方法

从整体的角度来看，数据仓库有以下实现方法。

(1) 自顶向下方法

在该方法中，首先应找出数据仓库解决方案所要满足的业务需求，把业务需求视为实现数据仓库的首要任务。数据仓库是一种功能而不是一种特征，数据仓库保存信息，并以外部工具易于显示和操作的方式组织这些信息。因此，如果不借助于可以利用这种功能的外部工具，最终用户就无法将这种功能嵌入到数据仓库中。这样，就很难界定该功能的范围，除非用广义上的业务术语，如“数据仓库将包含有关客户、供应商、市场、产品的信息”。自顶向下方法的优点和缺点如表 2-10 所示。

表 2-10 自顶向下方法的优点和缺点

优 点	缺 点
业务需求清楚地描绘出数据仓库实现的范围，因此是实现数据仓库解决方案的有效方法	机会有时超出了当前的业务范围
技术取决于业务	技术可以促进业务和竞争优势，但开始时对业务的促进是不明显的
易于向决策者提供数据仓库的收益情况	一旦数据仓库已经实现，可能就不再追求更高的目标

规划和实现数据仓库的自顶向下方法一般用于以下情况：

- 实现单位比较熟悉技术，并具有根据业务需求采用自顶向下方法开发应用程序的丰富经验。
- 决策层（总经理、决策者、投资者）完全清楚数据仓库的预测目标。
- 决策层（总经理、决策者、投资者）完全清楚数据仓库用做哪些机构的决策支持工具。

- 决策层（总经理、决策者、投资者）完全清楚数据仓库已经是业务过程中的一个子过程。

如果技术是成熟的和众所周知的，或者必须解决的业务问题是显而易见的，那么自顶向下方法是很有用的。采用自顶向下方法可以将技术和业务目标有机地结合起来。

（2）自底向上方法

自底向上方法一般从实验和基于技术的原型入手。先选择一个特定的、众所周知的业务问题的子集，再为该子集制订方案。实现自底向上方法一般是比较快的。自底向上方法可以使一个单位在发展时用尽可能少的经费和时间，就能在做出有效的投入之前评估技术的收益情况。在数据仓库领域，自底向上方法是快速实现数据集市、部门级数据仓库的有效手段。自底向上方法的优点和缺点如表 2-11 所示。

表 2-11 自底向上方法的优点和缺点

优 点	缺 点
实现的需求和开始时的需要远远超过自顶向下分析方法和长期考虑的范围	最初方案实现之后，最好回顾一下方案是如何服务于整个企业的
在企业了解数据仓库早期，该方法使企业无须巨大投入就可见到效益	单个自底向上工程项目的失败可能推迟潜在技术的实现
少数人集中工作在一个部门范围，可以加速实现决策过程	早期的小组应不断发展为较大的小组，以扩充最初方案的覆盖范围

规划和实现数据仓库的自底向上方法一般用于以下情况：

- 企业还没有确实掌握数据仓库技术，希望进行技术评估来决定运行该技术的方式、地点和时间。
- 企业希望了解实现和运行数据仓库所需要的各种费用情况。
- 企业在对数据仓库进行投资选择。

自底向上方法对于希望从数据仓库投资中快速得到回报的用户是非常有效的。该方法可以使企业充分利用各种技术，无须冒很大风险。

（3）联合方法

在以上两种方法的联合方法中，企业在保持自底向上方法的快速实现和基于应用的同时，还可以利用自顶向下方法的规划和决策性质。这种方法依赖于以下两个因素：

- 自顶向下的结构、标准和设计小组，可以从一个项目向另外一个项目传递知识，也可以把战术决策变为战略决策。
- 自底向上方法的项目小组，它直接负责在短期内实现一个集中的、部门级的商务解决方案。

联合方法具有以上两种方法的优点，但是难以作为一个项目来管理。该方法一般用于以下情形：

- 企业拥有经验丰富的设计师，有能力建立、证明、应用和维护数据结构、技术结构，以及企业模型，可以很容易地从具体（运作系统中的元数据）转移到抽象。
- 企业拥有固定的项目小组，完全清楚数据仓库技术应用的场所。他们可以清

楚地看到当前的商务需求。

联合方法适合数据仓库技术的快速试运行，并且保留了建立长远的决策方案的机会。

4. 数据仓库相关的数据存储技术

数据存储技术包含多介质存储设备的管理技术、数据存储的控制技术、数据的并行存储与管理技术、可变长技术和锁切换技术。

(1) 多介质存储设备的管理技术

在数据仓库中管理大批量数据时，为了达到效率和费用的平衡，通常要求数据仓库能够实现多存储介质设备的管理。即能够对磁盘、磁带等各种介质的存储设备实现有效的管理。

(2) 数据存储的控制

数据仓库的高效运行技术之一在于数据仓库设计者可以在物理块或物理页上对数据存储进行有效的控制。数据仓库设计者可以利用这一技术对数据存储的物理地址进行调整，使其更加适合数据仓库的使用要求。

(3) 数据的并行存储与管理

考虑数据仓库的多用户环境，数据仓库应该具备数据的并行存储与管理技术。这样才能使数据仓库的性能得到极大的提高。

(4) 可变长技术

在数据仓库中，如果经常发生变长数据的变动，将严重影响数据仓库的系统性能。因此，在数据仓库中需要有效管理变长数据的技术。

(5) 锁切换技术

由于数据仓库中的数据很少发生变化，因此，长期在加锁管理下进行数据仓库的操作将降低数据仓库的效率。为使数据仓库高效率地运行，要为数据仓库提供锁切换管理技术，使用户可以在需要的时候进行加锁操作或开锁操作。

5. 数据仓库技术的发展趋势

数据仓库技术的发展包括数据抽取、存储管理、数据表现和方法论等方面。

- 在数据抽取方面，未来的技术发展将集中于系统集成化。它将互联、转换、复制、调度、监控纳入标准化的统一管理以适应数据仓库本身或数据源可能的变化，使系统更便于管理和维护。
- 在存储管理方面，未来的发展将使数据库厂商明确推出数据仓库引擎，作为服务器产品与数据库服务器并驾齐驱。在这一方面，带有决策支持扩展的并行关系数据库将最具发展潜力。
- 在数据表现方面，数理统计的算法和功能将普遍集成到联机分析产品中，同时与 Internet/Web 技术紧密结合，推出适用于 Intranet 和终端免维护的数据仓库访问前端。
- 数据仓库实现过程的方法论将成为数据库设计的一个明确分支，并将成为管

理信息系统设计的必备内容。

计算机应用发展的数据仓库倾向是数据仓库发展的推动力。传统的联机事务处理系统并不单独考虑数据仓库，但实际应用对数据仓库所能提供的功能却早有需求。因此，许多事务处理系统近年来陷入一个两难的境地：如果在现有系统上增加有限的联机分析功能，包括复杂的报表和数据汇总操作，一方面会严重影响事务处理性能，另一方面统计分析又因系统结构上的种种限制而不能充分体现。在新一代的应用系统中，数据仓库在一开始便被纳入系统设计的考虑，联机分析应用于普遍的事务处理系统之中。在数据管理方面，联机事务处理和数据仓库在应用中相对独立，使联机事务处理系统本身更加简捷高效，同时统计分析也更为便利。面向行业的数理统计学向更为普遍的应用发展，并集成到应用系统的数据仓库解决方案中，它们将立足于数据仓库提供的丰富信息，更好地为业务决策服务。

6. 数据挖掘

数据挖掘与传统的数据分析（如查询、报表、联机应用分析）的本质区别是数据挖掘是在没有明确假设的前提下去挖掘信息、发现知识。数据挖掘所得到的信息应具有先知、有效和可实用3个特征。先前未知的信息是指该信息是预先未曾预料到的，即数据挖掘是要发现那些不能靠直觉发现的信息或知识，甚至是违背直觉的信息或知识，挖掘出的信息越是出乎意料，就可能越有价值。

如图2-13所示为数据挖掘体系结构图，它展示了数据挖掘的流程，说明了数据挖掘是怎样找到新规律的。

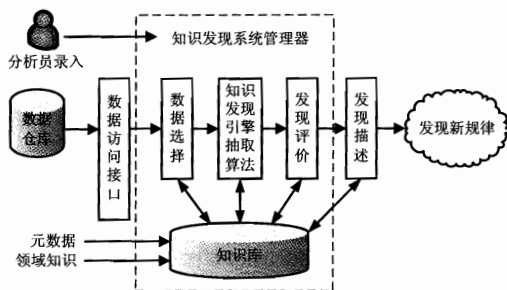


图 2-13 数据挖掘体系结构图

(1) 知识发现系统管理器

控制并管理知识发现过程，录入知识库中的信息用于驱动数据选择过程、抽取算法选择，以及使用过程和发现评价过程。

(2) 知识库

知识库包含源于多方面的必需的信息，可以将元数据输入数据仓库中，以描述数据仓库的数据结构，输入关键数据属性、规则和数据层次等。

(3) 数据访问接口

知识发现系统利用数据库的查询机制从数据仓库中提取数据，可使用 SQL 查询语言，结合知识库中的数据仓库元数据，指导从数据仓库中提取需要的数据。

(4) 数据选择

确定从数据仓库中需要抽取的数据及其结构。

(5) 知识发现引擎

将知识库中的抽取算法提供给抽取的数据，目的是要抽取数据元素间的模式和关系。

(6) 发现评价

分析员要寻找关注性的数据模式，选出那些关注性信息。

(7) 发现描述

发现描述部分提供两种功能，一种是以发现评价辅助分析员在知识库中保存所发现的信息，以备将来引用和使用；另一种是保持发现与决策者的通信。

2.2 典型试题分析

试题 1

设有员工实体 Employee (employeeID, name, sex, age, tel, departID)，其中 employeeID 为员工号，name 为员工姓名，sex 为员工性别，age 为员工年龄，tel 为员工电话，记录该员工的手机号码、办公室电话等，departID 为员工所在部门号，参照另一部门实体 Department 的主码 departID。

Employee 实体中存在派生属性 (1)。Employee 实体中还存在多值属性 (2)。对属性 departID 的约束是 (3)。

- (1) A. name, 原因是会存在同名员工
B. age, 原因是用属性 birth 替换 age 并可计算 age
C. tel, 原因是员工有多个电话
D. departID, 原因是实体 Department 已有 departID
- (2) A. name, 可以用 employeeID 区别
B. sex, 可以不作任何处理
C. tel, 可以将 tel 加上 employeeID 独立为一个实体
D. tel, 可以强制只记录一个电话号码
- (3) A. Primary Key, NOT NULL
B. Primary Key
C. Foreign Key
D. Candidate Key

试题分析

所谓派生属性，是表示其值可以从一个相关属性和属性集的值派生得到的属性，这个属性在实体中不是必需的。根据这个定义，age 是一个派生属性，因为用属性 birth (出生时间) 替换 age 并可计算 age。

多值属性是指可同时由多个值表示的属性。例如，包含关于雇员信息的数据库可

能包含关于他们个人兴趣的数据。一个雇员可能有几个兴趣：运动、电影、投资、烹调，并且由于这些值的任何一个或所有这些值可能同时是雇员的兴趣，所以这些数据应作为数据类型列表合成到资源。对于多值属性的处理，一般是提升为单独的一个或多个实体。

因为 departID 为员工所在部门号，参照另一部门实体 Department 的主码 departID，因此，在实体 Employee 中，departID 为外键。

试题答案

(1) B

(2) C

(3) C

试题 2

设关系模式 $R<U, F>$ ，其中 $U=\{A, B, C, D, E\}$ ， $F=\{A\rightarrow BC, C\rightarrow D, BC\rightarrow E, E\rightarrow A\}$ ，则分解 $\rho=\{R_1(ABCE), R_2(CD)\}$ 满足 (4)。

- (4) A. 具有无损联接性、保持函数依赖 B. 不具有无损联接性、保持函数依赖
C. 具有无损联接性、不保持函数依赖
D. 不具有无损联接性、不保持函数依赖

试题分析

首先看分解是否保持函数依赖。在 F 中有 4 个函数依赖。 $A\rightarrow BC$ 、 $BC\rightarrow E$ 和 $E\rightarrow A$ 在 R_1 中得到了保持， $C\rightarrow D$ 在 R_2 中得到了保持，因此分解是保持函数依赖的。

接下来，再根据以下定理判断是否为无损联接。

设 $\rho=\{R_1, R_2\}$ 是 R 的一个分解， F 是 R 上的函数依赖集，那么分解 ρ 相对于 F 是无损联接分解的充要条件是 $(R_1\cap R_2)\rightarrow(R_1-R_2)$ 或 $(R_1\cap R_2)\rightarrow(R_2-R_1)$ 。要注意的是，这两个条件只要任意一个条件成立就可以了，同时该定理只适用于“一分为二”的情况，即一个关系模式分解为两个关系模式。

在本题中， $R_1\cap R_2=\{C\}$ ， $R_1-R_2=ABE$ ， $R_2-R_1=D$ ，因为在 F 中有 $C\rightarrow D$ 成立，所以，分解是无损联接。

试题答案

(4) A

试题 3

在关于数据挖掘的描述中，正确的是 (5)。

- (5) A. 数据挖掘可以支持人们进行决策
B. 数据挖掘可以对任何数据进行
C. 数据挖掘与机器学习是统一的
D. 数据来源质量对数据挖掘结果的影响不大

试题分析

本题考查数据挖掘的基本概念，数据挖掘用于支持决策分析，同时数据挖掘是应

用于数据仓库之上的处理加工，对数据有一定的要求，如果数据源质量不好，必然影响到挖掘的结果。

试题答案

(5) A

试题 4

关系 $R(A, B, C, D)$ 和 $S(B, C, D)$ 进行笛卡儿运算，其结果集为 (6) 元关系。3 个实体及它们之间的多对多联系至少应转换成 (7) 个关系模式。

(6) A.4 B.3 C.6 D.7

(7) A.3 B.4 C.5 D.6

试题分析

关系 R 为 4 元关系，关系 S 为 3 元关系，因此，其笛卡儿积为 7 元关系。

3 个以上实体间的一个多元联系可以转换为一个独立的关系模式，与该联系相连的各实体的码和联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合。因此，3 个实体及它们之间的多对多联系至少应转换成 4 个关系模式。

试题答案

(6) D

(7) B

试题 5

关于诊疗科、医师、患者和治疗观察关系模式如下所示，其中带实下画线的表示主键，带虚下画线的表示外键。

诊疗科 (诊疗科代码, 诊疗科名称)

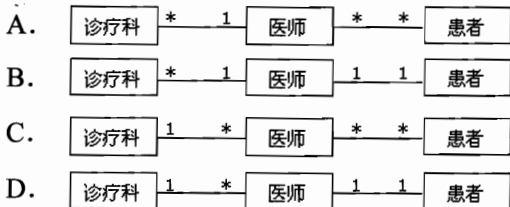
医师 (医师代码, 医师姓名, 诊疗科代码)

患者 (患者编号, 患者姓名)

治疗观察 (医师代码, 患者编号)

(8) 表示上述关系模式的 E-R 图。图中，*、1*、和 11 分别表示多对多、1 对多和 1 对 1 的联系。

(8)



显然, 根据常识, 医师和患者之间的关系为多对多的联系, 诊疗科和医师之间的关系为 1 对多的关系。

试题答案

(8) C

试题 6

若对表 2-12 按 (9) 进行运算, 可以得到表 2-13。

表 2-12 表 1

条形码	商品名	价格
01020210	牙刷	3
01020211	毛巾	10
01020212	毛巾	8
01020213	铅笔	0.5
02110200	钢笔	8

表 2-13 表 2

条形码	商品名	价格
01020211	毛巾	10
01020212	毛巾	8
02110200	钢笔	8

- (9) A. $\sigma_{\text{条形码} = '01020211' \vee '02110200'}$ (表 1) B. $\sigma_{\text{商品名} = \text{'毛巾'} \vee \text{'铅笔'}}$ (表 1)
 C. $\sigma_{\text{价格} \geq 8}$ (表 1) D. $\pi_{1,2,3}$ (表 1)

试题分析

表 2-13 保持了表 2-12 的所有属性和部分元组, 因此是选择操作。从 3 个选择操作来看, 只有 C 满足要求。

试题答案

(9) C

试题 7

对于关系模式 R (X, Y, Z), 下列结论错误的是 (10)。

- (10) A. 若 $X \rightarrow Y, Y \rightarrow Z$, 则 $X \rightarrow Z$ B. 若 $X \rightarrow Y, X \rightarrow Z$, 则 $X \rightarrow YZ$
 C. 若 $X \rightarrow Z$, 则 $XY \rightarrow Z$ D. 若 $XY \rightarrow Z$, 则 $X \rightarrow Z, Y \rightarrow Z$

试题分析

本题考查函数依赖的推理规则, 显然, 只有选项 D 是错误的。

试题答案

(10) D

试题 8

若系统中存在一个等待事务集 $\{T_0, T_1, T_2, \dots, T_n\}$ ，其中， T_0 正等待被 T_1 锁住的数据项 D_1 ， T_1 正等待被 T_2 锁住的数据项 D_2 ， \dots ， T_{n-1} 正等待被 T_n 锁住的数据项 D_n ， T_n 正等待被 T_0 锁住的数据项 D_0 ，则系统处于(11)的工作状态。

(11) A. 封锁 B. 死锁 C. 循环 D. 并发处理

试题分析

与操作系统中的进程调度类似，若在数据库系统中存在一个等待事务集 $\{T_0, T_1, T_2, \dots, T_n\}$ ，其中， T_0 正等待被 T_1 锁住的数据项 A_1 ， T_1 正等待被 T_2 锁住的数据项 A_2 ， \dots ， T_{n-1} 正等待被 T_n 锁住的数据项 A_n ， T_n 正等待被 T_0 锁住的数据项 A_0 ，则系统处于死锁的工作状态。

在数据库中，产生死锁的原因是两个或多个事务都已封锁了一些数据对象，然后又都请求对已为其他事务封锁的数据对象加锁，从而出现死等待。预防死锁的方法通常有一次封锁法（每个事务必须一次将所有要使用的数据全部加锁，否则就不能继续执行）和顺序封锁法（预先对数据对象规定一个封锁顺序，所有事务都按这个顺序进行封锁），死锁的诊断方法一般有超时法和等待图法。

试题答案

(11) B

试题 9

在某学校的综合管理系统设计阶段，教师实体在学籍管理子系统被称为“教师”，而在人事管理子系统被称为“职工”，这类冲突被称之为(12)。

(12) A. 语义冲突 B. 命名冲突 C. 属性冲突 D. 结构冲突

试题分析

根据局部应用设计好各局部 E-R 图之后，就可以对各分 E-R 图进行合并。合并的目的在于在合并过程中解决分 E-R 图中相互间存在冲突，消除分 E-R 图之间存在的信息冗余，使之成为能够被全系统所有用户共同理解和接受的统一的、精炼的全局概念模型。分 E-R 图之间的冲突主要有命名冲突、属性冲突和结构冲突 3 类。

选项 B 正确，因为命名冲突是指相同意义的属性，在不同的分 E-R 图上有着不同的命名，或是名称相同的属性在不同的分 E-R 图中代表着不同的意义，这些也要进行统一。

选项 C 不正确，因为属性冲突是指同一属性可能会存在于不同的分 E-R 图中，由于设计人员不同或是出发点不同，对属性的类型、取值范围和数据单位等可能会不一致，这些属性对旧的数据将来只能以一种形式在计算机中存储，这就需要在设计阶段进行统一。

选项 D 不正确, 因为结构冲突是指同一实体在不同的分 E-R 图中有不同的属性, 同一对象在某一分 E-R 图中被抽象为实体而在另一分 E-R 图中又被抽象为属性, 需要统一。

试题答案

(12) B

试题 10

关于数据库中关系性质的描述, 错误的是___(13)___。

- (13) A. 表中任意两行的值不能相同 B. 表中任意两列的值不能相同
C. 行在表中的顺序无关 D. 列在表中的顺序无关

试题分析

关系可以有 3 种类型: 基本关系 (通常又称为基本表或基表)、查询表和视图表。基本表是实际存在的表, 它是实际存储数据的逻辑表示。查询表是查询结果对应的表。视图表是由基本表或其他视图表导出的表, 是虚表, 不对应实际存储的数据。

基本关系具有以下 6 条性质:

- 列是同质的, 即每一列中的分量是同一类型的数据, 来自同一个域。
- 不同的列可出自同一个域, 称其中的每一列为一个属性, 不同的属性要给予不同的属性名。
- 列的顺序无所谓, 即列的次序可以任意交换。
- 任意两个元组不能完全相同。但在大多数实际关系数据库产品中, 如 Oracle 等, 如果用户没有定义有关的约束条件, 它们都允许关系表中存在两个完全相同的元组。
- 行的顺序无所谓, 即行的次序可以任意交换。
- 分量必须取原子值, 即每一个分量都必须是不可分的数据项。

试题答案

(13) B

试题 11

建立一个供应商、零件数据库。其中“供应商”表 S (Sno, Sname, Zip, City) 中的属性分别表示供应商代码、供应商名、供应商邮编、供应商所在城市, 其函数依赖为 $Sno \rightarrow (Sname, Zip, City)$, $Zip \rightarrow City$ 。“零件”表 P (Pno, Pname, Color, Weight, City), 表示零件号、零件名、颜色、重量及产地。表 S 与表 P 之间的关系 SP(Sno, Pno, Price, Qty) 表示供应商代码、零件号、价格、数量。

- “供应商”表 S 属于___(14)___。

(14) A. 1NF B. 2NF C. 3NF D. BCNF

- 若要求: 供应商代码不能为空, 且值是唯一的, 供应商的名也是唯一的; 零件号不能为空, 且值是唯一的; 一个供应商可以供应多种零件, 而一种零件

可以由多个供应商供应。请将下面的 SQL 语句空缺部分补充完整。

```
CREATE TABLE S(Sno CHAR(5) (15),  
Sname CHAR(30) UNIQUE,  
Zip CHAR(8),  
City CHAR(20)  
(16) );
```

- (15) A. FOREIGN KEY B. NOT NULL UNIQUE
C. FOREIGN KEY (Sno) D. PRIMARY KEY (Sname)
(16) A. NOT NULL B. NOT NULL UNIQUE
C. PRIMARY KEY (Sno) D. PRIMARY KEY (Sname)

试题分析

显然,表 S 的主键是 Sno,说明 S 至少是 2NF。但因为存在 $Sno \rightarrow Zip$ 和 $Zip \rightarrow City$,也就是说, City 是传递依赖于主键的,所以 S 不是 3NF。

在给定的 SQL 语句中,因为 Sno 是主键,所以(15)空应该填“PRIMARY KEY”,但给出的选项中没有这个词组。试题描述告诉我们,“供应商代码不能为空,且值是唯一的”,所以应该选择 B,而在(16)空处填写“PRIMARY KEY (Sno)”。

试题答案

- (14) B (15) B (16) C

试题 12

某高校管理信息系统的数据库设计过程中, (17) 阶段是在需求分析的基础上,对用户信息加以分类、聚集和概括,建立信息模型,并依照选定的数据库管理系统软件,转换成为数据的 (18),再依照软硬件环境,最终实现数据的合理存储。

- (17) A. 物理设计 B. 逻辑结构设计 C. 数据库实施 D. 概念结构设计
(18) A. 物理模式 B. 逻辑模式 C. 内模式 D. 概念模式

试题分析

通常将数据库设计分为需求分析、概念结构设计、逻辑结构设计和数据库物理设计 4 个阶段。

需求分析是指收集和分析用户对系统的信息需求和处理需求,得到设计系统所必需的需求信息,建立系统说明文档。其目标是通过调查研究,了解用户的数据要求和处理要求,并按一定格式整理形成需求说明书。

概念结构设计阶段的目标是对需求说明书提供的所有数据和处理要求进行抽象与综合处理,按一定的方法构造反映用户环境的数据及其相互联系的概念模型,即用户的数据模型或企业数据模型。这种概念数据模型与 DBMS 无关,是面向现实世界的、

极易为用户所理解的数据模型。为保证所设计的概念数据模型能正确、完全地反映用户的数据及其相互关系,便于进行所要求的各种处理,在本阶段设计中可吸收用户参与和评议设计。在进行概念结构设计时,可先设计各个应用的视图,即各个应用所看到的数据及其结构,然后再进行视图集成,以形成一个单一的概念数据模型。这样形成的初步数据模型还要经过数据库设计者和用户的审查与修改,最后形成所需的概念数据模型。

逻辑结构设计阶段的设计目标是把上一阶段得到的与 DBMS 无关的概念数据模型转换成等价的,并为某个特定的 DBMS 所接受的逻辑模型所表示的概念模式,同时将概念设计阶段得到的应用视图转换成外部模式,即特定 DBMS 下的应用视图。在转换过程中要进一步落实需求说明,并满足 DBMS 的各种限制。该阶段的结果是用 DBMS 所提供的数据库定义语言(DDL)写成的数据模式。逻辑设计的具体方法与 DBMS 的逻辑数据模型有关。逻辑模型应满足数据库存取、一致性及运行等各方面的用户需求。

物理设计阶段的任务是把逻辑设计阶段得到的满足用户需求的已确定的逻辑模型在物理上加以实现,其主要的内容是根据 DBMS 提供的各种手段,设计数据的存储形式和存取路径,如文件结构、索引的设计等,即设计数据库的内模式或存储模式。数据库的内模式对数据库的性能影响很大,应根据处理需求及 DBMS、操作系统和硬件的性能进行精心设计。

试题答案

(17) D

(18) B

试题 13

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(19)。

(19) A. 数据转换 B. 数据抽取 C. 数据清洗 D. 数据装载

试题分析

构建数据仓库的重要一环就是用户从数据源抽取所需的数据,经过数据清洗,最终按照预先定义好的数据仓库模型,将数据加载到数据仓库中去。

试题答案

(19) C

试题 14

(20) 可用于描述数据流图中数据存储及其之间的关系,最初用于数据库概念设计。在某学生选课系统中使用该工具来描述,学生的学号属于(21)。

(20) A. 实体关系图 B. 数据字典 C. IPO 图 D. 判定表

(21) A. 实体 B. 关系 C. 属性 D. 方法

试题分析

实体联系图可用于描述数据流图中数据存储及其之间的关系,最初用于数据库概

念设计。在实体联系图中，有实体、联系和属性 3 个基本部分。在某学生的选课系统中，学生属于实体，而学生的学号则属于该实体的属性。

试题答案

(20) A

(21) C

试题 15

在数据库设计的需求分析阶段应完成包括__(22)__在内的文档。

(22) 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

试题分析

本题考查数据库设计方面的相关知识。数据库的设计主要分为：需求分析阶段、概念设计阶段、逻辑设计阶段、物理设计阶段。

需求分析阶段的任务是对现实世界要处理的对象（组织、部门和企业等）进行详细调查，在了解现行系统的概况、确定新系统功能的过程中收集支持系统目标的基础数据及处理方法。需求分析是在用户调查的基础上，通过分析，逐步明确用户对系统的需求。在需求分析阶段应完成的文档是数据字典和数据流图。

概念设计阶段的任务是完成用户的数据模型，这种模型是与 DBMS 无关的概念模型，常见的有 E-R 模型。

逻辑设计阶段的任务是将概念模型转换成具体的关系模式。

物理设计阶段的任务是将关系模式加入 DBMS 的特性，成为具体某个 DBMS 的数据库。

试题答案

(22) C

试题 16

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__(23)__。

(23) 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

试题分析

本题考查对数据完整性约束方面基础知识的掌握。

完整性约束包括：实体完整性约束、参照完整性约束和用户自定义完整性约束 3 类。

实体完整性要求主键中的任一属性不能为空，同时主键不能有重复值。

参照完整性要求外键的值，要么为空，要么为对应关系主键值域。同时仅当参照关系中没有任何元组的外键值与被参照关系中要删除元组的主键值相同时，系统才可以执行删除操作，否则拒绝执行删除操作。

用户定义的完整性是针对某一具体数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求。一般用于限制某字段值的取值范围，此范围不涉及其他数据表的值。

从以上描述来看，题目的要求，以上3种完整性约束都无法达到目的。所以需要考虑触发器，触发器的功能一般比完整性约束要强得多。触发器的原理是通过编写相应的触发器脚本代码，来对某个字段值的变化进行监控，一旦值发生变化，则触发器脚本执行。在本题中，需要达到的效果是EMP中的工资产生变化，则需要判断变化值是否在P关系规定的范围之内，所以应在EMP上建立触发器。本题选C。

关于完整性约束的具体内容，请参看“2.1.5 完整性约束”。

试题答案

(23) C

试题 17

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 $F=\{A \rightarrow B, DE \rightarrow B, CB \rightarrow E, E \rightarrow A, B \rightarrow D\}$ 。(24) 为关系 R 的候选关键字。分解 (25) 是无损联接，并保持函数依赖的。

(24) A. AB

B. DE

C. CE

D. CB

(25) A. $p=\{R_1(AC), R_2(ED), R_3(B)\}$

B. $p=\{R_1(AC), R_2(E), R_3(DB)\}$

C. $p=\{R_1(AC), R_2(ED), R_3(AB)\}$

D. $p=\{R_1(ABC), R_2(ED), R_3(ACE)\}$

试题分析

本题考查如何求解候选关键字和对模式分解知识的掌握。

候选关键字使用 2.1.6 规范化理论中的图示法进行求解，对 R 关系模式画图如图 2-14 所示。

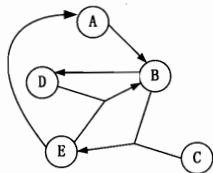


图 2-14 R 关系函数依赖示意图

图中 C 节点为 0 度节点，所以它必然被包含在候选关键字中，但仅有 C 节点并不能遍历全图，所以需要加入其他中间节点。若加入 B 节点，则 $BC \rightarrow E$ ， $E \rightarrow A$ ， $B \rightarrow D$ ，

能遍历全图。与此同时,加入 A、E 都能起到同样的效果。所以关系 R 有 3 个候选键: BC、EC, AC。

接下来是判断模式分解过程中的无损联接与保持函数依赖的问题。这个问题相对来说比较复杂。如果逐个判断每个选项的无损联接与保持函数依赖,无疑工作量是很大的。所以我们可以先观察这些选项有什么特点,通过观察发现 A 与 B 选项都存在单个字段的分解。在进行模式分解时,如果出现单字段,同时该字段未在其他分解的子关系模式中出现,并且函数依赖中有此字段的依赖关系,则说明此分解没有保持函数依赖。原因很简单,关于该字段的那个函数依赖,必然在分解中丢失了。所以 A 与 B 选项可以先排除。

然后判断 C 与 D 是否为无损联接。

对选项 C 构造初始的判定表,如表 2-14 所示。

表 2-14 模式分解 C 选项初始判定表

分解的关系模式	A	B	C	D	E
R ₁ (AC)	a ₁	b ₁₂	a ₃	b ₁₄	b ₁₅
R ₂ (ED)	b ₂₁	b ₂₂	b ₂₃	a ₄	a ₅
R ₃ (AB)	a ₁	a ₂	b ₃₃	b ₃₄	b ₃₅

由于 $A \rightarrow B$, 属性 A 的第 1 行和第 3 行相同,可以将第 1 行 b₁₂ 改为 a₂; 又由于 $B \rightarrow D$, 属性 B 的第 1 行和第 3 行相同,所以需要将属性 D 第 1 行 b₁₄ 和第 3 行 b₃₄ 改为同一符号,即取行号值最小的 b₁₄。修改后的判定表如表 2-15 所示。

表 2-15 模式分解 C 选项修改判定表

分解的关系模式	A	B	C	D	E
R ₁ (AC)	a ₁	a ₂	a ₃	b ₁₄	b ₁₅
R ₂ (ED)	b ₂₁	b ₂₂	b ₂₃	a ₄	a ₅
R ₃ (AB)	a ₁	a ₂	b ₃₃	b ₁₄	b ₃₅

反复检查函数依赖集 F,无法修改上表,所以选项 C 是有损联接的。

对选项 D 构造初始的判定表,如表 2-16 所示。

表 2-16 模式分解 D 选项初始判定表

分解的关系模式	A	B	C	D	E
R ₁ (ABC)	a ₁	a ₂	a ₃	b ₁₄	b ₁₅
R ₂ (ED)	b ₂₁	b ₂₂	b ₂₃	a ₄	a ₅
R ₃ (ACE)	a ₁	b ₃₂	a ₃	b ₃₄	a ₅

由于 $A \rightarrow B$, 属性 A 的第 1 行和第 3 行相同,可以将第 3 行 b₃₂ 改为 a₂; $E \rightarrow A$, 属性 E 的第 2 行和第 3 行相同,可以将属性 A 第 2 行 b₂₁ 改为 a₁; $AC \rightarrow E$, 属性 E 的第 2 行和第 3 行相同,可以将属性 E 第 1 行 b₁₅ 改为 a₅; $B \rightarrow D$, 属性 B 的第 1 行和第 3 行相同,所以需要将属性 D 第 1 行 b₁₄ 和第 3 行 b₃₄, 改为同一符号,即取行号值

最小的 b_{14} 。 $E \rightarrow D$ ，属性 E 的第 1~3 行相同，可以将属性 D 第 1 行 b_{14} 和第 3 行 b_{34} 改为 a_4 。修改后的判定表如表 2-17 所示。

表 2-17 模式分解 D 选项修改判定表

分解的关系模式	A	B	C	D	E
$R_1(ABC)$	a_1	a_2	a_3	a_4	a_5
$R_2(ED)$	a_1	b_{22}	b_{23}	a_4	a_5
$R_3(ACE)$	a_1	a_2	a_3	a_4	a_5

由于上表第一行全为 a，故分解无损。

所以本空应选 D。

试题答案

(24) C 或 D

(25) D

2.3 实战练习题

- (1) 不属于关系数据库管理系统。

(1) A. Oracle B. MS SQL Server C. DB2 D. IMS

- 数据的物理独立性是指当数据库的 (2) 。

(2) A. 外模式发生改变时，数据的物理结构需要改变

B. 内模式发生改变时，数据的逻辑结构不需要改变

C. 外模式发生改变时，数据的逻辑结构不需要改变

D. 内模式发生改变时，数据的物理结构不需要改变

- 在数据库系统中，数据的完整性是指数据的 (3) 。

(3) 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

- 在数据库设计的需求分析阶段，业务流程一般采用 (4) 表示。

(4) A. 数据流图 B. E-R 图 C. 程序结构图 D. 功能模块图

● 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 (5) 属性; 为了将数据库模式设计得更合理, 对于家庭成员属性 (6) 。

(5) A. 简单 B. 复合 C. 多值 D. 派生

(6) 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

● 希赛公司网上销售管理系统的数据库部分关系模式如下所示。其中, 客户号唯一标识一位客户, 产品号唯一标识一件产品, 订单号唯一标识一份订单。一份订单必须且仅对应一位客户, 一份订单可由一到多条订单明细组成, 一位客户可以有多个订单。

客户 (客户号, 姓名, 性别, 地址, 邮编)

产品 (产品号, 名称, 库存, 单价)

订单 (订单号, 时间, 金额, 客户号)

订单明细 (订单号, 产品号, 数量)

订单关系模式的主键为__(7)__; 订单明细关系模式的主键为__(8)__; 其中订单的外键为__(9)__。

(7) A. 订单号 B. 客户号 C. (订单号, 客户号) D. (订单号, 时间)

(8) A. 订单号 B. 产品号 C. (订单号, 产品号) D. (订单号, 数量)

(9) A. 客户号, 订单明细的外键为订单号

B. 客户号, 订单明细的外键为订单号和产品号

C. 订单号, 订单明细的外键为产品号

D. 订单号, 订单明细的外键为订单号和产品号

● 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)3个实体之间的关系如表2-18、表2-19和表2-20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。

表 2-18 部门表

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

表 2-19 商品表

商品号	商品名称	单价	库存量
30023	微机	4800	26
30024	打印机	1650	7
⋮	⋮	⋮	⋮
30101	毛巾	10	106
30102	牙刷	3.8	288
⋮	⋮	⋮	⋮

表 2-20 职工表

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

续表

职工号	姓名	住址
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
⋮	⋮	⋮

表 2-21 统计结果

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
⋮	⋮	⋮	⋮

部门关系不属于第三范式的原因是 (10)。如果用户要求得到表 2-21 所示的结果, 需要 (11), 并增加关系模式 (12)。

- (10) 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

- (11) 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

- (12) 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)

- 若关系模式中存在非主属性对码的部分依赖, 则该关系模式属于 (13)。

- (13) A. 1NF B. 2NF C. 3NF D. BCNF

● 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 2-22 和表 2-23 所示。

表 2-22 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 2-23 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 (14) ; 仓库关系的主键是 (15) ; 仓库关系 (16) , 为了解决这一问题, 需要将仓库关系分解为 (17) 。

(14) A. 商品号 B. 商品名称 C. 生产商 D. 单价

(15) A. 仓库号, 地址 B. 仓库号, 电话
C. 仓库号, 商品号 D. 地址, 电话

(16) 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

(17) 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)

● 关系数据库中, 实现实体之间的联系是通过表与表之间的公共 (18) 。

(18) A. 索引 B. 存储 C. 元组 D. 属性

● 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 (19) 。

(19) 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

● 由于软、硬件故障可能造成数据库中的数据被破坏,数据库恢复就是__(20)__.可以有多种方法实现数据库恢复,如定期将数据库作备份;在进行事务处理时,对数据更新(插入、删除、修改)的全部有关内容写入__(21)__;当系统正常运行时,按一定的时间间隔,设立__(22)__,把内存缓冲区内容还未写入到磁盘中的有关状态记录到该文件中;当发生故障时,根据现场数据内容及相关文件来恢复系统的状态。

(20) A. 重新安装数据库管理系统和应用程序

B. 重新安装应用程序,并将数据库做镜像

C. 重新安装数据库管理系统,并将数据库做镜像

D. 在尽可能短的时间内,把数据库恢复到故障发生前的状态

(21) A. 日志文件 B. 程序文件 C. 检查点文件 D. 图像文件

(22) A. 日志文件 B. 程序文件 C. 检查点文件 D. 图像文件

● __(23)__是一种信息分析工具,能自动地找出数据仓库中的模式及关系。

(23) A. 数据集市 B. 数据挖掘 C. 预测分析 D. 数据统计

● 在数据库设计的__(24)__阶段进行关系规范化。

(24) A. 需求分析 B. 概念设计 C. 逻辑设计 D. 物理设计

● 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)。

a. 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”唯一标识 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__(25)__。

(25) 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

b. 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:

SELELCT 产品名称, SUM(产品数量)

FROM P, I

WHERE P.产品号 = I.产品号 __(26)__

(26) 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

• 若对关系 $R(A, B, C, D)$ 和 $S(C, D, E)$ 进行关系代数运算, 则表达式 $\pi_{3,4,7}(\sigma_{4<5}(R \times S))$ 与 (27) 等价。

(27) A. $\pi_{C,D,E}(\sigma_{D<C}(R \times S))$ B. $\pi_{R,C,R,D,E}(\sigma_{R.D<S.C}(R \times S))$ C. $\pi_{C,D,E}(\sigma_{R.D<S.C}(R \times S))$ D. $\pi_{R,C,R,D,E}(\sigma_{D<C}(R \times S))$

企业战略数据模型可分为两种类型: (28) 描述日常事务处理中的数据及其关系; (29) 描述企业管理决策者所需信息及其关系。

(28) 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

(29) 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

2.4 练习题解析

试题 1 分析

题目给出的几种数据库管理系统中: Oracle、MS SQL Server、DB2 较为常见, 它们都属于关系型数据库管理系统。而 IMS 不是关系数据库管理系统, 它是 IBM 公司推出的层次型数据库管理系统。

试题 1 答案

(1) D

试题 2 分析

不同的数据库产品支持不同的数据模型, 使用不同的数据库语言, 建立在不同的操作系统上。数据的存储结构也各不相同, 但体系结构基本上都具有相同的特征, 采用“三级模式和两级映射”。

数据库系统在三级模式之间提供了两级映象: 模式/内模式映象、外模式/模式映象。正因为这两级映射保证了数据库中的数据具有较高的逻辑独立性和物理独立性。

数据的独立性是指数据与程序独立, 将数据的定义从程序中分离出去, 由 DBMS 负责数据的存储, 从而简化应用程序, 大大减少应用程序编制的工作量。数据的独立性是由 DBMS 的二级映像功能来保证的。数据的独立性包括数据的物理独立性和数据的逻辑独立性。

数据的物理独立性: 是指当数据库的内模式发生改变时, 数据的逻辑结构不变。由于应用程序处理的只是数据的逻辑结构, 这样物理独立性可以保证, 当数据的物理结构改变了, 应用程序不用改变。但是, 为了保证应用程序能够正确执行, 需要修改概念模式/内模式之间的映像。

数据的逻辑独立性：是指用户的应用程序与数据库的逻辑结构是相互独立的。数据的逻辑结构发生变化后，用户程序也可以不修改。但是，为了保证应用程序能够正确执行，需要修改外模式/概念模式之间的映像。

试题2 答案

(2) B

试题3 分析

数据库完整性是指数据库中数据的正确性、有效性和相容性（一致性）。数据库完整性由各种各样的完整性约束来保证，因此可以说数据库完整性设计就是数据库完整性约束的设计。数据库完整性约束可以通过 DBMS 或应用程序来实现，基于 DBMS 的完整性约束作为模式的一部分存入数据库中。通过 DBMS 实现的数据库完整性按照数据库设计步骤进行设计，而由应用软件实现的数据库完整性则纳入应用软件设计。

试题3 答案

(3) A

试题4 分析

在数据库设计的需求分析阶段，参与分析的主要人员是系统分析师和用户，由于数据库应用系统是面向企业和部门的具体业务，系统分析师一般并不了解，而用户一般不具有系统分析的能力，这就需要双方进行有效的沟通，使得设计人员对用户的各项业务了解和熟悉，进行分析和加工，将用户的业务转换成为设计人员所需要的信息组织，即以规范化的方式进行整理，形成对业务流程描述的文档数据和数据描述的文档数据字典，故选项 A 正确；E-R 图是概念设计阶段的文档，故选项 B 错误；程序结构图和功能模块图是应用程序设计阶段的文档，不是数据库设计的需求分析阶段的文档，故选项 C、D 错误。

试题4 答案

(4) A

试题5~6 分析

简单属性是原子的，不可再分的。复合属性可以细分为更小的部分（即划分为别的属性）。有时用户希望访问整个属性，有时希望访问属性的某个成分，那么在模式设计时可采用复合属性。本题职工实体集 Employee 的通信地址可以进一步分为邮编、省、市、街道。（5）空的正确答案为 B。

我们所举的例子中，定义的属性对于一个特定的实体都只有单独的一个值。例如，对于一个特定的职工，只对应一个职工号、职工姓名，这样的属性称为多值属性。为了将数据库模式设计得更合理，（6）空的家庭成员属性应该将职工亲属的姓名加上职工号设计成为一个独立的实体。

试题5~6 答案

(5) B

(6) D

试题 7~9 分析

根据试题描述,“单号唯一标识一份订单”所以,订单关系模式的主键为“单号”在订单关系模式中,“一份订单必须且仅对应一位客户”,而在客户关系模式中,“户号唯一标识一位客户”也就是说,“户号”客户关系模式的主键,因此,“户号”订单关系模式的外键。

因为“份订单可由一到多条订单明细组成”也就是说,在订单明细关系模式中,“单号”可以重复的,因此,需要与“品号”合起来作为主键。又因为“单号”单关系模式的主键,“品号”产品关系模式的主键(因为“号唯一标识一件产品”,所以,“单号”“品号”是订单明细关系模式的外键。

试题 7~9 答案

(7) A

(8) C

(9) B

试题 10~12 分析

在部门(部门号,部门名,负责人,电话)关系模式中,部门号是主码,所以不存在对码的部分依赖问题。根据试题假设,每个部门有一位负责人和一部电话,也就是说,存在“责人→电话”样的函数依赖,即存在对主码的传递函数依赖。

在表 2-21 中,有职工号、姓名、部门、月销售额 4 个属性,其中职工号和姓名可以由表 2-20 得出,但是,由表 2-19 至表 2-20 无法得出职工所在的部门,因此,应该在表 2-20 中增加“部门号”属性。同时,要计算出职工的月销售总额,还需要一个表来记录职工销售的数据。该表需要包括的属性有职工号、商品号、日期、数量。

试题 10~12 答案

(10) C

(11) D

(12) A

试题 13 分析

若关系模式 R 的每一个分量是不可再分的数据项,则关系模式 R 属于第一范式(1NF);若关系模式属于 1NF,且每一个非主属性完全依赖于码,则关系模式 R 属于 2NF。换句话说,当 1NF 消除了非主属性对码的部分依赖,则该关系模式属于 1NF。

试题 13 答案

(13) A

试题 14~17 分析

从试题中所给出的表格来看,商品关系的主键是商品号,仓库关系的主键是(仓库号,商品号)。显然,仓库关系存在冗余、修改操作的不一致,以及插入异常和删除异常。例如,仓库号为“01”的商品有 3 种,其地址就要重复 3 次,故存在冗余。为了解决仓库关系模式存在的问题,就需要进行模式分解,其中(45)空选项 A 存在的问题是仓库 2 不属于第三范式,因为存在非主属性对键的部分函数依赖,即“仓库号→电话”。选项 B 存在的问题是分解属于有损联接,即分解的新关系模式仓库 1 和仓库 2 无法恢复到原关系。选项 C 分解存在的问题与 A 类同。选项 C 分解即保持函数依赖,又是无损联接。

试题 14~17 答案

(14) A

(15) C

(16) D

(17) D

试题 18 分析

关系数据库中,实现实体之间的联系是通过表与表之间的公共属性。

试题 18 答案

(18) D

试题 19 分析

选项 A 是错误的,因为具有 DBA 特权的用户可操作数据库的所有资源。

选项 B 是错误的,因为选项 B 是将表 EMP 的查询权限赋予该用户,即全部记录,而题目只允许某用户查询表 EMP 中的部分记录。

选项 C 是错误的,因为编写查询表 EMP 的存储过程仍然是查询表 EMP 的所有记录。

选项 D 是正确的,因为是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。

试题 19 答案

(19) D

试题 20~22 分析

数据库恢复是指在数据库中的数据被破坏时,在尽可能短的时间内,把数据库恢复到故障发生前的状态。

在进行事务处理时,对数据更新(插入、删除、修改)的全部有关内容写入日志文件;当系统正常运行时,按一定的时间间隔,设立检查点文件,把内存缓冲区内容还未写入到磁盘中的有关状态记录到该文件中;当发生故障时,根据现场数据内容及相关文件来恢复系统的状态。

试题 20~22 答案

(20) D

(21) A

(22) C

试题 23 分析

本题考查的是数据挖掘的基本概念,关于数据挖掘的说明,请参看“2.1.9 数据仓库与数据挖掘”。

试题 23 答案

(23) B

试题 24 分析

本题考查数据库设计基础知识。

数据库设计通常分为 4 个阶段:需求分析、概念设计、逻辑设计、物理设计。

需求分析阶段的主要工作是分析当前和未来应用的数据要求。概念设计阶段将完成 E-R 建模。逻辑设计阶段将 E-R 模型转换成关系模式,并使用规范化理论对模式进行优化处理。物理设计阶段通常是完成将逻辑设计产生的关系模式结合 DBMS 的特性,形成能操作的数据库的过程。

试题 24 答案

(24) C

试题 25~26 分析

本题考查关系的完整性约束和 SQL 的基本知识及应用。

第(25)空考查数据库完整性约束。首先需要了解实体完整性、参照完整性和用户定义完整性的概念。

实体完整性:实体完整性要求主键中的任一属性不能为空,所谓空值是“不知道”或“无意义”的值。之所以要保证实体完整性,主要是因为,在关系中,每个元组的区分是依据主键值的不同,若主键值取空值,则不能标明该元组的存在。例如,对于学生关系 S (Sno, Sname, Ssex),其主键为 Sno,在插入某个元组时,就必须要求 Sno 不能为空。更加严格的 DBMS,则还要求 Sno 不能与已经存在的某个元组的 Sno 相同。

参照完整性:若基本关系 R 中含有与另一基本关系 S 的主键 PK 相对应的属性组 FK (FK 称为 R 的外键),则参照完整性要求,对 R 中的每个元组在 FK 上的值必须是 S 中某个元组的 PK 值,或者为空值。参照完整性的合理性在于,R 中的外键只能对 S 中的主键引用,不能是 S 中主键没有的值。例如,对于学生关系 S (Sno, Sname, Ssex)和选课关系 C (Sno, Cno, Grade)两个关系,C 中的 Sno 是外键,它是 S 的主键,若 C 中出现了某个 S 中没有的 Sno,即某个学生还没有注册,却已有了选课记录,这显然是不合理的。

用户定义的完整性:实体完整性和参照完整性适用于任何关系型 DBMS。除此之外,不同的数据库系统根据其应用环境的不同,往往还需要一些特殊的约束条件。用户定义的完整性就是针对某一具体数据库的约束条件,反映某一具体应用所涉及的数据必须满足的语义要求。

从以上概念说明可以看出“仓库关系 W 中的“负责人”引用员工关系的员工号”属于参照完整性,“库存关系 I 中的“仓库号,产品号”唯一标识 I 中的每一个记录”属于实体完整性,“员工关系 E 中的职称为“工程师”的月薪不能低于 3500 元”属于用户自定义完整性。

第(26)空要求“得到每种产品的名称和该产品的总库存量”,要达到该效果需要对数据表的数据进行分组统计。由于现在是要根据产品名称来统计库存量,所以分组的依据应是产品名称,所以应使用“GROUP BY 产品名称”。

试题 25~26 答案

(25) B

(26) C

试题 27 分析

本题考查关系代数运算方面的基础知识。

题目要求计算关系代数表达式 $\pi_{3,4,7}(\sigma_{4<5}(R \times S))$ 的结果集, 其中, $R \times S$ 的属性列名分别为: R.A, R.B, R.C, R.D, S.C, S.D 和 S.E, $\sigma_{4<5}(R \times S)$ 的含义是从 $R \times S$ 结果集中选取第 4 个分量(R.D)小于第 5 个分量(S.C)的元组, 故 $\sigma_{4<5}(R \times S)$ 与 $\sigma_{R.D < S.C}(R \times S)$ 等价。 $\pi_{3,4,7}(\sigma_{4<5}(R \times S))$ 的含义是从 $\sigma_{4<5}(R \times S)$ 结果集中选取第 3 列 R.C、第 4 列 R.D 和第 7 列 S.E (或 E), 故 $\pi_{3,4,7}(\sigma_{4<5}(R \times S))$ 与 $\pi_{R.C, R.D, E}(\sigma_{R.D < S.C}(R \times S))$ 等价。需要说明的是第 3 列 R.C 不能简写为 C, 因为关系 S 的第一列属性名也为 C, 故必须标上关系名加以区别; 同样, 第 4 列 R.D 也不能简写为 D, 因为关系 S 的第二列属性名也为 D, 故必须标上关系名加以区别。

试题 27 答案

(27) B

试题 28~29 分析

企业中使用的数据模型分两大类, 一类针对处理日常事务的应用系统, 即数据库; 另一类针对高层决策分析, 即数据仓库。

试题 28~29 答案

(28) B

(29) C

3

第 3 章

计算机硬件基础及嵌入式系统设计

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

- 嵌入式系统的特点。
- 嵌入式系统的硬件组成与设计。
- 嵌入式系统应用软件及开发平台。
- 嵌入式系统网络。
- 嵌入式系统数据库。
- 嵌入式操作系统与实时操作系统。

3.1 考点突破

计算机硬件基础及嵌入式系统设计部分从历年的考试情况来看，考点主要集中在以下几个方面：存储、串/并转换、寄存器、传输等方面。虽然计算机硬件基础部分在考纲中未做要求，但每次考试仍会考查，所以本章也会花一些篇幅对相关内容进行介绍。另外值得注意的是，嵌入式系统相关知识不仅在上午综合知识会考查，案例分析也会考查。

3.1.1 历年考试情况分析

在历年的考试试题中，有关嵌入式系统设计的试题如表 3-1 所示。

表 3-1 嵌入式系统设计试题分布表

题号	2009.11	2010.11	2011.11
9	中断	寄存器	CISC 与 RISC
10		内存容量计算	Cache 特点

(续表)

题号	2009.11	2010.11	2011.11
11	存储器	磁盘文件读取	虚拟存储器
12	串/并、并/串转换	异步传输	总线

按照知识点进行总结和归类的试题分布情况如表 3-2 所示。

表 3-2 嵌入式系统设计知识点归类表

知识点	2009.11	2010.11	2011.11
嵌入式系统概述	0	0	0
嵌入式操作系统	0	0	0
嵌入式系统分析与设计	0	0	0
嵌入式低功耗设计	0	0	0
存储系统	1	3	2
其他	3	1	2
合计	4	4	4

从表 3-2 中可以看出,嵌入式系统设计方面的内容在历年的考试中分值非常稳定,一直稳定维持在 4 分。所占分数比例的趋势如图 3-1 所示。

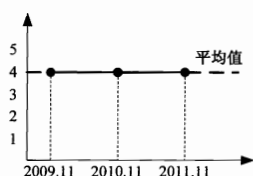


图 3-1 嵌入式系统设计历年试题比例趋势图

3.1.2 嵌入式系统概述

嵌入式系统是一种以应用为中心,以计算机技术为基础,可以适应不同应用的功能、可靠性、成本、体积和功耗等方面的要求,集可配置、可裁减的软硬件于一体的专用计算机系统。它具有很强的灵活性,主要由嵌入式硬件平台、相关支撑硬件、嵌入式操作系统、支撑软件和应用软件组成。其中,嵌入性、专用性和计算机系统是嵌入式系统的 3 个核心要素。

归纳起来,典型的嵌入式系统具有以下特点:

1. 系统专用性强

嵌入式系统是针对具体应用的专门系统。它的个性化很强,软件和硬件结合紧密。一般要针对硬件进行软件的开发和移植,根据硬件的变化和增减对软件进行修改。由于嵌入式系统总是用来完成某一特定任务的,整个系统与具体应用有机地结合在一起,升级换代也以更换整个产品的方式进行,因此,嵌入式产品一旦进入市场,一般具有较长的生命周期。

2. 系统实时性强

很多嵌入式系统对外来事件要求在限定的时间内及时做出响应，具有实时性。根据实时性的强弱，通常将嵌入式系统分为实时嵌入式系统和非实时嵌入式系统，其中大部分为实时嵌入式系统。

3. 软硬件依赖性强

嵌入式系统的专用性决定了其软硬件的互相依赖性很强，两者必须协同设计，以达到共同实现预定功能的目的，并满足性能、成本和可靠性等方面的严格要求。

4. 处理器专用

嵌入式系统的处理器一般是为某一特定目的和应用而专门设计的。通常具有功耗低、体积小和集成度高等优点，能够将许多在通用计算机上需要由板卡完成的任务和功能集成到芯片内部，从而有利于嵌入式系统的小型化和移动能力的增强。

5. 多种技术紧密结合

嵌入式系统通常是计算机技术、半导体技术、电力电子技术、机械技术与各行业的实际应用相结合的产物。通用计算机技术也离不开这些技术，但它们相互结合的紧密程度不及嵌入式系统。

6. 系统透明性

嵌入式系统在形态上与通用计算机系统差异甚大。它的输入设备往往不是常见的鼠标和键盘之类的设备，甚至没有输出装置，用户可能根本感觉不到它所使用的设备中有嵌入式系统的存在，即使知道，也不必关心嵌入式系统的相关情况。

7. 系统资源受限

嵌入式系统为了达到结构紧凑、高可靠性和低成本的目的，其存储容量、I/O 设备的数量和处理器处理能力都比较有限。

3.1.3 嵌入式操作系统

嵌入式操作系统（Embedded Operating System, EOS）是指运行在嵌入式系统上，支持嵌入式应用程序的操作系统，是用于控制和管理嵌入式系统中的硬件和软件资源、提供系统服务的软件集合。EOS 是嵌入式软件的一个重要组成部分，它的出现提高了嵌入式软件开发的效率和应用软件的可移植性，有力地推动了嵌入式系统的发展。

嵌入式操作系统有以下特点：

- **系统内核小。**由于嵌入式系统一般是应用于小型电子装置的，系统资源相对有限，所以内核较之传统的操作系统要小得多。例如，Enea 公司的 OSE 分布式系统，内核只有 5KB。
- **专用性强。**嵌入式系统的个性化很强，其中的软件系统和硬件的结合非常紧密，一般要针对硬件进行系统的移植，即使在同一品牌、同一系列的产品中也需要根据系统硬件的变化和增减不断进行修改。同时针对不同的任务，往往需要对系统进行较大的更改，程序的编译、下载要和系统相结合，这种修

改和通用软件的“升级”完全是两个概念。

- 系统精简。嵌入式系统一般没有系统软件和应用软件的明显区分，不要求其功能设计及实现上过于复杂，这样，一方面利于控制系统成本，另一方面也利于实现系统安全。
- 高实时性的系统软件是嵌入式软件的基本要求。而且软件要求固态存储，以提高速度；软件代码要求高质量和高可靠性。
- 嵌入式软件开发要想走向标准化，就必须使用多任务的操作系统。嵌入式系统的应用程序可以没有操作系统，直接在芯片上运行；但是为了合理地调度多任务、利用系统资源、系统函数，以及和专家库函数接口，用户必须自行选配 RTOS (RealTime Operating System) 开发平台，这样才能保证程序执行的实时性、可靠性，并减少开发时间，保障软件质量。
- 嵌入式系统开发需要开发工具和环境。由于其本身不具备自主开发能力，即使设计完成以后，用户通常也是不能对其中的程序功能进行修改的，必须有一套开发工具和环境才能进行开发，这些工具和环境一般是基于通用计算机上的软硬件设备，以及各种逻辑分析仪、混合信号示波器等。开发时往往有主机和目标机的概念，主机用于程序的开发，目标机作为最后的执行机，开发时需要交替结合进行。

1. VxWorks

VxWorks 是美国 WindRiver 公司的产品，是目前嵌入式系统领域中应用很广泛、市场占有率比较高的嵌入式操作系统。VxWorks 实时操作系统由 400 多个相对独立、短小精悍的目标模块组成，用户可根据需要选择适当的模块来裁剪和配置系统；提供基于优先级的任务调度、任务间同步与通信、中断处理、定时器和内存管理等功能，内建符合 POSIX（可移植操作系统接口）规范的内存管理，以及多处理器控制程序；并且具有简明易懂的用户接口，在核心方面甚至可以微缩到 8KB。

2. μ Clinux

μ Clinux 是一种优秀的嵌入式 Linux 版本，其全称为 Micro-Control Linux，从字面意思看是指微控制 Linux。同标准的 Linux 相比， μ Clinux 的内核非常小，但是它仍然继承了 Linux 操作系统的主要特性，包括良好的稳定性和移植性、强大的网络功能、出色的文件系统支持、标准丰富的 API，以及 TCP / IP 协议等。因为没有 MMU 内存管理单元，所以其多任务的实现需要一定的技巧。

3. VxWorks 与 μ Clinux 比较

VxWorks 与 μ Clinux 是目前两大主流嵌入式操作系统，它们的调度机制与通信机制方面各有特色，详细情况如表 3-3 所示。

表 3-3 VxWorks 与 μ Clinux 对比

操作系统	VxWorks	μ Clinux
内核抢占	是	是
优先级数量	256	64

(续表)

操作系统	VxWorks	μClinux
优先级变化	动态	动态
调度方法	基于优先级抢占式调度	基于固定优先级抢占式调度
同优先级调度	有	无
任务数量	256	64
时间可确定性	是	是
同步	信号量、互斥信号量、事件标志	信号量、互斥信号量、事件标志
通信量	消息队列、共享内存、管道	邮箱、消息队列
避免优先级反转	优先级继承	优先级置顶

3.1.4 嵌入式系统分析与设计

本节以一个多用途通用控制平台为例，对嵌入式系统的开发与设计过程进行描述。该平台是一个面向工业、企事业单位、生活社区和普通家庭用户，集成了移动通信、互联网技术、无线局域网和无线传感器网络等先进的网络技术，以及视频信息采集与处理技术的多用途控制平台。

对于家庭用户来说，可以通过移动通信网络随时随地了解家中状况，可以远程控制家里的电器设备，并且在设备出现了故障或紧急情况，该通用控制平台可以主动地通过手机或电话网通知用户或火警等相关部门。特别是出现火灾等极端状况时，控制平台可以启动有关设备进行扑救，同时将相关数据和现场图像或视频资料发送出去，提供给救援人员参考，以便协助救援。对于企业用户，该系统可以设置在仓库、车间和实验室等地点，并且多个平台设备之间可以共享数据和相互协作。

1. 需求分析

嵌入式系统一般都是面向产品开发的，对嵌入式系统进行需求分析是一项复杂而费时的的工作。需求分析阶段最重要的成果之一就是系统规格说明书。对于不同的项目，该文档形式可以灵活变化，但基本包含物理尺寸、操作环境、存放环境、指示灯装置、无线电标准、无线电功率和频率、数据传输速率、存储器、平均无故障时间、功耗、电源适配器、散热、系统复位、协议等内容。在进行需求分析时，可以采用 UML 进行需求描述。有关这方面的详细知识，请阅读 7.1.5 节。

2. 系统架构设计

嵌入式系统的架构由硬件架构和软件架构组成，随着嵌入式软件复杂度的日益提高，系统的响应速度和可靠性等重要指标不再仅仅由硬件架构所决定，软件架构的影响也逐步增大。描述系统如何实现规格说明中定义的功能是系统架构设计的主要目的。但是，在设计嵌入式系统的架构时，很难将软件和硬件完全分开。通常的处理方法是先考虑系统的软件架构，然后再考虑其硬件实现。系统架构的描述必须符合功能上和非功能上的需求，不仅要体现所要求的功能，还要满足成本、速度和功耗等非功

能约束。

软件架构的选择和设计很大程度上取决于嵌入式系统的架构，主要考虑以下几点因素：

- **系统的实时性要求。**对于硬实时系统，必须进行严格的时序分析。
- **采用的设计模型。**对于简单的中小型嵌入式系统，可以采用“先硬件后软件”的方法；对于复杂、大型嵌入式系统，必须采用软硬件协同设计的方法，还要充分考虑软件和硬件两个方面的互相制约和影响。
- **是否需要 EOS。**对于复杂的和日后需要继续开发和移植的系统，最好采用 EOS 来增强系统的可扩展性。

在所有满足系统功能和性能要求的方案中，应该优先选用最简单的架构。

3. 硬件子系统设计

嵌入式系统的开发环境由 4 个部分组成，分别是目标硬件平台、EOS、编程语言和开发工具，其中处理器和操作系统的选择应当考虑更多的因素，避免错误的决策影响项目的进度。

嵌入式系统设计的主要挑战是如何使互相竞争的设计指标同时达到最佳化。设计人员必须对各种处理器技术和 IC 技术的优缺点加以取舍。一般而言，处理器技术与 IC 技术无关，也就是说，任何处理器技术都可以使用任何 IC 技术来实现，但是最终器件的性能、NRE（Non-Recurring Engineering，一次性工程）成本、功耗和大小等指标会有很大的差异。通用的可编程技术提供了较大的灵活性，降低了 NRE 成本，建立产品样机与上市的时间较快；定制的技术能够提供较低的功耗、较好的性能、更小的体积和大批量生产时的低成本。根据这些原则，设计人员便可以对采用的处理器技术和处理器做出合理选择。

一般，全定制商用“通用处理器+软件”是大多数情况下都适用的一个选择。根据用户的需求和项目的需要，选择合适的通用嵌入式处理器，选择时需要考虑处理器的速度、技术指标、开发人员对处理器的熟悉程度、处理器的 I/O 功能是否满足系统的需求、处理器的调试、处理器制造商的支持可信度和封装形式等指标。

在硬件子系统的设计中，首先，将硬件划分为部件（或模块），并绘制部件连接框图；其次，对每个部件进行细化，将系统分成更多个可管理的小块，可以被单独实现。通常，系统的某些功能既可用软件实现也可用硬件实现，没有一个统一的方法指导设计人员决定功能的软、硬件分配，但是可以根据系统约束清单，在性能和成本之间进行权衡。

嵌入式系统中接口电路的设计需要考虑的因素主要有电平匹配问题、驱动能力和干扰问题。设计时需要注意 I/O 端口、硬件寄存器、内存映射、硬件中断和存储器空间分配等方面。总之，硬件设计人员应该给软件设计人员更多、更详细的信息，便于进行软件设计和开发。

4. 软件子系统设计

根据需求分析阶段的规格说明文档，确定系统计算模型，然后，对软件部分进行

分析与设计。嵌入式系统的软件设计需要根据应用的实时性要求、应用场合、可用资源情况等诸多约束,进行合理的任务划分,这是嵌入式系统应用软件设计的关键所在,直接影响软件设计的质量。在实时系统中,一个应用系统通常由一系列的任务构成,每个任务完成一个独立的功能,这些任务互相协作来共同实现系统的整体功能。

首先,明确系统的实时性指标,究竟是软实时还是硬实时,最坏情况下的实时时限是多少等;然后,确定任务大体的数目。任务划分的粒度和数目的合理性直接决定着调度开销和任务之间通信开销,以及系统响应时间的重要决定因素,这需要充分考虑极端情况下的各种因素。实时系统的软件应该尽可能的简化,过于复杂的软件设计会增加系统的复杂性,从而降低可靠性。

然后,进行可调度性分析。对已经划分好的任务根据实现的功能特点进行分类,识别出具有硬实时性要求的所有关键任务,并给所有的任务赋予适当任务优先级,根据每个任务执行的时间估计值在考虑调度开销余量的条件下进行粗略的可调度性分析,从而确保所有关键任务都满足调度时限。根据分析结果再对所有的任务进行适当的分裂与重组。对于周期相同的所有任务的功能进行适当的组合形成一个单独的任务,以降低事件分发机制的开销;对于若干固定顺序执行的任务要合成一个单一任务,避免同步开销;将若干有相同的事件源触发的任务也合并成一个任务,以减小事件分发机制的开销;对于计算密集型或以数据处理为主而不进行 I/O 操作的功能独立出来,由一个优先级的任务来实现;对任务之间的同步与互斥机制进行分析,尽量避免大粒度的互斥锁。

最后,在具体的实现上,根据系统特点确定任务的优先级,任务优先级分配的一般原则如下:

- **与中断的关联性。**凡是与中断服务程序相关联的任务应该安排尽量高的优先级,以保证及时处理异步事件,提高系统的实时响应能力。否则,CPU 可能被其他优先级高的任务长期占用,使得第二次中断发生时,上一次中断还没有处理,从而产生信号丢失,甚至是外部设备失败。
- **紧迫性。**越是紧迫的任务对响应时间的要求就越是严格,对于所有的紧迫任务,根据它们响应时间的先后顺序,安排优先级。
- **任务的频繁性。**对于周期性的任务,执行越频繁,则周期越短,允许耽搁的时间也就越短,故相应的优先级也应该越高,以保证处理的及时性。并且,任务的处理时间越短,优先级也应越高。
- **传递性。**信息处理流程的前驱任务的优先级应该高于后继任务的优先级,例如,数据的采集任务优先级应该高于数据处理任务的优先级。

5. EOS 的选择

在选择 EOS 时,也需要做多方面的考虑:

(1) EOS 的功能

根据项目需要的 EOS 功能来选择 EOS 产品,要考虑系统支持 EOS 的全部功能还是部分功能、是否支持文件系统和人机界面、是实时系统还是分时系统,以及系统是否可裁剪等因素。

(2) 配套开发工具

有些 RTOS 只支持该系统供应商的开发工具。也就是说,还必须向 EOS 供应商获取编

译器和调试器等；有些EOS使用广泛，且有第三方工具可用，因此，选择的余地比较大。

(3) EOS的可移植性

EOS到硬件的移植是一个重要的问题，是整个系统能否按期完工的关键因素，因此，要选择那些可移植性程度高的EOS，从而避免EOS难以向硬件移植而带来的种种困难，加速系统的开发进度。

(4) EOS的内存需求

均衡考虑是否需要额外RAM或EEPROM来迎合EOS对内存的较大要求。有些EOS对内存的要求是目标相关的，例如，Tornado/VxWorks等，开发人员能按照应用需求分配所需的资源，而不是为EOS分配资源。

(5) EOS附加软件包

EOS是否包含所需的软件部件，例如，网络协议栈、文件系统和各种常用外设的驱动等。

(6) EOS的实时性如何

有些EOS只能提供软实时性能，对于需要达到硬实时性能要求的系统就不适用；有些EOS即可满足软实时要求，也能满足硬实时要求，例如，MS Windows CE 2.0等。

(7) EOS的灵活性

EOS是否具有可裁剪性，即能否根据实际需要进行系统功能的裁剪。有些EOS具有较强的可裁剪性，例如，嵌入式Linux和ECos等。

6. 编程语言的选择

对于嵌入式系统编程语言的选择，需要考虑通用性、可移植性、执行效率和可维护性等方面。

(1) 通用性

随着微处理器技术的不断发展，其功能越来越专用，种类越来越多，但不同种类的微处理器都有自己专用的汇编语言。这就为系统开发人员设置了一个巨大的障碍，使系统编程更加困难，软件复用无法实现；而高级语言一般和具体机器的硬件结构联系较少，比较流行的高级语言对多数微处理器都有良好的支持，通用性较好。

(2) 可移植性

由于汇编语言和具体的微处理器密切相关，为某个微处理器设计的程序不能直接移植到另一个不同种类的微处理器上使用，因此，可移植性差；高级语言对所有微处理器都是通用的，因此，程序可以在不同的微处理器上运行，可移植性较好。这是实现软件复用的基础。

(3) 执行效率

一般来说，越是高级的语言，其编译器和开销就越大，应用程序也就越大、越慢。但单纯依靠低级语言（如汇编语言）来进行应用程序的开发，带来的问题是编程复杂、开发周期长。因此，存在一个开发时间和运行性能之间的权衡。

(4) 可维护性

通常,低级语言程序的可维护性不高。高级语言程序往往是模块化设计,各个模块之间的接口是固定的。因此,当系统出现问题时,可以很快地将问题定位到某个模块内,并尽快得到解决。另外,模块化设计也便于系统功能的扩展和升级。

在嵌入式系统的分析与设计过程中,需要完成一系列的文档,例如,技术文件目录、技术任务书、技术方案报告、产品规格、技术条件、设计说明书、试验报告和总结报告等,这些文档对完成产品设计和维护相当重要,需要按照通用计算机系统开发的文档管理标准进行管理。

3.1.5 嵌入式低功耗设计

嵌入式系统通常都是一些移动设备,不能像通用计算机系统那样,长期插接供电电源,因此,在进行嵌入式系统设计时,必须考虑其功耗问题。低功耗设计是嵌入式系统设计中的难点,是一个系统化的综合问题,必须从软件和硬件两个方面全面考虑,才能真正有效地降低功耗。

1. 基于硬件的低功耗设计

对于常用的典型 CMOS 集成电路,其功耗由静态功耗、静态漏电流功耗、内部短路功耗和动态功耗组成。通常,在 CMOS 器件的整体功耗中,动态功耗占 70%~90%,静态功耗与静态漏电流功耗一般不大于 2%,内部短路功耗在 10%~30%之间。从硬件方面降低器件工作时的功耗,主要从降低工作电压方面考虑。对于嵌入式系统设计人员来说,基于硬件的低功耗设计应该从如下几个方面考虑:

(1) 板级电路低功耗设计

板级电路的低功耗设计主要围绕处理器的低功耗特性和外围芯片的工作特点,设计处理器的供电电路和外围芯片的电源控制电路,处理器的供电设计允许改变其内核的输入电压,达到降低功耗的目的,外围芯片的电源控制则允许处理器能够根据实际需要对工作空闲的外围芯片的电源开启和关闭,从而降低其功耗。

(2) 选择低功耗处理器

处理器是嵌入式系统不可缺少的核心部件,选择处理器时除了参考其功能、性能、接口和指令集等指标外,还应该考虑功耗特性,在满足正常工作的前提下,尽量选择低电压工作的处理器,或者选择可以动态调整电压和工作频率的处理器。对于同样核心的处理器或其他外围电路,应该选用制造工艺更先进的器件。

(3) 总线的低功耗设计

对于嵌入式系统来说,总线的宽度越宽,功耗就越大,在可能情况下,应该优先选用低功耗的总线器件,例如,低电压差分信号器件(Low Voltage Differential Signal, LVDS)。LVDS 是一种小振幅差分信号技术,使用非常低的幅度信号通过一对差分信号传输数据,允许单个信道传输速率达到每秒数百兆位,因其特有的恒流源模式驱动,所以产生的噪声极低,功耗很小。随着 LVDS 器件信号速率的进一步提升,品种越来越丰富,在嵌入式系统设计中的应用也会越来越广泛。

(4) 接口驱动电路的设计

设计接口驱动电路时,要选用静态电流低的外围芯片,还要考虑上拉/下拉电阻的选取、悬空引脚的处理和缓冲器的使用等。电阻大小应该仔细计算或采用模拟工具进行仿真,悬空的引脚应该接地或接电源。接口芯片驱动能力如果能够满足需要,尽量减少使用缓冲器。

(5) 分区分时供电技术

嵌入式处理器的工作电压与外围芯片的工作电压通常不一致,可以将它们分别划在不同的供电区域内,并且可以部分关闭空闲分区内的电源,来降低系统功耗。

2. 基于软件的低功耗设计

在嵌入式系统的设计过程中,通过软件的设计优化,可在一定程度上降低系统的功耗,主要的技术措施有如下几种:

(1) 编译优化技术

对于实现同样的功能,不同实现算法所消耗的时间不同;使用的指令不同,消耗功率也不同。编译器可以通过特定处理器单条指令的功耗基本开销、连续执行不同指令开销的估算模型进行优化。

(2) 软件与硬件的协同设计

可以通过软件与硬件的功能再分配,将用硬件实现的功能由软件实现,从而减少硬件电路的规模,进而达到降低功耗的目的。这需要将基于软件实现某种功能的所需要功耗与硬件电路实现同样功能产生的功耗进行对比,详细分析之后进行综合考虑。

(3) 算法优化

对于软件实现的功能,应尽量采用时间复杂度低的算法,例如,快速傅里叶变换和快速排序算法等,来降低算法执行时间,从而将低功耗。

3.1.6 存储系统

在计算机体系中,存储系统是分层的,从快到慢,层次分别有:寄存器、Cache、内存(主存)、外存(包括磁盘、U盘、光盘等)。

1. 寄存器

寄存器是中央处理器(CPU)内的重要组成部分。寄存器是有限存储容量的高速存储部件,它们可用来暂存指令、数据和位址。在中央处理器的控制部件中,包含的寄存器有指令寄存器(IR)、程序计数器(PC)、程序状态寄存器(PSW)等。在中央处理器的算术及逻辑部件中,包含的寄存器有累加器(AC)。

指令寄存器(IR)用来保存当前正在执行的一条指令。当执行一条指令时,先把它从内存取到数据寄存器(DR)中,然后再传送至IR。指令划分为操作码和地址码字段,由二进制数字组成。为了执行任何给定的指令,必须对操作码进行测试,以便识别所要求的操作。指令译码器就是做这项工作的。指令寄存器中操作码字段的输出就是指令译码器的输入。操作码一经译码后,即可向操作控制器发出具体操作的特定信号。

程序计数器 (PC) 存储的是将要执行的下一条指令的地址, 所以通常也叫做指令计数器。当执行指令时, CPU 将自动修改 PC 的内容, 以便使其保持的总是将要执行的下一条指令的地址。

程序状态寄存器 (PSW) 又称标志寄存器, 这是一个存放条件码标志、控制标志和系统标志的寄存器。

地址寄存器用来保存当前 CPU 所访问的内存单元的地址。由于在内存和 CPU 之间存在着操作速度上的差别, 所以必须使用地址寄存器来保持地址信息, 直到内存的读/写操作完成为止。当 CPU 和内存进行信息交换, 即 CPU 向内存存/取数据时, 或者 CPU 从内存中读出指令时, 都要使用地址寄存器和数据缓冲寄存器。同样, 如果我们把外围设备的设备地址作为像内存的地址单元那样来看待, 那么, 当 CPU 和外围设备交换信息时, 我们同样使用地址寄存器和数据缓冲寄存器。

累加寄存器 (AC) 通常简称为累加器, 它是一个通用寄存器。其功能是: 当运算器的算术逻辑单元 (ALU) 执行算术运算或逻辑运算时, 为 ALU 提供一工作区。累加寄存器暂时存放 ALU 运算的结果信息。显然, 运算器中至少要有一个累加寄存器。目前, CPU 中的累加寄存器多达 16 个、32 个, 甚至更多。当使用多个累加器时, 就变成通用寄存器堆结构, 其中任何一个既可存放源操作数, 也可存放结果操作数。在这种情况下, 需要在指令格式中对寄存器号加以编址。

2. Cache

由于在 CPU 与存储系统间存在着数据传送带宽的限制, 因此在其中设置了 Cache (高速缓冲存储器, 通常速度比内存快), 以提高整体效率。但由于其成本更高, 因此 Cache 的容量要比内存小得多。

(1) Cache 原理、命中率、失效率

使用 Cache 改善系统性能的主要依据是程序的局部性原理。通俗地说, 就是一段时间内, 执行的语句常集中于某个局部。而 Cache 正式将访问集中的内容放在速度更快的 Cache 上, 以提高性能。引入 Cache 后, CPU 在需要数据时, 先找 Cache, 如果没有再找内存。

如果 Cache 的访问命中率为 h (通常 $1-h$ 就是 Cache 的失效率), 而 Cache 的访问周期时间是 t_1 , 主存储器的访问周期时间是 t_2 , 则整个系统的平均访存时间就应该是:

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

从公式中可以看出, 系统的平均访存时间与命中率有着很密切的关系。灵活地应用这个公式, 可以计算出所有情况下的平均访存时间。

例如: 设某流水线计算机主存的读/写时间为 100ns, 有一个指令和数据合一的 Cache, 已知该 Cache 的读/写时间为 10ns, 取指令的命中率为 98%, 取数据的命中率为 95%。在执行某类程序时, 约有 1/5 指令需要存/取一个操作数。假设指令流水线在任何时候都不阻塞, 则设置 Cache 后, 每条指令的平均访存时间约为多少?

其实这是应用该公式的简单数学题:

$$(2\% \times 100\text{ns} + 98\% \times 10\text{ns}) + 1/5 \times (5\% \times 100\text{ns} + 95\% \times 10\text{ns}) = 14.7\text{ns}$$

(2) Cache 存储器的映射机制

CPU 发生访存请求时, 会先让 Cache 判断是否包括, 如果命中 (即包括请求的内容) 就直接使用。这个判断的过程就是 Cache 地址映射, 这个速度应该尽可能快, 常见的映射方法有直接映射、全相联映射和组相联映射 3 种, 其原理如图 3-2 所示。

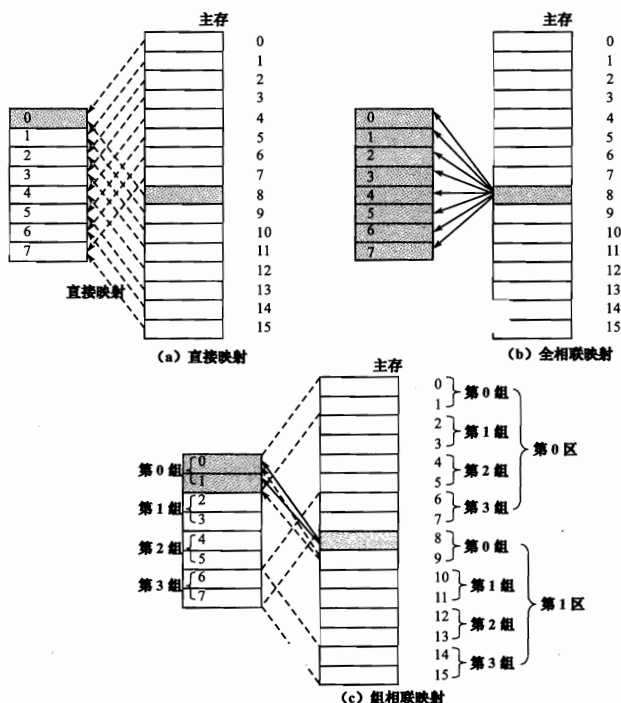


图 3-2 Cache 映射规则图

- 直接映射：是一种多对一的映射关系，但一个主存块只能够复制到 Cache 的一个特定位置上去。Cache 的行号 i 和主存的块号 j 有函数关系： $i=j\%m$ （其中 m 为 Cache 总行数）。例如，某 Cache 容量为 16KB（可用 14 位表示），每行的大小为 16B（可用 4 位表示），则说明其可分为 1024 行（可用 10 位表示）。则主存地址的最低 4 位为 Cache 的行内地址，中间 10 位为 Cache 行号。如果内存地址为 1234E8F8H 的话，那么最后 4 位就是 1000（对应十六进制数的最后一位），而中间 10 位，则应从 E8F（111010001111）中获取，得到 1010001111。
- 相联映射：将主存中一个块的地址与块的内容一起存于 Cache 的行中。速度更快，但控制复杂。
- 组相联映射：是前两种方式的折中方案。它将 Cache 中的块再分成组。然后通过直接映射方式决定组号，再通过相联映射的方式决定 Cache 中的块号。

要注意的是，在 Cache 映射中，主存和 Cache 存储器将均分成容量相同的块。

例如：容量为 64 块的 Cache 采用组相联方式映像，字块大小为 128 个字，每 4 块为一组。若主存容量为 4096 块，且以字编址，那么主存地址应该为多少位？主存

区号为多少位？这样的题目，首先根据主存与 Cache 块的容量需一致，因此内存也是 128 个字，因此共有 128×4096 个字，即 $2^{19}(2^7 \times 2^{12})$ 个字，因此主存地址需要 19 位；而内存所需要分为 $4096/64$ 块，即 2^6 ，因此主存区号需要 6 位。

(3) Cache 淘汰算法

当 Cache 数据已满，并且出现未命中情况时，就应淘汰一些老的数据，更新一些新的数据。选择淘汰什么数据的方法就是淘汰算法，常见的方法有 3 种：随机淘汰、先进先出 (FIFO) 淘汰 (淘汰最早调入 Cache 的数据)、最近最少使用 (LRU) 淘汰法。其中平均命中率最高的是 LRU 算法。

(4) Cache 存储器的写操作

在使用 Cache 时，需要保证其数据与主存一致，因此在写 Cache 时就需要考虑与主存间的同步问题，通常使用以下 3 种方法：写直达 (写 Cache 时，同时写主存)、写回 (写 Cache 时不马上写主存，而是等其淘汰时回写)、标记法。

3. 主存 (内存)

(1) 主存储器的种类

- RAM: 随机存储器，可读/写，断电后数据无法保存，只能暂存数据。
 - SRAM: 静态随机存储器，在不断电时信息能够一直保持。
 - DRAM: 动态随机存储器，需要定时刷新以维持信息不丢失。
- ROM: 只读存储器，出厂前用掩膜技术写入，常用于存放 BIOS 和微程序控制。
- PROM: 可编程 ROM，只能够一次写入，需用特殊电子设备进行写入。
- EPROM: 可擦除的 PROM，用紫外线照射 15~20 分钟可擦去所有信息，可写入多次。
- E²PROM: 电可擦除 ERPOM，可以写入，但速度慢。
- 闪存存储器: 现在 U 盘使用的种类，可以快速写入。

记忆时，抓住几个关键英文字母。A，即 Access，说明读写都行；O，即 Only，说明只读；P，即 Programmable，说明可通过特殊电子设备写入；E，即 Erasable，说明可擦写；E²说明是两个 E，第二个 E 是电子。

(2) 主存储器的组成

实际的存储器总是由一片或多片存储器配以控制电路构成的，如图 3-3 所示。其容量为 $W \times B$ ， W 是存储单元 (word，即字) 的数量， B 表示每个 word 由多少 bit (位) 组成。如果某一芯片规格为 $w \times b$ ，则组成 $W \times B$ 的存储器需要用 $(W/w) \times (B/b)$ 个芯片。

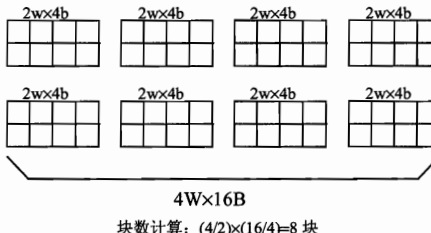


图 3-3 主存储器组成示意图

(3) 主存储器的地址编码

主存储器（内存）采用的是随机存取方式，需对每个数据块进行编码，而在主存储器中数据块是以 word 来标识的，即每个字一个地址，通常采用的是十六进制表示。例如，按字节编址，地址从 A4000H 到 CBFFFH，则表示有 $(CBFFF - A4000) + 1$ 个字节，28000H 个，也就是 163840 个字节，等于 160KB。

要注意的是，编址的基础可以是字节，也可以是字（字是由 1 个或多个字节组成的），要算地址位数，首先应计算要编址的字或字节数，然后求 2 的对数即可得到。

4. 磁盘

磁盘是最常见的一种外部存储器，它是由 1 至多个圆形磁盘组成的，其结构如图 3-4 所示。

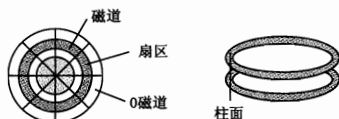


图 3-4 磁盘结构示意图

(1) 概念

- 磁道：磁道是一组记录密度不同的同心圆。
在一个磁盘中，从外到内，磁盘记录密度不断增加。同时，值得注意的是，0 磁道是磁盘最外圈的磁道。
- 扇区：磁盘上的每个磁道被等分为若干个弧段，这些弧段便是磁盘的扇区。
- 柱面：一个磁盘中，多个记录面相同的磁道组成柱面。如磁盘有 9 个记录面，则这 9 个记录面的 0 磁道可组成一个柱面。

(2) 公式

- 磁道数 = (外半径 - 内半径) × 道密度 × 记录面数。

注意：硬盘的第一面与最后一面是保护用的，要减掉。如 3 个双面的盘片的记录面数是 $3 \times 2 - 2 = 4$ 。

- 非格式化容量 = 位密度 × π × 最内圈直径 × 总磁道数。

注意：位密度是每道不同的，但每道的容量是相同的。0 道是最外面的磁道，其位密度最小。

- 格式化容量 = 每道扇区数 × 扇区容量 × 总磁道数。
- 平均数据传输速率 = 每道扇区数 × 扇区容量 × 盘片转速。
- 存取时间 = 寻道时间 + 等待时间，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。

(3) 数据存取过程

根据硬盘存放数据的规则，在向磁盘记录一个文件时，应将文件尽可能记录在同一柱面上，当一个柱面记录不下时，再记录到相邻柱面上。因此，当一个文件超出一个磁道容量时，剩下的部分应存于其他盘面的同一编号的磁道上，即同一柱面的其他磁道上。

为存取磁盘上的一个物理记录，必须给出 3 个参数：柱面号、磁头号（盘面号）、扇区号。磁盘机根据柱面号控制移动臂作径向运动，带动读/写头到达所需的柱面；从磁头号可确定哪一个磁头来读/写数据，然后便等待访问的信息块旋转到读/写头下进行存取。磁盘机实现这些功能的操作是：查找（将读/写头定位到指定柱面并选择指

定磁头)、搜索(指定磁头寻找访问的记录块)、读、写和控制等。

平均存取时间(Average Access Time)是反映磁盘数据操作速度的指标,单位是毫秒(ms)。它包括3个时间段,分别是平均寻道时间(Seek Time)、平均定位时间(Setting Time)和转动延迟(Rotational Latency),其中后两个又统称为等待时间。

寻道时间也称为查找时间,是磁头移动到目标磁道所需的时间。对于固定磁头磁盘而言,无须移动磁头,只需选择目标磁道对应的磁头即可。等待时间为等待读写的扇区旋转到磁头下方所用的时间。一般选用磁道旋转一周所用时间的一半作为平均等待时间。寻道时间由磁盘机的性能决定,目前主流硬盘典型的平均寻道时间(Average Seek Time, AST)一般在4ms左右,而转速则有5400r/m、7200r/m、15000r/m等。在考试当中,这些参数通常是由试题指定。

(4) 磁盘调度算法

- 先来先服务(FCFS):该算法根据进程请求访问磁盘的先后次序进行调度。其优点是公平、简单,且每个进程的请求都能依次得到处理,不会出现某一进程的请求长期得不到满足的情况。但此算法由于未对寻道进行优化,致使平均寻道时间可能较长。
- 最短寻道时间优先(SSTF):该算法选择这样的进程,其要求访问的磁道与当前磁头所在的磁道距离最近,以使每次的寻道时间最短,但这种调度算法却不能保证平均寻道时间最短。
- 扫描(SCAN)算法:SCAN算法不仅考虑到欲访问的磁道与当前磁道的距离,更优先考虑的是磁头的当前移动方向。例如,当磁头正在自里向外移动时,SCAN算法所选择的下一个访问对象应是其欲访问的磁道既在当前磁道之外,又是距离最近的。这样自里向外地访问,直到再无更外的磁道需要访问时才将磁臂换向,自外向里移动。由于这种算法中磁头移动的规律颇似电梯的运行,故又称为电梯调度算法。
- 循环扫描(CSCAN)算法:该算法规定磁头单向移动。例如,只自里向外移动,当磁头移到最外的被访问磁道时,磁头立即返回到最里的欲访磁道,即将最小磁道号紧接着最大磁道号构成循环,进行扫描。

3.2 典型试题分析

试题 1

嵌入式系统中采用中断方式实现输入/输出的主要原因是__(1)___。在中断时,CPU断点信息一般保存到__(2)___中。

- | | |
|--------------|-----------------|
| (1) A. 速度最快 | B. CPU 不参与操作 |
| C. 实现起来比较容易 | D. 能对突发事件做出快速响应 |
| (2) A. 通用寄存器 | B. 堆 |
| C. 栈 | D. I/O 接口 |

试题分析

本题主要考查嵌入式系统输入/输出方式的基础知识。

在一般的操作系统中,输入/输出方式主要有以下几种:

①程序控制方式：CPU 直接利用 I/O 指令编程，实现数据的 I/O。CPU 发出 I/O 命令，命令中包含了外设的地址信息和所要执行的操作，相应的 I/O 系统执行该命令并设置状态寄存器；CPU 不停地（定期地）查询 I/O 系统以确定该操作是否完成。由程序主动查询外设，完成主机与外设间的数据传送，方法简单，硬件开销小。

②程序中断方式：CPU 利用中断方式完成数据的 I/O，当 I/O 系统与外设交换数据时，CPU 无须等待也不必去查询 I/O 的状态，当 I/O 系统完成了数据传输后则以中断信号通知 CPU。CPU 然后保存正在执行程序的状态，转入 I/O 中断服务程序完成与 I/O 系统的数据交换。然后返回原主程序继续执行。与程序控制方式相比，中断方式因为 CPU 无须等待而提高了效率。在系统中具有多个中断源的情况下，常用的处理方法有：多中断信号线法、中断软件查询法、维菊链法、总线仲裁法和中断向量表法。

③DMA 方式：使用 DMA 控制器（DMAC）来控制和管理数据传输。DMAC 和 CPU 共享系统总线，并且具有独立访问存储器的能力。在进行 DMA 时，CPU 放弃对系统总线的控制而由 DMAC 控制总线；由 DMAC 提供存储器地址及必需的读写控制信号，实现外设与存储器之间进行数据交换。DMAC 获取总线方式主要有 3 种，分别是暂停方式、周期窃取方式和共享方式。

④通道：通道是一种通过执行通道程序管理 I/O 操作的控制器，它使主机与 I/O 操作之间达到更高的并行程度。在具有通道处理机的系统中，当用户进程请求启动外设时，由操作系统根据 I/O 要求构造通道程序和通道状态字，将通道程序保存在主存中，并将通道程序的首地址放到通道地址字中，然后执行“启动 I/O”指令。按照所采取的传送方式，可将通道分为字节多路通道、选择通道和数组多路通道 3 种。

⑤输入输出处理机（IOP）：也称为外围处理机（PPU），它是一个专用处理机，也可以是一个通用的处理机，具有丰富的指令系统和完善的中断系统。专用于大型、高效的计算机系统处理外围设备的 I/O，并利用共享存储器或其他共享手段与主机交换信息。从而使大型、高效的计算机系统更加高效地工作。与通道相比，IOP 具有比较丰富的指令系统，结构接近于一般的处理机，有自己的局部存储器。

嵌入式系统中采用中断方式实现输入/输出的主要原因是能对突发事件做出快速响应。在中断时，CPU 断点信息一般保存到栈中。

试题答案

(1) D

(2) C

试题 2

在嵌入式系统设计时，下面几种存储结构中对程序员是透明的是___(3)___。

(3) A. 高速缓存

B. 磁盘存储器

C. 内存

D. Flash 存储器

试题分析

本题主要考查嵌入式系统程序设计中存储结构的操作。

4 个选项中，高速缓存即 Cache，它处于内存与 CPU 之间，是为了提高访问内存时的速度而设置的，这个设备对于程序员的程序编写是完全透明的。

磁盘存储器与 Flash 存储器都属于外设，在存储文件时，需要考虑到该设备的情况，因为需要将文件内容存于相应的设备之上。

- (7) A. 指令寄存器 (IR) B. 程序计数器 (PC)
C. 算术逻辑单元 (ALU) D. 程序状态字寄存器 (PSW)

试题分析

控制器是分析和执行指令的部件,也是统一指挥和控制计算机各个部件按时序协调操作的部件。控制器的组成包含如下部分:① 程序计数器(PC);② 指令寄存器(IR);③ 指令译码器;④ 时序部件;⑤ 程序状态字寄存器(PSW);⑥ 中断机构。故C答案的算术逻辑单元(ALU)不属于控制器,是运算器。

试题答案

- (7) C

试题 7

在CPU中,____(8)____可用于传送和暂存用户数据,为ALU执行算术逻辑运算提供工作区。

- (8) A. 程序计数器 B. 累加寄存器 C. 程序状态寄存器 D. 地址寄存器

试题分析

本题考查CPU中常用寄存器的功能。这些寄存器的详细功能描述请参看“3.1.6 存储系统”中的寄存器部分。

所以本题应选B。

试题答案

- (8) B

试题 8

____(9)____是指按内容访问的存储器。

- (9) A. 虚拟存储器 B. 相联存储器
C. 高速缓存 (Cache) D. 随机访问存储器

试题分析

本题考查计算机系统存储器方面的基础知识。

计算机系统的存储器按所处的位置可分为内存和外存。按构成存储器的材料可分为磁存储器、半导体存储器和光存储器。按存储器的工作方式可分为读写存储器和只读存储器。按访问方式可分为按地址访问的存储器和按内容访问的存储器。按寻址方式可分为随机存储器、顺序存储器和直接存储器。

相联存储器是一种按内容访问的存储器。

试题答案

- (9) B

3.3 实战练习题

● 计算机执行程序时, 在一个指令周期的过程中, 为了能够从内存中读指令操作码, 首先是将__(1)___的内容送到地址总线上。

- (1) A. 程序计数器 (PC) B. 指令寄存器 (IR)
C. 状态寄存器 (SR) D. 通用寄存器 (GR)

● 内存按字节编址, 利用 $8K \times 4\text{bit}$ 的存储器芯片构成 $84000H \sim 8FFFFH$ 的内存, 共需__(2)___片。

- (2) A. 6 B. 8 C. 12 D. 24

● 某磁盘磁头从一个磁道移至另一个磁道需要 10ms 。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 100ms 和 2ms , 则读取一个 100 块的文件需要__(3)___ms 的时间。

- (3) A. 10200 B. 11000 C. 11200 D. 20200

● 计算机系统中, 在__(4)___的情况下一般应采用异步传输方式。

- (4) 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

● 在 CPU 与主存之间设置高速缓冲存储器 (Cache), 其目的是为了__(5)___。

- (5) A. 扩大主存的存储容量 B. 提高 CPU 对主存的访问效率
C. 既扩大主存容量又提高存取速度 D. 提高外存储器的速度

● 系统响应时间和作业吞吐量是衡量计算机系统性能的重要指标。对于一个持续处理业务的系统而言, 其__(6)___。

- (6) A. 响应时间越短, 作业吞吐量越小 B. 响应时间越短, 作业吞吐量越大
C. 响应时间越长, 作业吞吐量越大 D. 响应时间不会影响作业吞吐量

● 挂接在总线上的多个部件, __(7)___。

- (7) 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

3.4 练习题解析

试题 1 分析

本题考查指令的操作码。指令系统中的每一条指令都有一个操作码, 它表示该指令应进行什么性质的操作。不同的指令用操作码这个字段的不同编码来表示, 每一种

编码代表一种指令。组成操作码字段的位数一般取决于计算机指令系统的规模。

程序计数器(PC)用于记录需要执行的下一条指令操作码的地址,所以在读指令操作码时,应将程序计数器的内容送到地址总线上。

试题1 答案

(1) A

试题2 分析

本题的题型在软考中较为常见,其难度在于计算时需要注意技巧,如果不注意技巧,将浪费大量时间于无谓的计算过程。 $8\text{FFFFH}-84000\text{H}+1=(8\text{FFFFH}+1)-84000\text{H}=90000\text{H}-84000\text{H}=\text{C000H}$,转换为十进制为48K。由于内存是按字节编址,所以存储容量为: $48\text{K}\times 8\text{bit}$, $48\text{K}\times 8\text{bit}/(8\text{K}\times 4\text{bit})=12$ 。

试题2 答案

(2) C

试题3 分析

本题考查磁盘读取数据的基本原理。

通常要在磁盘上读取一个数据,首先需要寻道,即将磁头定位于目标磁道,然后通过磁盘的旋转,定位于要读取的信息之上,最后读取并传输数据。所以,读取每个数据需要耗费的平均时间为 $10\times 10+100+2=202\text{ms}$, 100 块的文件需要 20200ms。

试题3 答案

(3) D

试题4 分析

本题考查计算机系统中数据传输的方式。CPU 访问内存通常是同步方式, CPU 与 I/O 接口交换信息通常是同步方式, CPU 与 PCI 总线交换信息通常是同步方式, I/O 接口与打印机交换信息则通常采用基于缓存池的异步方式,因此答案为 D。

试题4 答案

(4) D

试题5 分析

由于 CPU 的速度比主存的读取速度快得多,为解决这种不匹配,在它们之间设置高速缓冲存储器(Cache),将主存中的内容事先调入 Cache 中, CPU 直接访问 Cache 的时间短得多,这样大大提高了 CPU 对主存的访问效率,也提高了整个计算机系统的效率。

试题5 答案

(5) B

试题6 分析

系统响应时间是指用户发出完整请求到系统完成任务给出响应的时间间隔。作业吞吐量是指单位时间内系统完成的任务量。若一个给定系统持续地收到用户提交的任

务请求，则系统的响应时间将对作业吞吐量造成一定影响。若每个任务的响应时间越短，则系统的空闲资源越多，整个系统在单位时间内完成的任务量将越大；反之，若响应时间越长，则系统的空闲资源越少，整个系统在单位时间内完成的任务量将越小。

试题 6 答案

(6) B

试题 7 分析

本题考查考生对总线概念的理解。总线是一个大家都能使用的数据传输通道，大家都可以使用这个通道，但发送数据时，是采用的分时机制，而接收数据时可以同时接收，也就是说，同一个数据，可以并行地被多个客户收取。如果该数据不是传给自己的，数据包将被丢弃。

试题 7 答案

(7) B

4

第4章

数据通信与计算机网络

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

- 数据通信的基本知识。
- 开放系统互连参考模型。
- 常用的协议标准。
- 网络互联与常用网络设备。
- 计算机网络的分类与应用。

4.1 考点突破

从历年的考试情况来看，本章的考点主要集中在网络设计相关的知识，偏宏观而非技术细节。

4.1.1 历年考试情况分析

在历年的考试试题中，有关数据通信与计算机网络知识的试题如表 4-1 所示。

表 4-1 数据通信与计算机网络知识试题分布表

题号	2009.11	2010.11	2011.11
13	网络层次化设计	网络层次化设计	网络层次化设计
14	物理网络设计	逻辑网络设计	综合布线
15	网络存储	网络系统生命周期	逻辑网络设计

按照知识点进行总结和归类的试题分布情况如表 4-2 所示。

表 4-2 数据通信与计算机网络知识归类表

知识点	2009.11	2010.11	2011.11
开放系统互联参考模型	0	0	0
TCP/IP 协议族	0	0	0
网络规划与设计	2	2	2
网络存储技术	1	0	0
网络应用	0	0	0
其他	0	1	1
合计	3	3	3

从表 4-2 中可以看出,数据通信与计算机网络知识方面的内容在历年的考试中分值非常稳定,一直稳定维持在 3 分。所占分数比例的趋势如图 4-1 所示。

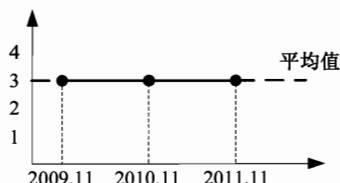


图 4-1 数据通信与计算机网络知识历年试题比例趋势图

4.1.2 开放系统互连参考模型

在开放系统互连参考模型（Open System Interconnection/Reference Model, OSI/RM）方面,考查的主要知识点是 OSI/RM 各层的功能和所涉及的协议。

OSI/RM 模型共分 7 层,从下往上分别是:物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。当接收数据时,数据是自下而上传输的;当发送数据时,数据是自上而上传输的。OSI/RM 7 层的主要功能如表 4-3 所示。

表 4-3 OSI/RM 模型的 7 层及主要功能

层次	名 称	主要功能	主要设备及协议
7	应用层	实现具体的应用功能	POP3、FTP、HTTP、Telnet、SMTP DHCP、TFTP、SNMP、DNS
6	表示层	数据的格式与表达、加密、压缩	
5	会话层	建立、管理和终止会话	
4	传输层	端到端连接	TCP、UDP
3	网络层	分组传输和路由选择	三层交换机, 路由器 ARP、RARP、IP、ICMP、IGMP
2	数据链路层	传送以帧为单位的信息	网桥, 交换机, 网卡 PPTP、L2TP、SLIP、PPP
1	物理层	二进制传输	中继器, 集线器

在网络数据通信的过程中,每一层要完成特定的任务。当传输数据的时候,每一

服务器集中管理，并负责处理客户端的 DHCP 要求；而客户端则会使用从服务器分配下来的 IP 环境数据。DHCP 透过“租约”的概念，有效且动态地分配客户端的 TCP/IP 设定。DHCP 分配的 IP 地址可以分为 3 种方式，分别是固定分配、动态分配和自动分配。

- NFS (Net File System, 网络文件系统) 是 FreeBSD 支持的文件系统中的一种，允许一个系统在网络上与他人共享目录和文件。通过使用 NFS，用户和程序可以像访问本地文件一样访问远端系统上的文件。
- Telnet (远程登录协议) 是登录和仿真程序，它的基本功能是允许用户登录进入远程主机系统。以前，Telnet 是一个将所有用户输入送到远方主机进行处理的简单的终端程序。它的一些较新的版本在本地执行更多的处理，于是可以提供更好的响应，并且减少了通过链路发送到远程主机的信息数量。
- DNS (Domain Name System, 域名系统) 用于命名组织到域层次结构中的计算机和网络服务。在 Internet 上域名与 IP 地址之间是一一对应的，域名虽然便于人们记忆，但机器之间只能互相认识 IP 地址，它们之间的转换工作称为域名解析，域名解析需要由专门的域名解析服务器来完成，DNS 就是进行域名解析的服务器。DNS 通过对用户友好的名称查找计算机和服务。当用户在应用程序中输入 DNS 名称时，DNS 服务可以将此名称解析为与之相关的其他信息，如 IP 地址。
- SNMP (Simple Network Management Protocol, 简单网络管理协议) 是为了解决 Internet 上的路由器管理问题而提出的，它可以在 IP、IPX、AppleTalk、OSI 及其他用到的传输协议上被使用。SNMP 事实上指一系列网络管理规范的集合，包括协议本身、数据结构的定义和一些相关概念。目前，SNMP 已成为网络管理领域中事实上的工业标准，并被广泛支持和应用，大多数网络管理系统和平台都是基于 SNMP 的。

2. 传输层

TCP/IP 的传输层大致对应于 OSI 模型的会话层和传输层，主要包括 TCP 和 UDP，这些协议负责提供流控制、错误校验和排序服务。所有的服务请求都使用这些协议。

- TCP (Transport Control Protocol, 传输控制协议) 是整个 TCP/IP 协议族中最重要的协议之一，它在 IP 提供的不可靠数据服务的基础上，采用了重发技术，为应用程序提供了一个可靠的、面向连接的、全双工的数据传输服务。TCP 一般用于传输数据量比较少，且对可靠性要求高的场合。
- UDP (User Datagram Protocol, 用户数据报协议) 是一种不可靠的、无连接的协议，可以保证应用程序进程间的通信，与同样处在传输层的面向连接的 TCP 相比较，UDP 是一种无连接的协议，它的错误检测功能要弱得多。可以这样说，TCP 有助于提供可靠性，而 UDP 则有助于提高传输的高速率。UDP 一般用于传输数据量大、对可靠性要求不是很高、但要求速度快的场合。

3. 网际层

TCP/IP 的网际层对应于 OSI 模型的网络层，包括 IP、ICMP、IGMP，以及 ARP 和 RARP。这些协议处理信息的路由及主机地址解析。

- IP (Internet Protocol, 网际协议) 所提供的服务通常被认为是无连接的和不可靠的, 因此把差错检测和流量控制之类的服务授权给了其他的各层协议, 这正是 TCP/IP 能够高效率工作的一个重要保证。网际层的功能主要由 IP 来提供, 除了提供端到端的分组分发功能外, IP 还提供了很多扩充功能。例如, 为了克服数据链路层对帧大小的限制, 网络层提供了数据分块和重组功能, 这使得很大的 IP 数据包能以较小的分组在网上传输。

网际层的另一个重要服务是在互相独立的局域网上建立互联网络, 即网际网。网间的报文来往根据它的目的 IP 地址通过路由器传到另一网络。

- ARP (Address Resolution Protocol, 地址解析协议) 用于动态地完成 IP 地址向物理地址的转换。物理地址通常是指主机的网卡地址 (MAC 地址), 每一网卡都有唯一的地址。
- RARP (Reverse Address Resolution Protocol, 反向地址解析协议) 用于动态完成物理地址向 IP 地址的转换。
- ICMP (Internet Control Message Protocol, 网际控制报文协议) 是一个专门用于发送差错报文的协议, 由于 IP 是一种尽力传送的通信协议, 即传送的数据可能丢失、重复、延迟或乱序传递, 所以 IP 需要一种尽量避免差错并能在发生差错时报告的机制。
- IGMP (Internet Group Management Protocol, 网际组管理协议) 允许 Internet 主机参加多播, 也即 IP 主机用做向相邻多目路由器报告多目组成员的协议。多目路由器是支持组播的路由器, 向本地网络发送 IGMP 查询。主机通过发送 IGMP 报告来应答查询。组播路由器负责将组播包转发到网络中所有组播成员。

4. 网络接口层

TCP/IP 的网络接口层大致对应于 OSI 模型的数据链路层和物理层, TCP/IP 不包含具体的物理层和数据链路层, 只定义了网络接口层作为物理层的接口规范。网络接口层处在 TCP/IP 的最底层, 主要负责管理为物理网络准备数据所需的全部服务程序和功能。该层处理数据的格式化并将数据传输到网络电缆, 是 TCP/IP 的实现基础, 其中可包含 IEEE 802.3 的 CSMA/CD、IEEE 802.5 的 TokenRing 等。

4.1.4 网络规划与设计

我们在网络建设前都要做一个需求分析工作, 否则, 网络建立起来就带有盲目性, 轻则造成网络资源浪费或网络瓶颈, 重则使网络瘫痪, 损失无法估量的数据资源。网络建设前的需求分析, 就是要规划网络建设所要做的工作。根据用户提出的要求, 进行网络的设计。可以这么说, 网络建设的好坏、快慢、可持续发展性等, 都将取决于网络实施前的规划工作如下:

- **网络的功能要求。**任何网络都不可能是一个可以进行各种各样工作的“万能网”, 因此, 必须针对每一个具体的网络, 依据使用要求、实现成本、未来发展、总预算投资等因素仔细地反复推敲, 尤其是分析出网络系统要完成的所有功能。
- **网络的性能要求。**根据对网络系统的相应时间、事物, 处理的实时性进行研

究, 确定系统需要的存储量及备用的存储量。根据网络的工作站权限、容错程度、网络安全性方面的要求等, 确定采取何种措施及方案。

- **网络运行环境的要求。**根据整个局域网运行时所需要的环境要求, 确定使用哪种网络操作系统、应用系统, 以及相应的应用软件和共享资源。
- **网络的扩充性和可维护性要求。**如何增加工作站、怎样与其他网络联网、对软件/硬件的升级换代有何要求与限制等, 都要在网络设计时加以考虑, 以保证网络的扩充性和可维护性。通常新建网络时都会给这个局域网提出一些有关使用寿命、维护代价等的要求。

在网络设计方面, 主要采用层次式方法。层次式网络设计在互联网组件的通信中引入了3个关键层的概念, 分别是核心层、汇聚层和接入层。

- 通常将网络中直接面向用户连接或访问网络的部分称为接入层, 将位于接入层和核心层之间的部分称为分布层或汇聚层。接入层的目的是允许终端用户连接到网络, 因此接入层交换机具有低成本和高端口密度特性。
- 汇聚层交换机是多台接入层交换机的汇聚点, 它必须能够处理来自接入层设备的所有通信量, 并提供到核心层的上行链路, 因此汇聚层交换机与接入层交换机比较, 需要更高的性能, 更少的接口和更高的交换速率。汇聚层是核心层和终端用户接入层的分界面, 汇聚层完成了网络访问策略控制、数据包处理、过滤、寻址, 以及其他数据处理的任务。
- 将网络主干部分称为核心层, 核心层的主要目的在于通过高速转发通信, 提供优化、可靠的骨干传输结构, 因此核心层交换机应拥有更高的可靠性、性能和吞吐量。核心层为网络提供了骨干组件或高速交换组件, 在纯粹的分层设计中, 核心层只完成数据交换的特殊任务。

4.1.5 网络存储技术

目前, 主流的网络存储技术主要有3种, 分别是直接附加存储(Direct Attached Storage, DAS)、网络附加存储(Network Attached Storage, NAS)和存储区域网络(Storage Area Network, SAN)。

1. 直接附加存储

DAS是将存储设备通过SCSI(Small Computer System Interface, 小型计算机系统接口)电缆直接连到服务器, 其本身是硬件的堆叠, 存储操作依赖于服务器, 不带有存储操作系统。因此, 有些文献也把DAS称为SAS(Server Attached Storage, 服务器附加存储)。

DAS的适用环境如下:

- 服务器在地理分布上很分散, 通过SAN或NAS在它们之间进行互连非常困难时。
- 存储系统必须被直接连接到应用服务器(例如, Microsoft Cluster Server或某些数据库使用的“原始分区”)上时。
- 包括许多数据库应用和应用服务器在内的应用, 它们需要直接连接到存储器上时。

由于 DAS 直接将存储设备连接到服务器上, 这导致它在传递距离、连接数量、传输速率等方面都受到限制。因此, 当存储容量增加时, DAS 方式很难扩展, 这对存储容量的升级是一个巨大的瓶颈; 另一方面, 由于数据的读取都要通过服务器来处理, 必然导致服务器的处理压力增加, 数据处理和传输能力将大大降低; 此外, 当服务器出现死机等异常时, 也会波及到存储数据, 使其无法使用。目前 DAS 基本被 NAS 所代替。

2. 网络附加存储

采用 NAS 技术的存储设备不再通过 I/O 总线附属于某个特定的服务器, 而是通过网络接口与网络直接相连, 由用户通过网络访问。NAS 存储系统的结构如图 4-3 所示。

NAS 存储设备类似于一个专用的文件服务器, 它去掉了通用服务器的大多数计算功能, 而仅提供文件系统功能, 从而降低了设备的成本。并且为方便存储设备到网络之间以最有效的方式发送数据, 专门优化了系统硬软件体系结构。NAS 以数据为中心, 将存储设备与服务器分离, 其存储设备在功能上完全独立于网络中的主服务器, 客户机与存储设备之间的数据访问不再需要文件服务器的干预, 同时它允许客户机与存储设备之间进行直接的数据访问, 所以不仅响应速度快, 而且数据传输速率也很高。

NAS 技术支持多种 TCP/IP 网络协议, 主要是 NFS (Net File System, 网络文件系统) 和 CIFS (Common Internet File System, 通用 Internet 文件系统) 来进行文件访问, 所以 NAS 的性能特点是进行小文件级的共享存取。在具体使用时, NAS 设备通常配置为文件服务器, 通过使用基于 Web 的管理界面来实现系统资源的配置、用户配置管理和用户访问登录等。

NAS 存储支持即插即用, 可以在网络的任一位置建立存储。基于 Web 管理, 从而使设备的安装、使用和管理更加容易。NAS 可以很经济地解决存储容量不足的问题, 但难以获得满意的性能。

3. 存储区域网络

SAN 是通过专用交换机将磁盘阵列与服务器连接起来的高速专用子网。它没有采用文件共享存取方式, 而是采用块 (Block) 级别存储。SAN 是通过专用高速网将一个或多个网络存储设备和服务器连接起来的专用存储系统, 其最大特点是将存储设备从传统的以太网中分离了出来, 成为独立的存储区域网络 (SAN) 的系统结构如图 4-4 所示。

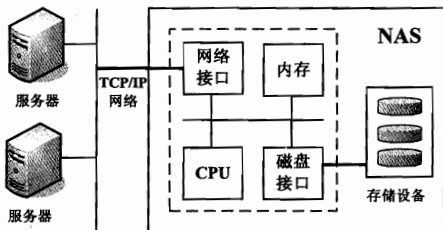


图 4-3 NAS 存储系统的结构

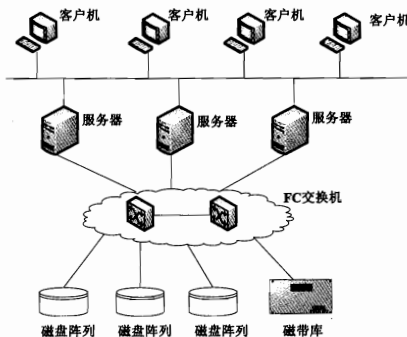


图 4-4 SAN 存储系统的结构

根据数据传输过程采用的协议，其技术划分为 FC SAN 和 IP SAN。另外，还有一种新兴的 IB SAN 技术。

(1) FC SAN

FC (Fiber Channel, 光纤通道) 和 SCSI 接口一样，最初也不是为硬盘设计开发的接口技术，而是专门为网络系统设计的，随着存储系统对速度的需求，才逐渐应用到硬盘系统中。光纤通道的主要特性有：热插拔性、高速带宽、远程连接、连接设备数量大等。它是当今最昂贵和复杂的存储架构，需要在硬件、软件和人员培训方面进行大量投资。

FC SAN 由 3 个基本的组件构成，分别是接口 (SCSI、FC)、连接设备 (交换机、路由器) 和协议 (IP、SCSI)。这 3 个组件再加上附加的存储设备和服务器就构成一个 SAN 系统。它是专用、高速、高可靠的网络，允许独立、动态地增加存储设备，使得管理和集中控制更加简化。

FC SAN 有两个较大的缺陷，分别是成本和复杂性，其原因就是因为使用了 FC。在光纤通道上部署 SAN，需要每个服务器上都要有 FC 适配器、专用的 FC 交换机和独立的布线基础架构。这些设施使成本大幅增加，更不用说精通 FC 协议的人员培训成本。

(2) IP SAN

IP SAN 是基于 IP 网络实现数据块级别存储方式的存储网络。由于设备成本低，配置技术简单，可共享和使用大容量的存储空间，因而逐渐获得广泛的应用。

在具体应用上，IP 存储主要是指 iSCSI (Internet SCSI)。作为一种新兴的存储技术，iSCSI 基于 IP 网络实现 SAN 架构，既具备了 IP 网络配置和管理简单的优势，又提供了 SAN 架构所拥有的强大功能和扩展性。iSCSI 是连接到一个 TCP/IP 网络的直接寻址的存储库，通过使用 TCP/IP 对 SCSI 指令进行封装，可以使指令能够通过 IP 网络进行传输，而过程完全不依赖于地点。

iSCSI 优势主要表现在：首先，建立在 SCSI、TCP/IP 这些稳定和熟悉的标准上，因此安装成本和维护费用都很低；其次，iSCSI 支持一般的以太网交换机而不是特殊的光纤通道交换机，从而减少了异构网络和电缆；最后，iSCSI 通过 IP 传输存储命令，因此可以在整个 Internet 上传输，没有距离限制。

iSCSI 的缺点在于，存储和网络是同一个物理接口，同时，协议本身的开销较大，协议本身需要频繁地将 SCSI 命令封装到 IP 包中，以及从 IP 包中将 SCSI 命令解析出来，这两个因素都造成了带宽的占用和主处理器的负担。但是，随着专门处理 iSCSI 指令的芯片的开发（解决主处理器的负担问题），以及 10G 以太网的普及（解决带宽问题），iSCSI 将有更好的发展。

(3) IB SAN

IB (Infini Band, 无限带宽) 是一种交换结构 I/O 技术，其设计思路是通过一套中心机构 (IB 交换机) 在远程存储器、网络及服务器等设备之间建立一个单一的连接链路，并由 IB 交换机来指挥流量。这种结构设计得非常紧密，大大提高了系统的性能、可靠性和有效性，能缓解各硬件设备之间的数据流量拥塞。而这是许多共享总线式技术没有解决好的问题，因为在共享总线环境中，设备之间的连接都必须通过指定的端口建立单独的链路。

IB 主要支持两种环境：模块对模块的计算机系统（支持 I/O 模块附加插槽）；在

数据中心环境中的机箱对机箱的互联系统、外部存储系统和外部局域网/广域网访问设备。IB 支持的带宽比现在主流的 I/O 载体（如 SCSI、FC 等）还要高，另外，由于使用 IPv6 的报头，IB 还支持与传统 Internet/Intranet 设施的有效连接。用 IB 技术替代总线结构所带来的最重要的变化就是建立了一个灵活、高效的数据中心，省去了服务器复杂的 I/O 部分。

IB SAN 采用层次结构，将系统的构成与接入设备的功能定义分开，不同的主机可通过 HCA（Host Channel Adapter，主机通道适配器）、RAID 等网络存储设备利用 TCA（Target Channel Adapter，目标通道适配器）接入 IB SAN。

IB SAN 主要具有如下特性：可伸缩的 Switched Fabric 互连结构；由硬件实现的传输层互连高效、可靠；支持多个虚信道；硬件实现自动的路径变换；高带宽，总带宽随 IB Switch 规模成倍增长；支持 SCSI 远程 DMA（Direct Memory Access，直接内存存取）协议；具有较高的容错性和抗毁性，支持热拔插。

4. 网络存储技术的选择

网络存储技术的目的都是为了扩大存储能力，提高存储性能。这些存储技术都能提供集中化的数据存储并有效存取文件；都支持多种操作系统，并允许用户通过多个操作系统同时使用数据；都可以从应用服务器上分离存储，并提供数据的高可用性；同时，都能通过集中存储管理来降低长期的运营成本。

因此，从存储的本质上来看，它们的功能都是相同的。事实上，它们之间的区别正在变得模糊，所有的技术都在用户的存储需求下接受挑战。在实际应用中，需要根据系统的业务特点和要求（例如，环境要求、性能要求、价格要求等）进行选择。

4.1.6 网络应用

本节将介绍常见网络应用的基本情况、相关概念及原理。

1. WWW（万维网）

WWW 是一个支持交互式访问的分布式超媒体系统。超媒体系统（在超文本的基础上，结合语音、图形、图像、动画等信息）直接扩充了传统的超文本系统（非线性的、用“链接”整合的信息结构）。Web 文档用文本标记语言（HTML）来撰写。除了文本外，文档还包括指定文档版面与格式的标签。在页面中可以包含图形、音频、视频等各种多媒体信息。

在 WWW 中，依赖于标准化的统一资源定位器（Uniform Resource Locator，URL）地址来定位信息的内容。在进行页面访问时采用超文本传送协议（HyperText Transfer Protocol，HTTP），其服务端口就是 HTTP 服务端口（80 端口）。首先，浏览器软件与 HTTP 端口建立一个 TCP 连接，然后发送 GET 命令，Web 服务器根据命令取出文档，发送给浏览器；浏览器释放连接，显示文档。

2. E-mail（电子邮件）

电子邮件是现在数据量、使用量最大的一个因特网应用，它用来完成人机之间的消息通信。与它相关的有以下 3 个协议。

- SMTP：简单邮件传送协议，用于邮件的发送，工作在 25 号端口上。
- POP3：邮局协议 V3.0，用于接收邮件，工作在 110 号端口上。

- IMAP: 邮件访问协议, 是用于替代 POP3 协议的新协议, 工作在 143 号端口上。

3. DNS (域名解析系统)

网络用户希望用有意义的名字来标识主机, 而不是 IP 地址。为了解决这个需求, 应运而生了一个域名解析系统 (DNS) (工作在 53 号端口)。它是运行在 TCP 之上, 负责将域名转换成实际相对应的 IP 地址, 从而在不改变底层协议的寻址方法的基础上, 为用户提供了一个直接使用符号名来确定主机的平台。

DNS 是一个分层命名系统, 名字由若干个标号组成, 标号之前用圆点分隔。最右边的是主域名, 最左边的是主机名, 中间的是子域名。例如: xinu.cs.purdue.edu 中, xinu 是主机名、cs 和 purdue 是子域名 (分别代表计算机系、普渡大学)、edu 是主域名。可以看出一个域名可以由几个段组成, 那么它们是怎样被赋值的呢? 它是由国际域名管理机构 InterNIC 来制定最高域的选择方法, 然后由逐层组织自己确定剩下的部分, 每级组织都可以增加一个段。常用域名如表 4-4 所示。

表 4-4 常用域名

域 名	应 用 于
com	商业组织
edu	教育机构
gov	政府组织
mil	军事组织
net	主要网络支持中心
org	非营利性组织
arpa	临时 ARPANET 域
int	国际组织
国家代码 (如 cn)	国家

注意: 通常我们在写域时, 最后是不加一个 “.” 的, 其实这只是一个缩写, 最后一个 “.” 代表的是 “根”, 如果采用全域名写法, 还需要加上这个小点。这在配置 DNS 时就会需要。

除了以上讲述的名字语法规则和管理机构的设立, 域名系统中还包括一个高效、可靠、通用的分布式系统用于名字到地址的映射。将域名映射到 IP 地址的机制由若干个称为名字服务器 (Name Server) 的独立、协作的系统组成。要理解域名服务器的工作原理最简单的方法就是将它们放置在与命名等级对应的树形结构中, 如图 4-5 所示。

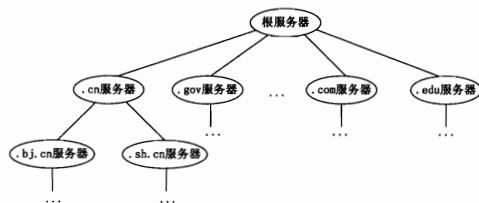


图 4-5 分布式 DNS 系统

4. FTP (文件传输协议)

- FTP 的传输模式包括: Bin (二进制) 和 ASCII (文本文件) 两种, 除了文本文件之外, 都应该使用二进制模式传输。

- FTP 应用的连接模式：在客户机和服务器之间需建立两条 TCP 连接，一条用于传送控制信息（21 端口），另一条用于传送文件内容（20 端口）。
- 匿名 FTP 的用户名为 anonymous。

5. DHCP（动态主机配置协议）

对于大中型网络而言，DHCP 可以减轻网管人员的负担，避免 IP 地址分配错误。它能让管理员快捷地验证 IP 地址和其他配置参数，可以为每个 DHCP 范围设置若干选项，并且如果主机物理上被移动到不同子网，DHCP 服务器也能够进行适当的配置。它还大大方便了便携机用户，移动到不同子网无须重新分配 IP 地址。

DHCP 服务最为基础的过程如图 4-6 所示。

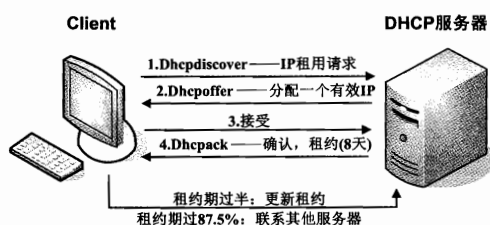


图 4-6 DHCP 服务流程

- 当 DHCP 客户机首次启动时，客户机向 DHCP 服务器发送一个 Dhcpdiscover 数据包，表达客户机的 IP 租用请求。
- DHCP 服务器收到 Dhcpdiscover 包后，就会从自己的地址范围中向那台主机提供（Dhcpoffer）一个还没有被分配的有效的 IP 地址。要注意的是：如果网络中有多台 DHCP 服务器，则客户机接受的将是最先收到的 Dhcpoffer。
- 最后，DHCP 服务器还会向客户机发送一个确认（Dhcpack），里面包含了最初发送的 IP 地址和其稳定期间的租约（默认值为 8 天）。
- 当租约过一半时，客户机会自动更新租约；当租约过了 87.5% 时，客户机仍然无法联系到当初的 DHCP 服务器，就会联系其他服务器，如果仍无法联系到，将会停用 IP。如图 4-7 所示是客户机 DHCP 状态迁移示意图。

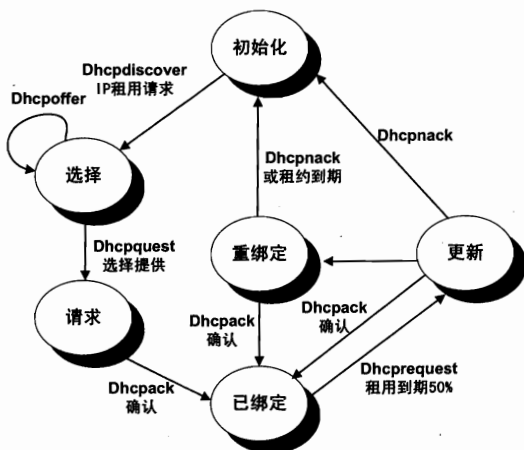


图 4-7 客户机 DHCP 状态迁移示意图

6. 其他应用

- **Gopher:** Internet 早期的一种全文检索服务, WWW 出现后, 被取代。
- **WebMail:** 指利用浏览器通过 Web 方式来收发电子邮件的服务或技术。
- **Usenet:** 新闻组是一个电子讨论组, 用户可以在这里与遍及全球的用户共享信息, 以及对某些问题的看法。
- **EZweb:** 日本现有的 3 家手机上网服务之一, 目前世界上最广泛和成功的 WAP 服务。
- **VOD:** 视频点播, 通过视频压缩、流技术、组播协议实现。
- **NetMeeting:** 网络会议, 通过视频压缩、流技术、组播协议实现。
- **VoIP:** Voice over Internet Protocol 是一种以 IP 电话为主, 并推出相应的增值业务的技术。VoIP 最大的优势是可以在 IP 网络上便宜地传送语音、传真、视频和数据等业务。它将通过 VoIP 网关与传统 PSTN 网络进行连接, 用来完成对声音信号进行压缩编码。
- **代理服务器:** 介于浏览器和 Web 服务器之间的一台服务器, 使用代理服务器之后, 浏览器就不是直接到 Web 服务器去获取网页信息了, 而是先访问代理服务器, 然后由代理服务器获取所需要的信息并传送给浏览器。它可以完成 IP 地址共享、信息缓存、信息转发等任务。

4.2 典型试题分析

试题 1

某 IP 网络连接如图 4-8 所示, 在这种配置下 IP 全局广播分组不能够通过的路径是 (1)。

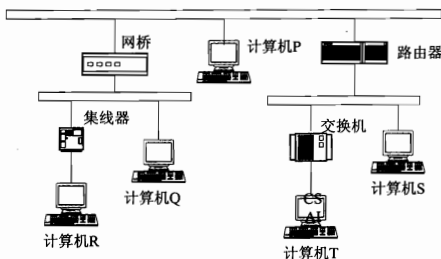


图 4-8 网络连接图

- (1) A. 计算机 P 和计算机 Q 之间的路径 B. 计算机 P 和计算机 S 之间的路径
C. 计算机 Q 和计算机 R 之间的路径 D. 计算机 S 和计算机 T 之间的路径

试题分析

在主干网上, 路由器的主要作用是路由选择。主干网上的路由器必须知道到达所有下层网络的路径。这需要维护庞大的路由表, 并对连接状态的变化做出尽可能迅速的响应。路由器的故障将会导致严重的信息传输问题。

在园区内部，路由器的主要作用是分隔子网。随着网络规模的不断扩大，局域网演变成以高速主干网和路由器连接的多个子网所组成的园区网。在其中，各个子网在逻辑上独立，而路由器就是唯一能够区分它们的设备，它负责子网间的报文转发和广播隔离，在边界上的路由器则负责与上层网络的连接。

交换机只能缩小冲突域，而不能缩小广播域。整个交换式网络就是一个大的广播域，广播报文散到整个交换式网络。而路由器可以隔离广播域，广播报文不能通过路由器继续进行广播。

P 和 S 被路由器隔开，属于不同的广播域。P 和 Q 中间有二层设备（网桥），不在同一个冲突域中，但在同一个广播域中。Q 和 R 中间只有一个一层设备（集线器），既在同一冲突域中，也在同一个广播域中。S 和 T 中间有一个二层设备（交换机），不在同一个冲突域中，但在同一个广播域中。因此，备选答案中，IP 全局广播分组不能够通过的只有计算机 P 和 S 之间的路径。

试题答案

(1) B

试题 2

确定网络的层次结构及各层采用的协议是网络设计中____(2)____阶段的主要任务。

- (2) 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

试题分析

显然，确定网络的层次结构及各层采用的协议是网络体系结构设计阶段的主要任务。

试题答案

(2) B

试题 3

在 IPv4 中，组播地址是____(3)____地址。

- (3) A. A 类 B. B 类 C. C 类 D. D 类

试题分析

IP 地址方案专门为组播划出一个地址范围，在 IPv4 中为 D 类地址，范围是 224. 0.0.0~239.255.255.255，并将 D 类地址划分为局部链接组播地址、预留组播地址和管理权限组播地址。

①局部链接地址：224. 0.0.0~224. 0.0.255，用于局域网，路由器不转发属于此范围的 IP 包。

②预留组播地址：224. 0.1. 0~238.255.255.255，用于全球范围或网络协议。

③管理权限地址：239. 0.0.0~239.255.255.255，组织内部使用，用于限制组播范围。

试题答案

(3) D

试题 4

Internet 中的协议应该满足规定的层次关系, 下面的选项中能正确表示协议层次和对应关系的是 (4)。

(4)

A.

SNMP	TFTP
UDP	TCP
IP	

B.

SNMP	HTTP
TCP	UDP
IP	

C.

HTTP	TFTP
TCP	UDP
IP	

D.

SMTP	TELNET
TCP	UDP
IP	

试题分析

Internet 协议要满足一定的封装关系, 上层协议封装在下层协议数据单元中传送。应用层协议 HTTP (超文本传输协议) 通过 TCP 连接发送, SNMP (简单网络管理协议) 利用 UDP 数据报传送, TFTP (简单文件传输协议) 也是利用 UDP 数据报传送, 而 TELNET (远程登录协议) 则是建立在 TCP 连接之上。

试题答案

(4) C

试题 5

POP3 协议采用 (5) 模式, 当客户机需要服务时, 客户端软件 (Outlook Express 或 FoxMail) 与 POP3 服务器建立 (6) 连接。

(5) A. Browser/Server B. Client/Server C. Peer to Peer D. Peer to Server

(6) A. TCP B. UDP C. PHP D. IP

试题分析

POP3 协议采用 Client/Server 模式, 当客户机需要服务时, 客户端软件 (Outlook Express 或 FoxMail) 与 POP3 服务器建立 TCP 连接。

试题答案

(5) B

(6) A

试题 6

在 Windows 操作系统中, 要实现一台具有多个域名的 Web 服务器, 正确的方法是 (7)。

- (7) A. 使用虚拟目录 B. 使用虚拟主机
C. 安装多套 IIS D. 为 IIS 配置多个 Web 服务端口

试题分析

在 Windows 操作系统中, Web 服务器只能安装一套 IIS 系统, 使用虚拟目录和多个 Web 服务器端口可以实现多个网站的发布, 但是其域名是相同的, 而使用虚拟主机可以实现一台具有多个域名和 Web 服务器。

试题答案

- (7) B

试题 7

当数据在两个 VLAN 之间传输时需要 (8) 设备。

- (8) A. 二层交换机 B. 网桥 C. 路由器 D. 中继器

试题分析

当数据在两个 VLAN 之间传输时需要通过路由器或三层交换机。因为每个 VLAN 是一个广播域, 在两个广播域之间进行数据交换需要第三层设备的支持。

试题答案

- (8) C

试题 8

某 DHCP 服务器设置的地址池 IP 从 192.36.96.101 到 192.36.96.150, 此时该网段下某 Windows 工作站启动后, 自动获得 169.254.220.167 这一 IP 地址, 这是因为 (9)。

- (9) A. DHCP 服务器提供保留的 IP 地址
B. DHCP 服务器不工作
C. DHCP 服务器设置租约时间太长
D. 网段内还有其他 DHCP 服务器, 工作站接到其他 DHCP 服务器响应的地址

试题分析

Windows 工作站启动后, 自动获得 169.254.220.167 地址, 而这个地址不在 DHCP 服务器设置的地址池范围内, 说明这个 DHCP 服务器没有工作。本题描述中所提到的 IP 地址 169.254.220.167 实际上是自动私有 IP 地址。当 DHCP 客户端无法与 DHCP 服务器通信时, 在 Windows 2000 以前的系统中, 如果计算机无法获取 IP 地址, 则自动配置成 IP 地址 0.0.0.0、子网掩码 0.0.0.0 的形式, 导致其不能与其他计算机进行通信。而对于 Windows 2000 以后的操作系统, 则在无法获取 IP 地址时自动配置成 IP 地址: 169.254. ×.×、子网掩码 255.255. 0.0 的形式, 这样, 可以使所有获取不到 IP 地址的计算机之间能够通信。

试题答案

(9) B

试题 9

在层次化网络设计方案中, 通常在 (10) 实现网络的访问策略控制。

(10) A. 应用层 B. 接入层 C. 汇聚层 D. 核心层

试题分析

在层次式网络设计方案中, 一般把网络分为 3 层, 分别是核心层、汇聚层和接入层。

核心层为网络提供了骨干组件或高速交换组件, 在分层设计中, 核心层只完成数据交换的特殊任务。核心层的功能主要是实现骨干网络之间的优化传输, 骨干层设计任务的重点通常是冗余能力、可靠性和高速的传输。网络的控制功能最好尽量少在骨干层上实施。核心层一直被认为是所有流量的最终承受者和汇聚者, 所以对核心层的设计及网络设备的要求十分严格。核心层设备将占投资的主要部分。

汇聚层是核心层和终端用户接入层的分界面, 汇聚层完成了网络访问策略控制、数据包处理、过滤、寻址, 以及其他数据处理的任务。汇聚层的功能主要是连接接入层节点和核心层中心。汇聚层设计为连接本地的逻辑中心, 仍需要较高的性能和比较丰富的功能。

接入层向本地网段提供用户接入, 是最终用户与网络的接口, 它应该提供即插即用的特性, 同时应该非常易于使用和维护。

从以上描述可以得知, 本题应选 C。

试题答案

(10) C

试题 10

在进行金融业务系统的网络设计时, 应该优先考虑 (11) 原则。

(11) A. 先进性 B. 开放性 C. 经济性 D. 高可用性

试题分析

先进性、开放性、经济性和高可用性都是网络设计时, 应该考虑的原则。但对于金融业务系统而言, 更加优先考虑高可用性。

试题答案

(11) D

试题 11

关于网络工程需求分析的论述, 正确的是 (12) 。

(12) A. 任何网络都不可能是一个能够满足各项功能需求的“万能网”
B. 必须采用最先进的网络设备, 获得最高的网络性能
C. 网络需求分析独立于应用系统的需求分析
D. 网络需求分析时可以先不考虑系统的扩展性

试题分析

我们在网络建设前都要做一个需求分析工作,否则,网络建立起来就带有盲目性,轻则造成网络资源浪费或网络瓶颈,重则使网络瘫痪,损失无法估量的数据资源。网络建设前的需求分析,就是要规划网络建设所要做的工作。根据用户提出的要求,进行网络的设计。可以这么说,网络建设的好坏、快慢、可持续发展性等,都将取决于网络实施前的规划工作。

①网络的功能要求。任何网络都不可能是一个可以进行各种各样工作的“万能网”,因此,必须针对每一个具体的网络,依据使用要求、实现成本、未来发展、总预算投资等因素仔细地反复推敲,尤其是分析出网络系统要完成的所有功能。

②网络的性能要求。根据对网络系统的相应时间、事物,处理的实时性进行研究,确定系统需要的存储量及备用的存储量。根据网络的工作站权限、容错程度、网络安全性方面的要求等,确定采取何种措施及方案。

③网络运行环境的要求。根据整个局域网运行时所需要的环境要求,确定使用哪种网络操作系统、应用系统,以及相应的应用软件和共享资源。

④网络的可扩充性和可维护性要求。如何增加工作站、怎样与其他网络联网、对软件/硬件的升级换代有何要求与限制等,都要在网络设计时加以考虑,以保证网络的可扩充性和可维护性。通常新建网络时都会给这个局域网提出一些有关使用寿命、维护代价等的要求。

试题答案

(12) A

试题 12

在 Linux 系统中,一般用__(13)___命令来测试另一台主机是否可达。

(13) A. ping B. ifconfig C. netstat D. uname

试题分析

本题考查的是 Linux 操作系统下常用网络管理与网络诊断命令的认识与使用。

①ping: 利用它可以检查网络是否能够连通,它可以很好地帮助我们分析判定网络故障。

②ifconfig: 调试计算机网络的常用命令,通常用它显示计算机中网络适配器的 IP 地址、子网掩码及默认网关等信息。

③netstat: 显示网络连接、路由表和网络接口信息,可以让用户得知目前都有哪些网络连接正在运行。

④uname: 显示当前操作系统名称。

试题答案

(13) A

试题 13

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（14）介质，所用时间最短。

- (14) A. 汽车 B. 双绞线 C. 多模光纤 D. 单模光纤

试题分析

一共有 25 盒磁带数据，每盒磁带数据量 40GB，则总共有 1000GB 的数据。由于甲、乙两地相距只有 1km，如果用汽车运输的话，按照 60km/h 计算，则只需要 1 分钟。而无论使用哪一种网络传输介质，都远远超过 1 分钟，因此，采用汽车运输，所用时间最短。

试题答案

- (14) A

试题 14

网络故障需按照协议层次进行分层诊断，找出故障原因并进行相应处理。查看端口状态、协议建立状态和 EIA 状态属于（15）诊断。

- (15) A. 物理层 B. 数据链路层 C. 网络层 D. 应用层

试题分析

物理层建立在通信媒体的基础上，实现系统和通信媒体的物理接口，为数据链路层实体之间进行透明传输，为建立、保持和拆除计算机和网络之间的物理连接提供服务。

物理层的故障转移表现在设备的物理连接方式是否恰当，以及连接电缆是否正确。确定路由器端口物理连接是否完好的最佳方法是使用 `show interface` 命令，检查每个端口的状态，解释屏幕输出信息，查看端口状态、协议建立状态和 EIA 状态。

试题答案

- (15) A

试题 15

（16）不是设备选型时应考虑的主要原则。

- (16) 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

试题分析

在进行设备的品牌、型号的选择时，应该考虑到产品技术指标、成本因素、原有设备的兼容性、产品的延续性、设备可管理性、厂商的技术支持等多方面的内容。在选择设备时，虽然要考虑技术的先进性，但并不是把是否采用最新技术作为主要原则。

试题答案

(16) D

试题 16

下列关于 Windows Server 2003 中域的叙述, 正确的是____(17)____。

- (17) A. 在网络环境中所有的计算机称为一个域
B. 同一个域中可以有多个备份域服务器
C. 每个域中必须有主域服务器和备份域服务器
D. 成员服务器支持活动目录数据库

试题分析

域(Domain)是一个共用“目录服务数据库”的计算机和用户的集合, 用于实现集中式管理。域是逻辑分组, 与网络的物理拓扑无关。域中只能有一个主域控制器, 但可以有零个或多个备份域控制器。

试题答案

(17) B

试题 17

以下网络存储模式中, 真正实现即插即用的是____(18)____。

- (18) A.DAS B.NAS C.Open SAN D.智能化 SAN

试题分析

即插即用, 顾名思义, 就是只要把设备接上, 不需要进行相关设置, 就可以直接使用。在以上给出的模式中, 只有 NAS 具备该功能。做题时, 首先可以排除的是 Open SAN 与智能化 SAN, 这两者都是需要进行复杂布署的。再看 DAS, DAS 是直接将存储连接在服务器上, 这种情况下, 是否为即插即用取决于硬件体系的支持, 以及是否有 RAID 的支持。而 NAS 的体系结构中, 存储是带了 CPU、内存等设备的, NAS 通过网线接入到网络中, 提供存储服务, 所以是即插即用型的。

试题答案

(18) B

试题 18

以下关于网络核心层的叙述中, 正确的是____(19)____。

- (19) A. 为了保障安全性, 应该对分组进行尽可能多的处理
B. 在区域间高速地转发数据分组
C. 由多台二、三层交换机组成
D. 提供多条路径来缓解通信瓶颈

试题分析

三层模型主要将网络划分为核心层、汇聚层和接入层，每一层都有着特定的作用：核心层提供不同区域或者下层的高速连接和最优传送路径；汇聚层将网络业务连接到接入层，并且实施与安全、流量负载和路由相关的策略；接入层为局域网接入广域网或者终端用户访问网络提供接入。其中，核心层是互联网络的高速骨干，由于其重要性，因此在设计中应该采用冗余组件设计，使其具备高可靠性，能快速适应变化。

在设计核心层设备的功能时，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的特性，以优化核心层获得低延迟和良好的可管理性。

核心层应具有有限的和一致的范围，如果核心层覆盖的范围过大，连接的设备过多，必然引起网络的复杂度加大，导致网络管理性降低；同时，如果核心层覆盖的范围不一致，必然导致大量处理不一致情况的功能都在核心层网络设备中实现，会降低核心网络设备的性能。

对于那些需要连接因特网和外部网络的网络工程来说，核心层应包括一条或多条连接到外部网络的连接，这样可以实现外部连接的可管理性和高效性。

试题答案

(19) B

试题 19

网络开发过程中，物理网络设计阶段的任务是__ (20) __。

- (20) 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

试题分析

网络的生命周期至少包括网络系统的构思计划、分析设计、实时运行和维护的过程。对于大多数网络系统来说，由于应用的不断发展，这些网络系统需要不断重复设计、实施、维护的过程。

网络逻辑结构设计是体现网络设计核心思想的关键阶段，在这一阶段根据需求规范和通信规范，选择一种比较适宜的网络逻辑结构，并基于该逻辑结构实施后续的资源分配规划、安全规划等内容。

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、运行环境等的确定，确保网络的物理连接符合逻辑连接的要求。在这一阶段，网络设计者需要确定具体的软硬件、连接设备、布线和服务。

现有网络体系分析的工作目的是描述资源分布，以便于在升级时尽量保护已有投资，通过该工作可以使网络设计者掌握网络现在所处的状态和情况。

需求分析阶段有助于设计者更好地理解网络应该具有什么功能和性能，最终设计出符合用户需求的网络，它为网络设计提供依据。

试题答案

(20) A

试题 20

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__(21)__存储方式最为合适。

- (21) A. 内置式存储 B. DAS
C. SAN D. NAS

试题分析

开放系统的直连式存储（Direct-Attached Storage, DAS）在服务器上外挂了一组大容量硬盘，存储设备与服务器主机之间采用 SCSI 通道连接，带宽为 10MB/s、20MB/s、40MB/s 和 80MB/s 等。直连式存储直接将存储设备连接到服务器上，这种方法难以扩展存储容量，而且不支持数据容错功能，当服务器出现异常时会造成数据丢失。

网络接入存储（Network Attached Storage, NAS）是将存储设备连接到现有的网络上，提供数据存储和文件访问服务的设备。NAS 服务器是在专用主机上安装简化了的瘦操作系统（只具有访问权限控制、数据保护和恢复等功能）的文件服务器。NAS 服务器内置了与网络连接所需要的协议，可以直接联网，具有权限的用户都可以通过网络访问 NAS 服务器中的文件。

存储区域网络（Storage Area Network, SAN）是一种连接存储设备和存储管理子系统的专用网络，专门提供数据存储和管理功能。SAN 可以被看做是负责数据传输的后端网络，而前端网络（或称为数据网络）则负责正常的 TCP/IP 传输。也可以把 SAN 看做是通过特定的互联方式连接的若干台存储服务器组成的单独的数据网络，提供企业级的数据存储服务。

试题答案

(21) C

4.3 实战练习题

- 关于 FTP 和 TFTP 的描述，正确的是__(1)__。

- (1) A. FTP 和 TFTP 都是基于 TCP
B. FTP 和 TFTP 都是基于 UDP
C. FTP 基于 TCP、TFTP 基于 UDP
D. FTP 基于 UDP、TFTP 基于 TCP

- __(2)__是错误的网络设备选型原则。

- (2) A. 选择网络设备，应尽可能选择同一厂家的产品。

B. 为了保证网络先进性, 尽可能选择性能高的产品

C. 核心设备的选取要考虑系统日后的扩展性

D. 网络设备选择要充分考虑其可靠性

● 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, (3) 安全性需求相对较弱。

(3) A. 政府网

B. 校园网

C. 企业网

D. 金融网

● 服务器的部署是网络规划的重要环节。某单位网络拓扑结构如图 4-9 所示, 需要部署 VOD 服务器、Web 服务器、邮件服务器, 此外还需要部署流量监控服务器对单位内部网络流量进行监控。VOD 服务器应部署在位置 (4), Web 服务器应部署在位置 (5), 流量监控服务器应部署在位置 (6)。上述服务器中, 流出流量最大的是 (7), 流入流量最大的是 (8)。

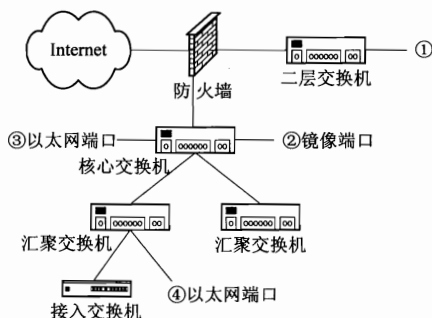


图 4-9 某单位网络拓扑结构

(4) A. ①

B. ②

C. ③

D. ④

(5) A. ①

B. ②

C. ③

D. ④

(6) A. ①

B. ②

C. ③

D. ④

(7) A. VOD 服务器

B. Web 服务器

C. 流量监控服务器

D. 邮件服务器

(8) A. VOD 服务器

B. Web 服务器

C. 流量监控服务器

D. 邮件服务器

● 在层次化网络设计方案中, (9) 是核心层的主要任务。

(9) A. 高速数据转发

B. 接入 Internet

C. 工作站接入网络

D. 实现网络的访问策略控制

● 对于校园网来说, (10)。

(10) A. 其核心是网络环境, 利用网络的人是关键

B. 其核心是应用, 网络教学资源是根本

C. 网络基础平台是否先进是评价校园网成功与否的重要指标

D. 校园网视其地域的大小, 可以是局域网, 也可以是广域网

● 设计骨干网时, 应该在性能和成本之间寻找平衡。以下叙述中, (11) 是正确的。

- (11) A. ATM 在局域网的所有应用可用 ELAN 来实现, 带宽效率高, 实时性好, 适宜用作园区网
B. 双星树结构的主干网, 虽然不能均衡负载, 成本也较高, 但具有更高的可用性
C. 由于建筑群布线路径复杂的特殊性, 一般直线距离超过 300 米的建筑物之间的千兆位以太网线路就必须用单模光纤
D. 如经费难以支持千兆位以太网, 可以采用 100Base-FX, 用双绞线建立快速以太网, 是非常经济实惠的选择

● 下面关于网络系统设计原则的论述, 正确的是 (12)。

- (12) A. 应尽量采用先进的网络设备, 获得最高的网络性能
B. 网络总体设计过程中, 只需要考虑近期目标即可, 不需要考虑扩展性
C. 网络系统应采用开放的标准和技术
D. 网络需求分析独立于应用系统的需求分析

● 在进行金融业务系统的网络设计时, 应该优先考虑 (13) 原则。在进行企业网络的需求分析时, 应该首先进行 (14)。

- (13) A. 先进性 B. 开放性 C. 经济性 D. 高可用性

- (14) A. 企业应用分析 B. 网络流量分析
C. 外部通信环境调研 D. 数据流向图分析

● 为了避免备份数据, 或转移存储数据占用过高网络带宽从而影响业务系统正常运作, (15) 首次采用了业务网络与存储网络分开的结构。

- (15) A. SAN B. NAS C. SCSI D. DAS

● 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 (16)。

- (16) 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 (17)。

- (17) 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 (18)。

- (18) 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

● 建筑物综合布线系统中的垂直子系统是指____(19)_____。

- (19) A. 由终端到信息插座之间的连线系统
- B. 楼层接线间的配线架和线缆系统
- C. 各楼层设备之间的互连系统
- D. 连接各个建筑物的通信系统

4.4 练习题解析

试题 1 分析

FTP 是网络上两台计算机传送文件的协议, 是通过 Internet 把文件从客户机复制到服务器上的一种途径。FTP 基于 TCP。

TFTP 是用来在客户机与服务器之间进行简单文件传输的协议, 提供不复杂、开销不大的文件传输服务。TFTP 设计的时候是进行小文件传输的。因此它不具备通常的 FTP 的许多功能, 它只能从文件服务器上获得或写入文件, 不能列出目录, 不进行认证, 它传输 8 位数据。TFTP 基于 UDP。

试题 1 答案

(1) C

试题 2 分析

对于中小规模的网络, 设备选型时应遵循以下一些基本原则:

① 标准化原则。所选择的设备必须基于国际标准或行业标准。因为只有基于标准的产品才有可能与其他厂商的产品互连互通 (需要指出的是, 并非只要基于标准的产品, 彼此之间才可以互连互通)。

② 技术简单性原则。对网络需求必须十分明确。对于普通用户而言, 在满足需求的前提下, 尽可能选择简单实用的技术和设备。否则, 今后的运行管理、故障诊断等, 都需要请专业人员, 开销巨大, 运行效果不一定好。例如, 选择设备时, 尽量选择同一厂家的设备, 这样实现起来会更加简单。

③ 环境适应性原则。不要轻信国外某些机构的评测报告, 其中不乏商业因素。而且, 即使是权威机构的评测报告, 也只是在特定网络环境下取得的结果, 不能作为产品选型的全部依据。

④ 可扩展性原则。对于核心设备的选择, 需要考虑日后的扩展性。

对于大型网络, 往往技术要求新、应用需求多, 可以根据自身网络环境的要求, 结合具体的设备和技术, 量身定做。在借鉴上述原则的基础上, 还需要考虑以下因素:

⑤可管理性原则。对于大型网络而言,这一点是至关重要的,它不仅关系到系统的性能指标,甚至关系到系统的可用性。主要考查网管系统对所选设备的监管、配置能力,以及设备可以提供的统计信息和故障检测手段,如骨干交换机必须具备端口镜像能力。这对于故障诊断,以及今后的网络规划具有特别重要的价值。

⑥容错冗余性原则。除了在网络设计时要考虑冗余,骨干设备的容错冗余也是必需的。所谓容错,就是设备的某一模块出现故障时,是否会影响其他模块,乃至其他设备的正常工作;是否支持热插拔;是否支持备份设备的自动切换等。所谓冗余,就是配置的设备,是否可以安装多个相同功能的模块,在工作正常的情况下实施负载分担,当其中一个出现问题时自动切换。

试题2 答案

(2) B

试题3 分析

网络系统的设计是受经费限制的,在考虑安全解决方案时必须考虑性能价格的平衡,而且不同的网络系统所要求的安全侧重点各有不同。在企业网、政府行政办公网、国防军工部门内部网、电子商务网站及VPN等网络方案设计中应重点体现安全性原则,确保网络系统和数据的安全运行。在社区网、城域网和校园网中,安全性的需求相对较弱。

试题3 答案

(3) B

试题4~8 分析

因为Web服务器是通过Internet供公众访问的,所以,它应该放在防火墙后面,即部署在位置①。流量监控服务器用来监视整个网络的流量情况,根据流量来更好地管理服务器,所以应部署在位置②。

因为VOD是视频点播,从用户端流入的只是简单的指令数据,而流出的是以G单位的视频数据,所以VOD服务器的流出流量是最大的。同样,Web服务器的流入也只是一些请求命令和交互命令数据。邮件服务器流入的是邮件数据,就单个单位而言,这个流量也比较小。流量监控服务器即要监控流入流量,也要监控流出流量,所以,它的流入流量是最大的。

试题4~8 答案

(4) C

(5) A

(6) B

(7) A

(8) C

试题9 分析

在层次化网络设计方案中,一般把网络分为3层,分别是核心层、汇聚层和接入层。

核心层为网络提供了骨干组件或高速交换组件,在分层设计中,核心层只完成数据交换的特殊任务。核心层的功能主要是实现骨干网络之间的优化传输,骨干层设计任务的重点通常是冗余能力、可靠性和高速的传输。网络的控制功能最好尽量在骨干层上实施。核心层一直被认为是所有流量的最终承受者和汇聚者,所以对核心层的设计及网络设备的要求十分严格。核心层设备将占投资的主要部分。

汇聚层是核心层和终端用户接入层的分界面,汇聚层完成了网络访问策略控制、数据包处理、过滤、寻址,以及其他数据处理的任务。汇聚层的功能主要是连接接入

层节点和核心层中心。汇聚层设计为连接本地的逻辑中心，仍需要较高的性能和比较丰富的功能。

接入层向本地网段提供用户接入，是最终用户与网络的接口，它应该提供即插即用的特性，同时应该非常易于使用和维护。

试题 9 答案

(9) A

试题 10 分析

对于任何网络而言，其核心都是应用。网络犹如高速公路，网络应用就好比是车辆，修路的目的是为了跑车。一般来说，校园网是一个局域网。对于校园网而言，其主要目的是为师生提供丰富的教学资源。要评价校园网是否成功，就要看是否达到了建设的目标，是否有适合于校园的应用，是否有丰富的教学资源供师生共享。

试题 10 答案

(10) B

试题 11 分析

骨干网（主干网）一般用来连接建筑群和服务器群，是网络的大动脉。骨干网技术的选择，要根据需求分析中的地理距离、信息流量和数据负载的轻重而定。连接建筑群的骨干网一般以光缆作为传输介质。

FDDI 基本属于过时技术，目前较少应用；ATM 是面向连接的网络，实时传输效率高，但在局域网的应用需要 ELAN 仿真来实现，效率低，不适宜用作局域网或园区网。千兆位以太网一般采用光缆作为传输介质，多种波长的单模和多模光纤分别用于不同的场合和距离，由于建筑群布线线路路径复杂的特殊性，一般直线距离超过 300 米的建筑物之间的千兆位以太网线路就必须要用单模光纤。如经费难以支持千兆位以太网，可以采用 100Base-FX，即用光纤建立快速以太网，其端口价格低，对光缆的要求也不高。

试题 11 答案

(11) C

试题 12 分析

本题关于网络系统设计的原则，只有选项 C“网络系统应采用开放的标准和技术”是正确的，其他均有误。下面分析其他选项的错误原因：

“A. 应尽量采用先进的网络设备，获得最高的网络性能”，这种说法不对，虽然采用先进的网络设备能获得不错的网络性能，但同时会增加网络的建设成本，在进行网络系统设计时应考虑成本因素，不能一味追求性能上的提升，而不计成本。

“B. 网络总体设计过程中，只需要考虑近期目标即可，不需要考虑扩展性”，这种说法不对，计算机网络技术发展非常快，应用过程中也会有需求不断提出，所以系统设计时需要考虑扩展性。

“D. 网络需求分析独立于应用系统的需求分析”，这种说法不对，网络系统不是一个独立的系统，它与应用系统有着密切的联系，如果在进行网络需求分析时独立于应用系统的需求分析，则最终完成的网络系统肯定与应用系统不能很好地协作。

试题 12 答案

(12) C

试题 13~14 分析

可用性、有效性和安全性是金融业务核心系统架构中被着重关注的三方面。数据量大、数据类型多样、业务需求多样、业务需求变化快和子系统繁多是金融业务的特点，因此金融业务核心系统架构中，可用性、有效性和安全性尤为重要。在复杂的金融业务环境中，只采用片面的策略来提高系统单方面的性能，会导致系统性能失衡，整体性能降低。因此在金融业务核心系统架构中要采用一定的策略保持可用性、有效性和安全性的平衡，以提升系统整体性能。而在进行网络设计时，其网络的高可用性是设计优先考虑。

企业内部网络的建设已经成为提升企业核心竞争力的关键因素。企业网已经越来越多地被人们提到，利用网络技术，现代企业可以在供应商、客户、合作伙伴、员工之间实现优化的信息沟通。这直接关系到企业能否获得关键的竞争优势。企业网络要求具有资源共享功能、通信服务功能、多媒体功能、远程 VPN 拨入访问功能。

所以在进行企业网络的需求分析时，对企业的需求、应用范围、基于的技术等，要从企业应用来进行分析。

试题 13~14 答案

(13) D

(14) A

试题 15 分析

SAN 结构中，业务网络与存储网络是分开的，并且是不同类型的网络，这样使得业务数据请求的传输不受存储数据传输的影响。

试题 15 答案

(15) A

试题 16 分析

本题主要考查大型局域网的层次和各个层次的功能，大型局域网通常划分为核心层、汇聚层和接入层，其中核心层在逻辑上只有一个，它连接多个分布层交换机，通常是一个园区中连接多个建筑物的总交换机的核心网络设备；汇聚层定义的网络的访问策略；接入层提供局域网接入功能，可以使用集线器代替交换机。

试题 16 答案

(16) A

试题 17 分析

本题主要考查网络设计方面的基础知识。根据网络系统设计的一般规则，在逻辑网络设计阶段的任务通常是根据需求规范和通信规范，实施资源分配和安全规划。其他几个选项都不是逻辑网络设计阶段的任务。

试题 17 答案

(17) C

试题 18 分析

本题主要考查网络系统生命周期的基础知识。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需求规范、通信规范、逻辑网络设计、物理网络设计、实施阶段。

试题 18 答案

(18) A

试题 19 分析

综合布线分六大子系统。

1. 工作区子系统 (Worklocation): 目的是实现工作区终端设备与水平子系统之间的连接, 由终端设备连接到信息插座的连接线缆所组成。工作区常用设备是计算机、网络集线器 (Hub 或 Mau)、电话、报警探头、摄像机、监视器、音响等。

2. 水平子系统 (Horizontal): 目的是实现信息插座和管理子系统 (跳线架) 间的连接, 将用户工作区引至管理子系统, 并为用户提供一个符合国际标准, 满足语音及高速数据传输要求的信息点出口。该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 (Administration): 本子系统由交连、互连配线架组成。管理间为连接其他子系统提供连接手段。交连和互连允许将通信线路定位或重定位到建筑物的不同部分, 以便能更容易地管理通信线路, 使在移动终端设备时能方便地进行插拔。互连配线架根据不同的连接硬件分楼层配线架 (箱) IDF 和总配线架 (箱) MDF, IDF 可安装在各楼层的干线接线间, MDF 一般安装在设备机房。

4. 垂直干线子系统 (Backbone): 目的是实现计算机设备、程控交换机 (PBX)、控制中心与各管理子系统间的连接, 是建筑物干线电缆的路由。该子系统通常是两个单元之间, 特别是在位于中央点的公共系统设备处提供多个线路设施。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。常用介质是大对数双绞线电缆和光缆。

5. 设备室子系统 (Equipment): 本子系统主要由设备间中的电缆、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。设备包括计算机系统、网络集线器 (Hub)、网络交换机 (Switch)、程控交换机 (PBX)、音响输出设备、闭路电视控制装置和报警控制中心等。

6. 建筑群子系统 (Campus): 该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。

试题 19 答案

(19) C

5

第 5 章

系统安全性与保密性设计

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

(1) 信息安全与保密

- 加密和解密。
- 身份认证（数字签名、密钥、口令）。
- 访问控制。
- 安全保密管理（防泄露、数字水印）。
- 安全协议（SSL、PGP、IPSec）。
- 系统备份与恢复。
- 防治病毒。

(2) 安全性规章与保护私有信息规则

- 信息系统安全法规与制度。
- 计算机防病毒制度。
- 保护私有信息规则。

5.1 考点突破

从历年的考试情况来看，本章的考点主要集中在安全相关的基本概念及网络安全方面。

5.1.1 历年考试情况分析

在历年的考试试题中，有关系统安全性与保密性设计知识的试题如表 5-1 所示。

表 5-1 系统安全性与保密性设计知识试题分布表

题号	2009.11	2010.11	2011.11
64	安全策略	网络安全漏洞	SNMP v3
65	对称加密算法	ARP	PGP

按照知识点进行总结和归类的试题分布情况如表 5-2 所示。

表 5-2 系统安全性与保密性设计知识归类表

知识点	2009.11	2010.11	2011.11
安全基础技术	1	0	1
网络安全	0	2	1
计算机病毒与木马	0	0	0
其他	1	0	0
合计	2	2	2

从表 5-2 中可以看出,系统安全性与保密性设计知识方面的内容在历年的考试中分值非常稳定,一直稳定维持在 2 分。所占分数比例的趋势如图 5-1 所示。

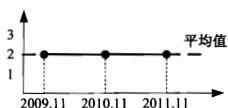


图 5-1 系统安全性与保密性设计知识历年试题比例趋势图

5.1.2 安全基础技术

在系统架构设计师的考试中,经常会考到与安全相关的一些基础概念及技术原理,这些知识将在本节进行详细描述。

1. 对称加密

对称加密采用了对称密码编码技术,它的特点是文件加密和解密使用相同的密钥,即加密密钥也可以用作解密密钥。对称加密算法的优点是:使用简单、加密解密快捷高效,而它也有致命的弱点:加密强度不高、密钥分发困难。

常见对称密钥加密算法如下。

- DES: 该算法主要采用了替换和移位的方式,密钥长度是 56 位,每次运算对 64 位数据块进行加密,该算法运行速度快、密钥易产生,曾是应用最为广泛的一种对称加密算法。
- 3DES: 又称三重 DES,该算法巧妙地利用了原有的 DES 算法,它使用两个 56 位的密钥 K1、K2,在加密操作时, K1 加密→K2 解密→K1 加密。解密操作时, K1 解密→K2 加密→K1 解密。这样做,相当于密钥长度加倍。
- 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。

2. 非对称加密

前面提到了对称加密技术，其中“对称”的意思是加密与解密使用了同样的密钥。顾名思义，非对称加密技术中所使用的加密与解密密钥是不同的。在非对称加密体系中，密钥是成对出现的，一对密钥包括一个公钥和一个私钥。如果用公开密钥对数据进行加密，只有用对应的私有密钥才能解密；如果用私有密钥对数据进行加密，那么只有用对应的公开密钥才能解密。非对称加密算法的优点在于：解决了对称密钥加密强度不高及密钥分发困难的问题，其缺点是：加密速度极慢，可对比称加密慢上千倍，甚至于数千倍。所以非对称加密算法通常只对极小的数据量进行加密，如对信息摘要进行加密，或用于加密对称密钥。

常见非对称密钥加密算法如下。

- RSA：最为常见的非对称加密算法，512 位密钥（或 1024 位密钥）、计算量极大、难破解。
- ECC：又称椭圆曲线密码体制，该算法是由 Sun 公司提出的。

3. 信息摘要

信息摘要简要地描述了一份较长的信息或文件，它可以被看做是一份长文件的“数字指纹”，信息摘要可以用于创建数字签名。

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。

4. 数字签名

数字签名（又称公钥数字签名、电子签章）类似于写在纸上的普通物理签名，但是使用了非对称加密技术实现，用于鉴别数字信息的作者身份。一套数字签名通常定义两种互补的运算，一个用于签名，另一个用于验证。

数字签名技术是对非对称加密技术与信息摘要的综合应用。通常的做法是：先对正文产生信息摘要，之后使用发送者 A 的私钥对该信息摘要进行加密，这就完成了签名。当接收者 B 收到签了名的摘要以后，会对摘要使用发送者 A 的公钥进行解密，若能解密，则表明该信息确实是由 A 发送的。以此保障信息的完整性（即不被篡改），以及保障信息的不可抵赖性（发送者 A 不能否认自己发过该信息。因为 B 收到的摘要能用 A 的公钥解密，这说明摘要用的是 A 的私钥加密的，而 A 的私钥只有 A 拥有）。

5. 数字证书

非对称加密技术的提出，解决了密钥传输的问题，但在实际应用过程中，我们遇到了一个新的问题：密钥只是一串数字或字符，并不能通过密钥得知它的主人是谁。这样就产生了安全隐患，所以提出了数字证书的概念。数字证书简单一点理解，就是密钥与身份信息的结合体。

数字证书的格式一般使用 X.509 国际标准。X.509 是广泛使用的证书格式之一，X.509 用户公钥证书是由可信赖的证书权威机构（CA——证书授权中心）创建的，并且由 CA 或用户存放在 X.500 的目录中。任何一个用户只要得到 CA 中心的公钥，就可以得到该 CA 中心为该用户签署的公钥。因为证书是不可伪造的，因此对于存放证

书的目录无须施加特别的保护。

因为用户数量多，因此会存在多个 CA 中心。但如果两个用户使用的是不同 CA 中心发放的证书，则无法直接使用证书；但如果两个证书发放机构之间已经安全地交换了公开密钥，则可以使用证书链来完成通信。

较通行的数字证书格式还有 PGP。

5.1.3 网络安全

网络安全知识是指与网络相关的一些安全性知识，包括：网络安全协议、常见的网络攻击、入侵检测技术、防火墙技术、VPN 技术等。

1. 网络安全协议

互联网是人类的一大创举，它将全世界各个地方的人及资源都联系到了一起。但同时，也为这些资源带来了很大的安全隐患。最初人们并未意识到这一点，只是简单地考虑如何将各地的计算机联系起来，将资源充分地共享，导致设计的通信协议都是没有考虑安全问题的。这些协议传输信息都是采用明文，如我们熟知的 IP、TCP、HTTP 等。所以后来针对这种情况开发了一系列的安全协议作为补充，我们需要了解这些协议的工作层次，以及基本特性，它们的工作层次关系如图 5-2 所示。

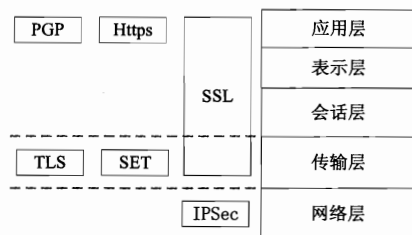


图 5-2 安全协议层次图

(1) PGP 协议

PGP (Pretty Good Privacy) 是一个基于 RSA 公钥加密体系的邮件加密协议。可以用它对邮件保密以防止非授权者阅读，它还能对邮件加上数字签名从而使收信人可以确信邮件发送者。PGP 采用了审慎的密匙管理，一种 RSA 和传统加密的杂合算法：一个对称加密算法 (IDEA)、一个非对称加密算法 (RSA)、一个单向散列算法 (MD5) 及一个随机数产生器 (从用户击键频率产生伪随机数序列的种子)，用于数字签名的邮件文摘算法，加密前压缩等，还有一个良好的人机工程设计。它的功能强大，有很快的速度。而且它的源代码是免费的。PGP 还可用于文件存储的加密。PGP 承认两种不同的证书格式：PGP 证书和 X.509 证书。

(2) SSL 协议

SSL (Security Socket Layer) 协议是 Netscape Communication 开发的安全协议，它工作于传输层以上 (跨越了多个网络层次)，用于在 Internet 上传送机密文件。SSL 协议由 SSL 记录协议、SSL 握手协议和 SSL 警报协议组成。

SSL 握手协议被用来在客户与服务器真正传输应用层数据之前建立安全机制，当客户与服务器第一次通信时，双方通过握手协议在版本号、密钥交换算法、数据加密算法和 Hash 算法上达成一致，然后互相验证对方身份，最后使用协商好的密钥交换算法产生一个只有双方知道的秘密信息，客户和服务器各自根据该秘密信息产生数据加密算法和 Hash 算法参数。

SSL 记录协议根据 SSL 握手协议协商的参数，对应用层送来的数据进行加密、压缩、计算消息鉴别码，然后经网络传输层发送给对方。

SSL 警报协议用来在客户和服务器之间传递 SSL 出错信息。

SSL 协议主要提供以下 3 方面的服务：

- 用户和服务器的合法性认证。认证用户和服务器的合法性，使得它们能够确信数据将被发送到正确的客户机和服务器上。客户机和服务器都有各自的识别号，这些识别号由公开密钥进行编号，为了验证用户是否合法，SSL 协议要求在握手交换数据时进行数字认证，以此来确保用户的合法性。
- 加密数据以隐藏被传送的数据。SSL 协议所采用的加密技术既有对称密钥技术，也有公开密钥技术。在客户机与服务器进行数据交换之前，交换 SSL 初始握手信息，在 SSL 握手信息中采用了各种加密技术对其加密，以保证其机密性和数据的完整性，并且用数字证书进行鉴别，这样就可以防止非法用户进行破译。
- 保护数据的完整性。SSL 协议采用 Hash 函数和机密共享的方法来提供信息的完整性服务，建立客户机与服务器之间的安全通道，使所有经过 SSL 协议处理的业务在传输过程中能全部完整、准确无误地到达目的地。

SSL 协议是一个保证计算机通信安全的协议，对通信对话过程进行安全保护，其实现过程主要经过如下几个阶段。

- 1) 接通阶段：客户机通过网络向服务器打招呼，服务器回应。
- 2) 密码交换阶段：客户机与服务器之间交换双方认可的密码，一般选用 RSA 密码算法，也有的选用 Diffie-Hellman 和 Fortezza-KEA 密码算法。
- 3) 会谈密码阶段：客户机器与服务器间产生彼此会谈的会谈密码。
- 4) 检验阶段：客户机检验服务器取得的密码。
- 5) 客户认证阶段：服务器验证客户机的可信度。
- 6) 结束阶段：客户机与服务器之间相互交换结束的信息。

发送时信息用对称密钥加密，对称密钥用不对称算法加密，再把两个包绑在一起传送过去。接收的过程与发送正好相反，先打开有对称密钥的加密包，再用对称密钥解密。因此，SSL 协议也可用于安全电子邮件。

(3) SET 协议

SET (Secure Electronic Transaction, 安全电子交易) 协议向基于信用卡进行电子化交易的应用提供了实现安全措施的规则。它是由 VISA 国际组织和 MasterCard 组织共同制定的一个能保证通过开放网络 (包括 Internet) 进行安全资金支付的技术标准。

SET 支付系统主要由持卡人 (Cardholder)、商家 (Merchant)、发卡行 (Issuing Bank)、收单行 (Acquiring Bank)、支付网关 (Payment Gateway)、认证中心 (Certificate Authority, CA) 共 6 个部分组成。对应的, 基于 SET 协议的网上购物系统至少包括电子钱包软件、商家软件、支付网关软件和签发证书软件。

SET 协议的工作流程如下:

1) 消费者利用自己的 PC 通过 Internet 选定所要购买的物品, 并在计算机上输入订货单, 订货单上需包括在线商店、购买物品名称及数量、交货时间及地点等相关信息。

2) 通过电子商务服务器与有关在线商店联系, 在线商店做出应答, 告诉消费者所填订货单的货物单价、应付款数、交货方式等信息是否准确, 是否有变化。

3) 消费者选择付款方式, 确认订单签发付款指令。此时 SET 开始介入。

4) 在 SET 中, 消费者必须对订单和付款指令进行数字签名, 同时利用双重签名技术保证商家看不到消费者的账号信息。

5) 在线商店接受订单后, 向消费者所在银行请求支付认可。信息通过支付网关到收单银行, 再到电子货币发行公司确认。批准交易后, 返回确认信息给在线商店。

6) 在线商店发送订单确认信息给消费者。消费者端软件可记录交易日志, 以备将来查询。

7) 在线商店发送货物或提供服务并通知收单银行将钱从消费者的账号转移到商店账号, 或通知发卡银行请求支付。在认证操作和支付操作中间一般会有一个时间间隔, 例如, 在每天的下班前请求银行结一天的账。

前两步与 SET 无关, 从第 3) 步开始 SET 起作用, 一直到第 6) 步, 在处理过程中通信协议、请求信息的格式、数据类型的定义等 SET 都有明确的规定。在操作的每一步, 消费者、在线商店、支付网关都通过 CA 来验证通信主体的身份, 以确保通信的对方不是冒名顶替, 所以, 也可以简单地认为 SET 规格充分发挥了认证中心的作用, 以维护在任何开放网络上的电子商务参与者所提供信息的真实性和保密性。

(4) IPSec 协议

IPSec 工作于网络层。它有两种加密方式: 第一种, 是将原 IP 数据包进行整体加密 (包括 IP 包头), 然后产生新的 IP 包进行传送; 第二种, 是将原 IP 数据包中的数据取出来, 加密, 然后通过原 IP 包头信息产生新的 IP 头。区别在于一个是整体加密, 另一个是数据加密。

IPSec 为 IP 网络通信提供透明的安全服务, 保护 TCP/IP 通信免遭窃听和篡改, 可以有效抵御网络攻击, 同时保持易用性。IPSec 有两个基本目标, 分别是保护 IP 数据包安全和为抵御网络攻击提供防护措施。IPSec 结合密码保护服务、安全协议组和动态密钥管理三者来实现上述两个目标, 不仅能为企业局域网与拨号用户、域、网站、远程站点, 以及 Extranet 之间的通信提供强有力且灵活的保护, 而且还能用来筛选特定数据流。

(5) TLS 协议

安全传输层协议 (TLS) 用于在两个通信应用程序之间提供保密性和数据完整性。

该协议由两层组成：TLS 记录协议（TLS Record）和 TLS 握手协议（TLS Handshake）。较低的层为 TLS 记录协议，位于某个可靠的传输协议（如 TCP）上面。TLS 记录协议提供的连接安全性具有以下两个基本特性：

- 私有——对称加密用以数据加密（DES、RC4 等）。对称加密所产生的密钥对每个连接都是唯一的，且此密钥基于另一个协议（如握手协议）协商。记录协议也可以不加密使用。
- 可靠——信息传输包括使用密钥的 MAC 进行信息完整性检查。安全哈希功能（SHA、MD5 等）用于 MAC 计算。记录协议在没有 MAC 的情况下也能操作，但一般只能用于这种模式，即有另一个协议正在使用记录协议传输协商安全参数。

2. 网络攻击

网络攻击可以分为两大类：被动攻击与主动攻击，其中主动攻击又可细化为中断、篡改和伪造。它们的工作原理如图 5-3 所示。



图 5-3 网络攻击类型示意图

被动攻击：主要是监听，如在电话线上接出一个分支，接上电话，便能监测到别人的通话。

中断（主动攻击）：通过阻隔源站向目的站点之间的通信，达到攻击的目的。例如，剪断电话线（网线），DDoS 攻击，当带宽被占满时，能产生服务器中断的效果。

篡改（主动攻击）：先获取源站点的信息，将信息进行修改，再以源站的身份向目的站点发送。例如，中间人攻击就属于这种形式。

伪造（主动攻击）：源站点并未向目的站点发送信息，在攻击方以源站点的身份向目的站点发送信息。

常见的网络攻击有以下几种。

（1）中间人攻击（MITM 攻击）

中间人攻击（Man-in-the-Middle Attack，简称“MITM 攻击”）是一种“间接”的入侵攻击，这种攻击模式是通过各种技术手段将受入侵者控制的一台计算机虚拟放置在网络连接中的两台通信计算机之间，这台计算机就称为“中间人”。然后入侵者把这台计算机模拟一台或两台原始计算机，使“中间人”能够与原始计算机建立活动连接并允许其读取或修改传递的信息，然而两个原始计算机用户却认为它们是在互相通信。通常，这种“拦截数据—修改数据—发送数据”的过程就被称为“会话劫持”（Session Hijack）。

（2）DDoS 攻击

DDoS 是 Distributed Denial of Service 的缩写，意思是“分布式拒绝服务”，那么

什么是拒绝服务（Denial of Service）呢？可以这么理解，凡是能导致合法用户不能够访问正常网络服务的行为都算是拒绝服务攻击。也就是说，拒绝服务攻击的目的非常明确，就是要阻止合法用户对正常网络资源的访问，从而达到攻击者不可告人的目的。DDoS 攻击就是通过很多“肉鸡”（被攻击者入侵过或可间接利用的主机）向受害主机发送大量看似合法的网络包，从而造成网络阻塞或服务器资源耗尽而导致拒绝服务，分布式拒绝服务攻击一旦被实施，攻击网络包就会犹如洪水般涌向受害主机，从而把合法用户的网络包淹没，导致合法用户无法正常访问服务器的网络资源。因此，拒绝服务攻击又被称为“洪水式攻击”。常见的 DDoS 攻击手段有 SYNflood、ACKflood、UDPFlood、ICMPflood、TCPflood、ConnectionsFlood、ScriptFlood、ProxyFlood 等。

（3）MAC/CAM 攻击

交换机主动学习客户端的 MAC 地址，并建立和维护端口和 MAC 地址的对应表，以此建立交换路径，这个表就是通常我们所说的 CAM 表。CAM 表的大小是固定的，不同的交换机的 CAM 表大小不同。MAC/CAM 攻击是指利用工具产生欺骗 MAC，快速填满 CAM 表，交换机 CAM 表被填满后，交换机以广播方式处理通过交换机的报文，这时攻击者可以利用各种嗅探攻击获取网络信息。CAM 表满了后，流量以洪泛方式发送到所有接口，也就代表 TRUNK 接口上的流量也会发给所有接口和邻接交换机，会造成交换机负载过大、网络缓慢和丢包，甚至瘫痪。

3. 入侵检测

入侵检测技术是为保证计算机系统的安全而设计与配置的一种能够及时发现并报告系统中未授权或异常现象的技术，是一种用于检测计算机网络中违反安全策略行为的技术。违反安全策略的行为有：入侵——非法用户的违规行为；滥用——用户的违规行为。

入侵检测系统所采用的技术可分为特征检测与异常检测两种。

（1）特征检测

特征检测也称为误用检测，假设入侵者活动可以用一种模式来表示，系统的目标是检测主体活动是否符合这些模式。它可以将已有的入侵方法检查出来，但对新的入侵方法无能为力。其难点在于如何设计模式既能够表达“入侵”现象又不会将正常的活动包含进来。

（2）异常检测

假设入侵者活动异常于正常主体的活动。根据这一理念建立主体正常活动的“活动简档”，将当前主体的活动状况与“活动简档”相比较，当违反其统计规律时，认为该活动可能是“入侵”行为。异常检测的难点在于如何建立“活动简档”及如何设计统计算法，从而不把正常的操作作为“入侵”或忽略真正的“入侵”行为。

4. 防火墙

防火墙是网络安全的第一道门户，可以实现内部网（信任网络）和外部不可信任网络之间，或者是内部网不同网络安全区域之间的隔离与访问控制，保证网络系统及

网络服务的可用性。狭义的防火墙是指安装了防火墙的软件或路由器系统，而广义的防火墙还包括整个网络的安全策略和安全行为。

防火墙是一种综合性的技术，涉及计算机网络技术、密码技术、安全技术、软件技术、安全协议、网络标准化组织的安全规范，以及安全操作系统等方面。近年来随着网络安全需求的不断升温，防火墙技术也发展迅速，并不断有新的信息安全技术和软件技术应用于包过滤、代理服务器、虚拟专用网、状态监测、加密技术和身份认证等这样的开发。

根据防火墙组建结构的不同，可以分为屏蔽路由器、双穴主机、屏蔽主机防火墙、屏蔽子网防火墙 4 种基本结构，以及一些变体，下面详细说明它们的优缺点与应用场合。

(1) 包过滤型防火墙

包过滤型防火墙工作于网络层，它对进出内部网络的所有信息进行分析，并按照一定的安全策略对进出内部网络的信息进行限制。这种防火墙的优点是处理速度快、费用低、对用户透明。缺点是维护比较困难，只能阻止少部分 IP 欺骗，不支持有效的用户认证，日志功能有限，过滤规则增加会大大降低吞吐量，无法对信息提供全面控制。

(2) 双宿网关防火墙

双宿网关防火墙由一台至少装有两块网卡的堡垒主机作为防火墙，位于内外网络之间，分别与内外网络相离，实现物理上的隔离。它有两种服务方式：一种是由用户直接登录到双宿主机上；另一种是在双宿主机上运行代理服务器。其安全性比屏蔽路由器高。但入侵者一旦得到双宿主机的访问权，内部网络就会被入侵，因此需具有强大的身份认证系统，才可以阻挡来自外部的不可信网络的非法入侵。

(3) 屏蔽主机防火墙

屏蔽主机防火墙是由包过滤型防火墙和双宿网关防火墙组合形成的一种防火墙，它强迫所有的外部主机与一个堡垒主机相连接，而不让它们直接与内部主机相连接。这种防火墙的优点是实现了网络层安全（包过滤）和应用层安全（代理），因此安全等级比屏蔽路由器要高。其缺点是堡垒主机可能被绕过，堡垒主机与其他内部主机间没有任何保护网络安全的东西存在，一旦被攻破，内网就将暴露。

(4) 屏蔽子网防火墙

屏蔽子网防火墙用了两个屏蔽路由器和一个堡垒主机，也称为“单 DMZ 防火墙结构”。这种防火墙的优点在于定义了“非军事区（DMZ）”网络后，支持网络层和应用层安全功能，在黑客攻破第一道防火墙，进入 DMZ 区后，只能对 DMZ 区进行破坏，无法影响到内部网络，所以这也是目前最安全的防火墙系统。

5. 安全扫描

安全扫描是指对计算机系统及网络端口进行安全性检查，它通常需要借助一个被称为“扫描器”的软件。扫描器并不是一个直接攻击网络漏洞的程序，它仅仅能够帮助管理员发现目标机的某些内在弱点，一个好的扫描器能够对得到的数据进行分析，帮助管理员查找目标主机的漏洞。它能够自动查找主机或网络，找到运行的服务及其相关属性，并发现这些服务潜在的漏洞。

从上面的描述中，我们可以发现安全扫描技术是一个帮助管理员找到网络隐患的工具，并不能直接解决安全问题，而且对未被业界发现的隐患也无法完全找到。

6. 日志审计系统

日志文件是包含关于系统消息的文件，这些消息通常来自于操作系统内核、运行的服务，以及在系统上运行的应用程序。日志文件包括系统日志、安全日志、应用日志等。现在的 Windows 和 UNIX（包括 Linux）系统都提供了较完善的日志系统。

日志审计系统则通过一些特定的、预先定义的规则来发现日志中潜在的问题，它可以用来事后亡羊补牢，也可以用来对网络安全攻击进行取证。显然它是一种被动式、事后的防护或事中跟踪的手段，很难在事前发挥作用。

7. PKI

PKI (Public Key Infrastructure, 公共密钥基础设施) 是 CA 安全认证体系的基础，是一种新的网络安全技术和安全规范，为安全认证体系进行密钥管理提供了一个平台。它能够为所有网络应用透明地提供采用加密和数字签名等密码服务所必需的密钥和证书管理。PKI 由认证中心、证书库、密钥备份及恢复系统、证书作废处理系统和客户端证书处理系统组成。

PKI 可以实现 CA 和证书的管理；密钥的备份与恢复；证书、密钥对的自动更换；交叉认证；加密密钥和签名密钥的分隔；支持对数字签名的不可抵赖性；密钥历史的管理等功能。PKI 技术的应用可以在认证、机密性、完整性和抗抵赖性方面发挥重要的作用。

PKI 技术主要借助“数字签名”技术实现以上这些方面的功能，数字签名是维护网络信息安全的一种重要方法和手段，在身份认证、数据完整性、抗抵赖性方面都有重要应用，特别是在大型网络安全通信中的密钥分配、认证和电子商务、电子政务系统中有重要作用。而且，它通过密码技术对电子文档进行电子形式的签名，是实现认证的重要工具。数字签名是只有信息发送方才能够进行的签名，是其他任何人无法伪造的一段数字串，这段特殊的数字串同时也是对相应的文件和信息真实性的一个证明。

签名是确认文件的一种手段，一般书面手工签名的作用有两个：一个是因为自己的签名难以否认，从而确认了文件已签署这一事实；二是因为签名不易仿冒，从而确认了文件是真实的这一事实。采用数字签名也能够确认以下两点：一是信息是由签名者发送的；二是信息自签发到收为止，没做任何修改。

数字签名的特点是它代表了文件的特征。如果文件发生变化，数字签名的值也会发生变化，不同的文件会得到不同的数字签名。数字签名是通过 Hash 函数与公开密钥算法来实现的，其原理如下：

- 1) 发送者首先将原文用 Hash 函数生成 128 位的数字摘要。
- 2) 发送者用自己的私钥对摘要进行加密形成数字签名，并且把加密后的数字签名附加在要发送的原文后面。
- 3) 发送者将原文和数字签名同时传给对方。
- 4) 接收者把收到的信息用 Hash 函数生成新的摘要，同时用发送者的公开密钥对

信息摘要进行解密。

5) 将解密后的摘要与新的摘要对比, 两者一致则说明传送过程中信息没有被破坏或篡改。

如果第三方冒充发送方发送了一个文件, 而接收方在对数字签名进行解密时使用的是发送方的公开密钥, 因此只要第三方不知道发送方的私用密钥, 解密后的数字摘要与计算机计算的新摘要必然是不同的。这就提供了一个安全的确认发送方身份的方法。

数字签名有两种, 一种是对整体信息的签名, 它是指经过密码变换的被签名信息整体; 另一种是对压缩信息的签名, 它是附加在被签名信息后或某一特定位置上的一段签名图样。若按照明文和密文的对应关系划分, 每一种又可以分为两个子类, 一类是确定性数字签名, 即明文与密文一一对应, 它对一个特定信息的签名不变化, 如 RSA 签名; 另一类是随机化或概率化数字签名, 它对同一信息的签名是随机变化的, 取决于签名算法中的随机参数的取值。一个明文可能有多个合法数字签名。

一个签名体制一般包含两个组成部分: 签名算法和验证算法。签名算法(也称签名密钥)是秘密的, 只有签名人掌握。而验证算法是公开的, 便于他人进行验证。

另外, 如果要基于 PKI 实现数据的保密性, 可以用对方的公钥对“原文+数字签名”所构成的信息包进行加密, 这样就可以保证对方只能使用自己的私钥进行解密, 从而达到保密性要求。

显然, 保障网络安全性与网络服务效率永远是一对矛盾体, 在计算机应用日益普及的今天, 要想网络系统安全可靠, 势必会增加许多控制措施和安全设备, 从而会或多或少的影响使用效率和方便性。

8. VPN

虚拟专用网(VPN)是企业网在 Internet 等公共网络上的延伸, 它通过一个私有的通道在公共网络上创建一个安全的私有连接。因此, 从本质上说 VPN 是一个虚信道, 它可用来连接两个专用网, 通过可靠的加密技术方法保证其安全性, 并且是作为一个公共网络的一部分存在的。实现 VPN 的关键技术主要有隧道技术、加解密技术、密钥管理技术和身份认证技术。如图 5-4 所示是一个 VPN 构成的原理示意图。

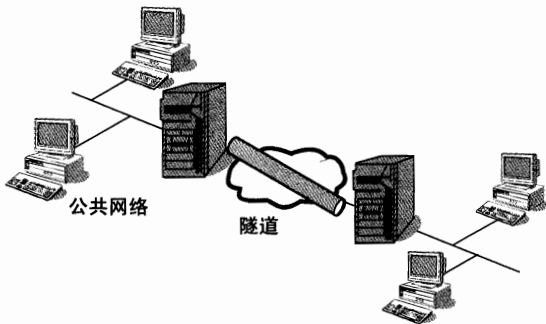


图 5-4 VPN 构成的原理示意图

隧道技术是 VPN 关键技术之一(此外还包括加解密技术、密钥管理技术和身份

认证技术），它是一种数据封装协议，也就是将一种协议封装在另一种协议中传输，从而实现被封装协议对封装协议的透明性。根据其工作的层次可以分为以下两类。

- 二层隧道技术：包括 PPP 基础上的 PPTP（点到点隧道协议）和 L2F（二层转发协议）、L2TP（二层隧道协议）。
- 三层隧道技术：主要代表是 IPSec（IP 层安全协议，它是 IPv4 和 IPv6 的安全标准）、移动 IP 和虚拟隧道协议（VTP）。

5.1.4 计算机病毒与木马

计算机病毒与木马是信息安全保障中不可避免的一个问题，本节将介绍常见病毒与木马的类型，以及命名规则，特点等内容。

1. 病毒与木马的概念

在计算机中，恶意代码主要分为两类，即病毒与木马。

- 病毒：编制或者在计算机程序中插入的破坏计算机功能或者破坏数据，影响计算机使用并且能够自我复制的一组计算机指令或者程序代码。
- 木马：计算机木马是一种后门程序，常被黑客用做控制远程计算机的工具。木马程序与一般的病毒不同，它不会自我繁殖，也并不“刻意”地去感染其他文件，它的主要作用是向施种木马者打开被种者计算机的门户，使对方可以任意毁坏、窃取文件，甚至远程操控计算机。木马与计算机网络中常常要用到的远程控制软件是有区别的。虽然二者在主要功能上都可以实现远程控制，但由于远程控制软件是“善意”的控制，因此通常不具有隐蔽性。木马则完全相反，木马要达到的正是“偷窃”性的远程控制，因此如果没有很强的隐蔽性的话，那么木马简直就是“毫无价值”的。

2. 病毒的分类

病毒的分类，有多种方式。

（1）按病毒存在的媒体分类

- 网络病毒：计算机网络传播感染网络中的可执行文件。
- 文件病毒：感染计算机中的文件。
- 引导型病毒：感染启动扇区（Boot）和硬盘的系统引导扇区。

（2）按病毒传染的方法分类

- 驻留型病毒：驻留内存，并一直处于激活状态。
- 非驻留型病毒：在得到机会激活时并不感染计算机内存。

（3）按病毒的危害分类

- 无危险型：减少磁盘的可用空间、减少内存、显示图像、发出声音，不影响系统。
- 危险型：造成严重的错误；删除程序、破坏数据、清除系统内存区和操作系统中重要的信息。

- 伴随型病毒：不改变文件本身，产生 EXE 文件的伴随体。
- “蠕虫”型病毒：通过计算机网络传播，占用系统内存。
- 寄生型病毒：练习型、诡秘型、变型病毒。

3. 病毒种类命名

每种病毒都有自己的名称，从名称通常可以判断出该病毒的类型。

(1) 系统病毒

系统病毒的前缀为 Win32、PE、Win95、W32、W95 等。这些病毒一般共有的特性是可以感染 Windows 操作系统的 *.exe 和 *.dll 文件，并通过这些文件进行传播。

(2) 蠕虫病毒

蠕虫病毒的前缀是 Worm。这种病毒的共有特性是通过网络或者系统漏洞进行传播，很大部分的蠕虫病毒都有向外发送带毒邮件，阻塞网络的特性。例如，冲击波（阻塞网络）、小邮差（发带毒邮件）等。

(3) 木马病毒、黑客病毒

木马病毒的前缀是 Trojan，黑客病毒前缀一般为 Hack。QQ 消息尾巴木马：Trojan.QQ3344，还有大家可能遇见比较多的针对网络游戏的木马病毒如 Trojan.LMir.PSW.60。

(4) 脚本病毒

脚本病毒的前缀是 Script。脚本病毒的共有特性是使用脚本语言编写，通过网页进行传播的病毒，如红色代码（Script.Redlof）。脚本病毒还会有如下前缀：VBS、JS（表明是何种脚本编写的），如欢乐时光（VBS.Happytime）、十四日（Js.Fortnight.c.s）等。

(5) 宏病毒

宏病毒也是脚本病毒的一种，由于它的特殊性，因此在这里单独算成一类。宏病毒的前缀是 Macro，第二前缀是 Word、Excel 其中之一。如 Macro.Word.WhiteScreen、美丽莎（Macro.Melissa）。

(6) 后门病毒

后门病毒的前缀是 Backdoor。该类病毒的共有特性是通过网络传播，给系统开后门，给用户计算机带来安全隐患。

(7) 病毒种植程序病毒

这类病毒的共有特性是运行时会从体内释放出一个或几个新的病毒到系统目录下，由释放出来的新病毒产生破坏。如冰河播种者（Dropper.BingHe2.2C）、MSN 射手（Dropper.Worm.Smibag）等。

(8) 破坏性程序病毒

破坏性程序病毒的前缀是 Harm。这类病毒的共有特性是本身具有好看的图标来诱惑用户点击，当用户点击这类病毒时，病毒便会直接对用户计算机产生破坏。如格式化 C 盘（Harm.formatC.f）、杀手命令（Harm.Command.Killer）等。

(9) 玩笑病毒

玩笑病毒的前缀是 Joke, 也称恶作剧病毒。这类病毒的共有特性是本身具有好看的图标来诱惑用户点击, 当用户点击这类病毒时, 病毒会做出各种破坏操作来吓唬用户, 其实病毒并没有对用户计算机进行任何破坏, 如女鬼 (Joke.Girl ghost) 病毒。

(10) 捆绑机病毒

捆绑机病毒的前缀是 Binder。这类病毒的共有特性是病毒作者会使用特定的捆绑程序将病毒与一些应用程序 (如 QQ、IE) 捆绑起来, 表面上看是一个正常的文件, 当用户运行这些捆绑病毒时, 会表面上运行这些应用程序, 然后隐藏运行捆绑在一起的病毒, 从而给用户造成危害, 如捆绑 QQ (Binder.QQPass.QQBin)、系统杀手 (Binder.killsys) 等。

关于病毒与木马, 考生除了解以上基本内容外, 还应该关注最近最为著名的病毒或木马的主要特征, 以前考试曾多次以当时最流行病毒为题。如 2007 年“熊猫烧香”病毒盛行, 所以在 2007 年 5 月的软考中就考到了“机器中熊猫烧香病毒的症状”。

5.2 典型试题分析

试题 1

用于在网络应用层和传输层之间提供加密方案的协议是__ (1) __。

- (1) A. PGP B. SSL C. IPSec D. DES

试题分析

本题考查安全协议的工作层次, 常见安全协议的工作层次请参看“5.1.3 网络安全”中“图 5-2 安全协议层次图”。

试题答案

- (1) B

试题 2

__ (2) __ 不属于 PKI 认证中心 (CA) 的功能。

- (2) A. 接收并验证最终用户数字证书的申请
B. 向申请者颁发或拒绝颁发数字证书
C. 产生和发布证书废止列表 (CRL), 验证证书状态
D. 业务受理点 LRA 的全面管理

试题分析

认证中心 (CA) 是电子商务体系中的核心环节, 是电子交易中信赖的基础。它通过自身的注册审核体系, 检查核实进行证书申请的用户身份和各项相关信息, 使网上交易的用户属性客观真实性与证书的真实性一致。认证中心作为权威的、可信赖的、

公正的第三方机构，专门负责发放并管理所有参与网上交易的实体所需的数字证书。

CA 的主要功能有证书发放、证书更新、证书撤销和证书验证。CA 的核心功能是发放和管理数字证书，具体描述如下：

- ① 接收验证最终用户数字证书的申请。
- ② 确定是否接受最终用户数字证书的申请。
- ③ 向申请者颁发或拒绝颁发数字证书。
- ④ 接收、处理最终用户的数字证书更新请求。
- ⑤ 接收最终用户数字证书的查询、撤销。
- ⑥ 产生和发布证书废止列表（CRL），验证证书状态。
- ⑦ 数字证书归档。
- ⑧ 密钥归档。
- ⑨ 历史数据归档。

通常，CA 中心会采用“统一建设，分级管理”的原则，分为多层结构进行建设和管理，即统一建立注册中心（RA）系统，各地级市及各行业可以根据具体情况设置不同层次的下级 RA 中心或本地注册中心（LRA）。各级下级 RA 机构统一接受 CA 中心的管理和审计，证书用户可通过 LRA 业务受理点完成证书业务办理。RA 系统负责本地管理员。用户的证书申请审核，并为 LRA 系统在各分支机构的分布建设提供策略支撑，完成 CA 中心的证书注册服务的集中处理。

试题答案

(2) D

试题 3

网络安全设计是保证网络安全运行的基础，网络安全设计有其基本的设计原则。以下关于网络安全设计原则的描述，错误的是 (3)。

- (3) A. 网络安全的“木桶原则”强调对信息均衡、全面地进行保护
- B. 良好的等级划分，是实现网络安全的保障
- C. 网络安全系统设计应独立进行，不需要考虑网络结构
- D. 网络安全系统应该以不影响系统正常运行为前提

试题分析

根据防范安全攻击的安全需求、需要达到的安全目标、对应安全机制所需的安全服务等因素，参照 SSE-CMM（系统安全工程能力成熟模型）和 ISO 17799（信息安全管理体系标准）等国际标准，综合考虑可实施性、可管理性、可扩展性、综合完备性、系统均衡性等方面，网络安全防范体系在整体设计过程中应遵循以下 9 项原则。

- ① 网络信息安全的木桶原则。网络信息安全的木桶原则是指对信息均衡、全面

地进行保护。“木桶的最大容积取决于最短的一块木板”。网络信息系统是一个复杂的计算机系统，它本身在物理上、操作上和管理上的种种漏洞构成了系统的安全脆弱性，尤其是多用户网络系统自身的复杂性、资源共享性使单纯的技术保护防不胜防。攻击者使用的“最易渗透原则”，必然在系统中最薄弱的地方进行攻击。因此，充分、全面、完整地系统的安全漏洞和安全威胁进行分析，评估和检测（包括模拟攻击）是设计信息安全系统的必要前提条件。安全机制和安全服务设计的首要目的是防止最常用的攻击手段，根本目的是提高整个系统的“安全最低点”的安全性能。

② 网络信息安全的整体性原则。要求在网络发生被攻击、破坏事件的情况下，必须尽可能地快速恢复网络信息中心的服务，减小损失。因此，信息安全系统应该包括安全防护机制、安全检测机制和安全恢复机制。安全防护机制是根据具体系统存在的各种安全威胁采取的相应的防护措施，避免非法攻击的进行。安全检测机制是检测系统的运行情况，及时发现和制止对系统进行的各种攻击。安全恢复机制是在安全防护机制失效的情况下，进行应急处理，并尽量及时地恢复信息，减少供给的破坏程度。

③ 安全性评价与平衡原则。对任何网络，绝对安全难以达到，也不一定是必要的，所以需要建立合理的实用安全性与用户需求评价与平衡体系。安全体系设计要正确处理需求、风险与代价的关系，做到安全性与可用性相容，做到组织上可执行。评价信息是否安全，没有绝对的评判标准和衡量指标，只能取决于系统的用户需求和具体的应用环境，具体取决于系统的规模和范围、系统的性质和信息的重要程度。

④ 标准化与一致性原则。系统是一个庞大的系统工程，其安全体系的设计必须遵循一系列的标准，这样才能确保各个分系统的一致性，使整个系统安全地互连互通，信息共享。

⑤ 技术与管理相结合原则。安全体系是一个复杂的系统工程，涉及人、技术、操作等要素，单靠技术或单靠管理都不可能实现。因此，必须将各种安全技术与管理机制、人员思想教育与技术培训、安全规章制度建设相结合。

⑥ 统筹规划，分步实施原则。由于政策规定、服务需求的不明朗，环境、条件、时间的变化，攻击手段的进步，安全防护不可能一步到位，可在一个比较全面的安全规划下，根据网络的实际需要，先建立基本的安全体系，保证基本的、必需的安全性。今后随着网络规模的扩大及应用的增加，网络应用和复杂程度的变化，网络脆弱性也会不断增加，调整或增强安全防护力度，保证整个网络最根本的安全需求。

⑦ 等级性原则。等级性原则是指安全层次和安全级别。良好的信息安全系统必然是分为不同等级的，包括对信息保密程度分级，对用户操作权限分级，对网络安全程度分级（安全子网和安全区域），对系统实现结构的分级（应用层、网络层、链路层等），从而针对不同级别的安全对象，提供全面、可选的安全算法和安全体制，以满足网络中不同层次的各种实际需求。

⑧ 动态发展原则。要根据网络安全的变化不断调整安全措施，适应新的网络环境，满足新的网络安全需求。

⑨ 易操作性原则。首先，安全措施需要人为去完成，如果措施过于复杂，对人的要求过高，本身就降低了安全性。其次，措施的采用不能影响系统的正常运行。

试题答案

(3) C

试题 4

关于网络安全，以下说法中正确的是__ (4) __。

- (4) A. 使用无线传输可以防御网络监听
- B. 木马是一种蠕虫病毒
- C. 使用防火墙可以有效地防御病毒
- D. 冲击波病毒利用 Windows 的 RPC 漏洞进行传播

试题分析

冲击波病毒是一种蠕虫病毒，病毒运行时会不停地利用 IP 扫描技术寻找网络上系统为 Windows 2000 或 Windows XP 的计算机，找到后就利用 RPC 缓冲区漏洞攻击该系统。一旦攻击成功，病毒体将会被传送到对方计算机中进行感染，使系统操作异常，不停重启，甚至导致系统崩溃。

木马是一种基于远程控制的黑客工具，分为控制端和服务端（被控制端），具有隐蔽性和非授权性的特点。所谓隐蔽性是指木马的设计者为了防止木马被发现，会采用多种手段隐藏木马，这样，服务端即使发现感染了木马，由于不能确定其具体位置，往往只能望“马”兴叹。所谓非授权性是指一旦控制端与服务端连接后，控制端将享有服务端的大部分操作权限，包括修改文件，修改注册表，控制鼠标、键盘等，而这些权力并不是服务端赋予的，而是通过木马程序窃取的。

使用无线传输，如果要防御网络监听，则需要使用专业设备。使用防火墙可以有效地防御网络攻击，但不能有效地防御病毒。

试题答案

(4) D

试题 5

许多黑客利用缓冲区溢出漏洞进行攻击，对于这一威胁，最可靠的解决方案是__ (5) __。

- (5) A. 安装防火墙
- B. 安装用户认证系统
- C. 安装相关的系统补丁
- D. 安装防病毒软件

试题分析

许多黑客利用缓冲区溢出漏洞进行攻击，对于这一威胁，最可靠的解决方案是及时安装相关的系统补丁，“堵住”漏洞。

试题答案

(5) C

试题 6

____(6)____无法有效防御 DDoS 攻击。

- (6) A. 根据 IP 地址对数据包进行过滤 B. 为系统访问提供更高级别的身份认证
C. 安装防病毒软件 D. 使用工具软件检测不正常的高流量

试题分析

分布式拒绝服务攻击 (Distributed Denial of Service, DDoS) 手段是在传统的 DoS 攻击基础之上产生的一类攻击方式。分布式拒绝服务攻击指借助于客户/服务器技术, 将多个计算机联合起来作为攻击平台, 对一个或多个目标发动 DoS 攻击, 从而成倍地提高拒绝服务攻击的威力。高速广泛连接的网络给大家带来了方便, 也为 DDoS 攻击创造了极为有利的条件。

对于此类隐蔽性极好的 DDoS 攻击的防范, 更重要的是用户要加强安全防范意识, 提高网络系统的安全性。专家建议可以采取的安全防御措施有以下几种。

①及早发现系统存在的攻击漏洞, 及时安装系统补丁程序。对一些重要的信息 (如系统配置信息) 建立和完善备份机制。对一些特权账号 (如管理员账号) 的密码设置要谨慎。通过这样一系列的举措, 可以把攻击者的可乘之机降低到最小。

②在网络管理方面, 要经常检查系统的物理环境, 禁止那些不必要的网络服务。建立边界安全界限, 确保输出的包受到正确限制。经常检测系统配置信息, 并注意查看每天的安全日志。

③利用网络安全设备 (如防火墙) 来加固网络的安全性, 配置好这些设备的安全规则, 过滤掉所有可能的伪造数据包。

④与网络服务提供商协调工作, 让网络服务提供商帮助实现路由的访问控制和对带宽总量的限制。

⑤当用户发现自己正在遭受 DDoS 攻击时, 应当启动自己的应付策略, 尽可能快地追踪攻击包, 并且及时联系 ISP 和有关应急组织, 分析受影响的系统, 确定涉及的其他节点, 从而阻挡从已知攻击节点的流量。

⑥如果用户是潜在的 DDoS 攻击受害者, 并且用户发现自己的计算机被攻击者用作主控端和代理端时, 用户不能因为自己的系统暂时没有受到损害而掉以轻心。攻击者一旦发现用户系统的漏洞, 这对用户的系统将是一个很大的威胁。所以, 用户只要发现系统中存在 DDoS 攻击的工具软件, 就要及时把它清除, 以免留下后患。

试题答案

(6) C

试题 7

IPSec VPN 安全技术没有用到____(7)____。

- (7) A. 隧道技术 B. 加密技术
C. 入侵检测技术 D. 身份认证技术

试题分析

虚拟专用网络（Virtual Private Network, VPN）提供了一种通过公用网络安全地对企业内部专用网络进行远程访问的连接方式。与普通网络连接一样，VPN 也由客户机、传输介质和服务器 3 部分组成，不同的是 VPN 连接使用隧道作为传输通道，这个隧道是建立在公共网络或专用网络基础之上的，如 Internet 或 Intranet。

VPN 可以实现不同网络的组件和资源之间的相互连接，利用 Internet 或其他公共互联网络的基础设施为用户创建隧道，并提供与专用网络一样的安全和功能保障。VPN 允许远程通信方、销售人员或企业分支机构使用 Internet 等公共互联网络的路由基础设施以安全的方式与位于企业局域网端的企业服务器建立连接。VPN 对用户端透明，用户好像使用一条专用线路在客户计算机和企业服务器之间建立点对点连接，进行数据的传输。

实现 VPN 的关键技术如下。

① 安全隧道技术（Tunneling）：一种通过使用互联网络的基础设施在网络之间传递数据的方式。使用隧道传递的数据（或负载）可以是不同协议的数据帧或包。隧道协议将这些其他协议的数据帧或包重新封装在新的包头中发送。新的包头提供了路由信息，从而使封装的负载数据能够通过互联网络传递。被封装的数据包在隧道的两个端点之间通过公共互联网络进行路由。被封装的数据包在公共互联网络上传递时所经过的逻辑路径称为隧道。一旦到达网络终点，数据将被解包并转发到最终目的地。隧道技术是指包括数据封装、传输和解包在内的全过程。

② 加/解密技术：VPN 利用已有的加/解密技术实现保密通信。

③ 密钥管理技术：建立隧道和保密通信都需要密钥管理技术的支撑，密钥管理负责密钥的生成、分发、控制和跟踪，以及验证密钥的真实性。

④ 身份认证技术：假如 VPN 的用户都要通过身份认证，通常使用用户名和密码，或者智能卡实现。

⑤ 访问控制技术：由 VPN 服务的提供者根据在各种预定义的组中的用户身份标识，来限制用户对网络信息或资源的访问控制的机制。

试题答案

(7) C

试题 8

某 Web 网站向 CA 申请了数字证书。用户登录该网站时，通过验证__（8）__，来确认该数字证书的有效性，从而__（9）__。

(8) A. CA 的签名 B. 网站的签名 C. 会话密钥 D. DES 密码

(9) A. 向网站确认自己的身份 B. 获取访问网站的权限

C. 和网站进行双向认证 D. 验证该网站的真伪

试题分析

数字证书采用公钥体制，即利用一对互相匹配的密钥进行加密和解密。每个用户将设定两个私钥（仅为本人所知的专用密钥，用来解密和签名）和公钥（由本人公开，

用于加密和验证签名)两个密钥,用以实现:

- ①发送机密文件:发送方使用接收方的公钥进行加密,接收方使用自己的私钥解密。
- ②接收方能够通过数字证书来确认发送方的身份,发送方无法抵赖。
- ③信息自数字签名后可以保证信息无法更改。

Web 网站向 CA 申请数字证书。用户登录该网站时,通过验证 CA 的签名,来确认该数字证书的有效性,从而验证该网站的真伪。

试题答案

(8) A (9) D

试题 9

某企业开发应用程序,要求调用传输层的安全协议保障应用通信的安全,下面可选的传输层安全协议是__(10)___。

(10) A. IPSec B. L2TP C. TLS D. PPTP

试题分析

本题考查安全协议的工作层次。

IPSec 工作于网络层, L2TP 与 PPTP 工作于数据链路层, TLS 工作于传输层,所以本题选 C。

试题答案

(10) C

试题 10

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

试题分析

本题考查网络安全协议的功能, SET 协议具备抗抵赖功能。关于 SET 协议的详细情况请参看“5.1.3 网络安全”。

试题答案

(11) A

试题 11

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

试题分析

IPSec 是虚拟专用网 (VPN) 的一个安全协议,它提供所有在网络层上的数据保护,提供透明的安全通信,而不属于传输层安全技术。

试题答案

(12) C

试题 12

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 (13) 区域。

- (13) A. 信任网络
C. 半信任网络

- B. 非信任网络
D. DMZ (非军事化区)

试题分析

DMZ (DeMilitarized Zone, 隔离区) 也称为非军事化区,它是为了解决安装防火墙后,外部网络不能访问内部网络服务器的问题,而设立的一个非安全系统与安全系统之间的缓冲区,这个缓冲区位于企业内部网络和外部网络之间的小网络区域内,在这个小网络区域内可以放置一些必须公开的服务器设施,例如,企业的 Web 服务器、FTP 服务器和论坛等。另一方面,通过 DMZ 区域,更加有效地保护了内部网络,因为这种网络部署与一般的防火墙方案相比,对攻击者来说又多了一道关卡。

试题答案

(13) D

试题 13

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 (14) 的内容。

- (14) A. 物理线路安全与网络安全
C. 物理线路安全与系统安全

- B. 网络安全与系统安全
D. 系统安全与应用安全

试题分析

网络安全体系设计是逻辑设计工作的重要内容之一,数据库容灾属于系统安全和应用安全考虑范畴。

试题答案

(14) D

试题 14

公司总部与分部之间需要传输大量数据,在保障数据安全的同时又要兼顾密钥算法效率,最合适的加密算法是 (15)。

- (15) A. RC-5

- B. RSA

- C. ECC

- D. MD5

试题分析

公司总部与分部之间通过 Internet 传输数据,需要采用加密方式保障数据安全。

- D. 保证可信网络内部信息不外泄
- 实现 VPN 的关键技术主要有隧道技术、加/解密技术、（9）和身份认证技术。如果需要在传输层实现 VPN，可选的协议是（10）。
- (9) A. 入侵检测技术 B. 病毒防治技术
C. 安全审计技术 D. 密钥管理技术
- (10) A. L2TP B. PPTP C. TLS D. IPSec
- 目前在网络上流行的“熊猫烧香”病毒属于（11）类型的病毒，感染该病毒后的计算机不会出现（12）的情况。
- (11) A. 目录 B. 引导区 C. 蠕虫 D. DOS
- (12) A. 执行文件图标变成熊猫烧香 B. 用户信息被泄露
C. 系统运行变慢 D. 破坏计算机主板
- 所谓网络安全漏洞是指（13）。
- (13) A. 用户的误操作引起的系统故障
B. 网络结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
- ARP 攻击造成网络无法跨网段通信的原因是（14）。
- (14) A. 发送大量 ARP 报文造成网络拥塞
B. 伪造网关 ARP 报文使得数据包无法发送到网关
C. ARP 攻击破坏了网络的物理连通性
D. ARP 攻击破坏了网关设备
- 下列安全协议中，用来实现安全电子邮件的协议是（15）。
- (15) A. IPSec B. L2TP C. PGP D. PPTP

5.4 练习题解析

试题 1 分析

常见的对称加密算法包括：DES、3DES、RC-5、IDEA。

常见的非对称加密算法包括：RSA、ECC。

关于对称加密算法与非对称加密算法的详细情况请参看“5.1.2 安全基础技术”。

试题 1 答案

- (1) D

试题 2 分析

数字签名技术是将摘要信息用发送者的私钥加密与原文一起传送给接收者。接收者只有用发送的公钥才能解密被加密的摘要信息，然后用 Hash 函数对收到的原文产生一个摘要信息，与解密的摘要信息对比。如果相同，则说明收到的信息是完整的，在传输过程中没有被修改，否则说明信息被修改过，因此数字签名能够验证信息的完整性。数字签名是个加密的过程，数字签名验证是个解密的过程。

数字签名技术是在网络系统虚拟环境中确认身份的重要技术，完全可以代替现实过程中的“亲笔签字”，在技术和法律上有保证。数字签名主要的功能是：保证信息传输的完整性、发送者的身份认证、防止交易中的抵赖发生。

试题 2 答案

(2) C

试题 3 分析

Kerberos 认证是一种使用对称密钥加密算法来实现通过可信第三方密钥分发中心的身份认证系统。客户方需要向服务器方递交自己的凭据来证明自己的身份，该凭据是由 KDC 专门为客户和服务器方在某一阶段内通信而生成的。凭据中包括客户和服务器方的身份信息和在下一阶段双方使用的临时加密密钥，还有证明客户方拥有会话密钥的身份认证者信息。身份认证信息的作用是防止攻击者在将来将同样的凭据再次使用。时间标记是检测重放攻击。

试题 3 答案

(3) B

试题 4 分析

包过滤型防火墙工作在 OSI 网络参考模型的网络层和传输层，它根据数据包头源地址、目的地址、端口号和协议类型等标志确定是否允许通过。只有满足过滤条件的数据包才被转发到相应的目的地，其余数据包则被从数据流中丢弃。

包过滤方式是一种通用、廉价和有效的安全手段。之所以通用，是因为它不是针对各个具体的网络服务采取特殊的处理方式，适用于所有网络服务；之所以廉价，是因为大多数路由器都提供数据包过滤功能，所以这类防火墙多数是由路由器集成的；之所以有效，是因为它能很大程度上满足了绝大多数企业安全要求。

在整个防火墙技术的发展过程中，包过滤技术出现了两种不同版本，称为“第一代静态包过滤”和“第二代动态包过滤”。

① 第一代静态包过滤类型防火墙。这类防火墙几乎是与路由器同时产生的，它是根据定义好的过滤规则审查每个数据包，以便确定其是否与某一条包过滤规则匹配。过滤规则基于数据包的报头信息进行制订。报头信息中包括 IP 源地址、IP 目标地址、传输协议（TCP、UDP、ICMP 等）、TCP/UDP 目标端口、ICMP 消息类型等。

② 第二代动态包过滤类型防火墙。这类防火墙采用动态设置包过滤规则的方法，避免了静态包过滤所具有的问题。这种技术后来发展成为包状态监测（Stateful

Inspection) 技术。采用这种技术的防火墙对通过其建立的每一个连接都进行跟踪, 并且根据需要可动态地在过滤规则中增加或更新条目。

包过滤方式的优点是不用改动客户机和主机上的应用程序, 因为它工作在网络层和传输层, 与应用层无关。但其弱点也是明显的: 过滤判别的依据只是网络层和传输层的有限信息, 因而各种安全要求不可能充分满足; 在许多过滤器中, 过滤规则的数目是有限制的, 且随着规则数目的增加, 性能会受到很大影响; 由于缺少上下文关联信息, 不能有效地过滤如 UDP、RPC 一类的协议; 另外, 大多数过滤器中缺少审计和报警机制, 它只能依据包头信息, 而不能对用户身份进行验证, 很容易受到“地址欺骗型”攻击。对安全管理人员素质要求高, 建立安全规则时, 必须对协议本身及其在不同应用程序中的作用有较深入的理解。因此, 过滤器通常是和应用网关配合使用, 共同组成防火墙系统。

试题4 答案

(4) D

试题5~6 分析

在 ISO 7498-2 中描述了开放系统互联安全的体系结构, 提出设计安全的信息系统的基础架构中应该包含 5 种安全服务(安全功能)、能够对这 5 种安全服务提供支持的 8 类安全机制和普遍安全机制, 以及需要进行的 5 种 OSI 安全管理方式。其结构如图 5-5 所示。

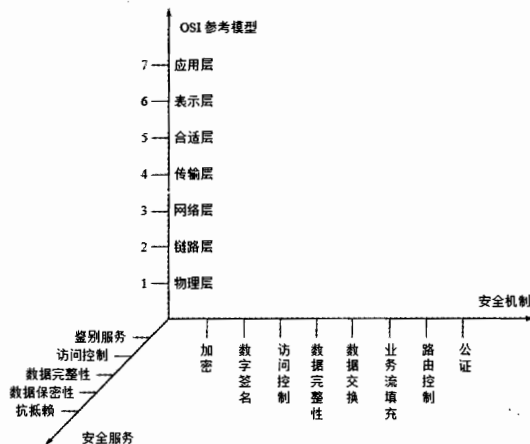


图 5-5 ISO 7498-2 结构

TCSEC 将安全分为 4 个方面, 分别是安全政策、可说明性、安全保障和文档。在美国国防部彩虹系列(Rainbow Series)标准中有详细的描述。该标准将以上 4 个方面分为 7 个安全级别, 从低到高依次为 D、C1、C2、B1、B2、B3 和 A 级。

A 类: 验证设计。形式化证明的安全。用于绝密级。

B 类:

B3: 安全域保护。提供可信设备的管理和恢复。即使计算机系统崩溃, 也不会泄密。

B2: 结构化安全策略模型。满足 B1 类要求外, 提供强制性控制。

B1: 安全标记保护。满足 C 类要求, 并提供数据标记。

C 类:

C2: 受控的访问保护。在 C1 之上, 增加了访问保护和审计跟踪功能。

C1: 自主安全保护。无条件的访问控制, 具有识别和控制的责任。

D 类: 最低保护。例如, 未加任何实际的安全措施。

试题 5~6 答案

(5) B

(6) D

试题 7 分析

无论采用什么形式的网络隔离, 其实质都是数据或信息的隔离。网络隔离的重点是物理隔离。人工方式隔离的一个特征, 就是内网与外网永不连接, 内网和外网在同一时间最多只有一个同隔离设备建立非 TCP/IP 的数据连接。

试题 7 答案

(7) D

试题 8 分析

安全审计包括识别、记录、存储、分析与安全相关行为的信息, 审计记录用于检查与安全相关的活动和负责人。安全审计系统就是根据一定的安全策略记录和分析历史操作事件及数据, 发现能够改进系统运行性能和系统安全的地方。安全审计的作用包括: 对潜在的攻击者起到震慑或警告的作用、检测和制止对安全系统的入侵、发现计算机的滥用情况、为系统管理员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及对已经发生的系统攻击行为提供有效的追究证据。安全审计系统通常有一个统一的集中管理平台, 支持集中管理, 并支持对日志代理、安全审计中心、日志、数据库的集中管理, 并具有事件响应机制和联动机制。

试题 8 答案

(8) D

试题 9~10 分析

隧道技术可以分别以第 2、3 层隧道协议为基础。第 2 层隧道协议对应 OSI 模型中的数据链路层, 使用帧作为数据交换单位。PPTP, L2TP 和 L2F (第 2 层转发) 都属于第 2 层隧道协议, 都是将数据封装在 PPP 帧中通过互联网络发送。第 3 层隧道协议对应 OSI 模型中的网络层, 使用包作为数据交换单位。IPoverIP 及 IPSec 隧道模式都属于第 3 层隧道协议, 都是将 IP 包封装在附加的 IP 包头中通过 IP 网络传送。

TLS (Transport Layer Security, 传输层安全性) 协议是 IETF 标准草案, 它基于 SSL 并与之相似。它的主要目标是在两个正在通信的应用程序之间提供保密性和数据完整性。与 SSL 一样, TLS 是独立于应用程序协议的, 其使用的加密算法的种类与

SSL 使用的相似。然而, TLS 标准把如何启动 TLS 握手和如何解释认证证书的决定权留给运行于其上层的协议的设计者和实现者来判断。TLS 协议的目标,按其优先级顺序来说,是密码安全性、互操作性和可扩展性。

试题 9~10 答案

(9) D (10) C

试题 11~12 分析

熊猫烧香是一种感染型的蠕虫病毒,它能感染系统中 exe、com、pif、src、html、asp 等文件,还能中止大量的反病毒软件进程并且会删除扩展名为 gho 的文件,该文件是系统备份工具 GHOST 的备份文件,使用户的系统备份文件丢失。

试题 11~12 答案

(11) C (12) D

试题 13 分析

本题主要考查网络安全漏洞的基本概念,网络安全漏洞通常是指网络结点的系统软件或应用软件在逻辑上的缺陷,因此本题应该选择 B。

试题 13 答案

(13) B

试题 14 分析

本题主要考查 ARP 攻击的定义和特点。ARP 攻击是针对以太网地址解析协议(ARP)的一种攻击技术,这种攻击可让攻击者取得局域网上的数据封包甚至可篡改封包,且可让网络上特定计算机或所有计算机无法正常连接。ARP 攻击造成网络无法跨网段通信的原因是伪造网关 ARP 报文使得数据包无法发送到网关。

试题 14 答案

(14) B

试题 15 分析

PGP (Pretty Good Privacy), 是一个基于 RSA 公钥加密体系的邮件加密软件。可以用它对邮件保密以防止非授权者阅读,它还能对邮件加上数字签名从而使收信人可以确认邮件的发送者,并能确信邮件不被篡改。它可以提供一种安全的通信方式,而事先并不需要任何保密的渠道用来传递密匙。它采用了一种 RSA 和传统加密的杂合算法,用于数字签名的邮件文摘算法、加密前压缩等,还有一个良好的人机工程设计。它的功能强大,有很快的速度。

试题 15 答案

(15) C

6

第6章

信息化基础

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

(1) 信息系统工程总体规划

- 总体规划目标、范围。
- 总体规划的方法论。
- 信息系统的组成。
- 信息系统的实现。

(2) 政府信息化与电子政务

- 电子政务的概念、内容和技术形式。
- 中国政府信息化的策略和历程。
- 电子政务建设的过程模式和技术模式。

(3) 企业信息化与电子商务

- 企业信息化的概念、目的、规划、方法。
- ERP 的主要模块和主要算法。
- 企业业务流程重组 (BPR)。
- CRM、PDM 在企业的应用。
- 知识管理。
- 企业应用集成。
- 全程供应链管理的思想。
- 商业智能。
- 电子商务的类型、标准。

(4) 信息资源管理

(5) 国际和国内有关信息化的标准、法律和规定

6.1 考点突破

从历年的考试情况来看,本章的考点主要集中在以下几个方面:企业应用集成的方式、商业智能、企业门户及信息化相关的一些概念。

6.1.1 历年考试情况分析

在历年的考试试题中,有关信息化基础知识的试题如表 6-1 所示。

表 6-1 信息化基础知识试题分布表

题号	2009.11	2010.11	2011.11
18	商业智能	企业门户	数据集成
19	企业应用集成	CRM	
20	EDI	企业应用集成	企业应用集成
21			
37			企业信息化建设
38			企业信息化方法

按照知识点进行总结和归类的试题分布情况如表 6-2 所示。

表 6-2 信息化基础知识归类表

知识点	2009.11	2010.11	2011.11
信息与信息化基本概念	0	0	0
政府信息化与电子政务	0	0	0
企业信息化与电子商务	1	1	2
应用集成技术	2	2	4
企业门户	0	1	0
合计	3	4	6

从表 6-2 中可以看出,信息化基础知识方面的内容在历年的考试中分值呈上升趋势。所占分数比例的趋势如图 6-1 所示。

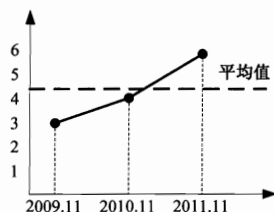


图 6-1 信息化基础知识历年试题比例趋势图

6.1.2 信息与信息化基本概念

从事计算机相关工作的人员,通常与信息及信息化是脱不开干系的,但很少有人能讲清楚信息与信息化到底是什么。不同的人对此有着不同的理解与定义。关于信息,常见的说法有如下几种。

- 维纳（Norbert Wiener）：信息就是信息，既不是物质也不是能量。
- 香农（Claude E.Shannon）：信息就是不确定性的减少。
- 哲学界：信息是事物普遍联系的方式。
- 其他：信息是事先不知道的报道。

而对于信息化，通常有以下说法：

- 信息化就是计算机、通信和网络技术的现代化。
- 信息化就是从物质生产占主导地位的社会向信息产业占主导地位社会转变的发展过程。
- 信息化就是从工业社会向信息社会演进的过程。

了解这两个概念，是对信息化进一步深入研究的基础。

6.1.3 政府信息化与电子政务

电子政务是政府机构应用现代信息和通信技术，将管理和服务通过网络技术进行集成，在网上实现政府组织结构和工作流程的优化重组，超越时间、空间与部门分隔的限制，全方位地向社会提供优质、规范、透明、符合国际水准的管理和服务。电子政务作为信息技术与管理的有机结合，成为当代信息化最重要的领域之一。

电子政务与传统政务相比，在办公手段、业务流程及与公众沟通的方式上都存在很大的区别。电子政务并不是要完全取代传统政务，也不是简单地将传统政务原封不动地搬到 Internet 上，而是要求政府部门运用网络和现代通信技术，对具体业务程序、工作方法、办公环境、组织和人员管理等进行优化和重组，打破传统政府的组织界限，使得政府部门之间、政府与社会公众之间可以通过各种电子化渠道相互沟通，并依据公众的需要，提供形式多样、方便快捷的服务方式。

随着信息技术的发展，信息在政治、经济和军事方面的作用日益显现。信息化已成为世界普遍关注的一个焦点，政府信息化作为国家信息化的基础，成为整个信息化中的关键，直接影响国家的竞争力和社会经济发展进程。政府信息化以电子政务这样一个更高层次的新面目出现，被认为是国家信息化最主要的推动力量。

1. 电子政务的业务模型

根据政府机构的业务形态来看，通常，电子政务主要包括 3 个应用领域，其业务模型可以用图 6-2 表示。

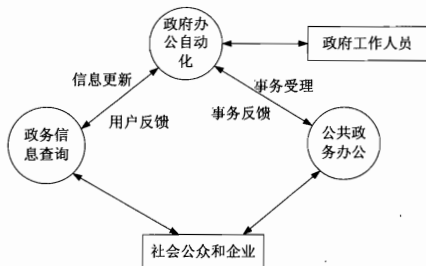


图 6-2 电子政务业务模型

(1) 政务信息查询

面向社会公众和企业组织，为其提供政策、法规、条例和流程的查询服务。

(2) 公共政务办公

借助互联网实现政府机构的对外办公，如申请、申报等，提高政府的运作效率，增加透明度。

(3) 政府办公自动化

以信息化手段提高政府机构内部办公的效率，如公文报送、信息通知和信息查询等。

在图 6-2 中，社会公众和企业主要通过政务信息查询及公共政务办公与电子政务平台建立沟通，相关事务处理请求通过 OA 系统中转给政府工作人员，政府工作人员可以通过 OA 系统进行政务处理及对政务信息查询系统的更新。通过对这一典型业务模型的分析，可以看出在电子政务系统中主要存在 3 种信息流，分别是政务办公信息流（存在于政府机构内部办公的过程中）、公共事务信息流（存在于政府机构对外办公的过程中）和政务咨询信息流（存在于社会公众和企业查询相关信息的过程中）。

2. 电子政务的应用模式

电子政务根据其服务的对象不同，可以分为以下 4 种模式：

(1) 政府对政府（Government to Government, G2G）

G2G 是指政府上下级之间、不同地区和不同职能部门之间实现的电子政务活动，包括国家和地方基础信息的采集、处理和利用，例如，人口信息、地理信息、资源信息等；政府之间各种业务流程所需要采集和处理的信息，例如，计划管理、经济管理、社会经济统计、公安、国防、国家安全等；政府之间的通信系统，包括各种紧急情况的通报、处理和通信系统。

(2) 政府对企业（Government to Business, G2B）

G2B 是政府向企业提供的各种公共服务，主要包括政府向企事业单位发布的各种方针、政策、法规和行政规定，即企事业单位从事合法业务活动的环境，包括产业政策、进出口、注册、纳税、工资、劳保、社保等各种规定；政府向企事业单位颁发的各种营业执照、许可证、合格证、质量认证等。

(3) 政府对公众（Government to Citizen, G2C）

G2C 实际上是政府面向公众所提供的服务。政府对公众的服务首先是信息服务，如让公众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等，还包括户口、各种证件的管理等政府提供的各种服务。

(4) 政府对公务员（Government to Employee, G2E）

G2E 是指政府与政府公务员即政府雇员之间的电子政务，也有学者把它称为内部效率效能电子政务模式。G2E 是政府机构通过网络技术实现内部电子化管理（如 OA 系统等）的重要形式，也是 G2G、G2B 和 G2C 的基础。G2E 主要是利用 Intranet 建立起有效的行政办公和员工管理体系，为提高政府工作效率和公务员管理水平服务。

6.1.4 企业信息化与电子商务

企业信息化就是企业应用信息技术及产品的过程，或者更确切地说，企业信息化是信息技术由局部到全局，由战术层次到战略层次向企业全面渗透，运用于流程管理、支持企业经营管理的过程。企业在信息化的过程中，会根据实际情况建设各种应用系统，从最初的办公自动化（Office Automation, OA）系统到目前流行的产品生命周期管理等，应用层次不一。随着信息技术的发展和信息化建设的推进，企业信息系统的规模在不断扩大，复杂程度在逐渐提高。

企业信息化是本章的考查重点，下面将详细介绍 ERP、CRM、企业应用集成、供应链管理、商务智能、电子商务等内容。

1. ERP

企业资源规划（Enterprise Resource Planning, ERP）是指建立在信息技术基础上，以系统化的管理思想，为企业提供决策和运营手段的管理平台。ERP 系统是将企业所有资源进行集成整合，并进行全面、一体化管理的信息系统。

目前，在 ERP 的应用方面，众多企业在汇集了不同行业、不同企业的管理需求特点、管理模式和管理经验之后，不断完善和发展自己的 ERP 系统应用产品，形成了百花齐放，百家争鸣的市场格局。

ERP 为企业提供的功能是多层面的和全方位的。在企业中，一般管理主要包括 3 个方面的内容，分别是生产控制、物流管理和财务管理。这三大系统本身就是一个集成体，相互之间有相应的接口，能够很好地整合在一起，这种整合使得原来只具备事后监控的信息系统，具有事前安排与实时控制的功能。另外，随着企业对人力资源管理的重视和加强，已经有越来越多的 ERP 供应商将人力资源管理也纳入了 ERP 系统。因此，典型的 ERP 系统的主要功能模块如下。

- 财会管理：包括会计核算和财务管理等模块。
- 物流管理：包括分销管理、库存控制和采购管理等模块。
- 生产控制管理：包括主生产计划、物料需求计划、能力需求计划、车间控制和制造标准等模块。
- 人力资源管理：包括人力资源规划的辅助决策、招聘管理、工资核算、工时管理和差旅费核算等模块。

ERP 通常有两种开发模式：

第一种模式是在 ERP 供应商的套装软件上进行二次开发。由于现在的 ERP 产品基本上采用模块化结构，允许用户进行个性化设计，所以，二次开发是可行的。这种方式实施时投资相对较少，而且项目的建设期明显缩短。套装软件往往由知名软件厂商开发，凭借技术实力雄厚的开发团队，套装软件具有良好的系统架构和稳定的系统性能，能够适应一定领域的市场需求，但面对的是管理水平参差不齐、竞争环境千差万别的各种企业。因此，套装软件在系统设计时往往采用行业的先进管理理念，这种理念不一定和企业原来的业务实践相一致。这就要求企业向这种标准靠拢，而实际上很难满足不同企业的个性化需求。

第二种模式是为企业定制开发 ERP 系统。这种开发方式是从企业的个性化需求出发,进行系统定制。这种定制开发的系统能够满足特定企业的需求,但鉴于开发者的技术实力和对企业业务实践、需求的了解程度,总是很难全面考虑系统的扩展性、稳定性等架构因素,系统不能快速适应企业的需求变化,开发周期较长,效率不高,投资较大,实施风险大。当然,随着信息技术的发展,这些不利因素也会逐渐得到缓解。例如,可以利用 Web Service 技术,集成企业原来在信息化建设中构建的各种彼此孤立的应用系统,降低开发成本和风险。

2. CRM

客户关系管理(Customer Relationship Management, CRM)将客户看做是企业的一项重要资产,客户关怀是 CRM 的中心,其目的是与客户建立长期和有效的业务关系,在与客户的每一个“接触点”上都更加接近客户、了解客户,最大限度地增加利润。CRM 的核心是客户价值管理,它将客户价值分为既成价值、潜在价值和模型价值,通过“一对一”营销原则,满足不同价值客户的个性化需求,提高客户忠诚度和保有率,实现客户价值持续贡献,从而全面提升企业赢利能力。

CRM 的功能可以归纳为 3 个方面:对销售、营销和客户服务 3 部分业务流程的信息化;与客户进行沟通所需要的手段(例如,电话、传真、网络等)的集成和自动化处理;对上面两部分功能所积累下的信息进行加工处理,产生客户智能,为企业决策提供支持。业界一致认为,市场营销和客户服务是 CRM 的支柱性功能。这些是客户与企业联系的主要领域,无论这些联系发生在售前、售中还是售后。

(1) 客户服务

客户服务是 CRM 的关键内容,是能否形成并保留大量忠诚客户的关键。随着市场竞争的深入,客户对服务的期望值也在不断地提高,已经超出传统的电话呼叫中心的范围。而呼叫中心正在向可以处理各种通信媒介的客户服务中心演变。电话互动必须与电子邮件、传真、网站,以及其他任何客户喜欢使用的方式相互整合。随着越来越多的客户进入互联网,通过浏览器来查看他们的订单或提出询问,自助服务的要求发展得也越来越快。

(2) 市场营销

营销自动化包括商机产生、商机获取和管理、商业活动管理和电话营销等。销售人员与潜在客户的互动行为、将潜在客户发展为真正客户并保持其忠诚度是使企业赢利的核心因素。随着互联网的发展,市场营销迅速从传统的电话营销转向网站营销和电子邮件营销。这些基于 Web 的营销活动给潜在客户更好的体验,使潜在客户以自己的方式、在方便的时间查看他们需要的信息。

(3) 共享的客户资料库

共享的客户资料库是企业的一种重要信息资源,它把市场营销和客户服务连接起来。如果一个企业的信息来源相互独立,那么这些信息中必然会存在大量重复、互相冲突的成分。这对企业的整体运营效率将产生负面影响;而动态的、能够被不同部门共享的客户资料库则是企业的一种宝贵资源,同时,它也是 CRM 的基础和依托。

(4) 分析能力

CRM 的一个重要方面在于它具有使客户价值最大化的分析能力。如今的 CRM 解决方案在提供标准报告的同时, 又可提供既定量又定性的即时分析。深入的智能分析需要统一的客户数据作为切入点, 并使所有企业业务应用系统融入分析环境中, 通过对客户数据的全面分析, 评估客户带给企业的价值, 以及衡量客户的满意度, 再将分析结果反馈给管理层, 这样, 便增加了信息分析的价值。企业决策者会参考这些信息, 做出更全面、更及时的商业决策。

3. 供应链管理

供应链是由供应商、制造商、仓库、配送中心和渠道商等构成的物流网络。同一个企业可能构成这个网络的不同结点, 但更多的情况下是由不同的企业构成这个网络中的不同结点。例如, 在某条供应链中, 某个企业可能既在制造商和仓库节点, 又在配送中心节点等占有位置。另外, 单个企业内部也同样存在一条供应链, 只不过处在各个节点上的不是其他企业, 而是该企业的各个部门。

供应链管理 (Supply Chain Management, SCM) 是一种集成的管理思想和方法, 它执行供应链中从供应商到最终用户的物流的计划和控制等职能。从单一的企业角度来看, 是指企业通过改善上、下游供应链关系, 整合和优化供应链中的信息流、物流和资金流, 以获得企业的竞争优势。

供应链管理是企业的有效性管理, 表现了企业在战略和战术上对业务流程的优化。整合并优化了供应商、制造商、零售商的业务效率, 使商品以正确的数量、正确的品质、在正确的地点、以正确的时间、最佳的成本进行生产和销售。SCM 包括计划、采购、制造、配送、退货五大基本内容。

(1) 计划

这是 SCM 的策略性部分。企业需要有一个策略来管理所有的资源, 以满足客户对产品的需求。好的计划是建立一系列的方法监控供应链, 使它能够有效、低成本地为顾客递送高质量和高价值的产品或服务。

(2) 采购

选择能为企业提供产品和服务的供应商, 与供应商建立一套定价、配送和付款流程, 并监控和改善管理。

(3) 制造

安排生产、测试、打包和准备送货所需的活动, 是供应链中测量内容最多的部分, 包括质量水平、产品产量和工人的生产效率等的测量。

(4) 配送

配送也称为物流, 是调整用户的订单收据、建立仓库网络、派递送人员提货并送货为顾客手中、建立产品计价系统、接收付款。

(5) 退货

退货是供应链中的问题处理部分。建立网络接收客户退回的次品和多余产品, 并在客户应用产品出问题提供支持。

总之, SCM 是从源头供应商到最终消费者的集成业务流程。它不仅为消费者带来有价值的产品和服务, 还为顾客带来有用的信息。

4. 商业智能

商业智能 (Business Intelligence, BI) 又称商务智能, 该概念于 1996 年最早由加特纳集团 (Gartner Group) 提出, 加特纳集团将商业智能定义为: 商业智能描述了一系列的概念和方法, 通过应用基于事实的支持系统来辅助商业决策的制定。商业智能技术提供使企业迅速分析数据的技术和方法, 包括收集、管理和分析数据, 将这些数据转化为有用的信息, 然后分发到企业各处。从技术应用的角度来看, 可将商业智能理解为数据仓库+OLAP+数据挖掘。

实施商业智能系统是一项复杂的系统工程, 整个项目涉及企业管理、运营管理、信息系统、数据仓库、数据挖掘和统计分析等众多门类的知识。因此, 用户除了要选择合适的软件工具外, 还必须按照正确的实施方法才能保证项目得以成功。商业智能项目的实施步骤大致如下:

(1) 需求分析

需求分析是商业智能实施的第一步, 在其他活动开展之前, 必须明确地定义企业对商业智能系统的期望和需求, 包括需要分析的主题、各主题可能查看的维度、需要发现企业哪些方面的规律等。

(2) 数据仓库建模

通过对企业需求的分析, 建立企业数据仓库的逻辑模型和物理模型, 并规划好系统的应用架构, 将企业各类数据按照分析主题进行组织和归类。

(3) 数据抽取

数据仓库建立后, 必须将数据从业务系统中抽取到数据仓库中, 在抽取的过程中还必须将数据进行转换和清洗, 以适应分析的需要。

(4) 建立商业智能分析报表

分析报表需要专业人员按照用户制订的格式进行开发, 用户也可自行开发。

(5) 用户培训和数据模拟测试

对用户进行培训, 在实际环境中对系统进行测试, 以便发现和修改问题。

(6) 系统改进和完善

在用户使用一段时间后, 可能会提出更多的、更具体的需求, 这时, 就需要按照上述步骤对系统进行重构或完善。

5. 电子商务

电子商务通常是指是在全球各地广泛的商业贸易活动中, 在 Internet 开放的网络环境下, 基于浏览器/服务器应用方式, 买卖双方不谋面地进行各种商贸活动, 实现消费者的网上购物、商户之间的网上交易和在线电子支付, 以及各种商务活动、交易活动、金融活动和相关的综合服务活动的一种新型的商业运营模式。

参与电子商务的实体有 4 类：顾客（个人消费者或集团购买）、商户（包括销售商、制造商、储运商）、银行（包括发卡行、收单行）及认证中心。

要实现完整的电子商务会涉及很多方面，除了买家、卖家外，还要有银行或金融机构、政府机构、认证机构、配送中心等机构的加入才行。由于参与电子商务的各方在物理上是互不谋面的，因此整个电子商务过程并不是物理世界商务活动的翻版，网上银行、在线电子支付等条件和数据加密、电子签名等技术在电子商务中发挥着重要的作用。

电子商务按从事商务活动的主体不同，分为 3 种类型，即企业间的电子商务（B2B）、企业与消费者之间的电子商务（B2C），以及消费者之间的电子商务（C2C）。

- 企业间的电子商务：即企业与企业（Business to Business, B2B）之间，通过 Internet 或专用网方式进行电子商务活动。如阿里巴巴、慧聪网等属于 B2B 类型。
- 企业与消费者（Business to Customer, B2C）之间的电子商务：即企业通过 Internet 为消费者提供一个新型的购物环境——网上商店，消费者通过网络在网上购物、在网上支付。由于这种模式节省了客户和企业双方的时间和空间，大大提高了交易效率，节省了不必要的开支，因此受到了很大程度的欢迎。如当当网、京东商城、China-pub 网上书店、淘宝商城等属于 B2C 类型。
- 消费者与消费者（Customer to Customer, C2C）之间的电子商务：主要体现在一些个人开办的网上商城和二手交易上，这种交易是在消费者之间发生的。如淘宝网就属于这种交易模式。

6.1.5 应用集成技术

在企业信息化建设的过程中，由于缺乏统一规划和总体布局，往往形成多个信息孤岛。信息孤岛使数据的一致性无法得到保证，信息无法共享和反馈，需要重复多次采集和输入。信息孤岛是企业信息化一个重要的负面因素，其主要原因既有技术因素也有管理因素，还有业务流程和标准方面的因素。如何将众多的信息孤岛联系起来，以便让不同的系统之间交互信息，是当前很多企业都面临的一个问题。

企业应用集成（Enterprise Application Integration, EAI）技术可以消除信息孤岛，它将多个企业信息系统连接起来，实现无缝集成，使它们就像一个整体一样。EAI 是伴随着企业信息系统的产生和演变的，企业的价值取向是推动 EAI 技术发展的原动力，而 EAI 的实现反过来也驱动企业竞争优势的提升。

1. 传统企业应用集成

传统的企业应用集成包括界面集成、数据集成、应用集成和过程集成，其中数据集成是其他集成方法的基础。

（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3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”,使之像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。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。过程集成的要素包括过程管理,过程建模和工作流。

2. EDI

EDI是英文Electronic Data Interchange的缩写,中文可译为“电子数据交换”。它是一种在公司之间传输订单、发票等作业文件的电子化手段。它通过计算机通信网络将贸易、运输、保险、银行和海关等行业信息,用一种国际公认的标准格式,实现各有关部门或公司与企业之间的数据交换与处理,并完成以贸易为中心的全部过程,它是20世纪80年代发展起来的一种新颖的电子化贸易工具,是计算机、通信和现代管理技术相结合的产物。国际标准化组织(ISO)将EDI描述成“将贸易(商业)或行政事务处理按照一个公认的标准变成结构化的事务处理或信息数据格式,从计算机到计算机的电子传输”。而ITU-T(原CCITT)将EDI定义为“从计算机到计算机之间的结构化的事务数据互换”。又由于使用EDI可以减少甚至消除贸易过程中的纸面文件,因此EDI又被人们通俗地称为“无纸贸易”。

根据功能,EDI可分为4类。

第一类常见的EDI系统是订货信息系统,是最基本的,也是最知名的EDI系统。它又可分为贸易数据互换系统(Trade Data Interchange, TDI),它用电子数据文件来传输订单、发货票和各类通知。

第二类常用的EDI系统是电子金融汇兑系统(Electronic Fund Transfer, EFT),即在银行和其他组织之间实行电子费用汇兑。EFT已使用多年,但它仍在不断地改进中。最大的改进是同订货系统联系起来,形成一个自动化水平更高的系统。

第三类常见的EDI系统是交互式应答系统(Interactive Query Response)。它可应用在旅行社或航空公司作为机票预定系统。这种EDI在应用时要询问到达某一目的地的航班,要求显示航班的时间、票价或其他信息,然后根据旅客的要求确定所要的航班,打印机票。

第四类是带有图形资料自动传输的EDI。最常见的是计算机辅助设计(Computer Aided Design, CAD)图形的自动传输。比如,设计公司完成一个厂房的平面布置图,将其平面布置图传输给厂房的主人,请主人提出修改意见。一旦该设计被认可,系统将自动输出订单,发出购买建筑材料的报告。在收到这些建筑材料后,自动开出收据。如美国一个厨房用品制造公司——Kraft Maid公司,在PC上以CAD设计厨房的平面布置图,再用EDI传输设计图纸、订货、收据等。

3. 应用集成数据交换方式

在进行应用集成时，很多时候都无法避免数据集成的问题，数据集成是其他应用层面集成的一个基础。数据集成其实就是进行数据交换、数据共享。常见的数据交换方式包括：共享数据库、消息传递、文件传输，不同的场合使用不同的数据交换方式。一般需要从数据交换的频度、时实度要求、数据量等方面来综合考虑采用哪一种交换方式。下面将详细介绍这几种方式。

(1) 共享数据库

共享数据库是指在应用集成时，让多个应用系统通过直接共享数据库的方式，来进行数据交换。这种方式，实时性强、可以频繁交互，数据的交换属于同步方式。由于这种方式涉及数据安全问题、并发控制问题、数据死锁问题，所以目前一般不采用这种方式进行集成。

(2) 消息传递

消息是软件对象之间进行交互和通信时所使用的一种数据结构。它可以独立于软件平台而存在。适用于数据量小、但要求频繁地、立即地、可靠地、异步地数据交换场合。

(3) 文件传输

文件传输指在进行数据交换时，直接将数据文件传送到相应位置，让目标系统直接读取相关数据。这种方式可以一次性传送大量信息，但不适合频繁进行数据传送。所以这种方式适用于数据量大、交换频度小、即时性要求低的情况。

6.1.6 企业门户

随着互联网的快速发展，企业门户（Enterprise Portal，EP）已经成为企业优化业务模式、扩展市场渠道、改善客户服务，以及提升企业形象和凝聚力的强有力手段。EP之所以具有极大的吸引力，关键在于它具备广泛的用途和灵活、全面的模型。随着电子商务的发展，EP已经成为新型办公环境的重要组成部分。

按照实际应用领域，EP可以划分为4类，分别是企业网站、企业信息门户、企业知识门户和企业应用门户。

1. 企业网站

随着互联网的兴起，企业纷纷建立自己的网站，供用户或企业员工浏览。这些网站往往功能简单，注重信息的单向传送，忽视用户与企业间、用户相互之间的信息互动。这些网站面向特定的使用人群，为企业服务，因此，可以被看做是EP发展的雏形。

2. 企业信息门户

企业信息门户（Enterprise Information Portal，EIP）是指在Internet环境下，把各种应用系统、数据资源和互联网资源统一集成到EP之下，根据每个用户使用特点和角色的不同，形成个性化的应用界面，并通过对事件和消息的处理传输把用户有机地联系在一起。EIP不仅仅局限于建立一个企业网站，提供一些企业和产品/服务的信息，更重要的是要求企业能实现多业务系统的集成，能对客户的各种要求做出快速响应，并且能对整个供应链进行统一管理。企业员工、合作伙伴、客户、供应商都可以通过EIP非常方便地获取自己所需的信息。对访问者来说，EIP提供了一个单一的访问入

口,所有访问者都可以通过这个入口获得个性化的信息和服务,可以快速了解企业的相关信息;对企业来说,EIP 既是一个展示企业的窗口,也可以无缝地集成企业的业务内容、商务活动、社区等,动态地发布存储在企业内部和外部的各种信息,同时还可以支持网上的虚拟社区,访问者可以相互讨论和交换信息。

3. 企业知识门户

企业知识门户(Enterprise Knowledge Portal, EKP)是企业员工日常工作所涉及相关主题内容的“总店”。企业员工可以通过 EKP 方便地了解当天的最新消息、工作内容、完成这些工作所需的知识等。通过 EKP,任何员工都可以实时地与工作团队中的其他成员取得联系,寻找到能够提供帮助的专家或者快速地连接到相关的门户。不难看出,EKP 的使用对象是企业员工,它的建立和使用可以大大提高企业范围内的知识共享,并由此提高企业员工的工作效率。当然,EKP 还应该具有信息收集、整理、提炼的功能,可以对已有的知识进行分类,建立企业知识库并随时更新知识库的内容。目前,一些咨询和服务型企业已经开始建立企业知识门户。

4. 企业应用门户

企业应用门户(Enterprise Application Portal, 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。从某种意义上说,可以把 EAP 看成是企业信息系统的集成界面。企业员工和合作伙伴可以通过 EAP 访问相应的应用系统,实现移动办公、进行网上交易等。

EIP、EKP 和 EAP 虽然能满足不同应用的需求,但随着企业信息系统复杂程度的增加,越来越多的企业需要能够将它们有机地整合在一起,形成一个通用型的企业门户。通用型的企业门户应该随访问者角色的不同,允许其访问企业内部网上的相应应用和信息资源。除此之外,还要提供先进的搜索功能、内容聚合能力、目录服务、安全性、应用/过程/数据集成、协作支持、知识获取、前后台业务系统集成等多种功能。给企业员工、客户、合作伙伴、供应商提供一个虚拟的工作场所。

6.2 典型试题分析

试题 1

企业系统规划方法(BSP)是指导公司建立信息系统的方法。一个企业的信息系统应当满足各个管理层次关于信息的需求。以下选项中不属于企业系统规划层次的是(1)。

- | | |
|--------------|----------|
| (1) A. 战略控制层 | B. 管理控制层 |
| C. 操作控制层 | D. 数据产生层 |

试题分析

企业系统规划方法(Business System Planning, BSP)最早由 IBM 公司于 20 世纪 70 年代研制并使用的一种企业信息系统开发的方法。虽然 30 多年过去了,但是,这种方法对于今天我国企业信息系统建设仍然具有一定的指导意义。

BSP 方法是企业战略数据规划方法和信息工程方法的基础,也就是说,后两种方法是在 BSP 方法的基础上发展起来的,因此,了解并掌握 BSP 方法对于全面掌握信息系统开发方法是有帮助的。BSP 方法的目的是提供一个信息系统规划,用以支持企业短期和长期的信息需求。

信息系统是一个企业的有机组成部分,并对企业的总体有效性起关键作用,一定要支持组织的企业需求并直接影响其目标,因而规划过程必须是企业战略转化的过程,信息系统的战略应当表达企业中各管理层次(战略计划层、管理控制层、操作控制层)的需求,必须向整个组织提供一致性的信息。信息系统应在组织机构和管理体制改变时保持工作能力。

试题答案

(1) D

试题 2

信息工程的基础是信息战略规划,规划的起点是将(2)和企业的信息需求转换成信息系统目标,实施信息系统工程是要为企业建立起具有稳定数据型的数据处理中心,以满足各级管理人员关于信息的需求,它坚持以(3)为信息处理的中心。

- (2) A. 事务处理 B. 现行人工和电算化混合的信息系统
C. 企业战略目标 D. 第一把手要求

- (3) A. 数据 B. 过程 C. 功能 D. 应用

试题分析

信息工程形成了“以数据为中心”,而不是“以应用为中心”的开发方法,并在方法中强调以数据为战略资源,以数据规划为基础的信息工程方法,它以主题数据库的组织和实施来实现,并提供直到系统完成的各阶段的实施方法。

信息战略规划是信息工程实施的起点,也是信息工程的基础。信息战略规划的目的是要评估实施信息工程的企业的信息需求,即为了实现企业的战略目标,保持企业的竞争优势,要对企业所需的信息进行评估。并在评估信息需求的基础上去建立这些信息需求的信息结构,进而建立支持相应信息结构的业务系统结构,并确定支持业务系统结构所必需的技术结构,最后提交信息战略规划的结果。

试题答案

(2) C (3) A

试题 3

信息战略规划报告应由 3 个主要部分组成:摘要、规划和附录。其中,摘要涉及的主题包括信息战略规划所涉及的范围、企业的业务目标和战略重点、信息技术对企业业务的影响、对现有信息环境的评价、推荐的系统战略、推荐的技术战略、推荐的组织战略、推荐的行动计划等,其中系统战略是关于(4)和(5)的总结。

- (4) A. 技术结构规划 B. 整体网络规划
C. 数据库结构规划 D. 信息结构规划

- (5) A. 业务系统结构规划 B. 机构结构规划
C. 过程结构规划 D. 系统发展规划

试题分析

信息战略规划报告的读者首先应是企业的高层管理者，因此，规划人员应以他们为对象来编写报告，不能将报告写成一份纯技术性的文件。

一般认为，信息战略规划报告应由3个主要部分组成。

① 摘要。摘要是从信息战略规划的主体抽取形成的，其目的是回答高层管理所最关心和有兴趣的问题。摘要通常不要多于5页，其内容应涉及信息战略规划所涉及的范围、企业的业务目标和战略重点、信息技术对企业业务的影响、对现有信息环境的评价、推荐的系统战略、推荐的技术战略、推荐的组织战略、推荐的行动计划。其中，系统战略是关于信息结构规划和业务系统结构规划的总结；技术战略是关于技术结构的总结；组织战略是关于信息系统组织进行机构改革的建议；行动计划是指要执行的主要项目，项目的持续时间，硬件设备获得的时间。

② 规划。信息战略规划是报告的主体内容，它详细说明执行摘要中相关的要点、所使用的表格、图形和插图表达的重要信息。其篇幅为40~70页，不宜过长。其主要内容包括：

- 阐述总体内容。包括规划的范围、规划的委托人、规划组成员。
- 业务环境描述。包括企业的任务、目标、关键成功因素、信息需求及组织结构。
- 评价现有信息环境，确定在满足业务环境需求方面存在的问题。
- 通过可选方案和推荐的信息结构、业务系统结构、技术结构，阐明其优点，确定问题的解决方案。
- 最后给出推荐的行动计划。

③ 附录。大部分规划的详细内容包含在附录中，并可考虑是否形成一个可用于存放技术信息的信息战略规划技术报告。

试题答案

- (4) D (5) A

试题4

关于电子政务与传统政务的比较，以下论述不正确的是 (6) 。

- (6) A. 办公手段不同 B. 与公众沟通方式存在差异
C. 业务流程一致 D. 电子政务是政务活动一种新的表现形式

试题分析

电子政务是指国家机关在政务活动中全面应用现代信息技术进行管理和办公，并向社会公众提供服务。电子政务建设不是简单地将政府原有的职能和业务流程计算机化或网络化。由于在信息化的背景下，政府获取信息、处理信息、传播信息的难度大大降低，使得政府在行为方式和组织结构等方面的优化重组成为现实。所以，电子政务是一项重要的政府创新，是政务活动的一种新的表现形式，它可以导致政府结构的

调整和业务流程的重组，实现资源的最优化配置。

传统的政务办公以纸质文件或传统媒体为信息传递、交流的媒介，而电子政务可以通过电子邮件、协同办公系统、WWW 网站等交换、发布信息，办公手段和其与公众沟通的手段有了重大的变化，变得交互性更强、效率更高。

试题答案

(6) C

试题 5

关于信息资源和信息资源管理，以下论述不正确的是___(7)___。

- (7) A. 信息资源与人力、物力、财力等资源一样，都是企业的重要资源
- B. 信息资源管理包括数据资源管理和信息处理管理
- C. 信息资源管理成为知识经济时代企业文化建设的重要组成部分
- D. 信息资源管理的基础是数据库管理

试题分析

信息资源管理是指组织在业务活动（如生产和经营活动）中对信息的产生、获取、处理、存储、传输和使用进行全面的的管理。

信息资源与人力、物力、财力和自然资源一样，都是企业的重要资源，应该像管理其他资源那样管理信息资源。信息资源管理包括数据资源管理和信息处理管理，前者强调对数据的控制（维护和安全），后者则关心企业管理人员如何获取和处理信息（流程和方法）。

信息资源管理的基础是数据管理。数据库管理仅仅负责物理数据库的设计、实现、安全性和维护工作；而数据管理在于确定数据规划、数据应用、数据标准、数据内容、数据范围等。

信息资源的有效管理必然使信息和信息技术渗透到组织的各个部门，影响到所有职员的工作与生活，使信息文化融入到企业文化中，成为知识经济时代企业文化建设的重要组成部分。

试题答案

(7) D

试题 6

在关于信息属性的叙述中，错误的是___(8)___。

- (8) A. 信息具有无限性和普遍性
- B. 信息具有依附性，信息必须依附于某种载体
- C. 信息具有相对性，即不同的认识主体从同一事物中获取的信息及信息量可能是不同的
- D. 信息具有可传递性，信息在空间的传递称为信息存储

试题分析

人们通过深入研究，发现信息的特征如下：

① 客观性。信息是客观事物在人脑中的反映。而反映的对象则有主观和客观的区别，因而，信息可分为主观信息和客观信息。主观信息，如决策、指令、计划等；客观信息，如国际形势、经济发展等信息。

② 普遍性。物质的普遍性决定了信息的普遍存在，因而信息无所不在。

③ 无限性。客观世界是无限的，反映客观世界的信息自然也是无限的。

④ 动态性。信息随着时间的变化而变化，因而是动态的。

⑤ 依附性。信息是客观世界的反映，因而要依附于一定的载体而存在，需要有物质的承担者。信息不能完全脱离物质而独立存在。

⑥ 变换性。信息通过处理可以实现变换或转换，使其形式和内容发生变化，以适应特定的需要。

⑦ 传递性。信息在时间上的传递就是存储，在空间上的传递就是转移或扩散。

⑧ 层次性。客观世界是分层次的，反映它的信息也是分层次的。

⑨ 系统性。信息可以表示为一种集合，不同类别的信息可以形成不同的整体。因而，可以形成与现实世界相对应的信息系统。

⑩ 转化性。信息的产生不能没有物质，信息的传递不能没有能量，有效地使用信息可以把信息转化为物质或能量。

试题答案

(8) D

试题 7

电子政务根据其服务的对象不同，可以分为 4 种模式。某政府部门内部的“办公自动化系统”属于 (9) 模式。

(9) A. G2B B. G2C C. G2E D. G2G

试题分析

根据其服务对象的不同，电子政务基本上可以分为 4 种模式：政府对政府的电子政务（G2G）、政府对企业的电子政务（G2B）、政府对公众的电子政务（G2C）、政府对公务员的电子政务（G2E）。

以上 4 种模式是一种互动关系。政府部门是以两种身份来参与信息化建设的，它既是公共信息平台的使用者，也是建设的组织者，这两种身份相辅相成，互相促进。

关于 4 种模式的详细解释请参看“6.1.3 政府信息化与电子政务”。

显然，政府部门内部的“办公自动化系统”服务属于 G2E。

试题答案

(9) C

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。

试题答案

(12) B

试题 10

企业每个业务过程都包含一定数量的业务活动。业务活动是企业功能分解后最基本的、不可再分解的最小功能单元。以下关于业务活动的叙述不正确的是 (13)。

- (13) A. 业务活动之间是相对独立的
B. 业务活动之间有清晰的时空界限
C. 每一个业务活动都应是可执行的
D. 业务活动可能会产生不确定的结果或几个结果

试题分析

业务活动是企业功能分解后最基本的、不可再分解的最小功能单元。业务活动之间是相对独立的，有清晰的时空界限，每一个业务活动都应是可执行的，其结果是确定的且唯一的。

试题答案

(13) D

试题 11

企业应用集成是一个战略意义上的方法，它从服务和信息角度将多个信息系统绑定在一起，提供实时交换信息和影响流程的能力。(14) 提供企业之间的信息共享能力，(15) 在用户使用角度能够对集成系统产生一个“整体”的感觉。

- (14) A. API 集成 B. 数据集成 C. 界面集成 D. 过程集成
(15) A. API 集成 B. 数据集成 C. 界面集成 D. 过程集成

试题分析

请参看试题 9 的分析。

试题答案

(14) B (15) C

试题 12

希赛公司拥有多个应用系统，分别采用不同的语言和平台，独立构建而成，企业需要集成来自不同系统的数据，并使用可定制格式的数据频繁地、立即地、可靠地、异步地传输数据。以下集成方式，最能满足这种要求的是 (16)。

- (16) A. 文件共享 B. 数据库共享 C. 远程方法调 D. 消息机制

试题分析

在本题中，希赛公司拥有多个应用系统，分别采用不同的语言和平台独立构建而成，由此造成了信息孤岛现象。现在需要集成来自不同系统的数据，并使用可定制格式的数据频繁地、立即地、可靠地、异步地传输数据。这种需要是一种数据集成，因此，在所给出的选项中，应该使用消息机制来实现数据集成，而其他3个选项都无法实现这个要求。

试题答案

(16) D

试题 13

应用集成是一项十分复杂的工作，必须针对具体情况选择合适的集成方法。某企业欲整合部门之间的报表信息，在年末进行数据集成与处理，并要求保留部门内部现有信息系统的数据格式不变。在这种情况下，采用___(17)___的方式最为适当。

(17) A. 共享数据库 B. 远程过程调用 C. 消息传递 D. 文件传输

试题分析

在试题所给出的情况下，数据集成的频率很低，且需要保持现有的数据格式，因此，采用文件传输的方式最为简单方便。

试题答案

(17) D

试题 14

下面关于 ERP 的叙述，不正确的是___(18)___。

- (18) A. ERP 为组织提供了升级和简化其所用的信息技术的机会
B. 购买使用一个商业化的 ERP 软件，转化成本高，失败的风险也很大
C. 除了制造和财务，ERP 系统可以支持人力资源、销售和配送
D. ERP 的关键是事后监控企业的各项业务功能，使得诸如质量、有效性、客户满意度、工作成果等可控

试题分析

ERP 系统是一个整合的程序集合，能够管理公司各个节点，支持组织内制造和财务等业务功能。灵活性和快速反应是企业开展商业竞争取胜的重要因素，而实施 ERP 能够使用户在第一时间获得信息，帮助企业更好地为客户服务，提高质量标准，评估市场状态，实施 ERP 会对整个组织带来巨大的冲击，其优缺点如下：

- ① 促使企业改变工作流程。
- ② 能够整合原有的旧系统，促使企业技术基础设施升级。
- ③ 提高对决策数据访问的效率。
- ④ 费用较高，时间较长。

⑤ 难以与其他系统集成,一旦选定 ERP 软件,很难向其他 ERP 系统转换。

ERP 的关键是实时监控企业功能,这使得诸如质量、有效性、客户满意度、工作成果和获利能力等关键事务得到及时分析。除了制造和财务外,ERP 系统还可以支持人力资源、销售和配送。

试题答案

(18) D

试题 15

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(19)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。

(19) A.数据 B.接口 C.过程 D.界面

试题分析

本题考查企业应用集成的方式和特点。

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而又独立运行的企业信息系统联合起来。目前市场上主流的集成模式有 3 种,分别是面向信息的集成、面向过程的集成和面向服务的集成。其中,面向过程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。

试题答案

(19) C

试题 16

电子数据交换(EDI)是电子商务活动中采用的一种重要的技术手段。以下关于 EDI 的叙述中,错误的是(20)。

- (20) A. EDI 的实施需要一个公认的标准和协议,将商务活动中涉及的文件标准化和格式化
B. EDI 的实施在技术上比较成熟,成本也较低
C. EDI 通过计算机网络,在贸易伙伴之间进行数据交换和自动处理
D. EDI 主要应用于企业与企业、企业与批发商之间的批发业务

试题分析

本题主要考查电子数据交换(EDI)的基本概念和特点。

电子数据交换是电子商务活动中采用的一种重要的技术手段。EDI 的实施需要一个公认的标准和协议,将商务活动中涉及的文件标准化和格式化;EDI 通过计算机网络,在贸易伙伴之间进行数据交换和自动处理;EDI 主要应用于企业与企业、企业与批发商之间的批发业务;EDI 的实施在技术上比较成熟,但是实施 EDI 需要统一数据

格式，成本与代价较大。

试题答案

(20) B

试题 17

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 (21) 活动，不属于商业智能范畴。

- (21) 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

试题分析

本题主要考查商业智能的基本概念。

商业智能是利用数据挖掘技术、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、识别数据模式、获取决策支持并得出结论。商业智能技术主要体现在“智能”上，即通过对大量数据的分析，得到趋势变化等重要知识，并为决策提供支持。选项 A、C、D 都是对数据进行分析，获得知识的过程；选项 B 仅仅是获取数据，并没有对数据进行分析，因此不属于商业智能范畴。

试题答案

(21) B

6.3 实战练习题

- 以下关于信息和信息化的论述中，不正确的是 (1) 。

- (1) A. 信息化就是开发利用信息资源，促进信息交流和知识共享，提高经济增长质量，推动经济社会发展转型的历史进程。
- B. 信息、材料和能源共同构成经济和社会发展的三大战略资源，这三者之间不可以相互转化
- C. 信息是“用以消除随机不确定的东西”
- D. 信息资源是重要的生产要素

• “企业系统规划方法”和“信息工程”都推荐建立表示数据类（主题数据库）和过程之间关系的 CU 矩阵 M 。其中若第 i 号过程产生第 k 号数据类，则 $M_{ik}=C$ ；若第 j 号过程使用第 k 号数据类，则 $M_{jk}=U$ 。矩阵 M 按照一定的规则进行调整后，可以给出划分系统的子系统方案，并可确定每个子系统相关的 (2) 和 (3)；同时

也可了解子系统之间的__(4)___。

- (2) A. 关系数据库 B. 层次数据库
C. 网状数据库 D. 共享数据库
- (3) A. 关系数据库 B. 网状数据库
C. 专业(私有)数据库 D. 子集数据库
- (4) A. 过程引用 B. 功能关系 C. 数据存储 D. 数据通信

● 某行政单位希望将位于单位内部信息系统中的新闻、信息等发布到互联网上,并实现一部分网上办公能力。对于这种应用集成需求,最适合的集成方式是__(5)___。

- (5) A. 数据集成 B. 界面集成 C. API 集成 D. 门户集成

● 对 ERP 项目最恰当的定位是__(6)___。

- (6) A. 信息系统集成项目 B. 管理变革项目
C. 技术改造项目 D. workflow 实施项目

● 建立企业信息系统应该遵循一定的原则,以下原则不适当的是__(7)___。

- (7) A. 必须支持企业的战略目标 B. 应该自上而下地规划和实现
C. 应该支持企业各个管理层的需求 D. 应该向整个企业提供一致的信息

● 在开发一个企业管理信息系统时,首先要进行用户调查,调查中收集的主要信息包括__(8)___。

- (8) A. 管理目标、人力资源、业务流程和数据流程信息
B. 组织结构、功能体系、业务流程和数据流程信息
C. 企业性质、客户资源、业务流程和数据流程信息
D. 管理目标、功能体系、业务流程和数据流程信息

● 电子政务根据其服务的对象不同,基本上可以分为 4 种模式。某市政府在互联网上提供的“机动车违章查询”服务,属于__(9)___模式。

- (9) A. G2B B. G2C C. G2E D. G2G

● 以下关于信息系统的论述中,正确的是__(10)___。

- (10) A. 信息系统可以是手工的,也可以是计算机化的
B. 信息系统就是计算机化的信息系统
C. 基于计算机的信息系统由硬件、软件、数据库、远程通信等组成,不包括人和规程
D. 信息系统计算机化一定能提高系统的性能

● 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__(11)___门户解决方案最为合适。

- (11) 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 (12)。

- (12) 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 (13)。

- (13) 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

● 希赛公司欲对其内部的信息系统进行集成，需要实现在系统之间快速传递可定制格式的数据包，并且当有新的数据包到达时，接收系统会自动得到通知。另外还要求支持数据重传，以确保传输的成功。针对这些集成需求，应该采用 (14) 的集成方式。

- (14) A. 远程过程调用 B. 共享数据库
C. 文件传输 D. 消息传递

● 以下关于企业信息化方法的叙述中，正确的是 (15)。

- (15) 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做是一种资本，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

6.4 练习题解析

试题 1 分析

从一定的意义上来说，物质（材料）、能量（能源）、信息都是人类生存和社会

发展所不可缺少的资源,其中,物质和能量是更为基本的资源,信息则是一种较为高级的资源。物质资源提供给人类的是各种材料,能量资源提供给人类的是各种动力,而信息资源提供给人类的是知识和智慧。

从潜在意义上讲,信息是可以转化的。它在一定的条件下,可以转化为物质、能量、时间及其他。信息可以转化,这当然需要条件,其中最主要的条件就是信息必须被人们有效地利用。没有这个条件,信息是不可能发生这种转化的。同样,“知识就是力量”也是需要这样的条件的。显然,正确而有效地利用信息,就可能在同样的条件下创造更多的物质财富,开发或节约更多的能量,节省更多的时间。在这方面,将有许多工作可做,有许多潜力可挖。

试题 1 答案

(1) B

试题 2~4 分析

BSP 认为数据类和过程是定义企业信息系统总体结构的基础,应该建立它们之间的内在联系,并可清除在考虑定义和内容时所产生的问题。过程/数据类矩阵(M)是建立二者联系的工具。其中,行表示数据类,列表示过程,并以字母 C 和 U 来表示过程对数据类的产生和使用。在矩阵中,首先按关键资源的生命周期顺序放置过程,开始是计划过程,然后是度量和控制过程,以及直接涉及产品的过程,最后是管理支持资源的过程;其次是根据过程产生数据的顺序来安排数据,开始是由计划过程产生的数据,接着把它所有数据列入矩阵,并在适当的行列交叉处填上 C 和 U 。

填写规则是:若其中第 i 号过程产生第 k 号数据类,则 $M_{ik}=C$;若第 j 号过程使用第 k 号数据类,则 $M_{jk}=U$ 。

这样,矩阵 M 按照一定的规则进行调整后,可以给出划分系统的子系统方案,并可确定每个子系统相关的共享数据库和专业(私有)数据库,同时也可了解子系统之间的数据通信。根据其对数据类的产生和使用特点,可将子系统分类如下:

- ① 产生数据类但不使用其他数据类的子系统。
- ② 使用其他数据类来产生一个数据类的子系统。
- ③ 使用数据类但不产生数据类的子系统。

试题 2~4 答案

(2) D

(3) C

(4) D

试题 5 分析

根据题目描述,该行政单位希望将位于单位内部信息系统中的相关新闻、信息等发布到互联网上,并实现一部分网上办公的能力,这是一个典型的信息门户的构建问题,因此应该采用门户集成的方式最为恰当。

试题 5 答案

(5) D

试题 6 分析

简要地说,企业的所有资源包括三大流:物流、资金流和信息流。ERP 也就是对

这3种资源进行全面集成管理的管理信息系统。概括地说,ERP是建立在信息技术基础上,利用现代企业的先进管理思想,全面地集成了企业所有资源信息,并为企业提供决策、计划、控制与经营业绩评估的全方位和系统化的管理平台。ERP系统是一种管理理论和管理思想,不仅仅是信息系统。它利用企业的所有资源,包括内部资源与外部市场资源,为企业制造产品或提供服务创造最优的解决方案,最终达到企业的经营目标。ERP理论与系统是从MRP-II(制造资源计划II)发展而来的,它的主线也是计划,但ERP已将管理的重心转移到财务上,在企业整个经营运作过程中贯穿了财务成本控制的概念。

在设计和开发ERP系统时,应该把握住一个中心、两类业务、三条干线的总体思路。一个中心就是以财务数据库为中心;两类业务就是计划与执行;三条干线则是指供应链管理、生产管理和财务管理。在ERP设计时常用的工具包括业务分析、数据流程图、实体关系图及功能模块图。

而实施ERP则是一场耗资大、周期长、涉及面广的系统工程。由于ERP软件原本是个实用性强、牵涉面较广的管理系统,在实施过程中应该采取规范的方法,严格按照ERP软件的实施方法论进行。ERP实施方法论的核心是实现管理思想革命和管理信息化技术提升。实施可以分为以下3个时期:

- ① 前期。主要是基础数据准备和标准化。
- ② 中期。进行交接面界定,业务流程重组。
- ③ 后期。实施适应期,实行手工与计算机(或新旧系统)并行作业,逐步解决不适应性。

整个实施过程包括项目启动、组建团队、设计、编码、测试、数据准备、软件安装、软件调试、项目试运行、项目正式运行等环节。具体来说包括以下工作。

- ① 明确观点、统一认识、建立实施团队。
- ② 明确目标和制定实施计划。
- ③ 根据企业人员知识结构和技术水平组织培训。
- ④ 根据企业现状进行业务需求分析。
- ⑤ 根据需求分析结果建模和进行原型分析。
- ⑥ 根据实际业务流程和具体情况进行系统功能和参数配置,以及系统实施。
- ⑦ 根据业务原型进行试运行试验,制订技术解决方案。
- ⑧ 调试环境、培训和测试。
- ⑨ 上线准备、数据准备。
- ⑩ 系统上线,投入运行。
- ⑪ 系统优化、周期性系统运行审查。

同时,ERP也随着管理思想、技术的提高而随之发展,其主要的发展趋势是:系统更加柔性化,从而更灵活地适应企业变化的需求;应用范围广泛化,渗透到各行各业;功能深度化;对供应链管理提供更好的支持;朝着Internet技术靠拢;组件技术应用更加深入;安全保密性更高;成本更低、易用性更好,以及系统自助化。

试题6答案

(6) B

试题7分析

支持整个企业需求的信息系统规模都比较大，这样大的系统应该是自上而下地规划，并自下而上地分步实现。这样，信息系统就可以按部就班地以模块化的方式进行建设，并照顾到企业的重点部门和资金投入的能力。

试题7答案

(7) B

试题8分析

在开发一个企业管理信息系统时，首先要进行用户调查，调查的范围不仅仅局限于信息和数据，还应该包括企业的生产、经营、管理等各个方面。具体来讲，调查中收集的主要信息包括组织结构与功能业务、数据和数据流程、业务流程、决策方式及过程、可用资源与限制条件、现存问题及改进。

试题8答案

(8) B

试题9分析

根据其服务对象的不同，电子政务基本上可以分为4种模式：政府对政府的电子政务（G2G）、政府对企业的电子政务（G2B）、政府对公众的电子政务（G2C）、政府对公务员的电子政务（G2E）。

以上4种模式是一种互动关系。政府部门是以两种身份来参与信息化建设的，它既是公共信息平台的使用者，也是建设的组织者，这两种身份相辅相成，互相促进。

① 政府对政府。在政府与政府之间，致力于政府办公系统自动化建设，促进信息互动、信息共享和资源整合，提高行政效率。主要包括电子法规政策系统、电子公文系统、电子司法档案系统、电子财政管理系统。

② 政府对企业。在政府与企业之间，致力于电子商务实践，营造安全、有序、合理的电子商务环境，引导和促进电子商务发展。主要包括电子采购与招标、电子税务、电子证照办理、信息咨询服务、中小企业电子服务。

③ 政府对公众。在政府与公众之间，致力于网络系统、信息渠道和在线服务的建设，为民众提供获取更便捷、质量更佳、内容更多元化的服务。主要包括教育培训服务、就业服务、电子医疗服务、社会保险网络服务、公民信息服务、交通管理服务、公民电子税务、电子证件服务。

④ 政府对公务员。随着电子政务的发展，政府部门对内部工作人员的电子政务被单独列出来，称为政府对公务员的电子政务模式。主要包括办公自动化系统、电子培训系统、业绩评价系统。

显然，“机动车违章查询”服务属于G2C。

试题9答案

(9) B

试题 10 分析

信息系统以系统思想为依据，以计算机为手段，由人和计算机等组成，进行数据收集、传递、处理、存储、分发，加工产生信息，为决策、预测和管理提供依据的系统。信息系统可以是手工的，也可以是计算机化的。

试题 10 答案

(10) A

试题 11 分析

企业门户是一个信息技术平台，这个平台可以提供个性化的信息服务，为企业提供一个单一的访问企业各种信息资源和应用程序的入口。现有的企业门户大致可以分为企业信息门户、企业知识门户和企业应用门户 3 种。其中企业信息门户重点强调为访问结构数据和无结构数据提供统一入口，实现收集、访问、管理和无缝集成。企业知识门户提供了一个创造、搜集和传播企业知识的平台，通过企业知识门户，员工可以与工作团队中的其他成员取得联系，寻找能够提供帮助的专家。企业应用门户是一个用来提高企业的集中贸易能力、协同能力和信息管理能力的平台。它以商业流程和企业应用为核心，将商业流程中功能不同的应用模块通过门户集成在一起，提高公司的集中贸易能力、协同能力和信息管理能力。

试题 11 答案

(11) C

试题 12 分析

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。在客户管理系统中，销售自动化是其中最为基本的模块，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、计划结果分析等功能。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有两种，分别是呼叫中心和互联网。CRM 系统能够与 ERP 系统在财务、制造、库存等环节进行连接，两者之间虽然关系比较独立，但由于两者之间具有一定的关系，因此会形成一定的闭环反馈结构。

试题 12 答案

(12) C

试题 13 分析

共享数据库是一种重要的企业应用集成方式，它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。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，能够在一定程度上缓解数据语义不一致的问题，但无法完全解决该问题。在共享数据库集成中，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，这会使数据库成为一个性能瓶颈。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。

试题 13 答案

(13) B

试题 14 分析

根据题干描述,该公司需要在应用集成后实现采用可定制的格式频繁地、立即地、可靠地、异步地传输数据包。远程过程调用一般是基于同步的方式,效率较低,而且容易失败;共享数据库和文件传输的集成方式在性能方面较差,系统不能保持即时数据同步,而且容易造成应用与数据紧耦合;消息传递的集成方式能够保证数据的异步、立即、可靠传输,恰好能够满足该公司的集成需求。

试题 14 答案

(14) D

试题 15 分析

本题考查信息化相关知识。

选项 A 描述错误,因为业务流程重构是对业务流程的优化,而非针对组织结构和工作方法。选项 B 描述错误,因为事务型数据库容易形成信息孤岛,而主题数据库不容易形成“信息孤岛”。C 选项描述错误,因为人力资源是把所有员工看做是一种资本,而非部分员工。

试题 15 答案

(15) D

7

第7章

系统开发基础

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

(1) 开发管理

- 项目的范围、时间、成本。
- 文档管理工作、配置管理。
- 软件开发的质量与风险。
- 软件的运行与评价。

(2) 需求管理

- 需求变更。
- 需求跟踪。
- 需求变更风险管理。

(3) 软件开发方法

- 软件开发生命周期。
- 软件开发模型（瀑布模型、演化模型、增量模型、螺旋模型、原型，构件组装模型、RUP、敏捷方法）。
- 构件与软件重用。
- 逆向工程。
- 形式化方法。

(4) 软件开发环境与工具

- 集成开发环境。
- 开发工具（建模工具、分析设计工具、编程工具、测试工具、项目管理工具等）。

(5) 设计方法

- 分析设计图示（DFD、ERD、UML、流程图、N-S图、PAD）。
- 结构化分析与设计。
- 模块设计。
- 面向对象的分析与设计。
- I/O设计、人机界面设计。
- 设计模式。

(6) 基于构件的开发

- 构件的概念与分类。
- 中间件技术。
- 典型应用架构（J2EE、.NET）。

(7) 应用系统构建

- 应用系统设计与开发（分析与设计方法的使用、外部设计、内部设计、程序设计、测试）。
- 软件包的使用（开发工具、运行管理工具、业务处理工具、ERP、群件、OA工具）。

(8) 测试与评审

- 测试评审方法。
- 验证与确认（V&V）。
- 测试自动化。
- 测试设计和管理方法。

7.1 考点突破

从历年的考试情况来看，本章的考点主要集中在以下方面：开发模型、开发环境、配置管理、需求管理、时间管理、面向对象基础、软件测试。

7.1.1 历年考试情况分析

在历年的考试试题中，有关系统开发基础知识的试题如表7-1所示。

表 7-1 系统开发基础知识试题分布表

题号	2009.11	2010.11	2011.11
21	用户文档		
22	配置项	范围定义	软件产品配置
23	需求变更	时间管理	软件质量保证
24	需求管理	变更管理	需求跟踪能力链
25	敏捷方法	CMM	需求定义方法

(续表)

题号	2009.11	2010.11	2011.11
26	项目管理工具	“4+1”视图模型	需求管理或需求开发
27			RUP
28	逆向工程	螺旋模型	
29		软件开发环境	实体类、边界类、控制类
30			
31		软件重用	最少知识原则
32	用例包含关系	开发模型	软件开发方法
33	UML 图		
34			
		有效性验证	系统设计
42	界面设计	系统测试	快速迭代原型开发
43	系统测试		静态分析
44	黑盒测试		

按照知识点进行总结和归类的试题分布情况如表 7-2 所示。

表 7-2 系统开发基础知识归类表

知识点	2009.11	2010.11	2011.11
开发管理	2	3	2
需求管理	2	0	4
软件开发方法	3	3	3
软件设计方法	4	2	4
测试与评审	2	3	3
软件开发环境与工具	2	2	0
软件过程改进	0	1	0
合计	15	14	15

从表 7-2 中可以看出,系统开发基础知识方面的内容在历年的考试中所占分值比例非常大。分数比例的趋势如图 7-1 所示。

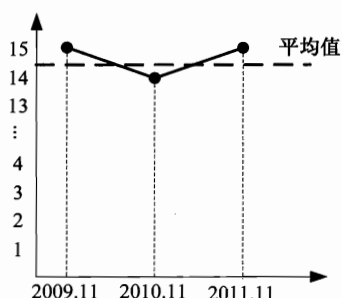


图 7-1 系统开发基础知识历年试题比例趋势图

7.1.2 开发管理

软件项目本身是复杂的,如果没有仔细地计划,复杂的项目是不可能成功的。一个

计划良好的项目将受到有效的控制,进展明显,而参加该项目的人员都会得到支持以进行其工作。软件项目本身也具有风险性,如果没有有效的风险管理也是不能成功的。

通常软件工程项目的管理比其他工程项目的管理更困难,这是因为:

- 软件产品不可见。开发的进度及产品的质量是否符合要求不明显,比较难把握。
- 没有标准的软件过程。尽管近几年来“软件过程改进”领域有许多进步,但由于团队、人员的个性化因素,还不存在一个放之四海皆准的标准化软件过程。
- 大型软件项目常常是一次性项目。由于这些项目都是“前无古人”的,因此缺乏可以借鉴的历史经验。

在 PMBOK 中提出了项目管理九大知识领域,此后这九大知识领域为业界普遍认同。九大知识领域中有四大核心领域:范围管理、时间管理、成本管理、质量管理,四大辅助领域:人力资源、沟通管理、风险管理、采购管理及负责整体协调的整体管理。

1. 整体管理

整体管理负责项目的全生命周期管理、全局性管理和综合性管理。全生命周期管理意味着项目整体管理过程负责管理项目的启动阶段直到项目收尾阶段的整个项目生命周期。全局性管理意味着项目整体管理过程负责管理项目的整体包括项目管理工作、技术工作和商务工作等。综合性管理意味着项目整体管理过程负责管理项目的需求、范围、进度、成本质量、人力资源、沟通、风险和采购。

整体管理的活动主要包括以下几个方面:

- 分析和理解范围。这包括产品需求、标准、假定、约束、项目干系人的期望和其他与项目相关的影响因素,以及在项目中如何管理和处理这些因素。
- 把产品需求和特定的标准明确记录在文档里。
- 利用计划过程组,结合项目的实际情况,制订系统的项目管理计划。
- 把完成项目需要做的工作恰当地分解为可管理的更小部分。
- 采取恰当的行动使项目按照整体的项目管理计划来实施。
- 对项目状态、过程和产品进行度量和监督。
- 分析并监控项目风险。

整体管理主要关心为达成项目目标所需的管理过程的互相配合,这些过程是为了完成一个项目的目标所要求的。项目整体管理的过程包括如下内容:

- 制订项目章程。制订一份正式批准项目或阶段的文件,并记录能反映干系人需要和期望的初步要求的过程。
- 制订项目管理计划。对定义、编制、整合和协调所有子计划所必需的行动进行记录的过程。
- 指导与管理项目执行。为实现项目目标而执行项目管理计划中所确定的工作的过程。
- 监控项目工作。跟踪、审查和调整项目进展,以实现项目管理计划中确定的绩效目标的过程。
- 实施整体变更控制——审查所有变更请求,批准变更,管理对可交付成果、组织过程资产、项目文件和项目管理计划的变更的过程。

- 结束项目或阶段——完结所有项目管理过程组的所有活动，以正式结束项目或阶段的过程。

2. 范围管理

范围管理确定在项目内包括什么工作和不包括什么工作，由此界定的项目范围在项目的全生命周期内可能因种种原因而变化，项目范围管理也要管理项目范围的这种变化。

范围管理主要包括以下过程：

- 收集需求。为实现项目目标而定义并记录干系人的需求的过程。
- 定义范围。制定项目和产品详细描述的过程。
- 创建工作分解结构。将项目可交付成果和项目工作分解为较小的、更易于管理的组成部分的过程。
- 核实范围。正式验收项目已完成的可交付成果的过程。
- 控制范围。监督项目和产品的范围状态、管理范围基准变更的过程。

3. 时间管理

时间管理又称进度管理，时间管理的职责是保证项目的所有工作都在一个指定的时间内完成。

时间管理主要包括以下过程：

- 定义活动。识别为完成项目可交付成果而需采取的具体行动的过程。
- 排列活动顺序。识别和记录项目活动间逻辑关系的过程。
- 估算活动资源。估算各项活动所需材料、人员、设备和用品的种类和数量的过程。
- 估算活动持续时间。根据资源估算的结果，估算完成单项活动所需工作时段数的过程。
- 制订进度计划。分析活动顺序、持续时间、资源需求和进度约束，编制项目进度计划的过程。
- 控制进度。监督项目状态以更新项目进展、管理进度基准变更的过程。

4. 成本管理

成本管理就是要确保在批准的预算内完成项目。

成本管理主要包括以下过程：

- 估算成本。对完成项目活动所需资金进行近似估算的过程。
- 制定预算。汇总所有单个活动或工作包的估算成本，建立一个经批准的成本基准的过程。
- 控制成本。监督项目状态以更新项目预算、管理成本基准变更的过程。

5. 质量管理

质量管理是指确定质量方针、目标和职责，并通过质量体系中的质量计划、质量控制、质量保证和质量改进来使其实现的所有管理职能的全部活动。

质量管理的主要包括以下过程：

- 规划质量。识别项目及其产品的质量要求和/或标准，并书面描述项目将如何达到这些要求和/或标准的过程。
- 实施质量保证。审计质量要求和质量控制测量结果，确保采用合理的质量标准和操作性定义的过程。
- 实施质量控制。监测并记录执行质量活动的结果，从而评估绩效并建议必要变更的过程。

6. 人力资源管理

人力资源管理是为项目正常开展，而进行的招聘（选择）项目组人员并进行有效组织、考核绩效支付报酬、进行有效激励、结合组织与个人需要进行有效开发以便实现最优组织绩效的全过程。

人力资源管理主要包括以下过程：

- 制订人力资源计划。识别和记录项目角色、职责、所需技能及报告关系，并编制人员配备管理计划的过程。
- 组建项目团队。确认可用人力资源并组建项目所需团队的过程。
- 建设项目团队。提高工作能力、促进团队互动和改善团队氛围，以提高项目绩效的过程。
- 管理项目团队。跟踪团队成员的表现，提供反馈，解决问题并管理变更，以优化项目绩效的过程。

7. 沟通管理

沟通管理是指项目组为项目干系人之间进行良好沟通而进行的管理活动。其内容包括沟通计划编制、信息分发、绩效报告、项目干系人管理。

沟通管理的主要包括以下过程：

- 识别干系人。识别所有受项目影响的人员或组织，并记录其利益、参与情况和成功的影响项目的过程。
- 规划沟通。确定项目干系人的信息需求，并定义沟通方法的过程。
- 发布信息。按计划向项目干系人提供相关信息的过程。
- 管理干系人期望。为满足干系人的需要而与之沟通和协作，并解决所发生的问题的过程。
- 报告绩效。收集并发布绩效信息（包括状态报告、进展测量结果和预测情况）的过程。

8. 风险管理

风险是指“损失或伤害的可能性”。风险包含这些特征：不确定性、针对未来、会带来损失。

项目风险管理通常包括风险识别、风险估计（风险评估）和风险驾驭（风险控制）三个主要活动。风险识别的主要工作是找到潜在风险并将其文档化，它包括项目风险、技术风险和商业风险三种。风险估计则是通过对各种风险发生的可能性和破坏性这两

个方面进行评估,并将它们按优先级进行排列。风险驾驭则是指利用某种技术,如原型化、软件自动化、软件心理学、可靠性工程学等方法设法避开风险。

风险管理就是要在风险成为影响项目成功的威胁之前,识别、着手处理并消除风险的源头。

风险管理的主要包括以下过程:

- 规划风险管理。定义如何实施项目风险管理活动的过程。
- 识别风险。判断哪些风险会影响项目并记录其特征的过程。
- 实施定性风险分析。评估并综合分析风险的发生概率和影响,对风险进行优先排序,从而为后续分析或行动提供基础的过程。
- 实施定量风险分析。就已识别风险对项目整体目标的影响进行定量分析的过程。
- 规划风险应对。针对项目目标,制定提高机会、降低威胁的方案和措施的过程。
- 监控风险。在整个项目中,实施风险应对计划、跟踪已识别风险、监测残余风险、识别新风险和评估风险过程有效性的过程。

在风险管理时,经常会应用风险曝光度(Risk Exposure)这个概念,它的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 $1000000 \times 0.5\% = 5000$ 元。

9. 采购管理

采购管理是为完成项目工作,从项目团队外部购买或获取所需的产品、服务或成果的过程。

采购管理主要包括以下过程:

- 规划采购。记录项目采购决策、明确采购方法、识别潜在卖方的过程。
- 实施采购。获取卖方应答、选择卖方并授予合同的过程。
- 管理采购。管理采购关系、监督合同绩效,以及采取必要的变更和纠正措施的过程。
- 结束采购。完成单次项目采购的过程。

10. 配置管理

配置管理是一个管理学科,它对配置项的开发和支持生存期给予技术和管理上的指导。配置管理的应用取决于项目的规模、复杂程序和风险大小。配置管理的目的在于运用配置标识、配置控制、配置状态和配置审计,建立和维护工作产品的完整性。

在系统架构设计师考试中,主要是考查基本概念层面的内容,所以下面将详细说明什么是配置项、基线、里程碑等概念,同时论述配置管理的意义,以及版本管理的相关内容。

(1) 配置项

信息系统中的文档和软件在其开发、运行、维护的过程中会得到许多阶段性的成果,并且每个文档、信息系统在开发和运行过程中还需要用到多种工具软件或配置。所有这些配置项都需要得到妥善的管理,决不能出现混乱,以便在提出某些特定的要

求时,将它们进行约定的组合来满足使用的目的。这些信息项是配置管理的对象,称为配置项。它们通常可以分成下面的6种类型。

- 环境类。信息系统开发、运行和维护的环境,如编译器、操作系统、编辑软件、管理系统、开发工具、测试工具、项目管理工具、文档编制工具等。
- 定义类。需求分析与系统定义阶段结束后得到的工件,如需求规格说明书、项目开发计划、设计标准或设计准则、验收测试计划等。
- 设计类。设计阶段得到的工件,如系统设计说明书、程序规格说明、数据库设计、编码标准、用户界面设计、测试标准、系统测试计划、用户手册。
- 编码类。编码及单元测试结束后得到的工件,如源代码、目标码、单元测试用例、数据及测试结果。
- 测试类。系统测试完成后的工作,系统测试用例,测试结果,操作手册,安装手册。
- 维护类。维护阶段产品的工作,以上任何需要变更的配置项。

每个配置项的主要属性有名称、标识符、状态、版本、作者、日期等。配置项是一个独立存在的信息项,我们可以把它看成一个元素,单独的一个元素发挥不了什么作用,但随着工作的进展,出于不同的要求,需要将这些元素进行不同的组合,这个组合称为配置,配置是一个信息系统产品在生存期各个阶段的不同形式(记录特定信息的不同媒体)和不同版本的程序、文档及相关数据的集合,或者说是配置项的集合,它具有完整的意义。

(2) 基线与里程碑

基线(Baseline)是项目生存期各开发阶段末尾的特定点,也称为里程碑(Milestone),在这些特定点上,阶段工作已结束,并且已经形成了正式的阶段产品。

建立基线的概念是为了把各开发阶段的工作划分得更加明确,使得本来连续开展的开发工作在这些点上被分割开,从而更加有利于检验和肯定阶段工作的成果,同时有利于进行变更控制。有了基线的规定就可以禁止跨越里程碑去修改另一开发阶段的工作成果,并且认为建立了里程碑,有些完成的阶段成果已被冻结。

作为阶段工作的正式产品,基线应该是稳定的,如作为设计基线的设计规格说明应该是通过评审的。如果还只是设计草稿,就不能作为基线,不能被冻结。

如果把软件看做是系统的一个组成部分,以下3种基线是最受人们关注的:功能基线、分配基线、产品基线。

提出基线的概念本来是为了更好地实现变更控制,但如果把每个基线都当成一个整体来看待会造成麻烦。因为一个变更很可能只涉及基线的很小部分。例如,假定某个大型软件中的一个模块修改了,如果将这一变更当做整个软件产品基线的变更,就很不方便。

事实上,基线可由多个配置项组成,一个配置项可以是一个文档,或者是一个可直接放在配置控制之下的工作产品,能够作为一个独立的基本部件加以修改。文档通常已被认为是独立可修改的部件了,但如有必要还可将其再加以细分,把文档中的章、节甚至段当做配置项来看待。

置于配置管理之下的工作产品包括将交付给客户的产品、指定的内部工作产品、采购的产品、工具和其他用于创建和描述这些工作产品的实体。

(3) 配置管理的意义

信息系统项目的对象是信息系统，它和传统的制造产品有着很大的差别，这些差别决定了信息系统项目必须相应地采取特殊的措施，否则将无法达其目标。信息系统是不可见抽象的智力产品，其规模日益扩大和复杂，参加的人员数量日益增多，沟通工作量也越来越大，并时时处于变化之中，并对系统开发人员的依赖相当大。由于这些原因，信息项目很容易造成信息的拥有者的版本不一致，或者需要的文件找不到，或者需求变化太快以致产品与需求不一致的现象。所有这些现象用配置管理的方法都是很容易实现的。

配置管理能够解决以下问题：

- 多重维护问题。一个文档的几个副本在不同的地方使用，或者若干个文档中都含有一些共同的内容。如果一个用户发现了一个文档出现了问题便直接进行修正，或几个用户发现了问题都各自做了修正，这样，文档就不一致了。

这是配置管理最容易解决的问题，当用户需要修改某文档时，必须从配置库中检出该文档，修改后再检入，每个用户需要该文档时都从配置库中检出当前最新的文档。

- 同时修改问题。多个用户对同一个文档进行修改，这时就有可能出现有的用户的变更消失了。要解决这个问题，有两个办法，一个是同时只允许一个人检出，另一个办法是将文档分成多个文档，避免编辑冲突。
- 丢失版本或不知版本。这个问题的解决要明确规定保留哪个版本，销毁哪个版本；采用一种系统化的文档标识版本，并控制版本的变更采用统一的备份规程。

(4) 版本管理

在配置管理中，所有的配置项都应列入版本控制的范畴。对于信息产品的版本有两个方面的意思，一是为满足不同用户的不同使用要求，如用于不同运行环境的系列产品。如适合 Linux、Windows、Solaris 用户的软件产品分别称为 Linux 版、Windows 版和 Solaris 版。它们在功能和性能上是相当的，原则上没有差别，或者说，这些是并列的系列产品。对于这类差别很小的不同版本，互相也称为变体（Variant）。

另一种版本的含义是在信息系统产品投产使用后，产品经过一系列的变更，如纠错，增加功能，提高性能的更改，而形成的一系列的顺序演化的产品，这些产品也称为一个版本，每个版本都可说出它是从哪个版本导出的演化过程。

必须注意到，修正后的新版本往往不能完全代替老版本，尽管新版本有某些优越的特性。因为一些用户仍然使用着老版本，并且不容易立刻做到以旧换新，否则可能会打扰老版本原有的工作环境。显然，多个版本被多个用户同时使用的情况是不可避免的现实。这就要求多个版本共存，这也是配置管理要解决的一个重要课题。

配置项的状态通常有 3 种，分别是草稿、正式发布和正在修改。一般来说，配置项版本控制的流程如下：

- 创建配置项。
- 修改处于草稿状态的配置项。
- 技术评审或领导审批。
- 正式发布。
- 变更，修改版本号。

版本管理要解决的第一个问题是版本标识，也就是为区分不同的版本，要给它们科学地命名。通常有2种版本命名的方法，分别是号码版本标识和符号版本标识。其中号码版本标识以数字表示，如用1.0、2.0、1.2、2.1.1等表示版本号；符号版本标识是将重要的版本属性有选择地给出，如Windows 2008、Delphi 2010将版本产生的时间给出。为了从版本标识上看到更多信息，可能给出更多的属性，如面向的客户群、开发语言、硬件平台、生成日期等。

配置管理中，版本包括配置项的版本和配置的版本，这两种版本的标识应该各有特点，配置项的版本应该体现出其版本的继承关系，它主要是在开发人员内部进行区分。另外，还需要对重要的版本做一些标记，如对纳入基线的配置项版本应该做一个标识。

7.1.3 需求管理

软件需求是指用户对新系统在功能、行为、性能、设计约束等方面的期望。为了获取到软件需求并将需求准确的实现成为可用的系统，人们提出了需求工程。需求工程包括了两个部分：需求开发（偏技术）和需求管理（偏管理）。

在需求开发阶段需要确定软件所期望的用户类型，获取每种用户类型的需求，了解实际的用户任务和目标，以及这些任务所支持的业务需求。同时还包括分析源于用户的信息，对需求进行优先级分类，将所收集的需求编写成为需求规格说明书和需求分析模型，以及对需求进行评审等工作；需求管理通常包括定义需求基线、处理需求变更和需求跟踪等方面的工作。这两个方面是相辅相成的，需求开发是主线，是目标；需求管理是支持，是保障。

1. 变更管理

在需求开发阶段，无论工作做得多么完美都无法避免在系统开发的过程中产生需求变更，需求变更是项目团队不希望看到的，但也是躲不过的。不过通过良好的管理方法，是可以尽量减少变更，从而控制风险的。

需要控制好变更，一般通过这样的思路进行：分析变更产生的原因以便对症下药，成立一个变更控制委员会以便形成统一的决策机构，制定变更流程让过程是合理甚至最优的。

（1）变更产生的原因

变更是指在项目的实施过程中，由于项目环境或者其他各种原因对项目的部分或项目的全部功能、性能、体系结构、技术、指标、集成方法和项目进度等方面做出改变。项目变更是正常的、不可避免的。在项目实施过程中，变更越早，损失越小；变更越迟，难度越大，损失也越大。项目在失控的情况下，任何微小变化的积累，最终都会对项目的质量、成本和进度产生较大影响，这是一个从量变到质变的过程。

变更产生的原因主要有以下几个方面：

- 项目外部环境发生变化，如政府政策的变化等。
- 项目总体设计、需求分析不够周密详细，有一定的错误或遗漏。
- 新技术的出现，设计人员提出了新的设计方案或新的实现手段。
- 建设单位由于机构重组等原因造成业务流程的变化。

(2) 变更控制委员会

变更控制委员会（Change Control Board, CCB）也称为配置控制委员会（Configuration Control Board），其任务是对建议的配置项变更做出评价、审批，以及监督已批准变更的实施。CCB 的成员通常包括项目经理、用户代表、质量控制人员、配置控制人员。这个组织不必是常设机构，可以根据工作的需要组成，其中的人员可以全职的，也可以是兼职的。

如果 CCB 不只是控制变更，而是承担更多的配置管理任务，那就应该包括基线的审定、标识的审定，以及产品的审定等工作，并且可能实际的工作需分为项目层、系统层和组织层来组建，使其完成不同层面的配置管理任务。

(3) 变更控制的流程

一般来说，变更控制应该遵循以下基本流程：

- **变更申请。**应记录变更的提出人、日期、申请变更的内容等信息。
- **变更评估。**对变更的影响范围、严重程度、经济和技术可行性进行系统分析。
- **变更决策。**由 CCB 决定是否实施变更。
- **变更实施。**由管理者指定的工作人员在受控状态下实施变更。
- **变更验证。**由配置管理人员或受到变更影响的人对变更结果进行评价，确定变更结果和预期是否相符、相关内容是否进行了更新、工作产物是否符合版本管理的要求。
- **沟通存档。**将变更后的内容通知可能会受到影响的人员，并将变更记录汇总归档。如提出的变更在决策时被否决，其初始记录也应予以保存。

2. 需求跟踪

需求跟踪是将单个需求和其他系统元素之间的依赖关系和逻辑联系建立跟踪，这些元素包括各种类型的需求、业务规则、系统架构和构件、源代码、测试用例及帮助文件等。

(1) 正向跟踪与反向跟踪

需求跟踪包括正向跟踪和反向跟踪，正向跟踪是指检查 SRS 中的每个需求是否都能在后继工作成果中找到对应点；反向跟踪也称为逆向跟踪，是指检查设计文档、代码、测试用例等工作成果是否都能在 SRS 中找到出处。具体来说，需求跟踪涉及 5 种类型，如图 7-2 所示。

图 7-2 中的箭头表示需求跟踪能力联系链，它能跟踪需求使用的整个周期，即从需求建议到交付的全过程。



图 7-2 5 类需求可跟踪

图 7-2 的左半部分表明,从用户原始需求可向前追溯到软件需求,这样就能区分出开发过程中或开发结束后由于变更受到影响的需求,也确保了 SRS 中包括所有用户需求。同样,可以从软件需求回溯到相应的用户原始需求,确认每个软件需求的出处。如果以用例的形式来描述用户需求,图 7-2 的左半部分就是用例和功能性需求之间的跟踪情况。

图 7-2 的右半部分表明,由于在开发过程中,软件需求转变为设计和编码等实现元素,所以通过定义单个软件需求和特定的产品元素之间的联系链,可以从软件需求追溯到产品元素。这种联系链使开发人员知道每个需求对应的产品元素,从而确保产品元素满足每个需求。第四类联系链是从产品元素回溯到软件需求,使开发人员知道每个产品元素存在的原因。绝大多数项目不包括与用户需求直接相关的代码,但开发人员应该知道为什么要写这一行代码。如果不能把设计元素、代码段或测试用例回溯到一个软件需求,就可能出现画蛇添足的现象。当然,如果某个孤立的产品元素表明了一个正当的功能,则说明 SRS 漏掉了一项需求。

第五类联系链是软件需求之间的跟踪,这种跟踪便于更好地处理软件需求之间的逻辑相关性,检查需求分解中可能出现的错误或遗漏。

(2) 需求跟踪矩阵

表示需求和其他系统元素之间的联系链的最普遍方式是使用需求跟踪(能力)矩阵。不论采用何种跟踪方式,都要建立与维护需求跟踪矩阵,它保存了需求与后继工作成果的对应关系。例如,从用户原始需求到软件需求之间的跟踪,可以采用如表 7-3 所示的矩阵。

表 7-3 用户原始需求到软件需求的跟踪矩阵示例

用例 原始需求	UC-1	UC-2	UC-3	...	UC-n
FR-1					
FR-2					
⋮					
FR-m					

对于从软件需求到下游工作产品之间的跟踪,可以采用如表 7-4 所示的矩阵。

表 7-4 软件需求到下游工作产品的跟踪矩阵示例

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

表 7-4 明确展示了每个用例是如何连接到一个或多个设计、编码和测试元素的。其中,设计元素可以是模型中的对象,如 DFD、E-R 图或类图等;代码模块可以是类中的方法、源代码文件名、过程或函数。需求跟踪矩阵中可以定义各种系统元素类型间的一对一、一对多和多对多关系,也就是说,允许在表 7-4 的一个单元格中填入多

个元素来实现这些特征。例如，一个代码模块对应一个设计元素，多个测试用例验证一个功能点，每个用例导致多个功能点等。

3. 需求风险管理

人们做事情总希望一帆风顺，做项目也是如此，总是希望项目进展顺利，按照计划如期交付。但现实却是残酷的，会有许多潜在威胁和阻碍项目按计划进行的因素，这就是风险。风险可能会给项目成本、进度、质量和团队工作效率等方面带来负面影响。当然，所谓“塞翁失马，焉知非福”，风险有时候也能给项目带来机会。

风险管理的目的就是希望让项目管理人员能够“掌控”风险，风险事件一旦发生，能够按照预先制订的应对计划有条不紊地处理风险。

(1) 需求阶段的风险源

系统分析师在进行需求开发的过程中，有时也会“陷自身于困境”，无意之中给项目带来风险。这些做法列举如下：

- 无足够用户参与。在需求获取的过程中，如果没有足够的用户参与，系统分析师所获得的需求就是片面的和不完整的，这样，在需求开发之初就埋下了风险。
- 忽略了用户分类。用户不止一个人，各类用户有自己的特点和需求，如果系统分析师不能针对所有主要用户进行分类，就必然会导致有的用户对产品感到失望。例如，菜单驱动操作对高级用户太低效了，但命令和快捷键又会使不熟练的用户感到困难。
- 用户需求的不断增加。需求蔓延有可能引起项目范围蔓延，而这是项目中的大忌，因为它会对项目成本、进度和质量等方面带来很大的负面影响，甚至直接导致项目失败。
- 模棱两可的需求。模棱两可的需求会使不同的项目干系人产生不同的期望，会使开发人员为错误问题而浪费大量时间。
- 不必要的特性。这是技术人员的一个通病，喜欢画蛇添足。经常发生的情况是，用户并不认为这些添加的“足”很有用，以致在其上耗费的努力白搭，浪费项目资源。
- 过于精简的 SRS。过于精简的 SRS 为用户和开发人员提供了“无限遐想”的机会，却给项目开发带来了无限的麻烦，导致不断的修改，项目完工遥遥无期。
- 不准确的估算。系统分析师在信息不充分的情况下，如果未经深思就对需求做出估算，则这种估算通常只是一种猜测而已。一旦传递给用户，他们却认为这是一种承诺。

(2) 需求阶段的主要风险及应对措施

项目风险管理的一个主要过程是识别风险，也就是事先要“预知”项目进展过程中可能会发生的风险，然后对其进行分析，采取相应措施。根据业内人士的经验，与需求有关的主要风险及其应对措施如表 7-5 所示。

表 7-5 与需求有关的风险及其应对措施

阶段	主要风险	风险应对措施
需求获取	产品视图与范围	在项目早期写一份项目视图与范围将业务需求涵盖在内,并将其作为新的需求及修改需求的指导
	需求开发所需时间	记录参与的每个项目中实际需求开发的工作量,这样就能知道所花的时间是否合适,并改进将来项目的工作计划
	忽略市场对产品的反馈信息	强调市场调查,建立原型,并运用客户核心小组来获得产品的反馈信息
	没有非功能需求	编写非功能需求文档和验收标准,作为可接受的标准
	客户反对产品需求	确定出主要的客户,并采用产品代表的方法来确保客户代表的积极参与,确保在需求决定权上有正确的人选
	期望需求	尽量识别并记录用户的期望,提出大量的问题来提示用户,以充分表达他们的想法和建议
	把已有的产品作为需求基线	将在逆向工程中收集的需求编写成文档,并让用户评审以确保其正确性
	给出期望的解决办法	从用户描述的解决方法中提炼出其本质需求
需求分析	划分需求优先级	评估每项新需求的优先级,并与已有的工作对比,以做出相应的决策
	带来技术困难的特性	分析每项需求的可行性,以确定是否能按计划实现
	不熟悉的技术、工具/平台	明确那些高风险的需求,并留出充裕时间进行学习、实验和测试原型
需求定义	系统分析师和用户对需求的不同理解	使用高水平的系统分析师;使用模型和原型,使一些模糊的需求变得清晰
	时间压力对待确定因素的影响	记录解决每项待确定因素的负责人的名字、如何解决的,以及解决的截止日期
	SRS 的完整性和正确性	以用户的任务为中心,采用用例技术获取需求;根据场景编写需求测试用例,建立原型;让用户代表对 SRS 和分析模型进行正式评审
	具有二义性的术语	建立一本术语和数据字典,用于定义所有的业务和技术词汇
	需求说明中包括了设计	仔细评审 SRS,以确保它是在强调“做什么”,而不是“怎么做”
需求验证	未经验证的需求评审	从用户代表方获得参与需求正式评审的承诺,并尽早通过非正式评审
	审查的有效性	对参与需求评审的所有人员进行培训,以使评审工作更加有效
需求管理	需求变更	将项目视图与范围文档作为变更的参照;用户积极参与需求获取过程;将那些易于变更的需求用多种方案实现,并在设计时注意其可修改性
	需求变更过程	建立规范的变更控制流程,并严格执行
	未实现的需求	使用需求跟踪能力矩阵或相关工具
	项目范围蔓延	在项目早期编制视图与范围文档,并得到用户确认;采用迭代式开发方法

系统分析师、系统架构设计师、项目管理人员可以利用表 7-5 来识别项目中的需求风险。但要注意的是,表 7-5 只是一个总结性的风险清单,具体到每一个项目,可能都有些不同,需要根据实际情况进行增加或删减。在风险应对措施方面,也需要根据经验和项目约束,进行调整或改进。

7.1.4 软件开发方法

软件生命周期与开发模型是进行软件开发时,指导性的一些规范原则,下面将详细介绍软件开发生命周期、常见开发模型、软件重用、逆向工程等内容。

1. 软件生命周期

软件生命周期(Software Life Cycle)是人们在研究软件开发过程时所发现的一种规律性的事实。如同人的一生要经历婴儿期、少年期、青年期、老年期直至死亡这样

一个全过程一样，一个软件产品也要经历计划、分析、设计、编程、测试、维护直至被淘汰这样一个全过程。软件的这一全过程被称为软件生命周期。

目前，软件生命周期各阶段的划分尚不统一，有的分得粗些，有的分得细些，所包含的实际内容也不完全相同。最具代表性的是以下 4 种：

①1970 年，Boehm 提出了如表 7-6 所示的软件生命周期模型。

表 7-6 Boehm 定义的软件生命周期模型

计划时期		开发时期					运行时期
问题定义	可行性研究	需求分析	总体设计	详细设计	编码	测试	维护

②在 1988 年制订和公布的国家标准《GB 8566—1988 计算机软件开发规范》中将软件生命周期划分为如表 7-7 所示的 8 个阶段。

表 7-7 GB 8566—1988 定义的软件生命周期模型

可行性研究与计划	需求分析	概要设计	详细设计	实现	组装测试	确认测试	使用和维护
----------	------	------	------	----	------	------	-------

③在 20 世纪 90 年代初有了软件工程过程的概念之后，于 1995 年制订和公布的国家标准《GB/T8566—1995 信息技术 - 软件生存期过程》定义了软件生命周期的 7 个主要过程，如表 7-8 所示。

表 7-8 GB/T 8566—1995 定义的软件生命周期模型

管理过程				
获取过程	供应过程	开发过程	运行过程	维护过程
支持过程				

其中，“获取过程”和“供应过程”分别描述了软件的“获取者”（用户）和“供应者”（开发者）在开发之前的主要活动和任务。而“管理过程”和“支持过程”则贯穿于整个软件生命周期。

④1999 年，Rational 软件公司的 3 位软件工程大师 Ivar Jacobson、Grady Booch 和 James Rumbaugh 联合编写了一部划时代的著作《统一软件开发过程》（*The Unified Software Development Process*），将他们多年研究所得的软件开发方法学融合在了一起。该书清楚地说明了支持整个软件生命周期的统一软件开发过程是一个用例驱动的、以架构为中心的、迭代与增量的开发过程。统一软件开发过程是在重复一系列组成软件生命周期的循环，每次循环都包括如下 4 个阶段和 5 种 workflows，分别如表 7-9 和表 7-10 所示。

表 7-9 UP 定义的软件生命周期模型的 4 个阶段

初始阶段（inception phase）	捕捉用例，思考系统架构……
细化阶段（elaboration phase）	细化用例，设计系统架构……
构造阶段（construction phase）	程序设计，实现，α 测试……
移交阶段（transition phase）	β 测试……

表 7-10 UP 定义的软件生命周期模型的 5 种工作流

需求工作流 (requirements workflow)	捕捉需求, 使开发导向正确的系统
分析工作流 (analysis workflow)	生成一个有助于架构设计的需求描述
设计工作流 (design workflow)	建立系统设计模型
实现工作流 (implementation workflow)	实现系统
测试工作流 (test workflow)	验证实现的结果

尽管软件生命周期中各阶段的划分标准不统一, 名称也不一致, 但主要包括了计划、分析、设计、编程、测试和维护等阶段。本书主要介绍分析、设计、测试和维护阶段的工作。

2. 瀑布模型

瀑布模型严格遵循软件生命周期各阶段的固定顺序: 计划、分析、设计、编程、测试和维护, 上一阶段完成后才能进入到下一阶段, 整个模型就像一个飞流直下的瀑布。

瀑布模型有许多优点: 可强迫开发人员采用规范的方法; 严格规定了各阶段必须提交的文档; 要求每个阶段结束后, 都要进行严格评审。

但瀑布模型过于理想化, 而且缺乏灵活性, 无法在开发过程中逐渐明确用户难以确切表达或一时难以想到的需求, 直到软件开发完成之后才发现与用户需求有很大距离, 此时必须付出高额的代价才能纠正这一偏差。

3. 快速原型模型

快速原型是指快速建立起来的可以在计算机上运行的程序(它所完成的功能往往是最终软件产品功能的一个子集), 或仅仅只是一个演示界面(不具备实际可运行的功能)。快速原型模型的第一步是快速建立一个能反映用户主要需求的软件原型, 让用户在计算机上使用它, 通过实际操作了解目标系统的概貌。开发人员按照用户提出的意见快速修改原型系统, 然后再次请用户试用……, 一旦用户认为这个原型系统确实能够满足他们的需求, 开发人员便可据此书写软件需求说明, 并根据这份文档开发出可以满足用户真实需求的软件产品。

快速原型法的特点如下:

- 快速原型法最显著的特点是引入了迭代的概念。
- 快速原型法自始至终强调用户的参与。
- 快速原型法在用户需求分析、系统功能描述及系统实现方法等方面允许有较大的灵活性。用户需求可以不十分明确, 系统功能描述也可以不完整, 对于界面的要求也可以逐步完善。
- 快速原型法可以用来评价几种不同的设计方案。
- 快速原型法可以用来建立系统的某个部分。
- 快速原型法不排斥传统生命周期法中采用的大量行之有效的方法和工具, 它是与传统方法互为补充的方法。

原型化方法基于这样一种客观事实: 并非所有的需求在系统开发之前都能准确地

说明和定义。因此，它不追求也不可能要求对需求的严格定义，而是采用了动态定义需求的方法。

快速原型法适用于需求不够明确的项目。具有广泛技能高水平的原型化人员是原型实施的重要保证。原型化人员应该是具有经验与才干、训练有素的专业人员。衡量原型化人员能力的重要标准是他是否能够从用户的模糊描述中快速获取实际的需求。

由于原型法开发需要适当的快速开发工具，需要用户密切配合，所以下列情况不适合使用原型法：

- 缺乏适用的原型开发工具。
- 用户不参与、不积极配合开发过程。
- 用户的数据资源缺乏组织和管理。
- 用户的软件资源缺乏组织和管理

4. 演化模型

演化模型也是一种原型化开发方法，但与快速原型模型略有不同。在快速原型模型中，原型的用途是获知用户的真正需求，一旦需求确定了，原型即被抛弃。而演化模型的开发过程，则是从初始模型逐步演化为最终软件产品的渐进过程。也就是说，快速原型模型是一种“抛弃式”的原型化方法，而演化模型则是一种“渐进式”的原型化方法。

5. 增量模型

增量模型是第三种原型化开发方法，但它既非“抛弃式”的，也非“渐进式”的，而是“递增式”的。增量模型把软件产品划分为一系列的增量构件，分别进行设计、编程、集成和测试。每个构件由多个相互作用的模块构成，并且能够完成特定的功能。如何将一个完整软件产品分解成增量构件，因软件产品特点和开发人员的习惯而异，但使用增量模型的软件体系结构必须是开放的，加入新构件的过程必须简单方便。

6. 螺旋模型

螺旋模型综合了瀑布模型和演化模型的优点，还增加了风险分析。螺旋模型包含4个方面的活动：制订计划、风险分析、实施工程、客户评估。这4项活动恰好可以放在一个直角坐标系的4个象限，而开发过程恰好像一条螺旋线。采用螺旋模型时，软件开发沿着螺旋线自内向外旋转，每转一圈都要对风险进行识别和分析，并采取相应的对策。螺旋线第一圈的开始点可能是一个概念项目。从第二圈开始，一个新产品开发项目开始了，新产品的演化沿着螺旋线进行若干次迭代，一直运转到软件生命期结束。

7. 喷泉模型

喷泉模型主要用于描述面向对象的开发过程。喷泉一词体现了面向对象开发过程的迭代和无缝隙特征。迭代意味着模型中的开发活动常常需要多次重复，每次重复都会增加或明确一些目标系统的性质，但却不是对先前工作结果的本质性改动。无缝隙是指在开发活动（如分析、设计、编程）之间不存在明显的边界，而是允许各开发活动交叉、迭代地进行。

8. 基于构件的模型

构件（Component）是一个具有可重用价值的、功能相对独立的软件单元。基于构件的软件开发（Component Based Software Development, CBSD）模型是利用模块化方法，将整个系统模块化，并在一定构件模型的支持下，复用构件库中的一个或多个软件构件，通过组合手段高效率、高质量地构造应用软件系统的过程。基于构件的开发模型融合了螺旋模型的许多特征，本质上是演化型的，开发过程是迭代的。基于构件的开发模型由软件的需求分析和定义、体系结构设计、构件库建立、应用软件构建、测试和发布 5 个阶段组成。

构件作为重要的软件技术和工具得到了极大的发展，这些新技术和工具有 Microsoft 的 DCOM、Sun 的 EJB、OMG 的 CORBA 等。基于构件的开发活动从标识候选构件开始，通过搜索已有构件库，确认所需要的构件是否已经存在，如果已经存在，就从构件库中提取出来复用；如果不存在，就采用面向对象方法开发它。在提取出来的构件通过语法和语义检查后，将这些构件通过胶合代码组装到一起实现系统，这个过程是迭代的。

基于构件的开发方法使得软件开发不再一切从头开始，开发的过程就是构件组装的过程，维护的过程就是构件升级、替换和扩充的过程，其优点是构件组装模型导致了软件的复用，提高了软件开发的效率；构件可由一方定义其规格说明，被另一方实现，然后供给第三方使用；构件组装模型允许多个项目同时开发，降低了费用，提高了可维护性，可实现分步提交软件产品。

缺点是由于采用自定义的组装结构标准，缺乏通用的组装结构标准，引入具有较大的风险；可重用性和软件高效性不易协调，需要精干的、有经验的分析人员和开发人员，一般的开发人员插不上手，客户的满意度低；过分依赖于构件，构件库的质量影响着产品质量。

构件中有一类较为特殊，即中间件。中间件工作于操作系统与应用程序之间，分布式应用软件借助这种软件在不同的技术之间共享资源。中间件有以下几种类型：

- 远程过程调用：它是一种广泛使用的分布式应用程序处理方法。应用程序使用 RPC 来远程执行一个位于不同地址空间里的过程，并且从效果上看和执行本地调用相同。要注意的是，这里的“远程”既可以指不同的计算机，也可以指同一台计算机上的不同进程。一个 RPC 应用可分为两个部分，分别是服务器和客户。这里的“服务器”和“客户”是指逻辑上的进程，而不是指物理计算机。
- 面向消息的中间件：利用高效可靠的消息传递机制进行平台无关的数据交换，并基于数据通信来进行分布式系统的集成。通过提供消息传递和消息排队模型，它可在分布式环境下扩展进程间的通信，并支持多种通信协议、语言、应用程序、硬件和软件平台。例如，IBM 的 MQSeries、BEA 的 MessageQ 等都属于面向消息的中间件产品。
- 事务处理监控器：也称为交易中间件，是当前应用最广泛的中间件之一。它能支持数以万计的客户进程对服务器的并发访问，使系统具有极强的扩展性，因此，适于电信、金融、证券等拥有大量客户的领域。在对效率、可靠

性要求严格的关键任务系统中具有明显优势。TPM 一般支持负载均衡,支持分布式两阶段提交,保证事务完整性和数据完整性,并具有安全认证和故障恢复等功能,能很好地满足应用开发的要求。

- 数据库访问中间件:通过一个抽象层访问数据库的技术,从而允许使用相同或相似的代码访问不同的数据库资源。例如,常见的 ODBC 与 JDBC 就属于数据库访问中间件。

9. 快速应用开发 (RAD) 模型

快速应用开发 (Rapid Application Development, RAD) 模型是一个增量型的软件开发过程模型,强调极短的开发周期。RAD 模型是瀑布模型的一个“高速”变种,通过大量使用可复用构件,采用基于构件的建造方法赢得快速开发。如果需求理解得好且约束了项目的范围,利用这种模型可以很快地创建出功能完善的“信息系统”。其流程从业务建模开始,随后是数据建模、过程建模、应用生成、测试及反复。

RAD 模型各个活动期所要完成的任务如下。

- 业务建模:以什么信息驱动业务过程运作?要生成什么信息?谁生成它?信息流的去向是哪里?由谁处理?可以辅之以数据流图。
- 数据建模:为支持业务过程的数据流,找数据对象集合,定义数据对象属性,与其他数据对象的关系构成数据模型,可辅之以 E-R 模型。
- 过程建模:使数据对象在信息流中完成各业务功能。创建过程以描述数据对象的增加、修改、删除、查找,即细化数据流图中的处理框。
- 应用程序生成:利用第四代语言 (4GL) 写出处理程序,重用已有构件或创建新的可重用构件,利用环境提供的工具自动生成并构造出整个应用系统。
- 测试与交付,由于大量重用,一般只做总体测试,但新创建的构件还是要测试的。
- 与瀑布模型相比, RAD 模型不采用传统的第三代程序设计语言来创建软件,而是采用基于构件的开发方法,复用已有的程序结构(如果可能的话)或使用可复用构件,或创建可复用的构件(如果需要的话)。在所有情况下,均使用自动化工具辅助软件创造。很显然,加在一个 RAD 模型项目上的时间约束需要“一个可伸缩的范围”。如果一个业务能够被模块化使得其中每一个主要功能均可以在不到 3 个月的时间内完成,那么它就是 RAD 的一个候选者。每一个主要功能可由一个单独的 RAD 组来实现,最后再集成起来形成一个整体。

RAD 模型通过大量使用可复用构件加快了开发速度,对信息系统的开发特别有效。但是像所有其他软件过程模型一样, RAD 方法也有其缺陷:

- 并非所有应用都适合 RAD。RAD 模型对模块化要求比较高,如果有哪一项功能不能被模块化,那么建造 RAD 所需要的构件就会有问题;如果高性能是一个指标,且该指标必须通过调整接口使其适应系统构件才能赢得, RAD 方法也有可能不能奏效。
- 开发者和客户必须在很短的时间完成一系列的需求分析,任何一方配合不当都会导致 RAD 项目失败。

- RAD 只能用于信息系统开发,不适合技术风险很高的情况。当一个新应用要采用很多新技术或当新软件要求与已有的计算机程序有较高的互操作性时,这种情况就会发生。

10. UP

UP (Unified Process, 统一过程方法) 是一个统一的软件开发过程,是一个通用过程框架,可以应付种类广泛的软件系统、不同的应用领域、不同的组织类型、不同的性能水平和不同的项目规模。UP 是基于构件的,这意味着利用它开发的软件系统是由构件构成的,构件之间通过定义良好的接口相互联系。在准备软件系统所有蓝图的时候,UP 使用的是统一建模语言(UML)。

与其他软件过程相比,UP 具有 3 个显著的特点:用例驱动、以基本架构为中心、迭代和增量。

UP 中的软件过程在时间上被分解为 4 个顺序的阶段,分别是初始阶段、细化阶段、构建阶段和交付阶段。每个阶段结束时都要安排一次技术评审,以确定这个阶段的目标是否已经满足。如果评审结果令人满意,就可以允许项目进入下一个阶段。

4 个阶段的核心任务分别如下。

(1) 初始阶段

- 明确地说明项目规模。这涉及了解环境及最重要的需求和约束,以便得出最终产品的验收标准。
- 计划和准备商业理由。评估风险管理、人员配备、项目计划和成本/进度/收益率折中的备选方案。
- 综合考虑备选构架,评估设计和自制/外购/复用方面的折中,从而估算出成本、进度和资源。此处的目标在于通过对一些概念的证实来证明可行性。该证明可采用可模拟需求的模型形式或用于探索被认为高风险区域的初始原型。初始阶段的原型设计工作应该限制在确信解决方案可行就可以了。该解决方案在细化和构建阶段实现。
- 准备项目的环境,评估项目和组织,选择工具,决定流程中要改进的部分。

(2) 细化阶段

- 快速确定构架,确认构架并为构架建立基线。
- 根据此阶段获得的新信息改进前景,对推动构架和计划决策的最关键用例建立可靠的了解。
- 为构建阶段创建详细的迭代计划并为其建立基线。
- 改进开发案例,定位开发环境,包括流程和支持构建团队所需的工具和自动化支持。
- 改进构架并选择构件。评估潜在构件,充分了解自制/外购/复用决策,以便有把握地确定构建阶段的成本和进度。集成了所选构架构件,并按主要场景进行了评估。通过这些活动得到的经验有可能导致重新设计构架、考虑替代设计或重新考虑需求。

(3) 构建阶段

- 资源管理、控制和流程优化。
- 完成构件开发并根据已定义的评估标准进行测试。
- 根据前景的验收标准对产品发布版进行评估。

(4) 产品化阶段（提交阶段）

- 执行部署计划。
- 对最终用户支持材料定稿。
- 在开发现场测试可交付产品。
- 制作产品发布版。
- 获得用户反馈。
- 基于反馈调整产品。
- 使最终用户可以使用产品。

基于 UP 的软件过程是一个迭代过程。通过初始、细化、构建和提交 4 个阶段就是一个开发周期，每次经过这 4 个阶段就会产生一代软件。除非产品退役，否则通过重复同样的 4 个阶段，产品将演化为下一代产品，但每一次的侧重点都将放在不同的阶段上。这些随后的过程称为演化过程。

在进度和工作量方面，所有阶段都各不相同。尽管不同的项目有很大的不同，但一个中等规模项目的典型初始开发周期应该预先考虑到工作量和进度间的分配，如表 7-11 所示。

表 7-11 UP 各阶段的工作量和进度分配

	初始阶段	细化阶段	构建阶段	提交阶段
工作量	5%	20%	65%	10%
进度	10%	30%	50%	10%

对于演进周期，初始和细化阶段就小得多了。能够自动完成某些构建工作的工具将会缓解此现象，并使得构建阶段比初始阶段和细化阶段的总和还要小很多。

11. XP 方法

XP 方法即极限编程，是敏捷方法中的一种。其他敏捷方法还有：自适应开发、水晶方法、特性驱动开发等，它们都有一个共同的特点，那就是都将矛头指向了“文档”，它们认为传统的软件工程方法文档量太“重”了，称为“重量级”方法，而相应的敏捷方法则是“轻量级”方法。

在 XP 方法中，提出了四大价值观：沟通、简单、反馈、勇气。五大原则：快速反馈、简单性假设、逐步修改、提倡更改、优质工作。

12 个最佳实践：

(1) 计划游戏

计划游戏的主要思想就是先快速地制订一份概要的计划，然后，随着项目细节的不断清晰，再逐步完善这份计划。计划游戏产生的结果是一套用户故事及后续的一两

次迭代的概要计划。

“客户负责业务决策，开发团队负责技术决策”是计划游戏获得成功的前提条件。也就是说，系统的范围、下一次迭代的发布时间、用户故事的优先级应该由客户决定，而每个用户故事所需的开发时间、不同可选技术的成本、如何组建团队、每个用户故事的风险及具体的开发顺序应该由开发团队决定。

（2）小型发布

XP 方法秉承的是“持续集成、小步快走”的哲学，也就是说每一次发布的版本应该尽可能地小，当然前提条件是每个版本有足够的商业价值，值得发布。

由于小型发布可以使得集成更频繁，客户获得的中间结果越频繁，反馈也就越频繁，客户就能够实时地了解项目的进展情况，从而提出更多意见，以便在下一次迭代中计划进去，以实现更高的客户满意度。

（3）隐喻

相对而言，隐喻比较令人费解。词典中的解释是：“一种语言的表达手段，它用来暗示字面意义不相似的事物之间的相似之处”。

如果能够找到合适的隐喻是十分快乐的，但并不是每一种情况都可以找到恰当的隐喻，因此，没有必要去强求，而是顺其自然。

（4）简单设计

强调简单的价值观，引出了简单性假设原则，落到实处就是“简单设计”实践。这个实践看上去似乎很容易理解，但却又经常被误解，许多批评者就指责 XP 忽略设计是不正确的。其实，XP 的简单设计实践并不是要忽略设计，而且认为设计不应该在编码之前一次性完成，因为那样只能建立在“情况不会发生变化”或者“我们可以预见所有的变化”之类的谎言的基础上的。

（5）测试先行

当笔者第一次看到“测试先行”这个概念的时候，第一感觉就是不解，陷入了“程序都还没有写出来，测试什么呀？”的迷思。于是，开始天马行空地寻求相关的隐喻，终于找到了能够启发我的泥瓦匠。首先，来看看两个不同泥瓦匠是如何工作的吧：

工匠甲：先拉上一根水平线，砌每一块砖时，都与这根水平线进行比较，使得每一块砖都保持水平。

工匠乙：先将一排砖都砌完，然后再拉上一根水平线，看看哪些砖有问题，对有问题砖进行适当的调整。

显然，读者都会觉得工匠乙的做法浪费时间。然而，仔细想想，自己平时在编写程序时其实就是按照工匠乙的方法的，甚至有时候比工匠乙还笨，是整面墙都砌完了，直接进行“集成测试”，经常让整面的墙倒塌。

对于有些团队而言，没有采用工匠甲的工作方法，甚至有时候程序员会以“开发工作太紧张”为理由，而忽略测试工作。这样，就导致了一个恶性循环，越是没空编写测试程序，代码的效率与质量越差，花在找 Bug、解决 Bug 的时间也越多，实际产

能大大降低。由于产能降低了，因此时间更紧张，压力更大。

(6) 重构

重构是一种对代码进行改进而不影响功能实现的技术，XP 需要开发人员在“闻到代码的坏味道”时，有重构代码的勇气。重构的目的是降低变化引发的风险、使得代码优化更加容易。通常重构发生在以下两种情况之下：

- 实现某个特性之前：尝试改变现有的代码结构，以使得实现新的特性更加简单。
- 实现某个特性之后：检查刚刚写完的代码后，认真检查一下，看是否能够进行简化。

在考虑重构时，应该养成编写并经常运行测试代码的习惯；要先编写代码，再进行重构；把每一次增加功能都当作一次重构的好时机；将每一个纠正错误当作一次重构的时机。重构技术是对简单设计的一个良好的补充，也是 XP 中重视“优质工作”的体现，这也是优秀的程序员必备的一项技能。

(7) 结对编程

结对编程是指在开发中，程序员每两人一组，当甲编码时，乙在一旁检查（即看着甲编码），乙编码时反之。自从 20 世纪 60 年代开始，就有类似的实践在进行，长年以来的研究结果给出的结论是：结对编程的效率反而比单独编程更高。一开始虽然会牺牲一些速度，但慢慢地，开发速度会逐渐加快。究其原因，主要是结对编程大大降低了沟通的成本，提高了工作的质量。具体表现在以下几个方面：

- 所有的设计决策确保了不是由一个人做出的。
- 系统的任何一个部分都肯定至少有两个人以上熟悉。
- 几乎不可能有两个人都忽略的测试项或者其他任务。
- 结对组合的动态性，是一个组织进行知识管理的好途径。
- 代码总是能够保障被评审过。

而且，XP 方法集成的其他最佳实践也能够使得结对编程更加容易进行：

- 编码标准可以消除一些无谓的分歧。
- 隐喻可以帮助结对伙伴更好地沟通。
- 简单设计可以使结对伙伴更了解他们所从事的工作。

结对编程技术被誉为 XP 保持工作质量、强调人文主义的一个最典型的实践，应用得当还能够使开发团队协作更加顺畅、知识交流与共享更加频繁、团队稳定性也会更加牢固。

(8) 集体代码所有制

由于 XP 方法鼓励团队进行结对编程，而且认为结对编程的组合应该动态地搭配，根据任务的不同、专业技能的不同进行最优组合。因此，每一个人都会遇到不同的代码，代码的所有制就不再适合于私有，因为那样会给修改工作带来巨大的不便。

所谓集体代码所有制，就是团队中的每个成员都拥有对代码进行改进的权利，每个人都拥有全部代码，也都需要对全部代码负责。同时，XP 强调代码是谁破坏的（修

改后出现问题），就应该由谁来修复。

（9）持续集成

持续集成要求 XP 团队每天尽可能多次地做代码集成，每次都在确保系统运行的单元测试通过之后进行。这样，就可以及早地暴露、消除由于重构、集体代码所有制所引入的错误，从而减少解决问题的痛苦。

（10）每周工作 40 小时

这是最让开发人员开心、管理者反对的一个最佳实践了，加班、再加班早已成为开发人员的家常便饭，也是管理者最常使用的一种策略。而 XP 方法认为，加班最终会扼杀团队的积极性，最终导致项目的失败，这也充分体现了 XP 方法关注人的因素比关注过程的因素更多一些。

不过，有一点是需要解释的，“每周工作 40 小时”中的“40”不是一个绝对数，它所代表的意思是团队应该保证按照“正常的时间”进行工作，不要加班。

（11）现场客户

现场客户是 XP 强调沟通的又一表现。其强调的是开发团队应该和客户能够随时沟通，可以是面谈、可以是在线聊天、可以是电话，当然面谈是必不可少的。其中的关键是当开发人员需要客户做出业务决策时，需要进一步了解业务细节时，能够随时找到相应的客户。

（12）编码标准

编码标准是一个“雅俗共享”的最佳实践，不管是代表重型方法论的 UP、PSP，还是代表敏捷方法的 XP，都认为开发团队应该拥有一个编码标准。XP 方法认为拥有编码标准可以避免团队在一些与开发进度无关的枝末细节问题上发生争论，而且会给重构、结对编程带来很大的麻烦。不过，XP 方法的编码标准的目的是不是创建一个事无巨细的规则列表，而是能够提供一个确保代码清晰，便于交流的指导方针。

12. 构件与软件重用

构件也称为组件，是一个功能相对独立的具有可重用价值的软件单元。在 OO 方法中，一个构件由一组对象构成，包含了一些协作的类的集合，它们共同工作来提供一种系统功能。可重用性（可复用性）是指系统和（或）其组成部分能在其他系统中重复使用的程度。软件开发的全生命周期都有可重用的价值，包括项目的组织、软件需求、设计、文档、实现、测试方法和测试用例，都是可以被重复利用和借鉴的有效资源。可重用性体现在软件的各个层次，通用的、可重用性高的软件模块往往已经由操作系统或开发工具提供，例如，通用库、标准构件和模板库等，它们并不需要程序员重新开发。

软件重用的形式可分为垂直式重用和水平式重用。水平式重用是重用不同应用领域中的软件元素，例如，数据结构、排序算法、人机界面构件等。标准函数库是一种典型的原始的水平式重用机制；垂直式重用是在一类具有较多公共性的应用领域之间重用软件构件。由于在两个截然不同的应用领域之间进行软件重用潜力不大，所以垂直式重用受到广泛关注。垂直式重用活动的主要关键点在于领域分析，即根据应用领

域的特征和相似性,预测构件的可重用性。一旦根据领域分析确认了构件的可重用价值,即可进行构件的开发,并对具有可重用价值的构件做一般化处理,使它们能够适应新的类似的应用领域。然后将构件和它们的文档存入可重用构件库,成为可供未来开发项目使用的可重用资源。

(1) CORBA 构件标准

CORBA 是由 OMG 制定的一个工业标准,其主要目标是提供一种机制,使得对象可以透明地发出请求和获得应答,从而建立一个异质的分布式应用环境。OMG 给出的以对象请求代理(Object Request Broker, ORB)为中心的对象管理结构如图 7-3 所示。

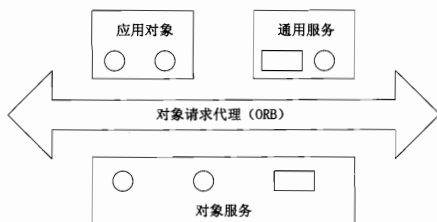


图 7-3 对象管理结构

在 OMG 的对象管理结构中,ORB 是一个关键的通信机制,它以实现互操作性为主要目标,处理对象之间的消息分布。对象服务实现基本的对象创建和管理功能,通用服务则使用对象管理结构所规定的类接口实现一些通用功能。针对 ORB,OMG 又进一步提出了 CORBA 技术规范,主要内容包括接口定义语言(Interface Definition Language, IDL)、接口池(Interface Repository, IR)、动态调用接口(Dynamic Invocation Interface, DII)和对象适配器(Object Adapter, OA)等。

CORBA 定义了一种面向对象的构件开发方法,使不同的应用系统可以共享构件。每个对象都将其内部操作细节封装起来,同时又向外界提供精确定义的接口,从而降低了应用系统的复杂性,也降低了软件开发费用。CORBA 的平台无关性实现了对象的跨平台引用,开发人员可以在更大的范围内选择最实用的对象加入到自己的应用系统之中。CORBA 的语言无关性使开发人员可以在更大的范围内相互利用别人的编程技能和成果。

(2) EJB 构件标准

EJB 是用于开发和部署多层结构的、分布式的、面向对象的 Java 应用系统的跨平台的构建架构。使用 EJB 编写的应用程序具有可扩展性和交互性,以及多用户安全的特性。这些应用只需要写一次,就可以发布到任何支持 EJB 规范的服务器平台上。

(3) COM/DCOM/COM+

Microsoft 的 COM 定义了构件和它们的客户之间互相作用的方式,使得构件和客户端无须任何中介构件就能相互联系。DCOM 扩展了 COM,使其能够支持在局域网、广域网甚至 Internet 上不同计算机的对象之间的通信。使用 DCOM,应用系统就可以在位置上达到分布性,从而满足客户和应用的需求。

因为 DCOM 是 COM 的无缝扩展,所以可以将基于 COM 的应用、构件、工具和知识转移到标准化的分布式计算领域中。在做分布式计算时,DCOM 处理网络协议的

低层次的细节问题,从而使开发人员能够集中精力解决用户所要求的问题。DCOM 具有语言无关性,任何语言都可以用来创建 COM 构件。Java、Visual C++、Visual Basic、Delphi、PowerBuilder 和 Cobol 等都能够和 DCOM 很好地相互作用。

DCOM 具有位置独立性,也就是说,DCOM 使得构件的位置对用户来说完全透明,用户无须知道构件的具体位置,无论构件是位于客户的同一个进程中,还是位于地球的另一端。在任何情况下,客户连接和调用构件的方法都是一样的。DCOM 不仅无须改变源码,而且无须重新编译程序。仅仅使用一个简单的再配置动作,就可以改变构件之间相互连接的方式。

COM+则不再局限于 COM 的组件技术,它更加注重于分布式网络应用的设计和实现,已经成为 Microsoft 系统平台策略和软件发展策略的一部分。COM+继承了 COM 几乎全部的优势,同时又避免了 COM 实现方面的一些不足。COM+倡导了一种新的概念,它把 COM 组件软件提升到应用层而不再是底层的软件结构,它通过操作系统的各种支持,使组件对象模型建立在应用层上,把所有组件的底层细节留给操作系统。

13. 逆向工程

逆向工程(Reverse Engineering)术语源于硬件制造业,相互竞争的公司为了了解对方设计和制造工艺的机密,在得不到设计和制造说明书的情况下,通过拆卸实物获得信息,软件的逆向工程也基本类似,不过,通常“解剖”的不仅是竞争对手的程序,而且还包括本公司多年前的产品。软件的逆向工程是分析程序,力图在比源代码更高抽象层次上建立程序的表示过程,逆向工程是设计的恢复过程。

与逆向工程相关的概念有重构、设计恢复、再工程和正向工程。

- 重构(Restructuring)。重构是指在同一抽象级别上转换系统描述形式。
- 设计恢复(Design Recovery)。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。
- 再工程(Re-Engineering)。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 3 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。
- 正向工程(Forward Engineering)。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。

一般认为,凡是在软件生命周期内将软件某种形式的描述转换成更为抽象形式的活动都可称为逆向工程。逆向工程的完备性可以用在某一个抽象层次上提供信息的详细程度来描述。逆向工程过程应该能够导出过程的设计模型(实现级,一种底层的抽象)、程序和数据结构信息(结构级,稍高层次的抽象)、对象模型、数据和控制流模型(功能级,相对高层的抽象)和 UML 状态图和部署图(领域级,高层抽象)。随着抽象层次的增高,完备性就会降低。抽象层次越高,它与代码的距离就越远,通过逆向工程恢复的难度就越大,而自动工具支持的可能性相对变小,要求人参与判断和推理的工作增多。

逆向工程不仅应用于软件开发，也应用于软件维护。对于一项具体的维护任务，一般不必导出所有抽象级别上的信息，例如，如果只是希望完成代码重构任务，则只需获得实现级信息即可。当然，若能进行深入分析，产生的代码质量会更好些。

7.1.5 软件设计方法

本节主要讲述软件设计相关知识，包括结构化分析与设计、面向对象分析与设计、人机界面设计。

1. 结构化分析与设计

这种方法采用结构化技术来完成软件开发的各项任务。该方法把软件生命周期的全过程依次划分为若干阶段，然后顺序地完成每个阶段的任务，与瀑布模型有很好的结合度，是与其最相适应的开发方法。

结构化分析方法学也称为生命周期方法学，它采用结构化分析、设计、编程来完成软件开发的各项任务。它具有阶段性、推迟实现、文档管理三大特点。目前，虽然面向对象开发方法已经取代了结构化方法成为软件方法学的主流，但对于一些功能需求非常明确而且不轻易改变的软件系统，结构化方法仍然比较有效。

(1) 结构化分析基础

结构化分析（解决“做什么”的问题）是一种面向数据流的需求分析方法。其基本思想是：自顶而下，逐层分解，把一个大问题分解成为若干个小问题，每个小问题再分解成若干个更小的问题。经过逐层分解，每个最底层的问题都是足够简单、容易解决的，于是复杂的问题也就迎刃而解了。在结构化分析时，常使用的工具包括数据流图和数据字典，整个分析的结果由一套分层的数据流图、一本数据词典、一组小说明（加工的逻辑说明）及软件需求规格说明书组成。

分层数据流图（DFD）是结构化分析中用得最多的工具。

(2) 结构化设计基础

结构化设计包括体系结构设计、接口设计、数据设计和过程设计等任务。它是一种面向数据流的设计方法，是以结构化分析阶段所产生的成果为基础，进一步自顶而下、逐步求精和模块化的过程。

- 概要设计：主要是设计软件的结构、确定系统是由哪些模块组成，以及每个模块之间的关系。它采用的是结构图（包括模块、调用、数据）来描述程序的结构，还可以使用层次图和 HIPO（层次图加输入/处理/输出图）。整个过程主要包括：复查基本系统模型、复查并精化数据流图、确定数据流图的信息流类型（包括交换流和事务流）、根据流类型分别实施变换分析或事务分析、根据软件设计原则对得到的软件结构图进一步优化。
- 详细设计：确定应该如何具体地实现所要求的系统，得出对目标系统的精确描述。它采用自顶向下、逐步求精的设计方式和单入口单出口的控制结构。经常使用的工具包括程序流程图、盒图、PAD 图（问题分析图）、PDL（伪码）。

(3) 模块设计原则

在结构化方法中，模块化是一个很重要的概念，它是将一个待开发的软件分解成为若干个小的简单部分——模块，每个模块可以独立地开发、测试。这是一种复杂问

题的“分而治之”原则，其目的是使程序的结构清晰、易于测试与修改。

具体来说，模块是指执行某一特定任务的数据结构和程序代码。通常将模块的接口和功能定义为其外部特性，将模块的局部数据和实现该模块的程序代码称为内部特性。而在模块设计时，最重要的原则就是实现信息隐蔽和模块独立。模块通常具有连续性，也就是意味着作用于系统的小变动将导致行为上的小变化，同时规模说明的小变动也将影响到一小部分模块。

- 信息隐蔽：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。通常我们将难的决策、可能修改的决策、数据结构的内部连接，以及对它所做的操作细节、内部特征码、与计算机硬件有关的细节等隐蔽起来。

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。

- 模块独立：模块独立是指每个模块完成一个相对独立的特定子功能，并且与其他模块之间的联系最简单。保持模块的高度独立性，也是在设计时的一个很重要的原则。通常我们用耦合（模块之间联系的紧密程度）和内聚（模块内部各元素之间联系的紧密程度）两个标准来衡量，我们的目标是高内聚、低耦合。

模块的内聚类型通常可以分为7种，根据内聚度从高到低排序如表7-12所示。

表 7-12 模块的内聚类型

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	所包含的任务必须在同一时间间隔内执行（如初始化模块）
逻辑内聚	完成逻辑上相关的一组任务
偶然内聚	完成一组没有关系或松散关系的任务

与此相对应，模块的耦合类型通常也分为7种，根据耦合度从低到高排序如表7-13所示。

表 7-13 模块的耦合类型

耦合类型	描 述
非直接耦合	没有直接联系，互相不依赖对方
数据耦合	借助参数表传递简单数据
标记耦合	一个数据结构的一部分借助于模块接口被传递
控制耦合	模块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	与软件以外的环境有关
公共耦合	多个模块引用同一个全局数据区
内容耦合	一个模块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

除了满足以上两大基本原则之外，通常在模块分解时还需要注意：保持模块的大小适中；尽可能减少调用的深度；直接调用该模块的个数应该尽量大，但调用其他模块的个数则不宜过大；保证模块是单入口、单出口的；模块的作用域应该在之内；功能应该是可预测的。

2. 面向对象分析与设计

这种方法引入了“对象”的概念，将数据和方法封装在一起，提高了模块的聚合度，降低了耦合度，更大程度上支持软件重用，是现在最流行和最具有发展前景的软件开发方法。

综观计算机软件发展史，许多新方法和新技术都是在编程领域首先兴起，进而发展到软件生命周期的前期阶段——分析与设计阶段。结构化方法经历了从“结构化编程”、“结构化设计”到“结构化分析”的发展历程，面向对象的方法也经历了从“面向对象的编程（Object-Oriented Programming, OOP）”、“面向对象的设计（Object-Oriented Design, OOD）”到“面向对象的分析（Object-Oriented Analysis, OOA）”的发展历程。1989 年之后，面向对象方法的研究重点开始转向软件生命周期的分析阶段，并将 OOA 和 OOD 密切地联系在一起，出现了一大批面向对象的分析与设计（OOA&D）方法。截止到 1994 年，公开发表并具有一定影响的 OOA&D 方法已达 50 余种。

由于各种 OOA 方法所强调的重点与该方法的主要特色不同，因此所产生的 OOA 模型从整体形态、结构框架到具体内容都有较大的差异。下面将介绍 OOA、OOD 及 OOP 所需要完成的工作，以及相关具有代表性的一些方法。

（1）OOA/OOD/OOP

OOA 的任务是了解问题域所涉及的对象、对象间的关系和操作，然后构造问题的对象模型。问题域是指一个包含现实世界事物与概念的领域，这些事物和概念与所设计的系统要解决的问题有关。在这个过程中，抽象是最本质和最重要的方法。针对不同的问题，可以选择不同的抽象层次，过简或过繁都会影响到对问题的本质属性的了解和解决。

OOD 在分析对象模型的基础上，设计各个对象、对象之间的关系（如层次关系、继承关系等）和通信方式（如消息模式）等，其主要作用是对 OOA 的结果作进一步的规范化整理，以便能够被 OOP 直接接受。

OOP 指系统功能的编码，实现在 OOD 阶段所规定的各个对象所应完成的任务。它包括每个对象的内部功能的实现，确立对象哪一些处理能力应在哪些类中进行描述，确定并实现系统的界面、输出的形式等。

（2）OMT 方法简介

1991 年，James Rumbaugh 在《面向对象的建模与设计》（*Object-Oriented Modeling and Design*）一书中提出了面向对象分析与设计的 OMT（Object Modeling Technique）方法。

OMT 方法的 OOA 模型包括对象模型、动态模型和功能模型。

对象模型表示静态的、结构化的系统的“数据”性质。它是对模拟客观世界实体的对象及对象彼此间的关系的映射，描述了系统的静态结构，通常用类图表示。

动态模型表示瞬时的、行为化的系统的“控制”性质，它规定了对象模型中的对象的合法变化序列，通常用状态图表示。

功能模型表示变化的系统的“功能”性质，它指明了系统应该“做什么”，因此更直接地反映了用户对目标系统的需求，通常用数据流图表示。

OMT方法的3个模型，分别从3个不同侧面描述了所要开发的系统：功能模型指明了系统应该“做什么”；动态模型明确了什么时候做（即在何种状态下接受了什么事件的触发）；对象模型则定义了做事情的实体。这3种模型相互补充、相互配合。

（3）Coad/Yourdon 方法

Coad/Yourdon方法特别强调OOA和OOD采用完全一致的概念和表示法，使分析和设计之间不需要表示法的转换。该方法的特点是表示简练、易学，对于对象、结构、服务的认定较系统、完整，可操作性强。

在Coad/Yourdon方法中，OOA的任务主要是建立问题域的分析模型。分析过程和构造OOA概念模型的顺序由5个层次组成，分别是类与对象层、属性层、操作层、结构层和主题层，它们分别表示分析的不同侧面。OOA需要经过5个步骤来完成整个分析工作，即标识对象类、标识结构与关联（包括继承、聚合、组合、实例化等）、划分主题、定义属性和定义操作。

OOD中将继续贯穿OOA中的5个层次和5个活动，它由4个部分组成，分别是人机交互组件、问题域组件、任务管理组件和数据管理组件，其主要的活动就是这4个组件的设计工作。

（4）Booch 方法

Booch认为软件开发是一个螺旋上升的过程，每个周期包括标识类和对象、确定类和对象的含义、标识关系、说明每个类的接口和实现4个步骤。它的模型中主要包括以下几种图，如表7-14所示。

表 7-14 模型中的图

	静态模型	动态模型
逻辑模型	类图 对象图	状态转换图 时序图
物理模型	模块图 进程图	

（5）OOSE

OOSE在OMT的基础上，对功能模型进行了补充，提出了“用例”的概念，最终取代了数据流图来进行需求分析和建立功能模型。OOSE方法采用5类模型来建立目标系统，分别是需求模型、分析模型、设计模型、实现模型和测试模型。

OOSE的开发活动主要分为3类，分别是分析、构造和测试。其中，分析过程分为需求分析和健壮性分析两个子过程，分析活动分别产生需求模型和分析模型；构造活动包括设计和实现两个子过程，分别产生设计模型和实现模型；测试过程包括单元测试、集成测试和系统测试3个过程，共同产生测试模型。

用例是 OOSE 中的重要概念, 在开发各种模型时, 它是贯穿 OOSE 活动的核心, 描述了系统的需求及功能。用例实际上是描述系统参与者 (既可以是用户, 也可以是与系统交互的其他系统) 对于系统的使用情况, 是从参与者的角度来确定系统的功能。因此, 首先必须分析、确定系统的参与者, 然后进一步考虑参与者的主要任务和使用方式, 再识别出所使用的事件, 即用例。

3. UML

随着面向对象技术的高速发展, 人们急需统一的方法学和建模规范, 而正是为了实现这一目标, Booch 方法、OOSE、OMT 三大主流 OOA 技术的创始人通过融合与整理, 形成了新的标准, 这就是统一建模语言 (UML), 现在已经纳为国际标准, 是软件系统建模的主要规范之一。UML 规范的结构如图 7-4 所示。

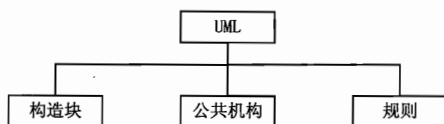


图 7-4 UML 规范的结构示意图

- 构造块: 包括事物构造块、关系和图。
 - 事物构造块: 包括结构构造块 (类、接口、协作、用例、活动类、构件、节点等)、行为构造块 (交互、状态机)、分组构造块 (包)、注释构造块。
 - 关系: 包括关联关系 (包括表示整体-部分关系的聚合、组合关系)、依赖关系、泛化关系 (表示一般/特列关系)、实现关系。
 - 图: 在 UML 2.0 中包括 14 种不同的图, 分为表示系统静态结构的静态模型 (包括类图、对象图、复合结构图、构件图、部署图、包图), 以及表示系统动态结构的动态模型 (包括用例图、活动图、状态机图、顺序图、通信图、定时图、交互概观图、制品图)。
- 规则: 也就是支配基本构造块如何放在一起的规则。最常见的语义规则有 3 种。
 - 命名: 也就是为事物、关系和图起名字; 从语义的有效性而言, 只要求是由字符、数字、下画线组成的唯一串; 另外也要求是唯一的。
 - 范围: 在 UML 1.0 中作用域包括所有者作用域 (Owner Scope) 和目标作用域 (Target Scope) 两类, 不过在 UML 2.0 中这个概念已经被简化。
 - 可见性: 用来表示编程语言中的 Public、Private、Protected 等修饰符。
- 公共机制: 是指达到特定目标的公共 UML 方法, 主要包括规格说明、修饰、公共分类和扩展机制 4 种。
 - 规格说明: 规格说明是元素语义的文本描述, 它是模型真正的“肉”。
 - 修饰: UML 为每一个模型元素设置了一个简单的记号, 还可以通过修饰来表达更多的信息。
 - 公共分类: 包括类元与实体 (类元表示概念, 而实体表示具体的实体)、接口和实现 (接口用来定义契约, 而实现就是具体的内容) 两组公共分类。

- 扩展机制：包括约束（添加新规则来扩展元素的语义）、构造型（用于定义新的 UML 建模元素）、标记值（添加新的特殊信息来扩展模型元素的规格说明）。

表 7-15 所示是对 UML 2.0 规范所规定的 14 种图的简要描述。

表 7-15 UML 2.0 的正式图形

图 名	功 能	备 注
类图	描述类、类的特性及类之间的关系	UML 1.x 原有
对象图	描述一个时间点上系统中各个对象的一个快照	UML 1.x 非正式图
复合结构图	描述类的运行时刻的分解	UML 2.0 新增
构件图	描述构件的结构与连接	UML 1.x 原有
部署图	描述在各个节点上的部署	UML 1.x 原有
包图	描述编译时的层次结构	UML 中非正式图
用例图	描述用户与系统如何交互	UML 1.x 原有
活动图	描述过程行为与并行行为	UML 1.x 原有
状态机图	描述事件如何改变对象生命周期	UML 1.x 原有
顺序图	描述对象之间的交互，重点在强调顺序	UML 1.x 原有
通信图	描述对象之间的交互，重点在于连接	UML 1 中的协作图
定时图	描述对象之间的交互，重点在于定时	UML 2.0 新增
交互概观图	是一种顺序图与活动图的混合	UML 2.0 新增
制品图	描述了一组制品及它们之间的关系	UML 2.0 新增

但要注意的是，UML 的作者们并没有把图作为 UML 的主要部分，因此各种图形并不是精确定义的，往往可以将一种图形中的成分合法地绘制到另一种图形中。在 UML 参考手册第 2 版中，将 UML 图划分为四大领域 9 种视图，如表 7-16 所示。

表 7-16 UML 视图和图

主要领域	视 图	图
结构	静态视图	类图
	设计视图	复合结构图、协作图、构件图
	用例视图	用例图
动态	状态视图	状态机图
	活动视图	活动图
	交互视图	顺序图、通信图
物理	部署视图	部署图
模型管理	模型管理视图	包图
	特性描述	包图

其中结构领域主要是对系统中的结构成员及其相互关系进行描述；动态领域则描述了系统随时间变化的行为；物理领域则描述了系统的计算资源和部署在这些资源上的系统工件；模型管理领域则说明了模型自身的分层组织结构。

4. 用户界面设计

用户界面设计是一项复杂的任务，它必须遵循一些“良好设计”的指导原则，下面介绍一些关键的用户界面设计原则。

(1) 用户控制

人机界面设计首先要确立用户类型。划分类别可以从不同的角度，视实际情况而定。确定类型后要针对其特点预测它们对不同界面的反应。这就要从多方面设计分析。用户应当感觉系统的运行在自己的控制之下。在图形界面或基于 Web 的界面中，用户指导程序的每一步执行；即使在程序进行某些处理或用户等待输出结果时，用户同样保持对控制的敏感度。

(2) 信息最小量

人机界面设计要尽量减少用户记忆负担，采用有助于记忆的设计方案。

(3) 帮助和提示

要对用户的操作命令做出反应，帮助用户处理问题。系统要设计有恢复出错现场的能力，在系统内部处理工作要有提示，尽量把主动权让给用户。

(4) 媒体最佳组合

多媒体界面的成功并不在于仅向用户提供丰富的媒体，而应在相关理论指导下，注意处理好各种媒体间的关系，恰当选用。

(5) 界面一致性

一致性要求用户界面遵循标准和常规的方式，让用户处在一个熟悉的和可预见的环境之中，这主要体现在命名、编码、缩写、布局及菜单、按钮和键盘功能在内的控制使用等。

(6) 界面容错性

一个好的界面应该以一种宽容的态度允许用户进行实验和出错，使用户在出现错误时能够方便地从错误中恢复。

(7) 界面美观性

界面美观性是视觉上的吸引力，主要体现在具有平衡和对称性、合适的色彩、各元素具有合理的对齐方式和间隔、相关元素适当分组、使用户可以方便地找到要操作的元素等。

(8) 界面可适应性

界面可适应性是指用户界面应该根据用户的个性要求及其对界面的熟知程度而改变，即满足定制化和个性化的要求。所谓定制化，是在程序中声明用户的熟知程度，用户界面可以根据熟知程度改变外观和行为；所谓个性化，是让用户按照自己的习惯和爱好设置用户界面元素。

7.1.6 测试与评审

软件测试是软件质量保证的主要手段之一，也是在将软件交付给客户之前所必须完成的步骤。目前，软件的正确性证明尚未得到根本的解决，软件测试仍是发现软件错误和缺陷的主要手段。

大量统计资料表明，目前软件测试所花费用已超过软件开发费用的 30%。

1. 软件测试基础

(1) 软件测试的目的

软件测试的目的就是在软件投入生产性运行之前，尽可能多地发现软件产品（主要是指程序）中的错误和缺陷。

为了发现程序中的错误，应竭力设计能暴露错误的测试用例。测试用例是由测试数据和预期结果构成的。一个好的测试用例是极有可能发现至今为止尚未发现的错误的测试用例。一次成功的测试是发现了至今为止尚未发现的错误的测试。

高效的测试是指用少量的测试用例，发现被测软件尽可能多的错误。

软件测试所追求的目标就是以尽可能少的时间和人力发现软件产品中尽可能多的错误。

(2) 软件测试准则

- 应该尽早地、不断地进行软件测试，把软件测试贯穿于开发过程的始终。
- 所有测试都应该能追溯到用户需求。从用户的角度看，最严重的错误是导致软件不能满足用户需求的那些错误。
- 应该从“小规模”测试开始，并逐步进行“大规模”测试。
- 应该在测试之前就制订出测试计划。
- 根据 Pareto 原理，80%的错误可能出现在 20%的程序模块中，测试成功的关键是怎样找出这 20%的模块。
- 应该由独立的第三方从事测试工作。
- 对非法和非预期的输入数据也要像合法的和预期的输入数据一样编写测试用例。
- 检查软件是否做了应该做的事仅是成功的一半，另一半是看软件是否做了不该做的事。
- 在规划测试时不要设想程序中不会查出错误。
- 测试只能证明软件中有错误，不能证明软件中没有错误。

(3) 软件测试分类

从测试阶段划分，可分为单元测试、集成测试、确认测试、系统测试。

从测试方法划分，可分为白盒测试、黑盒测试。

对于产品，可分为 α 测试与 β 测试。

在实际应用中，一旦纠正了程序中的错误后，还应选择部分或全部原先已测试过的测试用例，对修改后的程序重新测试，这种测试称为回归测试。

(4) 验证与确认

确认是运行前和变化后实施的评定，目的在于证明控制措施（或组合）能否达达预期控制水平，或满足可接受水平；而验证是在运行中或运行后进行的评定，目的在于证明确实达到了预期的控制水平或满足可接受水平。

2. V 模型

V 模型是一个著名的，以测试为驱动的开发模型，该模型强调开发过程中测试贯穿始终。V 模型如图 7-5 所示，从图中可以看出，单元测试的测试计划在编码阶段完成，集成测试计划在详细设计阶段完成，确认测试计划在概要设计阶段完成。

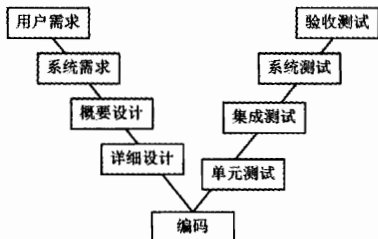


图 7-5 V 模型示意图

3. 单元测试

单元测试 (Unit Testing)，也称模块测试，通常可放在编程阶段，由程序员对自己编写的模块自行测试，检查模块是否实现了详细设计说明书中规定的功能和算法。单元测试主要发现编程和详细设计中产生的错误，单元测试计划应该在详细设计阶段制订。

单元测试期间着重从以下几个方面对模块进行测试：模块接口、局部数据结构、重要的执行通路、出错处理通路、边界条件等。

测试一个模块时需要为该模块编写一个驱动模块和若干个桩 (Stub) 模块。驱动模块用来调用被测模块，它接收测试者提供的测试数据，并把这些数据传送给被测模块，然后从被测模块接收测试结果，并以某种可以看见的方式 (如显示或打印) 将测试结果返回给测试者。桩模块用来模拟被测模块所调用的子模块，它接受被测模块的调用，检验调用参数，并以尽可能简单的操作模拟被调用的子程序模块功能，把结果送回被测模块。顶层模块测试时不需要驱动模块，底层模块测试时不需要桩模块。

模块的内聚程度高可以简化单元测试过程。如果每个模块只完成一种功能，则需要的测试方案数目将明显减少，模块中的错误也更容易预测和发现。

4. 集成测试

集成测试 (Integration Testing)，也称组装测试，它是对由各模块组装而成的程序进行测试，主要目标是发现模块间的接口和通信问题。例如，数据穿过接口可能丢失；一个模块对另一个模块可能由于疏忽而造成有害影响；把子功能组合起来可能不产生预期的主功能；个别看来是可以接受的误差可能积累到不能接受的程度；全程数据结构可能有问题等。集成测试主要发现设计阶段产生的错误，集成测试计划应该在概要设计阶段制定。

集成的方式可分为非渐增式和渐增式。

非渐增式集成是先测试所有的模块，然后把所有这些模块集成到一起，并把庞大的程序作为一个整体来测试。这种测试方法的出发点是可以“一步到位”，但测试者面对众多的错误现象，往往难以分清哪些是“真正的”错误，哪些是由其他错误引起

的“假性错误”，诊断定位和改正错误也十分困难。非渐增式集成只适合一些非常小的软件。

渐增式集成是将单元测试和集成测试合并到一起，它根据模块结构图，按某种次序选一个尚未测试的模块，把它同已经测试好的模块组合在一起进行测试，每次增加一个模块，直到所有模块被集成在程序中。这种测试方法比较容易定位和改正错误，目前在进行集成测试时已普遍采用渐增式集成。

渐增式集成又可分为自顶向下集成和自底向上集成。自顶向下集成先测试上层模块，再测试下层模块。由于测试下层模块时它的上层模块已测试过，所以不必另外编写驱动模块。自底向上集成先测试下层模块，再测试上层模块。同样，由于测试上层模块时它的下层模块已测试过，所以不必另外编写桩模块。这两种集成方法各有利弊，一种方法的优点恰好对应于另一种方法的缺点，实际测试时可根据软件特点及进度安排灵活选用最适当的方法，也可将两种方法混合使用。

5. 确认测试

确认测试（Validation Testing）主要依据软件需求说明书检查软件的功能、性能及其他特征是否与用户的需求一致。确认测试计划应该在需求分析阶段制订。

软件配置复查是确认测试的另一项重要内容。复查的目的是保证软件配置的所有成分都已齐全，质量符合要求，文档与程序完全一致，具有完成软件维护所必需的细节。

如果一个软件是为某个客户定制的，最后还要由该客户来实施验收测试（Acceptance Testing），以便确认其所有需求是否都已得到满足。由于软件系统的复杂性，在实际工作中，验收测试可能会持续到用户实际使用该软件之后的相当长的一段时间。

如果一个软件是作为产品被许多客户使用的，不可能也没必要由每个客户进行验收测试。绝大多数软件开发商都使用被称为 α （Alpha）测试和 β （Beta）测试的过程，来发现那些看起来只有最终用户才能发现的错误。

α 测试由用户在开发者的场所进行，并且在开发者的指导下进行测试。开发者负责记录发现的错误和使用中遇到的问题。也就是说， α 测试是在“受控的”环境中进行的。

β 测试是在一个或多个用户的现场由该软件的最终用户实施的，开发者通常不在现场，用户负责记录发现的错误和使用中遇到的问题并把这些问题报告给开发者。也就是说， β 测试是在“非受控的”环境中进行的。

经过确认测试之后的软件通常就可以交付使用了。

6. 系统测试

系统测试的对象是完整的、集成的计算机系统，系统测试的目的是在真实系统工作环境下，验证完整的软件配置项能否和系统正确连接，并满足系统/子系统设计文档和软件开发合同规定的要求。系统测试的技术依据是用户需求或开发合同，除应满足一般测试的准入条件外，在进行系统测试前，还应确认被测系统的所有配置项已通过测试，对需要固化运行的软件还应提供固件。

一般来说,系统测试的主要内容包括功能测试、健壮性测试、性能测试、用户界面测试、安全性测试、安装与反安装测试等,其中,最重要的工作是进行功能测试与性能测试。功能测试主要采用黑盒测试方法,性能测试主要验证软件系统在承担一定负载的情况下所表现出来的特性是否符合客户的需要,主要指标有响应时间、吞吐量、并发用户数和资源利用率等。

7. 回归测试

回归测试的目的是测试软件变更之后,变更部分的正确性和对变更需求的符合性,以及软件原有的、正确的功能、性能和其他规定的要求的不损害性。回归测试的对象主要包括以下4个方面:

- 未通过软件单元测试的软件,在变更之后,应对其进行单元测试。
- 未通过配置项测试的软件,在变更之后,首先应对变更的软件单元进行测试,然后再进行相关的集成测试和配置项测试。
- 未通过系统测试的软件,在变更之后,首先应对变更的软件单元进行测试,然后再进行相关的集成测试、配置项测试和系统测试。
- 因其他原因进行变更之后的软件单元,也首先应对变更的软件单元进行测试,然后再进行相关的软件测试。

8. 白盒测试

白盒测试,又称结构测试,主要用于单元测试阶段。它的前提是把程序看成装在一个透明的白盒子里,测试者完全知道程序的结构和处理算法。这种方法按照程序内部逻辑设计测试用例,检测程序中的主要执行通路是否都能按预定要求正确工作。

白盒测试常用的技术是逻辑覆盖,即考查用测试数据运行被测程序时对程序逻辑的覆盖程度。主要的覆盖标准有6种:语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖。

(1) 语句覆盖

语句覆盖是指选择足够多的测试用例,使得运行这些测试用例时,被测程序的每个语句至少执行一次。

很显然,语句覆盖是一种很弱的覆盖标准。

(2) 判定覆盖

判定覆盖又称分支覆盖,它的含义是,不仅每个语句至少执行一次,而且每个判定的每种可能的结果(分支)都至少执行一次。

判定覆盖比语句覆盖强,但对程序逻辑的覆盖程度仍然不高。

(3) 条件覆盖

条件覆盖的含义是,不仅每个语句至少执行一次,而且使判定表达式中的每个条件都取到各种可能的结果。

条件覆盖不一定包含判定覆盖,判定覆盖也不一定包含条件覆盖。

(4) 判定/条件覆盖

同时满足判定覆盖和条件覆盖的逻辑覆盖称为判定/条件覆盖。它的含义是,选取足够的测试用例,使得判定表达式中每个条件的所有可能结果至少出现一次,而且每个判定本身的所有可能结果也至少出现一次。

(5) 条件组合覆盖

条件组合覆盖的含义是,选取足够的测试用例,使得每个判定表达式中条件结果的所有可能组合至少出现一次。

显然,满足条件组合覆盖的测试用例,也一定满足判定/条件覆盖。因此,条件组合覆盖是上述5种覆盖标准中最强的一种。然而,条件组合覆盖还不能保证程序中所有可能的路径都至少经过一次。

(6) 路径覆盖

路径覆盖的含义是,选取足够的测试用例,使得程序的每条可能执行到的路径都至少经过一次(如果程序中有环路,则要求每条环路径至少经过一次)。

路径覆盖实际上考虑了程序中各种判定结果的所有可能组合,因此是一种较强的覆盖标准。但路径覆盖并未考虑判定中的条件结果的组合,并不能代替条件覆盖和条件组合覆盖。

9. 黑盒测试

黑盒测试,又称功能测试,主要用于集成测试和确认测试阶段。它把软件看做一个不透明的黑盒子,完全不考虑(或不了解)软件的内部结构和处理算法,它只检查软件功能是否能按照软件需求说明书的要求正常使用,软件是否能适当地接收输入数据并产生正确的输出信息,软件运行过程中能否保持外部信息(如文件和数据库)的完整性等。

常用的黑盒测试技术包括等价类划分、边值分析、错误推测和因果图等。

(1) 等价类划分

在设计测试用例时,等价类划分是用得最多的一种黑盒测试方法。所谓等价类,就是某个输入域的集合,对于一个等价类中的输入值来说,它们揭示程序中错误的作用是等效的。也就是说,如果等价类中的一个输入数据能检测出一个错误,那么等价类中的其他输入数据也能检测出同一个错误;反之,如果等价类中的一个输入数据不能检测出某个错误,那么等价类中的其他输入数据也不能检测出这一错误(除非这个等价类的某个子集还属于另一等价类)。

如果一个等价类内的数据是符合(软件需求说明书)要求的、合理的数据,则称这个等价类为有效等价类。有效等价类主要用来检验软件是否实现了软件需求说明书中规定的功能。

如果一个等价类内的数据是不符合(软件需求说明书)要求的、不合理或非法的数据,则称这个等价类为无效等价类。无效等价类主要用来检验软件的容错性。

黑盒测试中,利用等价类划分方法设计测试用例的步骤如下:

- ① 根据软件的功能说明,对每一个输入条件确定若干个有效等价类和若干个无

效等价类，并为每个有效等价类和无效等价类编号。

② 设计一个测试用例，使其覆盖尽可能多的尚未被覆盖的有效等价类。重复这一步，直至所有的有效等价类均被覆盖。

③ 设计一个测试用例，使其覆盖一个尚未被覆盖的无效等价类。重复这一步，直至所有的无效等价类均被覆盖。

应当特别注意，无效等价类用来测试非正常的输入数据，因此每个无效等价类都有可能查出软件中的错误，所以要为每个无效等价类设计一个测试用例。

(2) 边值分析

经验表明，软件在处理边界情况时最容易出错。设计一些测试用例，使软件恰好运行在边界附近，暴露出软件错误的可能性会更大一些。

通常，每一个等价类的边界，都应该着重测试，选取的测试数据应该恰好等于、稍小于或稍大于边界值。

将等价类划分法和边值分析法结合使用，更有可能发现软件中的错误。

(3) 错误推测

使用等价类划分和边值分析技术，有助于设计出具有代表性的、容易暴露软件错误的测试方案。但是，不同类型不同特征的软件通常又有一些特殊的容易出错的地方。错误推测法主要依靠测试人员的经验和直觉，从各种可能的测试方案中选出一些最可能引起程序出错的方案。

(4) 因果图

因果图法是根据输入条件与输出结果之间的因果关系来设计测试用例的，它首先检查输入条件的各种组合情况，并找出输出结果对输入条件的依赖关系，然后为每种输出条件的组合设计测试用例。

10. 自动化测试

自动化测试是把以人为驱动的测试行为转化为机器执行的一种过程。通常，在设计了测试用例并通过评审之后，由测试人员根据测试用例中描述的规程一步步执行测试，得到实际结果与期望结果的比较。在此过程中，为了节省人力、时间或硬件资源，提高测试效率，便引入了自动化测试的概念。

自动化测试在多类项目中都可采用，尤其是对于采用迭代型开发模型的项目。因为迭代型开发模型的系统不是一次性开发出来的，要经过多轮迭代，每一轮迭代都要进行测试。自动化测试只需要写好测试脚本再重复多次执行，即可完成测试任务，能极大的提升了测试效率。

11. 软件评审

评审是对软件元素或者项目状态的一种评估手段，以确定其是否与计划的结果保持一致，并使其得到改进。

软件评审又称技术评审，可以分为以下 3 种类型：

(1) 评审

评审是指一次正式的会议，在会议上向用户或其他项目干系人介绍一个或一组工作产品，以征求对方的意见和批准。

(2) 检查

检查是一种正式的评估方法，将由非制作者本人的个人或小组详细检查工作产品，以验证是否有错误、是否违反开发标准、是否存在其他问题。

(3) 走查

走查是一个评审过程，由某个开发人员领导一个或多个开发团队成员对他（或她）的工作产品进行检查，由其他成员针对技术、风格、可能的错误、是否违反开发标准和其他问题提出意见。

在实际工作中，技术评审可以分为正式评审和非正式评审。正式评审是指通过召开评审会的形式，组织多个专家，将工作产品涉及的人员集合在一起，并定义好评审人员的角色和职责，对工作产品进行正规的会议评审。而非正式评审并没有这种严格的组织形式，一般也不需要人员集合在一起评审，而是通过电子邮件、文件汇签，甚至是网络聊天等多种形式对工作产品进行评审。

非正式评审对于获得分散而随机的反馈是有效的，但非正式评审是非系统化的、不彻底的，它在实施过程中具有不一致性。非正式评审可以根据个人爱好的方式进行评审，而正式评审则需要遵循预先定义好的一系列步骤和过程。正式评审的内容需要记录在案，包括确定材料，评审小组对工作产品是否完整或者是否需要进一步工作的判定，以及对所发现的错误和所提出的问题的总结。

7.1.7 软件开发环境与工具

在整个软件开发过程中，要使用到一系列的工具，工具的使用让开发效率以及软件质量得到极大的提升。在软件设计师考试中，会涉及软件开发工具的基础内容，主要是识别某种工具是应用于开发的哪一个阶段。下面将按阶段来说明常见的软件开发工具。

1. 需求分析工具

需求分析工具主要了解两种：PowerDesigner 和 Rational Rose。

(1) PowerDesigner 简介

PowerDesigner 是一种系统分析设计 CASE 工具，同时也可以作为数据库设计工具进行使用。它是由美国原 Power Soft 公司（已于 1996 年并入 Sybase 公司）推出的发展版本。PowerDesigner 是一个集成的系统设计工具集，可以用于创建 UML 图、数据库的概念模型和物理模型并能针对某一程序开发语言（平台）快速地生成相应的对象和组件。

(2) Rational Rose 简介

Rational Rose 是运用最为广泛的 UML 建模 CSAAE 工具之一。它支持除对象图外所有的 UML 模型图。

2. 数据库设计工具

数据库设计工具主要是两种：PowerDesigner 和 ERwin，其中 PowerDesigner 在上

面已经提到。

ERwin 是美国 CA (Computer Associates International Inc.) 公司的拳头产品之一。ERwin 为企业数据需求和关系数据库设计提供了一个定义、管理和实现的平台。ERwin 将 ER 绘图工具与基于 Windows 的图形用户接口相结合, 这样的特点使得用户能够快速构建企业的数据解决方案。ERwin 支持概念模型和物理模型之间的无损转换。并能够根据物理模型生成具体的数据库或是数据库生成脚本, 并支持基于 XML 格式的数据模型输出。此外, ERwin 还与 CA 另外一个需求分析实施用的产品 BPwin 进行无缝集成, 用于帮助设计、生成和维护高质量的应用程序, 加速基于组件应用程序的开发。

3. 项目管理工具

(1) Microsoft Project

Microsoft Project 跟 Microsoft 的其他办公软件一样, 是一个功能强大、界面友好、易于使用的工具软件。就目前来说, Microsoft Project 在全球占据了近 2/3 的项目管理软件市场。它可以用来控制简单或者复杂的项目, 安排和追踪项目中所有的活动, 以及这些活动所涉及的资源、资金利用情况, 使我们对整个项目进展一目了然, 在一定程度上减轻项目的管理工作量, 同时对于提高项目的质量起到了促进的作用。

Microsoft Project 是一个综合性的工具, 可以进行多种管理, 如项目范围管理、项目时间管理、项目成本管理、人力资源管理等。

(2) Visual Source Safe (VSS)

Microsoft Visual Source Safe (以下简称 VSS) 用于项目配置管理, 它提供了完善的版本和配置管理功能, 以及安全保护和跟踪检查功能。VSS 将所有的项目源文件(包括各种文件类型)以特有的方式存入数据库。用户可以根据需要随时快速有效地共享文件。文件一旦被添加进 VSS, 它的每次改动都会被记录下来, 用户可以恢复文件的早期版本, 项目组的其他成员也可以看到有关文档最新版本, 并对它们进行修改。使用 VSS 来管理项目, 使得项目组成员间的沟通与合作更简易、直观。

(3) CVS

CVS (Concurrent Versions System) 与 VSS 一样, 也是用于项目配置管理。它是一个免费、开放、灵活、支持分支/合并、并行开发和集成, 便于团队协作大规模开发的版本控制工具。CVS 可以说是目前最为流行和首推的版本控制系统, 它同时也是一个开源项目。

4. 程序设计工具

(1) VS.NET 简介

Visual Studio .NET 是微软所出的 Windows 下的集成开发环境, 可以开发 Window DNA 和 IIS Web Application。包括多种语言及相应的项目工具, 如配置管理等。Visual Studio 系列都具备良好的图形界面, 并且提供了许多简化软件开发的功能。

(2) JBuilder 简介

JBuilder 是 Borland 公司为 Java 语言开发的一种 IDE 开发工具。

(3) Eclipse 简介

Eclipse 也是针对 Java 语言的开发工具, 较之 JBuilder, 它功能不如 JBuilder 强大,

但其优势也是非常明显的，Eclipse 是免费的，同时体积小，所以目前 Eclipse 的使用者也是非常多的。

(4) Delphi

Delphi 是美国 Borland 公司开发的工作在 Windows 平台下的开发工具，它的前身是在 DOS 下的产品 Borland Pascal。从产品名称上就可以知道，Borland Pascal 使用的是 Pascal 语言。从 Borland Pascal 5.0 版本开始，Borland 公司在传统 Pascal 的基础上加入了面向对象的功能。Delphi 是一个集成开发环境（IDE），使用的是由传统 Pascal 语言发展而来的 Object Pascal 语言。它在本质上是一个代码编辑器而不是一种语言，但由于 Delphi 几乎是市场上唯一一个使用 Pascal 语言的产品，因此有的时候 Delphi 也成为了人们称呼 Object Pascal 的代名词。Borland 公司已经把 Object Pascal 语言改称为 Delphi 语言。

7.1.8 软件过程改进

软件过程改进/过程改进（Software Process Improvement, SPI）帮助软件企业对其软件（制作）过程的改变（进）进行计划、（措施）制定及实施。它的实施对象就是软件企业的软件过程，也就是软件产品的生产过程，当然也包括软件维护之类的维护过程，而对于其他的过程并不关注。

1. ISO 9000 标准族

ISO 9000 标准族（如表 7-17 所示）是国际标准化组织中质量管理和质量保证技术委员会制订的一系列标准，现在共包括 20 个标准。

表 7-17 ISO 9000 标准族

① 质量术语标准			
ISO 8402			
② 标准选用与实施指南	③ 质量保证标准		④ 质量管理标准
ISO 9000 -1 选择与使用 -2 实施 -3 计算机软件 -4 可信性大纲	ISO 9001 设计、开发、生产、安装和服务 ISO 9002 生产、安装和服务 ISO 9003 最终检验和试验		ISO 9004 -1 指南 -2 服务指南 -3 流程性材料 -4 质量改进
⑤ 支持性技术标准			
ISO 10005 质量计划 ISO 10007 技术状态	ISO 10011 -1 审核 -2 审核员 -3 审核管理	ISO 10012 -1 测量设备 -2 测量过程	ISO 10013 质量手册

2. 能力成熟度模型（CMM）

CMM 模型是 CMU/SEI（卡内基梅隆大学/软件工程研究所）提出的软件过程成熟度模型，它描述和分析了软件过程能力的发展程度，确立了一个软件过程成熟程度的分级标准，它将软件过程成熟度分为 5 级，如表 7-18 所示。

表 7-18 CMM 过程成熟改进模型

级 别	特 点	关键过程域
初始级	软件过程的特点是无秩序的, 有时甚至是混乱的, 成功依赖于个人	
可重复级	已经建立了基本的项目管理过程, 可用于对成本、进度和功能特性进行跟踪。对类似的应用项目, 有章可循并能重复以往所取得的成功	需求管理, 软件项目策划, 软件项目跟踪与监督, 软件子合同管理, 软件质量保证, 软件配置管理
已定义级	用于管理的和工程的软件过程均已文档化、标准化, 并形成整个软件组织的标准软件过程	组织过程定义, 组织过程焦点, 培训大纲, 集成软件管理, 软件产品工程, 组织协调, 同行专家评审
已管理级	软件过程和产品质量有详细的度量标准	定量的过程管理, 软件质量管理
优化级	通过对来自过程、新概念和新技术等方面的各种有用信息的定量分析, 能够不断地、持续地进行过程改进	缺陷预防, 技术变更管理, 过程变更管理

在 CMM 内部, 每个成熟度等级 (除第一级外) 规定了不同的关键过程域 (KPA), 一个软件组织如果希望达到某一个成熟度级别, 就必须完全满足关键过程域所规定的要求, 即满足关键过程域的目标。而公共特性是对每个 KPA 的所有关键实施按照它们的属性进行分组。无论哪个 KPA, 它们的关键实施都统一按 5 个公共属性进行组织, 分别是执行约定、执行能力、实施活动、度量和分析、实施验证。

CMM 模型的最新版本是 CMMI, 它涉及面比 CMM 更广, 覆盖软件工程、系统工程、集成产品开发和系统采购, 如图 7-6 所示。它也将组织的成熟度分成 5 个级别, 但其中规定的 KPA 与 CMM 相比有所变化, 如表 7-19 所示。

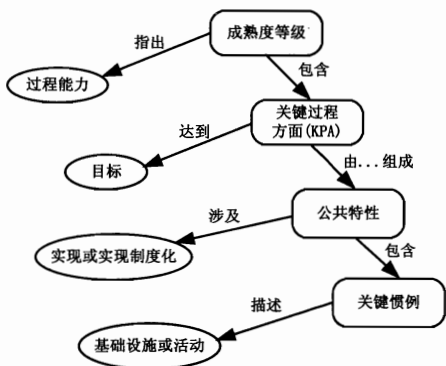


图 7-6 CMM 内部结构示意图

表 7-19 CMM 过程成熟度模型

级 别	关键过程域
初始级	
可重复级	需求管理 项目策划 项目监督和控制 供方协定管理 测量与分析 过程 and 产品质量保证 配置管理 组织培训
已定义级	需求开发 技术解决 产品集成 验证 确认 组织过程焦点 组织过程定义 组织培训 集成项目管理 风险管理 决策分析和决定
已管理级	组织过程性能 定量项目管理
优化级	组织变革和部署 原因分析和决定

3. CMMI

每一种 CMMI 模型都有两种表示法, 即阶段式和连续式。连续式表示法强调的是单个过程域的能力, 从过程域的角度考查基线和度量结果的改善, 其关键技术语是“能力”; 而阶段式表示法强调的是组织的成熟度, 从过程域集合的角度考查整个组织的

过程成熟度阶段，其关键术语是“成熟度”。阶段式模型也把组织分为以下5个不同的级别。

- 初始级：代表了以不可预测结果为特征的过程成熟度，过程处于无序状态，成功主要取决于团队的技能。
- 已管理级：代表了以可重复项目执行为特征的过程成熟度。
- 严格定义级：代表了以组织内改进项目执行为特征的过程成熟度。
- 定量管理级：代表了以改进组织性能为特征的过程成熟度。
- 优化级：代表了以可快速进行重新配置的组织性能和定量的、持续的过程改进为特征的过程成熟度。

7.2 典型试题分析

试题 1

需求工程帮助软件工程师更好地理解要解决的问题。下列开发过程中的活动，不属于需求工程范畴的是__（1）__。

- (1) A. 理解客户需要什么，分析要求，评估可行性
B. 与客户协商合理的解决方案，无歧义地详细说明方案
C. 向客户展现系统的初步设计方案，并得到客户的认可
D. 管理需求以至将这些需求转化为可运行的系统

试题分析

需求工程为以下工作提供了良好的机制：理解客户需要什么，分析要求，评估可行性，协商合理的解决方案，无歧义地详细说明方案，确认规格说明，管理需求以至将这些需求转化为可运行的系统。需求工程并不关心采用何种设计方案解决问题。

试题答案

- (1) C

试题 2

面向团队的需求收集方法能够鼓励合作，为解决方案的各个要素提供建议，协商不同的方法，以及说明初步的需求方案。下列关于面向团队的需求收集方法叙述，不恰当的是__（2）__。

- (2) A. 举行团队需求收集会议，会议由软件工程师、客户和其他利益相关者共同举办和参加
B. 拟定一个会议议程，与会者围绕需求要点，畅所欲言
C. 会议提倡自由发言，不需要特意控制会议的进度
D. 会议目的是为了识别问题，提出解决方案的要点，初步刻画解决方案中的需求问题

试题分析

在进行面向团队的需求分析时，通常会举行团队需求收集会议，会议由软件工程师、客户和其他利益相关者共同举办和参加；为会议拟订一个会议议程，既要涵盖所有的重要点，又要鼓励思维的自由交流；会议由一位主持人控制会议进度，并保证会议主题不被偏离；会议目的是为了识别问题，提出解决方案的要点，初步刻画解决方案中的需求问题。

试题答案

(2) C

试题 3

在关于逆向工程（Reverse Engineering）的描述中，正确的是（3）。

- (3) A. 从已经安装的软件中提取设计规范，用以进行软件开发
B. 按照“输出→处理→输入”的顺序设计软件
C. 用硬件来实现软件的功能
D. 根据软件处理的对象来选择开发语言和开发工具

试题分析

逆向工程（Reverse Engineering）术语源于硬件制造业，相互竞争的公司为了了解对方设计和制造工艺的机密，在得不到设计和制造说明书的情况下，通过拆卸实物获得信息，软件的逆向工程也基本类似，不过，通常“解剖”的不仅是竞争对手的程序，而且还包括本公司多年前的产品。软件的逆向工程是分析程序，力图在比源代码更高抽象层次上建立程序的表示过程，逆向工程是设计的恢复过程。

所以本题的正确答案是 A。关于逆向工程的详细说明，请参看“7.1.5 软件开发方法”中逆向工程相关内容。

试题答案

(3) A

试题 4

在选择开发方法时，不适合使用原型法的情况是（4）。

- (4) A. 用户需求模糊不清 B. 系统设计方案难以确定
C. 系统使用范围变化很大 D. 用户的数据资源缺乏组织和管理

试题分析

由于原型法开发需要适当的快速开发工具，需要用户密切配合，所以下列情况不适合使用原型法：

- ①缺乏适用的原型开发工具。
②用户不参与、不积极配合开发过程。

③用户的数据资源缺乏组织和管理。

④用户的软件资源缺乏组织和管理

所以本题应选 D，关于原型法的详细说明请参看“7.1.4 软件开发方法”中开发模型相关内容。

试题答案

(4) D

试题 5

某软件公司欲开发一个图像处理系统，在项目初期，开发人员对算法的效率、操作系统的兼容性和人机交互的形式等情况并不确定。在这种情况下，采用__(5)__方法比较合适。

(5) A. 瀑布式 B. 形式化 C. 协同开发 D. 快速原型

试题分析

很多时候，客户提出了软件的一些基本功能，但是没有详细定义输入、处理和输出需求；另一种情况下，开发人员可能对算法的效率、操作系统的兼容性和人机交互的形式等情况不确定。在这种情况下，原型开发方法是最好的解决方法。

试题答案

(5) D

试题 6

螺旋模型是一种演进式的软件过程模型，结合了原型开发方法的系统性和瀑布模型可控性特点。它有两个显著特点，一是采用__(6)__的方式逐步加深系统定义和实现的深度，降低风险；二是确定一系列__(7)__，确保项目开发过程中的相关利益者都支持可行的和令人满意的系统解决方案。

(6) A. 逐步交付 B. 顺序 C. 循环 D. 增量

(7) A. 实现方案 B. 设计方案 C. 关键点 D. 里程碑

试题分析

螺旋模型是一种演化式的软件过程模型，结合了原型开发方法的系统性和瀑布模型的可控性特点。它有两个显著特点，一是采用循环的方式逐步加深系统定义和实现的深度，同时降低风险；二是确定一系列里程碑，确保项目开发过程中的相关利益者都支持可行的和令人满意的系统解决方案。

试题答案

(6) C

(7) D

试题 7

极限编程是一种重要的敏捷开发方法，包含策划、设计、编码和测试 4 个框架活

动的规则和实践。下面关于极限编程的叙述，错误的是__（8）__。

- （8）A. 极限编程中使用的重要技术是重构，既包括设计技术的重构，也包括构建技术的重构
- B. 极限编程提倡在基本设计完成后，立即进行编码实现，并进行测试
- C. 极限编程活动中的关键概念之一是“结对编程”，推荐两个人面对同一台计算机共同开发代码
- D. 极限编程过程中建立的单元测试应当使用一个可以自动实施的框架，支持代码修改后即时的回归测试策略

试题分析

极限编程是一种重要的敏捷开发方法，包含规划、设计、编码和测试4个框架活动的规则和实践。极限编程中使用的重要技术是重构，既包括设计技术的重构，也包括构建技术的重构；极限编程提倡在基本设计完成后，团队不应该直接开始编码，而是开发一系列用于检测本次发布的包括所有故事（story）的单元测试；极限编程活动中的关键概念之一是“结对编程”，推荐两个人面对同一台计算机共同开发代码；极限编程过程中建立的单元测试应当使用一个可以自动实施的框架，支持代码修改后即时的回归测试策略。

试题答案

（8）B

试题 8

在结构化设计中，__（9）__描述了模块的输入输出关系、处理内容、模块的内部数据和模块的调用关系，是系统设计的重要成果，也是系统实施阶段编制程序设计任务书和进行程序设计的出发点和依据。

- （9）A. 系统流程图 B. IPO 图 C. HIPO 图 D. 模块结构图

试题分析

系统流程图是表达系统执行过程的描述工具；IPO图描述了模块的输入输出关系、处理内容、模块的内部数据和模块的调用关系；HIPO图描述了系统自顶向下的模块关系；模块结构图描述了系统的模块结构以及模块间的关系，同时也描述了模块之间的控制关系。

试题答案

（9）B

试题 9

下列聚合类型中内聚程度最高的是__（10）__，下列耦合类型中耦合程度最低的是__（11）__。

- （10）A. 偶然内聚 B. 时间内聚 C. 功能内聚 D. 过程内聚

- (11) A. 数据耦合 B. 控制耦合 C. 公共耦合 D. 内容耦合

试题分析

模块的独立程度有两个定性标准度量：内聚和耦合。内聚衡量模块内部各元素结合的紧密程度；耦合度量不同模块间互相依赖的程度。按照内聚程度从低到高排列，内聚包括偶然内聚、逻辑内聚、瞬时内聚、过程内聚、通信内聚、顺序内聚和功能内聚，其中功能内聚的内聚程度最高。按照耦合程度从低到高的排列，耦合包括非直接耦合、数据耦合、标记耦合、控制耦合、外部耦合、公共耦合和内容耦合，其中非直接耦合的耦合程度最低。

试题答案

- (10) C (11) A

试题 10

____(12)____方法以原型开发思想为基础,采用迭代增量式开发,发行版本小型化,比较适合需求变化较大或者开发前期对需求不是很清晰的项目。

- (12) A. 信息工程 B. 结构化
C. 面向对象 D. 敏捷

试题分析

本题考查开发模型相关知识。迭代、增量、发行版本小型化，以及需求变化较大等特点均与敏捷方法一致，所以本题应选 D。关于敏捷方法更详细的说明，请参看“7.1.4 软件开发方法”。

试题答案

- (12) D

试题 11

系统分析是信息系统开发重要的阶段。系统分析的困难不包括 (13) 。

- (13) A. 问题空间的理解 B. 系统建模工具的复杂性
C. 人与人之间的沟通 D. 环境的不断变化

试题分析

系统分析的困难主要来自 3 个方面：问题空间的理解、人与人之间的沟通和环境的不断变化。问题空间的理解，主要是指系统分析员缺乏足够的对象系统的专业知识，从而无法获取准确和全面的系统需求；人与人之间的沟通主要是指系统分析员和用户之间的沟通比较困难，用户虽然精通业务但往往不善于把业务过程明确地表达出来；环境的不断变化指的是信息系统生存在不断变化的环境中，环境对它不断提出新的要求。

试题答案

- (13) B

试题 12

____(14)____是一种最常用的结构化分析工具，它从数据传递和加工的角度，以图形的方式刻画系统内数据的运行情况。通常使用____(15)____作为该工具的补充说明。

(14) A. 数据流图 B. 数据字典 C. E-R 图 D. 判定表

(15) A. 数据流图 B. 数据字典 C. E-R 图 D. 判定表

试题分析

数据流图是一种最常用的结构化分析工具，它从数据传递和加工的角度，以图形的方式刻画系统内数据的运行情况。数据流图是一种能全面描述信息系统逻辑模型的主要工具，它可以用少数集中符号综合地反映出信息在系统中的流动、处理和存储的情况。

通常使用数据字典对数据流图加以补充说明。数据字典是以特定格式记录下来的、对系统的数据流图中各个基本要素的内容和特征所做的完整的定义和说明。

试题答案

(14) A (15) B

试题 13

用户界面设计中，设计原则不正确的是____(16)____。

(16) A. 为用户提供更多的信息和功能 B. 减少用户的记忆负担
C. 保持界面一致性 D. 置用户于控制之下

试题分析

用户界面设计是一项复杂的任务，它必须遵循一些“良好设计”的指导原则，下面是一些关键的用户界面设计原则：

① 用户控制。人机界面设计首先要确立用户类型。划分类型可以从不同的角度，视实际情况而定。确定类型后要针对其特点预测他们对不同界面的反应。这就要从多方面设计分析。用户应当感觉系统的运行在自己的控制之下。在图形界面或基于 Web 的界面中，用户指导程序的每一步执行；即使在程序进行某些处理或用户等待输出结果时，用户同样保持对控制的敏感度。

② 信息最小量。人机界面设计要尽量减少用户记忆负担，采用有助于记忆的设计方案。

③ 帮助和提示。要对用户的操作命令做出反应，帮助用户处理问题。系统要设计有恢复出错现场的能力，在系统内部处理工作要有提示，尽量把主动权让给用户。

④ 媒体最佳组合。多媒体界面的成功并不在于仅向用户提供丰富的媒体，而应在相关理论指导下，注意处理好各种媒体间的关系，恰当选用。

⑤ 界面一致性。一致性要求用户界面遵循标准和常规的方式，让用户处在一个熟悉的和可预见的环境之中，这主要体现在命名、编码、缩写、布局，以及菜单、按

钮和键盘功能在内的控制使用等。

⑥ 界面容错性。一个好的界面应该以一种宽容的态度允许用户进行实验和出错，使用户在出现错误时能够方便地从错误中恢复。

⑦ 界面美观性。界面美观性是视觉上的吸引力，主要体现在具有平衡和对称性、合适的色彩、各元素具有合理的对齐方式和间隔、相关元素适当分组、使用户可以方便地找到要操作的元素等。

⑧ 界面可适应性。界面可适应性是指用户界面应该根据用户的个性要求及其对界面的熟知程度而改变，即满足定制化和个性化的要求。所谓定制化，是在程序中声明用户的熟知程度，用户界面可以根据熟知程度改变外观和行为；所谓个性化，是使用户按照自己的习惯和爱好设置用户界面元素。

试题答案

(16) A

试题 14

以下关于需求管理的叙述中，正确的是___(17)___。

- (17) 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

试题分析

需求管理是一个对系统需求变更、了解和控制的过程。需求管理过程与需求开发过程相互关联，当初始需求导出的同时就启动了需求管理计划，一旦形成了需求文档的初稿，需求管理活动就开始了。

关于需求管理过程域内的原则和策略，可以参考：

① 需求管理的关键过程领域不涉及收集和分析项目需求，而是假定已收集了软件需求，或者已由更高一级的系统给定了需求。

② 开发人员在向客户以及有关部门承诺某些需求之前，应该确认需求和约束条件、风险、偶然因素、假定条件等。

③ 关键处理领域同样建议通过版本控制和变更控制来管理需求文档。

关于需求管理更详细的内容请参看“7.1.3 需求管理”。

试题答案

(17) A

试题 15

软件设计阶段的度量考虑了架构层次、构件层次和界面设计层次的问题，其中构

件层次的设计度量集中于软件构件的内部特性，（18）不属于构件层次设计度量。

- (18) A. 模块内聚 B. 模块耦合 C. 布局恰当性 D. 复杂度

试题分析

构件层次设计度量主要考虑模块的独立性问题，这包括内聚、耦合和复杂度。

试题答案

- (18) C

试题 16

复用应该是每个软件过程都不可缺少的部分。下列制品，不属于复用候选的软件制品是（19）。

- (19) A. 项目计划 B. 体系结构 C. 用户界面 D. 用户需求

试题分析

在软件复用的过程中，有高级的复用，如项目计划和体系结构的复用；也有低层的复用，如代码复用和界面复用等。显然，用户需求是用户自己的对软件的一些要求和目标，这是不能复用的。

试题答案

- (19) D

试题 17

面向功能的软件度量使用软件所提供功能的测量作为规范化值。下面关于面向功能的软件度量方法的描述，不正确的是（20）。

- (20) A. 面向功能度量是由 Albrecht 首先提出来的，他建议采用基于功能点的测量
B. 功能点是基于软件信息领域的可计算的（直接的）测量及软件复杂性的评估而导出的
C. “功能”不能直接测量，所以必须通过其他直接的测量来导出
D. “用户输入数”和“用户输出数”不属于面向功能的软件度量需要考虑的因素

试题分析

面向功能的软件度量是对软件和软件开发过程的间接度量，注意力集中于程序的功能性和实用性，而不是对 LOC 计数。该度量是由 Albrecht 首先提出来的。他提出了一种叫做功能点方法的生产率度量法，该方法利用有关软件数据域的一些计数度量和软件复杂性估计的经验关系式，导出功能点 FPs (Function Points)。

功能点通过填写图表格来计算。首先要确定 5 个数据域的特征，并在表格中相应位置给出计数。数据域的值以如下方式定义：

- ① 用户输入数：每个用户输入应是面向不同应用的输入数据，对它们都要进行

计数。输入数据应区别于查询数据，它们应分别计数。

② 用户输出数：各个用户输出是为用户提供的面向应用的输出信息，它们均应计数。在这里的“输出”是指报告、屏幕信息、错误信息等，在报告中的各个数据项不应再分别计数。

③ 用户查询数：查询是一种联机输入，它引发软件以联机方式产生某种即时响应。每一个不同的查询都要计数。

④ 文件数：每一个逻辑主文件都应计数。这里的逻辑主文件，是指逻辑上的一组数据，它们可以是一个大的数据库的一部分，也可以是一个单独的文件。

⑤ 外部接口数：对所有使用来将信息传送到另一个系统中的接口（即磁带、磁盘和可读写光盘上的数据文件）均应计数。

一旦收集到上述数据，就可以计算出与每一个计数相关的加权复杂性值。

试题答案

(20) D

试题 18

某个系统在开发时，用户已经定义了软件的一组一般性目标，但不能标识出详细的输入、处理及输出需求；开发者也可能暂时不能确定算法的有效性、操作系统的适应性或人机交互的形式。在这种情况下，采用__（21）__开发最恰当。

(21) A. 瀑布模型 B. 迭代模型 C. 原型模型 D. 螺旋模型

试题分析

从试题描述来看，用户对软件的需求并不明确，而且开发人员对算法也不确定，因此，适合使用原型模型。通过开始设计的原型，来明确用户的需求，来探索算法的有效性。通过原型的运行，来分析操作系统的适应性，以及分析人机交互的形式。

试题答案

(21) C

试题 19

需求分析是一种软件工程活动，它在系统级软件分配和软件设计间起到桥梁的作用。需求分析使得系统工程师能够刻画出软件的__（22）__、指明软件和其他系统元素的接口、并建立软件必须满足的约束。需求分析是发现、求精、建模和规约的过程。包括详细地精化由系统工程师建立并在软件项目计划中精化的软件范围，创建所需数据、信息和__（23）__，以及操作行为的模型，此外还有分析可选择的解决方案，并将它们分配到各软件元素中去。

(22) A. 功能和性能 B. 数据和操作 C. 实体和对象 D. 操作和对象

(23) A. 事件流 B. 消息流 C. 对象流 D. 控制流

试题分析

需求分析使得系统工程师能够刻画出软件的功能需求（明确所开发的软件必须具

备什么样的功能)、性能需求(明确待开发的软件的技术性能指标)、环境需求(明确软件运行时所需要的软、硬件的要求)、用户界面需求(明确人机交互方式、输入/输出数据格式)。需求分析要指明软件和其他系统元素的接口、并建立软件必须满足的约束。需求分析是发现、求精、建模和规约的过程。包括详细地精化由系统工程师建立并在软件项目计划中精化的软件范围,创建所需数据、信息和控制流,以及操作行为的模型。

试题答案

(22) A

(23) D

试题 20

黑盒测试注重于测试软件的功能性需求,主要用于软件的后期测试。__(24)__不能用黑盒测试检查出来。

(24) A. 功能不对或遗漏错误

B. 界面错误

C. 外部数据库访问错误

D. 程序控制结构错误

试题分析

黑盒测试注重于测试软件的功能性需求,把被测试对象看成一个黑盒子,测试人员完全不考虑程序的内部控制结构和处理过程,只在软件的接口(界面)处进行测试,依据需求规格说明书,检查程序是否满足功能要求。

试题答案

(24) D

试题 21

企业信息系统往往是一个具有业务复杂性和技术复杂性的大系统,针对其建设,系统分析首先要进行的工作是__(25)__.系统开发的目的是__(26)__。

(25) A. 获得当前系统的物理模型

B. 抽象出当前系统的逻辑模型

C. 建立目标系统的逻辑模型

D. 建立目标系统的物理模型

(26) A. 获得当前系统的物理模型

B. 抽象出当前系统的逻辑模型

C. 建立目标系统的逻辑模型

D. 建立目标系统的物理模型

试题分析

通常,软件开发项目是要实现目标系统的物理模型,即确定待开发软件系统的系统元素,并将功能和数据结构分配到这些系统元素中,它是软件实现的基础。但是,目标系统的具体物理模型是由它的逻辑模型经实例化(即具体到某个业务领域)得到的。与物理模型不同,逻辑模型忽视实现机制与细节,只描述系统要完成的功能和要处理的数据。作为目标系统的参考,需求分析的任务就是借助于当前系统的逻辑模型导出目标系统的逻辑模型,解决目标系统“做什么”的问题。

结合现有系统(当前)分析,进行新系统设计的过程如图 7-7 所示。



图 7-7 现有系统的研究和分析过程

① 获得当前系统的物理模型。当前系统可能是需要改进的某个已在计算机运行的数据处理系统，也可能是一个人工的数据处理过程。在这一步首先分析、理解当前系统是如何运行的，了解当前系统的组织机构、输入输出、资源利用情况和日常数据处理过程，并用一个具体模型来反映自己对当前系统的理解。这一模型应客观地反映现实世界的实际情况。

② 抽象出当前系统的逻辑模型。在理解当前系统“怎样做”的基础上，抽取其“做什么”的本质，从而从当前系统的物理模型抽象出当前系统的逻辑模型。

在物理模型中有许多物理因素，随着分析工作的深入，有些非本质的物理因素就成为不必要的负担，因而需要对物理模型进行分析，区分出本质的和非本质的因素，去掉那些非本质的因素即可获得反映系统本质的逻辑模型。

③ 建立目标系统的逻辑模型。分析目标系统与当前系统逻辑上的差别，明确目标系统到底要“做什么”，从当前系统的逻辑模型导出目标系统的逻辑模型。

④ 建立目标系统的物理模型。根据新系统的逻辑模型构建出相应的物理模型。

值得注意的是，原有系统可以是一个正在运行的软件系统，也可以是一个纯手工运作的流程。

试题答案

(25) A

(26) D

试题 21

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， (27) 属于用户文档。

(27) 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

试题分析

用户文档主要描述所交付系统的功能和使用方法，并不关心这些功能是怎样实现的。用户文档是了解系统的第一步，它可以让用户获得对系统准确的初步印象。

用户文档一般包括以下内容：

① 功能描述：说明系统能做什么。

② 安装文档：说明怎样安装这个系统及怎样使系统适应特定的硬件配置。

③ 使用手册：简要说明如何着手使用这个系统（通过丰富的例子说明怎样使用常用的系统功能，并说明用户操作错误是怎样恢复和重新启动的）。

④ 参考手册：详尽描述用户可以使用的系统设施，以及它们的使用方法，

- ② 对于未获得批准的变更, 不应该做设计和实现工作。
- ③ 变更应该由项目变更控制委员会决定实现哪些变更。
- ④ 项目风险承担者应该能够了解变更数据库的内容。
- ⑤ 决不能从数据库中删除或者修改变更请求的原始文档。
- ⑥ 每一个集成的需求变更必须能跟踪到一个经核准的变更请求。

试题答案

(29) C

试题 24

用边界值分析法, 假定 $10 < X < 30$, 那么 X 在测试中应取的边界值是 (30) 。

- (30) A. $X=11, X=29$ B. $X=9, X=10, X=30, X=31$
C. $X=10, X=30$ D. $X=9, X=31$

试题分析

边界值分析法是对各种输入、输出范围的边界情况设计测试用例的方法。

使用边界值分析法设计测试用例, 确定边界情况应考虑选取正好等于、刚刚大于或刚刚小于边界的值作为测试数据, 这样发现程序中错误的概率较大。

边界值分析法的使用要注意以下几点:

① 如果输入条件规定了取值范围或数据个数, 则可选择正好等于边界值、刚刚在边界范围内和刚刚超越边界外的值进行测试。

② 针对规格说明的每个输入条件, 使用上述原则。

③ 对于有序数列, 选择第一个和最后一个作为测试数据。

试题答案

(30) B

试题 25

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 (31) 。(32) 就是一种典型的项目管理工具。

- (31) 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
- (32) 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

试题分析

项目管理工具用来辅助软件的项目管理活动。通常项目管理活动包括项目的计划、调度、通信、成本估算、资源分配及质量控制等。一个项目管理工具通常把重点放在某一个或某几个特定的管理环节上，而不提供对管理活动包罗万象的支持。

项目管理工具具有以下特征：

- ① 覆盖整个软件生存周期。
- ② 为项目调度提供多种有效手段。
- ③ 利用估算模型对软件费用和工作量进行估算。
- ④ 支持多个项目和子项目的管理。
- ⑤ 确定关键路径、松弛时间、超前时间和滞后时间。
- ⑥ 对项目组成员和项目任务之间的通信给予辅助。
- ⑦ 自动进行资源平衡。
- ⑧ 跟踪资源的使用。
- ⑨ 生成固定格式的报表和裁剪项目报告。

成本估算工具就是一种典型的项目管理工具。

试题答案

(31) D

(32) B

试题 26

逆向工程导出的信息可以分为 4 个抽象层次，其中___(33)___可以抽象出程序的抽象语法树、符号表等信息；___(34)___可以抽象出反映程序段功能及程序段之间关系的信息。

(33) A. 实现级

B. 结构级

C. 功能级

D. 领域级

(34) A. 实现级

B. 结构级

C. 功能级

D. 领域级

试题分析

逆向工程导出的信息可分为如下 4 个抽象层次：

- ① 实现级：包括程序的抽象语法树、符号表等信息。
- ② 结构级：包括反映程序分量之间相互依赖关系的信息，如调用图、结构图等。
- ③ 功能级：包括反映程序段功能及程序段之间关系的信息。
- ④ 领域级：包括反映程序分量或程序与应用领域概念之间对应关系的信息。

试题答案

(33) A

(34) C

试题 27

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账号是否正确。用例“创建新订单”、“更新订单”与用例“核查客户账号”之间是 (35) 关系。

(35) A. 包含 (Include)

B. 扩展 (Extend)

C. 分类 (Classification)

D. 聚集 (Aggregation)

试题分析

用例是在系统中执行的一系列动作, 这些动作将生成特定参与者可见的价值结果。它确定了一个和系统参与者进行交互, 并可由系统执行的动作序列。用例模型描述的是外部执行者 (Actor) 所理解的系统功能。用例模型用于需求分析阶段, 它的建立是系统开发者和用户反复讨论的结果, 表明了开发者和用户对需求规格达成的共识。

两个用例之间的关系主要有两种情况: 一种是用于重用的包含关系, 用构造型 Include 表示; 另一种是用于分离出不同行为的扩展, 用构造型 Extend 表示。

① 包含关系: 当可以从两个或两个以上的原始用例中提取公共行为, 或者发现能够使用一个构件来实现某一个用例的部分功能是很重要的事时, 应该使用包含关系来表示它们。

② 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事情, 可以断定将这个用例分为一个主用例和一个或多个辅用例描述可能更加清晰。

试题答案

(35) A

试题 28

面向对象的设计模型包含以 (36) 表示的软件体系结构图, 以 (37), 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。

(36) A. 部署图

B. 包图

C. 协同图

D. 交互图

(37) A. 部署图

B. 包图

C. 协同图

D. 交互图

试题分析

面向对象的设计模型包含以包图表示的软件体系结构图, 以交互图表示的用例实

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。

试题答案

(36) B

(37) D

试题 29

基于构件的开发模型包括软件的需求分析定义、(38)、(39)、(40),以及测试和发布 5 个顺序执行的阶段。

(38) A. 构件接口设计

B. 体系结构设计

C. 元数据设计

D. 集成环境设计

(39) A. 数据库建模

B. 业务过程建模

C. 对象建模

D. 构件库建立

(40) A. 应用软件构建

B. 构件配置管理

C. 构件单元测试

D. 构件编码实现

试题分析

本题考查基于构件的软件开发模型的基础知识。

基于构件的开发模型利用模块化方法将整个系统模块化,并在一定构件模型的支持下复用构件库中的一个或多个软件构件,通过组合手段高效率、高质量地构造应用软件系统的过程。基于构件的开发模型融合了螺旋模型的许多特征,本质上是演化形的,开发过程是迭代的。基于构件的开发模型由软件的需求分析定义、体系结构设计、构件库建立、应用软件构建及测试和发布 5 个阶段组成。

试题答案

(38) B

(39) D

(40) A

试题 30

以下关于软件构件及其接口的叙述,错误的是(41)。

(41) A. 构件是软件系统中相对独立且具有一定意义的构成成分

B. 构件在容器中进行管理并获取其属性或者服务

C. 构件不允许外部对所支持的接口进行动态发现或调用

D. 构件可以基于对象实现,也可以不基于对象实现

试题分析

本题考查软件构件的基本概念。

软件构件是软件系统中具有一定意义的、相对独立的可重用单元。与对象相比,构件可以基于对象实现,也可以不作为对象实现。构件需要在容器中管理并获取容器提供的服务;客户程序可以在运行状态下利用接口动态确定构件所支持的功能并调用。

试题答案

(41) C

试题 31

在一个典型的基于 MVC (Model-View-Controller) 的 J2EE 应用中, 分发客户请求、有效组织其他构件为客户端提供服务的控制器由__(42)___实现。

(42) A. Entity Bean

B. Session Bean

C. Servlet

D. JSP

试题分析

J2EE 平台下实现 MVC, 系统的界面通常由 JSP 构件实现, 分发客户请求、有效组织其他构件为客户端提供服务的控件器由 Servlet 构件实现, 数据库相关操作由 Entity Bean 构件实现, 系统核心业务逻辑由 Session Bean 构件实现。

试题答案

(42) C

试题 31

集成测试有各种方法, 以下关于集成测试的描述中, 不正确的是__(43)___。

(43) A. 增量式集成测试容易定位错误, 排除错误

B. 非增量式集成测试不能充分利用人力, 会拖延工程进度

C. 增量式集成测试的强度大, 测试更彻底

D. 即使各个模块都通过了测试, 但系统集成以后仍可能出现错误

试题分析

集成测试也称组装测试, 它是对由各模块组装而成的程序进行测试, 主要目标是发现模块间的接口和通信问题。集成测试主要发现设计阶段产生的错误, 集成测试计划应该在概要设计阶段制定。集成的方式可分为非增量式和增量式。

非增量式集成测试又称整体拼装。使用这种方式, 首先对每个模块分别进行模块测试, 然后把所有模块组装在一起进行测试, 最终得到要求的软件系统。由于程序中不可避免地存在涉及模块间接口、全局数据结构等方面的问题, 所以一次试运行成功的可能性并不是很大。

增量式集成测试又称渐增式集成方式。首先对一个个模块进行模块测试, 然后将这些模块逐步组装成较大的系统, 在组装的过程中边连接边测试, 以发现连接过程中产生的问题。最后通过增值逐步组装成为要求的软件系统。

① 自顶向下的增值方式: 将模块按系统程序结构, 沿控制层次自顶向下进行集成。由于这种增值方式在测试过程中较早地验证了主要的控制和判断点。在一个功能划分合理的程序结构中, 判断常出现在较高的层次, 较早就能遇到。如果主要控制有问题, 尽早发现它能够减少以后的返工。

② 自底向上的增殖方式：从程序结构的最底层模块开始组装和测试。因为模块是自底向上进行组装的，对于一个给定层次的模块，它的子模块（包括子模块的所有下属模块）已经组装并测试完成，所以不再需要桩模块。在模块的测试过程中，需要从子模块得到的信息可以直接运行子模块得到。

③ 混合增殖式测试：自顶向下增殖的方式和自底向上增殖的方式各有优缺点。自顶向下增殖方式的缺点是需要建立桩模块。要使桩模块能够模拟实际子模块的功能将是十分困难的。同时涉及复杂算法和真正输入/输出的模块一般在底层，它们是最容易出问题的模块，到组装和测试的后期才遇到这些模块，一旦发现问题，导致过多的回归测试。而自顶向下增殖方式的优点是能够较早地发现在主要控制方面的问题。自底向上增殖方式的缺点是“程序一直未能作为一个实体存在，直到最后一个模块加上后才形成一个实体”。就是说，在自底向上组装和测试的过程中，对主要的控制直到最后才接触到。但这种方式的优点是不需要桩模块，而建立驱动模块一般比建立桩模块容易，同时由于涉及复杂算法和真正输入/输出的模块最先得到组装和测试，可以把最容易出问题的部分在早期解决。此外，自底向上增殖的方式可以实施多个模块的并行测试。

鉴于此，通常是把以上两种方式结合起来进行组装和测试。

④ 衍变的自顶向下的增殖测试：它的基本思想是强化对输入/输出模块和引入新算法模块的测试，并自底向上组装成为功能相当完整且相对独立的子系统，然后由主模块开始自顶向下进行增殖测试。

⑤ 自底向上一自顶向下的增殖测试：它首先对含读操作的子系统自底向上直至根节点模块进行组装和测试，然后对含写操作的子系统做自顶向下的组装与测试。

⑥ 回归测试：这种方式采取自顶向下的方式测试被修改的模块及其子模块，然后将这一部分视为子系统，再自底向上测试，以检查该子系统与其上级模块的接口是否适配。

增量式与非增量式测试的优点和缺点比较如表 7-20 所示。

表 7-20 增量式与非增量式测试的优点和缺点比较

	增量式集成测试	非增量式集成测试
错误定位	容易定位错误，排除故障	不容易定位错误
测试强度	先加入的模块经过多次测试，测试强度大，测试更彻底	测试强度小
测试的工作量	测试的工作量大	测试的工作量小
测试的进度	测试的过程长，进度慢	对各个模块可以并行测试，加快测试进度
测试辅助程序	自顶向下的增量需要编写桩模块；自底向上的增量需要编写驱动模块	每个中间模块的测试都需要编写驱动模块和桩模块

试题答案

(43) B

试题 32

在 UML 的通用机制中，____(44)____用于把元素组织成组；____(45)____是系统中

遵从一组接口规范且付诸实现的物理的、可替换的软件模块。

(44) A. 包 B. 类 C. 接口 D. 构件

(45) A. 包 B. 类 C. 接口 D. 构件

试题分析

本题是纯概念题，在 UML 的通用机制中，包用于把元素组织成组；构件是系统中遵从一组接口规范且付诸实现的物理的、可替换的软件模块。

试题答案

(44) A

(45) D

试题 33

对 OO 系统的技术度量的识别特征，Berard 定义了导致特殊度量的特征。其中 (46) 抑制程序构件的操作细节，只有对访问构件必需的信息被提供给其他希望访问的构件。

(46) A. 局部化 B. 封装 C. 信息隐藏 D. 继承

试题分析

面向对象的软件和用传统方法开发的软件有本质性不同，为此，对 OO 系统的技术度量必须调整以适应那些区别 OO 和传统软件的特征。Berard 定义了 5 个导致特殊度量的特征，分别是局部化、封装、信息隐蔽、继承和对象抽象技术。

① 局部化。局部化是软件的一个特征，它指明信息在程序中被集中的方式，例如，针对功能分解的传统方法围绕功能局部化信息，它们典型地以过程模块来实现。数据驱动方法围绕特定的数据结构局部化信息。在 OO 语境中，信息是通过封装数据和处理在类或对象的边界内而集中的。因为传统软件强调函数为局部化机制，软件度量着重于函数的内部结构或复杂性（如模块长度、内聚性或环路复杂性等）或函数间相互连接的方式（如模块耦合）。因为类是 OO 系统的基本单位，所以，局部化是基于对象的，因此，度量应该应用于作为一个完全实体的类（对象）。此外，在操作（函数、方法）和类间的关系没必要是一对一的。因此，反应类协作方式的度量必须能够适应一对多和多对一的关系。

② 封装。Berard 定义封装为“一组项的包装（或捆绑在一起），（对传统软件的）低层封装例子包括记录和数组，而子程序（例如，过程、函数、子例程和段落）是封装的中层机制。”

对 OO 系统，封装包含了类的责任，包括其属性（和针对聚合对象的其他类）和操作，以及由特定的属性值定义的类的状态。封装通过将测度的焦点从单个模块改变到数据（属性）和处理模块（操作）包而影响度量。此外，封装鼓励在高抽象层的测度。将此层次的抽象同传统的度量相比较，传统的着重于布尔条件的计数或代码行数。

③ 信息隐蔽。信息隐蔽隐瞒（或隐藏）程序构件的操作细节，只将对访问该构件必需的信息提供给那些希望访问它的其他构件。良好设计的 OO 系统应该鼓励信息隐蔽，因此，指明隐蔽所达到程度的度量应该提供了对 OO 设计质量的一个指标。

④ 继承。继承是使得某对象的责任能够传播到其他对象的机制，继承出现在类层次的所有层面上，通常，传统的软件不支持该特征。因为继承是很多 OO 系统的关键特征，所以很多 OO 度量是关注于它的。

⑤ 抽象。抽象是使得设计者能够关注程序构件（数据或过程）的本质性细节而不需考虑低层细节的机制。如 Berard 所说，抽象是一个相对概念，当我们移向更高的抽象级别时，我们忽略了越来越多的细节，即我们提供了对概念或项的更一般化的视图；当我们移向抽象的低层时，我们引入了更多的细节，即我们提供了概念或项的更特定的视图。

因为类是一种抽象，它可以在很多不同的细节级别上并以一系列不同的方式（例如，作为一个操作列表、作为一个状态序列、作为一系列协作）来观察，所以 OO 度量用类的测度（例如，每个应用的每个类的实例数、每个应用的参数化类数，以及参数化类和非参数化类的比率）来表示抽象。

试题答案

(46) C

试题 34

采用 UML 分析用户需求时，用例 UC1 可以出现在用例 UC2 出现的任何位置，那么 UC1 和 UC2 之间的关系是 (47)。

(47) A. include B. extend C. generalize D. call

试题分析

在 UML 的用例中，用例之间有 3 种关系，分别是 include（包含）、extend（扩展）和 generalize（泛化）。当两个用例之间具有继承关系时，意味着子用例可以继承父用例的行为，并可以在此基础上添加新的行为或重置父用例的行为，子用例可以出现在任何位置上。因此，UC1 和 UC2 之间的关系应该是泛化关系。

试题答案

(47) C

试题 35

关于 UML 中的状态图和活动图的叙述中，(48) 是错误的。

- (48) A. 状态图和活动图都是对系统的动态方面进行建模的图
B. 活动图是状态图的一种特殊情况，其中所有或大多数状态是活动状态
C. 状态图显示从状态到状态的控制流，活动图显示的是从活动到活动的控制流
D. 在对一个对象的生存期建模时，只能使用状态图

试题分析

状态图和活动图都是 UML 用于对系统动态方面进行建模的行为图，其中状态图

注重于由事件驱动的系统的变化状态；而活动图注重于从活动到活动的控制流。一个活动是一个状态机中进行的非原子的执行单元。活动图是状态机的一种特殊情况，其中全部或大多数状态是活动状态，并且全部或大多数转换是通过源状态中活动的完成来触发的。因为活动图是一种状态机，所以它可以适应状态机的全部特性。活动图和状态图在对一个对象的生命周期建模时都是有用的。

试题答案

(48) D

试题 36

UML 提供了 5 种对系统动态方面建模的图，其中 (49) 对系统行为组织和建模； (50) 对系统功能建模，它强调对象之间的控制流； (51) 之间是同构的。

- (49) A. 用例图 B. 顺序图 C. 协作图 D. 状态图
- (50) A. 用例图 B. 活动图 C. 状态图 D. 顺序图
- (51) A. 状态图和活动图 B. 用例图和活动图
- C. 顺序图和协作图 D. 活动图和协作图

试题分析

UML 提供了 5 种对系统动态方面建模的图。其中：

用例图描述了一组用例和参与者及它们之间的关系，对于系统行为的组织和建模特别重要。

交互图是顺序图和协作图的统称。顺序图强调消息的时间次序；协作图强调收发消息的对象的结构组织。这种是同构的，这意味着可以把一种图转换为另一种图而没有信息损失。

状态图显示了一个由状态、转换、事件和活动组成的状态机，它强调一个对象按事件次序发生的行为。

活动图显示了系统中从活动到活动的流，它强调对象之间的控制流，对系统的功能建模是非常重要的。

试题答案

(49) A (50) B (51) C

试题 37

UML 采用 4+1 视图来描述软件和软件开发过程，其中 (52) 描绘了所设计的并发与同步结构； (53) 表示软件到硬件的映射及分布结构；UML 中的类图可以用来表示 4+1 视图中的 (54)。

- (52) A. 逻辑视图 (Logical View) B. 实现视图 (Implementation View)
- C. 进程视图 (Process View) D. 部署视图 (Deployment View)

- (53) A. 逻辑视图 B. 实现视图 C. 进程视图 D. 部署视图
(54) A. 逻辑视图 B. 实现视图 C. 进程视图 D. 部署视图

试题分析

UML 采用 4+1 视图来描述软件和软件开发过程。

- ① 逻辑视图：以问题域的语汇组成的类和对象集合。
- ② 进程视图：可执行线程和进程作为活动类的建模，它是逻辑视图的一次执行实例，描绘了所设计的并发与同步结构。
- ③ 实现视图：对组成基于系统的物理代码的文件和组件进行建模。
- ④ 部署视图：把构件部署到一个组物理的、可计算的节点上，表示软件到硬件的映射及分布结构。
- ⑤ 用例视图：最基本的需求分析模型。

试题答案

- (52) C (53) D (54) A

试题 38

系统输入设计中应尽可能考虑人的因素，以下关于输入设计的一般原理中，错误的是 (55)。

- (55) A. 只让用户输入变化的数据 B. 使用创新的模式吸引用户的眼球
C. 表格中各个数据项应有提示信息
D. 尽可能使用选择而不是键盘输入的方式获取数据

试题分析

本题考查软件界面设计中的系统输入设计。人的因素在系统输入设计中扮演了很重要的角色。输入应该尽可能地简单，以降低错误发生的可能性，如对于范围可控的数据，使用选择的方式替代用户输入；只输入变化的数据等。输入应该尽可能使用已有含义明确的设计，需要采用模仿的方式而非创新。为了避免用户理解的二义性，应该对表格中输入的数据给出提示信息。

试题答案

- (55) B

7.3 实战练习题

● 需求工程活动产生软件运行特征的规约，指明软件和其他系统元素的接口并建立 (1)。

- (1) A. 数据流图和数据字典 B. 程序流程图

C. 体系结构模型

D. 软件必须满足的约束条件

● 有两种需求定义的方法——严格定义和原型定义，在关于这两种方法的描述中，不正确的是__ (2) __。

(2) A. 严格定义方法假定所有的需求都可以预先定义

B. 严格定义方法假定软件开发人员与用户之间的沟通存在障碍

C. 原型定义方法认为需求分析中不可避免地要出现很多反复

D. 原型定义方法强调用户在软件开发过程中的参与和决策

● 软件需求分析产生软件操作特征的规格说明，指明软件和其他系统元素的接口，建立软件必须满足的约束。下面对于软件需求分析的描述，不正确的是__ (3) __。

(3) A. 分析员研究系统规约和软件项目计划，并在系统语境内理解软件和复审，从而生成计划软件范围的估算

B. 需求分析使得系统工程师能够刻画出软件的功能和性能、指明软件和其他系统元素的接口、并建立软件必须满足的约束

C. 经过仔细的需求分析活动，分析员能够得到详细的系统规约

D. 需求分析能够为软件设计者提供可被翻译成数据、体系结构、界面和过程设计的模型

● 质量功能部署 (QFD) 是一种将客户要求转化成软件需求的技术。QFD 的目的是最大限度地提升软件工程过程中客户的满意度。为了这个目标，QFD 确认了 3 类需求，常规需求、__ (4) __ 和意外需求。

(4) A. 期望需求

B. 基础需求

C. 显式需求

D. 功能需求

● 需求分析的任务是借助于当前系统的物理模型导出目标系统的逻辑模型，解决目标系统“做什么”的问题。__ (5) __ 并不是需求分析的实现步骤之一。

(5) A. 获得当前系统的物理模型

B. 抽象出当前系统的逻辑模型

C. 建立目标系统的逻辑模型

D. 确定目标实现的具体技术路线

● 希赛网软件开发团队欲开发一套管理信息系统，在项目初期，用户提出了软件的一些基本功能，但是没有详细定义输入、处理和输出需求。在这种情况下，该团队在开发过程应采用__ (6) __。

(6) A. 瀑布模型

B. 增量模型

C. 原型开发模型

D. 快速应用程序开发 (RAD)

● 基于构件的开发 (CBD) 模型，融合了__ (7) __ 模型的许多特征。该模型本质是演化的，采用迭代方法开发软件。

(7) A. 瀑布

B. 快速应用开发 (RAD)

C. 螺旋

D. 形式化方法

● 统一软件开发过程是一种基于面向对象技术的软件开发过程，其特点是“用例

驱动，以架构为核心，迭代并增量”。统一软件开发过程定义了4种通用的开发阶段，它们按照过程顺序分别是：起始阶段、__（8）__、构建阶段和__（9）__，其中在构建阶段主要产生的文档有__（10）__。

（8）A. 分析阶段 B. 细化阶段 C. 设计阶段 D. 交付阶段

（9）A. 分析阶段 B. 细化阶段 C. 设计阶段 D. 交付阶段

（10）A. 初始用户手册 B. 用例模型 C. 项目计划 D. 设计模型

● 敏捷软件过程强调：让客户满意和软件尽早增量发布；小而高度自主的项目团队；非正式的方法；最小化软件工作产品，以及整体精简开发。__（11）__不是采用这种软件开发过程的原因。

（11）A. 难以提前预测哪些需求是稳定的和哪些需求会变化

B. 对于软件项目开发来说，设计和实现可以做到基本分离

C. 从制订计划的角度来看，分析、设计、实现和测试并不容易预测

D. 可执行原型和部分实现的可运行系统是了解用户需求和反馈的有效媒介

● 逆向工程过程的抽象层次是指可从源代码中抽取出来的设计信息的精制程度。抽象层次分为4层，其中，“最低层”抽象能够导出过程的设计表示文档，“低层”抽象能够导出程序和数据结构信息，“中层”能够导出__（12）__，“高层”抽象能够导出__（13）__。

（12）A. 实体关系模型

B. 程序和文档结构信息

C. 全部文档信息

D. 数据流和控制流模型

（13）A. 实体关系模型

B. 模块结构图

C. 完全的数据流图

D. 全部文档信息

● 详细的项目范围说明书是项目成功的关键。__（14）__不应该属于范围定义的输入。

（14）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__（15）__来进行活动定义。

（15）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

● 软件的逆向工程是一个恢复设计的过程，从现有的程序中抽取数据、体系结构和过程的设计信息。逆向工程的完备性可以用在某一个抽象层次上提供信息的详细程度来描述，在大多数情况下，抽象层次越高，完备性就越低。下列可以通过逆向工程恢复的制品中，完备性最低的是__（16）__。

（16）A. 过程的设计模型

B. 程序和数据结构

C. 对象模型、数据和控制流 D. UML 状态图和部署图

● (17) 把整个软件开发流程分成多个阶段, 每一个阶段都由目标设定、风险分析、开发和有效性验证及评审构成。

(17) A. 原型模型 B. 瀑布模型 C. 螺旋模型 D. V 模型

● 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 (18), 系统工程师侧重于 (19)。

(18) A. 实现视图 B. 进程视图 C. 逻辑视图 D. 部署视图

(19) A. 实现视图 B. 进程视图 C. 逻辑视图 D. 部署视图

● 软件的横向重用是指重用不同应用领域中的软件元素。(20) 是一种典型的、原始的横向重用机制。

(20) 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 (21)。

(21) 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

● 在实际的项目开发中, 人们总是希望使用自动工具来执行需求变更控制过程。下列描述中, (22) 不是这类工具所具有的功能。

(22) A. 可以定义变更请求的数据项及变更请求生存期的状态转换图

B. 记录每一种状态变更的数据, 确认做出变更的人员

C. 可以加强状态转换图使经授权的用户仅能做出所允许的状态变更

D. 定义变更控制计划, 并指导设计人员按照所制定的计划实施变更

● 黑盒测试法是根据软件产品的功能设计规格说明书, 通过运行程序进行测试, 证实每个已经实现的功能是否符合设计要求。如果某产品的文本编辑框允许输入1~255个字符, 采用 (23) 测试方法, 其测试数据为: 0个字符、1个字符、255个字符和256个字符。

(23) A. 等价类划分

B. 边界值分析

C. 比较测试

D. 正交数组测试

● 软件开发环境是支持软件产品开发的软件系统, 它由软件工具集和环境集成机制构成。环境集成机制包括: 提供统一的数据模式和数据接口规范的数据集成机制; 支持各开发活动之间通信、切换、调度和协同的 (24); 为统一操作方式提供支持的 (25)。

(24) A. 操作集成机制

B. 控制集成机制

- C. 平台集成机制 D. 界面集成机制
- (25) A. 操作集成机制 B. 控制集成机制
- C. 平台集成机制 D. 界面集成机制
- (26) 是一个独立可交付的功能单元，外界通过接口访问其提供的服务。
- (26) A. 面向对象系统中的对象 (Object)
- B. 模块化程序设计中的子程序 (Subroutine)
- C. 基于构件开发中的构件 (Component)
- D. 系统模型中的包 (Package)
- 在基于构件的软件开发中，(27) 描述系统设计蓝图以保证系统提供适当的功能；(28) 用来了解系统的性能、吞吐率等非功能性属性。
- (27) A. 逻辑构件模型 B. 物理构件模型
- C. 组件接口模型 D. 系统交互模型
- (28) A. 逻辑构件模型 B. 物理构件模型
- C. 组件接口模型 D. 系统交互模型
- 对象管理组织 (OMG) 基于 CORBA 基础设施定义了 4 种构件标准。其中，(29) 的状态信息是由构件自身而不是由容器维护。
- (29) A. 实体构件 B. 加工构件
- C. 服务构件 D. 会话构件
- 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 (30)。
- (30) A. 表示逻辑层 B. 应用逻辑层 C. 数据处理层 D. 数据层
- 系统输入设计中，采用内部控制方式以确保输入系统数据的有效性，(31) 用于验证数据是否位于合法的取值范围。
- (31) A. 数据类型检查 B. 自检位 C. 域检查 D. 格式检查
- 系统测试由若干个不同的测试类型组成，其中 (32) 检查系统能力的最高实际限度，即软件在一些超负荷情况下的运行情况；(33) 主要是检查系统的容错能力。
- (32) A. 强度测试 B. 性能测试 C. 恢复测试 D. 可靠性测试
- (33) A. 强度测试 B. 性能测试 C. 恢复测试 D. 可靠性测试
- 需求管理是 CMM 可重复级中的 6 个关键过程域之一，其主要目标是 (34)。
- (34) A. 对于软件需求，必须建立基线以进行控制，软件计划、产品和活动必须与软件需求保持一致
- B. 客观地验证需求管理活动符合规定的标准、程序和要求

- C. 策划软件需求管理的活动, 识别和控制已获取的软件需求
- D. 跟踪软件需求管理的过程、实际结果和执行情况
- UML 的事物是对模型中最具有代表性的成分的抽象, (35) 是模型的静态部分, 描述概念或物理元素; (36) 用来描述、说明和标注模型的任何元素。
- (35) A. 结构事物 B. 分组事物 C. 行为事物 D. 注释事物
- (36) A. 分组事物 B. 注释事物 C. 结构事物 D. 行为事物
- 希赛公司欲开发一个在线交易系统。为了能够精确表达用户与系统的复杂交互过程, 应该采用 UML 的 (37) 进行交互过程建模。
- (37) A. 类图 B. 顺序图 C. 部署图 D. 对象图
- 雇员类含有计算报酬的行为, 利用面向对象的 (38), 可以使得其派生类专职雇员类和兼职雇员类计算报酬的行为有相同的名称, 但有不同的计算方法。
- (38) A. 多态性 B. 继承性 C. 封装性 D. 复用性
- 面向对象分析的一项重要任务是发现潜在对象并进行筛选, 错误的做法是删除 (39)。
- (39) A. 系统范围之外的名词 B. 表示事件的名词
C. 不具有独特行为的名词 D. 一个对象的同义词
- 面向对象分析的任务不包含 (40)。
- (40) A. 建模系统功能 B. 发现并确定业务对象
C. 建模各对象的状态 D. 组织对象并确定对象间的关系
- 系统测试将软件、硬件、网络等其他因素结合, 对整个软件进行测试。 (41) 不是系统测试的内容。
- (41) A. 路径测试 B. 可靠性测试 C. 安装测试 D. 安全测试
- 软件测试是为了发现错误而执行程序的过程。黑盒测试法主要根据 (42) 来设计测试用例。
- (42) A. 程序内部逻辑 B. 程序内部功能
C. 程序数据结构 D. 程序流程图
- 复杂系统是指 (43)。
- (43) A. 通过对各子系统的了解不能对系统的性能做出完全的解释
B. 系统由大量的子系统组成
C. 系统的结构很复杂, 难于图示
D. 系统的功能很复杂, 难于用文字描述

● 静态分析通过解析程序文本从而识别出程序语句中可能存在的缺陷和异常之处；静态分析所包含的阶段中，(44)的主要工作是找出输入变量和输出变量之间的依赖关系。

(44) A. 控制流分析 B. 数据使用分析 C. 接口分析 D. 信息流分析

● 软件(45)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。

(45) A. 完善性维护 B. 适应性维护
C. 正确性维护 D. 预防性维护

7.4 练习题解析

试题 1 分析

需求工程活动产生软件运行特征的规约，指明软件和其他系统元素的接口并建立软件必须满足的约束条件。数据流图和数据字典只是这些约束条件的表示方法，而程序流程图和体系结构模型是设计阶段的工作。

试题 1 答案

(1) D

试题 2 分析

严格定义（预先定义）是目前采用较多的一种需求定义方法。在采用严格定义的传统结构化开发方法中，各个工作阶段排列成一个理想的线性开发序列，在每一工作阶段中，都用上一阶段所提供的完整、严格的文档作为指导文件，因此它本质上是一种顺序型的开发方法。

在传统的结构化开发中，需求的严格定义建立在以下的基本假设上：

① 所有需求都能够被预先定义。假设意味着，在没有实际系统运行经验的情况下，全部的系统需求均可通过逻辑推断得到。这对某些规模较小、功能简单的系统是可能的，但对那些功能庞大、复杂且较大的系统显然是困难的。即使事先做了深入细致的调查和分析，当用户见到新系统的实际效果时，也往往会改变原先的看法，会提出修改或更进一步增加系统功能的要求，所以再好的预先定义技术也会经常反复。这是因为人们对新事物的认识与理解将随着直观、实践的过程进一步加深，这是与人类认识世界的客观规律相一致的。所以，能够预先定义出所有需求的假设在许多场合是不能成立的。

② 开发人员与用户之间能够准确而清晰地交流。假设认为，用户与开发人员之间，虽然每人都有自己的专业、观点、行话，但在系统开发过程中可以使用图形/文档等通信工具进行交流，进行清晰、有效的沟通，这种沟通是必不可少的。可是，在实际开发中，往往对一些共同的约定，每个人可能都会产生自己的理解和解释。即使采用结构化语言、判定树、判定表等工具，仍然存在精确的、技术上的不严密感。这

将导致人们有意无意地带有个人的不同理解而各行其是，所以在多学科、多行业人员之间进行有效的通信交流是有一定困难的。

③ 采用图形/文字可以充分体现最终系统。在使用严格定义需求的开发过程中，开发人员与用户之间交流、通信的主要工具是定义报告，包括叙述文字、图形、逻辑规则和数据字典等技术工具。它们都是静止的、被动的，不能实际表演，很难在用户头脑中形成一个具体的形象。因此，要用静止的图形/文字描述来体现一个动态的系统是比较困难的。

除了所论述的情况外，上述基本假设还将导致严格定义的结构化开发方法存在以下缺陷。

首先是文档量大，由于在结构化方法的每个阶段都必须写出规范、严密的各种文档，这些文档虽然有助于开发人员之间、用户与开发人员间的通信交流，有助于开发过程的规范化，但由于编写文档花费大量人力和时间，导致系统开发周期增大。其次是开发过程可见性差，来自用户的反馈太迟。由于在需求定义、系统设计阶段都不能在用户终端显示新系统的实际效果，一直到系统实现阶段结束，用户才有机会通过对新系统的实际操作和体会来提出他们对新系统的看法和意见，但此时整个开发已近尾声，若想修改前几段的工作或修改需求定义，都将付出较大的代价，有时这种修改甚至会导致整个系统的失败。

综上所述，需求的严格定义的基本假设在许多情况下并不成立，传统的结构化方法面临着一些难以跨越的障碍。为此，需要探求一种变通的方法。

原型方法以一种与严格定义法截然不同的观点看待需求定义问题。原型化的需求定义过程是一个开发人员与用户通力合作的反复过程。从一个能满足用户基本需求的原型系统开始，允许用户在开发过程中提出更好的要求，根据用户的要求不断地对系统进行完善，它实质上是一种迭代的循环型的开发方式。

采用原型方法时需要注意以下几个问题：

① 并非所有的需求都能在系统开发前被准确地说明。事实上，要想严密、准确地定义任何事情都是有一定难度的，更不用说是定义一个庞大系统的全部需求。用户虽然可以叙述他们所需最终系统的目标及大致功能，但是对某些细节问题却往往不可能十分清楚。一个系统的开发过程，无论对于开发人员还是用户来说，都是一个学习和实践的过程，为了帮助他们在这个过程中提出更完善的需求，最好的方法就是提供现实世界的实例——原型，对原型进行研究、实践，并进行评价。

② 项目参加者之间通常都存在交流上的困难，原型提供了克服该困难的一个手段。用户和开发人员通过屏幕、键盘进行对话和讨论、交流，从他们自身的理解出发来测试原型，一个具体的原型系统，由于直观性、动态性而使得项目参加者之间的交流上的困难得到较好的克服。

③ 需要实际的、可供用户参与的系统模型。虽然图形和文字描述是一种较好的通信交流工具，但是，其最大缺陷是缺乏直观的、感性的特征，因而不易理解对象的全部含义。交互式的系统原型能够提供生动的规格说明，用户见到的是一个“活”的、实际运行着的系统。实际使用在计算机上运行的系统，显然比理解纸面上的系统要深

刻得多。

④ 有合适的系统开发环境。随着计算机硬件、软件技术和软件工具的迅速发展,软件的设计与实现工作越来越方便,对系统进行局部性修改甚至重新开发的代价大大降低。所以,对大系统的原型化已经成为可能。

⑤ 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。对系统改进的建议来自经验的发展,应该鼓励用户改进他们的系统,只有做必要的改变后,才能使用户和系统间获得更加良好的匹配,所以,从某种意义上说,严格定义需求的方法实际上抑制了用户在需求定义以后再改进的要求,这对提高最终系统的质量是有害的。另一方面,原型方法的使用并不排除严格定义方法的运用,当通过原型并在演示中得到明确的需求定义后,应采用行之有效的结构化方法来完成最终系统的开发。

试题 2 答案

(2) B

试题 3 分析

需求分析使得系统工程师能够刻画出软件的功能和性能、指明软件和其他系统元素的接口、并建立软件必须满足的约束。需求分析能够为软件设计者提供可被翻译成数据、体系结构、界面和过程设计的模型。分析员研究系统规约和软件项目计划,并在系统语境内理解软件和复审,从而生成计划软件范围的估算。

试题 3 答案

(3) C

试题 4 分析

QFD 确认了 3 类需求,分别是基本需求(常规需求)、期望需求和意外需求(兴奋需求)。其中期望需求指的是那些隐含在产品或系统中,可能由于非常基础以至于用户没有显式说明的需求。

试题 4 答案

(4) A

试题 5 分析

请参看“7.2 典型试题分析”中试题 21 的分析。

试题 5 答案

(5) D

试题 6 分析

瀑布模型也称为生命周期法,是生命周期法中最常用的开发模型,它把软件开发的过程分为软件计划、需求分析、软件设计、程序编码、软件测试和运行维护 6 个阶段,规定了它们自上而下、相互衔接的固定次序,如同瀑布流水,逐级下落。瀑布模型是最早出现的软件开发模型,在软件工程中占有重要的地位,它提供了软件开发的基本框架。瀑布模型主要用于需求明确或很少变更的项目。

原型法适合于用户没有肯定其需求的明确内容的时候。它是先根据已给的和分析的需求,建立一个原始模型,这是一个可以修改的模型(在生命周期法中,需求分析成文档后一般不再进行修改)。在软件开发的各个阶段都把有关信息相互反馈,直至模型的修改,使模型渐趋完善。在这个过程中,用户的参与和决策加强了,最终的结果是更适合用户的要求。这种原型法成败的关键及效率的高低,关键在于模型的建立及建模的速度。

增量模型融合了瀑布模型的基本成分(重复地应用)和原型的迭代特征。采用随着日程时间的进展而交错的线性序列。每一个线性序列产生软件的一个可发布的“增量”。当使用增量模型时,第一个增量往往是核心的产品,即实现了基本的需求,但很多补充的特性还没有发布。核心产品交用户使用,使用和/或评估的结果是下一个增量的开发计划。该计划包括对核心产品的修改,使其能更好地满足用户的需要,并发布一些新增的特点和功能。这个过程在每一个增量发布后不断重复,直到产生最终的完善产品。

RAD 是一个线性顺序的软件开发模型,强调极短的开发周期和可复用程序构件的开发。RAD 模型是瀑布模型的一个高速变种,通过使用基于构件的建造方法获得了快速开发。如果需求理解得很好,且约束了项目范围, RAD 模型使得一个开发组能够在很短时间内创建出功能完善的系统。RAD 方法主要用于信息系统应用软件的开发,它包含业务建模、数据建模、处理建模、应用生成、测试及反复 5 个阶段。

试题 6 答案

(6) C

试题 7 分析

基于构件的开发模型利用模块化方法将整个系统模块化,并在一定构件模型的支持下复用构件库中的一个或多个软件构件,通过组合手段高效率、高质量地构造应用软件系统的过程。

基于构件的开发模型融合了螺旋模型的许多特征,本质上是演化形的,开发过程是迭代的。基于构件的开发模型由软件的需求分析和定义、架构设计、构件库建立、应用软件构建及测试和发布 5 个阶段组成。

试题 7 答案

(7) C

试题 8~10 分析

统一过程适合于大、中型项目的开发,可以分为 4 个顺序的阶段,分别是初始阶段、细化阶段、构建阶段和移交阶段。

初始阶段的任务是为系统建立业务模型并确定项目的边界。在初始阶段,必须识别所有与系统交互的外部实体,定义系统与外部实体交互的特性。在这个阶段中所关注的是整个项目的业务和需求方面的主要风险。对于建立在原有系统基础上的开发项目来说,初始阶段可能很短。

细化阶段的任务是分析问题领域,建立健全的架构基础,淘汰项目中最高风险的

元素。在细化阶段，必须在理解整个系统的基础上，对架构做出决策，包括其范围、主要功能和诸如性能等非功能需求，同时为项目建立支持环境。

在构建阶段，要开发所有剩余的构件和应用程序功能，把这些构件集成为产品，并进行详细测试。从某种意义上说，构建阶段是一个制造过程，其重点放在管理资源及控制操作，以优化成本、进度和质量。构建阶段的主要任务是通过优化资源和避免不必要的报废和返工，使开发成本降到最低；完成所有所需功能的分析、开发和测试，快速完成可用的版本；确定软件、场地和用户是否已经为部署软件做好准备。在构建阶段，开发团队的工作可以实现某种程度的并行。即使是较小的项目，也通常包括可以相互独立开发的构件，从而使各团队之间实现并行开发。

当基线已经足够完善，可以安装到最终用户实际环境中时，则进入交付阶段。交付阶段的重点是确保软件对最终用户是可用的。交付阶段的主要任务是进行 β 测试，制作产品发布版本；对最终用户支持文档定稿；按用户的需求确认新系统；培训用户和维护人员；获得用户对当前版本的反馈，基于反馈调整产品，如进行调试、性能或可用性的增强等。根据产品的种类，交付阶段可能非常简单，也可能非常复杂。例如，发布现有桌面产品的新发布版本可能十分简单，而替换一个国家的航空交通管制系统可能就非常复杂。交付阶段结束时也要进行技术评审，评审目标是否实现，是否应该开始演化过程，用户对交付的产品是否满意等。

试题 8~10 答案

(8) B

(9) D

(10) D

试题 11 分析

敏捷软件过程主要有四大价值观：个体和交互胜过过程和工具；可以工作的软件胜过面面俱到的文档；客户合作胜过合同谈判；响应变化胜过遵循计划。这种价值观的前提是软件需求是难以提前确定的，而是会不断地发生变化，可以采用可执行原型和部分实现的可运行系统来了解用户需求，通过用户的反馈来明确需求。从制订计划的角度来看，分析、设计、实现和测试并不容易预测。

试题 11 答案

(11) B

试题 12~13 分析

逆向工程过程能够导出过程的设计模型（实现级，一种低层的抽象）、程序和数据结构信息（结构级，稍高层次的抽象）、对象模型、数据和控制流模型（功能级，相对高层的抽象）和 UML 状态图和部署图（领域级，高层抽象）。随着抽象层次增高，完备性就会降低。抽象层次越高，它与代码的距离就越远，通过逆向工程恢复的难度就越大，而自动工具支持的可能性相对变小，要求人参与判断和推理的工作增多。

所以本题选 D、A。关于逆向工程的详细说明，请参看“7.1.4 软件开发方法”中的逆向工程。

试题 12~13 答案

(12) D

(13) A

试题 14 分析

在初步项目范围说明书中已文档化的主要的可交付物、假设和约束条件的基础上准备详细的项目范围说明书，是项目成功的关键。范围定义的输入包括以下内容：

① 项目章程。如果项目章程或初始的范围说明书没有在项目执行组织中使用，同样的信息需要进一步收集和开发，以产生详细的项目范围说明书。

② 项目范围管理计划。

③ 组织过程资产。

④ 批准的变更申请。

所以项目文档管理方案不属于范围定义的输入。

试题 14 答案

(14) D

试题 15 分析

项目时间管理包括使项目按时完成所必需的管理过程。项目时间管理中的过程包括：活动定义、活动排序、活动的资源估算、活动历时估算、制定进度计划及进度控制。

为了得到工作分解结构（Work Breakdown Structure, WBS）中最底层的交付物，必须执行一系列的活动。对这些活动的识别及归档的过程就是活动定义。

鱼骨图（又称为 Ishikawa 图）是一种发现问题“根本原因”的方法，通常用来进行因果分析。

试题 15 答案

(15) B

试题 16 分析

逆向工程过程及用于实现该过程的工具的抽象层次是指可从源代码中抽取出来的设计信息的精密程度。理想地，抽象层次应该尽可能高，即逆向工程过程应该能够导出过程的设计表示（一种低层的抽象）；程序和数据结构信息（稍高一点层次的抽象）；数据和控制流模型（一种相对高层的抽象）；以及实体关系模型（一种高层抽象）。随着抽象层次增高，软件工程师获得更有助于理解程序的信息。

在试题给出的 4 个选项中，UML 状态图和部署图可以用来描述实体之间的关系，因此，其层次最高，完备性最低。

试题 16 答案

(16) D

试题 17 分析

本题考查开发模型基础知识，解这类题，需要对常见模型的核心特点有所了解。下面对选项中出现的模型做一个简单的总结。

原型模型：针对需求不明确、原型可抛弃。

瀑布模型：阶段明晰、无法应对需求不明确的情况。

螺旋模型：瀑布模型+演化模型、循环、里程碑、风险分析。

V 模型：测试模型、测试全程介入、测试计划提前。

把以上特点与题目描述进行对比，可以发现本题所描述的是螺旋模型。

试题 17 答案

(17) C

试题 18~19 分析

在 RUP 中采用“4+1”视图模型来描述软件系统的体系结构。“4+1”视图包括逻辑视图、实现视图、进程视图、部署视图和用例视图。

分析人员和测试人员关心的是系统的行为，因此会侧重于用例视图；最终用户关心的是系统的功能，因此会侧重于逻辑视图；程序员关心的是系统的配置、装配等问题，因此会侧重于实现视图；系统集成人员关心的是系统的性能、可伸缩性、吞吐率等问题，因此会侧重于进程视图；系统工程师关心的是系统的发布、安装、拓扑结构等问题，因此会侧重于部署视图。

试题 18~19 答案

(18) C

(19) D

试题 20 分析

软件重用是指在两次或多次不同的软件开发过程中重复使用相同或相似软件元素的过程。按照重用活动是否跨越相似性较少的多个应用领域，软件重用可以区分为横向重用和纵向重用。横向重用是指重用不同应用领域中的软件元素，例如数据结构、分类算法和人机界面构建等。标准函数是一种典型的、原始的横向重用机制。纵向重用是指在一类具有较多公共性的应用领域之间进行软部件重用。纵向重用活动的主要关键点是域分析：根据应用领域的特征及相似性预测软部件的可重用性。

试题 20 答案

(20) C

试题 21 分析

结构化分析方法是一种面向数据流的需求分析方法，其基本思想是自顶向下逐层分解。数据流图是进行结构化分析时所使用的模型，其基本成分包括数据流、加工、数据存储和外部实体。在进行结构化设计时，通过对数据流图进行变换分析和事务分析可以导出程序结构图。

数据库设计可以分为 4 个主要阶段：①用户需求分析。数据库设计人员采用一定的辅助工具对应用对象的功能、性能、限制等要求所进行的科学分析。②概念设计。概念结构设计是对信息分析和定义，如视图模型化、视图分析和汇总。对应用对象精确地抽象、概括而形成的独立于计算机系统的企业信息模型。描述概念模型的较理想的工具是 E-R 图。③逻辑设计。将抽象的概念模型转化为与选用的 DBMS 产品所支持的数据模型相符合的逻辑模型，它是物理设计的基础。包括模式初始设计、子模式设

计、应用程序设计、模式评价及模式求精。④物理设计。逻辑模型在计算机中的具体实现方案。

UML 是面向对象软件的标准化建模语言,其中状态图、活动图、顺序图和通信图可以用来对系统的动态行为进行建模。活动图展现了在系统内从一个活动到另一个活动的流程。活动图强调对象之间的控制流程。在活动图上可以表示分支和汇合。活动图与传统的程序流程图是不等价的。

试题 21 答案

(21) B

试题 22 分析

对许多项目来说,系统软件总需要不断完善,一些需求的改进是合理的而且不可避免,要使得软件需求完全不变更,也许是不可能的,但毫无控制的变更是项目陷入混乱、不能按进度完成或者软件质量无法保证的主要原因之一。

一个好的变更控制过程,给项目风险承担者提供了正式的建议需求变更机制。可以通过需求变更控制过程来跟踪已建议变更的状态,使已建议的变更确保不会丢失或疏忽。在实际中,人们总是希望使用自动工具来执行变更控制过程。有许多人使用商业问题跟踪工具来收集、存储、管理需求变更;可以使用工具对一系列最近提交的变更建议产生一个列表给变更控制委员会开会时做议程用。问题跟踪工具也可以随时按变更状态分类包裹变更请求的数目。

挑选工具时可以考虑以下几个方面:

- ① 可以定义变更请求的数据项。
- ② 可以定义变更请求生存期的状态转换图。
- ③ 可以加强状态转换图使经授权的用户仅能做出所允许的状态变更。
- ④ 记录每一种状态变更的数据,确认做出变更的人员。
- ⑤ 可以定义在提交新请求或请求状态被更新后应该自动通知的设计人员。
- ⑥ 可以根据需要生成标准的或定制的报告和图表。

试题 22 答案

(22) D

试题 23 分析

本题考查黑盒测试,常用的黑盒测试技术包括等价类划分、边值分析、错误推测和因果图等。关于这些技术的详细介绍,请参看“7.1.6 测试与评审”。

试题 23 答案

(23) B

试题 24~25 分析

软件开发环境 (Software Development Environment) 是支持软件产品开发的软件系统。它由软件工具集和环境集成机制构成,前者用来支持软件开发的相关过程、活动和任务;后者为工具集成和软件开发、维护和管理提供统一的支持,它通常包括数

据集成、控制集成和界面集成。数据集成机制提供了存储或访问环境信息库的统一的数据接口规范；界面集成机制采用统一的界面形式，提供统一的操作方式；控制集成机制支持各开发活动之间的通信、切换、调度和协同工作。

试题 24~25 答案

(24) B

(25) D

试题 26 分析

在基于构件的开发中，构件包含并扩展了模块化程序设计中的子程序、面向对象系统中对象或类和系统模型中包的思想，它是系统设计、实现和维护的基础。构件定义为通过接口访问服务的一个独立可交付的功能单元。平时我们所看到的 DLL 文件就是封装好的构件。

试题 26 答案

(26) C

试题 27~28 分析

在基于构件的软件开发中，逻辑构件模型用功能包描述系统的抽象设计，用接口描述每个服务集合，以及功能之间如何交互以满足用户需求，它作为系统的设计蓝图以保证系统提供适当的功能。物理构件模型用技术设施产品、硬件分布和拓扑结构，以及用于绑定的网络和通信协议描述系统的物理设计，这种架构用于了解系统的性能、吞吐率等许多非功能性属性。

试题 27~28 答案

(27) A

(28) B

试题 29 分析

对象管理组织 (OMG) 基于 CORBA 基础设施定义了 4 种构件标准。实体 (Entity) 构件需要长期持久化并主要用于事务性行为，由容器管理其持久化。加工 (Process) 构件同样需要容器管理其持久化，但没有客户端可访问的主键。会话 (Session) 构件不需要容器管理其持久化，其状态信息必须由构件自己管理。服务 (Service) 构件是无状态的。

试题 29 答案

(29) D

试题 30 分析

分布式系统开发分为 5 个逻辑计算层：表示层实现用户界面；表示逻辑层为了生成数据表示而必须进行的处理任务，如输入数据编辑等；应用逻辑层包括为支持实际业务应用和规则所需的应用逻辑和处理过程，如信用检查、数据计算和分析等；数据处理层包括存储和访问数据库中的数据所需的应用逻辑和命令，如查询语句和存储过程等；数据层是数据库中实际存储的业务数据。

试题 30 答案

(30) B

试题 31 分析

系统输入设计中,通常通过内部控制的方式验证输入数据的有效性。数据类型检查确保输入了正确的数据类型;自检位用于对主关键字进行基于校验位的检查;域检查用于验证数据是否位于合法的取值范围;格式检查按照已知的数据格式对照检查输入数据的格式。

试题 31 答案

(31) C

试题 32~33 分析

本题考查测试的相关概念,我们只要了解每一种测试的主要工作,就能解答此题。

恢复测试:恢复测试监测系统的容错能力。检测方法是采用各种方法让系统出现故障,检验系统是否按照要求能从故障中恢复过来,并在约定的时间内开始事务处理,而且不对系统造成任何伤害。如果系统的恢复是自动的(由系统自动完成),需要验证重新初始化、检查点、数据恢复等是否正确。如果恢复需要人工干预,就要对恢复的平均时间进行评估并判断它是否在允许的范围内。

强度测试:是对系统在异常情况下的承受能力的测试,是检查系统在极限状态下运行时,性能下降的幅度是否在允许的范围内。因此,强度测试要求系统在非正常数量、频率或容量的情况下运行。强度测试主要是为了发现在有效的输入数据中可能引起不稳定或不正确的数据组合。例如,运行使系统处理超过设计能力的最大允许值的测试例子;使系统传输超过设计最大能力的数据,包括内存的写入和读出等。

性能测试:检查系统是否满足系统设计方案说明书对性能的要求。性能测试覆盖了软件测试的各阶段,而不是等到系统的各部分所有都组装之后,才确定系统的真正性能。通常与强度测试结合起来进行,并同时软件、硬件进行测试。软件方面主要从响应时间、处理速度、吞吐量、处理精度等方面来检测。

可靠性测试:通常使用以下两个指标来衡量系统的可靠性:平均失效间隔时间(Mean Time Between Failures, MTBF)是否超过了规定的时限,因故障而停机时间(Mean Time To Repairs, MTTR)在一年中不应超过多少时间。

试题 32~33 答案

(32) A

(33) C

试题 34 分析

过程能力成熟度模型(Capability Maturity Model, CMM)在软件开发机构中被广泛用来指导软件过程改进。该模型描述了软件成立能力的5个成熟级别,每一级都包含若干关键过程域(Key Process Areas, KPA)。

CMM的第二级为可重复级,它包括6个关键过程域,分别是:需求管理、软件项目计划、软件项目跟踪和监督、软件分包合同管理、软件质量保证和软件配置管理。

需求管理的目标是为软件需求建立一个基线,提供给软件工程和管理使用;软件计划、产品和活动与软件需求保持一致。

试题 34 答案

(34) A

试题 35~36 分析

UML 有 3 种基本的构造块, 分别是事物(元素)、关系和图。事物是 UML 中重要的组成部分。关系把事物紧密联系在一起。图是很多有相互相关的事物的组。

UML 中的事物也称为建模元素, 包括结构事物、动作事物、分组事物和注释事物。这些事物是 UML 模型中最基本的面向对象的构造块。

① 结构事物。结构事物在模型中属于最静态的部分, 代表概念上等或物理上的元素。总共有 7 种结构事物:

首先是类, 类是描述具有相同属性、方法、关系和语义的对象的集合。一个类实现一个或多个接口。

第 2 种是接口, 接口是指类或组件提供特定服务的一组操作的集合。因此, 一个接口描述了类或组件的对外的可见的动作。一个接口可以实现类或组件的全部动作, 也可以只实现一部分。

第 3 种是协作, 协作定义了交互的操作, 是一些角色和其他元素一起工作, 提供一些合作的动作, 这些动作比元素的总和要大。因此, 协作具有结构化、动作化、维的特性。一个给定的类可能是几个协作的组成部分。这些协作代表构成系统的模式的实现。

第 4 种是用例, 用例是描述一系列的动作, 这些动作是系统对一个特定角色执行, 产生值得注意的结果的值。在模型中用例通常用来组织动作事物。用例是通过协作来实现的。

第 5 种是活动类, 活动类是这种类, 它的对象有一个或多个进程或线程。活动类和类很相像, 只是它的对象代表的元素的行为和其他的元素是同时存在的。

第 6 种是构件, 构件是物理上或可替换的系统部分, 它实现了一个接口集合。在一个系统中, 可能会遇到不同种类的构件, 如 DCOM 或 EJB。

第 7 种是节点, 节点是一个物理元素, 它在运行时存在, 代表一个可计算的资源, 通常占用一些内存和具有处理能力。一个组件集合一般来说位于一个节点, 但有可能从一个节点转到另一个节点。

② 动作事物: 动作事物是 UML 模型中的动态部分。它们是模型的动词, 代表时间和空间上的动作。总共有两种主要的动作事物。

第 1 种是交互(内部活动), 交互是由一组对象之间在特定上下文中, 为达到特定的目的而进行的一系列消息交换而组成的动作。交互中组成动作的对象的每个操作都要详细列出, 包括消息、动作次序(消息产生的动作)、连接(对象之间的连接)。

第 2 种是状态机, 状态机由一系列对象的状态组成。

内部活动和状态机是 UML 模型中最基本的两个动态事物元素, 它们通常和其他的结构元素、主要的类、对象连接在一起。

③ 分组事物。分组事物是 UML 模型中组织的部分，可以把它们看成是个盒子，模型可以在其中被分解。总共只有一种分组事物，称为包。包是一种将有组织的元素分组的机制。结构事物、动作事物甚至其他的分组事物都有可能放在一个包中。与组件（存在于运行时）不同的是包纯粹是一种概念上的东西，只存在于开发阶段。

④ 注释事物。注释事物是 UML 模型的解释部分。

试题 35~36 答案

(35) A

(36) B

试题 37 分析

显然，为了能够精确表达用户与系统的复杂交互过程，应该使用交互图。在 UML 中，交互图包括顺序图、通信图、定时图和交互概览图。顺序图强调消息的时间次序，通信图强调消息流经的数据结构，定时图强调消息跨越不同对象或角色的实际时间，交互概览图是顺序图和活动图的混合体。

试题 37 答案

(37) B

试题 38 分析

本题是一个纯概念题。在面向对象技术中，多态考虑的是类与类之间的层次关系，以及类自身内部特定成员函数之间的关系问题，是解决功能和行为的再抽象问题。多态是指类中具有相似功能的不同函数用同一个名称来实现，从而可以使用相同的调用方式来调用这些具有不同功能的同名函数。这也是人类思维方式的一种直接模拟，例如，一个对象中有很多求两个数最大值的行为，虽然可以针对不同的数据类型，写很多不同名称的函数来实现，但事实上，它们的功能几乎完全相同。这时，就可以利用多态的特征，用统一的标识来完成这些功能。这样，就可以达到类的行为的再抽象，进而统一标识，减少程序中标识符的个数。

试题 38 答案

(38) A

试题 39 分析

在 OOA 中，并不是所有的名词都表示问题域内有用的业务对象，通过删除对象的同义词、系统范围之外的名词、不具有独特行为的名词、不清楚的名词和另一个对象的行动或属性的名词来最终清理候选对象列表。

试题 39 答案

(39) B

试题 40 分析

OOA 基于用例模型，通过对象建模记录确定的对象、对象封装的数据和行为，以及对象之间的关系。OOA 包括 3 个活动，分别是建模系统功能、发现并确定业务对象、

组织对象并确定对象间的关系。

试题 40 答案

(40) C

试题 41 分析

系统测试是将已经确认的软件、计算机硬件、外设和网络等其他因素结合在一起,进行信息系统的各种集成测试和确认测试,其目的是通过与系统的需求相比较,发现所开发的系统与用户需求不符或矛盾的地方。系统测试是根据系统方案说明书来设计测试用例,常见的系统测试主要有功能测试、健壮性测试(可靠性测试)、性能测试、用户界面测试、安全性测试、安装与反安装测试等,其中,最重要的工作是进行功能测试与性能测试。功能测试主要采用黑盒测试方法,性能测试主要验证软件系统在承担一定负载的情况下所表现出来的特性是否符合客户的需要,主要指标有响应时间、吞吐量、并发用户数和资源利用率等。

从以上描述可以得知本题应选 A。A 选项所描述的路径测试是在单元测试阶段进行的一种白盒测试。

试题 41 答案

(41) A

试题 42 分析

软件测试是为了发现错误而执行程序的过程。黑盒测试也称为功能测试,这种测试将待测试的模块(软件)当成一个黑色的盒子,测试人员不清楚模块的内部设计,仅知道模块的功能。所以只是根据规格说明所规定的功能来设计测试用例,它不考虑程序的内部结构和处理过程。常用的黑盒测试技术有等价类划分、边值分析、错误猜测和因果图等。

试题 42 答案

(42) B

试题 43 分析

复杂系统的复杂之处主要在于其各子系统之间关联的复杂性。例如,人体本身就是一个复杂系统。虽然骨骼系统、神经系统、消化系统和血液循环系统等都有清晰的结构,可以清晰地描述其性能,但各子系统之间相互关联的机制却仍难以把握。

试题 43 答案

(43) A

试题 44 分析

静态分析通过解析程序文本从而识别出程序语句的各个部分,审查可能的缺陷和异常之处,静态分析包括五个阶段:控制流分析阶段找出并突出显示那些带有多重出口或入口的循环,以及不可达到的代码段;数据使用分析阶段突出程序中变量的使用

情况；接口分析阶段检查子程序和过程说明及它们使用的一致性；信息流分析阶段找出输入变量和输出变量之间的依赖关系；路径分析阶段找出程序中所有可能的路径并画出在此路径中执行的语句。

试题 44 答案

(44) D

试题 45 分析

本题考查软件维护相关知识。

软件维护的类型包括：改正性维护（正确性维护）、适应性维护、完善性维护、预防性维护。

改正性维护：在软件交付使用后，必然会有一部分隐藏的错误被带到运行阶段来。这些隐藏下来的错误在某些特定的使用环境下就会暴露出来。为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应当进行的诊断和改正错误的过程，就叫做改正性维护。

适应性维护：随着计算机的飞速发展，外部环境（新的硬、软件配置）或数据环境（数据库、数据格式、数据输入/输出方式、数据存储介质）可能发生变化，为了使软件适应这种变化，而去修改软件的过程就叫做适应性维护。

完善性维护：在软件的使用过程中，用户往往会对软件提出新的功能与性能要求。为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。这种情况下进行的维护活动叫做完善性维护。

预防性维护：为了提高软件的可维护性、可靠性等而提出的一种维护类型，它为以后进一步改进软件打下了良好的基础。通常，预防性维护定义为：“把今天的方法学用于昨天的系统以满足明天的需要”。也就是说，采用先进的软件工程方法对需要维护的软件或软件中的某一部分（重新）进行设计、编制和测试。

试题 45 答案

(45) C

8

第 8 章

软件架构设计

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

- 软件架构的概念。
- 软件架构的风格。
- 特定领域软件架构。
- 基于架构的软件开发方法。
- 软件架构评估。
- 软件产品线。
- 设计模式。

8.1 考点突破

从历年的考试情况来看，本章的考点主要集中在以下方面：基于构件的开发模型、架构风格、架构评估、架构与质量属性、设计模式。

8.1.1 历年考试情况分析

在历年的考试试题中，有关软件架构设计知识的试题如表 8-1 所示。

表 8-1 软件架构设计知识试题分布表

题号	2009.11	2010.11	2011.11
30	设计模式（Command 模式）		
31			
32			
33		工厂方法	Composite 模式

(续表)

题号	2009.11	2010.11	2011.11
34		工厂方法	Composite 模式
35	基于构件的开发模型	构件技术	
36		构件技术	
37			
38	构件技术	CORBA	
39	MVC 模式	分层架构	
40		客户机/服务器	
41	网络架构数据流图		
44		架构概念	架构风格
45	架构设计所处阶段	架构概念	设计模式
46	架构设计基本概念	ABSD	“4+1” 视图
47	软件架构需求		
48	ABSD		
49	软件架构文档	架构评审	ABSD
50	架构复审	过程控制风格	黑板风格
51	事件驱动与虚拟机风格	隐式调用风格	解释器风格
52		黑板风格	隐式调用风格
53	管道-过滤器风格	DSSA	中介者模式
54	隐式调用风格		命令模式
55	闭环控制架构	架构与质量属性	责任链模式
56	数据共享为中心的架构风格	架构与质量属性	架构与质量属性
57	架构模式	架构与质量属性	
58		架构与质量属性	架构与质量属性
59		责任链模式	
60	装饰模式	中介者模式	
61	访问者模式	代理模式	ATAM
62	架构评估方法 (ATAM)	架构评估	敏感点与风险点
63			

按照知识点进行总结和归类的试题分布情况如表 8-2 所示。

表 8-2 软件架构设计知识归类表

知识点	2009.11	2010.11	2011.11
软件架构的概念	4	2	0
软件架构的风格	9	4	4
分层架构	0	2	0
面向服务的架构 (SOA)	0	0	0
特定领域软件架构	0	2	0
软件产品线	0	0	0
基于架构的软件开发方法	5	6	4

(续表)

知识点	2009.11	2010.11	2011.11
软件架构与质量属性	0	4	6
软件架构评估	3	3	3
设计模式	5	5	6
其他	1	0	0
合计	27	28	23

从表 8-2 中可以看出, 软件架构设计知识方面的内容在历年的考试中所占分值比例是最高的, 达到了 27~28 分, 趋势如图 8-1 所示。

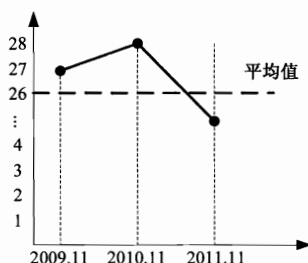


图 8-1 软件架构设计知识历年试题比例趋势图

8.1.2 软件架构的概念

架构一词并非计算机行业提出的, 早在 17 世纪, 就有人提出了架构的概念。这就是著名的“瑞典瓦萨战舰事件”。

17 世纪 20 年代, 瑞典与波兰交战, 当时的瑞典国王决心快速结束战争, 于是下令建造一艘史无前例的巨型战舰。战舰长 70 米, 载员 300 人, 两层甲板上共装 64 门重炮, 有着无可匹敌的火力。在著名的舰船设计师的精心设计, 造船工人的努力之下, 战舰终于完成。但遗憾的是该无敌战舰下水仅几十分钟, 由于鸣放了几响礼炮, 导致船身失去平衡而沉没, 当时船上 150 人有数十人丧生。巨舰的沉没带给人们很多的思索, 有人提出战舰的架构不合理, 其架构不足以支撑 64 门重炮, 而导致了悲剧的发生。

其实软件开发的发展过程中, 同样经历了“瑞典瓦萨战舰事件”。最初的软件开发是无章可循的, 全靠程序员技术水平来完成相应的系统开发。后来发现这种做法在项目一步步变大时, 风险会越来越高(项目失败率高), 所以提出了软件开发方法与软件开发模型。当模型得到广泛应用以后, 人们发现仅从需求指导设计的方式, 缺乏对系统的全盘把握(当软件进入运维期以后, 增加功能变得很复杂, 或是增加了过多的功能以后系统不堪重负而崩溃), 同时需求与设计之间有着一道鸿沟, 所以提出软件架构。

纵观软件架构技术的发展过程, 从最初的无结构设计到现行的基于架构的软件开发, 可以认为经历了 4 个阶段:

① 无架构设计阶段。以汇编语言进行小规模应用程序开发为特征。

② 萌芽阶段。出现了程序结构设计主题, 以控制流图和数据流图构成软件结构为特征。

③ 初级阶段。出现了从不同侧面描述系统的结构模型，以 UML 为典型代表。

④ 高级阶段。以描述系统的高层抽象结构为中心，不关心具体的建模细节，划分了架构模型与传统软件结构的界限，该阶段以 Kruchten 提出的“4+1”模型为标志。

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，由构件的描述、构件的相互作用（连接件）、指导构件集成的模式及这些模式的约束组成。软件架构不仅指定了系统的组织结构和拓扑结构，并且显示了系统需求和构件之间的对应关系，提供了一些设计决策的基本原理。

软件架构虽脱胎于软件工程，但其形成同时借鉴了计算机架构和网络架构中很多宝贵的思想和方法。近年来，软件架构已完全独立于软件工程，成为计算机科学的一个最新的研究方向和独立学科分支。软件架构研究的主要内容涉及软件架构描述、软件架构风格、软件架构评估和软件架构的形式化方法等。解决好软件的复用、质量和维护问题，是研究软件架构的根本目的。

8.1.3 软件架构的意义

对于软件项目的开发来说，一个清晰的软件架构是首要的。鉴于架构的重要性，Perry 将软件架构视为软件开发中第一类重要的设计对象，Barry Boehm 也明确指出：“在没有设计出架构及其规则时，那么整个项目不能继续下去，而且架构应该看做是软件开发中可交付的中间产品”。由此可见，架构在软件开发中为不同的人员提供了共同交流的语言，体现并尝试了系统早期的设计决策，并作为系统设计的抽象，为实现框架和构件的共享和重用、基于架构的软件开发提供了有力的支持。

1. 架构是项目干系人进行交流的手段

软件架构代表了系统的高层抽象，项目干系人能将它作为建立一个互相理解的基础，形成统一认识，互相交流。不同的项目干系人关心着系统的不同方面，而这些方面都受架构的影响，因此，架构可能是所有项目干系人都关心的一个重要因素。例如，用户关心系统是否满足可用性和可靠性需求；客户关心的是系统能否在规定时间内完成，并且开支在预算范围内；管理人员担心在经费支出和进度条件下，按此架构能否使开发团队成员在一定程度上独立开发，各部分的交互是否遵循统一的规范，开发进度是否可控；开发人员关心的是如何才能实现架构的各目标。

2. 架构是早期设计决策的体现

软件架构体现了系统的最早的一组设计决策，这些早期的约束比起以后的开发、设计、编码或运行及维护阶段的工作重要得多，对系统生命周期的影响也大得多。早期决策的正确性最难以保证，而且这些决策也最难以改变，影响范围也最大。

3. 架构明确了对系统实现的约束条件

所谓“实现”就是要用实体来显示出一个软件架构，即要符合架构所描述的结构性设计决策，分割成规定的构件，按规定方式互相交互。在具体实现时，必须按照架构的设计，将系统分成若干个组成部分，各部分必须按照预定的方式进行交互，而且每个部分也必须具有架构中所规定的外部特征。这些约束是在系统级或项目范围内做

出的，每个构件上工作的实现者是看不见的。这样一来，可以分离关注点，架构设计师不必是算法设计者或精通编程语言，他们只需重点考虑系统的总体权衡（tradeoff）问题，而构件的开发人员在架构给定的约束下进行开发。

4. 架构决定了开发和维护组织的组织结构

架构包含了对系统的最高层次的分解，因此一般被作为任务划分结构的基础。任务划分结构又规定了计划、调度及预算的单位，决定了开发小组内部交流的渠道、配置控制和文件系统的组织、集成与测试计划和过程等。各开发小组按照架构中对各主要构件接口的规定进行交流。一旦进入维护阶段，维护活动也会反映出软件架构，常由不同的小组分别负责对各具体部分的维护。

5. 架构制约着系统的质量属性

小的软件系统可以通过编程或调试措施来达到质量属性的要求，而随着软件系统规模的扩大，这种技巧也将越来越无法满足要求。因为在大型软件系统中，质量属性更多地是由系统结构和功能划分来实现的，而不再主要依靠所选用的算法或数据结构。可以使用对架构的评价来预测系统未来的质量属性，架构评估技术可以对按某架构开发出来的软件产品的质量及缺陷做出比较准确的预测。

6. 架构使推理和控制更改更简单

在整个软件生命周期内，每个架构都将更改划分为3类，分别是局部的、非局部的和架构级的变更。局部变更是最经常发生的，也是最容易进行的，只需修改某一个构件就可以实现。非局部变更的实现则需对多个构件进行修改，但并不改动软件架构。架构级的变更是指会影响各部分的相互关系，甚至要改动整个系统。所以，一个优秀的架构应该能使更改简单易行。

7. 架构有助于循序渐进的原型设计

一旦确定了架构，就可以对其进行分析，并将其按可执行模型来构造原型，以减少项目开发的潜在风险。

8. 架构可以作为培训的基础

在对项目组新成员介绍所开发的系统时，可以首先介绍系统的架构，以及对构件之间如何交互从而实现系统需求的高层次的描述，让项目新成员很快进入角色。

9. 架构是可传递和可复用的模型

软件架构体现了一个相对来说比较小又可理解的模型。软件架构级的复用意味着架构的决策能在具有相似需求的多个系统中发生影响，这比代码级的复用要有更大的好处。通过对架构的抽象，架构设计师能够对一些经过实践证明是非常有效的架构进行复用，从而提高设计的效率和可靠性。

8.1.4 软件架构的风格

Garlan 和 Shaw 对通用软件架构风格进行了分类，他们将软件架构分为数据流风格、调用/返回风格、独立构件风格、虚拟机风格和仓库风格。

1. 数据流风格

数据流风格包括批处理序列和管道/过滤器两种风格。

(1) 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。

(2) 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。管道/过滤器风格如图 8-2 所示。

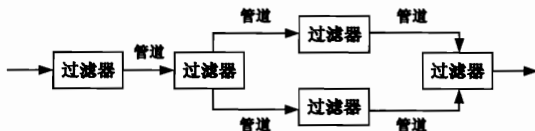


图 8-2 管道过滤器风格

管道/过滤器风格的软件体系结构具有许多很好的特点：

- 使得软构件具有良好的隐蔽性和高内聚、低耦合的特点。
- 允许设计者将整个系统的输入/输出行为看成是多个过滤器的行为的简单合成。
- 支持软件重用。只要提供适合在两个过滤器之间传送的数据，任何两个过滤器都可被连接起来。
- 系统维护和增强系统性能简单。新的过滤器可以添加到现有系统中来；旧的可以被改进的过滤器替换掉。
- 允许对一些如吞吐量、死锁等属性的分析。
- 支持并行执行。每个过滤器是作为一个单独的任务完成，因此可与其他任务并行执行。

但是，这样的系统也存在着若干不利因素。

- 通常导致进程成为批处理的结构。这是因为虽然过滤器可增量式地处理数据，但它们是独立的，所以设计者必须将每个过滤器看成一个完整的从输入到输出的转换；
- 不适合处理交互的应用。当需要增量地显示改变时，这个问题尤为严重；
- 因为在数据传输上没有通用的标准，每个过滤器都增加了解析和合成数据的工作，这样就导致了系统性能下降，并增加了编写过滤器的复杂性。

2. 调用/返回风格

调用/返回风格包括主程序/子程序、数据抽象和面向对象，以及层次结构。

(1) 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。

(2) 数据抽象和面向对象

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。

面向对象的系统有许多优点，并早已为人所知：

- 因为对象对其他对象隐藏它的表示，所以可以改变一个对象的表示，而不影响其他的对象。
- 设计者可将一些数据存取操作的问题分解成一些交互的代理程序的集合。

但是，面向对象的系统也存在着某些问题：

- 为了使一个对象和另一个对象通过过程调用等进行交互，必须知道对象的标识。只要一个对象的标识改变了，就必须修改所有其他明确调用它的对象。
- 必须修改所有显式调用它的其他对象，并消除由此带来的一些副作用。例如，如果 A 使用了对象 B，C 也使用了对象 B，那么，C 对 B 的使用所造成的对 A 的影响可能是料想不到的。

(3) 层次结构

层次系统的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。该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。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。在层次结构中，修改某一层，最多影响其相邻的两层（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。例如，网络通信协议和操作系统就属于层次结构。

层次系统有许多可取的属性：

- 支持基于抽象程度递增的系统设计，使设计者可以把一个复杂系统按递增的步骤进行分解。
- 支持功能增强，因为每一层至多和相邻的上下层交互，因此功能的改变最多影响相邻的上下层。
- 支持重用。只要提供的服务接口定义不变，同一层的不同实现可以交换使用。这样，就可以定义一组标准的接口，而允许各种不同的实现方法。

但是，层次系统也有其不足之处：

- 并不是每个系统都可以很容易地划分为分层的模式，甚至即使一个系统的逻辑结构是层次化的，出于对系统性能的考虑，系统设计师不得不把一些低级或高级的功能综合起来。

- 很难找到一个合适的、正确的层次抽象方法。

3. 独立构件风格

独立构件风格包括进程通信和事件驱动的系统。

(1) 进程通信

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、异步或同步方式，以及远程过程（方法）调用等。

(2) 事件驱动的系统

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（Implicit Invocation）来实现的。

隐式调用系统的主要优点如下：

- 为软件重用提供了强大的支持。当需要将一个构件加入现存系统中时，只需将它注册到系统的事件中。
- 为改进系统带来了方便。当用一个构件代替另一个构件时，不会影响到其他构件的接口。

隐式调用系统的主要缺点如下：

- 构件放弃了对系统计算的控制。一个构件触发一个事件时，不能确定其他构件是否会响应它。而且即使它知道事件注册了哪些构件的过程，它也不能保证这些过程被调用的顺序。
- 数据交换的问题。有时数据可被一个事件传递，但另一些情况下，基于事件的系统必须依靠一个共享的仓库进行交互。在这些情况下，全局性能和资源管理便成了问题。
- 既然过程的语义必须依赖于被触发事件的上下文约束，关于正确性的推理存在问题。

4. 虚拟机风格

虚拟机风格包括解释器和基于规则的系统。

(1) 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。

(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/数据选择器和工作内存，一般用在人工智能领域和 DSS 中。

5. 仓库风格

仓库风格包括数据库系统、黑板系统和超文本系统。

(1) 数据库系统

数据库系统是仓库风格最常见的形式。在数据库系统中,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。

(2) 黑板系统

黑板系统包括知识源、黑板和控制 3 个部分。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,例如,信号处理、问题规划和编译器优化等。

(3) 超文本系统

超文本系统中出现的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,它以节点为基本单位,链作为节点之间的联想式关联。超文本系统通常应用在互联网领域。

8.1.5 分层架构

在 IT 发展过程中,网络计算经历了从集中式计算模型到分布式计算模型的演变。在集中式计算技术时代,广泛使用的是大型机(或小型机)计算模型。它是通过一台物理上与宿主机相连接的非智能终端来实现宿主机上的应用程序。在多用户环境中,宿主机应用程序既负责与用户的交互,又负责对数据的管理。集中式的系统使用户能共享贵重的硬件设备,例如,磁盘、打印机和调制解调器等。但随着用户的增多,对宿主机能力的要求很高,而且开发人员必须为每个新的应用重新设计同样的数据管理构件。

20 世纪 80 年代以后,集中式结构逐渐被以 PC 为主的计算机网络所取代。PC 和工作站的采用,永远改变了协作计算模型,导致了分布式计算模型的产生。一方面,由于大型机系统固有的缺陷(如缺乏灵活性),无法适应信息量急剧增长的需求,并为整个企业提供全面的解决方案;另一方面,由于微处理器的日新月异,其强大的处理能力和低廉的价格使计算机网络迅速发展,用户可以选择适合自己需要的工作站、操作系统和应用程序。

1. 二层架构

客户机/服务器(Client/Server, C/S)架构是基于资源不对等,且为实现共享而提出来的,是 20 世纪 90 年代成熟起来的技术,C/S 架构定义了工作站(客户应用程序)如何与服务器相连,以实现数据和应用分布到多台计算机上。服务器负责有效地管理系统的资源,其主要任务集中于对 DBMS 的管理和控制,以及数据的备份与恢复;客户应用程序的主要任务是提供用户与数据库交互的界面,向服务器提交用户请求并接收来自服务器的信息,对存在于客户端的数据执行应用逻辑要求。这是一种“胖客户机(Fat Client)、瘦服务器(Thin Server)”的架构,其处理流程如图 8-3 所示。

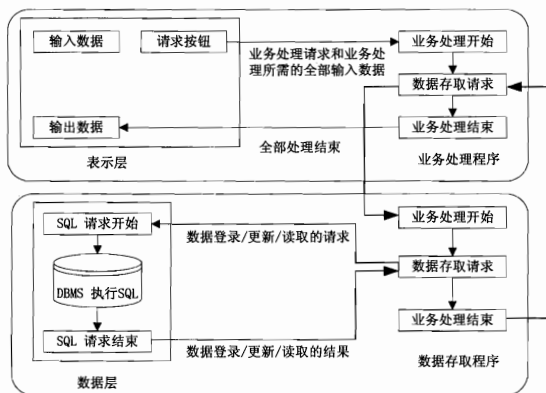


图 8-3 C/S 架构的一般处理流程

与集中式系统相比，C/S 架构的优点主要在于，系统的客户应用程序和服务器构件分别运行在不同的计算机上，系统中每台服务器都可以适合各构件的要求，这对于硬件和软件的变化显示出极大的适应性和灵活性，而且易于对系统进行扩充和缩小。在 C/S 架构中，系统中的功能构件充分隔离，客户应用程序的开发集中于数据的显示和分析，而服务器的开发则集中于数据的管理，不必在每一个新的应用程序中都要对一个 DBMS 进行编码。将大的应用处理任务分布到许多通过网络连接的低成本计算机上，以节约大量费用。

C/S 架构具有强大的数据操作和事务处理能力，模型思想简单，易于人们理解和接受。但随着企业规模的日益扩大，软件的复杂程度不断提高，C/S 架构逐渐暴露出以下缺点：

- 开发成本较高。C/S 架构对客户端软硬件配置要求较高，尤其是软件的不断升级，对硬件要求不断提高，增加了整个系统的成本。
- 客户端程序设计复杂。采用 C/S 架构进行软件开发，大部分工作量放在客户端的程序设计上，客户端显得十分庞大。
- 用户界面风格不一，使用繁杂，不利于推广使用。
- 软件移植困难。采用不同开发工具或平台开发的软件，一般互不兼容，不能或很难移植到其他平台上运行。
- 软件维护和升级困难。采用 C/S 架构的软件要升级，开发人员必须到现场为客户机升级，每个客户机上的软件都需维护。对软件的一个小小改动，每一个客户端都必须更新。
- 新技术不能轻易应用。因为一个软件平台及开发工具一旦选定，不可能轻易更改。
- 可扩展性差。C/S 架构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，软硬件的组合和集成能力有限。客户机的负荷太重，难以管理大量的客户机，系统的性能容易变坏。
- 系统安全性难以保证。因为客户端程序可以直接访问数据库服务器，那么，在客户端计算机上的其他程序也可想办法访问数据库服务器，从而使数据库的安全性受到威胁。

正是因为 C/S 架构有这么多缺点，因此，三层 C/S 架构应运而生。为了区分，而

把传统的 C/S 架构称为二层 C/S 架构。

2. 三层 C/S 架构

与二层 C/S 架构相比,在三层 C/S 架构中,增加了一个应用服务器。可以将整个应用逻辑驻留在应用服务器上,而只有表示层存在于客户机上。这种客户机称为瘦客户机(Thin Client)。三层 C/S 架构将应用系统分成表示层、功能层和数据层 3 个部分,如图 8-4 所示。

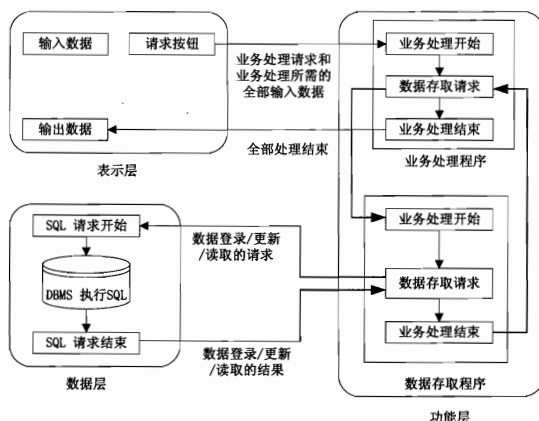


图 8-4 三层 C/S 架构的一般处理流程

(1) 表示层

表示层是系统的用户接口部分,担负着用户与系统之间的对话功能。它用于检查用户从键盘等输入的数据,显示输出的数据。为使用户能直观地进行操作,一般要使用图形用户界面,操作简单、易学易用。在变更用户界面时,只需改写显示控制和数据检查程序,而不影响其他两层。检查的内容也只限于数据的形式和取值的范围,不包括有关业务本身的处理逻辑。

(2) 功能层

功能层也称为业务逻辑层,是将具体的业务处理逻辑编入程序中。例如,在制作订购合同时计算合同金额、按照预定的格式配置数据、打印订购合同,而处理所需的数据则要从表示层或数据层取得。

(3) 数据层

数据层相当于二层 C/S 架构中的服务器,负责对 DBMS 的管理和控制。

三层 C/S 架构对这三层进行明确分割,并在逻辑上使其独立。在二层 C/S 架构中,数据层作为 DBMS 已经独立出来,所以,三层 C/S 架构的关键是要将表示层和功能层分离成各自独立的程序,并且还要使这两层间的接口简洁、明了。通常的做法是只将表示层配置在客户机中,如图 8-5 (a) 或 8-5 (b) 所示。如果像图 8-5 (c) 所示的那样连功能层也放在客户机中,与二层 C/S 架构相比,其程序的可维护性要好得多,但是其他问题并未得到解决。客户机的负荷太重,其业务处理所需的数据要从服务器传给客户机,所以系统的性能容易变坏。

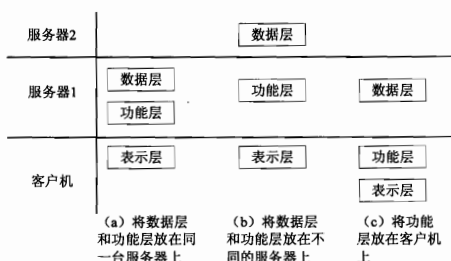


图 8-5 三层 C/S 架构的物理部署

如果将功能层和数据层分别放在不同的服务器中，如图 8-5 (b) 所示，则服务器之间也要进行数据传送。由于三层是分别放在各自不同的硬件系统上的，所以灵活性很高，能够适应客户机数目的增加和处理负荷的变动。例如，在追加新业务处理时，可以相应增加装载功能层的服务器（应用服务器）。因此，系统规模越大，这种形态的优点就越显著。

与传统的二层架构相比，三层 C/S 架构具有以下优点：

- 允许合理地划分三层的功能，使之在逻辑上保持相对独立性，从而使整个系统的逻辑结构更为清晰，能提高系统的可维护性和可扩展性。
- 允许更灵活、有效地选用相应的平台和硬件系统，使之在处理负荷能力上与处理特性上分别适应于结构清晰的三层，并且这些平台和各个组成部分可以具有良好的可升级性和开放性。
- 系统的各层可以并行开发，各层也可以选择各自最适合的开发语言，使之能并行且高效地进行开发，达到较高的性能价格比。对每一层的处理逻辑的开发和维护也会更容易些。
- 利用功能层可以有效地隔离表示层与数据层，未授权的用户难以绕过功能层而利用数据库工具或黑客手段去非法地访问数据层，这就为严格的安全管理奠定了坚实的基础。

但是，三层 C/S 架构各层间的通信效率若不高，即使分配给各层的硬件能力很强，其作为整体来说也达不到所要求的性能。此外，设计时必须慎重考虑三层间的通信方法、通信频度和数据量，这是三层 C/S 架构设计的关键问题。

3. B/S 架构

浏览器/服务器（Browser/Server, B/S）架构是三层 C/S 架构的一种实现方式，其具体结构为“浏览器/Web 服务器/数据库服务器”。B/S 架构利用不断成熟的 WWW 浏览器技术，结合浏览器的多种脚本语言，用通用浏览器就实现了原来需要复杂的专用软件才能实现的强大功能，并节约了开发成本。从某种程度上来说，B/S 架构是一种全新的软件架构。

在 B/S 架构中，除了数据库服务器外，应用程序以网页形式存放于 Web 服务器上，用户运行某个应用程序时，只需在客户端的浏览器中键入相应的网址，调用 Web 服务器上的应用程序，并对数据库进行操作，完成相应的数据处理工作，最后将结果通过浏览器显示给用户。基于 B/S 架构的软件，系统安装、修改和维护全在服务器端解决。用户在使用系统时，仅仅需要一个浏览器就可运行全部的模块，真正达到了“零客户

端”的功能，很容易在运行时自动升级。

但是，与 C/S 架构相比，B/S 架构也有许多不足之处，例如，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；安全性难以控制；采用 B/S 架构的系统，在数据查询等响应速度上，要远远地低于 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。

8.1.6 面向服务的架构（SOA）

面向服务的架构（Service Oriented Architecture, SOA）是一种新的架构风格，它具有松耦合和面向软件服务的特点，具有很高的重用性和灵活性。而要实现 SOA 的这些优点，Web Service 在其中扮演了重要的角色。

1. SOA 的概念

由于 SOA 是一个比较新的概念，不同厂商对 SOA 的定义也不一样，下面列举几个主要的定义。

- W3C: SOA 是指服务提供者完成一组工作，为服务使用者交付所需的最终结果。最终结果通常会使使用者的状态发生变化，但也可能使提供者的状态改变，或者双方都产生变化。
- Service-architecture.com: SOA 本质上是服务的集合，服务间彼此通信，这种通信可能是简单的数据传送，也可能是两个或更多的服务协作进行某些活动。服务间需要某些方法进行连接。所谓服务就是精确定义、封装完善、独立于其他服务所处环境和状态的函数。
- Looselycoupled.com: SOA 就是按需连接资源的系统。在 SOA 中，资源被作为可通过标准方式访问的独立服务，提供给网络中的其他成员。与传统的系统架构相比，SOA 规定了资源间更为灵活的松散耦合关系。
- Gartner: SOA 是一种 C/S 模式的软件设计方法，在 SOA 中，一项应用由软件服务和软件服务使用者组成。SOA 与大多数通用的 C/S 模型的不同之处，在于它着重强调软件构件的松散耦合，并使用独立的标准接口。
- META: SOA 是一种以通用为目的、可扩展、具有联合协作性的架构，所有流程都被定义为服务，服务通过基于类封装的服务接口委托给服务提供者，服务接口根据可扩展标识符、格式和协议单独描述。

从上述不同的定义中，可以得出 SOA 的几个关键特性：一种粗粒度、松耦合服务架构，服务之间通过简单、精确定义接口进行通信，不涉及底层编程接口和通信模型。

在 SOA 中，服务是封装成用于业务流程的可重用构件的 API。它提供信息或简化业务数据从一个有效的、一致的状态向另一个状态的转变。用于实现特定服务的流程并不重要，只要它响应用户的命令并为用户的请求提供高质量的服务就可以了。

通过定义的通信协议，可以调用服务来强调互操作性和位置透明性。一个服务表现为一个软件构件，因为从服务请求者的角度来看，它就像是一个自包含的函数。然而，实际上，服务的实现可能包括在一个企业内部的不同计算机上或者许多业务合作伙伴拥有的计算机上执行的很多步骤。就封装的软件而言，服务可能是一个构件，也

可能不是一个构件。如同类对象，请求者应用程序能够将服务看做是一个整体。

SOA 并不是一种现成的技术，而是一种架构和组织 IT 基础结构及业务功能的方法。SOA 是一种在计算环境中设计、开发、部署和管理离散逻辑单元（服务）的模型。

SOA 要求开发人员将应用设计为服务的集合，要求开发人员跳出应用本身进行思考，考虑现有服务的重用，或思索他们的服务如何能够被其他项目重用。但是，SOA 并不仅仅是一种开发方法，它还具有管理上的优点。例如，现在管理员可直接管理开发人员所构建的相同服务，这远胜于以往管理单个应用的方式。通过分析服务间的交互，SOA 可以帮助企业了解何时及为什么业务逻辑被切实执行了，这使管理员或分析师能够有针对性地优化业务流程。

2. SOA 的特征

实施 SOA 的关键目标是实现企业 IT 资产重用的最大化。要实现这一目标，就要在实施 SOA 的过程中牢记以下特征。

（1）可从企业外部访问

商业合作伙伴（外部用户）也能像企业内部用户一样访问相同的服务，商业合作伙伴采用 B2B（Business to Business）协议（例如，ebXML、RosettaNet 等）相互合作。当商业合作伙伴基于业务目的交换业务信息时，他们就参与了一次会话。会话是商业合作伙伴之间一系列的一条或多条业务信息的交换。会话类型（会话复杂或简单、长或短等）取决于业务目的。

除了 B2B 协议外，外部用户还可以访问以 Web 服务方式提供的企业服务。

（2）随时可用

当有服务使用者请求服务时，SOA 要求必须有服务提供者能够响应。大多数 SOA 都能够为门户应用之类的同步应用和 B2B 之类的异步应用提供服务。同步应用对于其所使用的服务具有很强的依赖性。

许多同步应用通常部署在前台，其最终用户很容易受到服务提供者短缺的影响。很多情况下，同步应用利用分布式服务提供者，这样可以响应更多的用户请求。但是，随着提供特定服务功能的服务器数量的增长，出现短缺的可能性也呈指数级上升。

相比之下，异步应用更为稳健，因为其采用队列请求设计，因此，可以容许出现服务提供者短缺或迟滞的情况。异步应用大多数情况下部署在后台，用户通常不会觉察到短暂的短缺。大部分情况下异步应用能够稳健应对短时间短缺，但是长时间短缺则会引发严重问题。在服务短缺解决、队列引擎将罕见的大量工作推到共享的应用资源中时，可能会出现队列溢出甚至服务死锁。

服务使用者要求提供同步服务时，通常是基于其自身理解或使用习惯。在多数情况下，采用异步模型可以达到同样的效果，但更能够体现 SOA 的最佳特性。

当然，并不是所有情况下都应当采用异步设计模式。但大多数情况下，异步消息可以确保系统在不同负荷下的伸缩性，在接口响应时间不是很短时尤其如此。

（3）粗粒度服务接口

粗粒度服务提供一项特定的业务功能，而细粒度服务代表了技术构件方法。例如，

向希赛教育的视频点播系统中添加一个客户是典型的粗粒度服务，而用户可以使用几个细粒度服务实现同一功能，例如，将客户名加入到点播系统中、添加详细的客户联系方式、添加点播信息等。

采用粗粒度服务接口的优点在于使用者和服务层之间不必再进行多次的往复，一次往复就足够。Internet 环境中有保障的 TCP/IP 会话已不再占据主导，建立连接的成本也过高，因此，在该环境中进行应用开发时粗粒度服务接口的优点更为明显。

除去基本的往复效率，事务稳定性问题也很重要。在一个单独事务中包含的多段细粒度请求可能使事务处理时间过长、导致后台服务超时，从而中止。与此相反，从事务的角度来看，向后台服务请求大块数据可能是获取反馈的唯一途径。

(4) 分级

一个关于粗粒度服务的争论是此类服务比细粒度服务的重用性差，因为粗粒度服务倾向于解决专门的业务问题，因此，通用性差、重用性设计困难。解决该争论的方法之一就是允许采用不同的粗粒度等级来创建服务。这种服务分级包含了粒度较细、重用性较高的服务，也包含粒度较粗、重用性较差的服务。

在服务分级方面，必须注意服务层的公开服务通常由后台系统或 SOA 平台中现有的本地服务组成。因此，允许在服务层创建私有服务是非常重要的。正确的文档、配置管理和私有服务的重用对于 IT 部门在 SOA 服务层快速开发新的公开服务的能力具有重要影响。

(5) 松散耦合

SOA 具有松散耦合构件服务，这一点区别于大多数其他的架构。该方法旨在将服务使用者和服务提供者在服务实现和客户如何使用服务方面隔离开来。

服务提供者和服务使用者间松散耦合背后的关键点是，服务接口作为与服务实现分离的实体而存在，这使得服务实现能够在完全不影响服务使用者的情况下进行修改。

大多数松散耦合方法都依靠基于服务接口的消息。基于消息的接口能够兼容多种传输方式（例如，HTTP、JMS、TCP/IP、MOM 等）。基于消息的接口可以采用同步和异步协议实现，Web 服务对于 SOA 服务接口来讲是一个重要的标准。

当使用者调用一个 Web 服务时，被调用的对象可以是 CICS 事务、DCOM 或 CORBA 对象、J2EE EJB 或 Tuxedo 服务等，但这与服务使用者无关，底层实现并不重要。

(6) 可重用的服务及服务接口设计管理

如果完全按照可重用的原则设计服务，SOA 将可以使应用变得更为灵活。可重用服务采用通用格式提供重要的业务功能，为开发人员节约了大量时间。设计可重用服务是与数据库设计或通用数据建模类似的最有价值的工作。由于服务设计是成功的关键因素，因此 SOA 实施者应当寻找一种适当的方法进行服务设计过程管理。

服务设计管理从根本上讲是服务设计问题，服务设计需要在两点间折中——走捷径的项目战术与企业构建可重用通用服务的长期目标。

超越项目短期目标进行服务接口的开发和评估是迈向精确定义服务接口的重要一步，同时还需要为接口文档、服务实现文档及所有重要的非功能性特征设立标准。

在大型组织中实现重用的一个先决条件是建立通用（设计阶段）服务库和开发流程，以保证重用的正确性和通用性。此外，对记述服务设计和开发的服务文档进行评估也是成功利用服务库的关键。

简言之，不按规则编写服务将无法保证可提供重用性的 SOA 的成功实施。在执行规则的过程中会产生财务费用，需要在制定 SOA 实施计划时加以考虑。

（7）标准化的接口

XML 和 Web Service 将 SOA 推向更高的层面，并大大提升了 SOA 的价值。尽管以往的 SOA 产品都是专有的、并且要求 IT 部门在其特定环境中开发所有应用，但 XML 和 Web Service 标准化的开放性使企业能够在所部署的所有技术和应用中采用 SOA。

Web Service 使应用功能得以通过标准化接口（WSDL）提供，并可基于标准化传输方式（HTTP 和 JMS）、采用标准化协议（SOAP）进行调用。例如，开发人员可以采用最适于门户开发的工具轻松创建一个新的门户应用，并可以重用 ERP 系统和定制化 J2EE 应用中的现有服务，而完全无须了解这些应用的内部工作原理。采用 XML，门户开发人员无须了解特定的数据表示格式，便能够在这些应用间轻松地交换数据。

当然，用户也可以不采用 Web 服务或 XML 来创建 SOA 应用，但是，这两种标准的重要性日益增加、应用日趋普遍，大多数的服务使用者都会将其作为企业的服务访问方法。

（8）支持各种消息模式

SOA 中可能存在以下消息模式。在一个 SOA 实现中，常会出现混合采用不同消息模式的服务。

- 无状态的消息：服务使用者向服务提供者发送的每条消息都必须包含服务提供者处理该消息所需的全部信息。这一限定使服务提供者无须存储服务使用者的状态信息，从而更易扩展。
- 有状态的消息：服务使用者与服务提供者共享服务使用者的特定环境信息，此信息包含在服务提供者和服务使用者交换的消息中。这一限定使服务提供者与服务使用者之间的通信更加灵活，但由于服务提供者必须存储每个服务使用者的共享环境信息，因此其整体可扩展性明显减弱。该限定增强了服务提供者和服务使用者的耦合关系，提高了交换服务提供者的服务难度。
- 等幂消息：向软件代理发送多次重复消息的效果和发送单条消息相同。这一限定使服务提供者和服务使用者能够在出现故障时简单地复制消息，从而改进服务可靠性。

（9）精确定义的服务接口

服务是由提供者和使用者间的契约定义的，契约规定了服务使用方法及使用者期望的最终结果。此外，还可以在其中规定服务质量。此处需要注意的关键点是，服务契约必须进行精确定义。

3. SOA 的优点和缺点

SOA 业务流程是由一系列业务服务组成的，可以更轻松地创建、修改和管理它来

满足不同时期的需要。了解了 SOA 的定义和基本特征后，再来看看 SOA 潜在的优点。

- 编码灵活性。可基于模块化的低层服务、采用不同组合方式创建高层服务，从而实现重用，这些都体现了编码的灵活性。此外，由于服务使用者不直接访问服务提供者，这种服务实现方式本身也可以灵活使用。
- 明确开发人员角色。例如，熟悉 BES（BlackBerry Enterprise Server，黑莓服务器企业版）的开发人员可以集中精力在重用访问层，协调层开发人员则无须特别了解 BES 的实现，而将精力放在解决高价值的业务问题上。
- 支持多种客户类型。借助精确定义的服务接口和对 XML、Web 服务标准的支持，可以支持多种客户类型，包括 PDA（Personal Digital Assistant，个人数码助理）、手机等新型访问渠道。
- 更易维护。服务提供者和服务使用者的松散耦合关系及对开放标准的采用确保了该特性的实现。
- 更好的伸缩性。依靠服务设计、开发和部署所采用的架构模型实现伸缩性。服务提供者可以彼此独立调整，以满足服务需求。
- 更高的可用性。该特性在服务提供者和服务使用者的松散耦合关系上得以体现，服务使用者无须了解服务提供者的实现细节，这样，服务提供者就可以在集群环境中灵活部署，服务使用者可以被转接到可用的例程上。
- 利用现有的资产。方法是将这些现有的资产包装成提供企业功能的服务。组织可以继续从现有的资源中获取价值，而不必重新从头开始构建。
- 更易于集成和管理复杂性。将基础设施和实现发生的改变所带来的影响降到最低限度。因为复杂性是隔离的，当更多的企业一起协作提供价值链时，这会变得更加重要。
- 更快地整合现实。通过利用现有的构件和服务，可以减少完成软件开发生命周期所需的时间。这使得可以快速地开展新的业务服务，并允许组织迅速地对改变做出响应和缩短开发时间。
- 减少成本和增加重用。通过以松散耦合的方式公开业务服务，企业可以根据业务要求更轻松地使用和组合服务。

作为一个具有发展前景的应用系统架构，SOA 尚处在不断的发展中，存在很多有待改进的地方。Stencil Group 公司的 Brent Sleeper 指出，SOA 在可靠性、安全性、编制、遗留系统支持、语义和性能方面还存在严重不足。

- 可靠性。SOA 还没有完全为事务的最高可靠性——不可否认性（Nonrepudiation）、消息一定会被传送且仅传送一次（once-and-only-once delivery）及事务撤回（Rollback）——做好准备。
- 安全性。在过去，访问控制只需要登录和验证，而在 SOA 环境中，由于一个应用程序的构件很容易去跟属于不同域的其他构件进行对话，所以确保迥然不同又相互连接的系统之间的安全性就复杂得多。
- 编排（Orchestration）。统一协调分布式软件构件以便构建有意义的业务流程是最复杂的，但它同时也最适合面向服务类型的集成，原因很显然，建立在 SOA 上面的应用程序可以被设计成可以按需要拆散、重新组装的服务。作为目前业务流程管理解决方案的核心，编排功能使 IT 管理人员能够通过已经部

署的套装或自己开发的应用软件的功能，把新的元应用软件（Meta-Application Software）连接起来。事实上，最大的难题不是建立模块化的应用软件，而是改变这些系统表示所处理数据的方法。

- 遗留系统支持。SOA 中提供集成遗留系统的适配器，遗留应用适配器屏蔽了许多专用性 API 的复杂性和晦涩性。一个设计良好的适配器的作用好比是一个设计良好的 SOA 服务，它提供了一个抽象层，把应用基础设施的其余部分与各种棘手问题隔离开来。一些厂商就专门把遗留应用软件“语义集成”到基于 XML 的集成架构中。但是，集成遗留系统的工作始终是一个挑战。
- 语义。定义事务和数据的业务含义，一直是 IT 管理人员面临的最棘手问题。语义关系是设计良好 SOA 架构的核心要素。就目前而言，没有哪一项技术或软件产品能够真正解决语义问题。为针对特定行业和功能的流程定义并实施功能和数据模型是一项繁重的任务，它最终必须由业务和 IT 管理人员共同承担。不过，预制构件和经过实践证明的咨询技能可以简化许多难题。
- 性能。这种怀疑观点通常针对两个方面：SOA 的分布性质和 Web 服务协议开销。不可否认，任何分布式系统的执行速度都不如独立式系统，这完全是因为网络的制约作用造成的。当然，有些应用软件无法容忍网络引起的延迟（例如，那些对实时性要求很高的应用软件），所以在应用 SOA 架构之前，搞清楚它的适用范围就显得很重要了。

4. SOA 的生命周期

由于 SOA 涉及业务的诸多方面，因此需要从一开始就对 SOA 项目进行细心的规划和设计。需要考虑项目的整个生命周期，从最初的阶段到第一个实现，再一直到可能的修订和重用。SOA 生命周期如图 8-6 所示。

(1) 建模

SOA 项目的第一步几乎和技术没有任何关系，所有事项都与用户的业务相关。面向服务的方法将业务所执行的活动视为服务，因此第一步是要确定这些业务活动或流程实际是什么。需要对业务架构进行记录，这些记录不仅可以用于规划 SOA，还可以用于对实际业务流程进行优化。通过在编写代码前模拟或建模业务流程，可以更深入地了解这些流程，从而有利于构建帮助执行这些流程的软件。

建模业务流程的程度将依赖于预期实现的深度。另外，这个程度还依赖于开发人员在开发团队中担任的角色。如果是企业架构师，将会对实际的业务服务进行建模。如果是软件开发人员，将可能对单个服务进行建模。

(2) 组装

对业务流程进行了建模和优化后，开发人员可以开始构建新的服务和/或重用现有的服务，然后对其进行组装以形成组合应用程序，从而实现这些流程。在建模过程中，

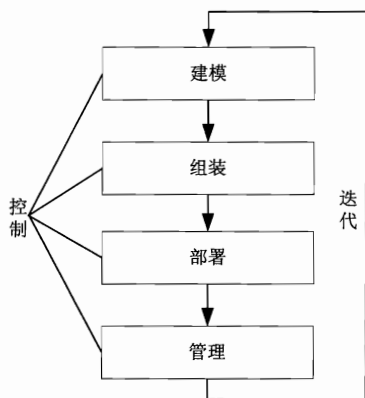


图 8-6 SOA 生命周期

已经确定了需要何种类型的服务，以及它们将访问何种类型的数据，已经存在某种形式的实现这些服务或访问该类数据所需的一些软件。

组装过程将要找到已经存在的功能，并为其添加服务支持。另外，还涉及创建提供功能和访问数据源所需的新服务，以便满足 SOA 涉及的业务流程范围内的需求。

(3) 部署

进行建模和组装后，要将组成 SOA 的资产部署到安全的集成环境中。此环境本身提供专门化的服务，用于集成业务中涉及的人员、流程和信息。这种级别的集成可帮助确保将公司的所有主要元素连接到一起协同工作。此外，部署工作还需要满足业务的性能和可用性需求，并提供足够的灵活性，以便吸纳新服务（并使旧服务退役），而不会对整个系统造成大的影响。

(4) 管理

部署之后，还需要从 IT 和业务两个角度对系统进行管理和监视。在管理过程中收集的信息用于帮助实时地了解业务流程，从而能更好地进行业务决策，并将信息反馈回生命周期，以进行持续的流程改进工作。将需要处理服务质量、安全、一般系统管理之类的问题。

在管理过程中，可以监视和优化系统，发现和纠正效率低下的情况和存在的问题。由于 SOA 是一个迭代过程，因此，在管理过程中，不仅要找出技术架构中有待改进的地方，而且还要找出业务架构中有待改进的地方。

完成管理过程后，就要开始新的建模过程了。在管理过程中收集的数据将用于重复合整个 SOA 生命周期，再次进行整个过程。

(5) 控制

SOA 可以包含来自组织的不同部门的服务，甚至还可能包含来自组织外的服务。如果没有恰当的控制，这种系统很容易失控。

控制对所有生命周期阶段起到巩固支撑作用，为整个 SOA 系统提供指导，并有助于了解系统全貌。它提供指导和控制，帮助服务提供者 and 使用者避免遇到意外情况。

5. SOA 与其他技术的关系

SOA 可以与许多其他技术结合在一起使用，然而，构件的封装和聚合在其中扮演着重要的角色。

(1) SOA 与编程语言

除了可能离不开 XML 和 WSDL 之外，SOA 并不是特定于语言的，可以用任何编程语言来实现服务，只要这种编程语言可以生成服务并且可以与 WSDL 结合在一起使用就可以了。SOAP 本身并不是绝对需要的，但它是通用的消息传递系统。因此，可以使用几乎任何一种编程语言和支持 WSDL 的平台来实现 SOA 中的成员服务。

(2) SOA 与 CORBA

基于 CORBA 的应用程序有许多构件必须连接到 SOA 中。虽然 CORBA 中的 IDL 在概念上类似于 WSDL，但它不是严格的，因而首先需要将其映射到 WSDL。另外，

需要使用更高级的 SOA 协议（例如，用于流程和策略管理的协议），而不是 CORBA 中的类似的概念。这是 CORBA 构件（表示为服务）需要与 SOA 服务交互的情况。在 CORBA 模型中，所有的独立子集仍然可以像以前一样工作。

由 OMG 提出并在许多产品中得以实现的模型驱动的架构（Model Driven Architecture, MDA）在一个更抽象的层次上与 SOA 的概念具有很强的相关性。MDA 基于这样的概念，任何软件流程都可以定义为模型甚至是元模型（即模型的模型），然后将这些模型和元模型转换成应用程序的实际构件。因此，MDA 创建了一个模型，这个模型先编译成软件应用程序，而软件应用程序接着又编译成可执行程序，这样就可以在平台上运行了。MDA 并不区分服务和对象这两个概念，但是它允许模型由其他子集模型本身组成。

（3）SOA 与自主计算

自主计算的概念应用于管理分布式服务架构的范围，具体来说，就是帮助维护策略和服务级协议，以及 SOA 系统的总体稳定性。

（4）SOA 与网络计算

一方面，网络计算可以以两个级别与 SOA 系统一起使用。网络是分布式计算的一种形式，它利用分布式特性和服务之间的交互来为 SOA 应用程序提供计算支持。在这种情况下，网络起到了框架的作用，其中实现了一些或所有单独的服务。因此，SOA 应用程序可以是网络服务的消费者。

另一方面，网络本身也可以构建在 SOA 之上。在这种情况下，每个操作系统服务都是构成整个 SOA 应用程序的成员，而 SOA 应用程序就是网络本身。因此，单独的网格构件既可以使用 Web 服务进行通信，又可以以 SOA 的方式进行交互。总而言之，网络系统可以是 SOA 本身，也可以提供服务来在其上构建应用程序级 SOA 模型。

（5）SOA 与 Web Service

在理解 SOA 和 Web Service 的关系上，经常发生混淆。根据 2003 年 4 月的 Gartner 报道，Yefim V. Natis 对这个问题是这样解释的：“Web Service 是技术规范，而 SOA 是设计原则。特别是 Web Service 中的 WSDL，是一个 SOA 配套的接口定义标准。这是 Web Service 和 SOA 的根本联系”。

从本质上来说，SOA 是一种架构模式，而 Web Service 是利用一组标准实现的服务。Web Service 是实现 SOA 的方式之一。用 Web Service 来实现 SOA 的好处是开发人员可以实现一个中立平台来获得服务，而且随着越来越多的软件商支持越来越多的 Web Service 规范，用户会取得更好的通用性。

相对来说，Web Service 是就现在而言最适合实现 SOA 的一些技术的集合，事实上，SOA 的火爆在很大程度上归功于 Web Service 标准的成熟和应用的普及，为广泛的实现 SOA 架构提供了基础。

6. Web Service

Web Service 是解决应用程序之间相互通信的一项技术。严格地说，Web Service 是描述一系列操作的接口。它使用标准的、规范的 XML 描述接口。这一描述中包括

了与服务进行交互所需要的全部细节,包括消息格式、传输协议和服务位置。而在对外的接口中隐藏了服务实现的细节,仅提供一系列可执行的操作,这些操作独立于软、硬件平台和编写服务所用的编程语言。Web Service 既可以单独使用,也可以同其他 Web Service 一起来实现复杂的业务功能。

(1) Web Service 相关技术

XML: XML 用严格的嵌套标记表示数据信息,特别适合在 Internet 环境中的多点数据交换环境下使用。在 B2B 电子商务应用环境中,XML 是一种非常优秀且极为适合的商务信息交换技术。B2B 解决方案的软件提供商制定了用于 B2B 应用之间交换商务信息的 XML 格式,包括 xCBL、cXML、BizTalk 等,这些交换格式已经运用在众多的 B2B 应用中。2000 年,UN/CEFACT 和 OASIS 开始联手制定全球电子商务信息交换格式 ebXML,确立了 XML 作为商务应用信息交换标准技术的地位。用 XML 来描述商务信息使得各种 B2B 应用在数据层上具有了开放集成的能力。

UDDI: UDDI 是一套面向 Web 服务的信息注册中心的实现标准和规范。创建 UDDI 注册中心的目的是实现 Web 服务的发布和发现。人们利用 UDDI 规范在 Web 上建立发现服务,这些发现服务为所有的请求者提供了一致的接口,使得已经发布的 Web 服务能通过编程被需要的请求者发现。UDDI 规范文本定义了 UDDI 操作入口站点(UDDI Operator Site)能够支持的 API 接口和 API 中用 XML 描述的数据结构的具体定义。

UDDI 注册中心是对所有提供公共 UDDI 注册服务站点的统称,在逻辑上它是一个统一体,但在物理上则以分布式系统体系结构实现。不同的站点之间采用对等网络结构实现,因此访问其中任意一个站点就等同于访问 UDDI 注册中心。一般情况下,访问 UDDI 操作入口站点所获得的结果是整个 UDDI 注册中心覆盖区域的所有信息,信息查询无须身份认证,但在 UDDI 操作入口站点上进行信息发布,则必须通过该 UDDI 操作入口站点自身的用户方能实施,同时,以后的更新、删除都必须通过这个操作入口站点,并使用初始发布时使用的用户名进行权限认证。

UDDI 注册中心的信息内容分为白页信息、黄页信息和绿页信息。白页中存放企业的地址、联系方式、企业身份识别等企业信息;黄页中存放基于标准分类的行业类别信息;绿页中存放 Web Service 的技术信息。多个合作站点之间可以无缝地共享注册信息。业务实体信息、服务信息、绑定信息和服务调用规范(businessEntity, businessService, bindingTemplate, tModel)的说明信息是 4 种主要的信息类型,它们是技术人员在需要使用合作伙伴所提供的 WebService 时必须了解的技术信息,这些元素构成了 UDDI 的信息结构。

UDDI 的接口包含查询 API 和发布 API。查询 API 用来快速地定位候选的业务实体、Web 服务及其调用规范和相关信息的细节。以 find_xx 命名的 API 提供了多种搜索标准,能对注册中心的数据进行广泛的搜索。如果事先已经知道所需服务的关键字,则可以直接调用以 get_xx 命名的 API 来得到相应的信息。发布 API 分为保存 API 和删除 API。一旦得到授权,一个独立的机构可以注册任意数量的 businessEntity 或 tModel 信息,也可以修改原先发布的信息。使用以 save_xx 命名的保存 API 来保存新信息,使用以 delete_xx 命名的删除 API 来删除已经在 UDDI 中心注册的信息。

SOAP: SOAP 完全继承了 XML 的开放性和描述可扩展性。SOAP 使用基于 TCP/IP

的应用层协议 HTTP、SMTP、FTP 等，可以与现有通信技术最大程度地兼容。SOAP 为使用 XML 在松散、分布的环境中对等地交换结构化和类型化信息提供了一个简单的机制。SOAP 本身并不定义任何应用语义，如编程模型或特定语义实现，它只定义一种简单的机制，通过一个模块化的包装模型和对模块中特定格式编码的数据重编码机制来表示应用语义。SOAP 的这项能力使其可被很多类型的系统用于从消息系统到 RPC 的延伸。

SOAP 规范主要由 SOAP 信封、编码规则、RPC 表示 3 部分组成。信封定义了整体 SOAP 消息的表示框架，可用于表示消息中的内容是什么，谁发送的，谁应当接收并处理它，以及这些处理操作是否是必需的。SOAP 编码规则定义了数据的编码机制，通过它来定义应用程序中需要使用的数据类型，并可用来交换由这些应用程序定义的数据类型所衍生的实例。SOAP RPC 定义了一个用于表示远程调用和响应的约定，例如，如何使用 HTTP 或 SMTP 协议与 SOAP 绑定，如何传输过程调用，在具体传输协议的哪个部分传输过程响应等。这 3 部分在功能上是正交且彼此独立的。

WSDL: WSDL 是描述 Web Service 的 XML 格式语言。WSDL 把 Web Service 定义为网络端点的集合。它有一个根元素，用类型 (Types)、消息 (Message)、端口类型 (Porttype)、绑定 (Binding)、端口 (Port) 和服务 (Service) 等元素来定义 Web Service。其中，“类型”是消息的数据类型定义，通常用来描述交换消息；“消息”代表待传输数据的抽象定义，由一个或多个部分组成；“端口类型”表示抽象操作的集合；“绑定”使操作和消息的具体协议与数据格式规范关联；“端口”指定一个用于绑定的地址，由此定义一个通信端点；“服务”则是相关端口的集合。

在 WSDL 中，端点和消息的抽象定义与具体的网络布置和数据格式绑定是相互分离的，这样就可以抽象定义消息和端口类型，实现它们的重用。

(2) Web Service 的特点

Web Service 具有以下一些主要特性。

- 互访性: Web Service 通过 SOAP 实现相互的访问，任何 Web Service 都可以与其他 Web Service 进行交互，避免了不同协议之间的相互转换。Web Service 可以用任何语言编写，因此开发者不需要更改开发环境就能开发新的 Web Service，同时还可以在新的 Web Service 中使用已有的 Web Service，而不必考虑 Web Service 的实现语言、运行环境等具体实现细节。例如，用 Delphi 编写的 Web Service 可以使用由 Visual C++ 编写的 Web Service，反之亦然。
- 普遍性: Web Service 使用 HTTP 和 XML 进行通信，任何支持这些技术的设备都可以拥有和访问 Web Service。Web Service 不仅在计算机网络上出现，而且将在电话、汽车、家用电器等设备中出现。现在，各主要设备和软件供应商都已宣布支持 SOAP 和周边 Web Service 技术，相信在未来，Web Service 将普遍存在于社会生活的各个领域。使用 Web Service，我们就能够通过网络在异地来指挥家中的电器设备工作，进行诸如煮饭、加温、降温等操作。
- 廉价性: Web Service 供应商提供的免费工具箱能够让开发者快速创建和部署自己的 Web Service，其中某些工具箱还可以让已有的构件方便地成为 Web Service，这样就降低了 Web Service 的开发费用，同时也加快了开发速度。

例如,微软的.NET平台就以支持Web Service为目标,因此用Visual Basic编写的组件可以很容易地作为Web Service发布和使用。

目前,Web服务存在的问题是有太多的技术标准,各软件制造商各自为政,人们也被众多的标准给搞糊涂了。这样就导致了用户宁肯等出现新的标准和兼容性提高后才大规模采用Web Service。

现在几乎每家软件制造商都在关注Web Service概念,并且形成了互不兼容的两大阵营:一个就是微软的.NET战略;另一个则是由IBM、BEA和其他一些公司支持的Sun公司的Java技术。大部分企业用户没有明显的偏向性,往往是同时选择两种技术,这样就要求两种Web服务的兼容性最终必须被解决。

Web Service的发展趋势,势必会渐渐走向服务自动整合的境界,也就是说Web Service可能会视情况,以动态的方式寻找更适当的服务进行整合。同时,Web Service的标准将趋向统一。

8.1.7 特定领域软件架构

早在20世纪70年代就有人提出程序族、应用族的概念,并开拓了对特定领域软件架构早期研究,这与软件架构研究的主要目的“在一组相关的应用中共享软件架构”也是一致的。为了解脱因为缺乏可用的软件构件,以及现有软件构件难以集成而导致软件开发过程中难以进行重用的困境,Mettala在1990年提出了特定领域软件架构(Domain Specific Software Architecture, DSSA),尝试解决这类问题。

简单地说,DSSA就是一个特定的问题领域中支持一组应用的领域模型、参考需求、参考架构等组成的开发基础,其目标就是支持在一个特定领域中多个应用的生成。

DSSA的必备特征如下:

- 一个严格定义的问题域和/或解决域。
- 具有普遍性,使其可以用于领域中某个特定应用的开发。
- 对整个领域的合适程度的抽象。
- 具备该领域固定的、典型的在开发过程中可重用元素。

从功能覆盖的范围角度有两种理解DSSA中领域的含义的方式:

- 垂直域:定义了一个特定的系统族,包含整个系统族内的多个系统,结果是在该领域中可作为系统的可行解决方案的一个通用软件架构。
- 水平域:定义了多个系统和多个系统族中功能区域的共有部分,在子系统级上涵盖多个系统族的特定部分功能,无法为系统提供完整的通用架构。

在垂直域上定义的DSSA只能应用于一个成熟的、稳定的领域,但这个条件比较难以满足;若将领域分割成较小的范围,则相对更容易,也容易得到一个一致的解决方案。

1. DSSA的活动

DSSA的主要活动可以分为3个阶段,分别是领域分析、领域设计和领域实现。

(1) 领域分析

这个阶段的主要目标是获得领域模型 (Domain Model)。领域模型描述领域中系统之间共同的需求。领域模型所描述的需求称为领域需求。在这个阶段中, 首先要进行一些准备性的活动, 包括定义领域的边界, 从而明确分析的对象; 识别信息源, 即领域分析和整个领域工程过程中信息的来源, 可能的信息源包括现存系统、技术文献、问题域和系统开发的专家、用户调查和市场分析、领域演化的历史记录等。在此基础上, 就可以分析领域中系统的需求, 确定哪些需求是被领域中的系统广泛共享的, 从而建立领域模型。当领域中存在大量系统时, 需要选择它们的一个子集作为样本系统。对样本系统需求的考察将显示领域需求的一个变化范围。一些需求对所有被考察的系统是共同的, 一些需求是单个系统所独有的。很多需求位于这两个极端之间, 即被部分系统共享。

(2) 领域设计

这个阶段的目标是获得 DSSA。DSSA 描述在领域模型中表示的需求的解决方案, 它不是单个系统的表示, 而是能够适应领域中多个系统的需求的一个高层次的设计。建立了领域模型之后, 就可以派生出满足这些被建模的领域需求的 DSSA。由于领域模型中的领域需求具有一定的变化性, DSSA 也要相应地具有变化性。它可以通过表示多选一的 (Alternative)、可选的 (Optional) 解决方案等来做到这一点。由于重用基础设施是依据领域模型和 DSSA 来组织的, 因此在这个阶段通过获得 DSSA, 也就同时形成了重用基础设施的规约。

(3) 领域实现

这个阶段的主要目标是依据领域模型和 DSSA 开发和组织可重用信息。这些可重用信息可能是从现有系统中提取得到的, 也可能需要通过新的开发得到。它们依据领域模型和 DSSA 进行组织, 也就是领域模型和 DSSA 定义了这些可重用信息的重用时机, 从而支持了系统化的软件重用。这个阶段也可以看做重用基础设施的实现阶段。

以上过程是一个反复的、逐渐求精的过程。在实施领域工程的每个阶段中, 都可能返回到以前的步骤, 对以前的步骤得到的结果进行修改和完善, 再回到当前步骤, 在新的基础上进行本阶段的活动。

2. DSSA 的建立过程

因所在的领域不同, DSSA 的创建和使用过程也各有差异, Tracz 曾提出一个通用的 DSSA 应用过程, 这些过程也需要根据所应用到的领域来进行调整。一般情况下, 需要用所应用领域的应用开发者习惯使用的工具和方法来建立 DSSA 模型。同时, Tracz 强调了 DSSA 参考架构文档工作的重要性, 因为新应用的开发和对现有应用的维护都要以此为基础。

DSSA 的建立过程分为 5 个阶段, 每个阶段可以进一步划分为一些步骤或子阶段。每个阶段包括一组需要回答的问题、一组需要的输入、一组将产生的输出和验证标准。该过程是并发的 (Concurrent)、递归的 (Recursive)、反复的 (Iterative), 或者说, 它是螺旋形 (Spiral) 的。完成该过程可能需要对每个阶段经历几遍, 每次增加更多的细节。

(1) 定义领域范围

本阶段的重点是确定什么在感兴趣的领域中及本过程到何时结束。这个阶段的一个主要输出是领域中的应用需要满足一系列用户的需求。

(2) 定义领域特定的元素

本阶段的目标是编译领域字典和领域术语的同义词词典。在领域工程过程的前一个阶段产生的高层块图将被增加更多的细节，特别是识别领域中应用间的共同性和差异性。

(3) 定义领域特定的设计和实现需求约束

本阶段的目标是描述解空间中有差别的特性。不仅要识别出约束，并且要记录约束对设计和实现决定造成的后果，还要记录对处理这些问题时产生的所有问题的讨论。

(4) 定义领域模型和架构

本阶段的目标是产生一般的架构，并说明构成它们的模块或构件的语法和语义。

(5) 产生、搜集可重用的产品单元

本阶段的目标是为 DSSA 增加构件使得它可以被用来产生问题域中的新应用。

DSSA 的建立过程是并发的、递归的和反复进行的。该过程的目的是将用户的需要映射为基于实现限制集合的软件需求，这些需求定义了 DSSA。在此之前的领域工程和领域分析过程并没有对系统的功能性需求和实现限制进行区分，而是统称为“需求”。如图 8-7 所示是 DSSA 的一个三层次系统模型。

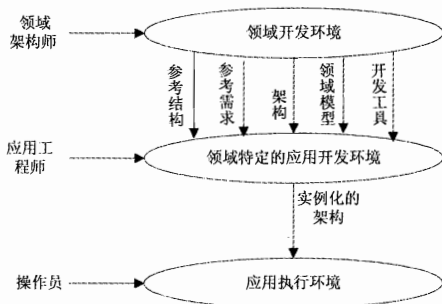


图 8-7 DSSA 的三层次系统模型

DSSA 的建立需要设计人员对所在特定应用领域（包括问题域和解决域）必须精通，他们要找到合适的抽象方式来实现 DSSA 的通用性和可重用性。通常，DSSA 以一种逐渐演化的方式发展。

8.1.8 软件产品线

软件产品线（Software Product Line）是一个产品集合，这些产品共享一个公共的、可管理的特征集，这个特征集能满足特定领域的特定需求。软件产品线是一个十分适合专业的开发组织的软件开发方法，能有效地提高软件生产率和质量，缩短开发时间，降低总开发成本。

软件产品线主要由两部分组成，分别是核心资源和产品集合。核心资源是领域工程的所有结果的集合，是产品线中产品构造的基础。核心资源必定包含产品线中所有产品共享的产品线架构，新设计开发的或者通过对现有系统的再工程得到的、需要在整个产品线中系统化复用的构件。与构件相关的测试计划、测试实例及所有设计文档，需求说明书、领域模型、领域范围的定义，以及采用 COTS 的构件也属于核心资源。

1. 产品线的过程模型

软件产品线的过程模型主要有双生命周期模型、SEI 模型和三生命周期模型。

(1) 双生命周期模型

双生命周期模型分成两个重叠的生命周期，分别是领域工程和应用工程，如图 8-8 所示。

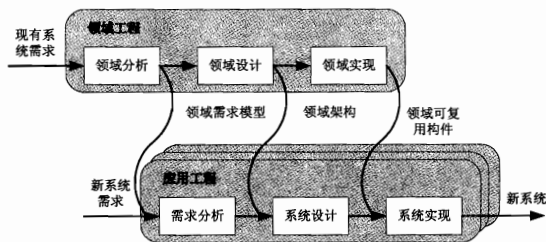


图 8-8 产品线的双生命周期模型

领域工程阶段的主要任务如下：

- 领域分析：利用现有系统的设一步计、架构和需求建立领域模型。
- 领域设计：用领域模型确定领域/产品线的共性和可变性，为产品线设计架构。
- 领域实现：基于领域架构开发领域可复用资源，例如，构件、文档和代码生成器等。

应用工程在领域工程结果的基础上构造新产品。应用工程需要根据每个应用独特的需求，经过以下阶段，生成新产品。

- 需求分析：将系统需求与领域需求比较，划分成领域公共需求和独特需求两部分，得出系统需求规格说明书。
- 系统设计：在领域架构基础上，结合系统独特需求，设计应用的软件架构。
- 系统实现：遵照应用架构，用领域可复用资源实现领域公共需求，用定制开发的构件满足系统独特需求，构建新的系统。

应用工程将产品线资源不能满足的需求返回给领域工程，以检验是否将之并入产品线的需求中。领域工程从应用工程中获得反馈或结合新产品的需求，进入又一次周期性发展，称为产品线的演化。

双生命周期模型定义了典型的产品线开发过程的基本活动、各活动内容和结果，以及产品线的演化方式。这种产品线方法综合了软件架构和软件重用的概念，在模型中定义了一个软件工程化的开发过程，目的是提高软件生产率、可靠性和质量，降低开发成本，缩短开发时间。

(2) SEI 模型

SEI 将产品线的基本活动分为 3 个部分，分别是核心资源开发（即领域工程）、产品开发（即应用工程）和管理，如图 8-9 所示。

从本质上来看，产品线开发包括核心资源库的开发和使用核心资源的产品开发，这两者都需要技术和组织的管理。核心资源的开发和产品开发可同时进行，也可交叉进行。例如，新产品的构建以核心资源库为基础，或者核心资源库可从已存在的系统中抽取。

在图 8-9 中，每个旋转环代表一个基本活动，3 个环连接在一起，不停地运动着。3 个基本活动交错连接，可以任何次序发生，且是高度重叠的。旋转的箭头表示不但核心资源库被用来开发产品，而且已存在的核心资源的修改，甚至新的核心资源常常可以来自产品开发。在核心资源和产品开发之间有一个强的反馈环，当新产品开发时，核心资源库就得到刷新。对核心资源的使用反过来又会促进核心资源的开发活动。另外，核心资源的价值通过使用它们的产品开发来得到体现。

SEI 模型的主要特点如下：

- 循环重复是产品线开发过程的特征，也是核心资源开发、产品线开发，以及核心资源和产品之间协作的特征。
- 核心资源开发和产品开发没有先后之分。
- 管理活动协调整个产品线开发过程的各个活动，对产品线成败负责。
- 核心资源开发和产品开发是两个互动的过程，3 个活动和整个产品线开发之间也是双向互动的。

(3) 三生命周期模型

三生命周期模型是对双生命周期模型的一种改进，主要针对大型软件企业的软件产品线开发，如图 8-10 所示。

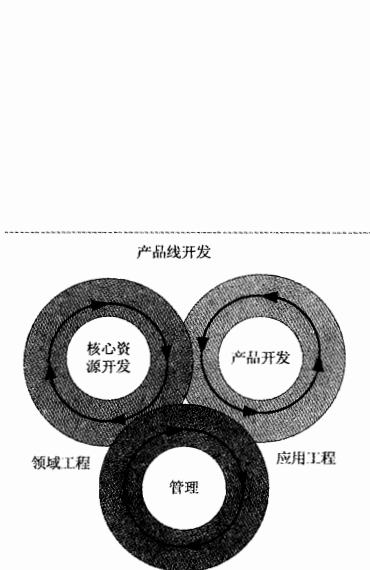


图 8-9 SEI 模型式上

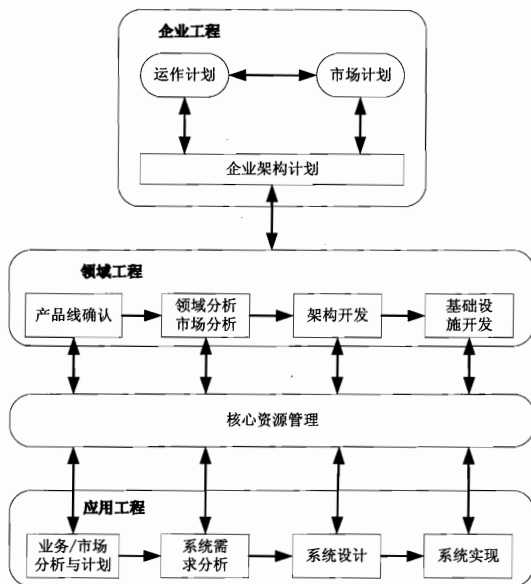


图 8-10 产品线的三生命周期模型

三生命周期模型为有多个产品线的大型企业增加了企业工程（Enterprise Engineering）流程，以便在企业范围内对所有资源的创建、设计和复用提供合理的规划。为了强调产品线工程在满足市场需求上与一般的系统复用的区别，在领域工程中增加了产品线确定作为起始阶段，和领域分析阶段、架构开发阶段、基础设施开发阶段组成整个领域工程，还为领域分析阶段增加了市场分析的任务。同样，为应用工程增加了业务/市场分析与规划。在领域工程和应用工程之间的双向交互中添加核心资源管理作为桥梁，核心资源管理和领域工程、应用工程之间的支持和交互是双向的，以便于产品线核心资源的管理和演化。

最后，需要说明的是，在上述3个过程模型中，软件产品线开发过程并没有明确描述如何复用遗留资源。实际上，大多数将要建立软件产品线的企业都积累了有产品线所在领域的大量应用代码和相关文档，这些代码和文档中包含的知识对领域工程来说是至关重要的。

2. 产品线的建立方式

软件产品线的建立需要企业有意识的、长期的努力才有可能成功。采用产品线方式开发软件时，开发人员可以划分为两组，分别是负责核心资源的小组和负责产品的小组。这也是产品线开发与独立系统开发的主要区别。

软件产品线的建立通常有4种方式，其划分依据有两个，第一个是企业采用演化方式（Evolutionary）还是革命方式（Revolutionary）引入产品线开发过程，第二个是企业基于现有产品开发还是开发全新的产品线。这4种方式基本特征如表8-3所示。

表 8-3 软件产品线建立方式基本特征

	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

（1）将现有产品演化为产品线

在基于现有产品架构设计的产品线架构的基础上，将特定产品的构件逐步地、越来越多地转化为产品线的共用构件，从基于产品的方法“慢慢地”转化为基于产品线的软件开发。这种方法的主要优点是通过对投资回收期的分解、对现有系统演化的维持，使产品线方法的实施风险降到了最小，但完成产品线核心资源的总周期和总投资都比使用革命方式要大。

（2）用软件产品线替代现有产品集

基本停止现有产品的开发，所有工作直接针对软件产品线的核心资源开发。遗留系统只有在符合架构和构件需求的情况下，才可以和新的构件协作。这种方法的目标是开发一个不受现有产品集限制的、全新的平台，总周期和总投资较演化方法要少，但因重要需求的变化导致的初始投资报废的风险加大。另外，基于核心资源的第一个产品面世的时间将会推后。

除了投资和周期方面的考虑外，现有产品集中软硬件结合的紧密程度，以及不同产品在硬件方面的需求的差异，也是产品线开发采用演化还是革命方式的决策依据。

对于软硬件结合密切，且硬件需求差异大的现有产品集，因无法满足产品线方法对软硬件同步的需求，只能采用革命方式替代现有产品集。

（3）全新软件产品线的演化

当企业进入一个全新的领域，要开发该领域的一系列产品时，同样也有演化和革命两种方式。演化方式将每一个新产品的需求与产品线核心资源进行协调。这种方式的好处是先期投资少，风险较小，第一个产品面世时间早。另外，因为是进入一个全新的领域，演化方法可以减少和简化因经验不足造成的初始阶段错误的修正代价；缺点是已有的产品线核心资源会影响新产品的需求协调，使成本加大。

（4）全新软件产品线的开发

设计师和开发工程师首先要得到产品线所有可能的需求，基于这个需求超集来设计和开发产品线核心资源。第一个产品将在产品线核心资源全部完成之后才开始构建。这种方式的优点是，一旦产品线核心资源完成后，新产品的开发速度将非常快，总成本也将减少；缺点是对新领域的需求很难做到全面和正确，使得核心资源不能像预期的那样支持新产品的开发。

从整体上来看，软件产品线的发展过程有3个阶段，分别是开发阶段、配置分发阶段和演化阶段。引起产品线架构演化的原因与引起任何其他系统演化的原因一样：产品线与技术变化的协调、现有问题的改正、新功能的增加、对现有功能的重组以允许更多的变化等。产品线的演化包括产品线核心资源的演化、产品的演化和产品的版本升级。

不管采用哪种产品线的建立方式，企业要成功实施产品线，主要取决于以下因素：

- 对要实施产品线的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（包括软件资源、人员组织和过程等）支持。

8.1.9 基于架构的软件开发方法（ABSD）

为软件系统设计一个体系结构不是一件容易的事，软件必须长期运行和具有自适应性，它必须支持广义的软件需求，详细的需求要等到最终产品开发完成后才能知道。而且，在作出最基本的设计决定时，就要进行初步的体系结构设计，这些决策一旦出现错误，就难以改正。为了有效地设计一个软件体系结构，软件设计师需要一个严格的设计方法，这些方法关注创造性的过程，为处理非确定软件需求提供策略，在设计过程中，为组织作出设计决策提供指导。

在本节中，我们将介绍基于体系结构的软件设计（Architecture-Based Software Design, ABSD）方法。ABSD方法为产生软件系统的概念体系结构提供基础，概念体系结构是由Hofmeister、Nord和Soni提出的4种不同的体系结构中的一种，它描述了系统的主要设计元素及其关系。概念体系结构代表了在开发过程中作出的第一个选择，相应地，它是达到系统质量和业务目标的关键，为达到预定功能提供了一个基础。

ABSD方法取决于决定系统的体系结构驱动。所谓体系结构驱动，是指构成体系

结构的业务、质量和功能需求的组合。使用 ABSD 方法，设计活动可以在体系结构驱动一决定就开始，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。设计活动的开始并不意味着需求获取和分析活动就可以终止，而是应该与设计活动并行。特别是在不可能预先决定所有需求时，例如产品线系统或长期运行的系统，快速开始设计是至关重要的。

ABSD 方法有 3 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体系结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。然而，对于设计方法来说，软件模板的使用是一个新概念，下面，我们进行简单的介绍。

软件模板是一个特殊类型的软件元素，包括描述所有这种类型的元素在共享服务和底层构造的基础上如何进行交互。软件模板还包括属于这种类型的所有元素的功能，这些功能的例子有：每个元素必须记录某些重大事件，每个元素必须为运行期间的外部诊断提供测试点等。在软件产品线系统中，软件模板显得格外重要，因为新元素的引入是一个通用的技术，这种技术用来使产品线体系结构适应一个特定的产品。

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，体系结构总是清晰的，这有助于降低体系结构设计的随意性。

1. ABSD 方法与生命周期

图 8-11 描述了 ABSD 方法在生命周期中的位置。尽管我们没有描述一个需求获取、组织或跟踪的特定方法，但还是假设一个需求阶段至少部分地完成，从需求阶段（包括功能需求、质量和业务需求、约束等）获得了输出。ABSD 方法的输出是 3 个视图的概念构件的集合，包括能够产生每个概念构件的假定、软件模板的集合和那些已经作出的具体实现的决策，我们把具体实现决策当作附加约束来维护。

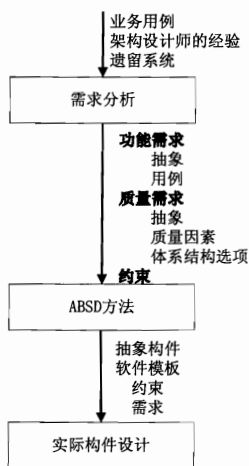


图 8-11 ABSD 方法与生命周期

在 ABSD 方法中，必须记录所有作出的决策及这些决策的原理，这有利于决策的可跟踪性和决策评审。

ABSD 方法的输入由下列部分组成:

- 抽象功能需求, 包括变化的需求和通用的需求。
- 用例 (实际功能需求)。
- 抽象的质量和业务需求。
- 质量因素 (实际质量和业务需求)。
- 体系结构选项;
- 约束。

下面, 我们描述需求阶段的假定输出, 即 ABSD 方法的输入。

(1) 抽象功能需求

ABSD 方法假定需求阶段的输出之一是功能需求的抽象描述, 包括这些需求的粗略变化的描述。当获取需求时, 考虑所有最终用户是重要的。

对一个特定系统来说, 通常有不同类型的最终用户。不同的系统管理员 (数据库管理员、系统管理员、网络管理员等) 都可以是最终用户。维护工程师也可以是系统的最终用户。总之, 一个最终用户就是当系统运行时使用系统的任何人员。

与抽象功能需求相联系的是对公共需求和与这些需求相关的粗略变化的描述, 在设计阶段, 理解这些需求之间的依赖关系是至关重要的。

我们必须在某种抽象级别上获取功能需求, 产品的详细需求往往要等具体产品开发完成后才能知道。当详细需求明确时, 抽象功能的获取为详细需求提供了分类。

(2) 用例

如前所述, 用例是一个或多个最终用户与系统之间的交互的具体表述, 在这里, 最终用户既可以是操作人员, 也可以是与系统进行交互操作的其他软件系统。虽然用例很容易找到和创建, 甚至可能有成百上千个, 但是, 因为我们需要分析用例, 所以必须限制用例的数量。在体系结构设计阶段, 只有重要的用例才有用。我们必须对所创建的用例进行分组、设置优先级, 以便筛选出最重要的用例, 剩下的用例可以在设计阶段的任何时候创建。

(3) 抽象的质量和业务需求

我们必须对待构建系统的质量和业务需求进行编号, 每个质量属性都包含一个特定的刺激, 以及希望得到的响应 (Response)。质量需求要尽量具体化。

(4) 体系结构选项

对每个质量和业务需求, 我们都要列举能够满足该需求的所有可能的体系结构。例如, 如果需求是支持一系列不同的用户界面, 则可能的体系结构选择就是把不同的用户界面分解成不同的构件。又如, 如果需求是保持操作系统的独立性, 则可能的体系结构选择就是构建虚拟的操作系统层, 接受所有的操作系统调用 (Invocation), 并解释之为当前操作系统所能支持。

在这个时候, 只需列举所有可能的选项, 而不需要对这些体系结构选项进行决策, 这种列举取决于设计师的经验, 既可来自某些书籍介绍, 也可直接来自设计师本身的实践。

(5) 质量场景

正如用例使功能需求具体化一样,质量场景使质量需求具体化。质量场景是质量需求的特定扩充。

与用例一样,质量场景也很容易找到和创建,可以创建很多个。我们必须对质量场景进行分组、设置优先级,只需验证最重要的质量场景。

(6) 约束

约束是一个前置的设计决策,设计过程本身包含决策。某些决策可以直接由业务目标导出而无须考虑对设计的影响。例如,如果一个公司在某个中间件产品上投入了大量资金,那么在产品的选择上就可以不必考虑其他决策。在需求获取阶段,约束主要来自系统的业务目标。

在某些特殊情况下,约束由遗留系统决定。今天,几乎没有软件系统不参考已有系统的,常见的情况是,新老系统同时并存,或者新系统替代老系统,但是必须尽可能重用老系统的功能。在设计阶段,虽然这些遗留系统处于被设计系统的外部,但设计师必须考虑遗留系统的特征。也就是说,在某种程度上,遗留系统影响着当前的设计,因此,理解遗留系统的结构和解决问题的技术都很重要。出于商业目的,可能要求重用遗留系统的构件,这种需求就变成了约束。

2. ABSD 方法的步骤

ABSD 方法大致分为 9 个步骤,在具体实施过程中可以根据实际情况进行步骤的裁剪。下面将详细叙述这些步骤的具体工作内容。

(1) 功能分解

一个设计元素有一组功能,这些功能必须分组。分解的目的是使每个组在体系结构内代表独立的元素。分解可以进一步细化。这种分解的标准取决于对一个特定的设计元素来说是很重要的性能。在不同的性能基础上,可以进行多重分解。

如果像通常的产品一样,在分解中起关键作用的性能要求是可修改的,则功能的分组可选择几个标准:

- 功能聚合。需求分组必须遵守“高内聚低耦合”的原则,这是对功能分解的一个标准技术。用例能够用来检查内聚和耦合,用来处理变化的性能因素也可用来检查内聚和耦合。
- 数据或计算行为的类似模式。在数据或计算行为上,如果有类似模式的功能,则应该分在同一组。这意味着展示类似行为取决于使用系统的特定领域。如果数据获取是一个功能,则一个类似模式可能是样本周期。如果一个特定功能是需要很大计算量的,则它应该与其他计算量的功能分在一组。用同一模式存取数据库的功能也应该分在一组。
- 类似的抽象级别。与硬件相近的功能不应该与那些抽象级别较高的功能分在一组。另一方面,处在同一抽象级别的功能应该分在一组。
- 功能的局部性。那些为其他服务提供服务的功能不应该与纯局部功能分在一组。

(2) 选择体系结构风格

每个设计元素有一个主要的体系结构风格或模式,这是设计元素如何完成它的功

能的基础。主要风格并不是唯一风格，为了达到特定目的，可以进行修改。体系结构风格的选择建立在设计元素的体系结构驱动基础上。因此，这个过程就是为设计元素决定体系结构驱动和从体系结构驱动及功能分解角度考虑问题，决定主要的体系结构风格。

在架构设计过程中，并不总是有现成的体系结构风格可供选择。在这种情况下，架构设计师应该设计一种新的风格来适合现有实际情况。有时，一个设计元素的功能包含一组相对独立的子功能，一个主要的风格不足以辩识。在这种情况下，选择一个具有与设计元素外界连接的独立的过滤器风格，是一种好的选择。

一旦选定了一个主要的体系结构风格，该风格必须适应基于属于这个设计元素的质量需求，体系结构选择必须满足质量需求。也就是说，要检查每一个质量需求，判断它是否与所分解的设计元素有关，如果有关，则选择一个选项与质量需求关联和应用到风格的选择之中。

当一个特定的质量需求与多个设计元素有关时，我们希望使用同样的选择，以使该选择在设计记录中与该设计元素相关联。

选择和细化一个体系结构风格的结果是一组体系结构构件类型，例如，可以是一个“客户”类型，该“客户”既与客户机所要计算的功能没有关系，也与存在多少个实例没有关系。这些问题都将在下一步决定。一些构件类型，特别是由质量需求导出的构件，可能与功能有关联，如“虚拟设备”。

为设计元素选择体系结构风格是一个重要的选择，这种选择在很大程度上依赖于架构设计师的个人设计经验。可供选择的风格有运行时风格（如客户/服务器结构）、开发时风格（如层次式结构），或者二者兼有（如三层结构）。在任何情况下，风格的选择可能产生附加功能，这些附加功能必须要加到功能组中。

（3）为风格分配功能

选择体系结构风格时产生了一组构件类型，我们必须决定这些类型的数量和每个类型的功能，这就是分配的目的。在功能分解时产生的功能组，应该分配给选择体系结构风格时产生的构件类型，这包括决定将存在多少个每个构件类型的实例，每个实例将完成什么功能。这样分配后产生的构件将作为设计元素分解的子设计元素。

每个设计元素的概念接口也必须得到标识，这个接口包含了设计元素所需的信息和在已经定义了体系结构风格内的每个构件类型所需要的数据和控制流。

重复以上3个步骤，需在各种质量属性之间进行折中处理，架构设计师必须判断在哪里折中会合适些。

（4）细化模板

被分解的设计元素有一组属于它的模板。在ABSD方法的初期，系统没有模板。当模板细化了以后，就要把功能增加上去。这些功能必须由实际构件在设计过程中加以实现。

对于存在的模板的每个功能来说，要考虑如下问题：这些功能的方方面面是否都由子设计元素处理了，或者这些功能仍然留在当前区域？该问题有两种可能答案：

- 这些功能仍然留在当前区域。此时，无须做任何事情。
- 功能将在各方面进行分解，满足当前区域。

子设计元素的功能也要进行检查，以决定是否应该加到模板中。这些是：

- 这些功能是可以被某些子设计元素所共享，而不是某个子设计元素专有。
- 每个子设计元素必须管理的职责，如错误处理、活动日志或为外部诊断提供检查点。

最后，需要检查模板的功能，以判断是否需要增加附加功能到系统任何地方的设计元素中。也就是说，要识别在该级别上已经存在的任何横向服务。模板包括了什么是一个好的设计元素和那些应该共享的功能。每种类型的功能可以需要附加支持功能，这种附加功能一旦得到识别，就要进行分配。

(5) 功能校验

用例用来验证设计元素，验证它们能否通过一定的结构达到目的。子设计元素的附加功能将可能通过用例的使用得到判断。然而，在功能分解的过程中，如果过度地使用用例，将几乎不能发现附加功能。

也可以使用变化因素，因为执行一个变化的难点取决于功能的分解。

从这种类型的校验出发，设计就是显示需求（通过用例）和支持修改（通过变化因素）。

(6) 检查并发视图

检查并发视图的目的是判断哪些活动是可以并发执行的。必须识别这些活动，以产生进程同步和资源竞争。

对并发视图的检查是通过虚拟进程来实现的。虚拟进程是通过程序、动态模块或一些其他的控制流执行的一条单独路径。虚拟进程与操作系统的进程概念不一样，操作系统的进程包括额外的地址空间的分配和调度策略。一个操作系统进程是几个虚拟进程的连接点，但每个虚拟进程不一定是操作系统进程。虚拟进程用来描述活动序列，使同步或资源竞争可以在多个虚拟进程之间进行。

用例用来检查两个用户的影响，并行性影响一个用户的活动。

发现同步和资源竞争可能增加新的功能。例如，资源管理可以增加，用来管理竞争。在这种情况下，新功能必须分配为设计元素的功能。

(7) 创建配置视图

如果在一个系统中，使用了多个处理器，则需要对不同的处理器配置设计元素，这种配置通过配置视图来进行检查。例如，我们检查网络对虚拟线程的影响，一个虚拟线程可以通过网络从一个处理器传递到另一个处理器。我们使用物理线程来描述在某个特定处理器中的线程。也就是说，一个虚拟线程是由若干个物理线程串联而成的。通过这种视图，我们可以发现一个单一的处理器的同步的物理线程和把一个虚拟线程从一个处理器传递到其他处理器上的需求。

在配置视图中，我们使用配置单元的概念，配置单元是能分配给处理器的最小设

计元素，其准确大小取决于设计元素的粒度。设计师必须作出哪个级别的设计元素组成一个配置单元的决策。如果一个设计元素的粒度大于配置单元，则需要对其进行分解，分配给数个处理器。如果一个设计元素的粒度小于或等于配置单元，则创建配置视图就意味着必须作出配置决策。

(8) 验证质量场景

一旦创建了3个视图，就要把质量场景应用到所创建的子设计元素上。对每个质量场景，都要考虑是否仍然满足需求，每个质量场景包括了一个质量属性刺激和所期望的响应。考虑到目前为止所作出的设计决策，看其是否能够达到质量属性的要求。

如果不能达到，则需重新考虑设计决策，或者设计师必须接受创建质量场景失败的现实。

(9) 验证约束

最后一步就是要验证所有的约束有没有互相矛盾的地方，对每一个约束，都需提问“该约束是否有可能实现？”。一个否定的回答就意味着对应的质量场景也不能满足。这时，需要把问题记录进文档，对导致约束的决策进行重新验证。

8.1.10 软件架构与质量属性

质量属性与软件架构是两个密不可分的概念。如何满足用户对各种质量属性的要求，是软件架构设计的基础。不同的质量属性可以用不同的策略来实现，本节后面的部分将详细叙述相关内容。但值得注意的一点是，并不是所有的质量属性目标都可以通过架构来实现的。例如：可修改性由划分功能的方式和模块中的编码技巧决定。其中“划分功能的方式”是由软件架构设计来决定的，但“编码技巧”是在实现阶段，由程序员的技术而决定的。又如：易用性包括界面的清晰易用，以及系统是否能为用户提供方便使用的功能，其中“系统是否能为用户提供方便使用的功能”取决于软件架构设计，但“界面的清晰易用”属于界面设计的范畴。

1. 可用性

可用性是指系统能够正常运行的时间比例。在软件架构设计过程中，可以采用一系列的控制可用性策略，让屏蔽或是修复错误，从而使系统的可用性达到更高的水平。常见的策略包括：

(1) 错误检测技术

- 命令/响应：由某个组件向被审查组件发出一个命令，如果在规定的时间内收到来自被审查组件的响应，则表示组件正常工作。
- 心跳：这种情况下，一个组件定期发出一个心跳消息，另一个组件收听该消息，如果超过时间未收到心跳消息，则认为组件出故障了。
- 异常：如果系统运行中抛出了异常，则说明出现了故障。

(2) 错误恢复

- 表决：使用多个功能相同的模块，同时来进行同样数据的处理，最后将结果进行对比，以少数服从多数的方式，从中选取处理结果作为最终结果，以此

修正错误。如进行加法运算处理，有3个加法模块：A、B、C，对于输入值 $3+7$ ，A的结果为10，B的结果为11，C的结果为10，则系统认为最终结果是10，这样，就将错误的运算结果进行了修正。

- 冗余：同样功能的设备有多个，当出现问题后，由备用设备替代工作。

(3) 错误预防

- 定期重置：该策略定期（或是达到某种参数值），则将组件重新启动。例如：很多网游的服务器会采用定期重启的策略，因为长时间运行，有可能导致内存泄露，从而导致错误的产生。如果提前重新启动，就能避免错误的产生。
- 进程监视器：一旦检测到进程中存在的错误，监视进程就可以删除非执行进程，并为该进程创建一个新的实例。

2. 可修改性

可修改性是能够快速地对系统性能价格比进行变更的能力。在软件架构设计过程中，可以采用一系列的控制可修改性策略，让理解及修改相应模块变得容易，从而使系统的可修改性达到更高的水平。常见的策略包括以下两种。

(1) 局部化修改

- 维持语义的一致性：语义的一致性指的是模块中责任之间的关系。目标是确保所有这些责任都能够协同工作，不需要过多地依赖其他模块。如高内聚低耦合就是为了语义一致性而提出来的。
- 预期期望的变更：考虑所预想的变更的集合提供了一个评估特定的责任分配的方法。
- 泛化该模块：使一个模块更通用，能够使它根据输入计算更广泛的功能。
- 限制可能的选择：修改的范围可能非常大，因此可能会影响很多模块。限制可能的选择将会降低这些修改所造成的影响。

(2) 防止连锁反应

- 信息隐藏：把某个实体的责任分解为更小的部分，并选择使哪些信息成为公有的，哪些信息成为私有的。可以通过指定的接口获得公有的责任。
- 维持现有的接口：如果B依赖于A的一个接口，则维持该接口及其语法能够使B保持不变，这一点是显而易见的，所以在需要修改增加新的功能服务时，可以采用添加接口的方式达到目的，而不建议采用修改接口的方式。
- 限制通信路径：限制与一个给定的模块共享数据的模块。

3. 性能

性能是系统的响应能力，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。在软件架构设计过程中，可以采用一系列提升性能策略，让系统中的事务在更短的时间内得到响应。常见的策略包括：

(1) 资源需求方面

- 提高计算效率：处理事件或消息中的一个步骤就是应用某个算法，改进在关键的位置所使用的算法将减少等待时间。

- 减少计算开销：如果没有资源请求，就可以减少处理需求。
- 管理事件率：如果可以降低监视环境变量处的采样频率，就可以减少需求，提高性能。
- 限制执行时间：限制用多少执行时间对事件做出响应。例如，限制迭代的数量就是限制执行时间的一个方法。

(2) 资源管理方面

- 引入并发：如果可以并行处理请求，就可以减少闭锁时间。可以通过在不同的线程上处理不同的事件流或创建额外的线程来处理不同的活动集来引入并发。
- 维持数据或计算的多个副本：客户机-服务器模式中的客户机是计算的副本。使用副本的目的是减少在中央服务器上进行所有的计算时出现的争用。
- 增加可用资源：速度更快的处理器、额外的处理器、额外的内存等。

(3) 资源仲裁方面

在资源仲裁方面，主要是对资源使用调度算法的选用问题。

- 先进先出：先进先出（FIFO）队列同等看待对资源的所有请求，并依次对其进行处理。
- 固定优先级调度：每个请求资源的源都有一个固定的优先级，按优先级来分配资源。
- 动态优先级调度：动态优先级包括时间片轮转和时限时间最早优先等不同策略。

4. 安全性

安全性是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。在软件架构设计过程中，可以采用一系列手段来提高系统的安全性。常见的手段包括：

(1) 抵抗攻击

- 对用户进行身份验证：身份验证的方式包括密码、数据证书、生物识别等。
- 对用户进行授权：授权能够保证经过身份验证的用户有权访问和修改数据或服务。
- 维护数据的机密性：应该对数据进行保护，以防止未经授权的访问。
- 限制暴露的信息：攻击者通常会利用暴露的某个弱点来攻击主机上的所有数据和服务。所以在设计时，可以限制服务在主机上的分配，每个主机上只能获得受限的服务。
- 限制访问：防火墙根据消息源或目的地端口来限制访问。

(2) 检测攻击

通过布署入侵检测系统来检测攻击。

(3) 从攻击中恢复

如何在攻击发生后，迅速恢复也是安全性设计时需要考虑的一个重要方面。

5. 可测试性

可测试性是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。在软件架构设计时，强调可测试性主要是为了在完成一个软件开发的增量后，能较为轻松地对软件进行测试。下面将介绍一些增强可测试性的策略。

- 将接口与实现分离：将接口与实现分离利于测试的进行。这样我们可以在缺失某个组件时，使用其替代品先占位，然后对系统的剩余部分进行测试。
- 特化访问接口：具有特化的测试接口允许通过测试工具并独立于其正常操作，来捕获或指定组件的变量值。

6. 易用性

易用性用于衡量用户使用一个软件产品完成指定任务的难易程度。要提高软件的易用性一般可以考虑将用户接口与应用的其余部分分离开来。

8.1.11 软件架构评估

软件架构设计是软件开发过程中关键的一步。对于当今世界上庞大而复杂的系统来说，没有一个合适的架构而要有一个成功的软件设计几乎是不可想象的。不同类型的系统需要不同的架构，甚至一个系统的不同子系统也需要不同的架构。架构的选择往往会成为一个系统设计成败的关键。但是，怎样才能知道为系统所选用的架构是否恰当呢？如何确保按照所选用的架构能顺利地开发出成功的软件产品呢？要回答这些问题并不容易，因为它受到很多因素的影响，需要专门的方法来对其进行评估。

1. 架构评估概述

软件架构评估可以只针对一个架构，也可以针对一组架构。在架构评估过程中，评估人员所关注的是系统的质量属性。

为了后面讨论的需要，本节先介绍两个概念，分别是敏感点（Sensitivity Point）和权衡点（Tradeoff Point）。敏感点是一个或多个构件（和/或构件之间的关系）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。例如，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。

从目前已有的软件架构评估技术来看，可以归纳为3类主要的评估方式，分别是基于调查问卷（或检查表）的方式、基于场景的方式和基于度量的方式。

（1）基于调查问卷（或检查表）的评估方式

基于调查问卷的评估方式与问卷调查是类似的，只不过这里需要调查的不是软件需求，而是有关软件架构的问题。这些问题可能涉及架构设计决策，也可能涉及架构文档，例如，架构的表示用的是何种架构描述语言（Architecture Description Language, ADL）；有的问题针对架构描述本身的细节，例如，系统的核心功能是否与界面分开。检查表中也包含一系列比调查问卷更细节和具体的问题，它们更趋向于考查某些关心

的质量属性。例如，对实时系统的性能进行考查时，很可能问到系统是否反复多次地将同样的数据写入磁盘等。

这一评估方式比较自由、灵活，可评估多种质量属性，也可以在软件架构设计的多个阶段进行。但是，由于评估的结果很大程度上来自评估人员的主观推断，因此，不同的评估人员可能会产生不同甚至截然相反的结果，而且评估人员对领域的熟悉程度、是否具有丰富的相关经验也成为评估结果是否正确的重要因素。

(2) 基于场景的评估方式

基于场景的评估方式主要应用在架构权衡分析法（Architecture Tradeoff Analysis Method, ATAM）、软件架构分析法（Software Architecture Analysis Method, SAAM）和成本效益分析法（Cost Benefit Analysis Method, CBAM）中。在架构评估中，一般采用刺激（Stimulus）、环境（Environment）和响应（Response）3方面来对场景进行描述。刺激是场景中解释或描述项目干系人怎样引发与系统的交互部分，环境描述的是刺激发生时的情况，响应是指系统是如何通过架构对刺激作出反应的。

基于场景的方式分析软件架构对场景的支持程度，从而判断该架构对这一场景所代表的质量需求的满足程度。例如，用一系列对软件的修改来反映易修改性方面的需求，用一系列攻击性操作来代表安全性方面的需求等。这一评估方式考虑到了所有与系统相关的人员对质量的要求，涉及的基本活动包括确定应用领域的功能和软件架构之间的映射，设计用于体现待评估质量属性的场景，以及分析软件架构对场景的支持程度。

不同的系统对同一质量属性的理解可能不同，例如，对操作系统来说，可移植性被理解为系统可在不同的硬件平台上运行，而对于普通的应用系统而言，可移植性往往是指该系统可在不同的操作系统上运行。由于存在这种不一致性，对一个领域适合的场景设计在另一个领域内未必合适，因此，基于场景的评估方式是特定于领域的。这一评估方式的实施者一方面需要有丰富的领域知识，以对某一质量需求设计出合理的场景；另一方面，必须对待评估的软件架构有一定的了解，以准确判断它是否支持场景描述的一系列活动。

(3) 基于度量的评估方式

度量是指为软件产品的某一属性所赋予的数值，例如，代码行数、方法调用层数和构件个数等。传统的度量研究主要是针对代码的，但近年来也出现了一些针对高层设计的度量，软件架构度量即是其中之一。

基于度量的评估技术都涉及3个基本活动。首先，需要建立质量属性和度量之间的映射原则，即确定怎样从度量结果推导出系统具有什么样的质量属性；然后，从软件架构文档中获取度量信息；最后，根据映射原则分析、推导出系统的某些质量属性。基于度量的评估方式提供更为客观和量化的质量评估，需要在软件架构的设计基本完成以后才能进行，而且需要评估人员对待评估的架构十分了解，否则不能获取准确的度量。

2. ATAM 评估方法

使用 ATAM 方法对软件架构进行评估的目的，是依据系统质量属性和业务需求评估设计决策的结果。ATAM 希望揭示出架构满足特定质量目标的情况，使架构设计师

更清楚地认识到质量目标之间的联系，即如何权衡多个质量目标。这些设计决策很重要，一直会影响到整个软件生命周期，并且在软件实现后很难修改这些决策。

(1) 评估参与者

在 ATAM 方法中，参加评估的人员主要有评估小组、项目决策者和其他项目干系人。

- 评估小组：该小组是所评估架构项目外部的小组，通常由 3~5 人组成，他们可能是开发组织内部的，也可能是外部的。评估小组的每个成员都要扮演大量的特定角色。
- 项目决策者：项目决策者对开发项目具有发言权，并有权要求进行某些改变，他们包括项目管理人员、重要的客户代表和架构设计师等。
- 项目干系人：包括关键模块开发人员、测试人员和用户等。

(2) 评估活动

整个 ATAM 评估过程包括 9 个步骤，如图 8-12 所示。

① 描述 ATAM 方法。评估小组负责人向参加会议的项目干系人介绍 ATAM 评估方法。在这一步中，要解释每个人将要参与的过程，并预留出解答疑问的时间，设置好其他活动的环境和预期结果。关键是要使每个人都知道要收集哪些信息，如何描述这些信息，将要向谁报告等。

② 描述业务动机。项目决策者从业务的角度介绍系统的概况，该描述应该包括系统最重要的功能、技术/管理/经济和政治方面的任何相关限制、与该项目相关的业务目标和上下文、主要的项目干系人，以及架构的驱动因素等。参加评估的所有人员必须理解待评估的系统。

③ 描述架构。首席设计师或设计小组要对架构进行详略适当的介绍，至少应该包括技术约束（例如，操作系统、硬件和中间件等）、将与本系统进行交互的其他系统、用以满足质量属性要求的架构方法等。这一步很重要，将直接影响到可能要做的分析及分析的质量。

④ 确定架构方法。ATAM 评估方法主要通过理解架构方法来分析架构，在这一步，由架构设计师确定架构方法，由分析小组捕获，但不进行分析。

⑤ 生成质量属性效用树。评估小组、设计小组、管理人员和客户代表一起确定系统最重要的质量属性目标，并对这些质量目标设置优先级和细化。这一步很关键，它对以后的分析工作起指导作用。即使是架构级的分析，也不一定是全局的，所以，需要集中所有相关人员的精力，注意架构的各个方面，这通常是通过构建效用树的方式来实现的。效用树的输出结果是对具体质量属性需求的优先级的确定，这种优先级

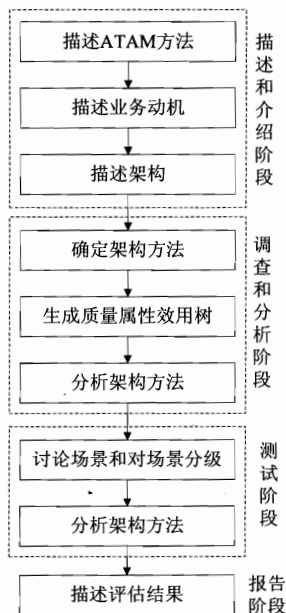


图 8-12 ATAM 方法的步骤

列表为 ATAM 评估方法的后面几步提供了指导,它告诉评估小组应该把有限的时间花在哪里,特别是应该到哪里去考察架构的方法与相应的风险、敏感点和权衡点。

⑥ 分析架构方法。一旦有了效用树的结果,评估小组可以对实现重要质量属性的架构方法进行考察。这是通过文档化这些架构决策和确定它们的风险、敏感点和权衡点等来实现的。在这一步中,评估小组要对每一种架构方法都考察足够的信息,完成与该方法有关的质量属性的初步分析。这一步的主要结果是一个架构方法或风格的列表,与之相关的一些问题,以及设计师对这些问题的回答。通常产生一个风险列表、敏感点和权衡点列表。在这一步结束时,评估小组应该对整个架构的绝大多数重要方面所做出的关键设计决策、风险列表、敏感点、权衡点有一个清楚的认识。

⑦ 讨论场景和对场景分级。场景在驱动 ATAM 测试阶段起主导作用。项目干系人进行两项相关的活动,分别是集体讨论用例场景和改变场景。用例场景是场景的一种,在用例场景中,项目干系人是一个终端用户,使用系统执行的一些功能。一旦收集了若干个场景后,必须设置优先级。评估人员通过投票表决的方式来完成,每个项目干系人分配相当于总场景数的 30% 的选择,且此数值只入不舍。例如,如果共有 17 个场景,则每个风险承担者将拿到 6 张选票,这 6 张选票的具体使用则取决于项目干系人,他可以把这 6 张票全部投给一个场景,或者每个场景 2~3 张票,还可以一个场景一张票等。

⑧ 分析架构方法。在收集并分析了场景之后,设计师就可把最高级别的场景映射到所描述的架构中,并对相关的架构如何有助于该场景的实现做出解释。在这一步中,评估小组要重复第⑥步中的工作,把新得到的最高优先级场景与尚未得到的架构工作产品对应起来。在第⑦步中,如果未产生任何在以前的分析步骤中都没有发现的高优先级场景,则在第⑧步就是测试步骤。

⑨ 描述评估结果。最后,要把 ATAM 分析中所得到的各种信息进行归纳,并反馈给项目干系人。这种描述一般要采用辅以幻灯片的形式,但也可以在 ATAM 评估结束之后,提交更完整的书面报告。在描述过程中,评估负责人要介绍 ATAM 评估的各个步骤,以及各步骤中得到的各种信息,包括业务环境、驱动需求、约束条件和架构等。最重要的是要介绍 ATAM 评估的结果。ATAM 的评估结果包括一个简洁的架构描述、表达清楚的业务目标、用场景集合捕获的质量属性、所确定的敏感点和权衡点的集合、有风险决策和无风险决策、风险主题的集合。

在具体的软件架构评估过程中,可以修改这 9 个步骤的顺序,以满足架构信息的特殊需求。也就是说,虽然这九个步骤按编号排列,但并不总是一个瀑布过程,评估人员可在这 9 个步骤中跳转或进行迭代。

3. CBAM 评估方法

CBAM 用来对架构设计决策的成本和收益进行建模,其基本思想是架构策略影响系统的质量属性,反过来这些质量属性又会为系统的项目干系人带来一些收益(称为“效用”),CBAM 协助项目干系人根据其投资收益率选择架构策略。CBAM 可以看做是 ATAM 的补充,在 ATAM 评估结果的基础上对架构的经济性进行评估。CBAM 的评估步骤大致如下:

① 整理场景。整理 ATAM 中获取的场景，根据业务目标确定这些场景的优先级，并选取优先级最高的 1/3 的场景进行分析。

② 对场景进行细化。为每个场景获取最坏情况、当前情况、期望情况和最好情况的质量属性响应级别。

③ 确定场景的优先级。项目干系人对场景进行投票，其投票是基于每个场景所期望的响应值，根据投票结果和票的权值，生成一个分值（场景的权值）。

④ 分配效用。对场景的响应级别确定效用表，根据架构策略涉及哪些质量属性及响应级别，形成相关的“策略-场景-响应级别”的对应关系。

⑤ 确定期望的质量属性响应级别的效用。即根据第⑤步的效用表及其对应关系，确定架构策略及其对应场景的效用表。

⑥ 计算各架构策略的总收益。根据第③步的场景的权值和第⑤步的架构策略效用表，计算出架构策略的总收益得分。

⑦ 根据受成本限制影响的投资收益率选择架构策略。根据开发经验估算架构策略的成本，结合第⑥步的收益，计算出架构策略的投资收益率，按投资收益率排序，从而确定选取策略的优先级。

4. SAAM 评估方法

SAAM 方法是最早形成文档并得到广泛使用的软件架构分析方法，最初是用来分析架构的可修改性的，但实践证明，SAAM 方法也可用于对许多其他质量属性及系统功能进行快速评估。SAAM 方法的目的是验证基本的架构假设和原则，评估架构固有的风险。SAAM 指导对架构的检查，使其主要关注潜在的问题点，如需求冲突等。SAAM 不仅能够评估架构对于特定系统需求的使用能力，也能用来比较不同的架构。

(1) 评估活动

与 ATAM 方法相比，SAAM 比较简单，这种方法易学易用，进行培训和准备的工作量都比较少。SAAM 评估可以分 6 个步骤进行，如图 8-13 所示。

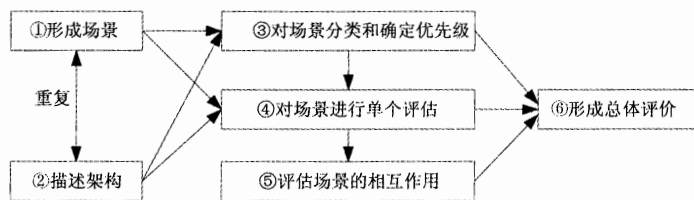


图 8-13 SAAM 方法的步骤

在这些步骤进行之前，通常有必要对系统做简要的介绍，包括对架构的业务目标的说明等。

① 形成场景。在形成场景的过程中，要注意全面捕捉系统的主要用途、系统用户类型、系统将来可能的变更、系统在当前及可预见的未来必须满足的质量属性等信息。只有这样，形成的场景才能代表与各种项目干系人相关的任务。形成场景是通过集中讨论来实现的，使项目干系人在一个友好的氛围中提出一些场景，这些场景反映

了他们的需求，也体现了他们对架构将如何实现需求的认识。

② 描述架构。在这一步，架构设计师应该采用参加评估的所有人员都能够充分理解的形式，对待评估的架构进行适当的描述。这种描述必须要说明系统中的运算和数据构件，以及它们之间的联系。除了要描述这些静态特性外，还要对系统在某段时间内的动态特征做出说明。描述既可采用自然语言，也可采用形式化的手段。

③ 场景的分类和优先级确定。场景可分为直接场景和间接场景（潜在场景）。直接场景是按照现有架构开发出来的系统能够直接实现的场景。与在设计时已经考虑过的需求相对应的直接场景能增进对架构的理解，促进对诸如性能和可靠性等其他质量属性的研究；间接场景就是需要对现有架构做某些修改才能支持的场景。间接场景对衡量架构对系统在演化过程中将出现的变更的适用情况十分关键。通过各种间接场景对架构的影响，可以确定架构在相关系统的生命周期内对不断演化的使用的适应情况。直接场景类似于用例，而间接场景有时也叫变更案例。评估人员通过对场景设置优先级，可以保证在评估的有限时间内考虑最重要的场景。这里的“重要”完全是由项目干系人及其所关心的问题确定的。项目干系人可以通过投票表达所关心的问题。

④ 对场景进行单个评估。一旦确定了要考虑的一组场景，就要把这些场景与架构的描述对应起来。对于直接场景而言，架构设计师需要讲清所评估的架构将如何执行这些场景；对于间接场景而言，设计师应说明需要对架构做哪些修改才能适应间接场景的要求。

⑤ 评估场景的相互作用。场景的相互作用暴露了设计方案中的功能分配。场景相互作用的多少与结构复杂性、耦合度、内聚性有关。同时，场景的相互作用能够暴露出架构设计文档未能充分说明的结构分解。

⑥ 形成总体评估。最后，评估人员要对场景和场景之间的交互作一个总体的权衡和评价，这一权衡反映组织对表现在不同场景中的目标的考虑优先级。根据对系统成功的相对重要性来为每个场景设置一个权值，权值的确定通常要与每个场景所支持的业务目标联系起来。

如果要比较多个架构，或者针对同一架构提出了多个不同的方案，则可通过权值的确定来得出总体评价。权值的设置具有很强的主观性，所以，应该让所有项目干系人共同参与，但也应合理组织，要允许对权值及其基本思想进行公开讨论。

（2）评估结果

SAAM 评估的主要有形输出包括以下两个方面：

① 把代表了未来可能做的更改的场景与架构对应起来，显现出架构中未来可能会表现出较高复杂性的地方，并对每个这样的更改的预期工作量做出评估。

② 理解系统的功能，对多个架构所支持的功能和数量进行比较。

如果所评估的是一个框架，SAAM 评估将指明框架中未能满足其修改性需求的地方，有时还会指出一种效果更好的设计。SAAM 评估也能对多个备选架构进行比较，明确其中哪一个架构能够较好地满足质量属性，而且做的更改较少、不会在未来导致太多的复杂的问题。

8.1.12 设计模式

所谓设计模式，简单地理解，是一些设计面向对象的软件开发的经验总结。一个设计模式事实上是系统地命名、解释和评价某一个重要的可重现的面向对象的设计方案。

在里程碑性著作《设计模式：可复用面向对象软件的基础》一书中首次将设计模式的思想应用于软件设计，描述了收录和描述设计模式的一个结构，收录了 23 个模式，并在这些模式的基础上提出了面向对象策略及方法的原则。

它将这 23 种模式分为 3 类：创建型模式，包括工厂方法（Factory Method）、抽象工厂（Abstract Factory）、单例（Singleton）、构建（Builder）、原型（Prototype）模式；结构型模式，包括适配器（Adapter）、合成（Composite）、装饰（Decorator）、代理（Proxy）、享元（Flyweight）、门面（Façade）、桥接（Bridge）模式；行为型模式，包括策略（Strategy）、模板方法（Template Method）、迭代器（Iterator）、责任链（Chain of Responsibility）、命令（Command）、备忘录（Memento）、状态（State）、访问者（Visitor）、解释器（Interpreter）、调停者（Mediator）、观察者（Observer）模式。

1. 简单工厂

简单工厂（Simple Factory）专门定义一个类的实例，被创建的实例通常都具有共同的父类。简单工厂模式又称为静态工厂方法（Static Factory Method）模式，属于创建型模式，通常它根据自变量的不同返回不同类的实例。

简单工厂模式的实质是由一个工厂类根据传入的参量，动态决定应该创建出哪一个产品类的实例。简单工厂模式实际上不属于 GoF 的 23 个模式，但它可以作为 GoF 的工厂方法（Factory Method）模式的一个引导。其 UML 类模型如图 8-14 所示，其涉及的参与者有 3 个：工厂角色、抽象产品角色和具体产品角色。

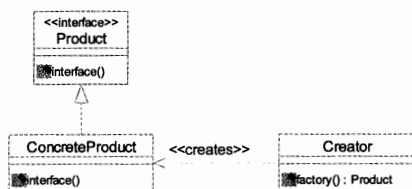


图 8-14 简单工厂模式的 UML 类模型

- 工厂（Creator）角色：是简单工厂模式的核心，它负责实现创建所有实例的内部逻辑。工厂类可以被外界直接调用，创建所需的产品对象。
- 抽象产品（Product）角色：是简单工厂模式所创建的所有对象的父类，它负责描述所有实例所共有的公共接口。
- 具体产品（Concrete Product）角色：是简单工厂模式的创建目标，所创建的对象都是充当这个角色的某个具体类的实例。

2. 工厂方法模式

工厂（Factory）模式又称为工厂方法模式，也叫虚拟构造（Virtual Constructor）函数模式或者多态工厂模式。在工厂模式中，父类负责定义创建对象的公共接口，而子类则负责生成具体的对象，这样做的目的是将类的实例化操作延迟到子类中完成，即由子类来决定究竟应该实例化哪一个类。

在简单工厂模式中，一个工厂类处于对产品类进行实例化的中心位置，它知道每个产品类的细节，并决定何时、哪一个产品类应当被实例化。简单工厂模式的优点是能够使客户端独立于产品的创建过程，并且在系统中引入新产品时无须对客户端进行修改，缺点是当有新产品要加入到系统中时，必须修改工厂类，以加入必要的处理逻辑。简单工厂模式的致命弱点就是处于核心地位的工厂类，因为一旦它无法确定要对哪个类进行实例化，就无法使用该模式，而工厂方法模式则可以很好地解决这一问题。工厂方法模式的 UML 类模型如图 8-15 所示。

其中的类或对象之间的关系如下。

- 产品（Product）角色：定义产品的接口。
- 真实产品（Concrete Product）角色：实现接口 Product 的类。
- 工厂（Creator）角色：声明工厂方法（Factory Method），返回一个产品。
- 真实工厂（Concrete Creator）角色：实现工厂方法（Factory Method），则客户调用，返回一个产品的实例。

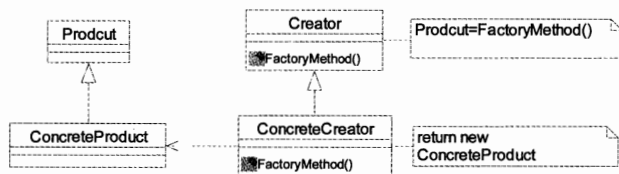


图 8-15 工厂方法模式的 UML 类模型

3. 抽象工厂

抽象工厂提供了一个创建一系列相关或相互依赖对象的接口，而无须指定它们具体的类。抽象工厂（Abstract Factory）模式又称为 Kit 模式，属于对象创建型模式。

抽象工厂模式与工厂模式最大的区别在于：工厂模式针对的是一个产品的等级结构，而抽象工厂模式则针对的是多个产品等级结构。正因为如此，在抽象工厂模式中经常会用到产品族这一概念，它指的是位于不同的产品等级结构中，并且功能相互关联的产品系列，其 UML 类模型如图 8-16 所示。其中的类或对象之间的关系如下。

- 抽象工厂（Abstract Factory）：声明生成抽象产品的方法。
- 具体工厂（Concrete Factory）：执行生成抽象产品的方法，生成具体的产品。

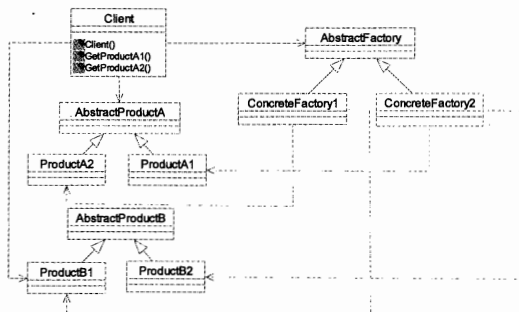


图 8-16 抽象工厂模式的 UML 类模型

4. 单例模式

单例（Singleton）模式确保其一个类只有一个实例，而且自行实例化并向整个系统提供这个实例。这个类称为单例类，它提供全局访问的方法。单例模式的要点有 3 个：一是某个类只能有一个实例；二是它必须自选创建这个实例；三是它必须自行向整个系统提供这个实例。单例模式的 UML 类模型如图 8-17 所示。

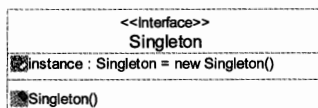


图 8-17 单例模式的 UML 类模型

类和对象之间的关系如下。

- 单例（Singleton）：提供一个 instance 方法，让客户可以使用它的唯一实例。内部实现只生成一个实例。

5. 构建模式

构建（Builder）模式将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。构建模式是一步步创建一个复杂的对象，它允许用户只通过指定复杂对象的类型和内容就可以构建它们，用户不知道内部的具体构建细节。构建模式的 UML 类模型如图 8-18 所示。

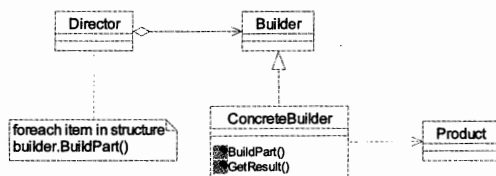


图 8-18 构建模式的 UML 类模型

其中的类或对象之间的关系如下。

- 抽象构建者（Builder）：为创建一个 Product 对象的各个部件指定抽象接口。
- 具体构建者（Concrete Builder）：实现 Builder 接口，构造和装配产品的各个部件；定义并明确它所创建的表示；提供一个返回这个产品的接口。
- 指挥者（Director）：构建一个使用 Builder 接口的对象。
- 产品（Product）：被构建的复杂对象，具体构建者创建该产品的内部表示并定义它的装配过程；包含定义组成部件的类，包括将这些部件装配成最终产品的接口。

6. 原型模式

原型（Prototype）模式指定创建对象的种类，并且通过复制这些原型创建新的对象。原形模式允许一个对象再创建另一个可定制的对象，根本无须知道任何创建的细节。工作原理是：通过将一个原型对象传给那个要发动创建的对象，这个要发动创建的对象通过请求原型对象复制自己来实施创建过程。原型模式的 UML 类模型如图 8-19 所示。

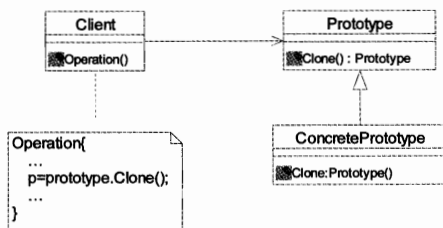


图 8-19 原型模式的 UML 类模型

其中类和对象之间的关系为：

- 抽象原型（Prototype）类：定义具有克隆自己的方法的接口。
- 具体原型（ConcretePrototype）类：实现具体的克隆方法。
- 客户（Client）类：通过克隆生成一个新的对象。

7. 适配器模式

适配器（Adapter）模式将一个接口转换成为客户想要的另一个接口，适配器模式使接口不兼容的那些类可以一起工作。适配器模式的 UML 类模型如图 8-20 所示。

其中类的定义及其关系如下。

- 目标（Target）抽象类：定义客户要用的特定领域的接口。
- 适配器公接口（Adapter）：调用另一个接口，作为一个转换器。
- 适配器母接口（Adaptee）：Adapter 需要接入。
- 客户（Client）调用类：协同对象符合 Adapter 适配器（公接口）。

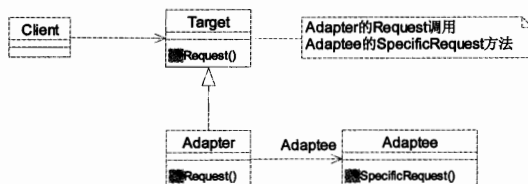


图 8-20 适配器模式的 UML 类模型

8. 合成模式

合成（Composite）模式组合多个对象形成树形结构以表示整体-部分的结构层次。合成模式对单个对象和合成对象的使用具有一致性。合成模式的 UML 类模型如图 8-21 所示。

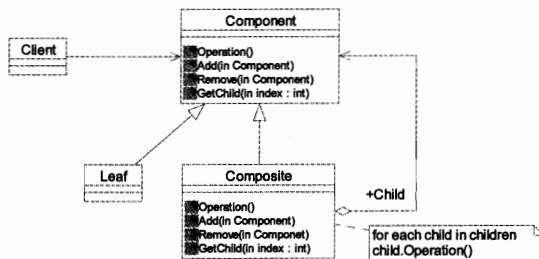


图 8-21 合成模式的 UML 类模型

其中类和对象的关系如下。

- 部件抽象接口 (Component)：为合成的对象声明接口；某些情况下，实现从此接口派生出所有类共有的默认行为；定义一个接口可以访问及管理它的多个部分 (GetChild)；如果有必要也可以在递归结构中定义一个接口访问它的父节点，并且实现它。
- 叶子部件 (Leaf)：在合成中表示叶节点对象，叶节点没有子节点；定义合成中原接口对象的行为。
- 合成 (Composite) 类：定义有子节点 (子部件) 的部件的行为；存储子节点 (子部件)；在 Component 接口中实现与子部件相关的操作。
- 客户 (Client) 应用程序：通过 Component 接口控制组合部分的对象。

9. 装饰模式

装饰 (Decorator) 模式动态地给一个对象增加其他职责，就增加对象功能来说，装饰模式比生成子类实现更为灵活。装饰模式的 UML 类模型如图 8-22 所示。

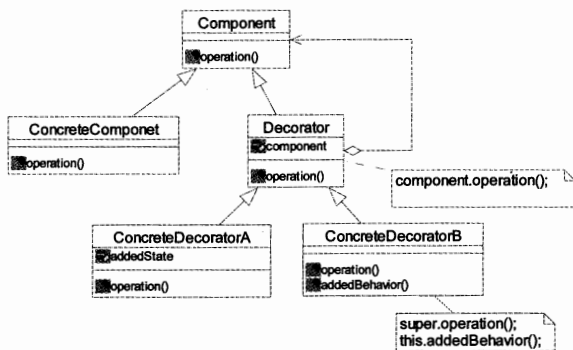


图 8-22 装饰模式的 UML 类模型

其中类和对象的关系如下。

- 部件 (Component)：定义对象的接口，可以给这些对象动态增加职责 (方法)。
- 具体部件 (Concrete Component)：定义具体的对象，Decorator 可以给它增加额外的职责 (方法)。
- 装饰抽象 (Decorator) 类：维护一个内有的 Component，并且定义一个与 Component 接口一致的接口。
- 具体装饰对象 (Concrete Decorator)：为内在的具体部件对象增加具体的职责 (方法)。

10. 代理模式

代理 (Proxy) 模式为其他对象提供一个代理或地方以控制对这个对象的访问。当客户向代理对象第一次提出请求时，将实例化为真实的对象，并且将请求传给它，以后所有的客户请求都经由 Proxy 传给封装了的真实对象。代理模式的其 UML 类模型如图 8-23 所示。

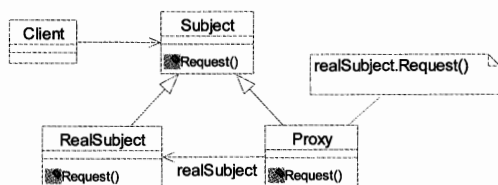


图 8-23 代理模式的 UML 类模型

其中的类和对象的关系如下。

- 代理（Proxy）：维护一个引用使得代理可以访问实体，如果 RealSubject 和 Subject 的接口相同，Proxy 会引用 Subject；提供一个与 Subject 的接口相同的接口，使得代理可以用来替代实体，这与适配器模式类似；控制对实体的访问并且负责创建及删除它。
- 抽象实体（Subject）接口：为 RealSubject 实体及 Proxy 代理定义相同的接口中，使得 RealSubject 在任何地方都可以使用 Proxy 来访问。
- 实体（RealSubject）：定义 Proxy 代理的实体。

11. 享元模式

享元（Flyweight）模式运用共享技术有效地支持大量细粒度的对象。系统只使用少量的对象，而这些对象都相近、状态变化很小、对象使用次数较多。享元模式的 UML 类模型如图 8-24 所示。

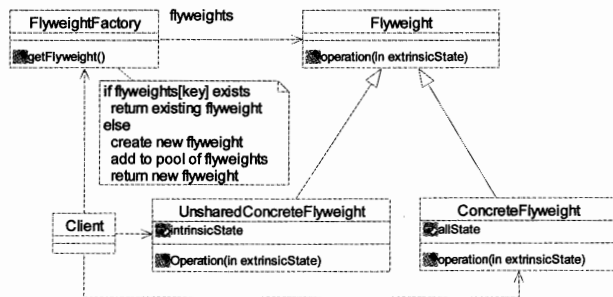


图 8-24 享元模式的 UML 类模型

其中类和对象的关系如下。

- 享元（Flyweight）类：声明一个接口，通过它可以接收外来的参数（状态），并对新状态做出处理（作用）。
- 具体享元（Concrete Flyweight）类：实现 Flyweight 的接口，并为内部增加存储空间。Concrete Flyweight 对象必须是可以共享的，它所存储的状态必须是内部的，即它独立存在于自己的环境中。
- 不共享的具体享元（Unshared Concrete Flyweight）类：不是所有的 Flyweight 子类都需要被共享，Flyweight 的共享不是强制的。在某些 Flyweight 的结构层次中，Unshared Concrete Flyweight 对象通常将 Concrete Flyweight 对象作为子节点。
- 享元类工厂（Flyweight Factory）：创建并管理 Flyweight 对象；确保享用

Flyweight, 当用户请求一个 Flyweight 对象时, FlyweightFactory 提供一个已创建的实例或者创建一个实例 (如果不存在)。

- 客户应用程序 (Client): 维持一个对 Flyweight 的引用; 计算或存储一个或多个 Flyweight 的外部状态。

12. 门面模式

门面 (Facade) 模式也称为外观模式, 提供一个统一的接口去访问多个子系统的多个不同的接口。门面模式定义了一个高层次的接口, 使得子系统更容易被使用。门面模式的 UML 类模型如图 8-25 所示。

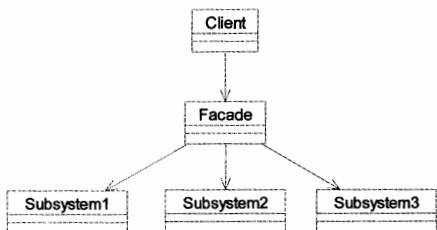


图 8-25 门面模式的 UML 类模型

其中类和对象的关系如下。

- 门面 (Facade) 类: 知道哪些子系统负责处理哪些请求; 将客户的请求传递给相应的子系统对象处理。
- 子系统 (Subsystem) 类: 实现子系统的功能; 处理由 Facade 传过来的任务; 子系统不用知道 Facade, 在任何地方都没有引用 Facade。

13. 桥接模式

桥接 (Bridge) 模式将抽象部分与实现部分分离, 使得它们两部分可以独立地变化。桥接模式的 UML 类模型如图 8-26 所示。

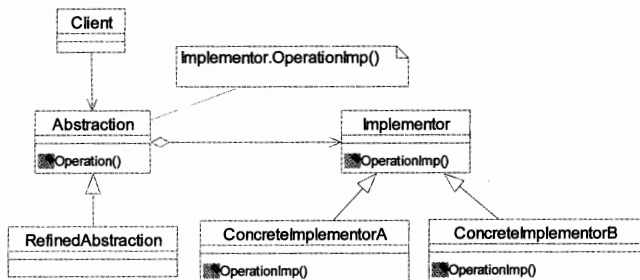


图 8-26 桥接模式的 UML 类模型

其中类和对象的关系如下。

- 抽象 (Abstraction) 类: 定义抽象类的接口, 维护一个实现抽象 (Implementor) 的对象。
- 扩充抽象 (Refined Abstraction) 类: 扩充由 Abstraction 定义的接口。
- 实现 (Implementor) 类接口: 定义实现类的接口, 这个接口不一定要与

Abstraction 的接口完全一致，事实上这两个接口可以完全不同，一般地讲，Implementor 接口仅提供基本操作，而 Abstraction 定义的接口可能会做更多更复杂的操作。

- 具体实现（ConcreteImplementor）类：具体实现 Implementor 的接口。

14. 策略模式

策略（Strategy）模式定义一系列的算法，将每一个算法封装起来，并让它们可以相互替换。策略模式让算法独立于使用它的客户而变化。策略模式的 UML 类模型如图 8-27 所示。

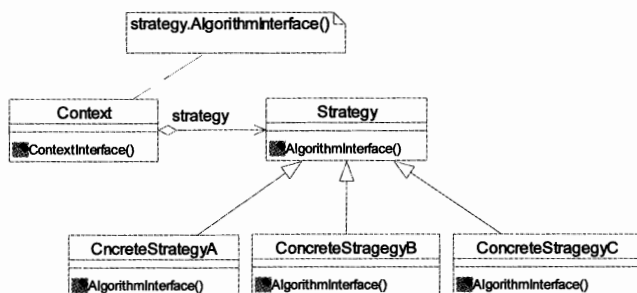


图 8-27 策略模式的 UML 类模型

其中类和对象的关系如下。

- 抽象策略（Strategy）类：定义一个公共的接口给所有支持的算法，Context 使用这个接口调用 ConcreteStrategy 定义的算法。
- 具体策略（Concrete Strategy）类：调用 Strategy 接口实现具体的算法。
- 上下文（Context）：用 ConcreteStrategy 对象配置其执行环境；维护一个对 Strategy 的引用实例；可以定义一个接口供 Strategy 存取其数据。

15. 模板方法模式

模板方法（Template Method）模式定义一个操作中算法的骨架，以将一些步骤延缓到子类中实现。模板方法让子类重新定义一个算法的某些步骤而无须改变算法的结构。模板方法模式的 UML 类模型如图 8-28 所示。

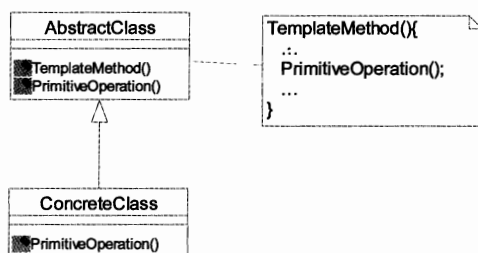


图 8-28 模板方法模式的 UML 类模型

其中类和对象的关系如下。

- 抽象类（AbstractClass）：定义一个抽象原始的操作，其子类可以重定义实

现算法的各个步骤；实现一个模板方法定义一个算法的骨架，此模板方法不仅可以调用原始的操作，还可以调用定义于 `AbstractClass` 中的方法或其他对象中的方法。

- 具体类（`ConcreteClass`）：实现原始的操作以完成子类特定算法的步骤。

16. 迭代器模式

迭代器（`Iterator`）模式提供一种方法可以访问聚合对象，而不用暴露这个对象的内部表示。迭代器模式的 UML 类模型如图 8-29 所示。

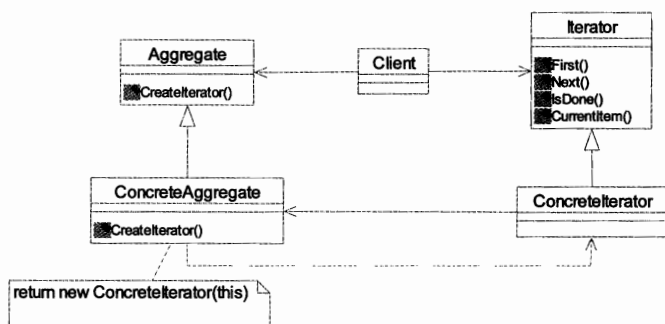


图 8-29 迭代器模式的 UML 类模型

其中类和对象的关系如下。

- 迭代器（`Iterator`）：定义访问和遍历元素的接口。
- 具体迭代器（`ConcreteIterator`）：实现迭代器的接口；在遍历时跟踪当前聚合对象中的位置。
- 聚合（`Aggregate`）：定义一个创建迭代器对象的接口。
- 具体聚合（`ConcreteAggregate`）：创建迭代器对象，返回一个具体迭代器实例。

17. 责任链模式

责任链（`Chain of Responsibility`）模式能避免请求发送者与接收者耦合在一起，让多个对象都有可能接收请求，将这些对象连接成一条链，并且沿着这条链传递请求，直到有对象处理为止。责任链模式的 UML 类模型如图 8-30 所示。

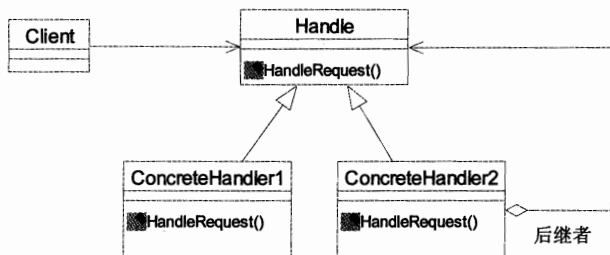


图 8-30 责任链模式的 UML 类模型

其中类和对象的关系如下。

- 传递者（`Handler`）接口：定义一个处理请求的接口；或实现链中下一个对象。

- 具体传递者（ConcreteHandler）：处理它所负责的请求；可以访问链中下一个对象；如果可以处理请求，就处理，否则将请求转发给后继者。
- 客户应用程序（Client）：向链中的对象提出最初的请求。

18. 命令模式

命令（Command）模式将一个请求封装成一个对象，因此可以参数化多个客户的不同请求，将请求排队，记录请求日志，并且支持撤销操作。命令模式的 UML 类模型如图 8-31 所示。

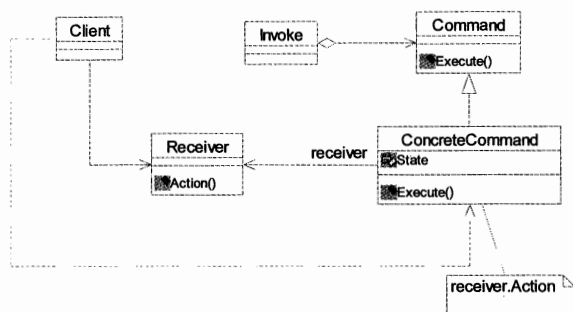


图 8-31 命令模式的 UML 类模型

其中类和对象之间的关系如下。

- 抽象命令（Command）类：声明执行操作的一个接口。
- 具体命令（ConcreteCommand）类：将一个接收者对象绑定于一个动作；实现 Execute 方法，以调用接收者的相关操作（Action）。
- 客户应用程序（Client）：创建一个具体命令类的对象，并且设定它的接收者。
- 调用者（Invoker）：要求一个命令对象执行一个请求。
- 接收者（Receiver）：知识如何执行关联请求的相关操作。

19. 备忘录模式

备忘录（Memento）模式在不破坏封装的前提下，捕获并且保存一个对象的内部状态，这样可以将对象恢复到原先保存的状态。备忘录模式的 UML 类模型如图 8-32 所示。

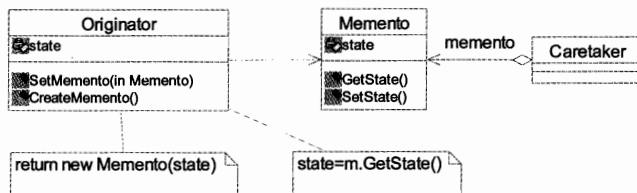


图 8-32 备忘录模式的 UML 类模型

其中类和对象的关系如下。

- 备忘录（Memento）：保持 Originator（原发器）的内部状态，根据原发器来决定保存哪些内部的状态；保护原发器之处的对象访问备忘录，备忘录可以

有效地利用两个接口，看管者只能调用狭窄（功能有限）的接口，即只能传递备忘录给其他对象，而原发器可以调用一个宽阔（功能强大）的接口，通过这个接口可以访问所有需要的数据，使原发器可以返回先前的状态。理想的情况是，只允许生成本备忘录的那个原发器访问本备忘录的内部状态。

- 原发器（Originator）：创建一个备忘录，记录它的当前内部状态；可以利用一个备忘录来恢复它的内部状态。
- 看管者（Caretaker）：只负责看管备忘录；不可以对备忘录的内容操作或检查。

Memento 模式经常在很多应用软件出现，按[Ctrl+Z]组合键会取消最后一次用户操作。如果不用 Memento 模式，Caretaker 对象要备份 Originator 对象的状态，Originator 就要具有所有需要访问成员的方法，当要恢复 Originator 对象的状态时，Caretaker 更要清楚 Originator 内部的结构。这种严紧的凝结的结果是，发生在 Originator 上的任何修改，Caretaker 都要作出相应的修改，这样就打破了对象封装的特点。如图 8-33 所示是没有采用 Memento 模式时的结构。

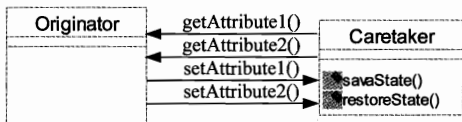


图 8-33 未采用备忘录模式时的 UML 类模型

Memento 模式是解决这种问题的最好办法，Memento 类成了 Originator 及 Caretaker 的媒介，封装保存 Originator 的备份状态，当 Originator 被提出备份请求时，它就会创建一个 Memento 对象返回给 Caretaker。Caretaker 不可以看到 Memento 对象的内部信息，需要时 Caretaker 可以返回备份的 Memento 对象给 Originator，让它恢复到备份状态，结构如图 8-34 所示。

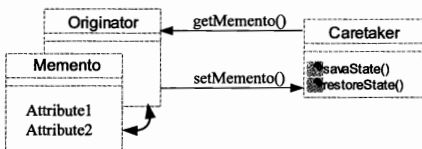


图 8-34 采用备忘录模式后的 UML 类模型

20. 状态模式

状态（State）模式能够使一个对象的内在状态改变时允许改变其行为，使这个对象看起来像是改变了其类。状态模式的 UML 类模型如图 8-35 所示。

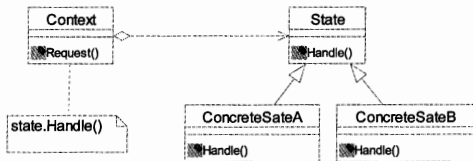


图 8-35 状态模式的 UML 类模型

其中类和对象的关系如下。

- 上下文(Context)类: 定义客户应用程序有兴趣的接口; 维护一个 ConcreteState (具体状态) 子类的实例对象。
- 抽象状态(State)类: 定义一个接口以封装与 Context 的一个特别状态(State) 相关的行为。
- 具体状态(ConcreteState)类: 每一个具体状态类实现一个 Context 的状态相关的行为。

21. 访问者模式

访问者(Visitor)模式说明一个操作执行于一个对象结构的成员中。访问者模式让我们定义一个类的新操作而无须改变它操作的这些成员类。访问者模式的 UML 类模型如图 8-36 所示。

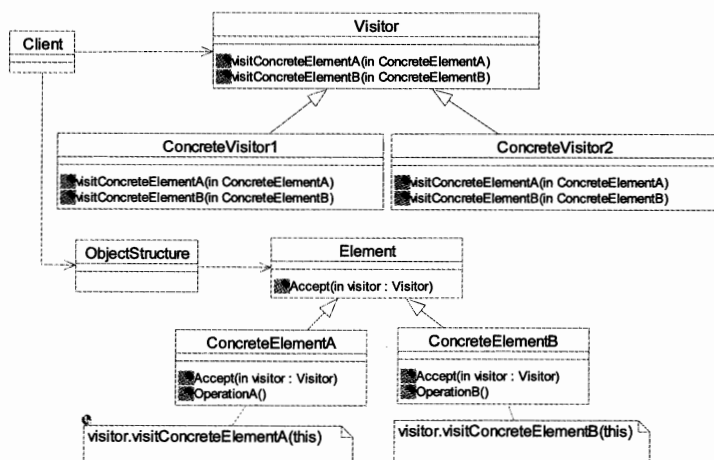


图 8-36 访问者模式的 UML 类模型

其中类和对象的关系如下。

- 抽象访问者(Visitor): 为对象结构类中的每一个 ConcreteElement 的类声明一个 Visit 操作, 这个操作的名称及标志(Signature)识别传出 Visit 请求给访问者的类。这就使得访问者可以找到被访问的元素的类, 也就可以直接经由其特有接口访问到元素(Element)。
- 具体访问者(ConcreteVisitor): 实现每个由 Visitor 声明的操作, 每个操作实现本算法的一部分, 而该算法片断乃是对应于结构中对象的类。ConcreteVisitor 为该算法提供了场景并存储它的局部状态。这一状态常常在遍历该结构的过程中累积结果。
- 元素(Element): 定义一个 Accept 操作, 它以一个访问者为参数。
- 具体元素(Concrete Element): 实现 Accept 操作, 该操作以访问者为参数。
- 对象结构(Object Structure)类: 能枚举它的元素; 可以提供一个高层的接口以允许访问者访问它的元素; 可以是一个合成模式或是一个集合, 如一个列表或一个无序集合。

22. 解释器模式

给出一种语言，定义这种语言的文法的一种表示，定义一个解释器（Interpreter），用它来解释使用这种语言的句子，这就是解释器模式，其 UML 类模型如图 8-37 所示。

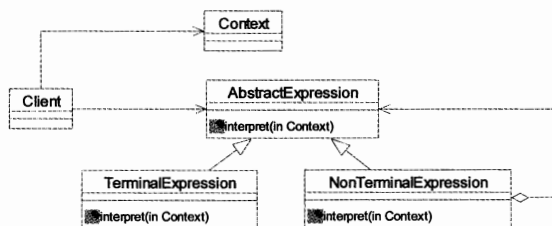


图 8-37 解释器模式的 UML 类模型

其中类和对象之间的关系如下。

- 抽象表达式（Abstract Expression）类：定义一个接口来执行解释操作。
- 终结符表达式（Terminal Expression）：实现文法中关联终结符的解释操作；文句中的每个终结符都需要一个实例。
- 非终结符表达式（NonTerminal Expression）：文法中的每一条规则 $R ::= R_1 R_2 \dots R_n$ 都需要一个非终结符表达式类；维护每一条规则 R_1 到 R_n 具有 AbstractExpression 接口实例；实现文法中关联非终结符的解释操作，采用递归的办法调用每一条规则 R_1 到 R_n 的解释操作。
- 上下文（Context）：包括解释器的所有全局信息。
- 客户应用程序（Client）：构建由这种语言表示的句子的抽象文法树，文法树由终结符表达式或者非终结符表达式的实例组成；调用文法树中的表达式实例的解释操作。

23. 调停者模式

调停者（Mediator）模式定义一个对象封装一系列多个对象如何相互作用。Mediator 使得对象之间不需要显式地相互引用，从而使其耦合更加松散。并且还让我们可以独立变化多个对象相互作用，其 UML 类模型如图 8-38 所示。

其中类和对象的关系如下。

- 抽象调停者（Mediator）：定义一个接口用于与各同事对象（Colleague）之间通信。
- 具体调停者（ConcreteMediator）：协调各个同事对象实现协作的行为；掌握并且维护它的各个同事对象引用。
- 同事（Colleague）类：每一个同事对象都引用一个调停者对象；每一个同事对象在需要和其他同事对象通信时，就与它的调停者通信。

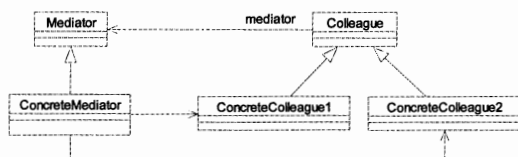


图 8-38 调停者模式的 UML 类模型

24. 观察者模式

观察者（Observer）模式定义对象间的一种一对多依赖关系，使得每当一个对象改变状态，则其相关依赖对象皆得到通知并被自动更新，其 UML 类模型如图 8-39 所示。

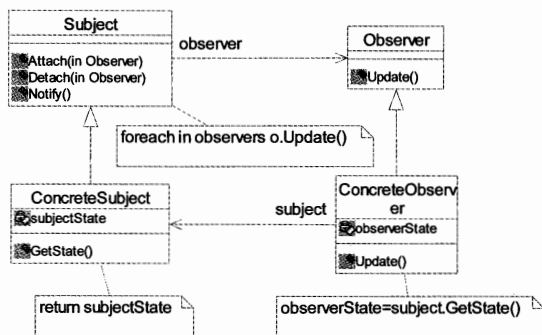


图 8-39 观察者模式的 UML 类模型

其中类和对象的关系如下。

- 被观察对象（Subject）：了解其多个观察者，任意数量的观察者可以观察一个对象；提供一个接口用来绑定及分离观察者对象。
- 具体被观察对象（Concrete Subject）：存储具体观察者（Concrete Observer）有兴趣的状态；当其状态改变时发送一个通知给其所有的观察者对象。
- 观察者（Observer）：定义一个更新接口，在一个被观察对象改变时应被通知。
- 具体观察者（Concrete Observer）：维护一个对 Concrete Subject 对象的引用；存储状态，状态需与 Concrete Subject 保持一致；实现观察者更新接口以保持其状态与 Concrete Subject 对象一致。

25. MVC 基础

MVC 是“Model-View-Controller”的缩写，中文翻译为“模式-视图-控制器”。它把一个应用的输入、处理、输出流程按照 Model、View、Controller 的方式进行分离，这样一个应用被分成 3 个层——模型层、视图层、控制层。

视图（View）代表用户交互界面，对于 Web 应用来说，可以概括为 HTML 界面，但有可能为 XHTML、XML 和 Applet。而一个应用可能有很多不同的视图，MVC 设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理，业务流程的处理交予模型（Model）处理。

模型（Model）就是业务流程/状态的处理，以及业务规则的制定。业务流程的处理过程对其他层来说是黑箱操作，模型接受视图请求的数据，并返回最终的处理结果。业务模型的设计可以说是 MVC 的核心。业务模型还有一个很重要的模型就是数据模型。数据模型主要指实体对象的数据保存（持续化）。

控制器（Controller）可以理解为从用户接收请求，将模型与视图匹配在一起，共同完成用户的请求。划分控制层的作用也很明显，它清楚地告诉我们，它就是一个分发器，选择什么样的模型，选择什么样的视图，可以完成什么样的用户请求。控制层

并不做任何数据处理。

模型、视图与控制器的分离,使得一个模型可以具有多个显示视图。如果用户通过某个视图的控制器改变了模型的数据,所有其他依赖于这些数据的视图都应反映到这些变化。因此,无论何时发生了何种数据变化,控制器都会将变化通知所有的视图,导致显示的更新。这实际上是一种模型的变化—传播机制。

8.2 典型试题分析

试题 1

服务组件体系结构 (Service Component Architecture, SCA) 是基于面向服务体系结构 (Service Oriented Architecture, SOA) 的思想描述服务之间组合和协作的规范。以下关于 SCA 的叙述,不正确的是 (1)。

- (1) A. SCA 定义了语言中立的服务组合方式,能够进行跨语言的服务调用
- B. SCA 加强组件的接口与传输协议的关联,提高组件的内聚性
- C. SCA 实现服务组件和其传输协议的绑定,这种绑定是可扩展的
- D. SCA 主要是为了满足软件集成的需要而创建的架构

试题分析

服务组件体系结构 (SCA) 是一个规范,它描述用于使用 SOA 构建应用程序和系统的模型。它可简化使用 SOA 进行的应用程序开发和实现工作。SCA 提供了构建粗粒度组件的机制,这些粗粒度组件由细粒度组件组装而成。SCA 将传统中间件编程从业务逻辑分离出来,从而使程序员免受其复杂性的困扰。它允许开发人员集中精力编写业务逻辑,而不必将大量的时间花费在更为底层的技术实现上。

SCA 方法的优势包括:简化业务组件开发;简化作为服务网络构建的业务解决方案的组装和部署;提高可移植性、可重用性和灵活性;通过屏蔽底层技术变更来保护业务逻辑资产;提高可测试性。

SCA 服务组件与传统组件的主要区别在于:服务组件往往是粗粒度的,而传统组件以细粒度居多;服务组件的接口是标准的,主要是 WSDL 接口,而传统组件常以具体 API 形式出现;服务组件的实现与语言是无关的,而传统组件常绑定某种特定的语言;服务组件可以通过组件容器提供 QoS 的服务,而传统组件完全由程序代码直接控制。

试题答案

- (1) B

试题 2

希赛公司欲实现一个数据处理软件,该软件需要从网络接收一组复杂的数据,然后分步进行解析和处理。在这种情况下,采用 (2) 的体系结构风格比较适合。

- (2) A. 远程过程调用 B. 层次化 C. 管道/过滤器 D. 共享数据

试题分析

层次系统组织成一个层次结构，每一层为上层服务，并作为下层客户。例如，四层的分层式体系结构可以分为应用软件、业务软件、中间件和系统软件。这种风格支持基于可增加抽象层的设计。这样，允许将一个复杂问题分解成一个增量步骤序列的实现。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，同样为软件重用提供了强大的支持。层次系统最广泛的应用是分层通信协议。在这一应用领域中，每一层提供一个抽象的功能，作为上层通信的基础。较低的层次定义低层的交互，最低层通常只定义硬件物理连接。

在管道/过滤器风格的软件体系结构中，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。这个过程通常通过对输入流的变换及增量计算来完成，所以在输入被完全消费之前，输出便产生了。因此，这里的构件被称为过滤器，这种风格的连接件就像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。此风格特别重要的过滤器必须是独立的实体，它不能与其他过滤器共享数据，而且一个过滤器不知道它上游和下游的标识。一个管道/过滤器网络输出的正确性并不依赖于过滤器进行增量计算过程的顺序。

根据以上介绍，在本题中，希赛公司欲实现一个数据处理软件，该软件需要从网络接收一组复杂的数据，然后分步进行解析和处理，适合采用的是管道/过滤器风格。

试题答案

(2) C

试题 3

回调(Call Back)函数是面向过程的程序设计语言中常用的一种机制，而设计模式中的___(3)___模式就是回调机制的一个面向对象的替代品。该模式的意图是___(4)___。

- (3) A. Strategy (策略) B. Adapter (适配器)
C. Command (命令) D. Observer (观察者)

- (4) A. 使原本由于接口不兼容而不能一起工作的那些类可以一起工作
B. 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
C. 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被自动更新
D. 使算法可独立于使用它的客户而变化

试题分析

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，对请求排队或记录请求日志，以及支持可撤销的操作。

Command 模式抽象出待执行的动作以参数化某对象，我们可用面向过程语言中的回调函数表达这种参数化机制。所谓回调函数，是指函数先在某处注册，而它将在稍

后某个需要的时候被调用。Command 模式是回调机制的一个面向对象的替代品。

Command 模式在不同的时刻指定、排列和执行请求。一个 Command 对象可以有一个与初始请求无关的生存期。如果一个请求的接收者可用一种与地址空间无关的方式表达,那么就可将负责该请求的命令对象传送给另一个不同的进程并在那里实现该请求。

Command 模式支持取消操作。Command 模式的 Excute 操作可在实施操作前将状态存储起来,在取消操作时这个状态用来消除该操作的影响。Command 接口必须添加一个 Unexecute 操作,该操作取消上一次 Execute 调用的效果。执行的命令被存储在一个历史列表中。可通过向后和向前遍历这一列表并分别调用 Unexecute 和 Execute 来实现重数不限的“取消”和“重做”。

Command 模式支持修改日志,这样当系统崩溃时,这些修改可以被重做一遍。在 Command 接口中添加装载操作和存储操作,可以用来保持变动的一个一致的修改日志。从崩溃中恢复的过程包括从磁盘中重新读入记录下来的命令并用 Execute 操作重新执行它们。

Command 模式用构建在原语操作上的高层操作构造一个系统。这样一种结构在支持事务的信息系统中很常见。一个事务封装了对数据的一组变动。Command 模式提供了对事务进行建模的方法。Command 模式有一个公共的接口,可以用同一种方式调用所有的事务。同时,使用该模式也易于添加新事务以扩展系统。

试题答案

(3) C

(4) B

试题 4

如图 8-40 (a) 所示的 UML 类图描绘的是设计模式中的 (5) 模式。如图 8-40 (b) 所示的 UML 类图描述了该模式的一种应用,其中与图 8-40 (a) 中的“Creator”对应的类是 (6)。

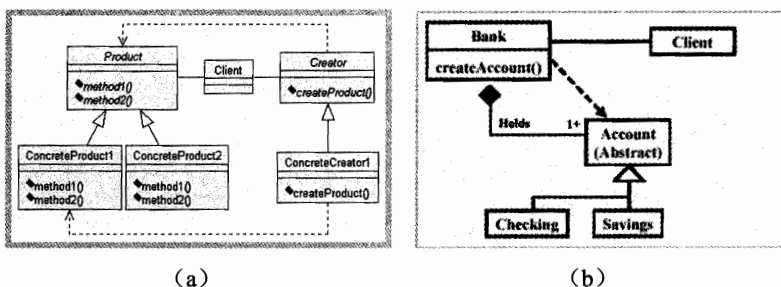


图 8-40 设计模式与类图

(5) A. Command B. Factory Method C. Composite D. Class Adapter

(6) A. Bank B. Account C. Checking D. Savings

试题分析

Command 模式将一个请求封装为一个对象,从而将不同的请求对数化并进行排队

或登记，以支持撤销操作。

Factory Method 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。

Composite 模式将对象组成树结构来表示局部和整体的层次关系。客户可以统一处理单个对象和对象组合。

Class Adapter 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。

试题答案

(5) B

(6) A

试题 5

设计模式 (Design Pattern) 是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。下面关于设计模式所倡导的基本原则的描述，错误的是 (7)。

- (7) A. 模块应对扩展开放，而对修改关闭
B. 优先使用继承，而不是组合
C. 要针对接口编程，而不是针对实现编程
D. 抽象不应该依赖于细节，细节应当依赖于抽象

试题分析

设计模式所倡导的基本原则如下。

① 开闭原则：一个软件实体应当对扩展开放，对修改关闭。在设计一个模块时，应当使这个模块可以在不被修改的情况下被扩展。关键在于抽象，抽象层要预见所有可能的扩展，因此，抽象层在任何扩展情况下都不会改变，即对修改关闭。同时，由于从抽象层导出一个或多个新类，可以有不同的实现，改变系统的的行为，此即为对扩展开发。简而言之，抽象层对修改关闭，通过扩展实现改变系统行为。

② 里氏代换原则：任何基类可以出现的地方，子类一定可以出现。

③ 依赖原则：要依赖于抽象，而不是具体实现。也可以这样说，要针对接口编程，不要针对实现编程。

④ 接口分离原则：应当为客户端提供尽量小的单独的接口，而不是提供大的接口。

⑤ 组合复用原则：要尽量使用组合，而不是继承关系达到复用目的。

⑥ 迪米特法则：又叫最少知识法则，就是说一个对象应当对其他对象有尽可能少的了解。

试题答案

(7) B

试题 6

在企业应用系统开发中，方法调用 (Method Invocation) 和消息 (Messaging) 机制是两种常用的数据处理与交换方式，下面关于这两种机制的描述，不正确的是

(8) 。

- (8) A. 方法调用一般具有同步特性, 而消息机制具有异步的特点
- B. 从可靠性方面考虑, 消息机制比方法调用更有优势
- C. 从效率方面考虑, 一般情况下消息机制比方法调用更有优势
- D. 消息调用机制可以支持多个数据的发送者和接收者, 更加灵活

试题分析

方法调用是指当某个事件发生时, 调用某个对象的方法来处理事件。消息机制是指当某个事件发生时, 把相关的消息发送给某些对象(一个或多个), 相关对象捕获消息后, 进行相应的处理。

从以上定义可以看出, 方法调用一般具有同步特性, 而消息机制具有异步的特点。在消息调用机制中, 消息是通过消息总线进行发送的, 可以支持多个数据的发送者和接收者, 更加灵活。

从效率方面考虑, 由于方法调用指定调用某个对象的方法, 同步进行, 所以效率要高一些。而消息机制由于采用异步方法, 数据不一定得到很快的处理, 具体要根据接收消息的对象的处理情况, 所以一般情况下效率会低一些。但是, 从可靠性方面考虑, 由于消息机制是通过系统总线来完成, 且可能有多个接收者, 所以比方法调用更有优势。

试题答案

(8) C

试题 7

JavaEE 是 Java 领域内企业级应用开发的框架与标准。下面关于采用 JavaEE 架构的描述, 不正确的是 (9) 。

- (9) A. JavaEE 定义了分布式环境中多层应用系统的架构, 是多种 Java 技术的混合体
- B. 具有典型的 3 层结构: 表现层、业务逻辑层和基础设施层
- C. 不同的应用系统对底层支持系统的要求可能不同, 因此每次开发时应该针对不同的应用需求对底层系统进行二次开发, 提供支持接口
- D. 要严格区分业务逻辑层和表现层, 尤其应该注意不要在表现层中混杂业务代码

试题分析

JavaEE 以前称为 J2EE, 可以帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。JavaEE 是在 JavaSE 的基础上构建的, 它提供 Web 服务、组件模型、管理和通信 API, 可以用来实现企业级的面向服务体系结构(SOA)和 Web 2.0 应用程序。JavaEE 定义了分布式环境中多层应用系统的架构, 是多种 Java 技术的混合体, 具有典型的 3 层结构: 表现层、业务逻辑层和基础设施层。在实际应用中, 基础设施层应该保持稳定, 要严格区分业务逻辑层和表现层, 尤其应该注意不要在

表现层中混杂业务代码。

试题答案

(9) C

试题 8

当不适合采用生成子类的方法对已有的类进行扩充时, 可以采用__ (10) __设计模式动态地给一个对象添加一些额外的职责; 当应用程序由于使用大量的对象, 造成很大的存储开销时, 可以采用__ (11) __设计模式运用共享技术来有效地支持大量细粒度的对象; 当想使用一个已经存在的类, 但其接口不符合需求时, 可以采用__ (12) __设计模式将该类的接口转换成我们希望的接口。

- | | |
|----------------------|-------------------|
| (10) A. 命令 (Command) | B. 适配器 (Adapter) |
| C. 装饰 (Decorate) | D. 享元 (Flyweight) |
| (11) A. 命令 (Command) | B. 适配器 (Adapter) |
| C. 装饰 (Decorate) | D. 享元 (Flyweight) |
| (12) A. 命令 (Command) | B. 适配器 (Adapter) |
| C. 装饰 (Decorate) | D. 享元 (Flyweight) |

试题分析

本题考查面向对象设计。

装饰模式主要的目的是在无法生成子类的情况下给一个对象动态地增加新的职责; 享元设计模式是共享大量细粒度的对象; 适配器设计模式则是将已有的接口转换为系统希望的接口形式。

试题答案

(10) C (11) D (12) B

试题 9

在 CORBA 体系结构中, __ (13) __ 属于客户端接口。

- | | |
|--------------------------|-----------------|
| (13) A. 静态 IDL Skeletons | B. POA |
| C. 静态 IDL Stubs | D. 动态 Skeletons |

试题分析

在 CORBA 体系结构中, ORB (Object Request Broker, 对象请求代理) 负责处理底层网络细节, 它可以运行在各种不同的底层网络协议上, 如 TCP/IP、IPX 和 SS7 等。在此基础上, ORB 实现了一系列的功能, 如对象定位、编组与解组、初始化服务和接口库等。它为客户提供和服务器端提供标准 API, 使得客户不用考虑底层网络细节, 通过对象引用来实现对远程对象的请求调用。

IDL (Interface Definition Language, 接口定义语言) 定义客户机和服务器之间的静态接口, 通过它实现了对象接口与对象实现的分离, 屏蔽了语言和系统软件带来的异构件。通过标准的 IDL 编译器, 可生成客户机端的 IDL 存根 (Stubs) 和服务端

的骨架 (Skeletons)，这两者就如同客户机端程序和服务器端程序连接 ORB 的粘合剂，IDL 存根提供了访问对象服务的静态接口，而骨架则包含了服务对象的静态接口并负责实现与对象实现中具体方法的连接。

IDL 存根被称为静态调用接口，由 IDL 编译器编译目标对象的 IDL 接口描述文件而自动产生，客户程序与它直接相连。IDL 存根的作用相当于本地调用，由存根向 ORB 透明地提供一个接口，以实现对操作参数的编码和解释。IDL 存根把请求从特定的编程语言的表示形式转换为适于传递到目标对象的形式进行通信传输。存根为客户提供了一种机制，使得客户能够不关心 ORB 的存在，而把请求交给存根，由存根负责对请求参数的封装和发送，以及对返回结果的接收和解封装。

静态 IDL 骨架是静态 IDL 存根在服务器端的对应，在请求的接收端提供与存根类似的服务。当 ORB 接收到请求时，由骨架将请求参数解封装，识别客户所请求的服务，（向上）调用服务器中的对象实现，当服务器完成了对请求的处理后，骨架把执行结果封装，并将结果返回给客户程序。

由于存根和骨架都是从用户的接口定义编译而来，所以它们都和具体的接口有关，并且，在请求发生前，存根和骨架早已分别被直接连接到客户程序和对象实现中去。为此，通过存根和骨架的调用被通称为静态调用。IDL 存根和 IDL 骨架之间没有必须配对的限制。

动态骨架接口 (Dynamic Skeleton Interface, DSI) 允许动态调用对象，对象实现需要实现动态调用例程的接口。DSI 是 DII (IDL 动态调用接口) 在服务器方的对应。与 DII 允许客户不通过存根就可以调用请求类似，DSI 允许用户在没有静态骨架信息的条件下来获得对象实现。DSI 从进入的消息找出调用的目标对象及相应的方法，并提供运行时的连接机制。

POA (Portable Object Adapter, 可携带对象适配器) 是一个引导客户端的请求到具体的对象应用的机制。POA 提供了标准的 API 去登记对象应用，或激活对象应用。POA 是灵活的 CORBA 编程模型模块，并且提供了大量的规则配置它的行为。

试题答案

(13) C

试题 10

某软件公司欲开发一个 Windows 平台上的公告板系统。在明确用户需求后，该公司的架构师决定采用 Command 模式实现该系统的界面显示部分，并设计 UML 类图，如图 8-41 所示。图中与 Command 模式中的“Invoker”角色相对应的类是 (14)，与“ConcreteCommand”角色相对应的类是 (15)。

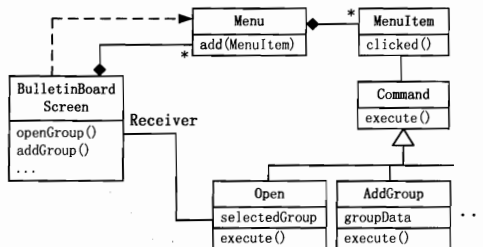


图 8-41 UML 类图

(14) A. Command

B. MenuItem

C. Open

D. ButktinBoardScreen

(15) A. Command

B. MenuItem

C. Open

D. BulktinBoardScreen

试题分析

Command（命令）模式是设计模式中行为模式的一种，它将“请求”封装成对象，以便使用不同的请求、队列或者日志来参数化其他对象。Command 模式也支持可撤销的操作。Command 模式的类图如图 8-42 所示。

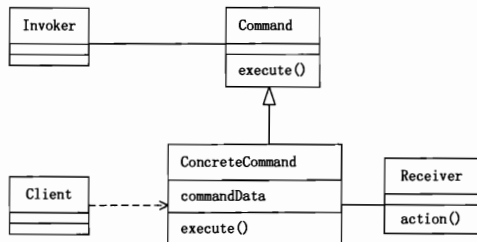


图 8-42 Command 类图

对于题目所给出的图，与“Invoker”角色相对应的类是 MenuItem，与“Concrete Command”角色相对应的类是 Open。

试题答案

(14) B

(15) C

试题 11

系统应用架构设计中，网络架构数据流图的主要作用是将处理器和设备分配到网络中。____(16)____不属于网络架构数据流图的内容。

(16) A. 服务器、客户端及其物理位置

B. 处理器说明信息

C. 单位时间的数据流大小

D. 传输协议

试题分析

应用架构建模中要绘制的第一个物理数据流图（PDFD）是网络架构 DFD，它们不显示单位时间的数据流量，需要显示的信息包括服务器及其物理位置、客户端及其物理位置、处理器说明、传输协议。

试题答案

(16) C

试题 12

软件架构贯穿于软件的整个生命周期，但在不同阶段对软件架构的关注力度并不相同，在____(17)____阶段，对软件架构的关注最多。

- (17) A. 需求分析与设计 B. 设计与实现
C. 实现与测试 D. 部署与变更

试题分析

本题主要考查软件架构对软件开发的影响和在生命周期中的关注力度。

软件架构贯穿于软件的整个生命周期，但在不同的阶段对软件架构的关注力度并不相同。其中需求分析阶段主要关注问题域；设计阶段主要将需求转换为软件架构模型；软件实现阶段主要关注将架构设计转换为实际的代码；软件部署阶段主要通过组装软件组件提高系统的实现效率。其中，设计与实现阶段在软件架构上的工作最多，也最重要，因此关注力度最大。

试题答案

- (17) B

试题 13

软件架构设计是降低成本、改进质量、按时和按需交付产品的关键活动。以下关于软件架构重要性的叙述中，错误的是__ (18) __。

- (18) A. 架构设计能够满足系统的性能、可维护性等品质
B. 良好的架构设计能够更好地捕获并了解用户需求
C. 架构设计能够使得不同的利益相关人(Stakeholders)达成一致的目标
D. 架构设计能够支持项目计划和项目管理等活动

试题分析

软件架构设计是降低成本、改进质量、按时和按需交付产品的关键因素。架构设计能够满足系统的性能、可维护性等品质；能够使得不同的利益相关人(Stakeholders)达成一致的目标；能够支持项目计划和项目管理等活动；能够有效地管理复杂性，等等。然而系统架构的给出必须建立在需求明确的基础上。

试题答案

- (18) B

试题 14

软件架构需求是指用户对目标软件系统在功能、行为、性能、设计约束等方面的期望。以下活动中，不属于软件架构需求过程范畴的是__ (19) __。

- (19) A. 设计构件 B. 需求获取
C. 标识构件 D. 架构需求评审

试题分析

软件架构需求是指用户对目标软件系统在功能、行为、性能和设计约束等方面的期望。需求过程主要是获取用户需求，标识系统中所要用到的构件，并进行架构需求

评审。其中标识构件又详细分为生成类图、对类图进行分组和将类打包成构件 3 步。软件架构需求并不应该包括设计构件的过程。

试题答案

(19) A

试题 15

基于架构的软件设计(ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。以下关于 ABSD 的叙述中,错误的是 (20)。

- (20) A. 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
- B. ABSD 方法是一个自顶向下,递归细化的过程
- C. ABSD 方法有 3 个基础:功能分解、选择架构风格实现质量和商业需求及软件模板的使用
- D. 使用 ABSD 方法,设计活动的开始意味着需求抽取和分析活动可以终止

试题分析

基于架构的软件设计(ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。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,并且设计活动的开始并不意味着需求抽取和分析活动可以终止,而是应该与设计活动并行。ABSD 方法有 3 个基础:第一个基础是功能分解,在功能分解中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体系结构风格来实现质量和商业需求;第三个基础是软件模板的使用。ABSD 方法是一个自顶向下,递归细化的过程,软件系统的架构通过该方法得到细化,直到能产生软件构件的类。

试题答案

(20) D

试题 16

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 (21)。

- (21) 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

试题分析

软件架构文档是对软件架构的一种描述,帮助程序员使用特定的程序设计语言实现软件架构。软件架构文档的写作应该遵循一定的原则,这些原则包括:文档要从使用者的角度进行编写;必须分发给所有与系统有关的开发人员;应该保持架构文档的即时更新,但更新不要过于频繁;架构文档中描述应该尽量避免不必要的重复;每次

架构文档修改都应该记录进行修改的原则。

试题答案

(21) A

试题 17

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 (22)。

- (22) 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

试题分析

架构复审是基于架构开发中一个重要的环节。架构设计、文档化和复审是一个迭代的过程。从这个方面来说, 在一个主版本的软件架构分析之后, 要安排一次由外部人员(用户代表和领域专家)参加的复审。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。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。

试题答案

(22) C

试题 18

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 (23) 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 (24) 风格的特点。

- (23) A. 虚拟机
 - B. 管道-过滤器
 - C. 事件驱动
 - D. 微内核-扩展
- (24) A. 虚拟机
 - B. 管道-过滤器
 - C. 事件驱动
 - D. 微内核-扩展

试题分析

Windows 操作系统在图形用户界面处理方面采用的是典型的“事件驱动”的架构风格。首先注册事件处理的是回调函数, 当某个界面事件发生时(如键盘敲击、鼠标移动等), 系统会查找并选择合适的回调函数处理该事件。Java 语言是一种解释型语言, 在 Java 虚拟机上运行, 这从架构风格上看是典型的“虚拟机”风格, 即通过虚拟架构屏蔽不同的硬件环境。

试题答案

(23) C

(24) A

试题 19

某软件开发公司负责开发一个 Web 服务器服务端处理软件,其核心部分是对客户端请求消息的解析与处理,包括 HTTP 报头分离、SOAP 报文解析等功能。该公司的架构师决定采用成熟的架构风格指导整个软件的设计,以下 (25) 架构风格,最适合该服务端处理软件。

- 、(25) A.虚拟机 B.管道-过滤器 C.黑板结构 D.分层结构

试题分析

根据题干描述,Web 服务器服务端的核心功能是数据处理,由于 Web 服务在数据传输方面具有协议分层的特征,即底层协议会包装上层协议(HTTP 协议体中包含整个 SOAP 消息内容),因此需要数据内容的逐步分解与分阶段处理。比较选项中的架构风格,由于管道-过滤器的架构风格支持分阶段数据处理,因此特别适合该服务端处理软件的要求。

试题答案

- (25) B

试题 20

希赛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,编辑程序可以自动滚屏到断点,变量监视器刷新变量数值。针对这样的功能描述,采用 (26) 的架构风格最为合适。

- (26) A.数据共享 B.虚拟机 C.隐式调用 D.显式调用

试题分析

根据题干描述,调试器在设置端点时,其本质是在断点处设置一个事件监听函数,当程序执行到断点位置时,会触发并调用该事件监听函数,监听函数负责进行自动滚屏、刷新变量数值等动作。这是一个典型的回调机制,属于隐式调用的架构风格。

试题答案

- (26) C

试题 21

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 (27) 。

- (27) 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

试题分析

采用闭环结构的软件通常由几个协作构件共同构成，且其中的主要构件彼此分开，能够进行替换与重用，但闭环结构通常适用于处理简单任务（如机器装配等），并不适用于复杂任务。分层结构的特点是通过引入抽象层，在较低层次不确定的实现细节在较高层次会变得确定，并能够组织层间构件的协作，系统结构更加清晰。

试题答案

(27) A

试题 22

一个软件的架构设计是随着技术的不断进步而不断变化的。以编译器为例，其主流架构经历了管道-过滤器到数据共享为中心的转变过程。以下关于编译器架构的叙述中，错误的是（28）。

- (28) A. 早期的编译器采用管道-过滤器架构风格，以文本形式输入的代码被逐步转化为各种形式，最终生成可执行代码
- B. 早期的编译器采用管道-过滤器架构风格，并且大多数编译器在词法分析时创造独立的符号表，在其后的阶段会不断修改符号表，因此符号表并不是程序数据的一部分
- C. 现代的编译器采用以数据共享为中心的架构风格，主要关心编译过程中程序的中间表示
- D. 现代的编译器采用以数据共享为中心的架构风格，但由于分析树是在语法分析阶段结束后才产生作为语义分析的输入，因此分析树不是数据中心的共享数据

试题分析

一个软件的架构设计是随着技术的进步而不断变化的。以编译器为例，其主流架构经历了管道-过滤器到数据共享为中心的转变过程。早期的编译器采用管道-过滤器架构风格，以文本形式输入的代码被逐步转化为各种形式，最终生成可执行代码。早期的编译器采用管道-过滤器架构风格，并且大多数编译器在词法分析时创造独立的符号表，在其后的阶段会不断修改符号表，因此符号表并不是程序数据的一部分。现代的编译器采用以数据共享为中心的架构风格，主要关心编译过程中程序的中间表示。现代的编译器采用以数据共享为中心的架构风格，分析树是在语法分析阶段结束后才产生作为语义分析的输入，分析树是数据中心的共享数据，为后续的语义分析提供了帮助。

试题答案

(28) D

试题 23

____(29)____的选择是开发一个软件系统时的基本设计决策；____(30)____是最底层的模式，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-计数是 C++ 管理动态资源时常用的一种____(31)____。

- (29) A. 架构模式 B. 惯用法 C. 设计模式 D. 分析模式
(30) A. 架构模式 B. 惯用法 C. 设计模式 D. 分析模式
(31) A. 架构模式 B. 惯用法 C. 设计模式 D. 分析模式

试题分析

模式是给定上下文中普遍问题的普遍解决方案，在架构方面，分高层的模式与低层的模式。

架构模式是软件设计中的高层决策，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；惯用法则是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，如引用-计数就是 C++ 语言中的一种惯用法。

试题答案

- (29) A (30) B (31) B

试题 24

某软件公司基于面向对象技术开发了一套图形界面显示构件库 Visual Component。在使用该库构建某图形界面时，用户要求为界面定制一些特效显示效果，如带滚动条、能够显示艺术字体的透明窗体等。针对这种需求，公司采用____(32)____最为灵活。

- (32) A. 桥接模式 B. 命令模式 C. 组合模式 D. 装饰模式

试题分析

本题考查设计模式的应用。每一种设计模式，都有自身的一些优势，以应对某些特定的应用场合。根据题目的描述，需要开发的是图形界面构件库，并要求为图形界面提供一些定制的特效，如带滚动条的图形界面，能够显示艺术字体且透明的图形界面等。这要求能够动态地对一个对象进行功能上的扩展，也可以对其子类进行功能上的扩展。对照选项中的 4 种设计模式，装饰模式最符合这一要求。

试题答案

- (32) D

试题 25

某软件公司承接了为某工作流语言开发解释器的工作。该工作流语言由多种活动节点构成，具有类 XML 的语法结构。用户要求解释器工作时，对每个活动节点进行一系列的处理，包括执行活动、日志记录、调用外部应用程序等，并且要求处理过程具有可扩展能力。针对这种需求，公司采用____(33)____最为恰当。

- (33) A. 适配器模式 B. 迭代器模式 C. 访问者模式 D. 观察者模式

试题分析

根据题目的描述,可以看出本题的核心在于对某个具有固定结构的活动节点需要多种处理能力,且处理能力可扩展,也就是说要求在不改变原来类结构(活动节点)的基础上增加新功能。对照4个选项,发现访问者模式最符合要求。

试题答案

(33) C

试题 26

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 (34)。

- (34) 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

试题分析

本题考查软件体系结构中的评估方法。

ATAM 是软件体系结构评估中的一种方法,主要对软件体系结构的设计结果进行评估。评估是软件系统详细设计、实现和测试之前的阶段工作,因此评估不涉及系统的实现代码和测试,因为评估是考查软件体系结构是否能够合适地解决软件系统的需求,并不对软件需求自身是否准确进行核实,而软件需求是否准确是需求评审阶段的工作。ATAM 并不是一种精确的评估方法,该方法表现的主要形式是评审会议。

试题答案

(34) D

试题 27

识别风险点、非风险点、敏感点和权衡点是 ATAM 方法中的关键步骤。已知针对某系统所做的架构设计中,提高其加密子系统的加密级别将对系统的安全性和性能都产生非常大的影响,则该子系统一定属于 (35)。

- (35) A. 风险点和敏感点 B. 权衡点和风险点
C. 权衡点和敏感点 D. 风险点和非风险点

试题分析

本题考查软件体系结构中的评估方法。

加密子系统的加密级别会对安全性和性能产生影响,一般而言,加密程度越高,安全性越好,但是其性能会降低;而加密程度越低,安全性越差,但性能一般会提高。因此,该子系统将在安全性和性能两个方面产生冲突,所以该子系统一定属于权衡点和敏感点。

试题答案

(35) C

8.3 实战练习题

● 如图 8-43 (a) 所示的 UML 类图描绘的是设计模式中的__ (1) __ 模式。如图 8-43 (b) 所示的 UML 类图描述了该模式的一种应用，其中与图 8-43 (a) 中的“Abstraction”对应的类是__ (2) __。

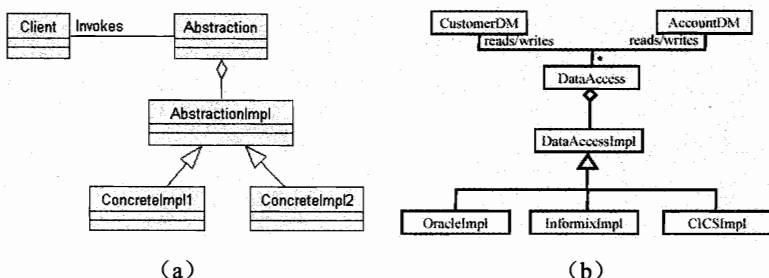


图 8-43 设计模式与类图

- (1) A. Command B. Bridge C. Composite D. Proxy
- (2) A. DataAccess B. DataAccessImpl C. CustomerDM D. AccountDM
- 软件的分层式体系结构把软件系统划分为 4 层，这 4 层结构自顶向下分别是__ (3) __。
- (3) A. 应用软件 B. 业务构件 C. 应用软件 D. 业务构件
- | | | | |
|------|------|------|------|
| 业务构件 | 应用软件 | 中间件 | 中间件 |
| 中间件 | 中间件 | 系统软件 | 应用软件 |
| 系统软件 | 系统软件 | 业务构件 | 系统软件 |
- 结构模板能够帮助分析员建立一个逐层细化的层次结构。结构环境图 (Architecture Context Diagram, ACD) 位于层次结构的顶层。在从 ACD 导出的__ (4) __中给出了各个专门子系统和重要 (数据与控制) 信息流。
- (4) A. 系统语境图 (SCD) B. 结构互连图 (AID)
- C. 结构流程图 (AFD) D. 结构图的规格说明 (ADS)
- 管道和过滤器模式属于__ (5) __。
- (5) A. 数据为中心的体系结构 B. 数据流体系结构
- C. 调用和返回体系结构 D. 层次式体系结构
- 某银行系统采用 Factory Method 方法描述其不同账户之间的关系，设计出的类图如图 8-44 所示。其中与 Factory Method 中的“Creator”角色相对应的类是__ (6) __；与“Product”角色相对应的类是__ (7) __。

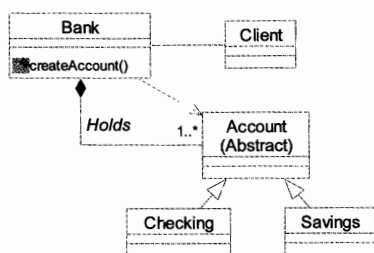


图 8-44 工厂方法模式类图

- (6) A. Bank B. Account C. Checking D. Savings
- (7) A. Bank B. Account C. Checking D. Savings

● 在客户机/服务器系统开发中, 采用__ (8) __时, 应将数据层和数据处理层放置于服务器, 应用逻辑层、表示逻辑层和表示层放置于客户机。

- (8) A. 分布式表示结构 B. 分布式应用结构
- C. 分布式数据和应用结构 D. 分布式数据结构

● 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__ (9) __。

- (9) 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人员和实现人员的工作

● 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__ (10) __。

- (10) 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

● 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__ (11) __来描述软件架构, 采用__ (12) __来描述需求。

- (11) 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
- (12) 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

● 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(13)架构风格最为合适。

- (13) A. 管道-过滤器 B. 隐式调用 C. 主程序-子程序 D. 解释器

● 希赛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图 8-45 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(14)。

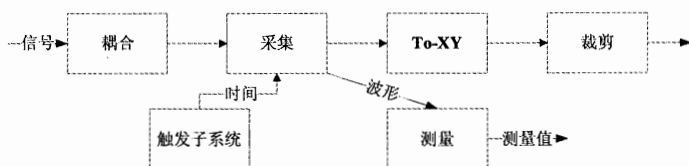


图 8-45 示波器内置软件架构图

- (14) 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

● 希赛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(15)架构风格最为合适。

- (15) A. 解释器 B. 过程控制 C. 分层 D. 管道-过滤器

● 在客户机/服务器系统开发中,采用(16)时,应将数据层和数据处理层放置于服务器,应用逻辑层、表示逻辑层和表示层放置于客户机。

- (16) A. 分布式表示结构 B. 分布式应用结构
C. 分布式数据和应用结构 D. 分布式数据结构

● 希赛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(17)架构风格最为合适。

- (17) A. 解释器 B. 主程序-子程序 C. 隐式调用 D. 管道-过滤器

● 希赛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(18)架构风格最为合适。

- (18) 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(DSSA)是在一个特定应用领域为一组应用提供组织结构参考的标准软件架构。实施 DSSA 的过程中包括一系列基本的活动,其中(19)活

动的主要目的是为了获得 DSSA。该活动参加人员中, (20) 的主要任务是提供关于领域中系统的需求规约和实现的知识。

(19) A. 领域需求 B. 领域分析 C. 领域设计 D. 领域实现

(20) A. 领域专家 B. 领域分析者 C. 领域设计者 D. 领域实现者

● 某服务器软件系统对可用性 (Availability)、性能 (Performance) 和可修改性 (Modification) 的要求较高, (21) 设计策略能提高该系统的可用性, (22) 设计策略能够提高该系统的性能, (23) 设计策略能够提高该系统的可修改性。

(21) A. Ping/Echo B. 限制访问 C. 运行时注册 D. 接口-实现分离

(22) A. 分层结构 B. 事务机制 C. 主动冗余 D. 队列调度

(23) A. 信息隐藏 B. 记录/回放 C. 任务模型 D. 回滚

● 某服务器软件系统能够正确运行并得出计算结果, 但存在“系统出错后不能在要求的时间内恢复到正常状态”和“对系统进行二次开发时总要超过半年的时间”两个问题, 上述问题依次与质量属性中的 (24) 相关。

(24) A. 可用性和性能 B. 性能和可修改性

C. 性能和可测试性 D. 可用性和可修改性

● 希赛公司欲开发一个软件系统的在线文档帮助系统, 用户可以在任何一个查询上下文中输入查询关键字, 如果当前查询环境下没有相关内容, 则系统会将查询按照一定的顺序转发给其他查询环境。基于上述需求, 采用 (25) 最为合适。

(25) A. 责任链模式 B. 桥接模式 C. 装饰模式 D. 适配器模式

● 希赛公司欲开发一套窗体图形界面类库。该类库需要包含若干预定义的窗格 (Pane) 对象, 例如 TextPane、ListPane 等, 窗格之间不允许直接引用。基于该类库的应用由一个包含一组窗格的窗口组成, 并需要协调窗格之间的行为。基于该类库, 在不引用窗格的前提下实现窗格之间的协作, 应用开发者应采用 (26) 最为合适。

(26) A. 备忘录模式 B. 中介者模式 C. 访问者模式 D. 迭代器模式

● 希赛公司开发一个文档编辑器, 该编辑器允许在文档中直接嵌入图形对象, 但开销很大。用户在系统设计之初提出编辑器在打开文档时必须十分迅速, 可以暂时不显示当前页面以外的图形。针对这种需求, 公司可以采用 (27) 避免同时创建这些图形对象。

(27) A. 代理模式 B. 外观模式 C. 桥接模式 D. 组合模式

● 正确识别风险点、非风险点、敏感点和权衡点是进行软件架构评价的关键步骤。其中 (28) 是实现一个特定质量属性的关键特征, 该特征为一个或多个软件构件所共有。“改变加密的级别可能会对安全性和性能都产生显著的影响”, 这是一个对系统 (29) 的描述。

(28) A. 风险点 B. 非风险点 C. 敏感点 D. 权衡点

(29) A. 风险点 B. 非风险点 C. 敏感点 D. 权衡点

结构模板把子系统处理划分成5个处理区域。每个子系统可以包含一个或多个系统元素（如硬件、软件、人），它们是系统工程师分配给子系统的。

试题4答案

(4) C

试题5分析

体系结构风格有九大类：

- ① 数据流系统：包括顺序批处理、管道和过滤器。
- ② 调用和返回系统：包括主程序和子程序、面向对象系统、层次结构。
- ③ 独立部件：包括通信进程、事件隐式调用。
- ④ 虚拟机：包括解释器、规则基系统。
- ⑤ 以数据为中心的系统：包括数据库、超文本系统、黑板系统。
- ⑥ 特殊领域风格：如过程控制、模拟器。
- ⑦ 特殊结构的风格：如分布式处理、状态转移系统。
- ⑧ 不同风格合成建立的异构结构。
- ⑨ 最初始、最基本的主程序/子程序。

管道和过滤器模式是为处理数据流的系统提供了一种模式。它是由过滤器和管道组成的。每个处理步骤都被封装在一个过滤器组件中，数据通过相邻过滤器之间的管道进行传输。每个过滤器可以单独修改，功能单一，并且它们之间的顺序可以进行配置。

试题5答案

(5) B

试题6~7分析

Factory Method 模式的意图是，定义一个用于创建对象的接口，让子类决定实例化哪一个类。Factory Method 是一个类的实例化延迟到其子类。Factory Method 模式的类图如图 8-46 所示。

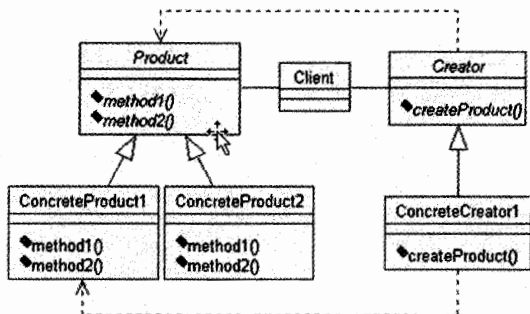


图 8-46 工厂方法模式类图

其中,

- 类 Product 定义了 Factory Method 所创建的对象接口。
- 类 ConcreteProduct 用于实现 Product 接口。
- 类 Creator 声明了工厂方法,该方法返回一个 Product 类型的对象。Creator 也可以定义一个工厂方法的默认实现,它返回一个默认的 ConcreteProduct 对象。
- 类 ConcreteCreator 重定义了工厂方法,以返回一个 ConcreteProduct 实例。

对照两张类图可以看出,与“Creator”角色相对应的类是 Bank;与“Product”角色相对应的类是 Account。

试题 6~7 答案

(6) A

(7) B

试题 8 分析

客户机/服务器系统开发时可以采用不同的分布式计算架构:分布式表示架构是将表示层和表示逻辑层迁移到客户机,应用逻辑层、数据处理层和数据层仍保留在服务器上;分布式数据架构是将数据层和数据处理层放置于服务器,应用逻辑层、表示逻辑层和表示层放置于客户机;分布式数据和应用架构数据层和数据处理层放置在数据服务器上,应用逻辑层放置于应用服务器上,表示逻辑层和表示层放置于客户机上。

试题 8 答案

(8) D

试题 9 分析

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,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,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,软件架构能够指导设计人员和实现人员的工作。一般在设计软件架构之初,会根据用户需求,确定多个候选架构,并从中选择一个较优的架构,并随着软件的开发,对这个架构进行微调,以达到最佳效果。

试题 9 答案

(9) A

试题 10 分析

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,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,并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,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。一般来说,软件架构设计活动将已标识构件集成到软件架构中,设计这些构件,但不予以实现。

试题 10 答案

(10) C

试题 11~12 分析

根据基于软件架构的设计的定义，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。它强调采用视角和视图来描述软件架构，采用用例和质量属性场景来描述需求。

试题 11~12 答案

(11) B

(12) C

试题 13 分析

本题主要考查软件架构设计策略与架构风格问题。根据题目描述，该软件系统特别强调用户定义系统中对象的关系和行为这一特性，这需要在软件架构层面提供一种运行时的系统行为定义与改变的能力，根据常见架构风格的特点和适用环境，可以知道最合适的架构设计风格应该是解释器风格。

试题 13 答案

(13) D

试题 14 分析

本题主要考查架构评审和软件架构设计的应用。根据图 8-45 中示波器的功能描述，结合示波器常见的功能和使用方式，可以看出图中仅有示波器内部处理的建模，而没有与外界的交互，如用户交互。而用户交互是示波器重要的功能，不能缺失，所以这是该架构存在的最大问题。

试题 14 答案

(14) C

试题 15 分析

本题主要考查架构风格与架构设计策略。根据题目描述，调温器需要实时获取外界的温度信息，并与用户定义的温度进行比较并做出动作。根据该系统的应用领域和实际需求，可以看出这是一个典型的过程控制架构风格的应用场景。

试题 15 答案

(15) B

试题 16 分析

客户机/服务器系统开发时可以采用不同的分布式计算架构：分布式表示架构是将表示层和表示逻辑层迁移到客户机，应用逻辑层、数据处理层和数据层仍保留在服务器上；分布式数据架构是将数据层和数据处理层放置于服务器，应用逻辑层、表示逻辑层和表示层放置于客户机；分布式数据和应用架构数据层和数据处理层放置在数据服务器上，应用逻辑层放置于应用服务器上，表示逻辑层和表示层放置于客户机。

试题 16 答案

(16) D

试题 17 分析

本题主要考查架构风格与架构设计策略。根据题目描述，漫步者机器人需要根据自身状态的外界环境进行自动调整，这是一个典型的根据外部事件进行响应的场景。比较 4 个候选项，隐式调用比较适合根据外部事件进行处理和动作的情景。

试题 17 答案

(17) C

试题 18 分析

本题主要考查架构风格与架构设计策略。根据题目描述，语音识别系统是一个十分典型的专家系统，其特点是求解的正确结果不止一个，求解过程比较复杂，需要通过专家知识和反馈逐步得到正确结果。因此对比 4 个候选项，黑板结构特别适合求解这类问题。

试题 18 答案

(18) C

试题 19~20 分析

本题主要考查特定领域软件架构的基本定义和基本活动。特定领域软件架构 (DSSA) 是在一个特定应用领域为一组应用提供组织结构参考的标准软件架构。实施 DSSA 的过程中包括一系列基本的活动，其中领域设计活动的主要目的是为了获得 DSSA。该活动参加人员中，领域专家的主要任务是提供关于领域中系统的需求规约和实现的知识。

试题 19~20 答案

(19) C

(20) A

试题 21~23 分析

本题主要考查质量属性以及实现质量属性的一般策略，不同策略主要针对一个或多个软件质量属性，其中 Ping/Echo 主要提高系统的可用性；限制访问主要提高系统的安全性；运行时注册主要提高系统的可修改性；接口-实现分离主要提高系统的可修改性；主动冗余提高系统的可靠性；队列调度主要提高系统的性能；信息隐藏主要提高系统的可修改性；记录-回放主要提高系统的可测试性，等等。

试题 21~23 答案

(21) A

(22) D

(23) A

试题 24 分析

本题主要考查软件质量属性的判断与应用。“系统出错后不能在要求的时间内恢复到正常状态”，这是对系统错误恢复能力的描述，属于系统可用性的范畴。“对系统进行二次开发时总要超过半年的时间”，这是对系统进行调整和维护方面能力的描述，属于系统可修改性的范畴。

试题 24 答案

(24) D

试题 25 分析

本题主要考查设计模式的理解与应用。根据题干描述，在线文档系统需要根据用户的查询需求逐步将查询请求依次传递，对比 4 个候选项，其中在责任链模式里，很多对象由每一个对象对其下家的引用而连接起来形成一条链。请求在这个链上传递，直到链上的某一个对象决定处理此请求。因此责任链模式是能够满足该要求的最好模式。

试题 25 答案

(25) A

试题 26 分析

本题主要考查设计模式的理解与应用。根据题干描述，应用系统需要使用希赛公司开发的类库，该应用系统是一组窗格组成，应用需要协调窗格之间的行为，并且不能引用窗格自身，在这种要求下，对比 4 个候选项，其中中介者模式用一个中介对象封装一系列的对象交互。中介者使用各对象不需要显式的相互调用，从而使其耦合松散。可以看出该模式最符合需求。

试题 26 答案

(26) B

试题 27 分析

本题主要考查设计模式的理解与应用。根据题干描述，该编辑器需要在文档中嵌入显示开销很大的图形对象，为了能够提高系统效率，需要避免同时创建这些图像。这对这些要求，对比候选项，可以发现代理模式可以解决直接访问对象时带来的问题，例如：要访问的对象在远程的机器上；对象创建开销很大，或者某些操作需要安全控制，或者需要进程外的访问等。因此代理模式是最为合适的设计模式。

试题 27 答案

(27) A

试题 28~29 分析

本题主要考查软件架构评价的理解和应用。正确识别风险点、非风险点、敏感点和权衡点是进行软件架构评价的关键步骤。其中敏感点是实现一个特定质量属性的关键特征，该特征为一个或多个软件构件所共有。系统权衡点会影响一个或多个属性，并对于多个属性来说都是敏感点。基于该定义，可以看出“改变加密的级别可能会对安全性和性能都产生显著的影响”正是一个对系统权衡点的描述。

试题 28~29 答案

(28) C

(29) D

试题 30~32 分析

本题考查提高质量属性的常见手段。

提高可用性的手段包括：命令/响应机制、心跳机制、异常处理机制、冗余机制等。

提高性能的手段包括：引入并发、维持数据或计算的多个副本、增加可用资源、控制采样频度、限制执行时间、固定优先级调度等。

提高安全性的手段包括：身份认证、限制访问、检测攻击、维护完整性等。

试题 30~32 答案

(30) A

(31) D

(32) C

9

第9章

应用数学

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

- 概率统计应用。
- 图论应用。
- 组合分析。
- 算法（数值算法与非数值算法）的选择与应用。
- 运筹方法（网络计划技术、线性规划、预测、决策、库存管理、模拟）。
- 数学建模。

9.1 考点突破

从历年的考试情况来看，本章的考点主要集中于数学建模、决策论、线性规划、图论。

9.1.1 历年考试情况分析

在历年的考试试题中，有关应用数学知识的试题如表 9-1 所示。

表 9-1 应用数学知识试题分布表

题号	2009.11	2010.11	2011.11
69	数学建模	数学建模	线性规划
70	转移矩阵	盈亏平衡点	线性规划

按照知识点进行总结和归类的试题分布情况如表 9-2 所示。

表 9-2 应用数学知识归类表

知识点	2009.11	2010.11	2011.11
数学建模	1	1	0
线性规划	0	0	2
图论	0	0	0
决策论	1	0	0
其他	0	1	0
合计	2	2	2

从表 9-2 中可以看出,应用数学知识方面的内容在历年的考试中分值非常稳定,一直稳定维持在 2 分。所占分数比例的趋势如图 9-1 所示。



图 9-1 应用数学知识历年试题比例趋势图

9.1.2 数学建模

当需要从定量的角度分析和研究一个实际问题时,人们就要在深入调查研究、了解对象信息、作出简化假设、分析内在规律等工作的基础上,用数学的符号和语言,把它表述为数学式子,也就是数学模型,然后用通过计算得到的模型结果来解释实际问题,并接受实际的检验。这个建立数学模型的全过程就称为数学建模。

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。当数学建模完成以后,可以产生相应的数学建模应用。如图 9-2 所示为数学、数学建模、数学建模应用之间的关系。

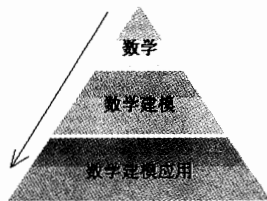


图 9-2 数学建模

应用数学去解决各类实际问题时,建立数学模型是十分关键的一步,同时也是十分困难的一步。建立数学模型的过程,是把错综复杂的实际问题简化、抽象为合理的数学结构的过程。要通过调查、收集数据资料,观察和研究实际对象的固有特征和内在规律,抓住问题的主要矛盾,建立起反映实际问题的数量关系,然后利用数学理论和方法去分析和解决问题。这就需要深厚而扎实的数学基础,敏锐的洞察力和想象力,对实际问题的浓厚兴趣和广博的知识面。

1. 数学建模的过程

虽然面临的各种实际问题不一样,但数学建模的基本过程基本上是一致的,可以遵循以下过程:

(1) 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。

(2) 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。

(3) 模型建立

在假设的基础上,利用适当的数学工具来刻画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。

(4) 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)。

(5) 模型分析

对所得的结果进行数学上的分析。

(6) 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。

模型检验的方法有多种:

- 实际案例数据对模型进行检验:将模型作为一个黑盒,通过案例数据的输入,检查其输出是否合理。这是应用人员常用的方法。
- 专家分析法:由经验丰富的专家根据模型自身的逻辑,再结合实际情况,分析是否会出现矛盾或问题。
- 计算机模拟分析法:有时,很难用实际案例或聘请专家来检验模型,例如,试验或实验的代价太大,难以取得实际案例,有的项目技术比较新,缺乏有经验的专家。这时,可采用该方法。先利用计算机来模拟实际问题,再在计算机上检验该数学模型。例如,对某种核辐射防护建立的数学模型,采用计算机模拟分析法来检验就十分有效。

(7) 模型应用

应用方式因问题的性质和建模的目的而异。

2. 数学建模的方法

构造模型是一种创造性劳动,成功的模型往往是科学与艺术的结晶,一般的建模方法和思路有以下4种。

- 直接分析法:根据对问题内在机理的认识,直接构造出模型。
- 类比法:根据类似问题的模型构造新模型。
- 数据分析法:通过试验,获得与问题密切相关的大量数据,用统计分析方法进行建模。

- 构想法：对将来可能发生的情况给出逻辑上合理的设想和描述，然后用已有的方法构造模型，并不断修正完善，直至比较满意为止。

9.1.3 线性规划

线性规划是研究在有限的资源条件下，如何有效地使用这些资源达到预定目标的数学方法。用数学的语言来说，也就是在一组约束条件下寻找目标函数的极值问题。

求极大值（或极小值）的模型表达如下：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_n \end{cases}, \quad x_i \geq 0, \quad 1 \leq i \leq n$$

在上述条件下，求解 x_1, x_2, \dots, x_n ，使满足下列表达式的 z 取极大值（或极小值）：

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

1. 图解法

解线性规划问题的方法有很多，最常用的有图解法、单纯形法、解方程法。图解法简单直观，而解方程法在两个不等式和三个不等式时能快速解决问题。下面，通过一个例子来说明这两种方法的应用。

某工厂在计划期内要安排生产 I、II 两种产品，已知生产单位产品所需的设备台时及 A、B 两种原材料的消耗如表 9-3 所示。

表 9-3 产品及原材料表

	I	II	总数
设 备	1	2	8 台时
原材料 A	4	0	16kg
原材料 B	0	4	12kg

该工厂每生产一件产品 I 可获利 2 元，每生产一件产品 II 可获利 3 元，问应该如何安排计划使该工厂获利最多？

该问题可用以下数学模型来描述：设 x_1, x_2 分别表示在计划期内产品 I、II 的产量，因为设备的有效台时是 8，这是一个限制产量的条件，所以在确定产品 I、II 的产量时，要考虑不超过设备的有效台时数，即可用不等式表示为

$$x_1 + 2x_2 \leq 8$$

同理，因原材料 A、B 的限量，可以得到以下不等式

$$4x_1 \leq 16$$

$$4x_2 \leq 12$$

该工厂的目标是在不超过所有资源限制的条件下, 如何确定产量 x_1, x_2 , 以得到最大的利润。若用 z 表示利润, 这时 $z = 2x_1 + 3x_2$ 。综上所述, 该计划问题可用数学模型表示为:

目标函数:

$$\max z = 2x_1 + 3x_2$$

满足约束条件:

$$x_1 + 2x_2 \leq 8$$

$$4x_1 \leq 16$$

$$4x_2 \leq 12$$

$$x_1, x_2 \geq 0$$

在以 x_1, x_2 为坐标轴的直角坐标系中, 非负条件 $x_1, x_2 \geq 0$ 是指第一象限。上述每个约束条件都代表一个半平面。例如, 约束条件 $x_1 + 2x_2 \leq 8$ 代表以直线 $x_1 + 2x_2 = 8$ 为边界的左下方的半平面。若同时满足 $x_1, x_2 \geq 0$, $x_1 + 2x_2 \leq 8$, $4x_1 \leq 16$ 和 $4x_2 \leq 12$ 的约束条件的点, 必然落在由这 3 个半平面相交组成的区域内, 如图 9-3 中的阴影部分所示。阴影区域中的每一个点 (包括边界点) 都是这个线性规划问题的解 (称可行解), 因而此区域是本题的线性规划问题的解的集合, 称它为可行域。

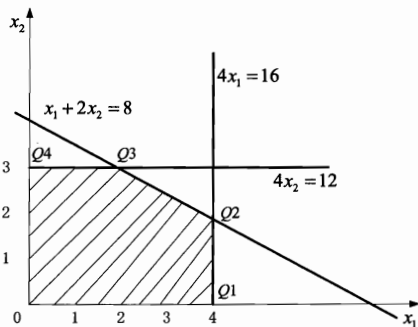


图 9-3 图解法

再分析目标函数 $z = 2x_1 + 3x_2$, 在坐标平面上, 它可表示以 z 为参数, $-2/3$ 为斜率的一组平行线:

$$x_2 = -\left(\frac{2}{3}\right)x_1 + \frac{z}{3}$$

位于同一直线上的点, 具有相同的目标函数值, 因此称它为等值线。当 z 值由小变大时, 直线沿其法线方向向右上方移动。当移动到 Q_2 点时, 使 z 值在可行域边界上实现最大化, 这就得到了本题的最优解 Q_2 , Q_2 点的坐标为 (4,2)。经过计算, 可以得

出 $z=14$ 。

这说明该厂的最优生产计划方案是：生产 4 件产品 I，2 件产品 II，可得最大利润为 14 元。

2. 关于解的讨论

在上述例题中，得到的最优解是唯一的，但对一般线性规划问题而言，求解结果还可能出现以下几种情况：无穷多最优解（多重解）、无界解（无最优解）、无可行解。当求解结果出现后两种情况时，一般说明线性规划问题的数学模型有错误。无界解源于缺乏必要的约束条件，无可行解源于矛盾的约束条件。

从图解法中直观地看到，当线性规划问题的可行域非空时，它是有界或无界凸多边形。若线性规划问题存在最优解，它一定在可行域的某个顶点得到；若在两个顶点同时得到最优解，则它们连线上的任意一点都是最优解，即有无穷多最优解。

3. 解方程法

使用解方程法解答时，如果只有两个不等式，则可以通过解不等式组直接求出最优解。但如果方程有 3 个不等式，则需要每两个不等式求解，然后将解套入到第 3 个不等式进行验证，以求得最优解。下面以表 9-3 所示问题为例进行求解。

$$\begin{cases} x_1 + 2x_2 \leq 8 \\ 4x_1 \leq 16 \\ 4x_2 \leq 12 \end{cases}$$

第一步：解 1 式与 2 式组成的不等式组，得到一组解： $x_1 \leq 4$ ， $x_2 \leq 2$ 。将结果套入 3 式进行验证，发现结果符合 3 式的要求，所以该解是一组合解。

第二步：解 1 式与 3 式组成的不等式组，得到一组解： $x_1 \leq 2$ ， $x_2 \leq 3$ 。将结果套入 2 式进行验证，发现结果符合 2 式的要求，所以该解也是一组合解。

第三步：解 2 式与 3 式组成的不等式组，得到一组解： $x_1 \leq 4$ ， $x_2 \leq 3$ 。将结果套入 1 式进行验证，发现结果不符合 1 式要求，所以该解不是合法解。

第四步：在合法解中找出最优解，将第一组解套入 $2x_1 + 3x_2$ ，即 $2 \times 4 + 3 \times 2$ ，得到的最大值为 14。而另一组解套入 $2x_1 + 3x_2$ ，即 $2 \times 2 + 3 \times 3$ ，得到的最大值为 13。所以应选： $x_1 \leq 4$ ， $x_2 \leq 2$ 。

4. 线性规划的适用性

线性规划模型用在原材料单一、生产过程稳定不变、分解型生产类型的企业是十分有效的，如石油化工厂等。对于产品结构简单、工艺路线短或者零件加工企业，有较大的应用价值。需要注意的是，对于机电类企业用线性规划模型只适用于作年度的总生产计划，而不宜用来做月度计划。这主要与工件在设备上的排序有关，计划期太短，很难安排过来。

一般来说，一个经济管理问题符合以下条件时，才能建立线性规划的模型：

- 要求解问题的目标函数能用数值指标来反映，且为线性函数。

- 存在着多种方案。
- 要求达到的目标是在一定约束条件下实现的, 这些约束条件可用线性等式或不等式描述。

9.1.4 图论

图论的知识范围十分广泛, 但考试要求掌握的知识点并不多, 主要了解两个方面的内容: 最小生成树和关键路径。

1. 最小生成树

树是图论中最主要的概念之一, 而且是最简单的图之一。它在计算机科学中应用非常广泛。我们从一个问题谈起, 如图 9-4 所示是通信线路图。

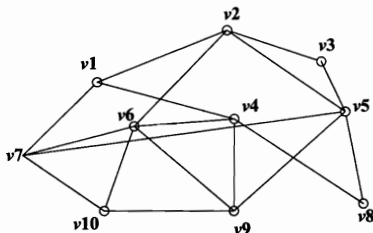


图 9-4 由线路相接的城市

其中, v_1, v_2, \dots, v_{10} 是 10 个城市, 线路只能在这里相接。不难发现, 只要破坏了几条线路, 立即使这个通信系统分解成不相连的两部分。但要问在什么情况下这 10 个城市依然保持相通? 不难知道, 至少要有 9 条线把这 10 个城市连接在一起, 显然这 9 条线是不存在任何回路的, 因而 9 条线少 1 条就会使系统失去连通性。

一个连通且无回路的无向图称为树。在树中度数为 1 的节点称为树叶, 度数大于 1 的节点称为分枝点或内点。一个无回路的无向图称为森林, 它的每一个连通分图是一棵树。

给定图 T , 以下关于树的定义是等价的:

- 无回路的连通图。
- 无回路且 $e=v-1$, 其中 e 为边数, v 为节点数。
- 连通且 $e=v-1$ 。
- 无回路且增加一条新边, 得到一个且仅一个回路。
- 连通且删去任何一个边后不连通。
- 每一对节点之间有一条且仅一条路。

在带权的图 G 的所有生成树中, 树权最小的那棵生成树, 称作最小生成树。

求连通的带权无向图的最小代价生成树的算法有普里姆 (Prim) 算法和克鲁斯卡尔 (Kruskal) 算法。

(1) 普里姆算法

设已知 $G=(V,E)$ 是一个带权连通无向图, 顶点 $V=\{0,1,2,\dots,n-1\}$ 。设 U 是构造生成

树过程中已被考虑在生成树上的顶点的集合。初始时, U 只包含一个出发顶点。设 T 是构造生成树过程中已被考虑在生成树上的边的集合, 初始时 T 为空。如果边 (ij) 具有最小代价, 且 $i \in U, j \in V-U$, 那么最小代价生成树应包含边 (ij) 。把 j 加到 U 中, 把 (ij) 加到 T 中。重复上述过程, 直到 U 等于 V 为止。这时, T 即为要求的最小代价生成树的边的集合。

普里姆算法的特点是当前形成的集合 T 始终是一棵树。因为每次添加的边是使树中的权尽可能小, 因此这是一种贪心的策略。普里姆算法的时间复杂度为 $O(n^2)$, 与图中边数无关, 所以适合于稠密图。

(2) 克鲁斯卡尔算法

设 T 的初始状态只有 n 个顶点而无边的森林 $T=(V, \emptyset)$, 按边长递增的顺序选择 E 中的 $n-1$ 安全边 (u,v) 并加入 T , 生成最小生成树。所谓安全边是指两个端点分别是森林 T 里两棵树中的顶点的边。加入安全边, 可将森林中的两棵树连接成一棵更大的树, 因为每一次添加到 T 中的边均是当前权值最小的安全边, MST 性质也能保证最终的 T 是一棵最小生成树。

克鲁斯卡尔算法的特点是当前形成的集合 T 除最后的结果外, 始终是一个森林。克鲁斯卡尔算法的时间复杂度为 $O(e \log_2 e)$, 与图中顶点数无关, 所以较适合于稀疏图。

2. 关键路径

在 AOV 网络中, 如果边上的权表示完成该活动所需的时间, 则称这样的 AOV 为 AOE 网络。如图 9-5 所示为一个具有 10 个活动的某个工程的 AOE 网络。图中有 7 个顶点, 分别表示事件 1~7, 其中, 1 表示工程开始状态, 7 表示工程结束状态, 边上的权表示完成该活动所需的时间。

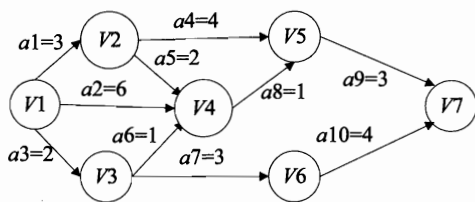


图 9-5 AOE 网络的例子

因 AOE 网络中的某些活动可以并行地进行, 所以完成工程的最少时间是从开始顶点到结束顶点的最长路径长度, 从开始顶点到结束顶点的最长路径称为关键路径 (临界路径), 关键路径上的活动称为关键活动。

为了找出给定的 AOE 网络的关键活动, 从而找出关键路径, 先定义几个重要的量:

$V_e(j)$ 、 $V_l(j)$: 顶点 j 事件最早、最迟发生时间。

$e(i)$ 、 $l(i)$: 活动 i 最早、最迟开始时间。

从源点 V_1 到某顶点 V_j 的最长路径长度, 称为事件 V_j 的最早发生时间, 记作 $V_e(j)$ 。 $V_e(j)$ 也是以 V_j 为起点的出边 $\langle V_j, V_k \rangle$ 所表示的活动 a_i 的最早开始时间 $e(i)$ 。

在不推迟整个工程完成的前提下, 一个事件 V_j 允许的最迟发生时间, 记作 $V_l(j)$ 。

显然, $l(i)=V_e(j)-(a_i)$, a_i 表示所需时间其中 j 为 a_i 活动的终点。满足条件 $l(i)=e(i)$ 的活动为关键活动。

求顶点 V_j 的 $V_e(j)$ 和 $V_l(j)$ 可按以下两步来做:

(1) 由源点开始向汇点递推

$$\begin{cases} V_e(1) = 0 \\ V_e(j) = \max \{V_e(i) + d(i, j)\}, < V_i, V_j \in E_1, 2 \leq j \leq n \end{cases}$$

其中, E_1 是网络中以 V_j 为终点的入边集合。

(2) 由汇点开始向源点递推

$$\begin{cases} V_l(n) = V_e(n) \\ V_l(j) = \min \{V_l(k) - d(j, k)\}, < V_j, V_k \in E_2, 2 \leq j \leq n-1 \end{cases}$$

其中, E_2 是网络中以 V_j 为起点的出边集合。

要求一个 AOE 的关键路径, 一般需要根据以上变量列出一张表格, 逐个检查。例如, 求图 9-5 所示的 AOE 的关键路径的表格如表 9-4 所示。

表 9-4 求关键路径的过程

V_i	$V_e(i)$	$V_l(i)$	a_i	$e(i)$	$l(i)$	$l(i)-e(i)$
V_1	0	0	$a_1(3)$	0	0	0
V_2	3	3	$a_2(6)$	0	0	0
V_3	2	3	$a_3(2)$	0	1	1
V_4	6	6	$a_4(4)$	3	3	0
V_5	7	7	$a_5(2)$	3	4	1
V_6	5	6	$a_6(1)$	2	5	3
V_7	10	10	$a_7(3)$	2	3	1
			$a_8(1)$	6	6	0
			$a_9(3)$	7	7	0
			$a_{10}(4)$	5	6	1

因此, 图 9-4 所示的关键活动为 a_1, a_2, a_4, a_8 和 a_9 , 其对应的关键路径有两条, 分别为 (V_1, V_2, V_5, V_7) 和 (V_1, V_4, V_5, V_7) , 长度都是 10。

9.1.5 决策论

决策就是决定的意思, 大至国家经济、政治, 小到个人生活, 凡是在有选择的地方就有决策。关于决策的重要性, 诺贝尔奖金获得者西蒙有一句名言“管理就是决策”。这就是说, 管理的核心是决策。

按决策环境分类, 可分为确定型决策 (决策环境是完全确定的, 作出的选择的结

果也是确定的），风险决策（决策的环境不是完全确定的，其发生的概率是已知的）和不确定型决策（将来发生结果的概率不确定，凭主观倾向进行决策）。

1. 不确定型决策

不确定型决策（非确定型决策）是指决策者对环境情况一无所知，决策者根据自己的主观倾向进行决策。根据决策者的主观态度不同，可分为 5 种准则，分别为悲观主义准则、乐观主义准则、折中主义准则、等可能性准则和后悔值准则。下面通过一个例题，具体介绍这些准则的含义和求解方法。

希赛公司需要根据下一年度宏观经济的的增长趋势预测决定投资策略。宏观经济增长趋势有不景气、不变和景气 3 种，投资策略有积极、稳健和保守 3 种，各种状态的收益如表 9-5 所示。

表 9-5 各种状态的收益

预计收益（单位：百万元人民币）		经济趋势预测		
		不景气	不变	景气
投资策略	积极	50	150	500
	稳健	150	200	300
	保守	400	250	200

在本题中，由于下一年度宏观经济的各种增长趋势的概率是未知的，所以是一个不确定型决策问题。

(1) 乐观主义准则

乐观主义准则也称为最大最大准则（maxmax 准则），其决策的原则是“大中取大”。持这种准则思想的决策者对事物总抱有乐观和冒险的态度，他决不放弃任何获得最好结果的机会，争取以“好中之好”的态度来选择决策方案。决策者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。在表 9-5 中，积极方案的最大结果为 500，稳健方案的最大结果为 300，保守方案的最大结果为 400。三者的最大值为 500，因此，选择其对应的积极投资方案。

(2) 悲观主义准则

悲观主义准则也称为最大最小准则（maxmin 准则），其决策的原则是“小中取大”。这种决策方法的思想是对事物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。在表 9-5 中，积极方案的最小结果为 50，稳健方案的最小结果为 150，保守方案的最小结果为 200。三者的最大值为 200，因此，选择其对应的保守投资方案。

(3) 折中主义准则

折中主义准则也称为赫尔威斯（Harwicz）准则，这种决策方法的特点是对事物既不乐观冒险，也不悲观保守，而是从中折中平衡一下，用一个系数 α （称为折中系数）来表示，并规定 $0 \leq \alpha \leq 1$ ，用以下算公式计算结果：

$$cv_i = \alpha \times \max\{a_{ij}\} + (1 - \alpha) \times \min\{a_{ij}\}$$

即用每个决策方案在各个自然状态下的最大效益值乘以 α ，再加上最小效益值乘以 $1-\alpha$ 。然后再比较 cv_i ，从中选择最大者。显然，折中主义准则的结果取决于 α 的选择。 α 接近于 1，则偏向于乐观； α 接近于 0，则偏向于悲观。

(4) 等可能准则

等可能准则也称为拉普拉斯 (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，就可以把每个状态出现的概率定为 $1/n$ (n 是自然状态数)，然后按照最大期望值准则决策。也就是说，把一个不确定型决策转换为风险决策。

(5) 后悔值准则

后悔值 (遗憾值) 准则也称为萨维奇 (Savage) 准则、最小机会损失准则。决策者在制订决策之后，如果不能符合理想情况，必然有后悔的感觉。这种方法的特点是每个自然状态的最大收益值 (损失矩阵取为最小值)，作为该自然状态的理想目标，并将该状态的其他值与最大值相减所得的差作为未达到理想目标的后悔值。这样，从收益矩阵就可以计算出后悔值矩阵。最后按照最大后悔值达到最小的方法进行决策，因此，也称为最小最大后悔值 (minmax)。在本题中，根据表 9-5 可以得出后悔值矩阵，如表 9-6 所示。

表 9-6 各种状态的后悔值

预计收益 (单位: 百万元人民币)		经济趋势预测		
		不景气	不变	景气
投资策略	积极	350	100	0
	稳健	250	50	200
	保守	0	0	300

在表 9-6 中，积极方案的最大后悔值为 350，稳健方案的最大后悔值为 250，保守方案的最大后悔值 300。三者的最小值为 250，因此，选择其对应的稳健投资方案。

2. 风险决策

风险决策是指决策者对客观情况不甚了解，但对将发生各事件的概率是已知的。在风险决策中，一般采用期望值作为决策准则，常用的有最大期望收益决策准则 (Expected Monetary Value, EMV) 和最小机会损失决策准则 (Expected Opportunity Loss, EOL)。

(1) 最大期望收益决策准则

决策矩阵的各元素代表“策略-事件”对的收益值，各事件发生的概率为 p_j ，先计算各策略的期望收益值 $\sum_i p_j a_{ij}$ ， $i=1,2,\dots,n$ ，然后从这些期望收益值中选取最大者，以它对应的策略为决策者应选择的决策策略。

(2) 最小机会损失决策准则

决策矩阵的各元素代表“策略-事件”对的损失值，各事件发生的概率为 p_j ，先

计算各策略的期望损失值 $\sum_i p_j a_{ij}'$, $i=1,2,\dots,n$, 然后从这些期望收益值中选取最小者, 以它对应的策略为决策者应选择的决策策略。

当 EMV 为最大时, EOL 便为最小。因此, 在决策时用这两个决策准则所得的结果是一致的。

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?

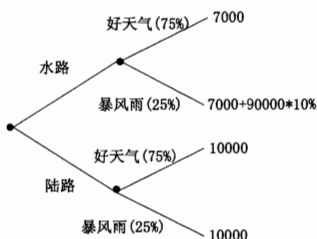


图 9-6 运输问题的决策树

这是一个风险决策问题, 其决策树如图 9-6 所示。

根据图 9-6, 走水路时, 成本为 7000 元的概率为 75%, 成本为 16000 元的概率为 25%, 因此, 走水路的期望成本为 $(7000 \times 75\%) + (16000 \times 25\%) = 9250$ 元; 走陆路时, 其成本为 $(10000 \times 75\%) + (10000 \times 25\%) = 10000$ 元。所以, 走水路的期望成本小于走陆路的成本, 应该选择走水路。

3. 转移矩阵

转移矩阵, 又称转移概率矩阵, 它是俄国数学家马尔科夫提出的, 他在 20 世纪初发现: 一个系统的某些因素在转移中, 第 n 次结果只受第 $n-1$ 的结果影响, 即只与当前所处状态有关, 而与过去状态无关。在马尔科夫分析中, 引入状态转移这个概念。所谓状态是指客观事物可能出现或存在的状态; 状态转移是指客观事物由一种状态转变为另一种状态。在考试当中, 主要掌握 2×2 的转移矩阵计算。

下面以一个例题说明该计算过程。

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:

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

其中, $p(A \rightarrow B)$ 是 A 的市场占有份额中转移给 B 的概率, 依此类推。这样, 两个月后的这种商品的市场占有率变化为_____。

- 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%

在解答这个问题时, 首先应该弄明白转移矩阵中数据的含义。

$p(A \rightarrow A)$ 代表：原来是 A 的份额，下个月转移到 A 的概率。对应的是 0.8。这也就是说：A 有 80%的份额能保留下来。

$p(A \rightarrow B)$ 代表：原来是 A 的份额，下个月转移到 B 的概率。对应的是 0.2。这也就是说：A 有 20%的份额会被 B 抢去。

$p(B \rightarrow A)$ 代表：B 可以抢 A 的份额，A 同样也可以抢 B 的。此处说明：B 的 40%被 A 抢走了。

$p(B \rightarrow B)$ 代表：原来是 B 的份额，下个月转移到 B 的概率。对应的是 0.6。这也就是说：B 有 60%的份额能保留下来。

从这里可以看出，过了一个月后。A 的份额由两部分组成：第一部分是自己保留下来的；第二部分是从 B 那边抢过来的。

具体如下：

$0.5 \times 0.8 + 0.5 \times 0.4 = 0.6$ ，所以 1 个月后 A 占 60%，B 占 40%。2 个月后的情况，应在 1 个月的数据基础上进行计算。即：

$0.6 \times 0.8 + 0.4 \times 0.4 = 0.64$ ，所以 2 个月后 A 占 64%，B 占 36%。从而得到答案 C。

9.2 典型试题分析

试题 1

如图 9-7 所示为某地区的运输网。

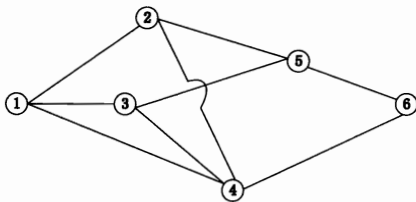


图 9-7 某地区的运输网

各节点之间的运输能力如表 9-7 所示（单位：万吨/小时）：

表 9-7 各节点之间的运输能力

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

从节点①到节点⑥的最大运输能力（流量）可以达到（1）万吨/小时。

(1) A.26

B.23

C.22

D.21

试题分析

为了便于计算，我们把表 9-7 的数据标记到图 9-7 上，形成图 9-8。

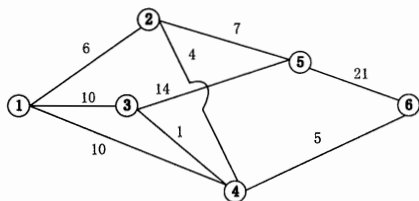


图 9-8 新的运输网

从图 9-8 中可以看出，只能从节点④和⑤到达节点⑥，其运输能力为 26。而只能从节点②和③到达节点⑤，且能满足最大运输量 21 (14+7)。但是，到达节点③的最大数量为 11 (10+1)，因此，节点⑤的最终输出能力为 18，即从节点①到节点⑥的最大运输能力为 23。最终的运输方案如图 9-9 所示。

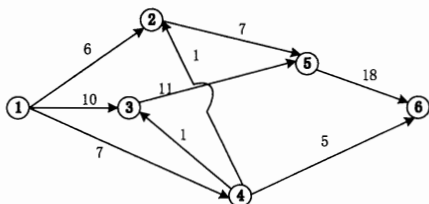


图 9-9 最终运输方案

试题答案

(1) B

试题 2

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 9-8 所示的系统 A 和 B 进行评估，那么乙认为(2)。

表 9-8 待评估情况表

		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

(2) 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

试题分析

点值评估方法将根据每一个价值的重要性为其指定一个加权值（以百分数表示），然后用每个因素来评估每一个候选的信息系统方案，得到一个等级分，范围为 0~100，其中 0 分表示候选方案根本没有达到价值因素的要求，100 分表示该候选方案完全达到了价值因素的要求。将每个候选系统的各个价值因素的得分累加，总分最佳的候选方案为最佳选择。

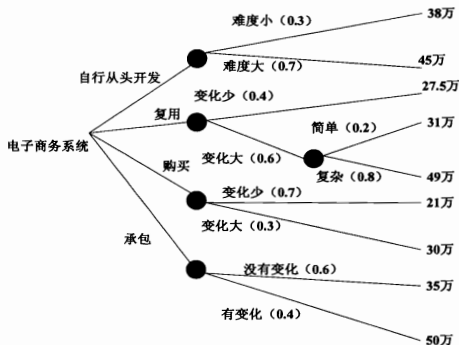
根据题目给出的表，相对于硬件（35%）和供应商支持（25%），软件这一因素被赋予最大的权重（40%），系统 A 的得分是 82.5，系统 B 的得分是 86.75（更接近 100），所以 B 是最佳方案。

试题答案

(2) B

试题 3

某企业拟进行电子商务系统的建设，有 4 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，项目经理提供了如图 9-10 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 (3)。



(3) 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

试题分析

在软件应用领域，直接获取（购买）计算机软件常常比自行开发的成本要低得多。软件工程管理者面临着做出自行开发还是购买的决策问题，而且由于存在多种可选的获取方案使得决策更加复杂。这些决策包括：从头开始构造系统；复用已有的“具有部分经验”的构件来构造系统；购买现在的开发软件，并进行修改已满足当前项目的需要；将软件开发承包给外面的开发商。

可以采用决策树分析来帮助管理者进行决策，题目给的图中路径上的数字是出现该情况的概率，树叶金额是出现该情况的成本，成本的期望值 = $\sum(\text{路径概率})_i \times (\text{估算的})$

路径成本) i ，其中 i 是决策树的某条路径。经过计算可知，采用购买方式预期方式预期成本最低，因此管理者选择建设方式的最佳决策是购买现成的软件产品。

试题答案

(3) C

试题 4

每个线性规划问题需要在有限个线性约束条件下，求解线性目标函数 F 何处能达到极值。有限个线性约束条件所形成的区域（可行解区域），由于其边界比较简单（逐片平直），人们常称其为单纯形区域。单纯形区域 D 可能有界，也可能无界，但必是凸集（该区域中任取两点，则连接这两点的线段全在该区域内）必有有限个顶点。

以下关于线性规划问题的叙述中，不正确的是（4）。

- (4) A. 若 D 有界，则 F 必能在 D 的某个顶点上达到极值
B. 若 F 在 D 中 A 、 B 点上都达到极值，则在 AB 线段上也都能达到极值
C. 若 D 有界，则该线性规划问题一定有一个或无穷多个最优解
D. 若 D 无界，则该线性规划问题没有最优解

试题分析

本题旨在从宏观上理解线性规划方法的原理与机制，特别是从二维、三维的直观理解推广到高维的理解。这种宏观、直观的理解对于深刻认识数学概念、方法是非常重要的，对于创新也会有重要的、奇特的启发作用。

很明显，有界区域内线性函数的值域肯定是有界的。从直观上可以理解，由于线性函数的平坦性，其极值一定会在边界上达到（许多教材上给出了严格证明）。直观的理解有助于形象地感悟某些理论研究的结论。由于单纯形区域的边界是逐片平直的，所以它对应的线性目标函数值域也会逐片平直的，人们可以想象，线性函数 F 会在 D 区域的顶点处达到极值。所以选项A是正确的。

由于单纯形区域是凸集，只要 A 、 B 两点在区域内，则线段 AB 全在该区域内。由于 $F(A)$ 与 $F(B)$ 在线性目标函数值域上，不难看出，线段 AB 中的任一点 C 对应的 $F(C)$ 就会落在 $F(A)$ 与 $F(B)$ 的连线上。所以选项B也是正确的。

选项C可以从选项A与B导出。线性规划问题要么无解，要么只有唯一的最优解，要么会有无穷多个最优解。因为如果有两个最优解，则这两个解的连线段上所有的解都是最优解。所以选项C也是正确的。

选项D不正确。若区域 D 无界，则线性规划问题可能无解，也可能有解（唯一解或无穷多个解）。

例如，线性规划问题：

$\text{Min } Z=X+Y \quad \text{s.t. } X \geq 0, Y \geq 0$ 的可行解区域是无界的，但在 $X=0, Y=0$ 时有唯一的最优解（极小值） $Z=0$ 。

又例如，线性规划问题：

$\text{Max } Z=2X$ s.t. $X \leq 4, X \geq 0, Y \geq 0$ 的可行解区域是无界的,但在 $X=4, Y \geq 0$ 处有无穷多个最优解(极大值) $Z=8$ 。

又例如,线性规划问题:

$\text{Max } Z=X+Y$ s.t. $X \leq 4, X \geq 0, Y \geq 0$ 的可行解区域是无界的,不存在最优解。

在坚实的理论基础上,直观、形象、宏观地看问题不仅能深刻理解问题的实质,有时还能启发新的思路,创立新的问题求解方法。

例如,用单纯形方法求解线性规划问题的过程,实际上就是在单纯形区域 D 的边界上先选一个初始顶点再通过迭代计算,沿着 D 的边界逐个顶点行进,直到达到最优解的那个顶点为止。

在企业实际应用中,一般会有大量的变量,区域 D 的顶点也很多,这种方法的计算量是很大的。

在直观上看,从区域 D 的一个顶点出发,沿 D 的边界前进直到最优解顶点,一般都是绕弯的。人们会想到,应该有更捷径的路,而这条路可能是从区域 D 内穿过去的。从区域 D 的一个点出发,沿什么方向走会使线性函数值 F 增长最快(或下降最快)呢?显然,应该沿函数 F 的梯度方向(或负梯度方向)前进,直到区域 D 的边界,会有更好的效果。据此,我们认为,可以获得比单纯形法更快的迭代求解方法。当然,单有直观思维是不够的,还需要在这种思维的指导下,去寻求实际可行的求解方法。现在,也确实有人按这种思路获得了新的解法。

试题答案

(4) D

试题 5

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(5)。

- (5) 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

试题分析

连续模型中,模型参数的微小变化都可能会导致计算结果的很大变化。

试题答案

(5) D

试题 6

甲、乙两个独立的网站都主要靠广告收入来支撑发展,目前都采用较高的价格销售广告。这两个网站都想通过降价争夺更多的客户和更丰厚的利润。假设这两个网站

在现有策略下各可以获得 1000 万元的利润。如果一方单独降价,就能扩大市场份额,可以获得 1500 万元的利润,此时,另一方的市场份额就会缩小,利润将下降到 200 万元。如果这两个网站同时降价,则他们都将只能得到 700 万元利润。这两个网站的主管各自经过独立的理性分析后决定, (6)。

- (6) A. 甲采取高价策略,乙采取低价策 B. 甲采取高价策略,乙采取高价策略
C. 甲采取低价策略,乙采取低价策 D. 甲采取低价策略,乙采取高价策略

试题分析

这是一个简单的博弈问题,可以表示为如图 9-11 所示的得益矩阵。

		A网站	
		高价	低价
B网站	高价	1000, 1000	200, 1500
	低价	1500, 200	700, 700

图 9-11 得益矩阵

由图 9-11 可以看出,假设 B 网站采用高价策略,那么 A 网站采用高价策略得 1000 万元,采用低价策略得 1500 万元。因此, A 网站应该采用低价策略。如果 B 网站采用低价策略,那么 A 网站采用高价策略得 200 万元,采用低价策略得 700 万元,因此 A 网站也应该采用低价策略。采用同样的方法,也可分析 B 网站的情况,也就是说,不管 A 网站采取什么样的策略, B 网站都应该选择低价策略。因此,这个博弈的最终结果一定是两个网站都采用低价策略,各得到 700 万元的利润。

这个博弈是一个非合作博弈问题,且两博弈方都肯定对方会按照个体行为理性原则决策,因此虽然双方采用低价策略的均衡对双方都不是理想的结果,但因为两博弈方都无法信任对方,都必须防备对方利用自己的信任(如果有的话)谋取利益,所以双方都会坚持采用低价,各自得到 700 万元的利润,各得 1000 万元利润的结果是无法实现的。即使两个网站都完全清楚上述利害关系,也无法改变这种结局。

试题答案

(6) C

试题 7

图 9-12 标明了 6 个城市(A~F)之间的公路(每条公路旁标注了其长度千米数)。为将部分公路改造成高速公路,使各个城市之间均可通过高速公路通达,至少要改造总计 (7) 千米的公路,这种总千米数最少的改造方案共有 (8) 个。

(7) A.1000

B.1300

C.1600

D.2000

(8) A.1

B.2

C.3

D.4

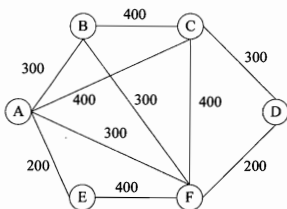


图 9-12 6 个城市(A~F)之间的公路

试题分析

这是一道求图的最小生成树问题，我们使用克鲁斯卡尔算法来解答，如图 9-13 所示。

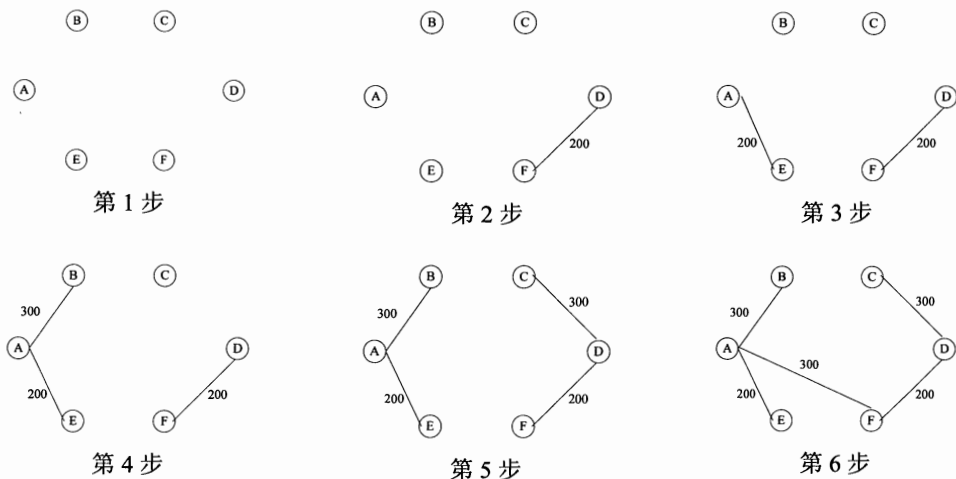


图 9-13 解答过程

到了第 5 步，就有了多种选择，既可以选择 AF，也可以选择 BF，因为其路程都是 300。我们给出的第 6 步是选择 AF 的结果。还有一种结果，就是在第 4 步时，不是选择 AB，而是选择 AF 或者 BF，则结果如图 9-14 所示。

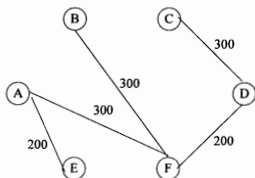


图 9-14 另外一种结果

从第 6 步的结果可以计算出，至少要改造的公路长度为 $200 \times 2 + 300 \times 3 = 1300$ 千米。

试题答案

(7) B

(8) C

试题 8

模型是现实世界的抽象或近似，主要包括叙述型、物理型、图解型和数学型等。无论开发何种模型，(9) 都是最关键的因素。

(9) A. 经济性

B. 简单性

C. 灵活性

D. 准确性

试题分析

模型是现实世界的抽象或近似，其最关键的因素就是准确性，也就是说，使模型要尽量准确地反映现实世界的实际情况。

试题答案

(9) D

试题 9

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(10)。

- (10) 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

试题分析

从实际问题中观察或测量得到的原始数据,通常是不太精确的,也难以完整。需要透过现象看本质,去伪存真,建立比较合理的模型,并求解。建模的过程通常是个渐进的过程。

首先,要根据原始数据初步判断应架构什么样的模型。例如,将一批二维数据画在平面坐标系内,观察它们的分布趋势,初步判断采用什么样的曲线进行拟合比较合适。写出大致的曲线函数表达式,其中必然带有待定的参数。

然后,通过原始数据来估计模型中的参数。算出了参数后,初步的模型就已经建立。但是,该模型是否符合实际,还需要用原始数据来检验。如果发现有些偏差,则需要调整模型或调整参数。一般的建模过程往往要反复多次经历上述过程,逐步优化得到比较合理、适用的模型,然后再选用适当的数值方法进行求解。

试题答案

(10) C

试题 10

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(11)套才能达到盈亏平衡点。

- (11) A. 13000 B. 16000 C. 18000 D. 20000

试题分析

设共销售了 n 套,则总成本=固定成本+每套产品的可变成本 $\times n=160000+2n$ 。销售总收入=单价 $\times n=10n$ 。盈亏平衡时,总成本=销售总收入,所以, $160000+2n=10n$,从而, $n=20000$ (套)。当销售量低于 2 万套时,会有亏损;当销售量超过 2 万套时就会有盈利。

试题答案

(11) D

9.3 实战练习题

● 某 IT 企业计划对一批新招聘的技术人员进行岗前脱产培训, 培训内容包括编程和测试两个专业, 每个专业要求在基础知识、应用技术和实际训练 3 个方面都得到提高。根据培训大纲, 每周的编程培训可同时获得基础知识 3 学分、应用技术 7 学分及实际训练 10 学分; 每周的测试培训可同时获得基础知识 5 学分、应用技术 2 学分及实际训练 7 学分。企业要求这次岗前培训至少能完成基础知识 70 学分, 应用技术 86 学分, 实际训练 185 学分。以上说明如表 9-9 所示。

表 9-9 技术培训表

	编程 (学分/周)	测试 (学分/周)	学分最低要求
基础知识	3	5	70
应用技术	7	2	86
实际训练	10	7	185

那么这样的岗前培训至少需要 (1) 周时间才能满足企业的要求。

- (1) A. 15 B. 18 C. 20 D. 23

● 某软件公司项目 A 的利润分析如表 9-10 所示。设贴现率为 10%, 第二年的利润现值是 (2) 元。

表 9-10 利润分析表

利润分析	第 零 年	第 一 年	第 二 年	第 三 年
利润值		¥ 889,000	¥ 1,139,000	¥ 1,514,000

- (2) A. 1,378,190 B. 949,167 C. 941,322 D. 922,590

● 某软件企业 2004 年初计划投资 1000 万元开发一套中间件产品, 预计从 2005 年开始, 年实现产品销售收入 1500 万元, 年市场销售成本 1000 万元。该产品的系统分析员张工根据财务总监提供的贴现率, 制作了如表 9-11 所示的产品销售现金流量。根据表中的数据, 该产品的动态投资回收期是 (3) 年, 投资收益率是 (4) 。

表 9-11 产品销售现金流量表

年度	2004	2005	2006	2007	2008
投资	1000	-	-	-	-
成本	-	1000	1000	1000	1000
收入	-	1500	1500	1500	1500
净现金流量	-1000	500	500	500	500
净现值	-925.93	428.67	396.92	367.51	340.29

- (3) A. 1 B. 2 C. 2.27 D. 2.73
- (4) A. 42% B. 44% C. 50% D. 100%

● 希赛公司的销售收入状态如表 9-12 所示, 该公司达到盈亏平衡点时的销售收入是 (5) (百万元人民币)。

表 9-12 希赛公司的销售收入状态

项目	金额 (单位百万元人民币)
销售收入	800
材料成本	300
分包费用	100
固定生产成本	130
毛利	270
固定销售成本	150
利润	120

(5) A. 560

B. 608

C. 615

D. 680

● 某工程包括 7 个作业 (A~G)，各作业所需的时间和人数，以及互相衔接的关系如图 9-15 所示 (其中虚线表示不消耗资源的虚作业)。

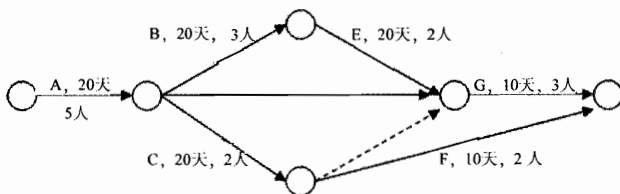
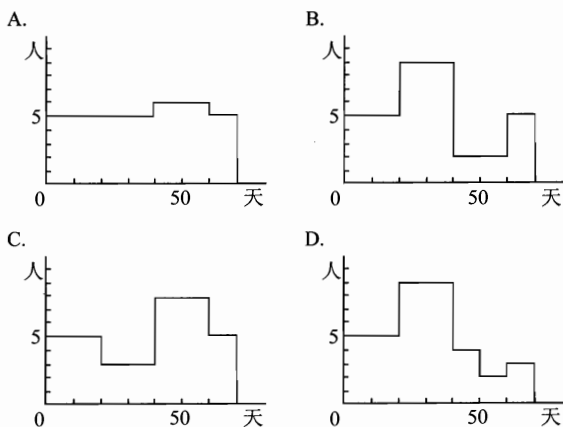


图 9-15 各作业所需的时间和人数以及互相衔接的关系

如果各个作业都按最早可能时间开始，那么，正确描述该工程每一天所需人数的图为 (6)。

(6)



● 系统分析师在收集、分析、处理数据时常会遇到零星异常数据 (野点、离群点)，即大大偏离其他数据值的数据。关于异常数据的叙述，不正确的是 (7)。

(7) A. 异常数据属于不良数据，应尽快找出来，修正或删除它

B. 处理大批数据时，其统计量中位数不易受零星异常数据的影响

C. 用最小二乘法进行线性拟合时，零星异常数据可能导致拟合结果完全失真

D. 测量误差、输入错误、程序运行错误等都可能产生异常数据

- 企业经常要对收集的原始数据进行处理, 数据处理的目的不包括___(8)___。

- (8) A. 增加信息量 B. 变换数据形式使其便于进一步处理
C. 便于使用者检索 D. 为管理人员提供决策支持

● 山区某乡的 6 个村之间有山路如图 9-16 所示, 其中的数字标明了各条山路的长度(千米)。

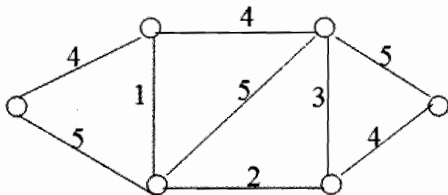


图 9-16 6 个村之间的山路

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___(9)___千米。

- (9) A. 11 B. 14 C. 18 D. 33

● 对实际应用问题建立了数学模型后, 一般还需要对该模型进行检验。通过检验, 尽可能找出模型中的问题, 以利于改进模型, 有时还可能会否定该模型。检验模型的做法有多种, 但一般不会___(10)___。

- (10) A. 利用实际案例数据对模型进行检验
B. 进行逻辑检验, 分析该模型是否会出现矛盾
C. 用计算机模拟实际问题来检验模型
D. 检验该模型所采用的技术能否被企业负责人理解

● 某类产品 n 种品牌在某地区的市场占有率常用概率向量 $u=(u_1, u_2, \dots, u_n)$ 表示(各分量分别表示各品牌的市场占有率, 值非负, 且总和为 1)。市场占有率每隔一定时间的变化常用转移矩阵 $P_{n \times n}$ 表示。设初始时刻的市场占有率为向量 u , 则下一时刻的市场占有率就是 uP , 再下一时刻的市场占有率就是 uP^2 , ...。如果在相当长时期内, 该转移矩阵的元素 s 均是常数, 则市场占有率会逐步稳定到某个概率向量 z , 即出现 $ZP=Z$ 。这种稳定的市场占有率体现了转移矩阵的特征, 与初始时刻的市场占有率无关。

假设占领某地区市场的冰箱品牌 A 与 B, 每月市场占有率的变化可用如一下常数转移矩阵来描述:

$$P = \begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix}$$

则冰箱品牌 A 与 B 在该地区最终将逐步稳定到市场占有率___(11)___。

- (11) A. (1/4, 3/4) B. (1/3, 2/3)

C. (1/2, 1/2)

D. (2/3, 1/3)

● 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(12)。

(12) 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

● 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 9-17 所示）运送到 F 地：

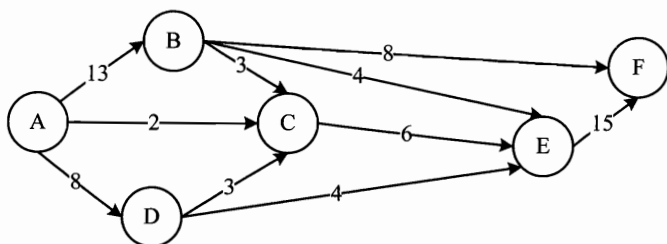


图 9-17 公路网示意图

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_____千人/小时。

(13) A. 20

B. 21

C. 22

D. 23

9.4 练习题解析

试题 1 分析

设安排编程培训 x 周，测试培训 y 周，则可以建立本题的线性规划模型如下：

目标函数： $x+y$ ，求最小值

约束条件： $3x+5y \geq 70$

$$7x+2y \geq 86$$

$$10x+7y \geq 185$$

非负条件： $x, y \geq 0$

该线性规划问题的图解法如图 9-18 所示。

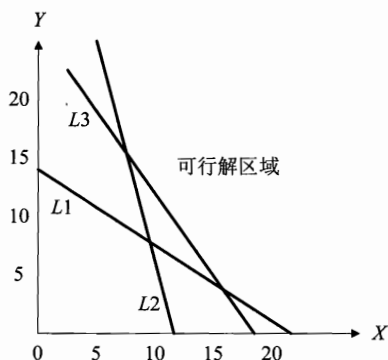


图 9-18 图解法求解线性规划问题

在坐标系第一象限内（因为要求 $x, y \geq 0$ ）画直线 $L1: 3x+5y=70$ （一定通过点(10, 14)与(70/3,0)），所以， $3x+5y \geq 70$ 表示在直线 $L1$ 之上的区域；画直线 $L2: 7x+2y=86$ （一定通过点(0,43)与(86/7,0)），所以， $7x+2y \geq 86$ 表示在直线 $L2$ 之上的区域；画直线 $L3: 10x+7y=185$ （一定通过点(0,185/7)与(20,18.5)），所以， $10x+7y \geq 185$ 表示在直线 $L3$ 之上的区域。上述 3 个约束条件及变量非负条件组成的可行解区域。

根据线性规划方法，目标函数的最小值一定会在可行解区的顶点处到达。因此，只要考察直线 $L1$ 与 $L3$ 的交点，以及直线 $L2$ 与 $L3$ 的交点处目标函数的值。

$L1$ 与 $L3$ 的交点满足：

$$3x+5y=70$$

$$10x+7y=185$$

可以求出可行解区的一个顶点为(15,5)，因此， $x+y=20$ 。

$L2$ 与 $L3$ 的交点满足：

$$7x+2y=86$$

$$10x+7y=185$$

可以求出可行解区的另一个顶点为(8,15)，因此， $x+y=23$ 。

比较这两个顶点处的 $x+y$ 值，就知道本题的最优解是 20 周。

试题 1 答案

(1) C

试题 2 分析

设在第 i 年的利润值为 $GP(i)$ ，第 i 年的贴现系数为 $DR(i)$ ，第 i 年的利润现值为 $NPV(i)$ ，则有 $NPV(i) = GP(i)/DR(i)$ 。因为贴现率为 10%，则第 2 年的贴现系数为 $(1+10\%)^2=1.21$ 。此时根据上述公式可求得第 2 年的利润现值是 941322 元。

试题 2 答案

(2) C

试题 3~4 分析

动态投资回收期和投资收益率是最重要的项目投资决策评价指标。投资回收期从项目的投建之日起,用项目所得的净收益来偿还原始投资所需要的年限。投资回收期分为静态投资回收期和动态投资回收期两种。静态投资回收期考虑资金的占用成本,使用项目建成后年贴现现金流量(即净现值)。

计算动态投资回收期的实用公式如下:

$T_p = \text{累计净现金流量折现值开始出现正值的年份数} - 1 + \frac{\text{上年累计净现金流量折现值}}{\text{当年净现金流量折现值}}$

在本题中,经简单计算表明,在第3年(2007年)中累计折现值开始大于0,因此,动态投资回收期为:

$$T_p = (3-1) + |428.67+396.92-925.93|/367.51 = 2.27$$

投资收益率反应企业投资的获利能力,等于动态投资回收期的倒数。

试题 3~4 答案

(3) C

(4) B

试题 5 分析

盈亏平衡点(又称保本点、盈亏分离点)是指企业经营处于不赢不亏状态所需达到的业务量(产量或销售量),即销售收入等于总成本,是投资或经营中一个很重要的数量界限。近年来,盈亏平衡分析在企业投资和经营决策中得到了广泛的应用。

因此,如果预期销售额与盈亏平衡点接近,则说明项目没有利润。盈亏平衡点越低,表明项目适应市场变化的能力越大,抗风险能力越强。

盈亏平衡点可以通过研究产品的单位售价(P)、单位可变成本(VC)和总固定成本(TFC)来计算。可变成本是与产量水平成比例变化的要素,通常包括原材料、劳动力成本和利用成本。固定成本是不随数量变化的费用。通常包括租金、保险费和财产税。盈亏平衡点的计算公式如下:

$$BEP = TFC/(P-VC)$$

在本题中,固定生产成本为130,固定销售成本为150,因此,总固定成本 TFC 为280。假设年销售产品 x 件,则单位售价为 $P=800/x$,单位可变成本为:

$$VC = (300+100)/x = 400/x$$

所以

$$BEP = 280/(800/x-400/x) = 280x/400 = 0.7x$$

即该公司生产和销售 $0.7x$ 件商品就可达到盈亏平衡,又因为商品的单位售价为 $800/x$,因此,该公司达到盈亏平衡点时的销售收入是

$$(800/x) \cdot 0.7x = 560$$

试题 5 答案

(5) A

试题6分析

从图 9-15 中可以看出, 在前 20 天时, 只有 A 作业在运行, 需要 5 人。在第 21~40 天, 作业 B、C、D 并行运行, 合计需要 $3+4+2=9$ 人。在第 41~50 天, 作业 E 和 F 并行, 需要 $2+2=4$ 人。在第 51~60 天, 只有作业 E 在运行, 需要 2 人。在第 61~70 天, 只有作业 G 在运行, 需要 3 人。

试题6答案

(6) D

试题7分析

异常数据属于不良数据, 应尽快找出来, 查找其原因是什么, 而不是删除它。

试题7答案

(7) A

试题8分析

数据处理是按一定目的, 用一定手段将所获得的原始信息进行加工处理。数据处理的目的是把信息的原始形式变换成便于观察、分析、查找、传递或易于进一步处理的形式; 经过筛选分类、提取过滤和编辑整理, 提高信息的质量; 对数据进行加工计算、分离和选择, 为管理人员提供管理、控制的依据; 将经过处理的数据存储起来, 以便于使用者检索; 发布、销售数据, 供客户使用。数据处理遵循“信息不增原理”, 即数据信号的任何处理、提炼都不能使信息量增加; 相反, 处理的结果常常会损失一些信息量, 处理的环节和次数越多, 损失的机会就越大。对有些用户来说, 最关心的是处理结果是否有用、有价值, 不管是否损失了信息量。例如, 对某个班级的考试成绩经过数据处理后, 获得了平均值、最高与最低值。虽然损失了信息量, 但领导看了觉得很有用。

试题8答案

(8) A

试题9分析

该题考查最小生成树相关知识。解题时, 可以采用克鲁斯卡尔算法, 从图中, 按边权值从小到大顺序来选择边, 当选取的边会形成环路时, 放弃该边的选择。选足 $n-1$ 条边时 (n 为图中的节点数), 即为解。依据该原则, 得到图 9-19, 加粗线组成最小生成树。

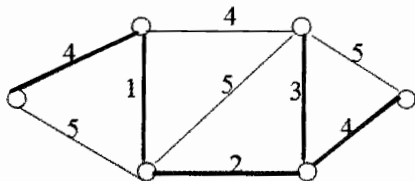


图 9-19 最小生成树示意图

所以电话线总长为： $1+2+3+4+4=14$ 。

试题 9 答案

(9) B

试题 10 分析

针对实际问题建立的数学模型往往是近似的，往往忽略了许多复杂因素。这种模型能否解决实际问题还需要检验。检验的方法有多种。

利用实际案例数据对模型进行检验是很常见的。将模型作为一个黑盒，通过案例数据的输入，检查其输出是否合理。这是应用人员常用的方法。

有时可以请专家来分析模型是否合理。经验丰富的专家一般会根据模型自身的逻辑，再结合实际情况，分析是否会出现矛盾或问题。

有时很难用实际案例或聘请专家来检验模型，例如，试验或实验的代价太大，难以取得实际案例，有的项目技术比较新，缺乏有经验的专家。这时，如果能利用计算机来模拟实际问题，再在计算机上检验该数学模型，这往往是一种有效的办法。例如，对某种核辐射防护建立的数学模型，采用计算机模拟方法来检验就十分有效。

企业负责人需要提供一切必要的支持来解决问题。至于解决过程中采用的技术问题，则需要由技术人员研究决定。企业负责人只需要听取汇报，从宏观上认可就可以，不需要理解其中的技术细节。

试题 10 答案

(10) D

试题 11 分析

根据题意，该地区冰箱品牌 A 与 B 每月占有率的变化描述为常数转移矩阵 P 。不管初始时刻这两种品牌的市场占有率（以概率向量来描述）如何，最终将稳定到概率向量 Z ，而且有关系式 $ZP=Z$ 。这表明， Z 的下一时刻仍然是 Z 。

设 $Z=(Z_1, Z_2)$ ，其中 $Z_1 \geq 0$ ， $Z_2 \geq 0$ ， $Z_1+Z_2=1$ ，从 $ZP=Z$ 可以列出方程：

$$0.8Z_1+0.4Z_2=Z_1$$

$$0.2Z_1+0.6Z_2=Z_2$$

根据上述条件，求解该方程，得到 $Z_1=2/3$ ， $Z_2=1/3$ 。

因此，冰箱品牌 A 与 B 在该地区最终将逐步稳定到市场占有率 $(2/3, 1/3)$ 。品牌 A 将占有 $2/3$ 的市场，品牌 B 将占有 $1/3$ 的市场。

试题 11 答案

(11) D

试题 12 分析

本题主要考查数学建模的基本过程，在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。模型分析工作主要包括模型的合理性分析、模型的误差分析和参数的灵敏性分析等，一般不包括模型的先进性分析。

试题 12 答案

(12) C

试题 13 分析

本题是架构考试中常见的一类计算题。该题解题关系是需要将图中节点的输入/输出流量调整平衡，因为只有输入/输出流量平衡才能表现出真实的运量。

如图 9-20 所示，对于节点 E，他的输出运力为 15，而所有输入运力之和为 14，则 E 的最大真实运力，只能达到 14，所以将 E 的输出运力修改为 14。对于 D 节点，其输出运力和为 7，而输入运力为 8，则需要平衡为 7。节点 B 也需要调，但情况比较复杂，我们需要综合分析 B 的输出运力与 C 的输出运力，分析可知，当 B 到 C 的运力调整为 1 时，既能达到节点运力的平衡，又能使运力最大，所以应调整为 1。当完成这些调整之后，可轻易得出结论，最大运力为 22。

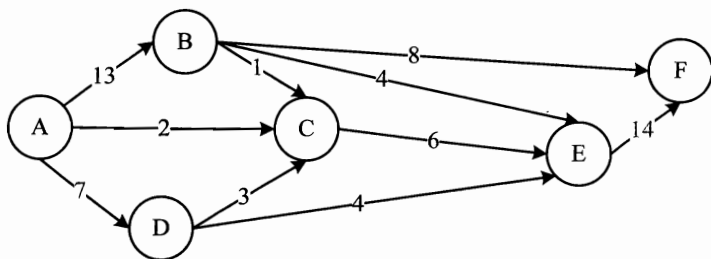


图 9-20 公路网运力调整示意图

试题 13 答案

(13) C

10

第 10 章

知识产权与标准化

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

- 标准化意识，标准化的发展，标准的生命周期。
- 国际标准、美国标准、国家标准、行业标准、地方标准、企业标准。
- 代码标准、文件格式标准、安全标准、软件开发规范和文档标准。
- 标准化机构。
- 知识产权。

10.1 考点突破

从历年的考试情况来看，本章所涉及的法律法规包括：著作权法、计算机软件保护条例、商标法、地理标志权、专利法、不正当竞争法、标准化法。这些法律法规从条款上来看，非常多，想要把它们全部弄明白，需要花费大量的时间，但从另外一个维度来看，考试当中，主要考查的问题是保护期限、知识产权人的确定、侵权判定及标准的类型，所以本章将从该角度来进行分析讲解。

10.1.1 历年考试情况分析

在历年的考试试题中，有关知识产权与标准化知识的试题如表 10-1 所示。

表 10-1 知识产权与标准化知识试题分布表

题号	2009.11	2010.11	2011.11
66	著作权保护期限	标准类型	著作权侵权问题
67	著作权法	商标法	商业秘密权
68	侵权判定	侵权判定	职务作品

按照知识点进行总结和归类的试题分布情况如表 10-2 所示。

表 10-2 知识产权与标准化知识归类表

知识点	2009.11	2010.11	2011.11
保护期限	1	0	0
知识产权人确定	0	0	1
侵权判定	2	1	1
标准化	0	1	0
其他	0	1	1
合计	3	3	3

从表 10-2 中可以看出,知识产权与标准化知识方面的内容在历年的考试中分值非常稳定,一直稳定维持在 3 分。所占分数比例的趋势如图 10-1 所示。

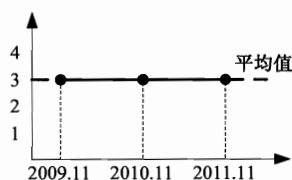


图 10-1 知识产权与标准化知识历年试题比例趋势图

10.1.2 保护期限

知识产权的保护是有时间性的,表 10-3 是对各种知识产权的保护期限的总结。

表 10-3 知识产权保护期限

客体类型	权利类型	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,不受保护
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	作者终生及其死亡后的 50 年(第 50 年的 12 月 31 日)。对于合作开发的,则以最后死亡的作者为准
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	著作权的保护期为 50 年(首次发表后的第 50 年的 12 月 31 日),若 50 年内未发表的,不予保护
注册商标		有效期是 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

对于前 4 项可以这样记忆：软件产品是作品的一种，公民拥有的为死后 50 年，单位拥有的为发表后 50 年；但公民的署名权、修改权（对于非软件作品还有保护作品完整权）是永远保护的。

10.1.3 知识产权人确定

通常情况下，谁拥有知识产权是显而易见的，但一旦涉及是在职完成、合作完成、委托完成、自然转让（公民知识产权人死亡后，在保护期内的知识产权根据继承法确定归属；单位知识产权人倒闭后，由承担其义务的组织拥有其知识产权）、专利及商标申请时都会有一些易混淆的点。表 10-4 对这些情况进行了总结。

表 10-4 知识产权人确定方法

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
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 就相同主题的发明做改进，可以获得国内优先权	

除此之外，还有一个小知识点：如果遇到作者不明的情况，那么作品原件的所有人可以行使除署名权以外的著作权，直到作者身份明确。

10.1.4 侵权判定

侵权判定是一个常考的知识点，在应对此类题型时，一方面靠常识判断，另一方面就得依据一些法律法规的条款了。下面将总结一些具有代表性的条款供大家参考。

1. 作品是否受保护

- 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。
- 口述作品（包括即兴的演说、授课、法庭辩论等以口头语言形式表现的作品）、摄影作品、示意图受著作权法保护。
- 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。
- 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，及其官方正式译文不受著作权法保护。

- 时事新闻不受著作权法保护。
- 历法、通用数表、通用表格和公式不受著作权法保护。

2. 侵权

- 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。
- 未经软件著作权人许可，修改、翻译其软件的。
- 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。

3. 不侵权

- 个人学习、研究或者欣赏；
- 适当引用，如写书时引用其他书籍适量内容，评论时引用，各种形式的新闻报道中引用；
- 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。

10.1.5 标准化

1. 标准

标准：对重复性的事务和概念所做的统一规定。以科学、技术和实践经验的综合成果为基础，经有关方面协商一致，由一个公认机构批准，以特定形式发布，作为共同遵守的准则和依据。

2. 标准化

标准化：是指在经济、技术、科学及管理等社会实践中，对重复性事物的概念通过制定、发布和实施标准达到统一，以获得最佳秩序和社会效益的活动。它是一门综合性学科，具有综合性、政策性和统一性的特点。

3. 标准化法

标准化法：确定了标准体系制和标准化管理体制，规定了制定标准的对象与原则，以及实施标准的要求，明确了违法行为的法律责任和处罚办法。

4. 标准化的过程

标准化的过程：一般包括标准产生（调查、研究、形成草案、批准发布）、标准实施（宣传、普及、监督、咨询）和标准更新（复审、废止或修订）3个子过程。

5. 标准的分类

按适用范围分：国际标准（ISO、IEC 等国际标准化组织）、国家标准（GB—中国、ANSI—美国、DIN—德国、BS—英国、JIS—日本、SIS—瑞典、NF—法国、SNV—瑞士、UNI—意大利、TOCTP—俄罗斯）、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（如中国军用标准、MIT-S—美国军用标准）、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。

按标准的性质分：技术标准、管理标准、工作标准。

按标准的作用分：基础标准（即为其他标准的基础）、产品标准、方法标准、安全标准、卫生标准、环境保护标准、服务标准……

按法律约束性分：强制标准（必须执行，有法律要求）、推荐标准。

6. 标准的编号

国际、国外标准代号及编号：虽然国际、国外标准代号形式各异，但基本结构是相同的：标准代号+专业类号+顺序号+年代号。标准代号通常使用缩写字母，专业类号有字母、数字、字母数字混合3种形式；顺序号和年代号与我国编码相似。

我国标准代号及编号：国家标准的代号由大写汉语拼音组成，强制性国家标准代号为 GB（国际实物标准代号为 GSB），推荐性国家标准代号为 GB/T。行业标准代号由汉语拼音大写字母组成（如航天 QJ，电子 SJ，机械 JB、金融 JR、军用标准 GJB），加上“/T”为推荐性标准，编号由主管部门申请，国务院标准化行政主管部门审查确定。地方标准代号由大写汉语拼音 DB 加上省级行政区划代码的前两位，加上“/T”一样说明为推荐性标准。企业标准代号由“Q/”加上企业代号组成。在考试时，国家标准、地方标准、企业标准都很容易识别，但行业标准较难识别，因为行业非常多，我们不可能将所有的行业代号都记下，所以行业标准通常用排除法识别（即不是国家标准、地方标准、企业标准则很有可能就是行业标准）。

另外还有一个小知识点需要注意，根据我国《国家标准管理办法》的规定：国家标准实施后，应当根据科学技术的发展和经济建设的需要，由该国家标准的主管部门组织有关单位适时进行复审，复审周期一般不超过5年。

10.2 典型试题分析

试题1

中国某企业与某日本公司进行技术合作，合同约定使用一项日本专利（获得批准并在有效期内），但该项技术未在中国申请专利，依照该专利生产的产品在（1）销售，中国企业需要向日本公司支付这项日本专利的许可使用费。

- (1) A. 中国
B. 日本
C. 其他国家和地区
D. 任何国家和地区

试题分析

知识产权具有严格的地域性特点,即各国主管机关依照其本国法律授予的知识产权,只能在本国领域内受法律保护。例如,中国专利局授予的专利权或中国商标局核准的商标专用权只能在中国领域内受保护,其他国家不给予保护,外国人在我国领域外使用中国专利局授权的发明专利,不侵犯我国专利权。我国法律对外国人的作品并不是都给予保护的,只保护共同参加国际条约国家的公民作品。同样,公约的其他成员国也按照公约规定,对我国公民和法人的作品给予保护。按照两国的双边协定,相互给予对方国民的作品保护。

依照该专利生产的产品在中国或其他国家和地区销售,中国企业不需要向日本公司支付这件日本专利的许可使用费。这是因为该日本公司未在中国或其他国家和地区申请该专利(试题中未指出。也就是说,该日本公司未在日本以外的国家和地区申请专利并获得批准,依照这件专利生产的产品在这些国家和地区销售,得不到这些国家和地区的专利保护),不受中国或其他国家和地区专利法的保护。因此,依照该专利生产的产品在中国或其他国家和地区销售,中国企业不需要向日本公司支付这项日本专利的许可使用费。

试题答案

- (1) B

试题 2

希赛公司所生产的 U 盘使用了其品牌产品“移动硬盘”的注册商标,那么,该公司(2)的商标专用权。

- (2) A. 享有了其 U 盘
B. 不享有其 U 盘
C. 享有移动硬盘和 U 盘
D. 不享有移动硬盘和 U 盘

试题分析

《商标法》第 2 章第 12 条规定:“同一申请人在不同类别的商品上使用同一商标的,应当按商品分类表提出注册申请”。第 13 条规定:“注册商标需要在同一类的其他商品上使用的,应当另行提出注册申请”。该公司在其生产的 U 盘上使用其他商品的注册商标,未另行提出商品注册申请,未经商标局核准注册,所以该公司不享有其 U 盘的商标专用权。

试题答案

- (2) B

试题 3

2005 年 5 月 4 日,张某向中国专利局提出发明专利申请;其后,张某对该发明作了改进,于 2006 年 5 月 4 日又就其改进发明自中国专利局提出申请时,可享有(3)。

- (3) A. 两项专利权
B. 优先使用权
C. 国际优先权
D. 国内优先权

试题分析

《中华人民共和国专利法》第二十九条规定：

申请人自发明或者实用新型在外国第一次提出专利申请之日起十二个月内，或者自外观设计在外国第一次提出专利申请之日起六个月内，又在中国就相同主题提出专利申请的，依照该外国同中国签订的协议或者共同参加的国际条约，或者依照相互承认优先权的原则，可以享有优先权。

申请人自发明或者实用新型在中国第一次提出专利申请之日起十二个月内，又向国务院专利行政部门就相同主题提出专利申请的，可以享有优先权。

《中华人民共和国专利法实施细则》第三十三条规定：

申请人在一件专利申请中可以要求一项或者多项优先权；要求多项优先权的，该申请的优先权期限从最早的优先权日起计算。

申请人要求本国优先权，在先申请是发明专利申请的，可以就相同主题提出发明或者实用新型专利申请；在先申请是实用新型专利申请的，可以就相同主题提出实用新型或者发明专利申请。但是，提出后一申请时，在先申请的主题有下列情形之一的，不得作为要求本国优先权的基础。

- (1) 已经要求外国优先权或者本国优先权的；
- (2) 已经被授予专利权的；
- (3) 属于按照规定提出的分案申请的。

申请人要求本国优先权的，其在先申请自后一申请提出之日起即视为撤回。

试题答案

- (3) D

试题 4

某商标注册人委托某印刷厂印制商标标识 20 万套，而印刷厂印制 25 万套，该印刷厂多印制 5 万套商标标识的做法属 (4) 行为。

- (4) A. 侵犯注册商标专用权
B. 伪造他人注册商标标识
C. 合法扩大业务范围
D. 不侵犯注册商标专用权

试题分析

《中华人民共和国商标法》第五十二条规定：

有下列行为之一的，均属侵犯注册商标专用权。

① 未经商标注册人的许可，在同一种商品或者类似商品上使用与其注册商标相同或者近似的商标的；

- ② 销售侵犯注册商标专用权的商品的。
- ③ 伪造、擅自制造他人注册商标标识或者销售伪造、擅自制造的注册商标标识的。
- ④ 未经商标注册人同意，更换其注册商标并将该更换商标的商品又投入市场的。
- ⑤ 给他人的注册商标专用权造成其他损害的。

在本题中，由于某商标注册人委托某印刷厂印制商标标识 20 万套，而印刷厂印制 25 万套，该印刷厂多印制的 5 万套属于“擅自制造他人注册商标标识”行为，因此，侵犯注册商标专用权。

试题答案

(4) A

试题 5

某软件企业根据市场需求，组织开发出一种新应用软件。为确保企业的市场竞争力，该企业对其所有员工进行了保密的约束。但某开发人员将该应用软件的程序设计技巧和算法通过论文向社会发表，使得软件企业丧失了 (5)。

(5) A. 发行权 B. 软件著作权 C. 商业秘密权 D. 专利权

试题分析

商业秘密是指不为公众所知悉、能为权利人带来经济利益、具有实用性并经权利人采取保密措施的技术信息和经营信息。根据这个定义，商业秘密应具备以下 4 个法律特征。

① 不为公众所知悉。这是讲商业秘密具有秘密性，是认定商业秘密最基本的要件和最主要的法律特征。商业秘密的技术信息和经营信息，在企业内部只能由参与工作的少数人知悉，这种信息不能从公开渠道获得。如果众所周知，那就不能称之为商业秘密。

② 能为权利人带来经济利益。这是讲商业秘密具有价值性，是认定商业秘密的主要要件，也是体现企业保护商业秘密的内在原因。一项商业秘密如果不能给企业带来经济价值，也就失去保护的意义。

③ 具有实用性。商业秘密区别于理论成果，具有现实的或潜在的使用价值。商业秘密在其权利人手里能应用，被人窃取后别人也能应用。这是认定侵犯商业秘密违法行为的一个重要条件。

④ 采取了保密措施。这是认定商业秘密最关键的条件。权利人对其所拥有的商业秘密应采取相应合理的保密措施，使其他人不采用非法手段就不能得到。如果权利人对拥有的商业秘密没有采取保密措施，任何人几乎随意可以得到，那么，就无法认定该秘密是否是权利人的商业秘密。

从题目说明来看，属于商业秘密泄露。

试题答案

(5) C

试题 6

在我国，实用新型和外观设计专利申请 (6)。

- (6) A. 须经过实质审查后授权 B. 经初审合格后即授权
C. 递交申请后即可授权 D. 经过形式审查和实质审查后才可授权

试题分析

根据我国专利法第四十条规定“实用新型和外观设计专利申请经初步审查没有发现驳回理由的，专利局应当作出授予实用新型专利权或者外观设计专利权的决定，发给相应的专利证书，并予以登记和公告。”实用新型和外观设计专利申请经初审合格后即授权。

试题答案

- (6) B

试题 7

李某购买了一张有注册商标应用软件的光盘，则李某享有____(7)____。

- (7) A. 注册商标专用权 B. 该光盘的所有权
C. 该软件的著作权 D. 该软件的复制权

试题分析

显然，李某只能拥有他购买的光盘的所有权，以及光盘中软件的使用权，而该软件的其他权利都不属于李某。

试题答案

- (7) B

试题 8

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____(8)____。

- (8) 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

试题分析

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公之于众。

试题答案

- (8) B

试题 9

为维护软件开发者的合法权益，解决自动保护可能产生的权属纠纷，我国实行了____(9)____。

- (9) A. 计算机软件著作权登记制度 B. 计算机软件标记制度
C. 计算机软件商标制度 D. 计算机软件产权审查制度

试题分析

为维护软件开发者的合法权益,解决自动保护可能产生的权属纠纷,我国实行了计算机软件著作权登记制度。

试题答案

(9) A

试题 10

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(10)。

- (10) 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

试题分析

《计算机软件保护条例》第二十八条规定:软件复制品的出版者、制作者不能证明其出版、制作有合法授权的,或者软件复制品的发行者、出租者不能证明其发行、出租的复制品有合法来源的,应当承担法律责任。

《计算机软件保护条例》第三十条规定:软件的复制品持有人不知道也没有合理理由应当知道该软件是侵权复制品的,不承担赔偿责任;但是,应当停止使用、销毁该侵权复制品。如果停止使用并销毁该侵权复制品将给复制品使用人造成重大损失的,复制品使用人可以在向软件著作权人支付合理费用后继续使用。

试题答案

(10) C

试题 11

我国的《著作权法》对一般文字作品的保护期是作者有生之年和去世后 50 年,德国的《版权法》对一般文字作品的保护期是作者有生之年和去世后 70 年。假如某德国作者已去世 60 年,以下说法中正确的是(11)。

- (11) A. 我国 M 出版社拟在我国翻译出版该作品,需要征得德国作者继承人的许可方可在我国出版发行
B. 我国 M 出版社拟在我国翻译出版该作品,不需要征得德国作者继承人的许可,就可在我国出版发行
C. 我国 M 出版社未征得德国作者继承人的许可,将该翻译作品销售到德国,不构成侵权
D. 我国 M 出版社未征得德国作者继承人的许可,将该翻译作品在我国销售,构成侵权

试题分析

本题考查知识产权方面的基础知识。按照《伯尔尼公约》的规定,一个成员国给

予其他成员国作品的版权保护期，应按照该成员国版权法的规定。依据我国著作权法的规定，该德国作者的作品已经超过法定版权保护期，不再受到版权保护。因此，出版社不需要征得德国作者继承人的许可，即可在我国出版发行该德国作者的作品。如果将该翻译出版作品未征得德国作者继承人的许可销售到德国，已构成侵权。这是因为德国的《版权法》规定作品的版权保护期是作者有生之年和去世后 70 年，作者去世 60 年，作品的保护期尚未超过，所以我国出版社若将该翻译出版作品未征得德国作者继承人的许可销售到德国，则构成侵权。

我国的《著作权法》对一般文字作品的保护期是作者有生之年和去世后 50 年，该作者已去世 60 年，超过了我国《著作权法》对一般文字作品的保护期，在我国也不再受著作权保护。所以我国 M 出版社不需要征得德国作者继承人的许可，即可在我国出版发行该德国作者的作品。

试题答案

(11) B

试题 12

(12) 不属于我国著作权法所保护的内容。

- (12) A. 为保护其软件著作权而采取的技术措施
B. 软件权利电子信息
C. 通过信息网络传播的软件
D. 采用反编译技术获得的软件

试题分析

本题考查知识产权方面的基础知识。我国著作权法采取列举方法，规定了侵权行为的表现形式。其中包括未经著作权人许可，复制、发行、表演、放映、广播、汇编、通过信息网络向公众传播其作品的行为；未经著作权人或者与著作权有关的权利人许可，故意避开或者破坏权利人为其作品、录音录像制品等采取的保护著作权或者与著作权有关的权利的技术措施的行为；未经著作权人或者与著作权有关的权利人许可，故意删除或者改变作品、录音录像制品等的权利管理电子信息的行为。这三种表现形式分别涵盖了试题中 A、B、C 选项。虽然利用反向编译技术、净室技术和反向工程技术等获得他人软件技术构思、技术方案并直接用于其软件产品中的行为是一种“复制”软件技术构思、技术方案的行为，但是对于这些行为在法律上不会受到制止。目前，我国对软件实施反编译是否合法还没有相应的法律规定。

由于著作权不保护思想，软件开发设计人员对体现在软件中的创造性的构思和技术方案不能得到保护。任何人都可以利用反向编译技术、净室技术和反向工程技术等获得他人软件所使用的思路、原理、结构、算法、处理过程和运行方法等设计要素，直接用于自己的软件产品中，这在著作权法上并不构成侵权。

试题答案

(12) D

试题 13

王某原是 X 公司的项目经理，在 X 公司任职期间主持开发了某软件，但未与 X 公司签定劳动合同及相应的保密协议。X 公司对该软件进行了软件著作权登记并获准。王某随后离职并将其在 X 公司任职期间掌握的该软件技术信息、客户需求及部分源程序等秘密信息提供给另一软件公司。王某的行为____(13)_____。

- (13) A. 既侵犯了科技公司的商业秘密权，又侵犯了科技公司的软件著作权
B. 既未侵犯科技公司的商业秘密权，又未侵犯科技公司的软件著作权
C. 侵犯了科技公司的商业秘密权
D. 侵犯了科技公司的软件著作权

试题分析

本题考查知识产权方面的基础知识，涉及著作权和商业秘密权的相关概念。王某作为公司的职员，在任职期间主存开发的软件为职务软件，公司对该软件享有软件著作权。王某将该软件源程序擅自提供给其他公司的行为已构成对公司软件著作权的侵犯。王某的行为将使得另一软件公司很快就会开发出类似的产品，在市场上与科技公司竞争，这样无疑会损害科技公司的利益。软件商业秘密包括软件技术秘密，如源程序、设计方法、技术方案、功能规划、开发情况和测试结果等；软件经营秘密，如经营方法、产销策略、客户情报（客户名单、客户需求）和软件市场分析等。商业秘密受到法律保护的依据是必须具备构成商业秘密的 3 个条件，即不为公众所知悉、具有实用性、采取了保密措施，缺少 3 个条件之一都会造成商业秘密丧失法律保护。公司未与王某签定劳动合同及相应的保密协议，可以认为科技公司主观上没有保守商业秘密的意愿，客观上没有采取相应的保密措施，那么公司的软件技术秘密和软件经营秘密就不具有保密性。所以，不认为王某侵犯了公司的商业秘密权。

试题答案

(13) D

试题 14

____(14)_____为推荐性地方标准的代号。

- (14) A. SJ/T B. Q/T11 C. GB/T D. DB11/T

试题分析

地方性标准代号有一个显著的特征——以字母“DB”开头，所以本题应选 D。

试题答案

(14) D

试题 15

以 GJB 冠名的标准属于____(15)_____。PSD、PAD 等程序构造的图形表示属于____(16)_____。

- (15) A. 国际标准 B. 国家标准 C. 行业标准 D. 企业规范

- (16) A. 基础标准 B. 开发标准 C. 文档标准 D. 管理标准

试题分析

根据《国家标准管理办法》第四条规定,国家标准的代号由大写汉语拼音字母构成,强制性国家标准的代号为“GB”,推荐性国家标准的代号为“GB/T”。GJB 是我国国家军用标准,属于行业标准。但是,要注意的是,GSB(国家实物标准)却是国家标准。

图形符号、箭头表示等都属于基础标准,如果需要,可以用在各种具体的标准中。

试题答案

- (15) C (16) A

10.3 实战练习题

● 甲公司将其开发的商业软件著作权经约定合法转让给乙公司,随后甲公司自行对原软件进行了改进,形成新版本后进行销售。那么,甲公司的行为___(1)___。

- (1) A. 不构成侵权,因为对原软件改进后形成了新版本
B. 不构成侵权,因为甲公司享有原软件的使用权
C. 不构成侵权,因为对原软件增加了新的功能
D. 构成侵权,因为甲公司不再享有原软件的使用权

● 根据知识产权法规的有关规定,下列选项中正确的说法是___(2)___。

- (2) 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___(3)___。

- (3) A. 甲公司 B. 乙公司 C. 程序员 D. 乙公司与程序员共有

● ___(4)___这一计算机软件著作权权利是不可以转让的。

- (4) A. 发行权 B. 复制权 C. 署名权 D. 信息网络传播权

● 甲公司开发的通信软件,使用“点波”牌商标,商标没有注册。2007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“点波”牌商标,并于2007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,___(5)___能获准“点波”牌商标注册。

- (5) A. 甲公司 B. 乙公司 C. 甲、乙公司都 D. 甲、乙公司都不

● 依据《计算机软件保护条例》,对软件的保护包括___(6)___。

- (6) 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

● 张某原是某软件公司的系统分析师,一直从事计算机网络端口优化处理的研发工作。2007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__(7)__所有。

- (7) A. 软件公司 B. 张某 C. 张某和软件公司共同 D. 社会公众

● 我国标准分为强制性标准和推荐性标准,标准的编号由标准代号、标准发布顺序号和标准发布年代号构成,__(8)__为推荐性行业标准的代号。

- (8) A. DB11/T B. Q/T11 C. GB/T D. SJ/T

● 以 ANSI 冠名的标准属于__(9)__。

- (9) A. 国家标准 B. 国际标准 C. 行业标准 D. 项目规范

● GJB 473—88(军用软件开发规范)是__(10)__。

- (10) A. 国家标准 B. 行业标准 C. 地方标准 D. 企业标准

● 《GB 8567—88 计算机软件产品开发文件编制指南》是__(11)__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。

- (11) 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

● 某软件企业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“昆仑”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__(12)__的保护。

- (12) 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__(13)__。

- (13) 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（14）。

- (14) 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

10.4 练习题解析

试题 1 分析

从题目描述来看，甲公司已经把著作权转让给了乙公司，此时甲不再拥有该软件的著作权，但甲公司却“对原软件作品提高和改善”，这属于“未经软件著作权人许可，修改、翻译其软件的”行为，侵犯了乙公司的权利，所以构成侵权。

试题 1 答案

(1) D

试题 2 分析

参看“表 10-3 知识产权保护期限”可知商业秘密无确定的保护期限。

试题 2 答案

(2) D

试题 3 分析

对于程序员在非职务期间创作的计算机程序，其著作权属于某项软件作品的开发单位，还是从事直接创作开发软件作品的个人，可按照《计算机软件保护条例》第十三条规定的三条标准确定。

- ① 所开发的软件作品不是执行其本职工作的结果。
- ② 开发的软件作品与开发者在单位中从事的工作内容无直接联系。
- ③ 开发的软件作品未使用单位的物质技术条件。

全部符合上述 3 条标准的，所开发软件的著作权即为雇员个人享有。雇员进行本职工作以外的软件开发创作，必须同时符合上述 3 个条件，才能算是非职务软件作品，雇员个人才享有软件著作权。常有软件开发符合前两个条件，但使用了单位的技术情报资料、计算机设备等物质技术条件的情况。处理此种情况较好的方法是对该软件著作权的归属应当由雇员双方协商确定，如对于公民在非职务期间利用单位物质条件创作的与单位业务范围无关的计算机程序，其著作权属于创作程序的作者，但作者许可第三人使用软件时，应当支付单位合理的物质条件使用费，如计算机机时费等。若协商不能解决，只能按上述 3 条标准作出界定。本试题中已明确，程序员利用了乙公司的物质和技术条件，其开发的应用程序的著作权属于乙公司。

试题 3 答案

(3) B

试题 4 分析

在计算机软件著作权中，署名权是不能转让的。

试题 4 答案

(4) C

试题 5 分析

根据商标法的规定，商标注册采取的是“谁先申请谁先获得”的方法。两个或者两个以上的商标注册申请人，在同一种商品或者类似商品上，以相同或者近似的商标申请注册的，初步审定并公告申请在先的商标；同一天申请的，初步审定并公告使用在先的商标，驳回其他人的申请，不予公告。

试题 5 答案

(5) B

试题 6 分析

《计算机软件保护条例》所保护的软件范围为计算机程序及其文档，不包括开发该软件的所用思想。

试题 6 答案

(6) C

试题 7 分析

根据专利法实施细则，职务发明创造是指：

- ① 在本职工作中做出的发明创造；
- ② 履行本单位交付的本职工作之外的任务所做出的发明创造；
- ③ 辞职、退休或者调动工作后 1 年内做出的，与其在原单位承担的本职工作或者原单位分配的任务有关的发明创造。

在本题中，因为张某是在退休后半年内研发出网络端口优化处理程序，所以，该项发明创造应属于张某原来所在的软件公司所有。

试题 7 答案

(7) A

试题 8 分析

标准的层次有国际标准（如 ISO）、国家标准（中国标准以 GB 开头，英国标准以 BS 开头，美国标准以 ANSI 开头）、地方标准（以 DB 开头，后接地方编号）、行业标准（如 IEEE）、企业标准（以 Q 开头）。标准的类型有推荐性标准和强制性标准。推荐性标准用“/T”表示。

试题 8 答案

(8) D

试题 9 分析

ANSI 为美国的国家标准，所以应选 A。

试题 9 答案

(9) A

试题 10 分析

GJB 是中国军用标准，军用标准属于行业标准。

试题 10 答案

(10) B

试题 11 分析

我国国家标准的代号由大写汉字拼音字母构成，强制性国家标准代号为 GB，推荐性国家标准的代号为 GB/T。

强制性标准是国家技术法规，具有法律约束性。其范围限制在国家安全、防止欺诈行为、保护人身健康与安全等方面。根据《标准化法》的规定，企业和有关部门对涉及其经营、生产、服务、管理有关的强制性标准都必须严格执行，任何单位和个人不得擅自更改或降低标准。对违反强制性标准而造成不良后果以至重大事故者，由法律、行政法规规定的行政主管部门依法根据情节轻重给予行政处罚，直至由司法机关追究刑事责任。

推荐性标准是自愿采用的标准。这类标准是指导性标准，不具有强制性，一般是为了通用或反复使用的目的，为产品或相关生产方法提供规则、指南或特性的文件。任何单位均有权决定是否采用，违犯这类标准，不构成经济或法律方面的责任。由于推荐性标准是协调一致的文件，不受政府和社会团体的利益干预，能更科学地规定特性或指导生产，我国《标准化法》鼓励企业积极采用推荐性标准。应当指出的是，推荐性标准一经接受并采用，或由各方商定后同意纳入经济合同中，就成为各方必须共同遵守的技术依据，具有法律上的约束性。

由行业机构、学术团体或国防机构制定，并适用于某个业务领域的标准。行业标准代号由国务院各有关行政主管部门提出其所管理的行业标准范围的申请报告，国务院标准化行政主管部门审查确定并正式公布该行业标准代号。已正式公布的行业代号：QJ（航天）、SJ（电子）、JB（机械）、JR（金融）等，暂无软件行业。行业标准代号由汉字拼音大写字母组成，再加上斜线 T 组成推荐性行业标准（如 SJ / T）。

试题 11 答案

(11) A

试题 12 分析

该企业在软件开发完成后就取得了著作权，该企业向国家专利局申请发明专利并

获得了专利权，此时该软件就可以同时受到著作权法、专利法的保护。

在《刑法》中明确规定了严重侵犯知识产权的行为应当承担刑事责任。在《计算机软件保护条例》第24条中，规定对侵权行为触犯刑律的，依照刑法关于侵犯著作权罪、销售侵权复制品罪的规定，依法追究刑事责任。例如，《刑法》中第213条规定，未经注册商标所有人许可，在同一种商品上使用与其注册商标相同的商标，情节严重的，处3年以下有期徒刑或者拘役，并处或者单处罚金；情节特别严重的，处3年以上7年以下有期徒刑，并处罚金；216条规定对假冒他人专利，情节严重的，处3年以下有期徒刑或者拘役，并处或者单处罚金；217条规定对以营利为目的，未经著作权人许可，复制发行其计算机软件的，违法所得数额较大或者有其他严重情节的，处3年以下有期徒刑或者拘役，并处或者单处罚金；违法所得数额巨大或者有其他特别严重情节的，处3年以上7年以下有期徒刑，并处罚金；218条规定对以营利为目的，销售明知是本法第217条规定的侵权复制品，违法所得数额巨大的，处3年以下有期徒刑或者拘役，并处或者单处罚金。所以，该软件产品能够得到刑法的保护。

商标权是商标所有人依法对其商标所享有的专有使用权。在我国，商标权的取得实行的是注册原则，即商标所有人只有依法将自己的商标注册后，商标注册人才能取得商标权，其商标才能得到法律的保护。该企业虽然对其软件产品已经冠以商品专用标识，但未进行商标注册，没有取得商标专用权，此时该软件产品就不能得到商标法的保护。

商标法虽然不能提供针对计算机软件的实质内容与表达的直接保护，却可以为软件提供商业化的保护。计算机软件的权利人可以通过商标法来实现对其所属软件的保护。首先，软件的权利人可以针对软件产品申请注册商标，并通过标注在其产品包装上的商标来表明其身份及商誉；其次，软件的权利人还可以通过技术措施在软件中设置其特有的商业标记，例如，通过技术手段将与注册商标相同的文字商标、图形商标或者其二者的结合出现在软件的界面上。这些措施不仅可以防止他人对商品化了的计算机软件实施侵权，也可以在遭受侵权后，利用商标法有效地打击侵权行为。尤其是当软件被盗版的时候，这些盗版软件经常是不标明商标、产地等，甚至是假冒商标，软件权利人可以依据商标法请求工商行政管理部门进行查处，在实践中，行政手段通常要比诉讼程序更简便、快捷，达到保护计算机软件的目的。当然，商标法还对侵犯软件商标权的行为应承担的民事及刑事责任进行了规定。

试题 12 答案

(12) C

试题 13 分析

张某开发的软件是在国际运输有限公司担任计算机系统管理员期间根据国际运输有限公司业务要求开发的“空运出口业务系统 V1.0”，即该软件是针对本职工作中明确指定的开发目标所开发的。根据《著作权法》第16条规定，公民为完成法人或者非法人单位工作任务所创作的作品是职务作品。认定作品为职务作品还是个人作品，应考虑两个前提条件：一是作者和所在单位存在劳动关系，二是作品的创作属于

作者应当履行的职责。职务作品分为一般职务作品和特殊的职务作品：一般职务作品的著作权由作者享有，单位或其他组织享有在其业务范围内优先使用的权利，期限为2年；特殊的职务作品，除署名权以外，著作权的其他权利由单位享有。所谓特殊职务作品，是指《著作权法》第16条第2款规定的两种情况：一是主要利用法人或者其他组织的物质技术条件创作，并由法人或者其他组织承担责任的工程设计、产品设计图、计算机软件、地图等科学技术作品；二是法律、法规规定或合同约定著作权由单位享有的职务作品。《计算机软件保护条例》也有类似的规定，在第十三条中规定了三种情况，一是针对本职工作中明确指定的开发目标所开发的软件；二是开发的软件是从事本职工作活动所预见的结果或者自然的结果；三是主要使用了法人或者其他组织的资金、专用设备、未公开的专门信息等物质技术条件所开发并由法人或者其他组织承担责任的软件。张某在公司任职期间利用公司的资金、设备和各种资料，且是从事本职工作活动所预见的结果。所以，其进行的软件开发行为是职务行为，其工作成果应由公司享有。因此，该软件的著作权应属于国际运输有限公司，但根据法律规定，张某享有署名权。

根据《计算机软件保护条例》第7条规定，软件登记机构发放的登记证明文件是登记事项的初步证明，只是证明登记主体享有软件著作权及订立许可合同、转让合同的重要的书面证据，并不是软件著作权产生的依据。该软件是张某针对本职工作中明确指定的开发目标所开发的，该软件的著作权应属于公司。明确真正的著作权人之后，软件著作权登记证书的证明力自然就消失了（只有审判机关才能确定登记证书的有效性。）。

为促进我国软件产业发展，增强我国软件产业的创新能力和竞争能力，1992年4月6日机械电子部发布了《计算机软件著作权登记办法》，鼓励软件登记并对登记的软件予以重点保护，而不是强制软件登记。软件登记可以分为软件著作权登记、软件著作权专有许可合同和转让合同的登记。软件著作权登记的申请人应当是该软件的著作权人，而软件著作权合同登记的申请人，应当是软件著作权专有许可合同和转让合同的当事人。如果未经软件著作权人许可登记其软件，或是将他人软件作为自己的软件登记的，或未经合作者许可、将与他人合作开发的软件作为自己单独完成的软件登记，这些行为都属于侵权行为，侵权人要承担法律责任。

试题 13 答案

(13) B

试题 14 分析

本题看似是考查著作权与商标权相关内容，但实际上是在考查一般争议处理的流程。对于任何争议基本上都是采取的先找主管行政管理部门进行仲裁，仲裁不成功再进行诉讼，而C选项的说法，刚好弄反了。

试题 14 答案

(14) C

11

第 11 章

系统配置与性能评价

根据考试大纲，本章要求考生掌握以下几个方面的知识点：

- 多层结构、分布式系统。
- 系统配置方法（双份、双重、热备份、容错、集群）。
- 性能计算（响应时间、吞吐量、TAT）。
- 性能设计（系统调整、Amdahl 解决方案、响应特性、负载均衡）。
- 性能指标（SPEC-Int、SPEC-Fp、TPC、Gibsonmix、响应时间）。
- 性能评估。
- 系统可靠性设计（容错技术、避错技术）。
- 系统可靠性指标与评估。

11.1 考点突破

从历年的考试情况来看，本章的考点主要集中于性能评估和系统可靠性设计，该方面的内容考查不仅限于综合知识考试，也有可能以案例分析及论文方式考查。

11.1.1 历年考试情况分析

在历年的考试试题中，有关系统配置与性能评价知识的试题如表 11-1 所示。

表 11-1 系统配置与性能评价知识试题分布表

题号	2009.11	2010.11	2011.11
16	基准程序测试	多处理器性能计算	负载均衡
17	计算机性能优化	系统性能评价	数据备份

按照知识点进行总结和归类的试题分布情况如表 11-2 所示。

表 11-2 系统配置与性能评价知识归类表

知识点	2009.11	2010.11	2011.11
系统性能评价	2	1	0
冗余技术	0	0	1
容错技术	0	0	1
系统可靠性计算	0	0	0
其他	0	1	0
合计	2	2	2

从表 11-2 中可以看出,系统配置与性能评价知识方面的内容在历年的考试中分值非常稳定,一直稳定维持在 2 分。所占分数比例的趋势如图 11-1 所示。



图 11-1 系统配置与性能评价知识历年试题比例趋势图

11.1.2 系统性能评价

无论是生产厂商还是用户,都需要某种方法来衡量计算机系统的性能,但由于系统很复杂、体系结构和实现的策略多样,因此很难采用统一的标准去评测所有的计算机。

1. 常用方法

- 时钟频率:即主频(常听到的 CPU 主频 2.8GHz 等),计算机的时钟频率在一定程度上反映了机器速度,一般来讲,主频越高,速度越快。但是相同频率、不同体系结构的机器,其速度可能会相差很多倍,所以该方法一般只用于同体系结构机器之间的性能对比。
- 指令执行速度:在早期,我们经常使用每秒执行的加法指令(由于当时各种指令的速度大致相同或等比例)总数来作为衡量其性能的重要指标,其单位为 KIPS(千条指令每秒)、MIPS(百万条指令每秒)。
- 等效指令法:也叫做吉普森混合法或混合比例算法。随着计算机指令系统的发展,使用单种指令的 MIPS 值的局限性日益暴露,后来就出现了改进的吉普森混合指令速度法。它通过统计各类指令在程序中所占的比例,进行折算。

数据处理速率(PDR)法:因为在不同程序中,各类指令使用频率是不同的,所以固定比例方法存在着很大的局限性;而且数据长度与指令功能的强弱对解题的速度影响极大。同时这种方法也不能反映现代计算机中高速缓冲存储器、流水线、交叉存储等结构的影响。具有这种结构的计算机的性能不仅与指令的执行频率有关,而且也与指令的执行顺序和地址的分布有关。

PDR 法采用计算 PDR 值的方法来衡量机器性能,PDR 值越大,机器性能越好。PDR 与每条指令和每个操作数的平均位数,以及每条指令的平均运算速度有关。PDR

值主要对 CPU 和主存储器的速度进行度量,但不适合衡量机器的整体速度,因为它没有涉及 Cache、多功能部件等技术对性能的影响。PDR 主要是对 CPU 和主存数据处理速度进行计算而得出的,它允许并行处理和指令预取的功能,这时,所取的是指令执行的平均时间。带有 Cache 的计算机,因为存取速度加快,其 PDR 值也就相应提高。PDR 不能全面反映计算机的性能,但它曾是美国及巴黎统筹委员会用来限制计算机出口的系统性能指标估算方法。1991 年 9 月停止使用 PDR,取而代之的是 CTP (综合理论性能)。

- 核心程序法:把应用程序中用得最频繁的那部分核心程序作为评价计算机性能的标准程序,在不同机器上运行,测其执行时间,作为各类机器性能评价的依据。

2. 基准测试程序

前面提到的性能评价方法主要是针对 CPU (有时包括主存),它没有考虑诸如 I/O 结构、操作系统、编译程序的效率等系统性能的影响,因此难以准确评价计算机的实际上作能力。基准程序法是目前一致承认的测试性能的较好方法,有多种多样的基准程序,如主要测试整数性能的基准程序逻辑、测试浮点性能的基准程序等。

- 整数测试程序: Dhrystone 是一个用来测试编译器和 CPU 处理整数指令和控制功能有效性的基准测试程序。
- 浮点测试程序:在计算机科学和工程应用领域,浮点计算工作量占很大比例,因此有许多此类基准测试程序。
 - 理论峰值浮点速度: MELOPS,与处理器时钟周期、并行流水线功能部件数相关,是直接计算出来的理论值。
 - Linpack 基准测试程序:主要测试向量性能和高速缓存性能。
 - Whetstone 基准测试程序:综合性测试程序,除测试浮点操作外,还测试整数计算和功能调用等性能。
- SPEC 基准程序:是由几十家世界知名计算机大厂商支持的非盈利的合作组织,开发共同认可的标准基准程序。
- TPC 基准程序:是事务处理委员会编写的,共包括 TPC-A、TPC-B、TPC-C、TPC-D 和 TPC-E 五种,每一种都有一定的适用范围。

3. 综合理论性能法 (CTP)

CTP 是美国政府为限制较高性能计算机出口所设置的运算部件综合性能估算方法。CTP 以每秒百万次理论运算 MTOPS 表示,从 1991 年 9 月 1 日起启用。CTP 的估算方法为首先算出处理部件每一计算单元(如定点加法单元、定点乘法单元、浮点加单元、浮点乘法单元)的有效计算率 R,再按不同字长加以调整,得出该计算单元的理论性能 TP,所有组成该处理部件的计算单元 TP 的总和即为综合理论性能 CTP。

4. 阿姆达尔解决方案

阿姆达尔定律: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。

阿姆达尔定律定义了采用特定组件所取得的加速比。假设使用某种改进了的组件，系统的性能就会得到提高，加速比的计算公式如下：

$$R = \frac{T_p}{T_i}$$

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。阿姆达尔定律为计算某些情况下的加速比提供了一种便捷的方法。加速比主要取决于两个因素：

- 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。
- 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。

原来的系统使用了改进功能后，其执行时间等于未改进部分的执行时间加上改进部分的执行时间，即

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

改进后整个系统的加速比为

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

例如，在某计算机系统中，假设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，则根据 Amdahl 定律，这样做可以使整个系统的性能提高

$$R = \frac{1}{(1 - 0.5) + 0.5 / 10} = 1.818 \text{ 倍}$$

11.1.3 冗余技术

冗余是指在正常系统运行所需的基础上加上一定数量的资源，包括信息、时间、硬件和软件。冗余是容错技术的基础，通过冗余资源的加入，可以使系统的可靠性得到较大的提高。主要的冗余技术有结构冗余（硬件冗余和软件冗余）、信息冗余、时间冗余和冗余附加 4 种。

1. 结构冗余

结构冗余是常用的冗余技术，按其工作方式，可分为静态冗余、动态冗余和混合冗余 3 种。

(1) 静态冗余

静态冗余又称屏蔽冗余或被动冗余，常用的有三模冗余和多模冗余。静态冗余通过表决和比较来屏蔽系统中出现的错误。例如，三模冗余是对三个功能相同，但由不同的

人采用不同的方法开发出的模块的运行结果进行表决，以多数结果作为系统的最终结果。即如果模块中有一个出错，这个错误能够被其他模块的正确结果“屏蔽”。由于无须对错误进行特别的测试，也不必进行模块的切换就能实现容错，故称为静态容错。

(2) 动态冗余

动态冗余又称主动冗余，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顶替它并重新运行。各备用模块在待机时，可与主模块一样工作，也可不工作。前者叫做热备份系统（双重系统），后者叫做冷备份系统（双工系统、双份系统）。在热备份系统中，两套系统同时、同步运行，当联机子系统检测到错误时，退出服务进行检修，而由热备份子系统接替工作，备用模块在待机过程中其失效率为 0；处于冷备份的子系统平时停机或者运行与联机系统无关的运算，当联机子系统产生故障时，人工或自动进行切换，使冷备份系统成为联机系统。在运行冷备份时，不能保证从系统断点处精确地连续工作，因为备份机不能取得原来的机器上当前运行的全部数据。

(3) 混合冗余

混合冗余技术是将静态冗余和动态冗余结合起来，且取二者之长处。它先使用静态冗余中的故障屏蔽技术，使系统免受某些可以被屏蔽的故障的影响。而对那些无法屏蔽的故障则采用主动冗余中的故障检测、故障定位和故障恢复等技术，并且对系统可以作重新配置。因此，混合冗余的效果要大大优于静态冗余和动态冗余。然而，由于混合冗余既要有静态冗余的屏蔽功能，又要有动态冗余的各种检测和定位等功能，它的附加硬件的开销是相当大的，所以混合冗余的成本很高，仅在对可靠性要求极高的场合中采用。

2. 信息冗余

信息冗余是在实现正常功能所需要的信息外，再添加一些信息，以保证运行结果正确性的方法。例如，检错码和纠错码就是信息冗余的例子。这种冗余信息的添加方法是按照一组预定的规则进行的。符合添加规则而形成的带有冗余信息的字称为码字，而那些虽带有冗余信息但不符合添加规则的字则称为非码字。当系统出现故障时，可能会将码字变成非码字，于是在译码过程中会将引起非码字的故障检测出来，这就是检错码的基本思想。纠错码则不仅可以将错误检测出来，还能将由故障引起的非码字纠正成正确的码字。

由此可见，信息冗余的主要任务在于研究出一套理想的编码和译码技术来提高信息冗余的效率。编码技术中应用最广泛的是奇偶校验码、海明校验码和循环冗余校验码。

3. 时间冗余

时间冗余是以时间（即降低系统运行速度）为代价以减少硬件冗余和信息冗余的开销来达到提高可靠性的目的。在某些实际应用中，硬件冗余和信息冗余的成本、体积、功耗、重量等开销可能过高，而时间并不是太重要的因素时，可以使用时间冗余。时间冗余的基本概念是重复多次进行相同的计算，或称为重复执行（复执），以达到故障检测的目的。

实现时间冗余的方法很多,但是其基本思想不外乎是对相同的计算任务重复执行多次,然后将每次的运行结果存放起来再进行比较。若每次的结果相同,则认为无故障;若存在不同的结果,则说明检测到了故障。不过,这种方法往往只能检测到瞬时性故障而不宜检测永久性的故障。

4. 冗余附加

冗余附加是指为实现上述冗余技术所需的资源和技术,包括程序、指令、数据,以及存放和调用它们的空间等。

11.1.4 容错技术

在开发任何一个系统时,设计者都希望系统在交付以后能正常稳定的运行,不出现故障,但要绝对避免错误产生,或是在设计阶段找出所有的错误是不可能实现的。所以产生了容错技术,容错技术简单一点讲,就是当系统产生错误时能让系统仍然正常运行,或是将损失尽可能降低的技术。容错技术既应用于硬件领域又应用于软件领域。

1. 软件容错

软件容错的基本思想是从硬件容错中引申而来,利用软件设计的冗余和多样化来达到屏蔽错误的影响,提高系统可靠性的目的。软件容错的主要方法是提供足够的冗余信息和算法程序,使系统在实际运行时能够及时发现程序设计错误,采取补救措施,以提高系统可靠性,保证整个系统的正常运行。

软件容错技术主要有 N 版本程序设计、恢复块方法和防卫式程序设计等。除上述 3 种方法外,提高软件容错能力也可以从计算机平台环境、软件工程和构造异常处理模块等不同方面达到。此外,利用高级程序设计语言本身的容错能力,采取相应的策略,也是可行的办法。例如, C++ 语言中的 `try_except` 处理法和 `try_finally` 中止法等。

(1) N 版本程序设计

N 版本程序设计是一种静态的故障屏蔽技术,采用前向恢复的策略,如图 11-2 所示。

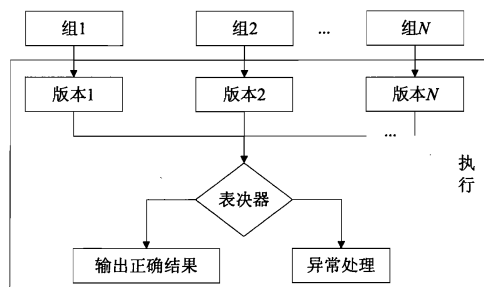


图 11-2 N 版本程序设计

N 版本程序的设计思想是用 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(小组)独立设计,使用不同的方法、不同的设计语言、不同的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器上相关错误的概率。

(2) 恢复块方法

恢复块方法是一种动态的故障屏蔽技术,采用后向恢复策略,如图 11-3 所示。

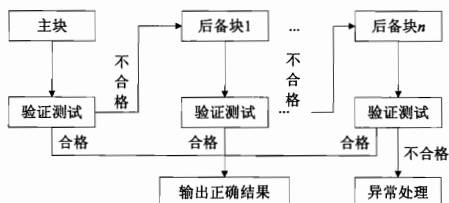


图 11-3 恢复块方法

恢复块方法提供具有相同功能的主块和几个后备块,一个块就是一个执行完整的程序段,主块首先投入运行,结束后进行验证测试,如果没有通过验证测试,系统经现场恢复后由后备块 1 运行。后备块 1 运行结束后也进行验证测试,如果没有通过验证测试,系统经现场恢复后由后备块 2 运行。重复这一过程,可以重复到耗尽所有的后备块,或者某个程序故障行为超出了预料,从而导致不可恢复的后果。

在程序设计时,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,使主块和后备块之间的共性错误降到最低程度。

(3) 防卫式程序设计

N 版本程序设计和恢复块方法都是基于设计冗余的思想,这给程序员和处理机都增加了许多工作,而且它们的结构本身又带来了一些问题和困难,例如,多版本程序设计中的相关性错误问题和恢复块方法中的验证测试的设计等。

防卫式程序设计是一种不采用任何传统的容错技术就能实现软件容错的方法,对于程序中存在的错误和不一致性,防卫式程序设计的基本思想是通过在程序中包含错误检查代码和错误恢复代码,使得一旦发生错误,程序就能撤销错误状态,恢复到一个已知的正确状态中去。例如, C++ 语言中的 `try_except` 处理法和 `try_finally` 中止法就属于防卫式程序设计。

11.1.5 系统可靠性计算

可靠性计算主要涉及 3 种系统,即串联系统、并联系统和冗余系统,其中,串联系统和并联系统的可靠性计算都非常简单,只要了解其概念,公式很容易记住。冗余系统已不是考试考查的重点,所以在此不详述。

1. 串联系统

假设一个系统由 n 个子系统组成,当且仅当所有的子系统都能正常工作时,系统才能正常工作,这种系统称为串联系统,如图 11-4 所示。

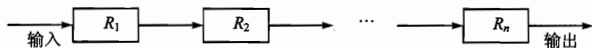


图 11-4 串联系统

设系统各个子系统的可靠性分别用 R_1, R_2, \dots, R_n 表示, 则系统的可靠性 $R = R_1 \times R_2 \times \dots \times R_n$ 。

如果系统的各个子系统的失效率分别用 $\lambda_1, \lambda_2, \dots, \lambda_n$ 来表示, 则系统的失效率 $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$ 。

2. 并联系统

假如一个系统由 n 个子系统组成, 只要有一个子系统能够正常工作, 系统就能正常工作, 如图 11-5 所示。

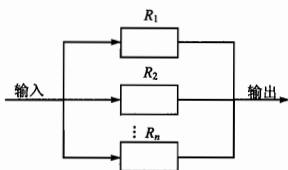


图 11-5 并联系统

设系统各个子系统的可靠性分别用 R_1, R_2, \dots, R_n 表示, 则系统的可靠性 $R = 1 - (1 - R_1) \times (1 - R_2) \times \dots \times (1 - R_n)$ 。

假如所有子系统的失效率均为 λ , 则系统的失效率 μ 为:

$$\mu = \frac{1}{\frac{1}{\lambda} \sum_{j=1}^n \frac{1}{j}}$$

在并联系统中只有一个子系统是真正需要的, 其余 $n-1$ 个子系统都被称为冗余子系统。该系统随着冗余子系统数量的增加, 其平均无故障时间也会增加。

11.2 典型试题分析

试题 1

某高可靠性计算机系统由如图 11-6 所示的冗余部件构成。若每个部件的千小时可靠度 R 均为 0.9, 则该计算机系统的千小时可靠度为 (1), 该计算机系统的失效率 λ 可使用 (2) 来计算。



图 11-6 某高可靠性计算机系统

- | | | | |
|------------------|---------------|-----------------|------------------|
| (1) A. 0.656 | B. 0.729 | C. 0.801 | D. 0.864 |
| (2) A. $\ln R/t$ | B. $-\ln R/t$ | C. $\log_2 R/t$ | D. $-\log_2 R/t$ |

注: t 表示时间

试题分析

本题的系统为一个串并联综合系统，我们可以先计算出中间 2 个并联系统的可靠度，根据并联公式 $R = 1 - (1 - R) \times (1 - R) \times \cdots \times (1 - R)$ ，可得到 3 个部件并联的可靠度为 $1 - (1 - R)$ ，2 个部件并联的可靠度为 $1 - (1 - R)$ 。

然后，再根据串联公式 $R = R \times R \times \cdots \times R$ ，可得到整个系统的可靠度为 $R * (1 - (1 - R)) * (1 - (1 - R)) * R = 0.9 * 0.999 * 0.99 * 0.9 = 0.8010981$

失效率与可靠度的关系为 $R = e^{-\lambda t}$ ，解这个等式，可得 $\lambda = -\ln R / t$ 。

试题答案

(1) C

(2) B

试题 2

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___(3)___。

- (3) 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

试题分析

软件容错的主要目的是提供足够的冗余信息和算法程序，使系统在实际运行时能够及时发现程序设计错误，采取补救措施，以提高软件可靠性，保证整个计算机系统的正常运行。

软件容错技术主要有恢复块方法、N 版本程序设计和防卫式程序设计等。

试题中提到的一致性检查和能力检查也是软件冗余的方法，其中，一致性检查是先预测程序的运行结果，然后将程序运行中和运行后的结果与预测值进行比较，根据比较结果判断软硬件的故障。能力检查是用诊断程序检查系统各个部件的功能是否符合设计的要求，例如，用诊断程序读写存储器的各个存储单元以检查其读写和存储功能是否正常，又如，用一组数据检查运算逻辑部件的功能是否正常。

试题答案

(3) B

试题 3

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___(4)___。

- (4) A. 硬件冗余 B. 信息冗余 C. 时间冗余 D. 人员冗余

试题分析

实现容错技术的主要手段是冗余。冗余是指实现系统规定功能是多余的那部分资源,包括硬件(结构)、软件(冗余附加)、信息和时间。

① 结构冗余。结构冗余是通常用的冗余技术,按其工作方式,可分为静态冗余、动态冗余和混合冗余3种。

- 静态冗余。常用的有三模冗余和多模冗余。静态冗余通过表决和比较来屏蔽系统中出现的错误。例如,三模冗余是对三个功能相同但由不同的人采用不同的方法开发出的模块的运行结果进行表决,以多数结果作为系统的最终结果。即如果模块中有一个出错,这个错误能够被其他模块的正确结果“屏蔽”。由于无须对错误进行特别的测试,也不必进行模块的切换就能实现容错,故称为静态冗余。
- 动态冗余。动态冗余的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顶替它并重新运行。这里须有检测、切换和恢复过程,故称其为动态冗余。每当一个出错模块被其备用模块顶替后,冗余系统相当于进行了一次重构。各备用模块在其待机时,可与主模块一样工作,也可不工作。前者叫做热备份系统,后者叫做冷备份系统。在热备份系统中备用模块在待机过程中的失效率为0。
- 混合冗余。它兼有静态冗余和动态冗余的长处。

② 信息冗余。为检查或纠正信息在运算或传输中的错误,须外加一部分信息,这种现象称为信息冗余。

③ 时间冗余。指以重复执行指令(指令复执)或程序(程序复算)来消除瞬时错误带来的影响。

④ 冗余附加技术。指为实现上述冗余技术所需的资源和技术,包括程序、指令、数据、存放和调动它们的空间和通道等。

试题答案

(4) D

试题 4

下面关于计算机性能的各种评估方法的论述中,正确的是__(5)___。

- (5) 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

试题分析

请参看“11.1.2 系统性能评价”章节相关内容。

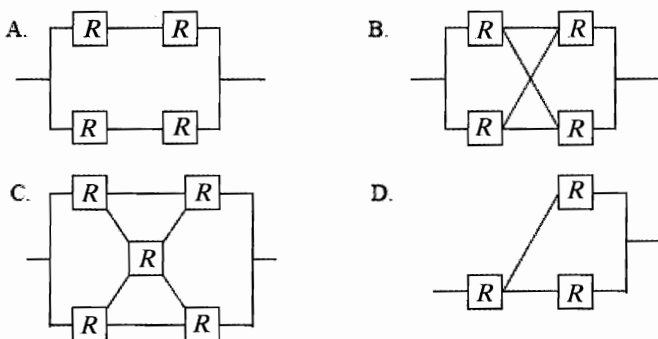
试题答案

(5) D

试题 5

在下列所示系统中每个部件的可靠度都等于 R ($0 < R < 1$)，则通过分析比较就可以判断，系统可靠度最高的是 (6)。

(6)



试题分析

如果对上述每个系统直接计算其可靠度，则会非常复杂，也很费时间。但目标要求并不是计算，而是判断并选择最高可靠度的系统。对这样的问题，用定性方法处理是非常简单有效的。在实际工作中，高水平的人常能跳出繁杂的数量层，在定性的层次考虑问题，常能很快地获得宏观的结论。本题就是典型的例子。

为方便描述，设 R_A 、 R_B 、 R_C 、 R_D 分别为系统 A~D 的可靠度。

显然，在任何系统中，如果将其中某个部件增加可靠度，其他部件的可靠度不变，则整个系统的可靠度将有所增加或保持不变；如果将系统中某个部件降低可靠度，其他部件的可靠度不变，则整个系统的可靠度将有所降低或保持不变。

在系统 C 中，如果将位于中央的那个部件换成一个总是失效的部件，则该系统就成为系统 A；如果将该部件换成一个总是可靠的部件，则该系统就成为系统 B；因此， $R_B \geq R_C \geq R_A$ 。

在系统 B 中，如果将左上角的那个部件换成一个总是失效的部件，则该系统就成为系统 D。因此， $R_B \geq R_D$ 。

综合上述比较分析可知，在题中所述的 4 个系统中，系统 B 具有最高的可靠度。

试题答案

(6) B

试题 6

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 (7) 组成。

(7) A. 缓冲区 B. 控制器 C. 加速器 D. 表决器

试题分析

参看“11.1.4 容错技术”的“ N 版本程序设计”可知： N 模冗余系统中需要有一个表决器。

试题答案

(7) D

试题 7

系统可靠性的简单度量是平均故障间隔时间 (MTBF)，其计算公式是 (8)；软件可用性是指在某个给定时间点上程序能够按照需求执行的概率，其定义为 (9)。(MTTF: Mean Time To Failure; MTTR: Mean Time To Repair)

- (8) A. MTTF+MTTR B. MTTF-MTTR
C. MTTR-MTTF D. MTTF×MTTR
- (9) A. 可用性=MTTF / (MTTF+MTTR) ×100%
B. 可用性=MTTR / (MTTF+MTTR) ×100%
C. 可用性=MTTF / (MTTF-MTTR) ×100%
D. 可用性=MTTR / (MTTF-MTTR) ×100%

试题分析

可用性是指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。可用性的计算公式如下：

$$\text{可用性} = \text{MTTF} / (\text{MTTF} + \text{MTTR}) \times 100\%$$

其中，MTTF 指平均失效等待时间，MTTR 指失效平均修复时间，MTBF 指平均失效间隔时间 MTBF，在这三者之间，有如下公式成立：

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

因此，如果缩短 MTTR，也就缩短了 MTBF。而延长 MTBF 和缩短 MTTR，都可延长 MTTF，再根据有效性的计算公式，这样可以提高有效性。

试题答案

(8) A (9) A

试题 8

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 3 个阶段： (10)。

- (10) 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

试题分析

系统的吞吐率是指系统在每秒所处理的事务数，当系统的负荷很小时，吞吐率增长比较快，然后随着负荷的增大，系统的吞吐率的增长会逐渐变缓。当系统的负荷达到临界点时，系统的吞吐率会下降。

试题答案

(10) A

试题 9

假设某磁盘子系统包括以下部件：10 个磁盘，每个磁盘的 MTTF（平均无故障时间）为 1,000,000 小时；1 个 SCSI 控制器，MTTF 是 500,000 小时；1 个电源，MTTF 是 200,000 小时；1 个风扇，MTTF 是 200,000 小时；1 个 SCSI 缆线，MTTF 是 1,000,000 小时。假设每个部件的故障是独立的，整个系统的 MTTF 约为 (11) 年。

(11) A.2

B.3

C.4

D.5

试题分析

在试题所叙述的磁盘子系统中，显然，各部件组成一个串联系统。要注意的是，这里虽然有 10 个磁盘，但它们并不是采用阵列形式，因为试题并没有提到这一点。因此，我们可以认为，只要有 1 个磁盘出现故障，整个系统就出现故障了。

根据失效率 λ 和 MTTF 的关系： $MTTF=1/\lambda$ ，我们可以求出各部件的失效率，然后根据串联系统失效率的计算方法得出整个系统的失效率。最后，再按照 $\lambda=1/MTTF$ 得出整个系统的 MTTF。具体计算公式如下：

$$1/(10/1000000+1/500000+1/200000+1/200000+1/1000000)=41667 \text{ 小时} = 4.76 \text{ 年}。$$

试题答案

(11) D

试题 10

以下关于系统兼容性的叙述，正确的是 (12) 。

- (12) A. 若两种计算机指令系统与体系结构不同，则无法实现高级语言程序兼容
- B. 操作系统的内核在不同体系结构的计算机上是不能兼容的
- C. 操作系统的外层在不同类型的计算机上是难以实现兼容的
- D. 不同体系结构的计算机之间可以实现编译程序兼容

试题分析

高级语言不依赖于具体的计算机指令系统与体系结构，使用高级语言编写的程序可以根据不同的编译系统编译成不同的计算机体系上的指令。因此，不同体系结构的计算机之间难以实现编译程序兼容。

内核是操作系统最基本的部分，它是为众多应用程序提供对计算机硬件的安全访问的一部分软件，这种访问是有限的，并且内核决定一个程序在什么时候对某部分硬件操作多长时间。因为是直接对硬件进行操作，所以操作系统的内核在不同体系结构的计算机上是不能兼容的。内核通常提供一种硬件抽象的方法来完成对硬件的直接操作，硬件抽象隐藏了复杂性，为应用软件和硬件提供了一套简洁、统一的接口，使程序设计更为简单。所以，操作系统的外层在不同类型的计算机上是可以实现兼容的。

试题答案

(12) B

试题 11

按照开放的接口、服务和支持的规范而实现的系统称为开放系统。开放系统环境中的人机界面、系统管理工具、通信服务和安全性等方面都是按公开标准实现的, 这种环境有利于实现应用软件的__(13)___。

- (13) A. 可移植性、可裁剪性和互操作性 B. 可靠性、可用性和可维护性
C. 兼容性、安全性和可理解性 D. 完整性、可扩充性和可推广性

试题分析

根据开放系统的定义, 因为开发系统都是按照开放的接口、服务和支持的规范而实现, 开放系统环境中的人机界面、系统管理工具、通信服务和安全性等方面都是按公开标准实现的。因此, 这种环境有利于实现应用软件的可移植性、可裁剪性和互操作性。

试题答案

(13) A

试题 12

为测量高负载大型计算机系统的性能, 最适宜的方法是__(14)___。

- (14) A. 查看运行日志 B. 硬件监控
C. 软件监控 D. 查看做业记账系统

试题分析

运行日志记录的是作业的活动、操作过程及运行时的问题, 并不是测量计算机系统性能的。

作业记账系统记录的是各作业的运行时间、运行期间所用的软硬件资源量, 也不是测量计算机系统性能的。

硬件监控与软件监控都可以测量计算机系统性能, 但对于高负载的计算机系统来说, 软件监控的开销很大, 会严重降低系统性能。

试题答案

(14) B

试题 13

用 3 个相同的元件组成如图 11-7 所示的一个系统。

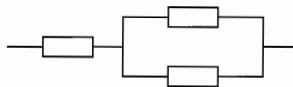


图 11-7 系统图

如果每个元件能否正常工作是相互独立的, 每个元件能正常工作的概率为 p , 那么此系统的可靠度 (元件或系统正常工作的概率通常称为可靠度) 为 (15)。

- (15) A. $p^2(1-p)$ B. $p^2(2-p)$ C. $p(1-p)^2$ D. $p(2-p)^2$

试题分析

图 11-7 的右边是一个并联系统, 其可靠度为 $1-(1-p)^2$, 然后再与左边的部分组成串联系统, 因此整个系统的可靠度为 $p \times [1-(1-p)^2] = p^2(2-p)$ 。

试题答案

- (15) B

试题 14

以下关于基准测试的叙述中, 正确的是 (16)。

- (16) 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

试题分析

各种类型的计算机都具有自己的性能指标, 计算机厂商当然希望自己研制的计算机有较高的性能。同样的计算机, 如果采用不同的评价方法, 所获得的性能指标也会不同。因此, 用户希望能有一些公正的机构采用公认的评价方法来测试计算机的性能。这样的测试称为基准测试, 基准测试采用的测试程序称为基准程序(Benchmark)。基准程序就是公认的标准程序, 用它来测试多种计算机系统, 比较和评价它们的性能, 定期公布测试结果, 供用户选购计算机时参考。

对计算机进行负载测试就是运行某种诊断程序, 加大负载, 检查哪个设备会发生故障。

在程序模块测试后进行的集成测试, 主要测试各模块之间的接口是否正常起作用。

白盒测试就是根据程序内部结构和内部逻辑, 测试其功能是否正确。

试题答案

- (16) C

试题 15

以下关于计算机性能改进的叙述中, 正确的是 (17)。

- (17) 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

- 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____(4)____。
 - (4) A. 软件备份 B. CRC 校验 C. 程序卷回 D. 指令复执
- SPEC 计算机性能测试有不同的方法, 吞吐率测试是指对____(5)____的测试。
 - (5) 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
- MIPS (每秒百万次指令数) 和 MFLOPS (每秒百万次浮点运算数) 是衡量 CPU 性能的两个指标, 其中____(6)____。
 - (6) A. MIPS 适合衡量向量处理机的性能, MFLOPS 适合衡量标量处理机的性能
 - B. MIPS 适合衡量标量处理机的性能, MFLOPS 适合衡量向量处理机的性能
 - C. MIPS 反映计算机系统的峰值性能, MFLOPS 反映计算机系统的持续性能
 - D. MIPS 反映计算机系统的持续性能, MFLOPS 反映计算机系统的峰值性能
- 采用软件冗余的方法提高系统的可靠性, 需要设计 N 个相同功能的程序模块, 这些模块必须____(7)____。
 - (7) A. 由同一组程序员按照相同的规格说明进行编写
 - B. 由同一组程序员按照不同的规格说明进行编写
 - C. 由不同的程序员按照相同的规格说明进行编写
 - D. 由不同的程序员按照不同的规格说明进行编写
- 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
- 下列关于软件可靠性的叙述, 不正确的是____(9)____。
 - (9) A. 由于影响软件可靠性的因素很复杂, 软件可靠性不能通过历史数据和开发数据直接测量和估算出来
 - B. 软件可靠性是指在特定环境和特定时间内, 计算机程序无故障运行的概率
 - C. 在软件可靠性的讨论中, 故障指软件行为与需求的不符, 故障有等级之分
 - D. 排除一个故障可能会引入其他的错误, 而这些错误会导致其他的故障

● 假定求浮点数平方根 (FPSQR) 的操作在某台机器上的一个基准测试程序中占总执行时间的 20%, FP 运算指令所用时间占总执行时间的 50%。采用两种优化 FPSQR 的方法, 第一种方法是增加专门的 FPSQR 硬件, 可以将 FPSQR 的操作速度提高为原来的 10 倍; 第二种方法是提高所有 FP (浮点) 运算指令的执行速度到原来的 1.6 倍, 从而提高求浮点数平方根操作的速度。可以通过计算这两种方法对基准测试程序的加速比来比较这两种方法的优劣。以下叙述正确的是 (10)。

- (10) A. 第一种方法的加速比是 1.23, 效果较好
B. 第二种方法的加速比是 1.23, 效果较好
C. 第一种方法的加速比是 1.22, 效果较好
D. 第二种方法的加速比是 1.22, 效果较好

● 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 (11)。

- (11) A. 5 B. 10 C. 15 D. 20

● 以下关于系统性能的叙述中, 不正确的是 (12)。

- (12) 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

● 数据备份是信息系统运行管理时保护数据的重要措施。(13) 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。

- (13) A. 增量备份 B. 差异备份
C. 完全备份 D. 按需备份

11.4 练习题解析

试题 1 分析

在试题给出的系统中, 客户机之间是并联的 (任何一台客户机出现故障, 对其他客户机没有影响), 同理, 打印机之间也是并联关系。然后, 客户机、服务器、打印

机之间组成一个串联关系。因此，我们可以把该系统简化为如图 11-9 所示的形式。

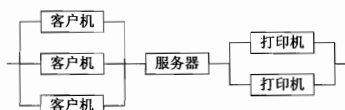


图 11-9 简化后的系统

已知服务器、各客户机及各打印机的可用性分别为 a 、 b 、 c ，因此整个系统的可用性为 $R = (1 - (1 - b))a(1 - (1 - c)) = a(1 - (1 - b))(1 - (1 - c))$ 。

试题 1 答案

(1) D

试题 2 分析

实现容错技术的主要手段是冗余。冗余是指实现系统规定功能是多余的那部分资源，包括硬件（结构）、软件（冗余附加）、信息和时间。

结构冗余是通常用的冗余技术，按其工作方式，可分为静态冗余、动态冗余和混合冗余 3 种。热备系统和冷备系统属于这 3 种当中的动态冗余。

关于冗余技术的详细情况，请参看“11.1.3 冗余技术”。

试题 2 答案

(2) A

试题 3 分析

如果系统的失效率为 λ ，则系统的平均故障间隔时间 $MTBF = 1/\lambda$ 。

试题 3 答案

(3) A

试题 4 分析

软件备份属于软件冗余范畴。信息冗余是在实现正常功能所需要的信息之外再添加一些信息，以保证运行的结果正确。所有的纠错码和检错码都属于信息冗余技术。程序卷回是从出错的地方重新执行程序，属于时间冗余技术。指令复执也是时间冗余技术，就是重新执行出错的指令。

试题 4 答案

(4) B

试题 5 分析

SPEC 对计算机性能的测试有两种方法：一种是测试计算机完成单个任务有多快，称为速度测试；一种是测试计算机在一定时间内能完成多少个任务，称为吞吐率测试。SPEC 的两种测试方法又分为基本的和非基本的两类。基本的是指在编译程序的过程中严格限制所用的优化选项；非基本的是可以使用不同的编译器和编译选项以得到最好的性能，这就使得测试结果的可比性降低。

试题 5 答案

(5) B

试题 6 分析

请参看“11.1.2 系统性能评价”。

试题 6 答案

(6) B

试题 7 分析

N 版本程序设计是一种静态的故障屏蔽技术,采用前向恢复的策略,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 份程序是由不同的人,按相同的需求文档,使用不同的方法独立设计并实现出来的。

试题 7 答案

(7) C

试题 8 分析

系统吞吐率不单取决于 CPU 的速度,还与内外存交换速度、磁盘存取速度等计算机的基本性能有关,也与应用的程序性能有关。因此, A 是错误的。

增加磁盘容量与 CPU 处理时间没有直接的关系,所以, B 也是错误的。

如果事务处理平均响应时间很长,就需要我们去分析其中的原因,然后根据原因采取相应的措施。如果是因为外围设备导致系统瓶颈,则才去提高外围设备的性能。因此, C 是错误的。

根据 20-80 法则,一个程序中 20%的代码使用了该程序所占资源的 80%;一个程序中 20%的代码占用了总运行时间的 80%;一个程序中 20%的代码使用了该程序所占内存的 80%。从这个规律出发,我们在做程序优化的时候,就有了针对性。比如想提高代码的运行速度,根据这个规律可以知道其中 20%的代码占用了 80%的运行时间,因此我们只要找到这 20%的代码,并进行相应的优化,那么程序的运行速度就可以有较大的提高。要想找出那 20%的代码,可以使用性能测试工具,检查程序中各个模块所分配内存的使用情况,以及每个函数所运行的时间等。

试题 8 答案

(8) D

试题 9 分析

软件可靠性是软件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,也就是软件无故障运行的概率。这里的故障是软件行为与需求的不符,故障有等级之分。软件可靠性可以通过历史数据和开发数据直接测量和估算出来。在软件开发中,排除一个故障可能会引入其他的错误,而这些错误会导致其他的故障,因此,在修改错误以后,还需要进行回归测试。

试题 9 答案

(9) A

试题 10 分析

Amdahl 定律: 系统中某一部件由于采用某种更快的执行方式后, 整个系统性能的提高与这种执行方式的使用频率或占总执行时间的比例有关。**Amdahl 定律**定义了由于采用特殊的方法所能获得的加速比的大小。

Amdahl 定律中, 加速比与两个因素有关: 一个是计算机执行某个任务的总时间中可被改进部分的时间所占的百分比, 即 (可改进部分占用的时间/改进前整个任务的执行时间), 记为 fe , 它总小于 1。另一个是改进部分采用改进措施后比没有采用改进措施前性能提高的倍数, 即 (改进前改进部分的执行时间/改进后改进部分的执行时间), 记为 re , 它总大于 1。

Amdahl 定律既可以用来确定系统中对性能限制最大的部件, 也可以用来计算通过改进某些部件所获得的系统性能的提高。**Amdahl 定律**指出, 加快某部件执行速度所获得的系统性能加速比, 受限于该部件在系统中所占的重要性。

假设我们对机器进行某种改进, 那么机器系统的加速比就是:

系统加速比 = 改进后系统性能/改进前系统性能

或者

系统加速比 = 改进前总执行时间/改进后总执行时间

系统加速比依赖于以下两个因素:

① 可改进部分在原系统计算时间中所占的比例。例如, 一个需运行 60 秒的程序中, 有 20 秒的运算可以加速, 那么该比例就是 20/60。这个值用“可改进比例”表示, 它总是小于等于 1 的。

② 可改进部分改进以后的性能提高。例如, 系统改进后执行程序, 其中可改进部分花费 2 秒时间, 而改进前该部分需花费 5 秒, 则性能提高为 5/2。用“部件加速比”表示性能提高比, 一般情况下, 它是大于 1 的。

部件改进后, 系统的总执行时间等于不可改进部分的执行时间加上可改进部分改进后的执行时间, 系统加速比为改进前与改进后总执行时间之比, 即:

系统加速比 = 改进前总执行时间/改进后总执行时间 = $1/((1-fe) + fe/re)$

Amdahl 定律有 3 个推论:

① **Amdahl 定律**表达了一种性能增加的递减规则: 如果仅仅对计算机中的某一部分做性能改进, 则改进越多, 系统获得的效果越小。

② 如果只针对整个任务的一部分进行优化, 那么所获得的加速比不大于 $1/(1-fe)$ 。

③ **Amdahl 定律**告诉我们如何衡量一个“好”的计算机系统: 具有高性价比的计算机是一个带宽平衡的系统, 而不是看它使用的某些部件的性能。

在本题中第一种方法 $fe=0.2$, $re=10$, 系统加速比 = $1/((1-fe) + fe/re) = 1/((1-0.2) +$

$0.2/10)=1.22$ 。

第二种方法 $fe=0.5, re=1.6$, 系统加速比 $= 1/((1-fe) + fe/re) = 1/((1-0.5)+0.5/1.6)=1.23$ 。

所以第二种方法效果更好。

试题 10 答案

(10) B

试题 11 分析

本题主要考查多处理机系统的性能上限计算问题, 根据计算公式 $p = \frac{n}{1+(n-1)a}$, 可以看出当 $a=0.1$ 时, 公式变为 $p = \frac{10n}{10+(n-1)}$, 对该表达式进行求导, 得到该式的极限为 10。

试题 11 答案

(11) B

试题 12 分析

本题主要考查系统性能评估的主要方法和需要注意的问题。对运行系统进行评估的主要目的是评价信息系统在性能方面的表现, 找出系统可能存在的性能瓶颈。其中, 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等, 评价 Web 服务器的主要性能指标有最大并发连接数、响应延迟和吞吐量等。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。

试题 12 答案

(12) C

试题 13 分析

本题考查数据备份的相关知识。

数据备份从备份量来分, 可以分为完全备份、增量备份、差异备份。

- 完全备份: 备份所有数据。即使两个备份时间点之间数据没有任何变动, 所有数据还是会被备份下来。
- 增量备份: 跟完全备份不同, 增量备份在做数据备份前会先判断数据的最后修改时间是否比上次备份的时间晚。如果不是, 则表示该数据并没有被修改过, 这次不需要备份。所以该备份方式, 只记录上次备份之后的变动情况, 而非完全备份。
- 差异备份: 差异备份与增量备份一样, 都只备份变动过的数据。但前者的备份是针对上次完整备份后, 曾被更新过的。

从以上对备份方式的分析可以得知: 增量备份可针对上次任何一种备份进行。

试题 13 答案

(13) A

12

第 12 章

案例分析

根据考试大纲，系统架构设计师考试中的案例分析试题涉及以下内容：

- 系统规划：包括系统项目的提出与可行性分析、系统方案的制定、评价和改进、新旧系统的分析和比较、现有软件、硬件和数据资源的有效利用。
- 软件架构设计：XML 技术、基于架构的软件开发过程、软件质量属性、架构模型（风格）、特定领域软件架构、基于架构的软件开发方法、架构评估、软件产品线、系统演化。
- 设计模式：设计模式的概念、设计模式的组成、模式和软件架构、设计模式分类、设计模式的实现。
- 系统设计：处理流程设计、人机界面设计、文件设计、存储设计、数据库设计、网络应用系统的设计、系统运行环境的集成与设计、中间件、应用服务器、性能设计与性能评估、系统转换计划。
- 软件系统建模：系统需求、建模的作用和意义、定义问题（目标、功能、性能等）与归结模型（静态结构模型、动态行为模型、物理模型）、结构化系统建模、数据流图、面向对象系统建模、统一建模语言（UML）、数据库建模、E-R 图、逆向工程。
- 分布式系统设计：分布式通信协议的设计、基于对象的分布式系统设计、基于 Web 的分布式系统设计、基于消息和协同的分布式系统设计、异构分布式系统的互操作性设计。
- 嵌入式系统设计：实时系统和嵌入式系统特征、实时任务调度和多任务设计、中断处理和异常处理、嵌入式系统开发设计。
- 系统的可靠性分析与设计：系统的故障模型和可靠性模型、系统的可靠性分析和可靠度计算、提高系统可靠性的措施、系统的故障对策和系统的备份与恢复。
- 系统的安全性和保密性设计：系统的访问控制技术、数据的完整性、数据与文件的加密、通信的安全性、系统的安全性设计。

12.1 历年考试情况分析

在历年的考试试题中，系统架构设计师案例分析试题一共 5 道题，试题一为必答题，试题二到试题五选答题（四选二），每道试题 25 分，满分为 75 分。每道试题的结构基本相同，一般首先是一段说明性文字（即案例背景介绍），然后跟随 3~4 个问题。历年考试的试题如表 12-1 所示。

表 12-1 系统架构设计师案例分析试题分布表

考试时间	主要知识点	考查内容说明
2009.11	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
2010.11	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
2011.11	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

从表 12-1 中可以看出，在系统架构设计师案例分析试题中，主要的考点集中于：架构设计、系统建模、嵌入式系统设计、安全性与可靠性设计等方面。

12.2 典型试题分析

本节对一些典型的试题进行分析和解答，以便考生更好地掌握相关知识点。

12.2.1 结构化软件系统建模

希赛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12-1所示。

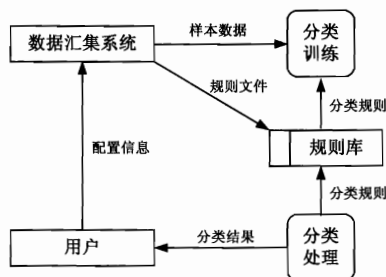


图 12-1 情报分类子系统的1层数据流图

项目组经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12-1所示的数据流图中存在的错误。

【问题1】

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2】

请分析指出图12-1所示的数据流图中存在的错误及其原因，并针对图12-1所示的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3】

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字以内说明在设计高质量的数据流图时应考虑的3个原则。

试题分析

这是一道流程图与数据流图的试题，主要考查系统流程图与数据流图的区别，以及数据流图的基本应用。这个试题对于以前参加过软件设计师考试的考生而言就非常容易了。数据流图是必考题，每年都会考到。

问题1主要考查系统流程图与数据流图的区别。系统流程图是描绘物理系统的传统工具。它的基本思想是用图形符号以黑盒子形式描绘系统里面的每个部件(程序、文件、数据库、表格、人工过程等)。系统流程图表达的是部件的信息流程，而不是表示对信息进行加工处理的控制过程。

数据流图就是采用图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

数据流图和流程图的区别主要表现于：数据流图用于系统的逻辑建模，而流程图用于物理建模。数据流图展现的是全局处理过程，而流程图往往是局部的处理过程。流程图是顺序执行的处理过程，即一个时间点，通常是处理一项事务，而数据流图的

处理过程是并行的，多个处理流程同时进行。数据流图展现的是系统数据流，而流程图展现的是系统控制流。

问题 2 是找错误数据流及根据低层数据流图来画高层数据流图。这是软件设计师的经典考查形式。

在找错误数据流时，首先需要了解错误数据流的类型：一类是语法错误，包括外部实体之间、数据存储之间或外部实体与数据存储之间不经过加工而存在直接数据流；另一类是逻辑错误，包括数据黑洞（只有输入没有产生输出）、灰洞（输入不足以产生输出）和无输入。

- “分类训练”加工属于数据黑洞错误。
- “分类处理”加工属于无输入错误。
- “规则文件”和“配置信息”数据流属于语法错误。

对于复杂系统，一般采用分层数据流图逐步细化系统的内部处理逻辑。0 层数据流图是 1 层数据流图更高级别的抽象。“分类训练”和“分类处理”加工属于内部加工，“分类规则”数据流属于内部数据流，抽象为“情报分类子系统”加工。其中，“样本数据”、“规则文件”和“配置信息”为输入数据流；“分类结果”为输出数据流。

问题 3 属于纯概念题，高质量的数据流图首先要求满足数据流一致性要求，即数据流的输入/输出应该匹配，输入数据的信息量足以产生输出数据，所有的输出数据都能够从输入数据中找到数据的来源。对于复杂性系统，可以采用 DFD 分层结构将系统的复杂性逐步分解降低，以提高数据流图的有效性。高质量的数据流图应该简化复杂的接口，最小化接口或连接的数目。

试题参考答案

【试题 1】

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建模阶段。

【问题 2】

如图 12-2 所示，数据流图中存在的错误有以下 4 种：

- ① “分类训练”加工：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。



图 12-2 情报分类子系统的 0 层数据流图

【问题 3】

高质量数据流图设计时应考虑的 3 个原则如下：

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。如果要知道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？

12.2.2 联合需求分析会议

某软件公司接受 A 公司委托开发一个软件任务，该任务由张工负责。张工预计在 4 周内完成对系统的需求分析，并形成需求规格说明书。张工委派了项目组的小刘来负责需求信息的获取。

两周后，小刘向张工汇报了他进行需求分析的过程及结果。小刘采用问卷调查的方式向 A 公司的 50 名工作人员搜集信息。他首先准备了问卷的初稿，并请 A 公司的相关管理人员进行了测试和修正；然后将问卷分发给 A 公司的每位工作人员，并要求他们在一周内返还问卷。但到目前为止，小刘只收回了 7 份问卷。小刘认为自己是完全按照问卷调查的步骤和要求实施的，而问卷的返还率仍然很低。张工听完后，给小刘分析了失败的原因，并提出了一些能够提高问卷返还率的建议。

但是为了不耽误项目的进度，张工决定采用 JRP (Joint Requirements Planning) 的方法再次进行需求调查，张工作为 JRP 的主持人。最终在第 4 周完成了需求规格说明书，并决定了系统后续阶段的开发计划，如图 12-3 所示。

该项目组除了张工之外，还有 2 名全职的开发人员，可以承担项目中的任何任务，并且承担同一任务的开发人员总是在一起工作。预计的开发时间中已经包含了编写文档的时间。张工决定采用迭代模型，在 160 天内完成这 3 个模块的设计、实现与测试。

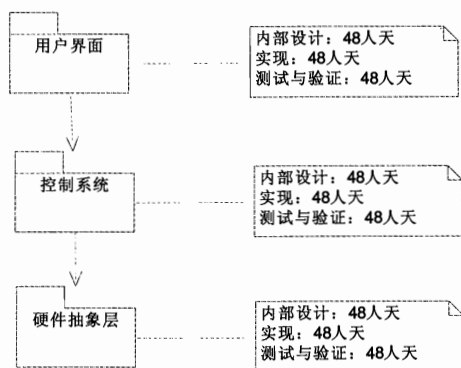


图 12-3 系统开发计划示意图

【问题 1】

用 150 字以内的文字，说明张工给小刘提出的提高问卷返还率的可能措施。

【问题 2】

请用 300 字以内文字简要说明 JRP 的基本思想及保证 JRP 顺利实施的基本原则。

【问题 3】

假设：

- ① 整个开发实施两轮迭代。
- ② 每个任务都被划分为 2 个子任务（例如，实现可以划分为实现 1 和实现 2），对应两轮迭代。
- ③ 完成每个子任务需要花费 24 人天。
- ④ 整个系统的集成测试、改正错误及验证需要花费 48 人天。
- ⑤ 第一轮迭代结束时，形成版本 v0.5；第二轮迭代结束时，整个系统的开发任务全部完成，形成版本 v1.0。

根据上述假设，给出采用迭代模型开发的各里程碑及其完成时间（标出在第几天完成）与交付产品。

试题分析

收集系统需求的方法有很多种，问卷调查是其中使用得较多的一种。它的优点是：可以快速收集信息；收集信息所花费的成本较低；由于问卷调查是不记名的，大多数人可能会给出真实的反馈；而问卷调查的结果也比较好整理和统计。它的一个较大的缺点是：问卷返还率通常会比较低。所以在采用问卷调查的方式收集需求时，除了组织适当的问题，选择合适的调查人群之外，一定要事先考虑到如何解决问卷返回率低的问题。

JRP 是另外一种常用的需求获取方法。JRP 基本思想是通过召开一系列高度结构化的分组会议，快速地分析问题、定义需求。它是 JAD（Joint Application Development，联合应用开发）技术的一个子集。JRP 的主要意图是收集需求，而不是对需求进行分析和验证。

试题参考答案

【问题 1】

为了提高问卷返还率，可采取以下措施：

- ① 向所有的工作人员解释问卷的目的，以及如何使用这些信息。
- ② 说明这份问卷是公司的每个工作人员都要回答的。
- ③ 拜托相关领导督促他所管辖的工作人员回答问卷，并及时返还。
- ④ 尽量参加一次这个公司的全体会议，在会议上解答工作人员们提出的问题，并解释这些信息的用处。
- ⑤ 更改问卷中的问题，尽量减少回答问卷所花费的时间。
- ⑥ 设置一些奖品或奖励，激励大家及时返还问卷。

【问题 2】

JRP 基本思想是通过召开一系列高度结构化的分组会议，快速地分析问题、定义需求。它是 JAD (Joint Application Development) 技术的一个子集。JRP 的主要意图是收集需求，而不是对需求进行分析和验证。实施 JRP 时应把握以下主要原则：

- ① 在 JRP 实施之前，应制定详细的议程，并严格遵照议程进行。
- ② 按照既定的时间安排进行。
- ③ 尽量完整地记录会议期间的内容。
- ④ 在讨论期间尽量避免使用专业术语。
- ⑤ 充分运用解决冲突的技能。
- ⑥ 会议期间应设置充分的间歇时间。
- ⑦ 鼓励团队取得一致意见。
- ⑧ 保证参加 JRP 的所有人员能够遵守实现约定的规则。

【问题 3】

第 24 天：完成用户界面的设计 1、控制系统的设计 1 和控制系统的实现 1。

第 48 天：完成硬件抽象层的设计 1、用户界面的实现 1 和控制系统的测试 1。

第 72 天：完成硬件抽象层的实现 1、测试 1 和用户界面的测试 1。

第一次迭代完成，交付产品：系统的 v0.5。（所有构件的子任务 1 都完成）。

第 96 天：完成用户界面、控制系统、硬件抽象层的设计 2。

第 120 天：完成控制系统、硬件抽象层的实现 2；用户界面的测试 2。

第 144 天：完成硬件抽象层的测试 2、用户界面的实现 2 和用户界面的测试 2。

交付产品：系统的 α 版（所有构件的子任务 2 都完成）。

第 160 天：完成集成测试、用户验证及改正错误。

第二轮迭代完成，交付产品：系统的 v1.0。

12.2.3 电子政务

电子政务是指政府机构利用信息化手段来实现政府职能。

某市房地产交易网站是市建设委员会实施电子政务的门户，网站包括以下栏目：项目公示、业务办理、信息发布、通知公告、政策法规、房地产经纪、在线答疑等，其中业务办理栏目中又包括申办预售许可、期房网上签约、申请预售登记、权属登记申请、现房网上签约、经纪机构管理、评估行业管理等项目，多数的业务办理项目需要管理部门多级审批。

【问题 1】

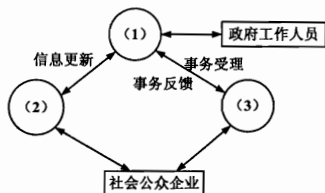


图 12-4 电子政务业务模型

一般而言，电子政务业务分为 3 个领域，电子政务业务模型如图 12-4 所示（箭头表示信息的流向）。请在图（1）、（2）、（3）空中填写恰当的内容。

【问题 2】

电子政务根据其服务的对象不同，基本上可以分为 4 种模式，即 G2G、G2B、G2C、G2E。请根据本题中房地产交易网站的栏目内容，说明该市建设委员会的电子政务系统包括了哪些模式？为什么？

【问题 3】

本题中的电子政务项目在进行需求分析时，系统分析师需要有效地获取需求，进行需求建模。需求建模包括域建模、用例建模、组件和服务建模、性能建模等。请用 300 字以内文字分别简要叙述什么是用例建模、组件和服务建模、性能建模。

【问题 4】

系统分析师必须能够与具有不同背景的利益相关者（如政府各个部门、房地产开发企业、购房者等）进行沟通交流，以提取和细化需求，并向这些利益相关者描述系统的体系结构。请用 50 字以内文字简要叙述常用的沟通交流技巧。

试题分析

电子政务是指政府机构利用信息化手段，实现各类政府职能。其核心是：应用信息技术，提高政府事务处理的信息流效率，改善政府组织和公共管理。

【问题 1】

根据政府机构的业务形态来看，通常，电子政务主要包括以下 3 个应用领域。

① 政务信息查询：面向社会公众和企业组织，为其提供政策、法规、条例和流程的查询服务。

② 公共政务办公：借助互联网实现政府机构的对外办公，如申请、申报等，提

高政府的运作效率，增加透明度。

③ 政府办公自动化：以信息化手段提高政府机构内部办公的效率，如公文报送、信息通知和信息查询等。

其业务模型如图 12-5 表示。

在图 12-5 中，社会公众和企业主要通过政务信息查询，以及公共政务办公与电子政务平台建立沟通，相关事务处理请求通过办公自动化系统中转给政府工作人员，政府工作人员可以通过办公自动化系统进行政务处理及对政务信息查询系统的更新。通过对这一典型业务模型的分析，可以看出在电子政务系统中主要存在 3 种信息流。

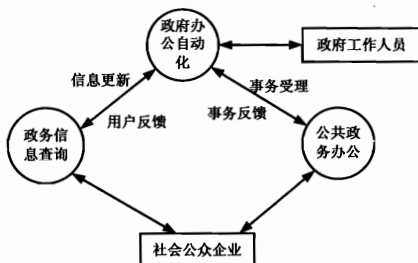


图 12-5 电子政务业务模型

① 政务办公信息流：主要存在于政府机构内部办公的过程中。

② 公共事务信息流：主要存在于政府机构对外办公的过程中。

③ 政务咨询信息流：主要存在于社会公众和企业查询相关信息的过程中。

【问题 2】

第 2 个问题考查电子政务业务的分类，这是大家都知道的内容了：政府对政府、政府对公务员、政府对企业、政府对公众。根据这些含义，对比试题描述中的功能，逐个划分就可以了。

【问题 3】

第 3 个问题考查域建模、用例建模、组件和服务建模、性能建模的概念。

域建模是指为问题域创建相应的模型并且把它划分为若干个内聚组的过程。域模型是一种用于理解问题域的工具。要构造域模型，必须完成下列工作：

- ① 标识并确定参与者（实体）及其操作（活动）的特征。
- ② 标识管理操作（规则）的策略。
- ③ 收集有关实现这些操作、来自这些操作或者记录这些操作（构件/数据）的信息。
- ④ 将相关的要素划分为子域。
- ⑤ 确定结果域（核心的/通用的/外部的）及它们之间交互的特征。

用例模型描述了各种参与者（人和其他系统）和要分析的系统之间的主要交互。用例应该说明系统如何支持域和业务流程模型中的业务流程。用例模型应该将系统放到上下文环境中，显示系统和外部参与者之间的边界，并描述系统和参与者之间的关键交互。用例建模可以描述利益相关者（如用户和维护人员）所看到的系统行为。

用例建模建立反映系统行为的动态模型，用例模型描述了各种参与者和要分析的系统之间的主要交互。

组件模型为子系统、模块和组件的层次结构分配需求和职责。服务模型将应用程

序定义为一组位于外部边界（用例）、架构层之间的抽象服务接口，并且提供了通用的应用程序和基础结构（安全、日志记录、配置等）。

性能建模是通过各种各样的方式来度量性能。必须考虑性能建模过程中其他的几个方面：

- ① 构建和部署应用程序的速度如何？
- ② 构建、维护和运行需要多少花费？
- ③ 该应用程序能在多大程度上满足其需求？
- ④ 对于必须使用该应用程序的人来说，需要为其付出多大的开销？
- ⑤ 该应用程序会对其他应用程序和基础结构产生怎样的影响？

关于这些问题（和无数的其他问题，这取决于具体情况）的答案，对一个成功的应用程序来说是至关重要的，并且通常称其为应用程序在架构上的质量。对这些质量进行建模是很困难的，甚至比性能的标准质量更困难。

【问题 4】

第 4 个问题考查获得需求的沟通与交流技巧，这是经常考查的一个内容。

① 用户访谈。用户访谈是最基本的一种需求捕获手段，其形式包括结构化和非结构化两种，结构化是指事先准备好一系列问题，有针对地进行；而非结构化则是只列出一个粗略的想法，根据访谈的具体情况发挥。最有效的访谈是结合这两种方法进行，毕竟不可能把什么都一一计划清楚，应该保持良好的灵活性。

② 用户调查。用户访谈时最大的难处在于很多关键的人员时间有限，不容易安排过多的时间；而且客户面经常较广，不可能一一访谈。因此，我们就需要借助“用户调查”这一方法，通过精心设计要问的问题，然后下发到相关的人员手里，让他们填写答案。这样就可以有效地克服前面提到的两个问题。

但是与用户访谈相比，用户调查最大的不足就是缺乏灵活性；而且双方未见面，分析人员无法从他们的表情等其他动作来获取一些更隐性的信息；还有就是客户有可能在心理上会不重视一张小小的表格，不认真对待，从而使得反馈的信息不全面。因此较好的做法是将这两种技术结合使用。具体来说，就是先设计问题，制作成为用户调查表，下发填写完后，进行仔细地分组、整理、分析，以获得基础信息，然后再针对这个结果进行小范围的用户访谈，作为补充。

③ 联合讨论会。这是一种相对来说成本较高的需求获取方法，但也是十分有效的一种。它通过联合各个关键客户代表、分析人员、开发团队代表一起，通过有组织的会议来讨论需求。通常该会议的参与人数为 6~18 人，召开时间为 1~5 小时。

其他的方法还有通过原型进行演示，通过书面或电子邮件等手段与用户交流等。

试题参考答案

【问题 1】

(1) 政府办公自动化（或办公自动化系统）

(2) 政务信息查询（或政务信息发布系统）

(3) 公共政务办公（或政务业务办理系统）

【问题 2】

G2B，栏目中有申办预售许可、申请预售登记等，针对房地产开发商企业。

G2C，栏目中有权属登记申请等，主要是针对购房个人。

G2E，因为题目中指出多数业务办理项目需要政府主管部门多级审批，所以系统后台还包括办公自动化系统。

【问题 3】

用例建模描述各种参与者（人和其他系统）和系统之间的主要交互。用例建模可以描述利益相关者（如用户和维护人员）所看到的系统行为。

组件建模确定系统的子系统、模块和组件结构，为子系统、模块分配需求和职责，每个组件元素作为一个自包含的单元，用于开发、部署和执行。服务建模提供了通用的应用程序，并将应用程序定义为一组抽象服务接口。

性能建模是对系统的性能进行度量，为每个组件确定性能指标。包括执行时间、资源使用、开发复杂性、维护复杂性等质量属性。

【问题 4】

调查、访谈、演示、组交互（会议）、书面交流（电子邮件）等。

12.2.4 软件系统架构

希赛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。如图 12-6 所示给出了该系统的简化示意图。表 12-2 描述了各种系统输入的含义。

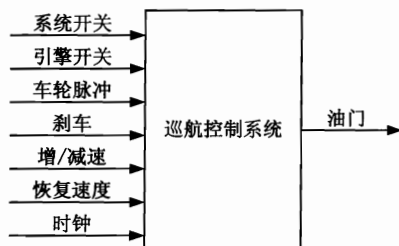


图 12-6 定速巡航控制系统简化示意图

表 12-2 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】

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 2】

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 3】

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。

试题分析

软件系统架构风格多种多样,如何根据系统的要求选择合适的架构风格,是系统架构设计师需要掌握的重要技能。

问题 1 考查概念层面的问题,根据卡耐基·梅隆大学软件工程研究所的定义,软件架构风格是描述某一类特定应用领域中软件系统组织方式和惯用方式。

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 难度比问题 1 要大一些。需要根据面向对象的风格特点与控制环路的风格特点,来分析系统功能应如何拆分成构件。

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、速度计和车轮等对象。整个计算过程通过对象之间的合理交互完成期望的功能。

控制环路的架构风格将以控制器为核心,期望速度、车轮脉冲、时钟和油门等作为构件。通过不断地反馈机制实现对系统的控制。

问题 3 主要从应用场景的角度比较两种架构风格。面向对象系统比较适合事件驱动的场景,特别是离散突发事件的处理;而控制环路则适合连续事件的处理,如维持恒定车速等。

试题参考答案

【问题1】

软件架构风格是描述某一类特定应用领域中软件系统组织方式和惯用方式。

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】

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如下：

- 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。

具体的计算过程如下：

- 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】

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，这种情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，直至达到恒定速度。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。

12.2.5 SOA

某银行开通了网上银行业务，其网上贷款业务流程如下：

① 客户在网上填写姓名、电子邮件地址、贷款类型、贷款金额、身份证号、通讯地址等信息，提交贷款申请。

② 在指定的时间内，客户会收到银行的电子邮件，通知贷款是否被批准。

③ 银行根据客户提交的信息，创建贷款申请任务，创建工作由运行在主机上的CICS（客户信息控制系统）完成，同时需要从第三方获得客户的信用审查信息。

④ 由信贷员对该项贷款申请业务进行审批，然后由风险检查系统评估该项贷款的风险程度，风险大的贷款申请被拒绝。

⑤ 无论批准或者拒绝，结果都会通过邮件系统递交给客户。对于拒绝的贷款申请，还要通知贷款申请任务进行有关操作。

⑥ 除了信贷员审批环节需要人机交互外，业务是自动进行的。

【问题 1】

上述网上贷款业务采用 SOA 架构来实现。上述业务流程中涉及哪些功能单元？什么是 SOA？本题中的案例采用 SOA 具有哪些优点？请用 200 字以内文字说明。

【问题 2】

请在答题纸上将以下关于 SOA 的叙述填写完整。

SOA 不是一个新鲜事物，但它却是传统的面向对象模型的替代模型。相比较而言，面向对象的模型是 (1) 耦合和 (2) 粒度的，而 SOA 是 (3) 耦合和 (4) 粒度的。SOA 系统原型的一个典型例子是 (5) (CORBA)，它已经出现很长时间了，其定义的概念与 SOA 相似。

随着 Web Services 的成熟，现在的 SOA 已经有所发展，这些进展是以 (6) 为基础的。在 Web Services 中，通过 (7) 来描述接口，与 CORBA 中的 (8) (接口描述语言) 相比，它动态性更强、灵活度更高。

SOA 还具有管理上的优点。例如，现在管理员可直接管理开发人员所构建的 (9)，这远胜于以往管理单个应用的方式。通过分析 (10) 间的交互，SOA 可以帮助企业了解何时以及什么业务逻辑被切实执行了，这使管理员能够有针对性地优化业务流程。

【问题 3】

上述网上贷款系统能够实际应用的基本前提之一是满足金融领域的安全性需求。

该系统必须要满足哪些安全方面的需求？请用 200 字以内的文字简要说明。

试题分析

本题的试题背景为网上银行贷款业务，实际考查 SOA 的理论和应用，试题共 3 个小问题。

【问题 1】

第一个问题要求考生回答系统有哪几个功能模块、SOA 的定义和优点。系统的功能模块可以根据试题描述中给出的流程直接得出。

- ① 提交贷款申请：客户在网上填写姓名、电子邮件地址、贷款类型、贷款金额、身份证号、通讯地址等信息。
- ② 信用审查：从第三方获得客户的信用审查信息。
- ③ 创建贷款申请任务：银行根据客户提交的信息，创建贷款申请任务。
- ④ 风险评估：由风险检查系统评估贷款的风险程度。
- ⑤ 审批贷款申请：由信贷员对该项贷款申请业务进行审批。
- ⑥ 客户通知：无论批准或者拒绝，结果都会通过邮件系统递交给客户。

有关 SOA 的定义和优点，请读者阅读本系列丛书中的《系统分析师考试系统分析与设计论文试题分类精解》相关内容。

【问题 2】

第二个问题是对第一个问题的补充，是一个填空题，进一步考查 SOA 的概念，以及与相关概念的比较。

SOA 不是一个新鲜事物，但它却是传统的面向对象模型的替代模型。相比较而言，面向对象的模型是紧耦合和细粒度的，而 SOA 是松耦合和大粒度的。SOA 系统原型的一个典型例子是通用对象请求代理结构（CORBA），它已经出现很长时间了，其定义的概念与 SOA 相似。

随着 Web Services 的成熟，现在的 SOA 已经有所发展，这些进展是以 XML 为基础的。在 Web Services 中，通过 WSDL 来描述接口，与 CORBA 中的 IDL（接口描述语言）相比，它动态性更强、灵活度更高。

SOA 还具有管理上的优点。例如，现在管理员可直接管理开发人员所构建的相同服务，这远胜于以往管理单个应用的方式。通过分析服务间的交互，SOA 可以帮助企业了解何时以及什么业务逻辑被切实执行了，这使管理员能够有针对性地优化业务流程。

【问题 3】

第三个问题是一个安全性试题，考生可以从网络安全、信息安全、交易安全等方面进行回答。具体的安全性问题，请读者阅读本书第 5 章的相关内容。

试题参考答案

【问题 1】

- ① 该网上贷款业务至少涉及贷款申请、信用审查、信贷员审批、风险检查、电

子邮件传送等功能单元。

② SOA 即面向服务的体系结构是一个软件架构模型，它将业务的不同功能单元（称为服务）通过服务之间的接口（和契约）联系起来。接口独立于实现服务的硬件平台、操作系统和编程语言。

③ 复用银行的各种应用资源（如软件资产）；增强银行各个业务的集成性和灵活性；业务流程变更时便于快速构建应用系统。

【问题 2】

- | | |
|----------------------|------------------|
| (1) 紧 | (2) 细（小） |
| (3) 松 | (4) 大 |
| (5) 通用对象请求代理结构 | (6) XML（可扩展标记语言） |
| (7) WSDL（Web 服务描述语言） | (8) IDL |
| (9) 相同服务 | (10) 服务 |

【问题 3】

- ① 验证（系统有关角色的身份识别）。
- ② 签名（创建及验证类似手写签名的电子签名）。
- ③ 授权（如信贷员是否具有审批权）。
- ④ 完整性（发送的数据与接收到的数据是否一致）。
- ⑤ 机密性（与业务无关的人员不能读取事务中的数据）。
- ⑥ 审查（把所有事务记录下来，以便事后验证）。
- ⑦ 不可否认性（能由第三方求证事务中发送及收到的是否是同一数据）。
- ⑧ 威胁预防（防止间谍程序登录、攻击系统）。

12.2.6 软件质量属性

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】

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 6 种

不同的软件质量属性名称，并解释其含义。

【问题2】

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。

试题分析

这是一道软件质量特性的试题，软件质量特性是软件架构以及软件架构设计师的一个重要关注点。因为如果在软件架构的设计阶段不考虑软件质量特性，则产生的软件质量隐患是在后期的设计与开发中无法弥补的。

软件质量特性主要包括以下几个方面。

- ① 功能性：系统所能完成期望工作的能力。
- ② 性能：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。
- ③ 可用性：系统能够正常运行的时间比例。
- ④ 可靠性：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。
- ⑤ 健壮性：在处理或环境中，系统能够承受压力或变更的能力。
- ⑥ 安全性：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。
- ⑦ 可修改性：能够快速地对较高的性能价格比对系统进行变更的能力。
- ⑧ 可变性：体系结构经扩充或变更成为新体系结构的能力。
- ⑨ 易用性：衡量用户使用一个软件产品完成指定任务的难易程度。
- ⑩ 可测试性：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。
- ⑪ 互操作性：系统与外界或系统与系统之间的相互作用能力。

问题1是纯概念题，从以上的属性中任选6个作答即可。

问题2难度稍大，需要结合题目给出的案例，来分析系统有哪些质量属性的需求，同时需要给出实现该质量属性的策略。下面逐一分析题目给出的场景。

① 在线交易平台必须在1s内完成客户的交易请求。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（减少计算复杂度等）、资源管理（并发、数据复制等）和资源调度（先进先出队列、优先级队列等）。

② 该平台必须严格保证客户个人信息和交易信息的保密性和安全性。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（授权、认证和限制访问等）、攻击检测（入侵检测等）、从攻击中恢复（部分可用性策略）和信息审计等。

③ 当发生故障时,该平台的平均故障恢复时间必须小于 10s。该要求主要对应可用性,可以采用的架构设计策略有 Ping/Echo、心跳、异常和主动冗余等。

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。添加新功能或进行平台升级必须在 6 小时内完成。该要求主要对应可修改性,可以采用的架构设计策略有软件模块泛化、限制模块之间通信、使用中介和延迟绑定等。

试题参考答案

【问题 1】

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如下:

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对系统性能价格比进行变更的能力。

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】

① 在线交易平台必须在 1s 内完成客户的交易请求。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(减少计算复杂度等)、资源管理(并发、数据复制等)和资源调度(先进先出队列、优先级队列等)。

② 该平台必须严格保证客户个人信息和交易信息的保密性和安全性。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(授权、认证和限制访问等)、攻击检测(入侵检测等)、从攻击中恢复(部分可用性策略)和信息审计等。

③ 当发生故障时, 该平台的平均故障恢复时间必须小于 10s。该要求主要对应可用性, 可以采用的架构设计策略有 Ping/Echo、心跳、异常和主动冗余等。

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。添加新功能或进行平台升级必须在 6 小时内完成。该要求主要对应可修改性, 可以采用的架构设计策略有软件模块泛化、限制模块之间通信、使用中介和延迟绑定等。

12.2.7 J2EE

某软件公司承担了某中小型企业应用软件开发任务, 进度要求紧迫。为了按时完成任任务, 选择合适的企业应用系统开发架构非常重要。项目组在进行方案论证时, 项目组成员提出了两种开发思路。

① 刘工建议采用 J2EE 5.0 和 EJB 3.0 进行开发。理由是 J2EE 定义了标准的应用开发体系结构和部署环境, EJB 是 J2EE 的基础和核心。J2EE 5.0 的主要目标是简化开发, 相比 EJB 2.1, EJB 3.0 具有很多改进和提高。

② 杜工建议采用 Struts、Spring 和 Hibernate 轻量级开源框架相结合的方式。理由是随着 Java 开源项目阵营的发展壮大, 一些基于 POJO (Plan Old Java Objects) 的开源框架被广泛地引入到 Java 企业应用开发中来, 与重量级的 EJB 框架相比, 这些轻量级的框架有很多优点。

项目组仔细比较分析了两种方案的特点、优点和不足之处。认为杜工和刘工的建议都合理, 但是从结合当前项目实际情况出发, 最后决定采用杜工的建议。

【问题 1】

Java 企业应用框架一般被划分为 3 个层次, 请用 150 字以内文字说明都有哪 3 个层次? 功能分别是什么?

【问题 2】

请用 200 字以内的文字叙述 Struts、Spring 和 Hibernate 开源框架特点和结合方式。

【问题 3】

请用 200 字以内的文字说明基于 Struts、Spring 和 Hibernate 的轻量级框架与基于 EJB 的重量级框架解决问题的侧重点有什么不同?

试题分析

【问题 1】

Java 企业应用框架一般被划分为 3 个层次: 表现层、业务逻辑组件层和持久层。

表现层用来建立应用系统的界面, 对应视图 (View)。例如, 表现层采用 JSF (Java Server Face), JSF 的开发流程的核心是事件驱动, 组件和标签的封装程度非常高, 很多典型应用已经不需要开发者去处理 HTTP, 整个过程是通过 IoC (依赖注入) 来实现的。

业务逻辑组件层用来开发应用逻辑, 对应控制器 (Controller)。例如, 业务逻辑组件层采用 EJB 3.0 的 Session Bean。EJB 3.0 允许开发者使用耦合松散的组件来开发

应用,实现一个 EJB 所有使用的类和接口都减少了。

持久层是实现持久化存储,对应模型(Model)。例如,采用 EJB 3.0 实体 Bean 持久化模型,吸收了 Hibernate 的一些思想采用 O/R Mapping 模式。

【问题 2】

Struts 是基于模型-视图-控制器(MVC)模式的开源框架,主要用于企业应用中表示层的实现。借助于 Struts,开发人员开源把主要精力集中在业务处理上,简化遵循 MVC 设计模式的 Web 应用开发工作,很好地实现代码重用,提高开发效率。Struts 框架包括:

① 模型(Model)。Struts 中模型是一个 Action 类,开发者通过其实现业务逻辑,同时用户请求通过控制器向 Action 的转发过程是基于由 struts-config.xml 文件描述的配置信息的。

② 视图(View)。视图是由与控制器配合工作的一整套 JSP 定制标签库构成,利用它们可以快速建立应用系统的界面。

③ 控制器(Controller),本质上是一个 Servlet,将客户端请求转发到相应的 Action 类。

④ 一堆用来做 XML 文件解析的工具包。

Spring 是业务组件层轻量级框架。Spring 框架是一个基于 IoC(依赖注入)和 AOP(面向方面编程)的架构。用户可以通过 Spring 来利用普通 Java 对象(POJO)编程,使用依赖注入解析 POJO 间的依赖性,然后使用面向方面编程将服务与它们相关联。采用依赖注入使得它可以很容易地实现 Bean 的装配,提供简洁的 AOP,并据此实现事务管理等。Spring 的核心要点是:支持不绑定到特定 J2EE 服务的可重用业务和数据访问对象。这样的对象可以在不同的 J2EE 环境(Web 或 EJB)、独立应用程序、测试环境之间重用。

Hibernate 属于持久层框架。Hibernate 是一个开源的 O/R Mapping 框架,它对 JDBC 进行了非常轻量级的对象封装,可以在应用 EJB 的 J2EE 框架中取代 CMP,完成数据持久化的重任。

【问题 3】

作为重量级框架,EJB 框架则强调高可伸缩性,适合于开发大型企业应用。在 EJB 体系结构中,一切与基础结构服务相关的问题和底层分配问题都由容器来处理。另外,EJB 容器通过减少数据库访问次数,以及分布式处理等方式提供了专门的系统性能解决方案,能够充分解决系统性能问题。

轻量级框架侧重于减少开发的复杂度,相应地,它的处理能力便有所减弱(如事务功能弱、不具备分布式处理能力),比较适用于开发小型企业应用。采用轻量框架后,一方面因为采用基于 POJOs 的方法进行开发,使应用不依赖于任何容器,这可以提高开发调试效率;另一方面轻量级框架多数是开源项目,开源社区提供了良好的设计和许多快速构建工具,以及大量现成可供参考的开源代码,这有利于项目的快速开发。

试题参考答案

【问题 1】

表现层、业务逻辑层和持久层。

表现层用来建立应用系统的界面，对应视图（View）。

业务逻辑层用来开发应用逻辑，对应控制器（Controller）。

持久层是实现持久化存储，对应模型（Model）。

【问题2】

Struts 框架是基于模型-视图-控制器（MVC）模式的开源框架，主要用于企业应用中表示层的实现。借助于 Struts，开发人员可以把主要精力集中在业务处理上，简化遵循 MVC 设计模式的 Web 应用开发工作，很好地实现代码重用，提高开发效率。

Spring 框架是业务组件层轻量级框架。

Hibernate 属于持久层框架。Hibernate 是一个开源的 O/R Mapping 框架，它对 JDBC 进行了非常轻量级的对象封装，可以在应用 EJB 的 J2EE 框架中取代 CMP，完成数据持久化的重任。

Struts 框架+Spring 框架+Hibernate 框架实现了表现层、业务逻辑组件层和持久层的结合。

【问题3】

轻量级框架侧重于减少开发的复杂度，相应地，它的处理能力便有所减弱（如事务功能弱、不具备分布处理能力），比较适用于开发中小型企业应用。

采用轻量级框架后，一方面因为采用基于 POJOs 的方法进行开发，使应用不依赖于任何容器，这可以提高开发调试效率；另一方面轻量级框架多数是开源项目，开源社区提供了良好的设计和许多快速构建工具，以及大量现成可供参考的开源代码，这有利于项目的快速开发。

而作为重量级框架 EJB 则强调高可伸缩性，适合于开发大型企业应用。

在 EJB 体系结构中，一切与基础结构服务相关的问题和底层分配问题都由容器来处理。另外，EJB 容器通过减少数据库访问次数，以及分布式处理等方式提供了专门的系统性能解决方案，能充分解决系统性能问题。

轻量级框架的产生并非是对重量级框架的否定，可以说二者是互补的。轻量级框架旨在开发更强大、功能更完备的企业应用；而新 EJB 3.0 则在努力简化 J2EE 使 EJB 不仅是擅长处理大型企业系统，也利用开发中小型系统，这也是 EJB 轻量化的一种努力。对于大型企业应用及将来可能涉及能力扩展的中小型应用，采用结合使用轻量级框架和重量级框架相也不失为一种较好的解决方案。

12.2.8 数据库安全与数据库备份

银行金融系统对数据库中的数据安全要求很高，必须在技术层面上采用必要的措施，以保证数据的有效和不丢失。数据库的备份与恢复是保证数据安全的一种基本方法，一般将数据库备份分为冷备份、热备份和数据导出，而数据导出根据导出数据的范围，又可以分为完全导出、增量导出和累计导出。

该金融系统上线初期，整个业务系统采用了内网方式运行，不与外网发生联系。为

了保证数据库内部数据的安全，该银行信息中心的刘工提出的备份方案如表 12-3 所示。

表 12-3 备份方案

时间	星期一	星期二	星期三	星期四	星期五	星期六	星期日
备份方式	完全导出	增量导出	增量导出	增量导出	累计导出	增量导出	增量导出

信息中心的李工对刘工的方案提出了异议，认为数据库一旦发生故障，刘工的方案无法做到数据库的实时恢复，会导致业务数据的丢失，银行应该采用数据库冷备份和热备份相结合的方式来完成数据库的备份。

【问题 1】

针对金融数据库安全，从技术层面上分析应重点考虑哪些因素？

【问题 2】

请用 200 字以内的文字评价刘工的备份策略的优缺点。

【问题 3】

请分别说明冷备份和热备份的概念，并补充表 12-4 中的（1）~（6）空。

表 12-4 冷备份和热备份的比较

备份方式	优点	缺点
冷备份	快速简单，维护方便	（1）
	可以恢复到备份时的时间点上	单独使用时，只能恢复到备份时的时间点上
	（2）	（3）
热备份	（4）	如果热备份出错，所得结果不能用于时间点的恢复
	可以实现实时/秒级恢复	（5）
	（6）	维护比较困难

【问题 4】

银行金融系统需要支持数据库的实时/秒级恢复（即恢复到故障之前的时间点上），请设计合理的备份策略，并说明其基本恢复过程。

试题分析

本题主要考查数据库故障、备份的概念和分类，以及备份的策略。

问题 1 是安全方面的，对于金融数据库而言，其安全隐患主要是两个方面：一方面是本身的系统产生了故障；另一方面是来自外部的攻击。如果清楚这一点，解题也就容易了。常见的数据库内部故障包括事务故障、系统故障和介质故障。而常见的外部攻击包括 SQL 注入攻击、系统（软件）漏洞、黑客攻击、计算机病毒等。

问题 2 要求分析刘工的备份策略的优缺点。分析优缺点，前提是了解“完全导出、增量导出、累计导出”的含义。

完全导出是一次性将数据库中所有数据都导出来。

增量导出是只导出自上次导出操作（包括完全导出、增量导出、累计导出）之后数据库中变化了的数据。

累计导出是只导出自上次完全导出之后数据库中变化了的数据。

有了以上的知识基础，后面的分析就容易多了。刘工的备份策略仅采用了数据导出的备份方式，其优点是简单易行，实现了数据的完整导出，甚至可以在运行期间做到按需备份。

其缺点是一旦发生故障，会丢失过多的数据，如在星期三的中午数据库出现故障，则只能恢复到星期二备份时的数据，星期二导出后到星期三中午的数据会全部丢失；如果数据量非常大，则数据导出所花费的时间和空间会很大。

问题3也属于纯概念问题，不在此赘述，请参看参考答案。

问题4是进行备份策略的选用，在做此策略时，需要注意冷备份与热备份相结合，这样既可以兼顾效率，又能让系统在出现故障后准确地恢复到故障前的状态，使损失降到最低。

试题参考答案

【问题1】

对数据库内部故障的考虑，包括事务故障、系统故障和介质故障。

对数据库外部各种攻击的考虑，包括计算机病毒、利用软件漏洞的黑客攻击和SQL注入攻击等。

【问题2】

刘工的备份策略仅采用了数据导出的备份方式，其优点是简单易行，实现了数据的完整导出，甚至可以在运行期间做到按需备份。

其缺点是一旦发生故障，会丢失过多的数据，例如，在星期三的中午，数据库出现故障，则只能恢复到星期二备份时的数据，星期二导出后到星期三中午的数据会全部丢失；如果数据量非常大，则数据导出所花费的时间和空间会很大。

【问题3】

冷备份是在数据库关闭的情况下，对数据库中的关键文件（数据文件、日志文件等）进行复制。热备份是在数据库运行的情况下，对数据库中的关键数据进行备份，要求数据库管理系统提供支持。

- (1) 备份时，数据库关闭，不能做其他工作。
 - (2) 与热备份相结合，实现实时/秒级恢复。
 - (3) 不能按数据库中的表或某个用户进行恢复。
 - (4) 备份时数据库仍然可用。
 - (5) 不能出错，否则可能会引起数据库无法恢复。
 - (6) 可对几乎所有数据库实体做恢复，速度快。
- (1)、(3) 次序无关；(4)、(6) 次序无关。

【问题4】

银行系统如果需要在支持数据库的实时恢复或秒级恢复，需要采用冷备份和热备份

相结合的方式来进行备份。基本的备份策略如表 12-5 所示。

表 12-5 基本的备份策略

时 间	备份方式
星期一	冷备份
星期二	热备份
星期三	热备份
星期四	热备份
星期五	热备份
星期六	热备份
星期日	热备份

冷备份与热备份的交替间隔时间可以根据自身特点来进行调整。

当数据库出现故障，需要恢复时，基本的步骤如下：

- ① 恢复最近一次的冷备份数据。
- ② 按照时间顺序，恢复热备份中备份的数据和日志文件。
- ③ 找到机器上未损坏的最后一个日志文件，将数据库恢复到该日志文件对应的时间点。

12.2.9 数据中心建设

某大型企业在全国各城市共有 40 个左右的分支机构，这些机构已经建设了基于大型关系数据库的信息管理系统，每天负责独立地处理本区域内的业务，并实时存储业务数据。

每个机构的数据库服务器从 PC Server（最小 512MB 内存）到 RISC 小型机（最大 4GB 内存）配置不一，操作系统也不一样，数据库产品类型也不统一。

每个机构平均每天产生约 4MB 的数据，数据表中包括 TEXT 字段；每季度一次性产生约 100MB 的统计和报表数据。

为了加强管理，实现对下属机构业务数据的异地存储备份，保证数据的安全及恢复，同时对全国业务数据进行挖掘分析，该企业拟在总部建设数据中心。数据中心通过公共广域网与各个分支机构组成 VPN，每个分支机构到数据中心的逻辑信道带宽为 128kbps。

【问题 1】

(1) 为保证数据上传的顺序、稳定、安全、并发，并解决数据库的异构问题，系统应采用下列哪种技术（请在下列 A、B、C、D 四个选项中选择）？

- | | |
|----------------|------------|
| A. 交易中间件技术 | B. Web 中间件 |
| C. Web Service | D. 插装技术 |

(2) 为保证分支机构可靠、高效地向数据中心汇总业务数据，避免单点故障，除了考虑广域网线路采用备份外，在数据中心还应采用什么技术？用 100 字以内文字说明。

【问题 2】

假设各个分支机构的历史数据已经通过某种方式顺利地存储到了数据中心。对于增量数据的汇总更新，总部的王副总工提议采用数据库复制技术和数据库的触发器技术获取数据更新日志的方式来完成，但张总认为这样做不仅增加了成本，而且不能解决全部问题。请用 300 字以内文字说明张总的理由。

【问题 3】

最后，为解决数据中心数据更新与同步问题，张总安排总部的系统分析师李工来设计方案，并指出可以对王副总工的意见加以完善，如果需要也可以修改各地的业务系统。李工提出的方案具有较好的可执行性，并获得了大家的认可。请用 400 字以内文字叙述系统分析师李工的方案要点。

试题分析

现在的数据中心与早期的数据中心相比，因采用了数据仓库、关系数据库、网络冗余、负载均衡、存储冗余、中间件等技术，不仅能够对整个业务系统的异地数据实行集中备份，满足数据安全及故障恢复要求，满足信息系统数据的一致性、完整性的要求，并且其支撑的业务系统具有高可靠性、高可用性、高安全性、高实时性等特点，企业还可以对异地机构的业务进行全面、有效、实时的管理和监控。

问题 1 主要考查在异构环境下，如何利用现有的成熟技术来解决各个分支机构的数据上传到数据中心的问题。

① 交易中间件技术。中间件是介于应用系统和系统软件之间的一类软件，它使用系统软件所提供的基础服务，衔接网络上的应用系统，能够达到资源共享、功能共享的目的。交易中间件是面向交易事务处理的中间件，它负责正确传递交易，对交易完整性进行管理，调度应用程序的运行，保证整个系统运行的高效性。根据 X/Open DTP 模型，本地交易的管理由数据库系统来完成，交易中间件主要完成分布式计算环境下交易的监控和管理。数据库虽然在本地交易管理的功能已经很强大，但通过广域网进行分布式交易管理并不是数据库的强项，特别是一笔交易涉及异构数据库时更是如此。

把数据上传看做是交易处理，交易中间件可以满足问题 1 的要求。有关中间件的更加详细的内容，请读者阅读《系统分析师技术指南》。

② Web 中间件即应用服务器，是实现基于 Web 应用的中间层平台，不仅是业务逻辑组件的容器，还提供了内容管理、负载均衡、容错、连接池等功能。有关应用服务器的更加详细的内容，请阅读《系统分析师技术指南》。

③ Web Service 是一个崭新的松散耦合的分布式计算模型。每个服务节点利用 XML、SOAP、UDDI、WSDL 和 WSFL 等标准，对外提供某类功能，这些服务功能可以通过绑定 HTTP 和 SOAP 等技术来访问，从外部使用者的角度来看，Web Service 是一种部署在 Web 上的对象或组件。它是一种面向服务的模型，适合解决分布式计算和处理，不适合应用于数据同步。有关 Web Service 的更加详细的内容，请阅读《系统分析师技术指南》。

④ 插桩技术是白盒测试技术的一种。在被测系统中植入测试代码或测试硬件，以

捕获系统运行时的可能路径,这种方法称为插桩。插桩技术主要用于系统的覆盖分析。

为避免数据中心的单点故障,数据中心数据库服务器采用多机集群 Cluster 和数据库并行处理技术,存储设备采用全冗余的 SAN 结构(或者全冗余的存储结构)。

问题 2 主要考查采用数据库复制技术和数据库触发器技术解决增量数据的汇总有什么缺点或不足。问题涉及了数据库系统的两个概念:数据库复制技术和数据库触发器技术。

数据库复制技术是分布式数据库技术中的一项重要技术,其主要功能是向分布式环境中的各个数据库节点提供数据复制,保持数据的一致性或者控制与维护数据冗余。数据复制一般有同步、异步方式,体系机构有主从(一个主点,多个复制点)、对等、分布式(多个主点,一个复制点)等。采用数据库复制技术,需要各个节点安装数据库复制服务器,在异构环境下,还要求复制点增加异构数据同步服务,开销较大。

数据库触发器技术使数据库管理系统能够在一些数据库相关事件发生时触发预先定义的操作,触发器机制可以增强引用完整性,加强复杂业务的规则,或者监控数据库的变动,并执行一定的数据操作,实现信息管理的自动化。

题目中给出了这个企业现有的信息系统环境:一个通过广域网连接起来的数据库异构的、操作系统异构的分布式系统,而且有些分支机构的数据服务器配置较低。从给出的具体数据来看,该企业的分支机构业务繁忙,每天的数据量较大,并且数据中还包括 TEXT 字段(可存储 Word 文档等),每季度还产生相当可观的统计、报表数据(由数据库原始数据生成)。

从题目的要求来看,采用的技术一方面不能影响业务系统,另一方面数据不能因为网络故障、服务器繁忙等原因丢失,而且成本不能太高。

采用数据库复制技术,各地需要安装专门的复制服务器,增加成本,维护管理较为复杂,同时,太多分支机构使得中心的数据库复制服务器压力大,因此在本题要求的条件下,不宜采用数据库复制技术。

采用数据库的触发器技术虽然能够实时记录数据库的数据变化,但不能捕获数据表中 TEXT 字段的 UPDATE 动作,并且对于每季度一次的统计报表数据,采用数据库的触发器技术来记录数据库的变化,占用数据库资源太多,可能影响某些机构的日常业务处理。

问题 3 主要考查如何解决问题 2 所涉及的缺点或不足。在问题 3 的说明中,特别提到了如果需要则可以修改业务系统。

为了解答这个问题,我们首先要明确增量数据更新的过程。

各地上传的数据更新日志是中心数据更新同步数据的依据。各地系统将数据更新操作以 SQL 语句的形式记录到数据更新日志中,由交易中间件系统发送该日志到数据中心的交易中间件服务器上。中心的交易中间件服务器开启同步服务进程,实时处理各个机构提交的数据更新日志,必要时对由于数据库异构产生的 SQL 语句差别进行修改,向中心数据库系统提交数据更新 SQL,完成数据中心的数据更新。

其次,我们还要明确数据更新日志获得的方式,重点解决 TEXT 字段问题和报表统计数据问题。

① 针对多数业务数据的更新，各地数据库采用触发器技术，通过触发器捕获记录或字段的增删改操作，以标准的 SQL 命令保存到数据更新日志中。

② 改造各地原有业务系统，当发生数据表 TEXT 字段修改时，在修改字段的同一事务中，将该动作增加到数据更新日志中，数据中心根据记录抽取该字段指向的内容。

③ 对每季度产生的报表统计数据，改造各地原有业务系统，在数据更新日志中保存生产数据的条件，数据中心根据记录一次性抽取满足条件的数据。

④ 针对个别机构数据库服务器配置较低，采用触发器技术可能造成资源不足的情况，升级该机构的数据库服务器。

试题参考答案

【问题 1】

(1) 交易中间件技术。

(2) 数据中心数据库服务器采用多机集群 Cluster 和数据库并行处理技术，存储设备采用全冗余的 SAN 结构（或者全冗余的存储结构）。

【问题 2】

采用数据库复制技术，各地需要安装专门的复制服务器，增加成本，维护管理较为复杂，同时，太多分支机构使得中心的数据库复制服务器压力大，因此在本题要求的条件下，不宜采用数据库复制技术。

采用数据库的触发器技术虽然能够实时记录数据库的数据变化，但不能捕获数据表中 TEXT 字段的 UPDATE 动作，并且对于每季度一次的统计报表数据，采用数据库的触发器技术来记录数据库的变化，占用数据库资源太多，可能影响某些机构的日常业务处理。

【问题 3】

① 针对多数业务数据的更新，各地数据库采用触发器技术，通过触发器捕获记录或字段的增删改操作，以标准的 SQL 命令保存到数据更新日志中。

② 改造各地原有业务系统，当发生数据表 TEXT 字段修改时，在修改字段的同一事务中，将该动作增加到数据更新日志中，数据中心根据记录抽取该字段指向的内容。

③ 对每季度产生的报表统计数据，改造各地原有业务系统，在数据更新日志中保存生产数据的条件，数据中心根据记录一次性抽取满足条件的数据。

④ 针对个别机构数据库服务器配置较低，采用触发器技术可能造成资源不足的情况，升级该机构的数据库服务器。

12.2.10 分布式数据库系统

随着传统的数据库技术的成熟和计算机网络技术的发展，分布式数据库系统的研究与开发受到人们越来越多的关注。分布式数据库支持数据独立性和分布透明性。用户不必关心数据的逻辑分区，不必关心数据物理位置分布的细节，也不必关心副本的一致性问題。

【问题 1】

请用 200 字以内的文字叙述分布式数据库的主要特性和优缺点。

【问题 2】

在数据库中，某个业务表中的数据量很大，急速膨胀，在这样的情况下，为了保证高的数据响应速度，根据数据的(1)和(2)原则，可以对数据表进行分片设计。一般有两种分片方法：(3)和(4)。

请在空(1)、(2)、(3)和(4)处填写恰当的内容并请用 200 字以内的文字叙述数据表分片满足的条件并给出简要说明。

【问题 3】

分布式数据库中各局部数据库应满足集中式数据库的基本需求，除此以外还应保证数据库的全局数据(5)、并发操作的(6)和故障的全局(7)。

请在空(5)、(6)和(7)处填写恰当的内容并请用 300 字以内的文字简述保持数据一致性的方法。

试题分析

分布式数据库在逻辑上是一个统一的整体，在物理上则是分别存储在不同的物理节点上。分布式数据库具有以下几个特点：

① 数据独立性与位置透明性。数据独立性是数据库方法追求的主要目标之一，分布透明性指用户不必关心数据的逻辑分区，不必关心数据物理位置分布的细节，也不必关心重复副本（冗余数据）的一致性问题，同时也不必关心局部场地上数据库支持哪种数据模型。分布透明性的优点是明显的。有了分布透明性，用户的应用程序书写起来就如同数据没有分布一样。当数据从一个场地移到另一个场地时不必改写应用程序。当增加某些数据的副本时也不必改写应用程序。数据分布的信息由系统存储在数据字典中。用户对非本地数据的访问请求由系统根据数据字典予以解释、转换、传送。

② 集中和节点自治相结合。在集中式数据库中，为了保证数据库的安全性和完整性，对共享数据库的控制是集中的，并设有 DBA 负责监督和维护系统的正常运行。在分布式数据库中，数据的共享有两个层次：一是局部共享，即在局部数据库中存储局部场地上各用户的共享数据。这些数据是本场地用户常用的；二是全局共享，即在分布式数据库的各个场地也存储可供网中其他场地的用户共享的数据，支持系统中的全局应用。因此，相应的控制结构也具有两个层次：集中和自治。分布式数据库系统常常采用集中和自治相结合的控制结构，各局部的 DBMS 可以独立地管理局部数据库，具有自治的功能。同时，系统又设有集中控制机制，协调各局部 DBMS 的工作，执行全局应用。

③ 支持全局数据库的一致性和可恢复性。分布式数据库中各局部数据库应满足集中式数据库的一致性、可串行性和可恢复性。除此以外还应保证数据库的全局一致性、并行操作的可串行性和系统的全局可恢复性。这是因为全局应用要涉及两个以上节点的数据。因此，在分布式数据库系统中，一个业务可能由不同场地上的多个操

作组成。这些操作要比集中式数据库复杂和困难得多，分布式数据库系统必须解决这些问题。

④ 复制透明性。用户不用关心数据库在网络中各个节点的复制情况，被复制的数据的更新都由系统自动完成。在分布式数据库系统中，可以把一个场地的数据复制到其他场地存放，应用程序可以使用复制到本地的数据在本地完成分布式操作，避免通过网络传输数据，提高了系统的运行和查询效率。但是对于复制数据的更新操作，就要涉及对所有复制数据的更新。

⑤ 易于扩展性。在大多数网络环境中，单个数据库服务器最终会不满足使用。如果服务器软件支持透明的水平扩展，那么就可以增加多个服务器来进一步分布数据和分担处理任务。

分布式数据库的主要优点如下：

- ① 具有灵活的体系结构。
- ② 适应分布式的管理和控制机构。
- ③ 经济性能优越。
- ④ 系统的可靠性高、可用性好。
- ⑤ 局部应用的响应速度快。
- ⑥ 可扩展性好，易于集成现有系统。

分布式数据库的主要缺点如下：

- ① 系统开销大，主要花在通信部分。
- ② 复杂的存取结构，原来在集中式系统中有效存取数据的技术，在分成式系统中都不再适用。
- ③ 数据的安全性和保密性较难处理。

数据分片将数据库整体逻辑结构分解为合适的逻辑单位（片段），然后由分配模式来定义片段及其副本在各场地的物理分布，其主要目的是提高访问的局部性，有利于按照用户的需求组织数据的分布和控制数据的冗余度。

数据分片应遵循的准则如下：

- ① 完整性。全局关系的所有数据都必须分配到各个片段中，不允许某些数据属于全局关系但不属于任何片段。
- ② 重构性。分裂后各个片段可以重构原来的全局关系。
- ③ 不相交性。全局关系中的每个元组仅属于一个片段，不能在多个片段中重复出现。此规则不是必需的，因为在有冗余 DDBS 中数据可有多个副本。但片段中的部分元组重复将会使数据的更新操作变得复杂，为简化操作控制，片段间一般是不相交的。

分片的方式有多种，水平分片和垂直分片是两种基本的分片方式，混合分片和导出分片是较复杂的分片方式。

① 水平分片：把一个全局关系中的元组分裂成多个子集，每个子集为一个片段。分片条件由关系中的属性值表示。对于水平分片，重构全局关系可通过关系的并操作实现。

② 垂直分片：把一个全局关系按列分裂成多个子集，应满足不相交性（关键字除外）。对于垂直分片，重构全局关系可通过连接运算实现。

③ 导出分片：它也是一种水平分片，但分片的限定条件不是该关系中的属性，而是由与该关系有联系的其他关系导出的。

④ 混合分片：是在分片中采用水平分片、垂直分片和导出分片 3 种形式的混合。

问题 3 是 3 个填空题。这个问题在前面已经给出了答案：全局的一致性、可串行性和可恢复性。

试题参考答案

【问题 1】

分布式数据库具有物理分布性、位置自治性与协作性，支持数据独立性、集中与自治相结合的控制、适度数据冗余度、分布的事务管理。

优点包括具有灵活的体系结构、分布式的管理和控制、系统的高可靠性和可用性、高扩展性、局部处理、响应速度快、经济性能优越。

缺点包括系统开销通信较大、复杂的存取结构、保持数据一致性算法复杂、数据的安全性和保密性要求高。

【问题 2】

(1) 时间局部性

(2) 空间局部性

(3) 水平分片

(4) 垂直分片

数据表分片满足的条件如下：

完备性条件：必须把全局关系的所有数据映射到片段中，决不允许有属于全局关系的数据却未被映射到任何一个片段。

可重构条件：必须保证能够由同一个全局关系的各个片段来重建该全局关系。对于水平分片可用并操作重构全局关系；对于垂直分片可用连接操作重构全局关系。

不相交条件：要求一个全局关系被分割后所得的各个数据片段互不重叠（对垂直分片的主键除外）。

【问题 3】

(5) 一致性

(6) 可串行性

(7) 可恢复性

保持数据库一致性的方法，要点如下：数据副本的一致性、保证分布式事务的 ACID 属性、故障恢复的一致性。

12.2.11 Web 2.0 技术

Web 2.0 是目前 Internet 上 Web 应用的新模式，其概念由 Tim O'Reilly 于 2003 年

首先提出。目前 Internet 上已经有很多基于 Web 2.0 的 Web 应用系统, 如 Facebook、Wikipedia、Blogger、del.icio.us、YouTube 和 Flickr 等, 其应用涵盖各个方面。

【问题 1】

请用 200 字以内的文字叙述基于 Web 2.0 的 Web 应用与传统 Web 应用的重要不同点, 并给出简要解释(列举五个方面的不同点即可)。

【问题 2】

基于 Web 2.0 的 Web 应用十分强调用户的体验与感受, Ajax 是目前广泛使用的表现层技术。请用 200 字以内的文字简要介绍 Ajax 的概念和核心思想, 并说明在使用 Ajax 时, 从效率方面考虑需要注意的问题(列举 3 个即可)。

【问题 3】

某房地产信息网站欲整合 Google 提供的地图信息(Google Maps)和 Craigslist(美国最大的分类广告网站)提供的房地产信息, 为用户提供新的房地产信息增值服务。负责网站建设的工程师提出了两种解决方案, 第一种方案拟采用传统的应用系统集成方法, 集成来自不同源的数据。另外一种采用基于 Web 2.0 的 Mashup 技术, 从不同的数据源抽取数据, 聚合并转换数据, 在不同的上下文使用, 具体如表 12-6 所示。请阅读并完成表中空白部分(标号(1)~(8)处)。

表 12-6 方案比较

考虑因素 \ 解决方案	传统的系统集成方案	基于 Web 2.0 的 Mashup 技术
对现有平台与系统的要求	需要传统平台 API 层的支持, 如 Java 平台、JavaEE 和 .NET 等, 集成过程复杂	仅需要用 (1) 构建应用程序, 集成过程相对简单。
新特性的支持与功能扩展	需要采用编程的方式完成系统功能扩展, 涉及数据存储层、业务逻辑层和表现层	直接使用 Mashup 技术 (2), 创建新的特性与功能
表现层支持	传统集成方式下的表现层大多采用 (3), 当客户端发送请求时, 需要 (4)	使用 Ajax 技术调用 (5), 浏览器客户端不需要在每次与服务器通信时都重新加载整个页面, 动态特性强
底层集成技术支持	使用传统的多层企业应用集成技术, 涉及不同的集成层次	采用 (6) 思想集成底层系统, 强调 (7), 以 (8) 形式集成并暴露有系统的能力

试题分析

这是一道关于 Web2.0 和 Ajax 方面的试题, 共 3 个问题。Web 2.0 是最近几年谈得很多的一个话题, 我们可以凭常识解答本题。

第 1 个问题考查 Web 2.0 的应用与传统 Web 应用有什么区别。因为这个问题最近谈得多, 所以就相对简单了。例如, 传统 Web 应用通过浏览器浏览大量网页, 而 Web 2.0 的应用除了网页外, 加上很多通过 Web 分享的其他内容, 更加互动; 传统 Web 应用的模式为读, 而 Web 2.0 的应用的模式为写和贡献; 传统 Web 应用的主要内容单元为网页, 而 Web 2.0 的应用的主要内容单元为帖子/记录; 传统 Web 应用的形态为静态, 而 Web 2.0 的应用形态为动态; 传统 Web 应用的浏览方式为浏览器, 而 Web 2.0 的应用的浏览方式为浏览器、RSS 阅读器和其他工具; 传统 Web 应用的内容创建者为

网站编辑，而 Web 2.0 的应用的内容创建者可以是任何人，等等。

第 2 个问题是关于 Ajax 的，涉及 Ajax 的概念和核心思想。

Ajax 是 Asynchronous JavaScript and XML（以及 DHTML 等）的缩写，用来描述一组技术，它使浏览器可以为用户提供更为自然的浏览体验。借助于 Ajax，可以在用户单击按钮时，使用 JavaScript 和 DHTML 立即更新用户界面，并向服务器发出异步请求，以执行更新或查询数据库。当请求返回时，就可以使用 JavaScript 和 CSS 来相应地更新用户界面，而不是刷新整个页面。最重要的是，用户甚至不知道浏览器正在与服务器通信，Web 站点看起来是即时响应的。

Ajax 是由几种蓬勃发展的技术以新的方式组合而成的。Ajax 包含以下技术：

- ① 基于 XHTML 和 CSS 标准的表示。
- ② 使用文档对象模型（Document Object Model）进行动态显示和交互。
- ③ 使用 XML 和 XSLT 做数据交互和操作。
- ④ 使用 XMLHttpRequest 进行异步数据接收。
- ⑤ 使用 JavaScript 将上述技术绑定在一起。

Ajax 的核心是 JavaScript 对象 XMLHttpRequest。该对象在 Internet Explorer 5.0 中首次引入，它是一种支持异步请求的技术。简而言之，XMLHttpRequest 使你可以使用 JavaScript 向服务器提出请求并处理响应，而不阻塞用户。

传统的企业应用集成方案一般需要现有系统平台 API 层的支持，如 Java 平台、JavaEE 和 .NET 等，集成过程涉及面较广，相对复杂。而基于 Web 的应用与数据集成时，仅需要使用由现有 Web 应用程序（如 Google Maps）公开的基于 Web 的 API（或者 Web 服务）来构建应用程序，避免了复杂的集成过程。

因此，（1）处应该填写“仅需要使用现有 Web 应用程序公开的，基于 Web 的 API（或 Web 服务）构建应用程序，集成过程相对简单”。基于 Web 2.0 和 Mashup 技术，可以直接 Mashup 两个或者更多的 Web API，以创建某种新的特性与功能，因此在（2）处应该填写“使用 Mashup 技术集成两个或者更多的 Web API，创建新的特性与功能”。对于（3）和（4）传统集成方式下的表现层大多采用静态展现技术，当客户端发送请求时，往往需要刷新整个网页。对于（5），传统集成方式的客户端大多采用静态展现技术，当客户端发送请求时，往往需要刷新整个网页。对于（6）、（7）、（8），基于 Web 2.0 和 Mashup 技术的集成方案使用 SOA 的思想集成底层系统，强调功能暴露与服务组合，以服务的形式集成并暴露现有系统的能力。

试题参考答案

【问题 1】

基于 Web 2.0 的 Web 应用与传统 Web 应用的重要不同点如下：

- ① 互联网作为平台，使用浏览器和网络服务，为数量巨大的客户提供服务。
- ② 利用集体智慧，鼓励用户参与与贡献，聚集大众智慧。

- ③ 数据是下一个 Intel Inside, 数据管理已经成为 Web 2.0 应用的核心。
- ④ 软件发布周期的终结, 新版本的快速发布和用户参与的开发模式。
- ⑤ 轻量级编程模型, 支持松散结合的轻量级编程模型, 考虑聚合而不是协调。
- ⑥ 软件超越单一设备。
- ⑦ 丰富的客户体验, 强调可供编程性和丰富的用户体验。

【问题 2】

(1) Ajax 并不是一种全新的技术, 而是几种现有技术的组合:

- ① XHTML 和 CSS。
- ② 使用文档对象模型 (Document Object Model) 作动态显示和交互。
- ③ 使用 XML 和 XSLT 做数据交互和操作。
- ④ 使用 XMLHttpRequest 进行异步数据接收。
- ⑤ 使用 JavaScript 将它们绑定在一起。

(2) 在使用 Ajax 时, 从性能方面考虑, 需要注意以下几个问题:

- ① 替换部分而不是全部 HTML 网页。
- ② 不要在没有必要的情况下使用轮询计时器。
- ③ 根据不同的情况, 考虑传递不同格式的数据 (JavaScript、XML、HTML 和 XHTML 等)。
- ④ 充分利用客户端的动态特性和计算能力减轻服务端负载。

【问题 3】

- (1) 使用现有 Web 应用程序公开的, 基于 Web 的 API (或 Web 服务)。
- (2) 集成两个或者更多的 Web API。
- (3) 静态展现技术。
- (4) 刷新整个网页。
- (5) 基于 Web 的 API。
- (6) SOA。
- (7) 功能暴露与服务组合。
- (8) 服务。

12.2.12 虚拟化技术

希赛公司计划建设一个企业数据中心, 该数据中心支持 A、B、C、D 和 E 这 5 项业务, 各业务完全独立运行, 各项业务运行的操作系统均不相同。在单台某型服务器上, 除了为系统开销预留 30% 的 CPU 占用率之外, 各业务在繁忙时段 (白天) 及

非繁忙时段（夜间）的 CPU 占用率如表 12-7 所示。

表 12-7 各业务的 CPU 占用率

业务类型	A	B	C	D	E
繁忙时段	30%	35%	45%	45%	45%
非繁忙时段	15%	20%	40%	33%	40%

现希赛公司拟全部采用该类型服务器，张工和李工据此提出了不同的设计方案。张工认为需要采购 5 台服务器，每项业务分配一台服务器。但李工提出了一种基于虚拟化技术的解决方案，采用的服务器数量少于 5 台。

【问题 1】

在数据中心中应用虚拟化技术会带来哪些收益？请至少给出 3 项。

【问题 2】

李工的方案需要配置几台服务器？应如何设计？

【问题 3】

为了进一步节省电力，按照李工的方案，非繁忙时段最少只需几台服务器工作即可满足业务负载要求？应如何设计？

试题分析

在计算机领域，虚拟化通常是指计算元件在虚拟的（而不是真实的）基础上运行。虚拟化技术可以扩大硬件的容量，简化软件的重新配置过程。CPU 的虚拟化技术可用单 CPU 模拟多 CPU 的并行处理过程，允许一个平台同时运行多个操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。

虚拟化技术与多任务及超线程技术完全不同。多任务是指在一个操作系统中多个程序同时并行运行，在虚拟化技术中可以同时运行多个操作系统，并且每一个操作系统中都有多个程序在运行，每一个操作系统都运行在一个虚拟的 CPU 或者是虚拟主机上；而超线程技术只是单 CPU 模拟双 CPU 来平衡程序的运行性能，这两个模拟出来的 CPU 不能分离，只能协同工作。

虚拟化技术可以有不同的实现方法，如完全虚拟化、准虚拟化、操作系统层虚拟化等。

最流行的虚拟化方法使用名为 Hypervisor（虚拟机管理程序）的一种软件，在虚拟服务器和底层硬件之间建立一个抽象层。在完全虚拟化的环境下，Hypervisor 运行在裸硬件上，充当主机操作系统；而由 Hypervisor 管理的虚拟服务器运行客户端操作系统（Guest OS）。Hypervisor 可以捕获 CPU 指令，为指令访问硬件控制器和外设充当中介。因此，完全虚拟化技术几乎能让任何一款操作系统不用改动就能安装到虚拟服务器上，其主要缺点是给处理器带来的较大开销。

完全虚拟化是处理器密集型技术，因为它要求 Hypervisor 管理各个虚拟服务器，并让它们彼此独立。减轻这种负担的一种方法就是改动客户操作系统，让它以为自己运行在虚拟环境下，能够与 Hypervisor 协同工作，这种方法就叫准虚拟化（Pare-Virtualization）。

Xen 是开源准虚拟化技术的一个例子。操作系统作为虚拟服务器在 Xen Hypervisor 上运行之前,它必须在核心层面进行某些改变。因此,Xen 适用于 BSD、Linux、Solaris 及其他开源操作系统,而不适合对类似于 Windows 的专有操作系统进行虚拟化处理,因为它们无法改动。准虚拟化技术的优点是性能高。经过准虚拟化处理的服务器可与 Hypervisor 协同工作,其响应能力几乎不逊于未经过虚拟化处理的服务器。

实现虚拟化还有一个方法,就是在操作系统层面增添虚拟服务器功能。Solaris Container 就是这样的一个例子,Virtuozzo/OpenVZ 是面向 Linux 的软件方案。

就操作系统层的虚拟化而言,没有独立的 Hypervisor 层。相反,主机操作系统本身就负责在多个虚拟服务器之间分配硬件资源,并且让这些服务器彼此独立。一个明显的区别是,如果使用操作系统层虚拟化,所有虚拟服务器必须运行同一操作系统(不过每个实例有各自的应用程序和用户账户)。

虽然操作系统层虚拟化的灵活性比较差,但本机速度性能比较高。此外,由于架构在所有虚拟服务器上使用单一、标准的操作系统,管理起来比异构环境要容易。

当前虚拟化技术的优势已经得到了业界的普遍认可,虚拟化技术所具备的提高资源利用率和节能环保的特性也得到了越来越多高性能计算机用户的青睐,虚拟化技术的普及和推广成为信息化技术发展的必然趋势。据业内专家分析,3 个方面的因素使虚拟化技术迅速走向成熟:首先,信息化产业的快速发展使得服务器硬件技术有了巨大进步;其次,高端服务器所承载的软件应用环境已经逐步发展成熟;第三,也是最重要的一点,由于服务器技术的快速发展,以及信息化产业发展对于数据中心的规模要求越来越高,由此导致数据中心面临着越来越严重的发展瓶颈,而虚拟化由于具有提高资源利用率及节能环保、可进行大规模数据整合等特点而成为信息化技术发展的又一项具有战略意义的新技术。

【问题 1】

将虚拟化技术应用于企业数据中心将有效提高服务器的资源利用率,支持运行在不同操作系统之上的多个业务共享同一台服务器,从而减少企业数据中心中的服务器数量,最终降低企业数据中心建设的硬件成本,节约场地面积,减少能耗。

【问题 2】

尽管业务 A、B、C、D 与 E 运行的操作系统各不相同,但虚拟化技术支持同一台服务器上同时运行多个不同的操作系统实例,因而在不超出一台服务器 CPU 处理能力的条件下,多个业务在虚拟化的支持下能够共享同一台服务器。

【问题 3】

虚拟化技术中的业务负载迁移机制支持企业数据中心依据业务负载的涨落动态迁移业务负载,将轻载服务器上的业务迁移到其他服务器,然后这些服务器进入硬件休眠,进一步降低数据中心的能耗。

试题参考答案

【问题 1】

在数据中心中应用虚拟化技术将降低数据中心建设中的硬件成本,节约数据中心

占用的场地面积，减少能耗。

【问题 2】

李工的方案需要配置 4 台服务器，其中，业务 A 和 B 共享 1 台服务器，业务 C、D、E 各自配置 1 台服务器。

【问题 3】

按照李工的方案，非繁忙时段最少只需 3 台服务器工作即可满足业务负载要求，其中，业务 D 在非繁忙时段迁移到业务 A 与 B 工作的那台服务器上，业务 C 和 E 在非繁忙时段继续驻留在各自原来的服务器上。

12.2.13 嵌入式软件体系架构

希赛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 12-7 所示，并将软件设计工作交给了李工，要求其在 3 周内完成软件总体设计工作，给出总体设计方案。



图 12-7 宇航嵌入式设备软件架构

【问题 1】

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 2】

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 VxWorks 5.5 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 12-8 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 12-8 两种操作系统的主要差异

比较类型	VxWorks 5.5	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 3】

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。

试题分析

这是一道嵌入式系统架构试题，题目以宇航嵌入式设备的研制为背景。考查软件需求分析、软件架构、框架设计、可靠性设计及嵌入式操作系统相关的一些知识。题目所述的项目中提出了一种新型的架构 TLS，由于是项目中提出的架构，大家必然对这种架构感觉比较陌生，但从图中，我们可以很明显地看到，架构是分层的结构。这也就回到了我们熟悉的技术框架之内了。

首先看第 1 问，问题要求说明“公司制定的 TLS 软件架构的层次特点”，这需要从层次型架构的特点，以及题目已给出的说明两方面着手进行说明。层次型架构总的来说，特点有：下层为上层提供服务，层次与层次之间有着良好的独立性，结合题目的说明，可以总结出以下的特点：

- 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。

基于这种思想，TLS 结构框架实现了宇航设备中软件功能服务，这样考生就可以完整地叙述各层软件的设计内涵，即：

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。

接下来看第2问,该问考查嵌入式操作系统的相关知识。VxWorks 和 Linux 是两个常见的嵌入式操作系统,其特点如表 12-9 所示。

表 12-9 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	②支持多进程、多线程操作
实时性	③硬实时系统	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	支持	支持

通过比较,显然选用类似于 Linux 的嵌入式操作系统适用本题要求。理由包括3点:

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(这点是系统安全性要求),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。

最后看第3问,作为宇航系统的嵌入式设备,硬/软件故障是宇航系统最为关注的内容,根据宇航系统的特点和系统组成,故障一般分为3类,即硬件故障、应用软件故障和操作系统故障,在设计中,应考虑这3类故障的处理方法。硬件故障一般包括 CPU 运算错误、存储器访问/越界错误、MMU 配置错误、定时器计数错误和内总线错误等;应用软件故障一般包括计算越界、除 0、溢出和超时等各种异常情况;操作系统故障一般包括越权访问、死锁、资源枯竭、调度超时、配置越界和操作系统异常等。

一般情况下,宇航系统的故障分为瞬态故障和永久故障,瞬态故障是指偶然发生的错误,而永久故障是指发生后不可消失的错误。在容错系统中,故障一般是由瞬态故障向永久故障转变,将鉴别瞬态故障和永久故障的算法称为滤波算法。在嵌入式实时系统中,常用的滤波算法包括门限算法、递减算法、递增算法和周期滤波算法等。

故障的鉴别目的是实现宇航系统的容错与重构。常用的容错算法是 $N+1$ 备份、冷备、温备和热备。 $N+1$ 备份是指 N 个通用模块之一的任何一个模块发生故障后,将故障模块的任务迁移到备份模块运行;冷备、温备和热备是3种备份方式,可根据宇航系统的总体备份时间或重要程度安排不同的备份算法。

试题参考答案

【问题1】

TLS 结构框架的主要特点如下:

- 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如下：

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】

两种操作系统的差异如表 12-10 所示。

表 12-10 两种操作系统的主要差异

比较类型	VxWorks 5.5	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】

（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。

12.2.14 实时系统

希赛公司承担了一项嵌入式系统软件开发项目。该项目主要用于车载电子系统中监视发动机及燃油系统等系统工作状况，并通过综合仪表显示给驾驶员。经过多次与用户沟通，形成以下技术要求：

(1) 本项目的硬件平台由主处理机模块和多种接口模块组成，底板采用标准 VME 总线（硬件结构图见图 12-8），具体硬件模块配置如下：

① 主处理机模块（CPM）采用 PowerPC755，主频 266 MHz，配有 SDRAM 存储器和 FLASH 存储器；提供一个定时/计数器；支持 16 级中断和二级 Cache。

② 输入输出模块（IOC）支持 16 路 RS422 接口信号，传输速率不低于 115.2 kbps，IOC 模块与 CPM 模块的数据交换采用 64KB 双端口存储器。

③ 离散量接口模块（DAM）支持 64 路开关型离散量输入输出接口；2 路频率量输入；12 位 A/D 转换器和 12 位 D/A 转换器。

④ 图形处理模块（GPM）用于显示图形，支持 OpenGL 标准接口软件。

⑤ MBI 模块主要提供 1553B 外总线接口，PSM 为电源支持模块。

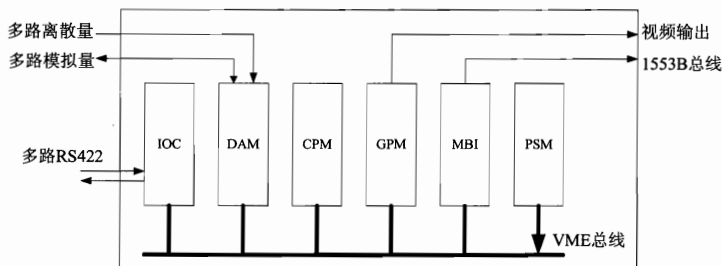


图 12-8 车载电子监视系统硬件结构图

(2) 本项目软件主要工作在 CPM 模块中，完成对外部设备的数据采集、分析和相应的控制，将监视结果以图形方式显示给驾驶员。该系统的软件主要包括外部接口驱动软件、VME 数据传输软件、处理软件、图形显示和外总线（1553B）数据交换软件。要求数据的采集必须确保每帧数据无丢失，并在本帧内完成数据的处理工作，本

帧信息显示给驾驶员的时刻最晚不能超过下一帧。详细的技术要求如下:

① 16路RS422接口主要完成对汽车燃油系统、动力系统和驾驶员命令的数据采集与控制。RS422数据传输格式将以32B为基本数据块,分别以20ms、40ms、60ms、1s这4个不同周期交换数据。

② 64路离散量数据主要监控发动机工作状态,要求每10ms采集一次,并在下一个10ms周期内将发动机状况显示到驾驶员座舱。

③ 多路模拟量数据主要采集发动机转速、油量及汽车的其他数据,为驾驶员监控汽车状态提供必要的量化数据。模拟量数据的刷新频率为1s。

④ 外总线(1553B)主要完成该系统与汽车其他电子系统的数据交换。

公司将本项目交给项目主管李工实施,要求李工按技术要求完成本项目的软件设计工作,公司根据合同关于“数据的采集必须确保每帧数据无丢失,并在本帧内完成数据的处理工作,本帧信息显示给驾驶员的时刻最晚不能超过下一帧”的要求,提醒李工设计中重点考虑整个系统的实时性问题。李工完成设计后,提交公司评审,会上就李工设计中存在的缺陷展开了激烈讨论,最终达成一致。

【问题1】

李工在设计IOC模块软件时指出:为了使CPM模块能够及时处理RS422数据,在IOC与CPM间的双端口存储器中为每个422通道设计一级缓冲,当某通道接收一个字节时,就将数据放入缓冲,由CPM接收(其结构见图12-9)。这样的好处在于每当有数据输入时,CPM模块可立即读取,而增大双口缓冲的目的是在CPM来不及处理时可防止数据的丢失。同时,IOC中的程序相对简单、实时性好、可以不考虑422通道的数据传输周期,只要按查询方式对16路422输入进行查询读取即可。

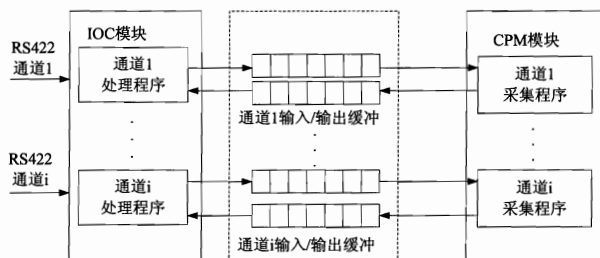


图 12-9 双口存储器工作示意图

会上,王工提出了强烈的反对意见,认为这种设计方法绝对不可取,这种方法只考虑了软件的简单与便利,而没有针对软件特点考虑问题,在设计中对实时性、负载平衡分配以及实现算法上均存在缺陷,必须改进。

(1) 你认为李工的设计在实时性、负载平衡分配和实现算法上存在怎样的缺陷?详细说明理由。

(2) 请给出IOC模块和CPM模块之间双口存储器数据交换方法。

【问题2】

如果CPM模块中采用了嵌入式实时操作系统,请根据本项目的需求,用300字

以内的文字说明 CPM 模块中数据采集任务应如何划分？系统对任务调度的最小调度周期如何计算？各个任务周期及优先级如何定义？

【问题 3】

请用 150 字以内的文字说明针对 IOC 模块的多路 RS422 接口的数据采集，其驱动程序使用中断方式或查询方式的主要差别是什么？主要优缺点有哪些？

试题分析

本题主要考查考生对以下实时系统基本技术知识的掌握。

① 因为嵌入式实时系统与实际应用要求结合性很高，为了满足系统总的要求，设计软件时，应充分考虑硬件平台的配置和系统的实际技术特点，掌握好这一点尤其重要。

② 实时系统必须保证在规定的时间内完成预先规定处理工作。而本题给出的多个数据采集、处理时间要求，是嵌入式实时系统设计中主要面临的设计问题。对仅有一个处理器而言，如何安排好多个任务协调处理的顺序，即任务的优先顺序，是考查考生综合设计能力最有效的方法。

③ 本题给出的硬件平台结构根据系统的处理量，按功能划分成了多个处理单元，考生应从题中可以隐含看出。那么，在软件设计中应充分发挥本题所给出的条件，学会使用负载平衡方法分配每个处理机所承担的任务，达到各个处理单元的负载基本相同，以增强系统的实时性。

【问题 1】

本问题主要考查考生对本题的理解能力和具体设计中如何充分发挥各自处理单元的能力。设计时应重点考虑 IOC 处理模块和主处理模块间的任务接口功能的划分问题。李工给出的方案不是不可行。只能说是不是最优方法。李工设计方案中的缺陷主要体现在如下两个方面：其一是系统实时性不是太好；其二是 IOC 和 CPM 工作负载不平衡，如果 CPM 能够有充分的处理能力（速度极快），完全可以按每个字节接收 16 路 RS422 数据，那么 IOC 就可以不用智能模块设计了，这样也可降低成本。但是，考生应清楚，这样的设计要求 CPM 要不断响应 IOC 模块的数据到达请求，势必使 CPM 消耗大量的时间处理此事务，而根据本题给出的技术说明“RS422 数据传输格式将以 32B 为基本数据块”，因此，李工的设计方法是不可取的。只有 CPM 模块按每帧 32B 一次接收完成，才可以大大降低 CPM 模块任务负载，确保 CPM 的大部分时间处理自己的工作。

【问题 2】

本问题主要考查考生针对本题提出的不同周期任务的周期要求，在采用操作系统情况下，如何设计 CPM 中不同任务的工作时序。这里主要考查考生对操作系统的掌握程度，理解操作系统中任务调度的抢占式优先级调度的基本原理，并根据具体任务的时间关系，画出任务的优先级分配图。这也是考核考生对实时系统的基础知识。

【问题 3】

查询方式是处理器直接利用 I/O 指令编程，实现数据的输入输出。处理器发出 I/O

命令, 命令中包含了外设的地址信息和所要执行的操作, 相应的 I/O 系统执行该命令并设置状态寄存器; 处理器不停地(定期地)查询 I/O 系统以确定该操作是否完成。由程序主动查询外设, 完成主机与外设间的数据传送, 方法简单, 硬件开销小。

中断方式是处理器利用中断方式完成数据的输入/输出, 当 I/O 系统与外设交换数据时, 处理器无须等待也不必去查询 I/O 的状态, 当 I/O 系统完成了数据传输后则以中断信号通知处理器。处理器然后保存正在执行程序的现场, 转入 I/O 中断服务程序完成与 I/O 系统的数据交换。再返回原主程序继续执行。与查询方式相比, 中断方式因为处理器无须等待而提高了效率。

试题参考答案

【问题 1】

(1) 从本题的第(2)项要求中的描述①可以看出, RS422 的数据是按 32B 为基本数据块进行传输的, 这说明数据块应是连续发送的。

① 实时性: 李工提出的在 IOC 接收到一个字节后就发送给 CPM 模块, 这不但影响了 IOC 的实时性, 同时也影响了 CPM 的实时性。

② 负载均衡分配: 从命题上可以看出 IOC 是智能设备, 可以独立完成 RS422 的数据采集, 按李工设计方法, IOC 的处理能力没有充分发挥, 而加大了 CPM 模块负载量, 使得两模块的负载不平衡。

③ 实现算法: 前两个设计缺陷的存在, 使得实现算法上软件的功能模块设计不完整、增大了 CPM 模块上的软件设计复杂度、没有有效地利用硬件资源。

(2) IOC 和 CPM 之间的数据交换设计办法

① 首先, 将 IOC 和 CPM 两模块的任务负载量划分均衡, 具体使 IOC 按每个 RS422 通道的 32B 为基本数据块进行接收, 一次统一提交给 CPM 模块。

② 在 IOC 模块中为每个通道设计 1~2 个 32B 缓冲区。IOC 在接到一个字节后, 将数据放入每个 32B 缓冲区, 当一个完整的数据块接收完成后, 一次性地将数据发送给 CPM 模块。

③ IOC 的软件可以使用中断方式或查询方式进行数据接收, 一旦数据块的头字节到达后, 使用查询方式连续将 32 个字节接收完成。

④ CPM 模块的通道采集程序可一次性读取完成的数据块, 这样, 降低了 CPM 的数据采集负载。

【问题 2】

根据本题第(2)条要求中对数据采集的技术要求, 按照采集任务周期划分, CPM 模块中的软件应划分成: 1 个 10ms 任务、1 个 20ms 任务、1 个 40ms 任务、1 个 60ms 任务、2 个 1s 任务和 1 个非周期任务。其中:

- 10ms 任务主要负责 64 路离散量数据的采集、处理和显示。
- 20ms 任务主要负责具有 20ms 数据交换周期的 RS422 通道的数据采集。
- 40ms 任务主要负责具有 40ms 数据交换周期的 RS422 通道的数据采集。

- 60ms 任务主要负责具有 60ms 数据交换周期的 RS422 通道的数据采集。
- 1s 任务有 2 个。一个任务主要负责具有 1s 数据交换周期的 RS422 通道的数据采集；另一个任务主要负责多路模拟量数据采集。
- 非周期任务主要负责该系统与汽车其他电子系统的数据交换。

系统对任务调度的最小调度周期应为所有任务周期的最大公约数，即 10ms、20ms、40ms、60ms、1s 的最大公约数是 10ms。

根据实时系统常用的小周期任务优先调度的算法。CPM 上 7 个任务的优先级顺序应为：

非周期任务 < 1s 任务 1 = 1s 任务 2 < 60ms 任务 < 40ms 任务 < 20ms 任务 < 10ms 任务
(优先级最低) \longrightarrow (优先级最高)

【问题 3】

(1) 主要差别

中断方式是在程序接收或发送每一个字节时，均产生中断信号，发中断主要通知处理器一个字节已经发送完成；接收中断主要通知处理器 RS422 接口中有一个字节数据达到。

查询方式主要用程序读取 RS422 接口的寄存器，判别接口是否有数据到达或接口发送缓冲区是否空。

(2) 优缺点

中断方式：及时响应数据、不会产生数据丢失、系统开销大、实现较复杂。

查询方式：软件实现简单、接收数据快、系统开销小、不能及时响应。

在实现时，可根据具体要求，将两种方式结合使用。

12.2.15 信息系统安全

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，不会发生信息泄露和数据损坏。
- ④ 防止来自于互联网上的各种恶意攻击。
- ⑤ 业务系统涉及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】

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 2】

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 3】

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。

试题分析

这是一道信息系统安全的试题。对于任何一个系统而言，安全性都是至关重要的，可以说一个系统的安全性如果得不到保障，那么该系统的功能越强大，造成的危害也就越大。

问题 1 是概念题，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露。
- 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 考到了认证方式，认证方式是大家平时接触得非常多的一项技术。如最简单的就是用户名/密码方式、持卡方式的 IC 门卡、生物技术认证方式的指纹、网上银行通常用例的数字证书等。

问题 3 属于技术应用型的题。给出了一些现象，让考生分析原因，分析缺陷。从描述来看，我们可以很明显得知系统缺乏安全审计的策略。而在安全体系中，审计占有非常重要的地位，安全审计系统可以帮助发现系统入侵和漏洞、帮助发现系统性能上的不足、为一些安全案件提供有效的追究证据。同时缺乏抗抵赖的机制。

试题参考答案

【问题 1】

- 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露。
- 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】

目前主要的认证方式有以下 3 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】

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。

12.3 实战练习题

试题 1

希赛公司欲针对 Linux 操作系统开发一个 KWIC（Key Word In Context）检索系

统。该系统接收用户输入的查询关键字，依据字母顺序给出相关帮助文档并根据帮助内容进行循环滚动阅读。在对 KWIC 系统进行需求分析时，公司的业务专家发现用户后续还有可能采用其他方式展示帮助内容。根据目前需求，公司的技术人员决定通过重复剪切帮助文档中的第一个单词并将其插入到行尾的方式实现帮助文档内容的循环滚动，后续还将采用其他的方法实现这一功能。

在对 KWIC 系统的架构进行设计时，公司的架构师王工提出采用共享数据的主程序-子程序的架构风格，而李工则主张采用管道-过滤器的架构风格。在架构评估会议上，大家从系统的算法变更、功能变更、数据表示变更和性能等方面对这两种方案进行评价，最终采用了李工的方案。

【问题 1】

在实际的软件项目开发中，采用恰当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，并对主程序-子程序和管道-过滤器这两种架构风格的特点进行描述。

【问题 2】

请完成表 12-11 中的空白部分（用+表示优、-表示差），对王工和李工提出的架构风格进行评价，并指出采用李工方案的原因。

表 12-11 王工与李工的架构风格评价

评价要素 \ 架构风格	共享数据的主程序-子程序	管道-过滤器
算法变更	-	(1)
功能变更	(2)	+
数据表示变更	(3)	(4)
性能	(5)	(6)

【问题 3】

如图 12-10 所示是李工给出的架构设计示意图，请将恰当的功能描述填入图中的 (1) ~ (4)。

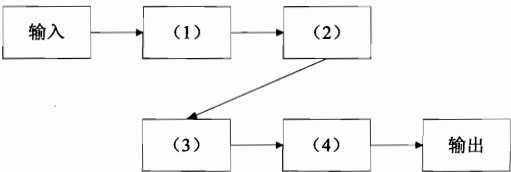


图 12-10 李工给出的架构示意图

试题 2

希赛软件公司承担了某项国家重点项目的研制工作，任务重点是参考国外汽车电子发展趋势，开发某汽车公司的企业汽车电子基础软件的架构，逐步实现汽车企业未

来技术发展规划。

该公司接收此项任务后，调动全体技术人员深入收集国外相关技术资料，经过多方调研和分析，公司提出遵照国际组织最新推出的 AUTOSAR 规范，按统筹规划、分步骤实施的原则，实现汽车公司的基础软件架构设计。图 12-11 给出了 AUTOSAR 规范所定义的工作包，图中灰色部分代表本项目工作所包含的内容，即软件架构和基础软件。

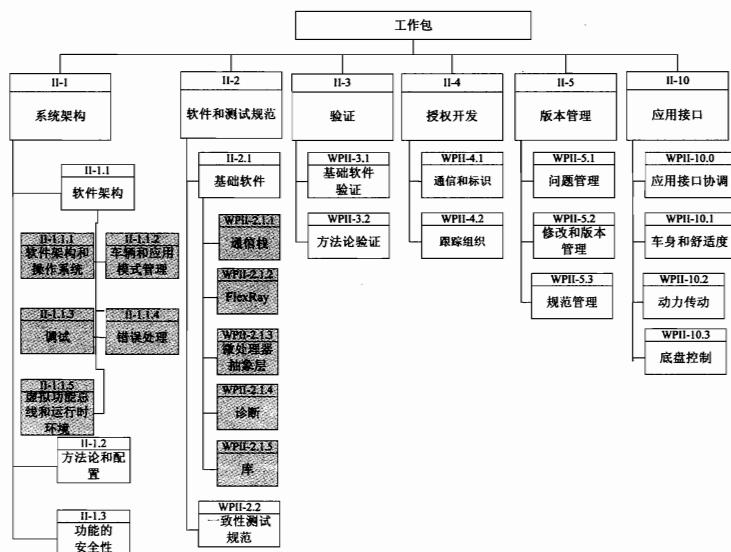


图 12-11 AUTOSAR 定义的工作包

【问题 1】

AUTOSAR 规范中要求，汽车电子软件开发流程应尽量满足并发、可多次迭代的特性。为了定义汽车电子的软件开发过程，公司李工和王工分别提出了两种软件开发流程，其开发流程如图 12-12 和 12-13 所示（图中 ECU 是指汽车电子中的电子控制单元）。请说明李工和王工谁定义的流程更符合 AUTOSAR 的规定，并说明理由。

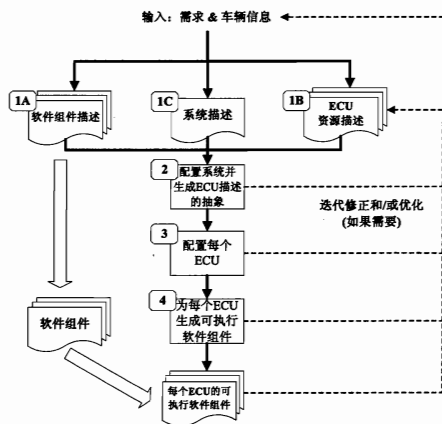


图 12-12 李工设计的流程

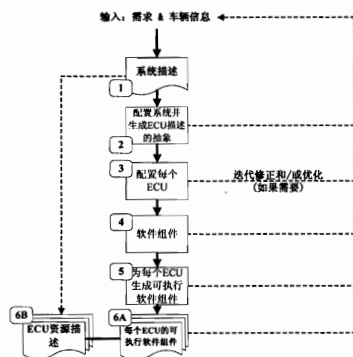


图 12-13 王工设计的流程

【问题 2】

图 12-11 中的 II-1.1.1 项中定义了软件架构和操作系统的要求，图 12-14 所示是满足 AUTOSAR 定义的操作系统各功能模块的层次结构，请说明 (1) ~ (5) 箭头所标的具体操作含义。

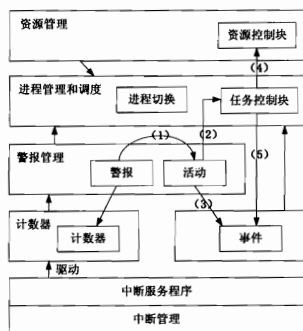


图 12-14 AUTOSAR 定义的操作系统结构

【问题 3】

AUTOSAR 是一种开放式架构，用 150 字以内的文字，说明采用 AUTOSAR 架构的主要优点，并说明汽车电子 ECU 覆盖汽车的哪 3 个领域。

试题 3

TeleDev 是一个大型的电信软件开发公司，公司内部采用多种商业/开源的工具进行软件系统设计与开发工作。为了提高系统开发效率，公司管理层决定开发一个分布式的系统设计与开发工具集成框架，将现有的系统设计与开发工具有效集成在一起。集成框架开发小组经过广泛调研，得到了如下核心需求：

① 目前使用的系统设计与开发工具的运行平台和开发语言差异较大，集成框架应无缝集成各个工具的功能。

② 目前使用的系统设计与开发工具所支持的通信协议和数据格式各不相同，集成框架应实现工具之间的灵活通信和数据格式转换。

③ 集成框架需要根据实际的开发流程灵活、动态地定义系统工具之间的协作关系。

④ 集成框架应能集成一些常用的第三方实用工具，如即时通信，邮件系统等。

集成框架开发小组经过分析与讨论，最终决定采用企业服务总线（ESB）作为集成框架的基础架构。

【问题 1】

ESB 是目前企业级应用集成常用的基础架构。请列举出 ESB 的 4 个主要功能，并从集成系统的部署方式、待集成系统之间的耦合程度、集成系统的可扩展性 3 个方面说明为何采用 ESB 作为集成框架的基础架构。

【问题 2】

在 ESB 基础架构的基础上，请根据题干描述中的 4 个需求，说明每个需求应该采用何种具体的集成方式或架构风格最为合适。

【问题 3】

请指出在实现工具之间数据格式的灵活转换时，通常采用的设计模式是什么，并对实现过程进行简要描述。

试题 4

某软件公司承担了某大型企业应用系统集成任务，该企业随着信息化的进展，积累了许多异构的遗产信息系统，这些系统分别采用 J2EE、.NET 等技术进行开发，分布在不同的地理位置，采用不同的协议进行数据传输。企业要求集成后的系统能够实现功能整合，并在组织现有功能的基础上提供增值服务。为了按时完成任务，选择合适的企业应用集成方法和架构非常重要。项目组在讨论方案时，提出了两种集成思路。

① 刘工建议采用传统的应用集成方法，将应用集成分为多个层次，并采用消息代理中间件连接遗产系统。

② 王工建议采用基于 SOA 的方法进行应用集成，将现有遗产系统采用 Web Service 的方式进行包装，暴露统一格式的接口，并采用企业服务总线（ESB）进行连接。

项目组仔细分析比较了两种方案的优点和不足后，认为刘工和王工的建议都合理，但是结合当前项目的实际情况，最后决定采用王工的建议。

【问题 1】

请分析比较两种方案优点和不足，完成表 12-12 中的空白部分。

表 12-12 两种方案的优点和不足

集成方案 考虑因素	刘工建议的集成方案	王工建议的集成方案
拟采取的集成方法	涉及不同的集成层次，集成方法复杂多样	(1)
对企业集成需求的符合程度	(2)	强调功能的暴露与服务的组合，便于提供增值服务
集成系统体系结构	(3)	基于总线结构的体系结构，系统的耦合度低
集成系统的可扩展性	遗产系统集成方法多样，系统耦合度高，可扩展性较差	(4)

【问题 2】

针对该企业的集成实际情况，请用 200 字以内的文字叙述王工建议中企业服务总线（ESB）应该具有的基本功能。

【问题 3】

王工的方案拟采用 Web Service 作为基于 SOA 集成方法的实现技术。请根据该系统的实际情况，用 300 字以内的文字说明系统应该分为哪几个层次，并简要说明每个层次的功能和相关标准。

试题 5

电子商务是构建在 Internet 上的新商务模式。企业和个人使用电子商务来降低交易成本，提高商品和信息流的周转速度，提高客户服务水平，并且使得制造商、供应商和客户之间可以紧密合作。企业信息化是实施企业电子商务的基础，特别是在企业流程再造、供应链管理、客户关系管理等方面需要依托信息技术进行整合和优化。

【问题 1】

一个完整的电子商务系统必须处理销售生命周期中消费者经历的多个阶段。请用 200 字以内的文字简要叙述有哪些阶段。

【问题 2】

供应链管理过程是企业的一个关键业务过程。请用 200 字以内的文字回答：供应链管理主要包括哪 3 个子过程？电子商务环境下，供应链管理过程的整合、优化有何意义？

【问题 3】

一个大型电子商务项目正处于建设方案征集、论证阶段，某系统集成商为了赢得客户的信任，需要提供一份建议方案文档，对客户的需求进行响应（包括问题 1、问题 2 和问题 3 所涉及的内容）。高质量的建议方案能够显示出集成商在处理客户 RFP（Request For Proposal）方面的能力、实力和专业性，而创建一个高质量的建议方案，需要调配众多的资源，按照计划执行。请用 300 字以内文字简要叙述如何创建一份高质量的建议方案文档。

试题 6

希赛公司是一家电子商务公司，其主要业务是在线购物，包括书籍、服装、家电和日用品等。随着公司业务规模不断增大，公司决策层决定重新设计并实现其网上交易系统，公司负责系统开发的王工和李工分别给出了两种不同的设计方案，如图 12-15 和 12-16 所示。

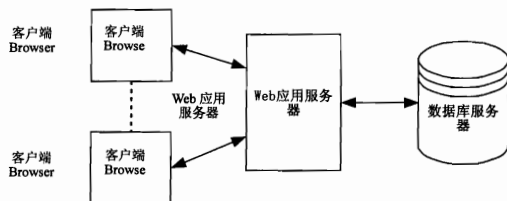


图 12-15 王工设计方案的体系结构设计示意图

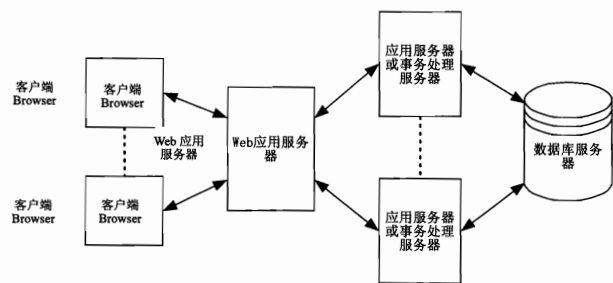


图 12-16 李工设计方案的体系结构设计示意图

公司的架构师和开发者针对这两种设计方案，从服务器负载情况、业务逻辑的分
离性、系统可靠性、实现简单性等方面进行讨论与评估，综合考虑最终采用了李工给
出的方案。

【问题 1】

请分析比较王工、李工两种方案的优点和不足，完成表 12-13 中的空白部分。

表 12-13 两种方案的比较

体系结构方案 评价因素	王工建议的体系结构方案	李工建议的体系结构方案
服务器负载	Web 服务器需要同时处理业 务逻辑与数据库访问，负担较重	(1)
业务逻辑的分离性	(2)	采用多个应用服务器专门进行业 务逻辑处理，做到业务逻辑与其他 代码分离
系统可靠性	采用单台 Web 服务器，整个 系统的可靠性较差	(3)
实现简单性	主要采用 JSP、ASP 等脚本语 言实现系统，比较简单	(4)

【问题 2】

对数据库的访问是该系统开发中需要特别注意的一个问题，O/R 映射是一种常用
的数据库访问编程技术。请用 200 字以内的文字说明 O/R 映射的含义，并指出采用
O/R 映射的 3 个主要好处。

【问题 3】

性能是 Web 应用系统的一个重要质量属性。请用 200 字以内的文字说明 3 个主要
影响 Web 应用系统性能的因素，针对每个因素提出解决方案以提高系统性能。

试题 7

traveler.com 是一家在线旅游信息服务公司，其主要业务是为自助旅游者提供关于
旅游线路及周边信息的服务。随着公司业务不断发展，公司用户要求提供基于位置
的增值旅游信息服务，即希望能够在给定位置（利用 GPS 全球定位系统获取）的情
况下得到周边的地理位置、住宿、餐饮和交通等旅游相关信息。针对该需求，公司技

术人员对现有系统的架构和运行模式进行了认真分析, 决定采用 Mashup (一种基于互联网的内容聚合技术) 集成来自其合作网站 (设为 A、B、C、D) 的信息, 满足用户的需求。具体实现方式如下:

- ① 利用 A 网站提供的地图信息, 得到用户位置相关的周边地理信息。
- ② B 网站根据用户的位置信息向其提供周边的住宿信息。
- ③ C 网站根据用户的位置信息向其提供周边的餐饮信息。
- ④ D 网站根据用户的位置信息向其提供周边的公交线路等信息。

【问题 1】

如图 12-17 所示是公司进行 Mashup 的流程示意图, 请阅读并补充图中数字标出部分的内容。

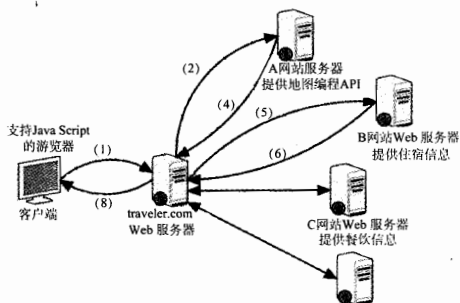


图 12-17 Mashup 流程示意图

- (1) 用户向 Traveler 网站请求服务, 请求页面提供用户的位置信息。
- (2) _____。
- (3) _____。
- (4) A 网站向 Traveler 网站返回用户所处位置周边的地图信息。
- (5) _____。
- (6) B 网站向 Traveler 网站返回用户所处位置周边的住宿信息。
- (7) _____。
- (8) Traveler 网站向用户返回用户所处位置周边整合的旅游信息。

【问题 2】

目前, 互联网上已经存在很多 Mashup 应用, 各大网站也纷纷提供了各种格式的 Mashup 数据接口。请用 300 字以内的文字给出 3 种目前经常使用的 Mashup 数据接口, 并加以简单说明。

【问题 3】

在实现 Mashup 应用时, 进行内容聚合的物理位置是一个十分重要的因素。目前很多 Mashup 站点都选择在客户端机器上进行内容聚合, 构成所谓的胖互联网应用程序

序 (Rich Internet Application, RIA)。请你用 200 字以内的文字说明在客户端进行内容聚合的优点。

试题 8

RMO 是一家运动服装制造销售公司,计划在 5 年时间内将销售区域从华南地区扩展至全国范围。为了扩大信息技术对于未来业务发展的价值,公司邀请咨询顾问帮助他们制订战略信息系统规划。经过评审,咨询顾问给出的战略规划要点之一是建立客户关系支持系统 (CRSS)。RMO 公司决定由其技术部成立专门的项目组负责 CRSS 的开发和维护工作。

项目组在仔细调研和分析了系统需求的基础上,确定了基于互联网的 CRSS 系统架构。但在确定系统数据架构时,张工认为应该采用集中式的数据架构,给出的理由是结构简单、易维护且开发及运行成本低;而刘工建议采用分布式的数据架构,并提出在开发中通过“局部数据库+缓存”的读写分离结构实现,具有较好的运行性能和可扩展性。

项目组经过集体讨论,考虑到公司的未来发展规划,最终采用了刘工的建议。

【问题 1】

请用 300 字以内的文字,说明张工和刘工提出的数据架构的基本思想。

【问题 2】

在刘工建议的基础上,为了避免 CRSS 系统的单点故障,请用 200 字以内文字简要说明如何建立 CRSS 的数据库系统;对于数据的读取、添加、更改和删除操作分别如何实现。

【问题 3】

RMO 公司销售区域将在未来 5 年大面积扩展,其潜在客户数量也会因此大幅度增加,所以良好的可扩展性是 CRSS 系统所必需的质量属性。请分别说明在集中式和分布式数据架构下,可以采用哪些方法提升系统的可扩展性。

试题 9

随着信息系统的深入应用,它在政府或企业中发挥着越来越重要的作用。由此也产生了一些问题:一旦由于故障、甚至灾难导致信息系统局部或全部瘫痪,就会对相关的政府或企业造成重大的损失,因此需要业务持续和灾难恢复规划。

业务持续和灾难恢复规划涉及一些特定的或相关的规划,当正常的信息处理业务突然中断时,用来减轻甚至避免其所带来的影响。它们用来保证维持组织运作的关键系统的可用性。

某大型网上书店通过 Internet 为用户提供网上图书查询及订购等多种服务。由于每天的业务量非常多,因此对业务有持续性要求,并且需要具备灾难恢复功能。在对原有系统的改进方案中提供:

- ① 对于一些关键应用系统,采用双机冗余热备的方式进行保护。
- ② 由于资金问题,公司并不打算建设自己专有的用于备份数据的备份中心(假

设这里的备份中心仅用于备份数据），而是决定租用其他公司提供的商业备份中心，每隔一段时间，把公司的业务数据备份到备份中心。

【问题 1】

请用 150 字以内文字，说明双机冗余热备方式主要解决的是系统运行中的哪些问题？在选择双机冗余热备产品时通常需要考虑哪些问题？

【问题 2】

请用 300 字以内文字，从技术方面说明备份中心的作用，以及在租用商业备份中心时的注意事项。

【问题 3】

请用 300 字以内文字，分析公司向备份中心备份数据的时间间隔的选取、公司日常业务系统的运行性能，以及在灾难发生时数据损失情况三者之间的关系。

试题 10

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的数据进行多次分析、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 12-18 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。

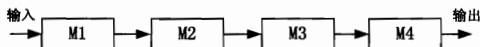


图 12-18 王工建议的软件系统模块示意图

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，假设各个模块的可靠度均为 0.99，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为 $0.99^4=0.96$ ，即整个软件系统的可靠度下降明显；第二，软件系统模块全部采用串联结构，一旦某个模块失效，则意味着整个软件系统失效。

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 12-19 所示。

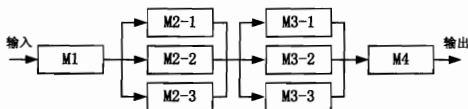


图 12-19 李工建议的软件系统模块示意图

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

【问题 1】

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】

请解释李工提出的动态冗余和 N 版本程序设计技术,给出如图 12-18 所示模块 M2 采用如图 12-19 所示的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】

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。

试题 11

甲公司承担了一项为宇航系统配套生产高性能嵌入式计算机系统的任务,用户要求该系统要具有高速并发处理能力、低功耗、高可靠性,并可以有效地防止系统故障的蔓延。根据用户对本项目的要求,甲公司成立了软/硬件两个项目组,总体设计由硬件组承担,负责高性能嵌入式计算机系统体系结构设计,软件组负责确定软件的技术需求和应用软件开发平台的软件设计工作。

在处理器选型方面,硬件组王工与软件组张工在讨论采用哪种 CPU 体系结构方面发生争议。目前,流行的处理器结构包括单核结构、多处理器结构、超线程结构、多核结构、共享 Cache 的多核结构和超线程技术的多核结构共 6 种,如图 12-20 所示。

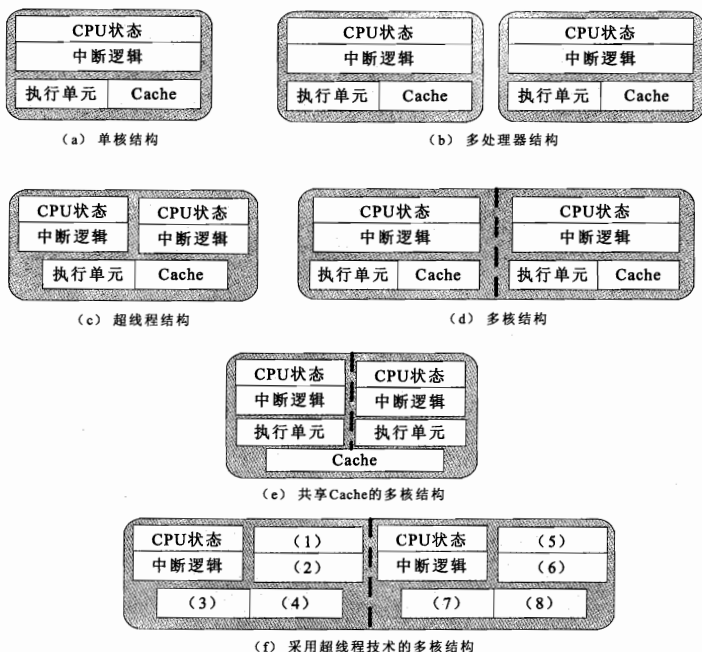


图 12-20 多种处理器结构

【问题 1】

王工提出,根据用户要求,本嵌入式系统应具有高速并行处理能力,采用多处理

器结构比较适合，主要理由是多处理器结构设计简单、可支持多个进程在不同处理器上并发处理；而张工提出，必须分清“多处理器结构”与“多核结构”的优点和缺点，多处理器结构虽然支持多进程的并发处理，但没有直接实现多线程并发执行；多核结构可以直接实现多线程并发执行。要提高应用的并行性就必须利用多个硬件资源的并行工作，建议采用超线程技术的多核结构的处理器。请填写图 12-20 (f) 中的 (1)~(8)，并用 300 字以内的文字对上述 6 种处理器结构的工作原理进行简要描述。

【问题 2】

在本项目中，如果计算机采用了多线程技术实现应用程序的并发操作，程序设计人员将面临多种挑战，其主要原因在于：程序会同时发生多个动作，对这些同时发生的动作及它们之间的交互进行管理将面临 4 方面的挑战，即同步、通信、负载均衡和可伸缩性，用 200 字以内的文字对上述 4 种挑战进行简要描述，并给出如表 12-14 所示的常用并发程序设计模式的分解方式。

表 12-14 常用并行程序设计模式的分解方式

模式	分解方式
任务级并行模式	(1)
分治模式 (Divide and Conquer)	(2)
几何分解模式	(3)
流水线模式	(4)
波峰 (Wavefront) 模式	(5)

【问题 3】

为了防止系统故障的蔓延，解决应用进程间的相互影响，在超线程技术的多核结构需要选择一种嵌入式操作系统作为本高性能嵌入式计算机系统的运行环境，请用 200 字以内的文字说明选择的操作系统应具备哪些主要功能。

试题 12

希赛公司最近签订了一份控制系统软件的研制合同。合同技术要求如下：

① 本控制系统的软件运行平台拟采用嵌入式实时操作系统，控制系统的工作周期为 10ms，要求控制软件能够精确地按事先预定的时间顺序完成数据的采集、处理和输出。

② 为了提高整个系统的可靠性，控制系统采用容错处理，即用两个相对独立的计算机承担相同的工作，通过交叉对比，实现控制的双余度管理；当任何一台计算机出现故障时，系统可以进行单余度控制。系统结构如图 12-21 所示。

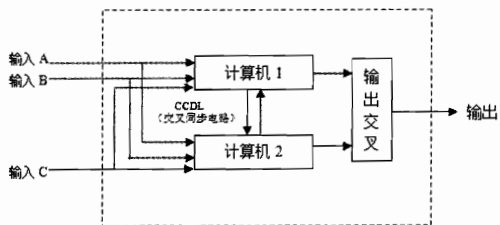


图 12-21 系统结构图

③在 10ms 工作周期内的时间分配如表 12-15 所示。

表 12-15 10ms 工作周期内的时间分配

功 能	时间要求 (ms)
余度数据采集	1.5
输入交叉对比	1
数据处理	4
比较监控	1
输出交叉对比	1.5
其他	1

公司总工程师将本项任务交付给项目主管李工程师实施，要求李工按双方合同要求完成本项目的策划、需求分析和方案设计工作，重点强调系统的可靠性要求，并特别提示：目前比较流行的嵌入式操作系统普遍采用可抢占式优先级调度策略，不能满足用户关于“能够精确地按事先预定的时间顺序完成数据的采集、处理和输出”的要求。

【问题 1】

李工接到任务后，认为本项目比较简单，很快就安排 3 名技术人员分别负责数据采集/输出模块、数据处理模块和比较监控模块的编写。总工程师听到汇报后，认为李工的方案和安排不妥，理由是李工忽视了系统的可靠性要求，对系统需求的理解不够深入。为实现系统关于可靠性方面的需求：

- ①你认为在组织结构、人员分工、设计开发等方面应做出哪些安排和规定？
- ②请写出关于余度表决算法的考虑。

【问题 2】

请用 200 字以内的文字说明李工应如何设计优先级调度策略，并按照如表 12-14 所示的要求实现任务流程处理。

【问题 3】

如图 12-22 所示为控制软件的数据流程关系，请根据本题给出的各个功能模块，填补其中的空缺处。

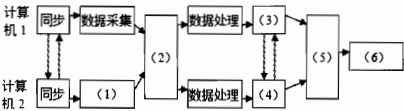


图 12-22 控制软件的数据流程

【问题 4】

请用 150 字以内的文字说明什么是系统失步，系统失步后应如何处理。

试题 13

某研究所承担了一项宇航嵌入式系统的研制任务，本项任务技术难度大、可靠性要求高、进度要求紧，预计软件规模约在一万行以上。而该所原先承担的相关项目均是用汇编语言编写（软件规模较小，一般是几百行）。项目主管工程师将软件开发的

论证工作分别布置给了王工和李工，要求他们根据本项目的技术要求，在一周内分别提出软件的实施方案。

① 该宇航嵌入式系统由数据处理机、信号处理机及数据采集模块组成。信号处理机主要通过数据采集模块将大量不同时间周期的外部数据采集进来，进行预处理（如 FFT 变换）后提交给数据处理机；数据处理机根据功能要求完成各种计算处理工作，并将结果输出。为了达到数据处理机和信号处理机的标准化，这两类处理机均采用同种处理机的体系结构（如 PowerPC7410）。整个系统工作的最大周期为 1s，最小周期为 5ms。数据处理机上任务分布如表 12-16 所示。

表 12-16 数据处理机上任务分布

工作周期（ms）	任务数（个）	共占时间（ms）
5	1	1
10	2	4
40	2	8
50	3	9
1000	2	3

② 一周后，王工和李工提交了各自的实施方案，其共同之处在于二者均提出采用操作系统及相应的软件开发环境。但是，王工和李工在选择什么样的操作系统及开发环境方面出现了较大分歧。王工的实施方案指出，为了满足本项目的技术要求，并有利于本所软件的长期发展，应选用具有硬实时处理能力的嵌入式实时操作系统及开发环境（如 VxWorks）；而李工的实施方案指出，操作系统主要完成对计算机资源的管理与分配工作，考虑到人们对操作系统及软件开发工具的熟悉程度，本项目应选用具有大众化的、开发人员容易上手的嵌入式操作系统（例如，Windows CE、Linux）。

③ 在两人争执不下的情况下，项目主管工程师组织了相关专家对王工和李工提交的实施方案进行了评审。最后，专家组一致认为王工的方案切实可行，操作系统的选型论证充分、客观，可以作为本所未来几年中相关项目的软件开发的基本环境。同时专家组还就具体实施选择嵌入式实时操作系统及开发环境产品时应着重考虑的相关问题提出了几点建议。

【问题 1】

为什么专家组一致认为王工的实施方案切实可行？请用 200 字以内文字简要说明。

【问题 2】

选择实时嵌入式操作系统及开发环境时应考虑哪几方面问题？请用 200 字以内文字简要论证。

【问题 3】

根据本项目中任务的时间性能参数，请用 200 字以内文字（也可用图示）描述任务的被调度关系，并分析系统是否可调度（暂不考虑操作系统的时间开销）。

试题 14

希赛公司承担了一项大型物料配送中心的研发项目，使用自动化设备进行货物的

存取。系统主要由两部分组成，其一是装有智能设备的配送车，用于自动化装卸货物；其二是用于管理和调度这些配送车的物料配送中心，如图 12-23 所示。

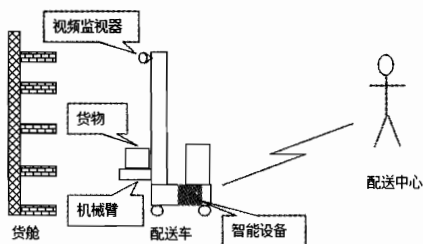


图 12-23 物料配送中心系统

智能设备接受配送中心发出的存取货指令，根据货舱地址和舱位位置等参数，自动控制配送车行走，并调整机械臂到达目标舱位，当存取完货物后，再次等待配送中心指令。其中，视频监视器用于采集工作现场的视频数据，该数据用于智能设备识别配送车当前位置。

公司经理将智能设备的研制任务交给了李工。李工投入了大量的时间，通过一个月的努力，提交了智能设备的设计方案。但是，出乎李工预料，该方案被专家评审会否定了。评审意见指出，李工的设计方案实施成本高，并且在技术上有缺陷，建议重新设计。

【问题 1】

李工的设计方案指出，考虑到智能设备工作的环境条件非常恶劣，并且处理机要处理大量的视频信息，所以，智能设备的硬件设计应考虑以下几点：

①由于智能设备要处理大量的图像数据，智能设备中的处理机应采用目前先进的、具有很强数据处理和信号处理能力的 MPC7410 处理器，并采用双处理机结构，以增加设备的容错能力。

②为了使智能设备适应不同环境条件，其设备的环境温度指标必须满足军用标准要求（-55~+70℃），因此，元器件必须经过严格筛选，并考虑芯片散热问题。

③由于智能设备与配送中心的数据交换采用无线通信，而配送中心又要控制多台配送车，为了保证通信数据的安全性，避免相互干扰，设计中应考虑采用安全措施，实施数据加解密操作，密钥由配送中心统一管理与分发。

请用 300 字以内的文字说明上述 3 点设计考虑有何问题。

【问题 2】

如图 12-23 所示是李工在设计方案中给出的智能设备工作状态转换图。

①请指出图中的两处错误（在图中圈出）并用 200 字以内的文字说明理由。

②给出正确的状态转换图。

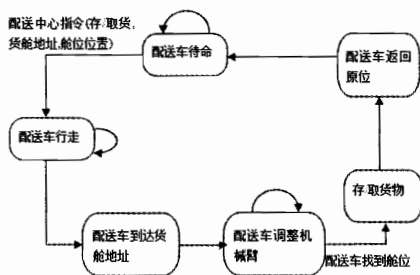


图 12-23 智能设备工作状态转换图

【问题 3】

针对李工的设计缺陷，请用 300 字以内的文字说明本项目应如何进行正确设计。

试题 15

某市政府部门有近 30 个委、办、局，分别承担着法定的政府职能，这些部门都有信息中心，负责本单位信息系统的建设和信息资源的采集、维护等工作，同时向业务部门提供信息化技术支持。该市的电子政务建设发展很快，统一规划实施了政务内网，实现了各个政府部门百兆以上带宽的互联互通，并在此基础上规划信息资源网。该政务内网的运行维护成本较高，所以在政务内网开通之后，市财政逐渐削减了各个委办局自身信息系统的维护费用，市政府也要求尽快实现信息资源的共享，以发挥电子政务网的效益。

【问题 1】

该市决定将各个委办局的信息系统采取物理集中的方式进行统一管理，一些部门对此提出了异议，主要理由是集中管理后，由于管理部门对业务应用不了解，难以保证对本单位业务进行有力支持。

请用 300 字以内文字，从技术角度论述该市实现信息系统统一管理是否可行。

【问题 2】

该市在信息系统集中管理建设过程中，需要在信息系统中重点规划建设哪些内容？

【问题 3】

发挥信息系统效益的关键是信息资源的有机共享，请给出该市政务信息资源共享的建议（200 字以内）。

试题 16

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。

【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, 填入图 12-24 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。

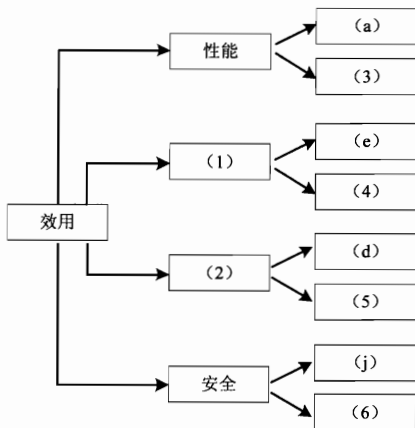


图 12-24 在线交易系统效用树

【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) 中各选出一个对系统架构风险、敏感点和权衡点最为恰当的描述。

12.4 练习题参考答案

试题 1 参考答案

【问题 1】

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。

主程序-子程序架构风格中，所有的计算构件作为子程序协作工作，并由一个主程序顺序地调用这些子程序，构件通过共享存储区交换数据。

管道-过滤器架构风格中，每个构件都有一组输入和输出，构件接受数据输入，经过内部处理，然后产生数据输出。这里的构件称为过滤器，构件之间的连接件称为数据流传输的管道。

【问题 2】

王工与李工的架构风格评价如表 12-17 所示。

表 12-17 王工与李工的架构风格评价

评价要素 \ 架构风格	共享数据的主程序-子程序	管道-过滤器
算法变更	—	(1) +
功能变更	(2) —	+
数据表示变更	(3) —	(4) —
性能	(5) +	(6) —

根据题干描述：“用户后续还有可能采用其他方式展示帮助内容”，因此 KWIC 系统对功能变更要求较高。

根据题干描述：“...，后续还将采用其他的方法实现这一功能”，因此 KWIC 系统对实现某一个功能的算法变更要求较高。

KWIC 是一个支持用户交互的窗口界面程序，因此对性能要求并不高。

KWIC 系统的显示帮助内容为文本，数据的表示基本不变，因此对数据表示变更要求不高。

综合上述分析，可以看出应该采用李工提出的管道-过滤器架构风格。

【问题 3】补充完整的架构示意图如图 12-25 所示。

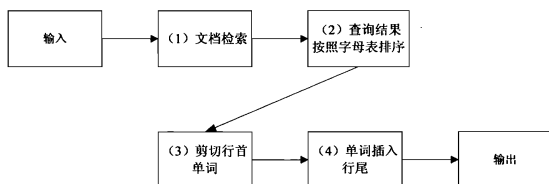


图 12-25 补充完整的架构示意图

试题 2 参考答案

【问题 1】

李工设计的流程符合 AUTOSAR 要求，理由如下：

李工设计的流程是将软件组件描述、系统描述和 ECU 资源描述同时定义，而王工设计的流程仅仅只做系统描述。

王工设计的流程没有考虑软件组件的描述，只是简单将软件组件作为第 4 步被集成。

李工设计的 ECU 软件开发流程的优势是明确了顶层定义阶段，并行度高，迭代清晰。

【问题 2】

- ① 操作系统的警报管理发现 ECU 系统出错时，启动错误处理程序。
- ② 错误处理程序将具体动作交由进程管理完成对发生错误的任务进行处理。
- ③ 错误处理程序产生一个错误事件。
- ④ 任务控制块处理程序调用资源管理功能，实现硬件资源重分配。
- ⑤ 任务控制块处理程序通知事件管理，对错误事件进行应答。

【问题 3】

采用 AUTOSAR 开放式架构的优点是：

- ① 可以有效支持多厂家汽车电子基础软件的研制。
- ② 有利于软件的重用，可根据不同的 ECU 结构，通过数据配置，自动生成各种 ECU 软件组件。
- ③ AUTOSAR 定义的软件框架，支持了汽车电子软件的全生存周期，包括构架、开发、测试、验证、授权、版本和接口。
- ④ AUTOSAR 规范覆盖整个汽车电子的三大领域：动力、底盘、车身。

试题 3 参考答案

【问题 1】

ESB 的主要功能包括：

- ① 应用程序的位置透明性。
- ② 传输协议转换。

- ③ 消息格式转换。
- ④ 消息路由。
- ⑤ 消息增强。
- ⑥ 安全支持。
- ⑦ 监控和管理。

采用 ESB 作为集成框架，能够实现灵活的部署结构，包括 CS 结构、P2P 结构等。

采用 ESB 作为集成框架，待集成系统只需要和总线进行联系，彼此之间不需要互相通信，这样就大大降低了系统的耦合程度。

采用 ESB 作为集成框架，在加入新的待集成系统时，只需要采用插件的方式实现传输协议和数据格式的适配即可，系统的可扩展性较强。

【问题 2】

对于需求（1）来说，由于需要共享系统的功能，并且系统的运行平台与语言差异较大，应该采用面向服务的方式进行功能集成，可以将工具的功能包装为服务，实现跨语言与跨平台访问。

对于需求（2）来说，工具所支持的通信协议和数据格式各不相同，并需要实现工具之间的灵活通信协议和数据格式交换，因此应该基于消息总线，以协议及数据适配器的方式实现灵活的通信协议和数据格式转换。

对于需求（3）来说，集成框架需要根据实际的软件系统开发流程，灵活、动态地定义系统设计与开发工具之间的协作关系，因此应该引入 workflow 定义语言及其引擎来动态描述工具之间的协作关系。

对于需求（4）来说，应该采用界面集成的方法对第三方工具进行集成，绕过工具内部的复杂处理逻辑。

【问题 3】

在实现工具之间数据格式的灵活转换时，通常采用适配器设计模式。即应首先定义一个统一的数据转换接口类，然后针对不同的数据格式转换需求定义对应的实际转换类，实际转换类需要继承数据转换接口类，并实现接口转换类定义的接口。

试题 4 参考答案

【问题 1】

- ① 将现有系统看做抽象的服务的提供者，集成方法统一明确。
- ② 不同层次的集成方法关注点不同，功能组合方面能力较弱。
- ③ 一般为中心辐射型（hub-and-spoke），系统之间的耦合度较高。
- ④ 集成系统具有模块化，松耦合的特点，可扩展性较好。

【问题 2】

针对该企业的集成实际情况，ESB 的基本核心功能包括：

- ① 提供位置透明性的路由和寻址服务。
- ② 控制服务寻址和命名的管理功能。
- ③ 至少一种形式的消息传递范型（如请求/响应、发布/订阅等）。
- ④ 支持至少一种可以广泛使用的传输协议和协议转换。
- ⑤ 支持服务提供的多种集成方式。如 Java 2 连接器、Web 服务、异步通信、适配器等。

【问题 3】

在采用 Web Service 作为 SOA 的实现技术时，该系统应该至少分为 6 个层次：底层传输层、服务通信协议层、服务描述层、服务层、业务流程层和服务注册层。

① 底层传输层主要负责消息的传输机制，HTTP、JMS 和 SMTP 都可以作为 Web Service 的消息传输协议，其中 HTTP 使用最广。

② 服务通信协议层的主要功能是描述并定义服务之间进行消息传递所需的技术标准，常用的标准是 SOAP 协议，还有新出现的 REST 协议。

③ 服务描述层主要以一种统一的方式描述服务的接口与消息交换方式，相关的标准是 WSDL。

④ 服务层的主要功能是将遗留系统进行包装，并通过发布的 WSDL 接口描述被定位和调用。

⑤ 业务流程层的主要功能是支持服务发现、服务调用和点到点的服务调用，并将业务流程从 Web Service 的底层调用抽象出来。相关的标准是 WS-BPEL（BPEL4WS）。

⑥ 服务注册层的主要功能是使得服务提供者能够通过 WSDL 发布服务定义，并支持服务请求者查找所需的服务信息。相关的标准是 UDDI。

试题 5 参考答案

【问题 1】

- ① （商品）搜索和识别，或商品浏览、搜索等。
- ② 选择和协商，或讨价还价、咨询等。
- ③ 购买。
- ④ 产品或服务递送，或物流等。
- ⑤ 售后服务（支持）。

【问题 2】

供应链管理主要包括：

- ① 需求计划，预测市场需求。
- ② 供应计划，合理地分配企业资源来满足需求。

③ 需求实现，快速、有效地满足需求。

电子商务环境下，供应链管理过程的整合、优化，可以使组织：

① 通过消除或者减少订购和递送过程中耗时、耗劳动力的步骤，增加收入和降低成本。

② 通过提供更多、更详细的有关发货日期和交易状态等信息增加客户的忠诚度，提升客户满意度。

③ 随着客户订单信息的传送速度和准确性的提升，减少公司在中间生产、存储和运输点的库存需求，使供应链内的库存降低。

【问题3】

① 要有一个方案编写（创建）计划。

② 制订方案编写的财务预算。

③ 确定方案编写团队。

④ 制订和实施有效的文档管理。

⑤ 进行技术解决方案开发，详细规划出实际的 IT 解决方案。

⑥ 确定项目的人力资源安排和时间进度。

⑦ 进行方案的质量保证审核。

⑧ 合理进行方案报价。

⑨ 清晰描述方案的条款（如支付、交货等）、条件和方案前提、假设。

⑩ 方案创建活动结束确认，并交付给客户。

试题6 参考答案

【问题1】

① Web 服务器处理用户请求，应用服务器处理业务逻辑与数据库访问，负载较为均衡。

② 业务逻辑与数据库访问都位于 Web 服务器中。业务与逻辑没有分离。

③ 采用多台应用服务器，系统的可靠性较高。

④ 需要将脚本语言与面向对象编程语言相结合，相对复杂。

【问题2】

O/R 映射指的是对象/关系映射，是一种编程技术，将关系数据库中的关系型数据与面向对象编程语言中类型系统定义的数据进行格式转换。采用对象/关系映射主要有以下3点好处：

① 可以将业务逻辑与数据逻辑分离。

② 可以使得开发人员采用面向对象的方式访问底层关系型数据库。

- ③ 能够做到上层应用与底层的具体的数据库无关，两者解耦合。

【问题 3】

影响 Web 应用系统性能的 3 个主要因素分别是：

① 数据库的连接与销毁。可以采用数据池的方式缓存数据库连接，实现数据库连接复用，提高系统的数据访问效率。

② 构件或中间件的加载与卸载。可以采用分布式对象池的方式缓存创建开销大的对象，实现对象复用，用以提高效率。

③ 线程的创建与销毁。可以采用线程池的方式缓存已经创建的线程，提高系统的反应速度。

试题 7 参考答案

【问题 1】

(2) Traveler 接受请求，将请求分解为对多个网站的数据请求调用，并为发起调用进行准备。

(3) 调用 A 网站提供的 Web 编程接口，给出（提供）用户位置信息，请求地图信息。

(5) 调用 B 网站的内容信息获取接口，给出（提供）用户地理信息，请求用户周边的住宿信息。

(7) 根据用户的请求信息进行数据内容聚合。

【问题 2】

目前常见的 Mashup 接口主要有以下 3 种：

① 聚合内容（Really Simple Syndication, RSS）式的 Mashup 接口。一种用于对网站内容进行描述和同步的格式，是目前使用最为广泛的 Web 资源发布方式，可以被称为资源共享模式的延伸。

② 表述性状态转移（Representational State Transfer, REST）式的 Mashup 接口。REST 从资源的角度来看待整个网络，分布在各处的资源由统一资源标识符（Unified Resource Identifier, URI）确定，而客户端的应用通过 URI 来获取资源的表示。

③ 基于简单对象访问协议（Simple Object Access Protocol, SOAP）的 Web 服务式 Mashup 接口。一种基于 XML 的数据格式定义，用来进行 Web 服务调用过程中的参数调用和返回。

【问题 3】

在客户端进行内容聚合的优点主要有以下两点：

① 从 Mashup 服务器存储的角度来说，对服务器所产生的负载较轻，因为数据可以直接从内容提供者那里传送到客户端。

②从网络传输的角度来说,在基于 Ajax 等技术和应用模型的基础上,客户端页面只请求需要更新的内容,而不用刷新整个页面,从而减少网络数据的通信量。

试题8 参考答案

【问题1】

① 张工提出的集中式数据架构是由一个处理器、与它相关联的数据存储设备,以及其他外围设备组成,它被物理地定义到单个位置。系统提供数据处理能力,用户可以在同样的站点上操作,也可以在地理位置隔开的其他站点上通过远程终端来操作。系统及其数据管理被某个或中心站点集中控制。

② 刘工提出的分布式数据架构使用多个计算机系统上的多个局部数据库系统构成,数据可以在多个不同的局部数据库中进行传送,由不同的数据库管理系统软件进行管理,运行在多种不同的计算机上,支持多种不同的操作系统。这些机器位于(或分布在)不同的地理位置并通过多种通信网络连接在一起。企业数据可以分布在不同的计算机上,一个应用程序可以操作位于不同地理位置的机器上的数据。

【问题2】

读写分离架构利用了数据库的复制技术,将数据的读和写分布在不同的处理节点上,从而达到提高可用性和扩展性的目的。

CRSS 的分布式数据库系统需要由多个局部数据库系统、多个热备份数据库系统和多个数据缓存组成。局部数据库负责数据的写入,多个热备份数据库系统用以解决单点故障的问题,数据缓存负责为应用提供所读取的数据。

① 读取数据:应用访问缓存,如果命中则返回,否则从局部数据库系统中读取数据并将数据加载到缓存后返回。

② 添加数据:采用延迟加载策略,应用将数据直接写入局部数据库。

③ 更改数据:应用更改局部数据库中的数据,将缓存中的数据标记为失效。

④ 删除数据:应用删除局部数据库中的数据,将缓存中的数据标记为失效。

【问题3】

张工提出的集中式数据架构通过向上扩展(Scale Up)提升系统的可扩展性。具体的实现方式包括硬件扩容(增加 CPU 数量、内存容量、磁盘数量)和硬件升级(更换为高端主机或高速磁盘等)。

刘工提出的分布式数据架构通过向外扩展(Scale Out)提升系统的可扩展性。具体的实现方式包括数据复制、数据垂直切分(或/和)水平切分、缓存和全文搜索。

试题9 参考答案

【问题1】

采用双机冗余热备方式,当本地某个系统发生故障时,系统能够自动快速地切换到正常的系统,通过本地故障恢复确保系统持续提供服务。目前有许多不同的厂家提供双机冗余热备的产品,采用双机冗余热备方式,涉及以下众多因素:

① 双机热备产品适用的规模。

- ② 支持的操作系统。
- ③ 支持的数据库系统。
- ④ 对正常业务系统的性能影响。
- ⑤ 提供的 GUI 管理工具功能易用性。
- ⑥ 能够完全实现多应用多级切换（应用级切换），适用多种应用并存的系统，某一应用的切换可以不对其他应用产生影响。
- ⑦ 集中管理配置能力。
- ⑧ 远程监控和管理能力。
- ⑨ 切换速度。
- ⑩ 磁盘管理方面功能。

【问题 2】

备份中心的主要作用是在灾难发生时能够有效地保护数据。在租用商业备份中心时应该注意以下几点：

- ① 备份中心与公司的系统有充分远的距离（几十甚至几千千米）。
- ② 备份中心具有抗灾能力。
- ③ 备份中心要有足够的带宽确保与数据中心的数据同步。
- ④ 合理选择备份的时间段。

【问题 3】

人们总是希望信息系统能够在日常的运行中具有高性能，在灾难发生时能够不损失任何数据，但事实上却很难实现这一目标。

如果系统向异地备份中心的备份时间段越小，就会更加及时地把数据备份到备份中心，能够在灾难发生时损失的数据越少，但是会对系统日常运行性能产生越大的影响。极端情况下，如果每次有了新的交易就向异地备份中心备份数据，当灾难发生时，只会损失最新发生的交易。但是备份时间越小，就要求系统的备份链路有很高的带宽，需要付出的代价也越大。另一方面，这也会影响数据中心的性能。

因此，在实际中需要根据具体的业务情况备份时间段、业务系统的运行性能，以及灾难发生时数据损失这三者的关系，做出较为合理的决定。当然，要达到最优化通常非常困难。

试题 10 参考答案

【问题 1】

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】

动态冗余又称主动冗余，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。前者叫热备份系统（双重系统），后者叫冷备份系统（双工系统、双份系统）。

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。

M2 采用动态冗余后的可靠度为：

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

李工给出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。

$$R = 0.99 * 0.999999 * 0.999999 * 0.99 = 0.98$$

【问题 3】

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。

试题 11 参考答案

【问题 1】

填写结果如图 12-25 所示。

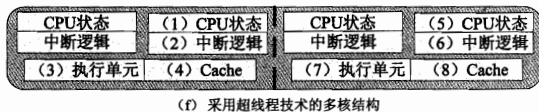


图 12-25 填写后的结果

(1) 单核结构：计算机中仅有一个物理处理器，不支持应用软件的并行执行，因为在任何时间点上，CPU 都只能执行一个指令流。

(2) 多处理器结构：支持真正意义上的并行执行，因为多个线程或进程能够在多

个处理器上同时执行。

(3) 超线程结构 (SMT)：实际上计算机中只有一个实际的物理处理器，但从软件角度来看，存在多个逻辑处理器，支持操作系统和应用程序将多个线程调度到多个逻辑处理器上，就像多处理器系统一样。从微体系结构的角度看，逻辑处理器的指令是固定的，并且在共享的执行资源上同时执行。

(4) 多核结构：采用单芯片多处理器 (CMP) 的设计，此种结构不是重用单个处理器中某些处理器资源，而是在单个处理器芯片内实现两个或更多的“执行核”。这些执行核都是相互独立的处理器，并具有自己的执行集合及体系结构资源。

(5) 共享 Cache 的多核结构：与多核结构工作方式相一致，主要差别在于设计时将这此执行核设计成可共享片上的 Cache。

(6) 采用超线程技术的多核结构：主要是将多核结构与超线程结构相结合，从而将逻辑处理器的数量增加到执行核的两倍。

【问题 2】

(1) 同步：是指两个或多个线程协调其行为的过程。如一个线程停下来等待另一个线程完成某项任务。

(2) 通信：是指与线程之间交换数据相关的带宽和时延问题。

(3) 负载均衡：是指多个线程之间工作量分布的情况。负载均衡能够使各线程的工作量均衡分配。

(4) 可伸缩性：是衡量在性能更加强劲的系统上运行软件时能否有效利用更多线程的指标。例如，如果一个应用程序是面向 4 核系统编写的，那么当程序在 8 核系统上运行时，其性能是否能够线性增长（即增加一倍）。

常用并发程序设计模式的分解方式如表 12-18 所示。

表 12-18 常用并行程序设计模式的分解方式

模式	分解方式
任务级并行模式	(1) 任务分解
分治模式 (Divide and Conquer)	(2) 任务分解或数据分解
几何分解模式	(3) 数据分解
流水线模式	(4) 数据流分解
波峰 (Wavefront) 模式	(5) 数据流分解

【问题 3】

(1) 嵌入式操作系统必须是强实时系统，实时性应达到毫秒级。

(2) 操作系统应支持存储器保护能力，支持进程间的资源独立性（或支持时间、空间的隔离）。

(3) 操作系统内核应支持多核结构的处理器，并支持多个逻辑处理的方式。

(4) 支持高速 Cache 的管理。

试题 12 参考答案

【问题 1】

(1) 成立两个独立的软件开发小组。

两个小组合作编写统一的系统需求稳当和软件功能说明书。

两个小组分别进行软件开发, 最好使用不同的程序设计语言(如 C 语言、C++语言和 ADA 语言)。

(2) 余度表决算法一般采用主从式。

以下假设设计计算机 1 为主机, 计算机 2 为从机。

对输入数据进行交叉对比, 如果一致, 则用计算机 1 的数据为数据处理; 如果不一致, 有历史数据时, 计算机 1 用历史数据为数据处理; 无历史数据时, 若计算机 1 和计算机 2 的数据中有一个在门限值之内, 则使用该数据作为数据处理, 若无门限值之内的数据, 则计算机 1 使用默认数据进行处理。交叉对比不一致, 应建立故障记录, 若故障记录超过门限值, 系统变为单余度工作方式。

对输出数据进行交叉对比, 如果一致, 用计算机 1 的数据为输出数据; 如果不一致, 且计算机 1 和计算机 2 均无历史数据, 则计算机 1 用默认数据作为输出数据。交叉对比不一致, 应建立故障记录, 若故障记录超过门限值, 系统变为单余度工作方式。

【问题 2】

设计时, 将本项目的所有任务设定为统一的优先级, 避免优先级抢占; 根据 10ms 工作周期内的时间节点划分, 设定操作系统的时钟节拍为 0.5ms, 设计出一个软件定时器; 根据软件定时器的计时情况, 在事先安排好的时间节点上, 分别启动相应的任务工作; 如果在规定的时间内任务工作未完成, 则按超时处理。原则上使用上一周期的数据。

【问题 3】

- | | | |
|----------|----------|--------|
| (1) 数据采集 | (2) 交叉对比 | (3) 同步 |
| (4) 同步 | (5) 交叉对比 | (6) 输出 |

【问题 4】

计算机 1 和计算机 2 进行同步操作时, 相互间未能同步上, 则称为失步。

在一个工作周期开始时, 两个计算机在时间上必须同步, 一旦发生失步, 为了确保系统的稳定, 本周期的工作将采用上一周期的数据进行; 如果输出失步, 则输出交叉对比功能无效, 使用上一周期的输出数据作为本周期的输出。

试题 13 参考答案**【问题 1】**

① 宇航嵌入式系统是一种硬实时系统, 要求对外部事件做出快速响应(一般在毫秒级), 因此, 硬实时操作系统在实时处理和相应的方面优于一般嵌入式操作系统, 故王工提出的选用具有硬实时处理能力的嵌入式操作系统方案可行。

② 从资源的约束性来讲, 宇航嵌入式系统由于受空间和重量的限制, 操作系统

代码不宜过于庞大，选择具有可裁剪功能的操作系统更适应本项目。

③ 宇航嵌入式系统要求软件可靠性要高，而王工选择的操作系统已经过相关部门验证，其可靠性高于李工所选的操作系统。

【问题 2】

选择嵌入式实时操作系统及开发环境应考虑以下几方面：硬实时性、可扩展性、接口的开放性、可靠性、可裁剪性、环境的可用性、兼容性。

【问题 3】

因为 $\sum C_i/T_i = 1/5 + 4/10 + 8/40 + 9/50 + 3/1000 = 983/1000 < 1$ ，因此系统是可调度的。调度次序为 $A \rightarrow E \rightarrow B \rightarrow D \rightarrow C \rightarrow B \rightarrow C \rightarrow D \rightarrow D \rightarrow E$ ，其中，A 代表 5ms 任务，B 代表 10ms 任务，C 代表 40ms 任务，D 代表 50ms 任务，E 代表 1000ms 任务（调度次序可改变，不唯一）。

试题 14 参考答案

【问题 1】

1. 关于处理机选型问题

① 智能设备工作比较单一，虽然要处理图像信息，使用目前较廉价的信号处理芯片就可以承担，采用最新处理机芯片，虽然处理速度快，但成本也相对高。

② 最新处理机芯片，往往使芯片的集成规模增大，带来了设计的复杂性，同时使项目的风险难于预测。

③ 配送车的主要功能就是搬运物料，其可靠性相比其他系统要低得多，使用双机容错备份技术只能会加大成本，不会带来任何意义。

2. 关于环境条件问题

① 物料配送中心的环境通常和人们的生活环境相当， $-55 \sim +70^{\circ}\text{C}$ 是军用设备的环境标准要求，因此，智能设备满足工业环境标准即可。

② 按军用设备的环境标准设计智能设备，当然设备的环境适用性更强，但必须采用军品档芯片，其价格通常为工业档芯片的几倍。

③ 环境要求越高，元器件筛选成本越大。

3. 关于数据通信的安全性问题

智能设备与配送中心的通信采用普通数据交换方式即可，以数据的加/解密方式确保安全性完全没有必要。

【问题 2】

(1) 两处不正确的转换如图 12-27 所示。

① 处不正确：配送车到达货舱地址是动作，不是状态。

② 处不正确：配送车在存/取货物状态完成后，只能是待命状态，配送车返回原

位状态不存在。

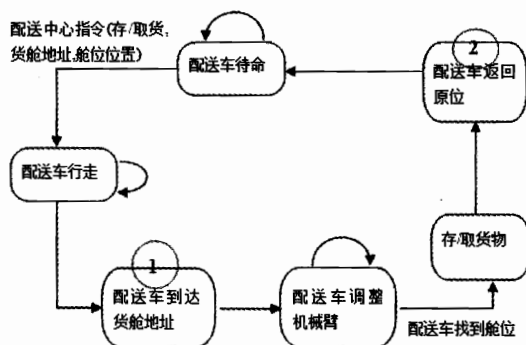


图 12-27 不正确的状态

(2) 正确的状态转换图如图 12-28 所示。

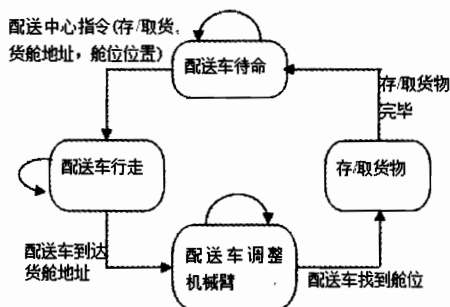


图 12-28 正确的状态图

【问题 3】

(1) 在设计方案初期, 应充分考察大型物料配送中心的现场环境, 根据配送中心所处地理的位置, 确定智能设备的环境温度要求, 可以遵循工业标准 ($-40\sim+60^{\circ}\text{C}$) 或民用标准 ($-20\sim+50^{\circ}\text{C}$) ;

(2) 通过分析智能设备的计算量, 合理评估处理机处理能力需求。

处理机芯片选型原则:

- ① 较通用的 DSP 处理机芯片。
- ② 已被广泛应用, 技术成熟、且价格适中。
- ③ 处理能力满足智能设备计算量需求。
- ④ 关键芯片可选用工业档, 其他芯片可选用民用档。

(3) 根据智能设备的功能要求, 进行软件需求分析。主要包括:

- ① 完成功能定义。
- ② 制订正确状态转换图, 并多于用户沟通, 得到用户认可。

- ③ 定义智能设备与配送中心数据交换格式。
- ④ 明确智能设备与配送中心的控制命令及参数。
- (4) 编制设计方案, 并请同行专家及用户评审。

试题 15 参考答案

【问题 1】

该市实行各个委办局信息系统统一管理可行。

该市电子政务网性能优良, 信息系统物理部署点位置不会影响应用, 也不存在信息传输瓶颈。

集中管理后, 不仅能共享网络、主机的硬件设备, 还能集中进行高可靠性配置、安全管理, 以较低的成本向各个应用系统提供较高的可用性和安全性。

集中管理后, 各个部门的运行维护和管理费用将大大降低, 可节省该市用于政府部门信息系统维护的整体费用。

应用系统位于信息系统层次结构的上层, 底层的集中共享, 能够对业务提供更好的支撑。而各个部门的应用对信息系统的要求可由本部门提出, 纳入到集中管理部门对信息系统的统一规划中。

【问题 2】

- (1) 灾备系统(容灾系统)。
- (2) CA 认证系统(身份识别系统)。
- (3) 入侵检测系统。
- (4) 安全审计系统。
- (5) 防火、防盗等物理安全措施。
- (6) 高可用性设施(如多机集群、网络冗余、电源冗余等)。
- (7) 较好性能的网络管理系统, 监控网络流量。

【问题 3】

- (1) 制定全市政务信息的分类、分级标准。
- (2) 结合各部门的职责, 制定信息的采集原则, 如关键业务产生关键信息, 权威部门产生权威信息, 从源头上保证信息的质量。
- (3) 制定信息的维护、更新原则, 如谁生产谁维护。
- (4) 建设信息的交换平台, 规划信息交换目录, 制定信息的交换办法。
- (5) 建设政务信息仓库, 开发信息资源的综合应用。
- (6) 优化政务流程。

试题 16 参考答案

【问题 1】

表 12-19 效用树答案表

编号	答案
(1)	可修改性
(2)	可用性
(3)	(f)
(4)	(l)
(5)	(h)
(6)	(b)

【问题 2】

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。

题干描述中，(k) 描述的是系统架构风险；(c) 描述的是敏感点；(g) 描述的是权衡点。

13

第 13 章

架构设计论文

根据考试大纲，信息系统项目管理论文考试的出题范围定位于以下 6 个方面：

- 系统建模：包括定义问题与归结模型；结构化系统建模；面向对象系统建模；数据库建模。
- 软件架构设计：包括软件架构设计；特定领域软件架构；基于架构的软件开发方法；软件演化。
- 系统设计：包括处理流程设计；系统人机界面设计；文件设计；存储设计；数据库设计；网络应用系统的设计；系统运行环境的集成与设计；系统性能设计；中间件；应用服务器。
- 分布式系统设计：包括分布式通信协议的设计；基于对象的分布式系统设计；基于 Web 的分布式系统设计；基于消息和协同的分布式系统设计；异构分布式系统的互操作性设计。
- 系统的可靠性分析与设计：包括系统的故障模型和可靠性模型；提高系统可靠性的措施；系统的故障对策和系统的备份与恢复。
- 系统的安全性和保密性设计：包括系统的访问控制技术；数据的完整性；数据与文件的加密；通信的安全性；系统的安全性设计。

13.1 历年考试情况分析

在历年的考试试题中，系统架构设计师论文试题一共 4 道，由考生选择其中一道试题作答，满分为 75 分。历年考试的试题如表 13-1 所示。

表 13-1 系统架构设计师论文试题分布表

考试时间	考试题目	知识点说明
2009.11	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
2010.11	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
2011.11	论模型驱动架构在系统开发中的应用	模型驱动架构 (Model Driven Architecture, 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	需求获取技术及每种技术的基本方法

13.2 典型试题分析

本节对一些典型的试题进行分析和解答, 以便考生更好地掌握相关知识点。

13.2.1 论软件体系结构风格及其应用

软件体系结构设计的一个核心问题是如何有效地使用重复的体系结构模式, 即达到软件体系结构级的软件重用。软件体系结构风格 (Software Architecture Style) 是描述软件系统组织方式的常用模式, 在实践中已经被多次应用。按照 Shaw 和 Garlan 的说法, “一种体系结构风格定义了构件类型和连接件类型的词汇表, 以及它们如何组合的约束条件”。体系结构风格通常分为数据流 (Data Flow) 风格、调用/返回 (Call/Return) 风格、独立构件 (Independent Components) 风格、虚拟机 (Virtual Machines) 风格和仓库 (Repositories) 风格五大类。在实际应用中, 随着软件系统规模的扩大和复杂, 一个系统往往会同时使用多类体系结构风格, 这些风格可以交叉组合、彼此重叠。

请围绕“软件体系结构风格及其应用”论题, 依次从以下 3 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及你在其中所担任的主要工作。
- ② 请说明以上软件体系结构风格分类中每一类有哪些经典的软件体系结构设计风格, 并就其中至少三类论述其具体的软件体系结构风格的构件、连接件类型和组合约束要求等结构特征及其应用特点。
- ③ 具体阐述你参与管理和开发的项目中在体系结构设计时选择使用软件体系结构风格的情况, 包括选择的依据、多个风格组合使用的情况等, 以及最终实际效果。

写作要点

关于常见的软件体系结构风格，请参看“8.1.4 软件架构的风格”。

本题考查的重点是软件体系结构的分类，每类风格的构件、连接件和组合约束情况，以及在项目中具体如何根据实际情况选择某一种或几种软件体系结构风格。具体来说，关于“软件体系结构风格及其应用”的论文，大致包括以下几个方面：

① 简要叙述你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。

② 说明软件体系结构风格分类中每一类有哪些经典的软件体系结构设计风格，并就其中至少三类论述其具体的软件体系结构风格的构件、连接件类型和组合约束要求等结构特征及其应用特点。

③ 结合项目的实际状况，指出在体系结构设计时选择使用软件体系结构风格的情况，包括选择的依据、多个风格组合使用的情况等，要给出实际的效果及分析。

13.2.2 论开放系统应用的互操作性技术

如今建设的计算机应用信息系统往往已是基于网络的开放式信息处理平台。其中，应用的互操作性技术已成为迫切需要解决的一个关键技术。根据你对当前开放系统环境的认识和工程系统的开发的经验，论述下列3个问题：

① 简述应用互操作性与应用可移植性这两个相关而又不相同的概念。具体说明你所采用的开放式网络体系结构和遇到的应用互操作性问题。

② 具体叙述你为了保证开放系统的应用互操作性，采用过哪些主要技术和措施，它们的效果如何？

③ 简述开放式网络体系结构的一般设计原则，以及应用互操作性技术在其中的地位与发展方向。

写作要点

本题主要考查考生对互操作性与应用可移植性这两个概念的理解与应用。

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。

可移植性使软件系统适用于多种硬件平台、用户界面、操作系统、编程语言或编译器。为了实现可移植性，需要按照硬件无关的方式组织软件系统，将其他软件系统和环境提取出来。可移植性是系统能够在不同计算环境下运行的能力。这些环境可能是硬件、软件，也可能是两者的结合。在关于某个特定计算环境的所有假设都集中在一个构件中时，系统是可移植的。如果移植到新的系统需要做些更改，则可移植性就是一种特殊的可修改性。

本文写作要点，大致包括以下几个方面：

① 简述应用互操作性与应用可移植性这两个概念的异同。

② 具体说明你所采用的开放式网络体系结构和遇到的应用互操作性问题（如数据库的互操作问题）。为了保证开放系统的应用互操作性，采用过哪些主要技术和措施，它们的效果如何。

③ 简述开放式网络体系结构的一般设计原则，以及应用互操作性技术在其中的地位与发展方向。

13.2.3 论多层分布式结构系统的开发

传统的应用系统模式是“主机/终端”或“客户机/服务器”。随着中间件技术和Web技术的发展，这些传统模式已经不能适应新的环境。目前设计大型系统大多采用多层分布式结构，如C/A/S和B/A/S，应根据系统具体需求和运行环境的不同选择合适的结构。

请围绕“多层分布式机构系统的开发”论题，依次从以下3个方面进行论述：

① 简要叙述你参与分析设计的多层分布式结构系统及你担任的主要工作。

② 简要说明多层分布结构分类的依据及多层分布式结构的特点，并指出你参与分析设计的系统属于其中的哪种结构，以及选择这种结构的原因。

③ 具体论述你在开发该系统时采用了哪些方法、策略与工具来实现所选的结构。

写作要点

① 简要介绍你所参与分析设计的多层分布式结构系统（说明该系统采用的是C/A/S还是B/A/S），以及你在系统中担任的主要工作。

② 说明多层分布式结构分类的依据。

从概念上说，任何信息系统应用都可以映射到以下5层。

- 表现层：是实际的用户界面。
- 表现逻辑层：为了生成表现层而必须进行的处理。
- 应用逻辑层：包括支持实际业务应用和规则所需的所有逻辑和处理。
- 数据处理层：包括用来存储和访问往来于数据库的数据所需的所有命令和逻辑。
- 数据层：数据库中实际存储的数据。

C/A/S结构是指将数据层和数据处理层放置在各自的服务器上，应用逻辑层放置在各自的服务器上，表现逻辑层和表现层放置在客户端上，这也称为三层或多层客户/服务器计算。

在B/A/S结构中，表现层和表现逻辑层在客户端Web浏览器中使用从某个Web服务器下载的内容实现，表现逻辑层连接到运行在应用服务器上的应用逻辑层，应用逻辑层最终连接到后台的数据库服务器。

③ 多层分布式结构系统的主要特点。

- 安全性：中间层隔离了客户对数据服务器的访问，保护了数据库的安全。
- 稳定性。

- 易维护性：由于业务逻辑位于单独的一层，当业务规则发生变化后，客户端程序基本不作改动。
 - 快速响应：通过负载均衡及中间层缓存数据能力，可以提高对客户端的响应速度。
 - 系统扩展灵活：基于多层分布式结构，当业务增大时，可以在中间部署更多的应用服务器，提高对客户端的响应，而所有变化对客户端透明。
- ④ 多层分布式结构系统的开发主要考虑 4 方面的技术。
- 开发环境：开发人员需要一种创建新组件，并将已有组件加以集成的开发环境。
 - 应用程序的集成。
 - 应用程序的配置：分布式结构的开发需要配置平台的支持，以便在用户剧增时能有效地扩展，并保持系统的稳定。
 - 选择适合的中间件。

13.2.4 论设计模式在软件开发中的应用

设计模式描述了在特定场景下解决一般设计问题的类和相互通信的对象。一个设计模式命名、抽象并确定了一个通用设计结构的主要方面，这些设计结构能被用来构造可复用的面向对象设计。现在，设计模式已经广泛地应用在软件开发中。

请围绕“设计模式在软件开发中的应用”论题，依次对以下 3 个方面进行论述：

- ① 概要叙述你参与分析和开发的应用项目，以及你所担任的主要工作。
- ② 简要介绍设计模式的基本概念及分类，详细说明在你所参与分析和开发的应用项目中应用了哪些设计模式、方法，以及选用它们的原因。
- ③ 分析并讨论使用设计模式的效果，并分析和评价设计模式对软件开发的影响。

写作要点

所谓设计模式，简单地理解，是一些设计面向对象的软件开发的经验总结。设计模式事实上是系统地命名、解释和评价某一个重要的可重现的面向对象的设计方案。关于设计模式的详细解释，请参看第 8 章。

本文的写作要点，大致包括以下几个方面：

- ① 结合自己参与分析和开发的应用项目或从事的实际工作，说明设计模式对系统开发的作用。
- ② 较为详细地说明设计模式的基本分类，并结合项目实际工作，说明在系统开发中所采用的设计模式，以及选择这些设计模式的原因。
- ③ 结合项目实际工作，举例说明设计模式的使用效果，针对设计模式对软件开发的影响展开分析与讨论。

13.2.5 论基于场景的软件体系结构评估方法

大型复杂软件系统开发所关注的问题之一是质量，在软件系统的早期设计阶段，选择合适的体系结构对系统许多关键质量属性（如可用性、可修改性、性能、安全性、易用性等）起着决定性的影响。不恰当的软件体系结构将给项目开发带来灾难。因此，尽早分析和评估一个系统的体系结构非常重要。软件体系结构分析和评估的目的是为了识别体系结构中潜在的风险，验证系统的质量需求在设计中是否得到体现，预测系统的质量并帮助开发人员进行设计决策。

软件体系结构的评估通常是指评估参与者在评估过程中利用特定评估方法对系统质量属性进行分析与评估。基于调查问卷或检查表的评估和基于场景（Scenarios）的评估是目前主要的两类评估方式。利用场景评估技术进行软件体系结构评估的主流方法包括 SAAM（Scenario-based Architecture Analysis Method）、ATAM（Architecture Tradeoff Analysis Method）和 CBAM（Cost Benefit Analysis Method）。SAAM 方法最初用于比较不同的体系结构，后来用于指导对体系结构的检查，使其主要关注潜在的问题，如需求冲突，或仅从某一参与者观点出发的不全面的系统设计。ATAM 方法在揭示出结构满足特定质量目标的同时，也能反映出质量目标之间的联系，从而权衡多个质量目标。CBAM 方法可以看做是 ATAM 方法的补充，在其评估结果上对软件体系结构的经济性进行评估。

请围绕“基于场景的软件体系结构评估方法”论题，依次从以下3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，包括角色、工作内容等。

② 请从评估目的、评估参与者、评估活动或过程、评估结果等几个方面对 SAAM 或 ATAM 评估方法进行分析。

③ 结合你参与的实际工作和项目的实际情况，具体阐述你在进行体系结构设计和评估时，采用了什么评估方法，如何具体实施，最终实际效果如何。

写作要点

利用场景评估技术进行软件体系结构评估的主流方法包括 SAAM、ATAM 和 CBAM，对此的详细描述请参看“8.1.8 软件架构评估”。

本文的写作要点，大致包括以下几个方面：

① 简要叙述你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，特别是要介绍在软件体系结构设计方面的情况。

② 从评估目的、评估参与者、评估活动或过程、评估结果等几个方面对 SAAM 或 ATAM 评估方法进行分析（只要写一种评估方法就可以了）。

③ 结合你参与的实际工作和项目的实际情况，说明对软件体系结构进行评估的原因，并具体阐述你在进行体系结构设计和评估时采用的评估方法和具体实施的过程和步骤，对最终实际效果进行说明。

13.2.6 论基于 DSSA 的软件架构设计与应用

软件架构设计的一个重要课题是如何解决软件重用问题。特定领域软件架构 (Domain Specific Software Architecture, DSSA) 是一种有效实现特定领域软件重用的手段。按照 Tracz 的说法, DSSA 就是一个特定的问题领域中由领域模型、参考需求、参考架构等组成的开发基础架构, 其目标就是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。领域分析的主要目的是获得领域模型, 领域模型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 DSSA, DSSA 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依据领域模型和 DSSA 开发和组织可重用信息。

请围绕“基于 DSSA 的软件架构设计与应用”论题, 依次从以下 3 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 就你所熟悉的领域, 请给出针对该特定领域, 在基于 DSSA 的软件设计开发中所涉及的领域模型、参考需求和参考架构以及相应的支持环境或设施。
- ③ 具体阐述你参与管理和开发的项目中使用 DSSA 的情况, 包括领域分析、领域设计和领域实现等活动是如何具体实施的, 最终实际效果如何。

写作要点

这是一道关于 DSSA 的试题。DSSA 的概念以及基本活动, 题目中都进行了简要的说明。值得注意的是在论文的写作选材方面, 作者的项目需要具备一定的行业特色。如电力行业、通信行业、铁路系统等。关于 DSSA 所具备的特征, 以及如何用 DSSA 的体系来具体创建系统的软件架构。关于 DSSA 的详细描述请参看“8.1.7 特定领域软件架构”。

本文写作要点, 大致包括以下几个方面:

- ① 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。
- ② 应结合自己所熟悉的领域, 定义领域范围, 确定领域应用需要满足的用户需求; 定义领域特定的元素、领域字典和领域术语; 定义领域特定的设计和实现需求约束; 在此基础上, 定义领域模型, 产生该领域的参考架构, 并说明构件的语法和语义; 最后, 产生、搜集可重用的产品单元, 为 DSSA 增加构件, 为问题域实现新应用提供支持。这个 DSSA 的建立过程是并发、递归和反复进行的。

所给出的 DSSA 应该具备以下 4 个方面的特征:

- 一个严格定义的问题域和/或解决域。
- 具有普遍性, 使其可以用于领域中某个特定应用的开发。
- 对整个领域能有合适程度的抽象。
- 具备该领域固定的、典型的在开发过程中的可重用元素。

- ③ 需要结合项目实际, 指出在架构设计时使用 DSSA 的情况, 包括领域分析、领域设计和领域实现等活动是如何具体实施的, 要给出实际的效果并进行分析。

13.2.7 论信息系统建模方法

系统模型在软件开发中扮演着重要的角色。可为已有的系统创建模型，以便更好地理解这些系统；也可以针对待开发的系统创建模型，作为记录业务需求或技术设计的方法。模型是建立信息系统的基础。恰当地运用信息系统建模方法，是成功地进行软件开发的一个关键环节。

请围绕“论信息系统建模方法”论题，依次从以下3个方面进行论述。

- ① 概要叙述你参与管理和开发的信息系统项目，以及你在其中所承担的主要工作。
- ② 论述常见的信息系统建模方法的主要内容（包括每种建模方法的核心思想及所创建的模型）。
- ③ 具体阐述你参与管理和开发的项目中选择使用的信息系统建模方法及选择该方法的原因，给出具体的实施过程和实施效果。

写作要点

这是一道关于信息系统的建模的试题。值得注意的是，有不少考生把该试题的建模方法理解成了开发模型与架构，这都是不正确的。信息系统的建模方法可以分为：面向过程的建模、面向数据的建模、面向信息的建模、面向决策的建模和面向对象的建模5种。

面向过程的建模方法是把过程看做系统模型的基本部分，数据是随着过程而产生的。最有影响的面向过程的设计方法是 Yourdon 设计法。

面向数据的建模方法把模型的输入输出看成是最为重要的，因此，首先定义的是数据结构，而过程模块是从数据结构中导出的，即功能跟随数据。最有影响的面向数据的设计方法是 Jackson 设计法。

面向信息的建模方法与面向数据建模方法的区别就是信息和数据的区别。信息和数据都是信息系统中最基本的术语，数据是指记载下来的事实，是客观实体属性的值，而信息是构成一定含义的一组数据。面向信息建模方法是从整个系统的逻辑数据模型开始的，通过一个全局信息需求视图来说明系统中所有基本数据实体及其相互关系，然后，在此基础上逐步构造整个模型，信息模型记录系统运作所需的信息实体，如人员、地点、事物、观念等，为分析现行系统提供信息的图形化表示。数据建模的目的是设计和实现满足系统信息需求的数据库结构，即数据建模支持系统设计。

决策支持系统由数据库、模型库和各自的管理系统组成。决策支持系统模型需要反映的问题是系统的决策制订原则和机理、系统的组织机构和人员配置。通过对决策系统的建模，企业的领导可以对企业有一个细致的了解，从而发现其中问题。如组织结构臃肿、职权划分不清、权力范围不合理等，据此进行相应的改革。比较成熟的决策支持系统建模方法有 Petri 网和 GRAI 法。

面向对象的分析方法是利用面向对象的信息建模概念，如实体、关系、属性等，同时运用封装、继承、多态等机制来构造模拟现实系统的方法。传统的结构化设计方法的基本点是面向过程，系统被分解成若干个过程。而面向对象的方法是采用构造模型的观点，在系统的开发过程中，各个步骤的共同的目的是建造一个问题域的模型。

在面向对象的设计中，初始元素是对象，然后将具有共同特征的对象归纳成类，组织类之间的等级关系，构造类库。在应用时，在类库中选择相应的类。

在完成该论文时，首先应对上面的方法进行介绍，然后说明自己的系统中采用了哪种方法，具体是怎么样来使用这种方法的。

13.2.8 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3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。

写作要点

软件容错的主要目的是提供足够的冗余信息和算法程序，使系统在实际运行时能够及时发现程序设计错误，采取补救措施，以提高软件可靠性，保证整个计算机系统的正常运行。

软件容错技术主要有恢复块方法和N版本程序设计，另外还有防卫式程序设计等。对于该知识点的详细描述请参看“11.1.3 容错技术”。

在进行本文的写作时，要注意以下几个方面：

- ① 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4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。
- ③ 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）。
 - 给出本软件的可靠性指标。
 - 分析出软件容错的实时性指标，如恢复时间、重组（或称重构）时间。

13.2.9 论数据仓库的设计与实现

数据仓库已经成为数据分析和联机分析处理中日趋重要的平台。然而，数据仓库的设计与实现过程面临许多技术上的挑战，如多个异种数据源的集成带来的困难等。

请围绕“论数据仓库的设计与实现”这一论题，依次从以下3个方面进行论述：

- ① 概要地叙述你参与分析和开发的软件项目，以及你所担任的主要工作。
- ② 具体讨论你在从事数据仓库的设计时是如何进行规划和分析的，详细描述数据仓库设计、数据集成和测试，以及部署数据仓库的过程。
- ③ 分析并讨论你在数据仓库设计与实现过程中遇到过的主要问题，并简要展望数据仓库技术的发展趋势。

写作要点

数据仓库是面向业务主题的、集成的、不可更新的，并且会随着时间、业务的变化而不断变化，这些特性决定了数据仓库的设计不同于传统的数据库设计方法。数据仓库系统的原始需求通常不是很明确，并且不断变化、增加，所以，数据仓库的建立是一个过程，从建立简单的基本框架着手，不断丰富和完善整个系统。这一过程将由以下几部分构成：需求分析、概念模型设计、逻辑模型设计、物理模型设计和数据仓库生成。关于数据仓库的详细描述请参看“2.1.9 数据仓库与数据挖掘”。

在撰写本文时，需要注意以下几个方面：

- ① 在文章中首先用 400~600 字的篇幅简要叙述作者参与分析和开发的软件项目概要和所担任的工作。
- ② 具体讨论在从事数据仓库的设计时是如何进行规划和分析的，详细描述数据仓库设计、数据集成和测试，以及部署数据仓库的过程。
- ③ 简要讨论在数据仓库设计与实现过程中遇到过的主要问题，以及解决问题的方法。
- ④ 最后，简要展望数据仓库技术的发展趋势。

13.2.10 论计算机网络的安全性

随着计算机网络，特别是各类互联网络的普遍使用，计算机网络的安全性及其实现方法受到了人们极大的关注。为了保障网络服务的可用性和网络信息的完整性，目前已提出了许多对网络服务的种类与范围等实施控制的技术与方法。

- ① 简述你参与过的计算机网络应用系统的概要和你所担任的工作，包括你在网络安全性与保密性方面所做过的主要工作。
- ② 详细论述你采用的保障网络安全性的技术与方法，并且着重说明你所采用的专用软件、硬件与管理措施的综合解决方案。
- ③ 论述保障网络安全性与提高网络服务效率之间的关系，同时简要评估你所采用的网络安全性措施的效果。

写作要点

计算机网络应用系统不仅与计算机系统结构有关，还与运行的环境、人员因素和社会因素有关，具体来说，应该包括以下内容：

- ① 硬件安全，是指保护计算机系统硬件的安全，保证其自身的可靠性。
- ② 软件安全，是指保护软件和数据不被篡改、破坏和非法复制。通常又可分为操作系统安全、数据库安全、网络软件安全和应用软件安全。
- ③ 运行安全，是指保护系统能够连续和正常运行。
- ④ 安全立法，是指利用国家机器，通过网络与信息安全立法，对计算机网络方面的犯罪行为进行约束和制裁。

从技术方面而言，硬件、软件的安全主要是通过系统供应商来保证的，在选择时应该经过系统的检查与评测，在设计时充分考虑到网络性的安全，相对而言网络安全的矛盾更加凸显。另外由于计算机网络的运行主要是通过网络和信息技术来支撑的，因此计算机网络安全运行的核心内容就是网络上的信息安全问题。在网络信息安全方面的主要特征如下：

- ① 保密性。保证信息不泄露给未经授权的用户或供其利用。
- ② 完整性。防止信息被未经授权的人篡改，保证真实的信息从真实的信源无失真地传到真实的信宿。
- ③ 可用性。保证信息及信息系统确实为授权使用者所用，防止由于计算机病毒或其他人为因素造成网络 and 系统无法正常运行而拒绝服务或为敌手所用。
- ④ 控性。对信息内容及信息系统实施安全监控管理，防止非法修改。
- ⑤ 抗抵赖性。保证信息行为人不能否认自己的行为。

而在网络安全的防护方面，主要的技术手段包括：防火墙、入侵检测、病毒扫描、安全扫描、日志审计、网页防篡改、私自拨号检测、PKI 技术和服务等。关于这些方面的详细描述请参看“5.1.3 网络安全”。

本文的写作要点，大致包括以下几个方面：

- ① 简述你参与过的计算机网络应用系统的概要（并应适当地说明该系统的应用背景与目标）和你所担任的工作，包括你在网络安全性与保密性方面所做过的主要工作。
- ② 具体叙述采用的保障网络安全性的技术与方法，并且着重说明你所采用的专用软件、硬件与管理措施的综合解决方案。
- ③ 根据系统实际运行情况，简要评估所采用的网络安全性措施的效果。
- ④ 简单论述保障网络安全性与提高网络服务效率之间的关系（显然是一种相互制约的关系）。

13.2.11 论 Web Service 技术的应用与发展趋势

随着 Internet 技术的发展，一种新的分布式计算模型——Web Service 应运而生。

在支持基于 Web 的新型应用方面, Web Service 体系结构具有独特的优势和广泛的应用前景,但在实际应用中仍然存在一些问题。

请围绕“Web Service 技术的应用与发展趋势”论题,依次从以下 3 个方面进行论述:

- ① 简述你参与的此类应用项目,以及你在其中所担任的主要工作。
- ② 论述你所参与的应用项目的特点及选用 Web Service 体系结构的准则,结合实例详细叙述 Web Service 技术在你参与的项目中的应用、所带来的好处及遇到的问题。
- ③ 分析并讨论 Web Service 技术的特点、应用前景及其存在的主要问题,并简要展望 Web Service 技术的发展趋势。

写作要点

本题涉及 Web Service 技术,关于该技术的详细介绍请参看“8.1.6 面向服务的架构(SOA)”。

完成本文的写作,需要注意以下要点:

- ① 用 400~600 字简单介绍你参与分析和开发的 Web Service 项目,以及你所担任的主要工作。
- ② 具体论述你所参与的应用项目的特点,以及选用 Web Service 体系结构的准则,结合实例详细叙述 Web Service 技术在你参与的项目中的应用、所带来的好处及遇到的问题。
- ③ 分析并讨论 Web Service 技术的特点、应用前景及其存在的主要问题,并简要展望 Web Service 技术的发展趋势。

13.2.12 论系统的可靠性

系统的可靠性是指系统在运行过程中抗干扰(系统的故障和人为的故障)和保证其正常工作的能力,它体现在系统工作的连续性和工作的正确性。一般说,系统的可靠性包括查错、纠错能力;在错误干扰下不发生崩溃及重新恢复和重新启动的能力;硬件与软件的可靠性;保证数据的存储与计算精度等。

根据你实际参与设计的经验论述下列 3 个问题:

- ① 简要叙述你参与设计的系统的概要和你所担任的工作。
- ② 具体叙述你参与设计的系统中,为了提高系统的可靠性,遇到了哪些问题,你是如何采取措施,解决这些问题的。
- ③ 简要叙述你的这些措施的效果如何,你现在认为还有哪些应进一步改进的地方,以及如何改进。

写作要点

本题所涉及的系统可靠性知识可参看“第 11 章 系统配置与性能评价”。

在撰写本文时,要注意以下几个方面:

① 在文章中首先用 400~600 字的篇幅简要叙述自己参与设计的系统的概要和所担任的工作。

② 简单说明系统可靠性的重要性。

③ 具体叙述为了提高系统的可靠性,采用了哪些措施,遇到了哪些问题,是如何解决这些问题的。

④ 最后应根据项目的实际执行情况,说明系统可靠性的效果如何(要有具体数据),以及如何改进。

13.2.13 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。软件可靠性与其他质量属性一样,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 3 个方面进行论述。

① 概要叙述你参与实施的软件开发项目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及你如何应用到实际项目中,效果如何。

写作要点

这是一道关于软件可靠性设计的试题,主要考查影响软件可靠性的因素及软件可靠性设计技术和应用。本题需要特别注意的一点是:文章不能以“系统可靠性设计与应用”为主线组织内容。“系统可靠性设计”与“软件可靠性设计”的区别在于:系统可靠性包括了系统硬件的可靠性和系统软件的可靠性。

软件的可靠性设计技术主要有:恢复块方法、N 版本程序设计、防卫式程序设计。相关技术知识请参看“11.1.4 容错技术”。

13.2.14 论软件需求获取技术及应用

软件需求是指用户对新系统在功能、行为、性能、设计约束等方面的期望。软件需求获取是一个确定和理解不同的项目干系人的需求和约束的过程。需求获取是否科学、准备充分,对获取的结果影响很大,这是因为大部分用户无法完整地描述需求,而且也不可能看到系统的全貌。因此,掌握各种不同的需求获取技术,并且熟练地在

实践中运用它，并与用户有效合作，是十分重要的。

请围绕“需求获取技术及应用”论题，依次从以下三个方面进行论述。

① 简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 详细说明目前有哪些比较常用的需求获取技术，说明每种需求获取技术的基本方法。

③ 详细论述在你参与分析和开发的软件项目中所采取的需求获取技术，以及选取这些技术的原因，并说明需求获取的具体实施步骤。

写作要点

一、常用的需求获取技术：用户访谈、用户调查、现场观摩、阅读历史文档、联合讨论会（联合需求计划）等。

（1）用户访谈。用户访谈是最基本的一种需求获取手段，其形式包括结构化和非结构化两种，结构化是指事先准备好一系列问题，有针对地进行；非结构化则是只列出一个粗略的想法，根据访谈的具体情况发挥。最有效的访谈是结合这两种方法进行。用户访谈具有良好的灵活性，用较宽广的应用范围，但是也存在着许多困难，诸如客户经常较忙，你难以安排到时间；面谈时信息量大，记录较为困难；沟通需要很多技巧，同时需要分析员有足够的领域知识；另外，在访谈时会遇到一些对于组织来说比较机密和敏感的话题。因此，这看似简单的技术，也需要分析人员拥有足够多的经验和较强的沟通能力。

（2）用户调查。用户访谈时最大的难处在于很多关键的人员时间有限，不容易安排过多的时间；而且客户面经常较广，不可能一一访谈。因此，我们就需要借助用户调查，通过精心设计要问的问题，然后下发到相关的人员手里，让他们填写答案。这样就可以有效地克服前面提到的两个问题。但是与用户访谈相比，用户调查最大的不足就是缺乏灵活性；而且双方未见面，分析人员无法从他们的表情等其他动作来获取一些更隐性的信息；还有就是客户有可能在心理上会不重视一张小小的表格，不认真对待从而使得反馈的信息不全面。因此较好的做法是将这两种技术结合使用。具体来说，就是先设计问题，制作成为用户调查表，下发填写完后，进行仔细的分组、整理、分析，以获得基础信息，然后再针对这个结果进行小范围的用户访谈，作为补充。

（3）现场观摩。对于许多较为复杂的流程和操作而言，是比较难以用言语表达清楚的，而且这样做也会显得很低效。因此，针对这一现象，分析团队可以就一些较复杂、较难理解的流程、操作采用现场观摩的方法来获取需求。具体来说，就是走到客户的工作现场，一边观察，一边听客户的讲解，甚至可以安排人员跟随客户工作一小段时间。这样就可以使得分析人员更加直观地理解需求。

（4）阅读历史文档。这种方式也称为“文档考古”。对于一些数据流比较复杂的，工作表单较多的项目，有时是难以通过语言，或者通过观察来了解需求细节的。这个时候就可以借助于阅读历史文档的方法，对历史存在的一些文档进行研究，从中获得所需的信息。这个方法的主要风险是历史的文档可能与新系统的流程、数据有一些不

吻合的地方，并且还可以承载一些原有系统的缺陷。要想有效地避免和发现这些问题，就需要分析人员能够运用自己的聪明才智，将其与其他需求捕获技术结合对照。还有一个负面因素就是，这些历史的文档中记载的信息有可能涉及客户的商业秘密，因此对数据信息的保密也是分析人员基本的职业道德。

(5) 联合讨论会。这是一种相对来说成本较高的需求获取方法，但也是十分有效的一种。它通过联合各个关键客户代表、分析人员、开发团队代表一起，通过有组织的会议来讨论需求。通常该会议的参与人数为 6~18 人，召开时间为 1~5 小时。在会议之前，应该将与讨论主题相关的材料提前分发给所有将要参加会议的人。在会议开始之后，首先应该花一些时间让所有的与会者互相认识，以使交流在更加轻松的气氛下进行。会议的最初，就是针对所列举的问题进行逐项专题讨论，然后对原有系统、类似系统的不足进行开放性交流，第三步则是大家在此基础上对新的解决方案进行一番设想，在过程中将这些想法、问题、不足记录下来，形成一个要点清单。第四步就是针对这个要点清单进行整理，明确优先级，并进行评审。这种联合讨论会将会起到群策群力的效果，对于一些问题最有歧义的时候、对需求最不清晰地领域都是十分有用的一种技术。而且最大的难度就是会议的组织，要做到言之有物，气氛开放，否则将难以达到预想的效果。

二、结合项目实际工作，举例说明你在获取需求时分别采用了哪些需求获取技术；详细说明你选择这些技术的原因及具体实施步骤。

13.3 实战练习题

试题 1 论面向服务的体系结构在系统集成中的应用

面向服务的体系结构 (Service Oriented Architecture, SOA) 作为一种体系结构模型，将应用程序的不同功能单元通过一些良好定义的接口联系起来。接口是采用中立的方式进行定义的，它独立于实现服务的硬件平台、操作系统和编程语言。这使得构建服务可以以一种统一和通用的方式进行交互。

请围绕“SOA 在系统集成中的应用”论题，依次从以下的 3 个方面进行论述：

- ① 概要叙述你参与分析与开发的系统集成项目，以及你在其中所担任的主要工作。
- ② 详细论述 SOA 中的关键技术，以及你熟悉的工具和环境对 SOA 的支持。
- ③ 通过你的切身实践详细论述 SOA 在系统集成中发挥的作用和优势。

试题 2 论软件的静态演化和动态演化及其应用

软件演化 (Software Evolution) 是指软件在其生命周期内的更新行为和过程。演化是一系列贯穿软件生命周期始终的活动，系统需求改变、功能实现增强、新功能加入、软件架构改变、软件缺陷修复、运行环境改变均要求软件系统能够快速适应变化，具有较强的演化能力。软件静态演化 (Static Evolution) 和动态演化 (Dynamic Evolution)

是目前软件演化的两种重要类型。

请围绕“软件的静态演化和动态演化及其应用”论题，依次从以下3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及你在其中所担任的主要工作。
- ② 请分别对软件静态演化和动态演化的特点进行论述，说明两种软件演化类型各自的优缺点及其应用场合，并举例说明各自的常见演化技术手段。
- ③ 具体阐述你参与管理和开发的项目中所进行的软件演化活动的特点、演化的类型，以及所采取的对应演化技术手段，说明具体实施过程及实际应用的效果。

试题3 论大规模分布式系统缓存设计策略

大规模分布式系统通常需要利用缓存技术减轻服务器负载、降低网络拥塞、增强系统可扩展性。缓存技术的基本思想是将客户最近经常访问的内容在缓存服务器中存放一个副本，当该内容下次被访问时，不必建立新的数据请求，而是直接由缓存提供。良好的缓存设计，是一个大规模分布式系统能够正常、高效运行的必要前提。在进行大规模分布式系统开发时，必须从一开始就针对应用需求和场景对系统的缓存机制进行全面考虑，设计一个可伸缩的系统缓存架构。

请围绕“大规模分布式系统缓存设计策略”论题，依次从以下3个方面进行论述：

- ① 概要叙述你参与实施的大规模分布式系统开发项目及你所担任的主要工作。
- ② 从不同的用途和应用场景考虑，请详细阐述至少两种常见的缓存工作模式，并说明每种工作模式的适应场景。
- ③ 阐述你在设计大规模分布式系统的缓存机制时遇到了哪些问题，如何解决。

试题4 论基于REST服务的Web应用系统设计

REST (REpresentational State Transfer) 是指从几种基于网络的架构风格衍生出来的一种混合架构风格，它是目前互联网的核心架构风格。基于REST服务 (RESTful Service) 的Web应用系统设计任务主要包括：识别并设计REST风格的服务，采用面向服务的思想进行REST服务集成。采用这种方法设计的Web应用系统能够结合REST风格和面向服务思想的优点，近年来受到了广泛的关注。

请围绕“基于REST服务的Web应用系统设计”论题，依次从以下3个方面进行论述：

- ① 概要叙述你参与实施的Web应用系统开发项目及你所承担的主要工作。
- ② 简要叙述与传统的Web服务相比，采用REST服务构建的Web应用具有哪些优势和不足。
- ③ 阐述你在设计基于REST服务的Web应用系统时遇到了哪些问题，如何解决。

试题5 论中间件技术在软件开发中的作用

随着软件应用越来越复杂，软件开发“一切从零开始”的开发模式也越来越不能满足应用的要求。一些著名的软件公司推出自己的中间件产品，如通信中间件、数据

库中间件等为软件开发提供有力的支持。

请围绕“中间件技术在软件开发中的作用”这个论题，依次从以下3个方面进行论述：

- ① 概要叙述你参与分析和开发的应用项目，以及你所担任的主要工作。
- ② 具体讨论你在软件开发过程中选取中间件的原则和标准，结合实例详细叙述中间件技术在你的软件开发中的作用。
- ③ 假如你在一个软件公司中能够起到影响公司决策的作用，请你结合市场和技术因素，为你的公司设计一个有自主知识产权的中间件的方案，并评价你的方案。

试题6 论基于 Web 的数据库应用系统的开发技术

浏览器和 HTTP 协议在全球因特网的成功应用，促进了企业中的 B/S 结构的迅速推广。基于 Web 的数据库应用系统通常采用三层（或更多层）C/S 结构，允许用于各类不同的平台和选择相应的数据库应用环境。

请围绕“基于 Web 的数据库应用系统的开发技术”论题，依次从以下3个方面进行论述：

- ① 概要叙述你参与分析和开发的软件项目，以及你所担任的主要工作。
- ② 具体讨论你在基于 Web 的数据库应用系统开发中所采用的主要技术，包括开发过程中遇到的问题和所采取的措施。
- ③ 分析你所采用的开发技术的具体效果。现在你认为应当做哪些方面的改进，以及如何加以改进？

试题7 论数据库的安全性设计

通常，信息系统的安全性指的是采取必要的安全保护方面的技术和管理措施，保护硬件、软件和数据不因偶然的或恶意的原因而受到破坏、更改和泄露。其中数据库的安全性设计有着十分重要的作用（包括物理安全与逻辑安全）。

根据你实际参与开发和设计的经验，论述下列3个问题：

- ① 简述你开发与设计的软件概要和你所担任的工作。
- ② 具体叙述你在数据库应用系统开发中是如何使用安全性设计技术的，包括采用过哪些主要技术，遇到过哪些实际问题，采取过哪些相应的措施。
- ③ 简述你所采用的数据库安全性设计技术的效果，以及你有哪些主要的使用体会和进一步的设想。

试题8 论数据挖掘技术的应用

随着信息技术的高速发展，各组织机构积累的数据量急剧增长。如何从海量的数据中提取有用的知识成为当务之急。数据挖掘（Data Mining）就是为顺应这种需要应运而生发展起来的数据处理技术，是知识发现的关键步骤。数据挖掘就是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程。

请围绕“论数据挖掘技术的应用”论题，依次从以下3个方面进行论述：

- ① 概要叙述你参与分析和开发的软件项目及你所担任的主要工作。
- ② 数据挖掘的主要任务是什么？具体论述你在项目中使用数据挖掘技术所解决的问题。
- ③ 数据挖掘的方法主要有哪些？分析并讨论你所选择的数据挖掘方法，简述其具体实现过程和实际应用效果。

试题9 论企业信息系统的的功能

企业信息化能够实现高效率、低成本的运行，越来越受到人们的重视。目前，它已经渗透到企业活动的各个阶段。在传统的企业活动中，所涉及的不同企业或企业内部不同人员在相互交往中往往需要直接见面。但在企业信息化中，更多的是通过计算机网络这一媒介，而不需要直接见面。这就要求在设计 and 实现企业信息系统时除了针对具体业务需求进行详细的分析，保证满足具体的业务需求之外，还要加强信息系统安全方面的考虑。

请围绕“企业信息系统的的功能”论题，依次从以下3个方面进行论述：

- ① 概要叙述你参与分析设计的企业信息系统及你所担任的主要工作。
- ② 深入讨论在企业信息系统中，通常需要着重解决的安全问题。
- ③ 结合你所参与分析设计的企业信息系统及其特点，详细论述你所采用的企业信息系统安全解决方案，并客观地分析、评价你的解决方案。

试题10 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3个方面进行论述：

- ① 简要概述你参与实施的软件开发项目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。

13.4 练习题写作要点

13.4.1 试题1写作要点

面向服务的体系结构是一种新的体系结构风格，它具有松耦合和面向软件服务的特点，具有很高的重用性和灵活性。关于 SOA 的详细介绍请参看“8.1.6 面向服务的架构（SOA）”。

在撰写本文时，要注意以下几个方面：

① 简单介绍你参与分析与开发的系统集成项目情况和背景，以及你在其中所担任的主要工作，说明为什么要使用 SOA。

② 详细论述 SOA 中的关键技术，以及你熟悉的工具和环境对 SOA 的支持。要注意的是不要逐个地对技术进行讨论，而只是根据你的项目实际情况，具体地讨论 2~3 个技术的应用就可以了。

③ 根据你的项目应用情况，详细介绍 SOA 在系统集成中发挥的作用和优势。

13.4.2 试题2写作要点

本题考查的知识点是软件演化。一个软件系统开发完毕正式投入使用之后，如果需求发生变化，或者要将该系统移植到另一个环境运行，且新环境的需求也有相应的变化时，就要对软件进行修改，这就是软件演化。软件演化是一个程序不断调节以满足新的软件需求的过程，也就是对一个已有软件不断进行修改、补充、完善以适用新需求和环境变化的过程。由于软件演化一词并不多见，所以难倒了很多考生。其实换一种讲法，可能大家就倍感亲切了——“软件升级”，其实演化的本质就是在升级。既然是升级，静态演化与动态演化是怎么回事也就好理解了，即升级时是否停止系统的运行。所以如果有了上面的基础概念理解，写该论文的方向也就明晰了。

文章共分 3 个部分组织内容。

第一部分，简要叙述作者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。

第二部分，偏理论，需要介绍静态演化和动态演化的基本情况，写作时可参考以下材料。

静态演化（Static Evolution），静态演化是指软件在停机状态下的演化。其优点是不用考虑运行状态的迁移，同时也没有活动的进程需要处理。然而停止一个应用程序就意味着中断它提供的服务，造成软件暂时失效。

软件静态演化是指发生在应用程序停止时的软件修改和更新，即一般意义上的软件维护和升级。静态演化的优点是没有状态迁移或活动线程的问题要解决，缺陷是停止应用程序意味着停止它所提供的服务，也就是使软件系统暂时失效。在软件交付之后，静态演化（类似于一般意义上的软件维护）就成为软件变更的一个常规过程。变更可以是一种更正代码错误的简单变更，也可以是更正设计错误的较大范围的变更，

还可以是对描述错误进行修正或提供新需求这样的重大改进。有3种不同的软件维护：改正性维护、适应性维护和完善性维护。维护过程一般包括变更分析、版本规划、系统实现和向客户交付系统等活动。

在面向对象技术中，使用子类型方法来扩展程序，它适合于软件静态演化和代码重用。子类型化一个类意味着保留父类中的参数和方法，并尽可能地增加新的参数和方法。另外，使用重载和多态性作为主要的演化机制。实际上，建立类的新版本，最简单的机制是创建它的子类，然后重载需要变更的方法，最后，使用多态性调用新创建的方法。在基于构件的软件技术中，构件采取接口和实现相分离技术，构件之间只能通过接口进行通信，这使得具有兼容接口的不同构件实现可以相互取代，从而成为软件静态演化的一条途径。

动态演化（Dynamic Evolution），动态演化是指软件在执行期间的软件演化。其优点是软件不会存在暂时的失效，有持续可用性的明显优点。但由于涉及状态迁移等问题，比静态演化从技术上更难处理。

动态演化是最复杂也是最有实际意义的演化形式。动态演化使得软件在运行过程中，可以根据应用需求和环境变化，动态地进行软件的配置、维护和更新，其表现形式包括系统元素数目的可变性、结构关系的可调节性和结构形态的动态可配置性。软件的动态演化特性对于适应未来软件发展的开放性、动态性具有重要意义。

动态演化是指软件在运行期间的演化。在许多重要的应用领域中，如金融、电力、电信及空中交通管制等，系统的持续可用性是一个关键性的要求，运行时刻的系统演化可减少因关机和重新启动而带来的损失和风险。此外，越来越多的其他类型的应用软件也提出了运行时刻演化的要求，在不必对应用软件进行重新编译和加载的前提下，为最终用户提供系统定制和扩展的能力。

动态演化可分为两种类型：预设的和非预设的。在 Web 环境中，软件应用常常需要处理多种类型的信息，因此它们常被设计为可以动态下载并安装插件以处理当前所面临的新类型的信息；而分布式 Web 应用也常常需要增减内部处理节点的数目以适应多变的负载。这些动态改变都是软件设计者能够预先设想到的，可实现为系统的固有功能。另有一些必须对系统配置进行修改和调整的情况是直到系统投入运行以后才发现的，这就要求系统能够处理在原始设计中没有完全预料到的新需求。这种情况下一般需要关闭整个系统，重新开发、重新装入并重新启动系统。然而，为了进行局部的修改而关闭整个系统在某些情况下是不允许的（如关键运行系统）或者代价太高。精心设计的动态演化技术可以在不关闭整个系统的前提下修改系统的结构配置，并尽量使未受影响的部分继续工作以提高系统的可用度。

为支持软件的动态演化性，已在语言、机制和环境等方面做了大量工作。在程序语言的层次上，引进各种机制以支持软件动态演化，例如，动态装载技术允许增加代码到已运行的程序中，延迟绑定是在运行时而不是编译时决定类和对象的绑定。Java hotswap 允许在运行时改变方法：当一个方法终止时这个方法的新版本可以有效地替换旧版本，在类层次上代码的二进制兼容被支持。Gilgul 语言也允许更换运行时对象。但程序语言层次上的动态演化机制仅局限于函数、类方法和对象等小粒度的替换，只支持预设的有限变更，变更由事件触发。

通过标准化运行级构件的规约, 依靠构件运行平台(中间件平台)提供的基础设施, 使软件在构件层次上的动态演化成为可能。中间件中具有的如命名服务、反射技术和动态适配等机制, 为运行态构件的动态替换和升级提供支撑, 从而推动了软件动态演化的发展。命名服务就是给构件实例提供一个名称, 以便客户通过这些名称来获取构件实例。对工业标准构件 EJB 和 CORBA 构件的引用都可以通过中间件平台的命名服务进行。同一构件标识可以被映射到多个构件实例, 从而根据具体情境对某一名字的构件引用导向到不同的构件实例。反射技术是系统的一种自描述(Self-Representation)和自推理的技术, 它提供了关于自身行为的表示, 这种表示可以被检查和调整, 且与它所描述的系统行为是因果相联(Causally Connected)的。因果相联, 意味着对自身表示的改动将立即反映在系统的实际状态和行为中, 反之亦然。将反射性引入中间件能够以可控的方式开放平台内部的实现, 从而提高中间件的定制能力和运行时的适应能力。动态适配机制中比较著名的是 CORBA 提供的动态接口服务: 动态调用接口 DII 和动态骨架接口 DSI。前者支持动态客户请求调用, 而后者支持将请求动态指派(Dispatch)给构件。因此, 软件构件化技术使得软件具有良好的构造性, 软件演化的粒度更大。中间件技术则为基于构件的软件动态演化提供了坚实的基础设施和方便的操作界面。

第三部分需要结合作者自身的项目情况进行描述, 这部分应是详细描述的部分。

13.4.3 试题 3 写作要点

① 论文中要具体介绍项目的总体需求(特别是应用需求中对缓存机制的要求)、系统的逻辑与物理架构、采用的技术等内容和担任的实际工作。

② 从不同的用途和应用场景来考虑, 大体上可以将缓存分为 3 种工作模式, 即单实例缓存模式(Single Instance)、复制模式(Replication Cache)和分区模式(Partition Cache)。每种工作模式都有其适应的场景和优缺点。

- 单实例模式。单实例模式是一种较为简单的缓存模式, 多个应用服务器共享一个中央的缓存服务器。通过共享缓存的数据, 能够极大提高系统的性能。该模式的主要限制在于缓存服务器的内存大小和节点增加之后服务器的处理能力和网络带宽。该模式的适应场景是: 对缓存的要求比较简单; 系统的吞吐量和数据量不大; 性能要求不高。
- 复制模式。复制模式将缓存的数据复制到多台机器上, 对于单一缓存服务器性能出现问题的情况下, 可以通过缓存复制的方式将压力分解到多个缓存服务器。该模式的工作原理是: 缓存客户端可以访问自己的缓存服务器, 多个缓存服务器之间的数据是彼此同步的, 对于性能要求更高的场景, 这样的部署架构能够获得更高的吞吐能力。该模式的适应场景是: 数据量不是特别大; 需要极高的性能; 数据改动的频率不是特别大。
- 分区模式。当需要缓存的数据已经超过一台服务器的内存上限时, 可以考虑采用分区模式对数据进行线性缩放, 也就是通过增加缓存服务器来解决数据增长和压力增加的情况。在分区模式中, 其架构是无分享架构(Shared Nothing Architecture, SNA), 每个节点之间数据彼此独立, 一个节点出现故障后不会影响到其他节点。在出现某个节点宕机或者其他故障的情况下, 致使这部

分的分区缓存无法使用，并不妨碍其他数据节点数据的正常工作。该模式的适应场景是：总体数据量较大，已经超出了单个缓存服务器的内存上限；系统缓存要求具有很大的可伸缩性；客户端数量庞大，单个客户端对缓存数据的数据量要求不大。

③ 进行大规模分布式系统缓存机制设计时可能遇到的问题包括如何缓存服务器的工作模式选择；高可用性的设计考虑；缓存一致性与分布式算法；对象状态同步的考虑；缓存钝化/激活/过期和初始化，等等。

13.4.4 试题 4 写作要点

① 论文中要具体介绍项目的总体需求（特别是质量属性需求）、Web 应用系统的逻辑与物理拓扑结构、采用的技术等内容和承担的实际工作。

② REST (REpresentational State Transfer) 是指从几种基于网络的架构风格衍生出来的一种混合架构风格，目前 Web 的体系结构正是基于 REST 风格的。REST 风格中的特点是客户端 / 服务器、无状态、缓存、统一接口、分层系统和按需代码。REST 组件通过以一种数据格式转移资源的表述进行通信，可以基于接收者的能力和期待的内容，以及资源的性质动态地选择不同的表述。

与传统的 Web 服务相比，REST 服务主要有以下优势：

- REST 服务基于 W3C/IETF 的标准与规范（包括 HTTP、XML、URI 和 MIME 等），其实现技术简单、成熟。
- REST 服务基于 URI 和超链接技术，不需要通过集中式的服务信息仓库即可发现服务资源。
- REST 服务支持缓存，具有无状态的特性，这些使得 REST 服务能够支持大量客户端，构建的应用系统具有较强的伸缩性。
- REST 服务基于轻量级的 Web 框架，仅仅需要基本的开发工具支持，构建过程简单且成本较低。
- REST 服务的测试相对简单，采用浏览器即可完成服务功能测试。

与传统的 Web 服务相比，REST 服务主要存在如下不足：

- REST 服务倡导的 REST 风格与实际实现尚存在一定差距。例如，高层 REST 服务倡导使用 GET、PUT、POST 和 DELETE 所有 4 个统一接口，在 REST 实现部分通常只能采用 GET 和 POST 接口，因为大多数的代理和防火墙会屏蔽其他接口；并且 XHTML 表单中只能使用 GET 和 POST 接口。
- REST 服务要求所有的输入参数都必须在 URI 中传递，这样会产生对参数容量大小的限制（目前的大小是 4KB）。如果超出该数量，会导致 HTTP 协议错误（错误代码 414：Request-URI too long）。
- 在 URI 中表达复杂类型的参数比较困难，且目前对 URI 中的参数不存在一种公认的编组（marshalling）和解编（un-marshalling）方法。

③ 进行基于 REST 服务的 Web 应用系统的设计时可能遇到的问题包括：如何识别并设计 REST 风格服务；构建 REST 服务的运行环境，HTTP 服务器与应用服务器

选型等；富客户端表现方式及编程语言的选择；系统逻辑与物理拓扑结构的分析与设计等。

13.4.5 试题 5 写作要点

中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户/服务器的操作系统之上，管理计算资源和网络通信。从定义中可以看出：

- ① 中间件是一类软件，而非一种软件。
- ② 中间件不仅要实现互联，还要实现应用之间的互操作。
- ③ 中间件是基于分布式处理的软件，最突出的特点是其网络通信功能。

1. 中间件的分类

根据功能的不同，中间件可以细化为以下一些种类：

① 通信处理（消息）中间件。在不同平台之间通信，实现分布式系统中可靠的、高效的、实时的跨平台数据传输，称为消息中间件。这是中间件中唯一不可缺少的，是需求量最大的中间件产品，目前在大部分操作系统中已包含了其部分功能。

② 事务处理（交易）中间件。在分布式事务处理系统中要处理大量的事务，常常在系统中要同时进行上万笔事务。在联机事务处理系统（OLTP）中，每笔事务常常要多台服务器的程序顺序地协调完成，一旦中间发生某种故障时，不但要完成恢复工作，而且要自动切换系统，达到系统永不停机，实现高可靠性运行；同时要使大量事务在多台应用服务器实时并发运行，并进行负载平衡地调度，实现昂贵的小型机和大型计算机系统同等的功能，为了实现这个目标，要求系统具有监视和调度整个系统的功能。一个事务处理平台，根据 X/OPEN 的参数模型规定，应由事务处理中间件、通信处理中间件，以及数据存取管理中间件 3 部分组成。

③ 数据存储管理中间件。在分布式系统中，重要的数据都集中存放在数据服务器中，它们可以是关系型、复合文档型、具有各种存放格式的多媒体型，或者是经过加密或压缩存放的，该中间件为在网络上虚拟缓存、格式转换、解压等带来了方便。

④ Web 服务中间件。浏览器图形用户界面已成为公认规范，然而它的会话能力差、不能做数据写入、受 HTTP 协议的限制等，必须进行修改和扩充，这形成了 Web 服务器中间件。

⑤ 安全中间件。一些军事、政府和商务部门上网的最大障碍是安全保密问题，而且不能使用国外提供的安全措施（如防火墙、加密、认证等），必须用国产的产品。产生不安全因素是由操作系统引起的，但必须要用中间件去解决，以适应灵活多变的要求。

⑥ 跨平台和体系结构的中间件。当前，开发大型应用软件通常采用基于体系结构和构件的技术，在分布系统中，还需要集成各节点上的不同系统平台上的构件或新老版本的构件，由此产生了体系结构中间件，功能最强的是 CORBA，可以跨任意平台，但是太庞大；Java Bean 较灵活简单，很适合于做浏览器，但运行效率差；DCOM

模型主要适合 Windows 平台,已广泛使用。实际上国内新建系统主要是 UNIX(包括 Linux)和 Windows,因此针对这两个平台建立相应的中间件要实用得多。

⑦ 专用平台中间件。为特定应用领域设计参考模式,建立相应体系结构,配置相应的构件库和中间件,为应用服务器开发和运行特定领域的关键任务(如电子商务、网站等)。

⑧ 其他中间件。现在出现了一些中间件,如数据流中间件、门户中间件,以及为某些专业领域如银行、电信等开发的专用中间件。另外还有一些更高层的中间件,多用于系统整合,包括企业应用集成中间件、工作流中间件、门户中间件等是多种中间件的组合。

2. 中间件的优点及其应用

中间件作为一大类系统软件,与操作系统、数据库管理系统并称“三套车”,其重要性是不言而喻的,中间件的优点应该说是有目共睹的。它的优越性体现在以下几个方面:缩短应用的开发周期;节约应用的开发成本;减少系统初期的建设成本;降低应用开发的失败率;保护已有的投资;简化应用集成;减少维护费用;提高应用的开发质量;保证技术进步的连续性;增强应用的生命力。

具体地说,中间件屏蔽了底层操作系统的复杂性,使程序开发人员面对一个简单而统一的开发环境,减少了程序设计的复杂性,将注意力集中在自己的业务上,不必再为程序在不同系统软件上的移植而重复工作,从而大大减少了技术上的负担。

中间件带给应用系统的不只是开发的简便、开发周期的缩短,还减少了系统的维护、运行和管理的工作量,同时减少了计算机总体费用的投入。Standish 的调查报告显示,由于采用了中间件技术,应用系统的总建设费用可以减少 50%左右。在网络经济、电子商务大发展的今天,从中间件获得利益的不只是 IT 厂商,IT 用户同样是赢家,并且是更有把握的赢家。

其次,中间件作为新层次的基础软件,其重要作用是将不同时期、在不同操作系统上开发的应用软件集成起来,彼此像一个天衣无缝的整体协调工作,这是操作系统、数据库管理系统本身做不了的。中间件的这一作用,使得在技术不断发展之后,我们以往在应用软件上的劳动成果仍然物有所用,节约了大量的人力、财力的投入。

3. 中间件的发展趋势

中间件作为构筑企业信息系统和电子商务系统的基石和核心技术,向着标准化和构件化方向发展。具体来看,有以下 3 种发展趋势:

① 规范化。在中间件的发展过程中,做得最好的一件事情就是规范的制定。对于不同类型的中间件,目前都有一些规范可以遵循,如消息类的 JMS,对象类的 CORBA 和 COM/DCOM,交易类的 XA、OTS、JTA/JTS,应用服务器类的 J2EE,数据访问类的 ODBC 和 JDBC,Web 服务有 SOAP、WSDL、UDDI 等。这些规范的建立极大地促进了中间件技术的发展,同时保证了系统的扩展性、开放性和互操作。

② 构件化和松耦合。除了已经得到较为普遍应用的 CORBA、DCOM 等适应 Intranet 的构件技术外,随着企业业务流程整合和电子商务应用的发展,中间件技术朝

着面向 Web、松耦合的方式发展。基于 XML 和 Web 服务的中间件技术,使得不同系统之间、不同应用之间的交互建立在非常灵活的基础上。XML 是一种可扩展的源标识语言,它提供了一种新的定义标识语言标准。XML 技术非常适合于异构系统间的数据交换,因此在国际上已经被普遍采纳为电子商务的数据标准。而 Web 服务作为基于 Web 技术的构件,在流程中间件的控制和集成下可以灵活、动态地被组织成为跨企业的商务应用。

③ 平台化。目前,一些大的中间件厂商在已有的中间件产品基础上,都提出了完整的面向互联网的软件平台战略计划和应用解决方案。

4. 选用中间件的原则

① 技术成熟度。不同的中间件的发展历史都不同。由于中间件涉及的技术面广,与操作系统、网络、数据库和应用都有关系,因此中间件从诞生到成熟需要 2~3 年的时间。中间件的技术成熟度是一个很需要关注的问题。

② 先进性。中间件的技术先进性主要看它是不是符合国际的潮流,是否采用了各种标准化的做法。比如,它能不能支持 HTML、JavaScript、XML 这样一些国际标准的语言环境。因为大量的应用软件都是在语言环境下发展起来,现在计算机上的这些语言环境已经很成熟,手机上也开始有了 Java,机顶盒如果不支持这些标准的话,将来很难做到应用服务的通用。如果中间件技术上能达到国际开放标准水平的话,就很容易向上发展了。包括 MHP 和 MHEG 和美国的 OPENCABEL 在内的一系列中间件标准,只要中间件支持 JavaScript、HTML 和 XML 等通用语言,这 3 种常用中间件的主流模块都可以移植到我们的应用服务里。

③ 与遗留应用的结合度。如果选用中间件构造的应用与传统的遗留应用要建立联系的话,就必须考虑这个问题。

④ 使用的难易程度。每类中间件使用的难度也不尽相同。有些中间件只是你的应用的一部分,而有些中间件将会给你的应用一个新的体系结构。应用的现状、应用开发队伍的水平、中间件本身的复杂程度等,都会影响中间件的使用效果。

⑤ 成本。不同类的中间件的成本不同,如把消息中间件只是用在局域网上实现数据访问显然代价太大,把交易中间件用在非交易处理类系统中去传递消息同样不可取。另外也要考虑技术培训对开发成本带来的影响。

⑥ 技术方向。有的中间件比较传统,而有的中间件是发展方向,中间件的选择与对软件技术发展趋势的把握紧密相关。如果企业已决定用构件技术或 Web 技术,就要选择相关的中间件。

⑦ 适用性。好的中间件要带一个非常丰富的应用库。中间件的一个最大作用就是重复使用已经开发过的应用软件模块,通常这些模块都是以库单元的方法来实现的,中间件在实际应用中越成功,开发的应用程序越多,它带的库就越丰富,使用者就可以越简便地开发或修改自己需要的新服务,这就是知识的积累。

通过以上分析,可以对中间件技术有一个全面的了解,除了基础技术的了解,写好本文,还要注意以下要点:

① 用 400~600 字简单介绍你参与分析和开发的应用项目，以及你所担任的主要工作。

② 简单介绍中间件的种类及作用。

③ 具体讨论你在软件开发过程中选取中间件的原则和标准，结合实例详细叙述中间件技术在你的软件开发中的作用。

④ 结合市场和技术因素，为你的公司设计一个有自主知识产权的中间件的方案，并评价你的方案。

13.4.6 试题 6 写作要点

完成本题的写作，需要对 Web 开发的相关架构有充分的了解，相关知识可参看“8.1.5 分层架构”。除了解技术知识，在写作方面还要注意以下几个方面的要点：

① 在文章中首先用 400~600 字的篇幅简要叙述自己参与分析和开发的系统概要和所担任的工作。

② 具体讨论在基于 Web 的数据库应用系统开发中所采用的主要技术，包括开发过程中遇到的问题和所采取的措施。特别要注意描述三层结构的逻辑和物理部署，讨论数据是如何在三层之间传送的。

③ 最后要根据项目实际情况，分析所采用的开发技术的具体效果，你现在认为还有哪些可以改进的地方（例如，应用服务器性能的改进等），如何加以改进。

13.4.7 试题 7 写作要点

数据库的安全性是指保护数据不被破坏和不被非法窃取，包括物理安全性和逻辑安全性。

1. 物理安全性

要保证数据库的物理安全性，除了健全机房管理制度，提高防火、防盗、防雷击等基础设施以外，最重要的就是要有一个好的数据库备份策略。

数据库备份是指对重要的数据制作一份拷贝，并将其存储在安全的地方。备份系统为一个目的而存在：尽可能快地全盘恢复运行计算机系统所需的数据。根据系统安全需求可选择的备份机制有：场地内高速度、大容量自动的数据存储、备份与恢复；场地外的数据存储、备份与恢复；对系统设备的备份。备份不仅在网络系统硬件故障或人为失误时起到保护作用，还在入侵者非授权访问、攻击网络和破坏数据完整性时起到保护作用，同时也是系统灾难恢复的前提之一。

一般的数据备份操作有 3 种：一是全盘备份，即将所有文件写入备份介质；二是增量备份，只备份那些上次备份之后更改过的文件，这是最有效的备份方法；三是差分备份，备份上次全盘备份之后更改过的所有文件，其优点是只需两组磁带就可以恢复最后一次全盘备份的磁带和最后一次差分备份的磁带。

在确定备份的指导思想和备份方案之后，需要选择安全的存储媒介和技术进行数

据备份。一般来说, 备份的硬件设备包括磁盘、磁带和光盘塔。而利用磁盘备份的方法常用的是磁盘镜像、磁盘双工和磁盘阵列, 其中磁盘镜像是指系统中使用两块一样大小的磁盘, 在写入数据时, 先将数据写入源盘, 然后再写入镜像盘, 这样镜像盘中的数据与源盘的数据一模一样, 从而起到了备份的作用。而磁盘双工是两个互为备份的盘同时写入。磁盘阵列是指将多个类型、容量、接口甚至品牌一致的专用磁盘或普通磁盘连成一个阵列, 使其能以某种快速、准确和安全的方式来读写磁盘数据。其主要功能是可以提高网络数据的可用性及存储容量, 并能够免除单块硬盘故障所带来的灾难性后果。因此, 磁盘阵列是一种高效、快速、易用的网络存储备份设备。

备份的技术有“冷备份”和“热备份”两种。热备份是指“在线”的备份, 即下载备份的数据还在整个计算机系统和网络中, 只不过传到另一个非工作的分区或另一个非实时处理的业务系统中存放, 如前面提到的磁盘镜像、磁盘双工和磁盘阵列都属于热备份。

“冷备份”则是指“不在线”的备份, 下载的备份存放到安全的存储媒介中, 而这种存储媒介与正在运行的整个计算机系统和网络没有直接联系, 在系统恢复时重新安装, 有一部分原始的数据长期保存并作为查询使用。磁带就是一种冷备份的设备。热备份的优点是调用快, 使用方便, 在系统恢复中需要反复调试时更显优势, 但投资比较昂贵。冷备份则弥补了热备份的一些不足, 二者优势互补, 相辅相成, 冷备份在回避风险中还具有便于保管的特殊优点。

在进行备份的过程中常使用备份软件, 它一般应具有以下功能: 保证备份数据的完整性, 并具有对备份介质的管理能力; 支持多种备份方式, 可以定时自动备份, 还可以设置备份自动启动和停止日期; 支持多种校验手段(如字节校验、CRC 循环冗余校验、快速磁带扫描), 以保证备份的正确性; 提供联机数据备份功能; 支持 RAID 容错技术和图像备份功能。

与备份同样重要的是在备份完成之后对它们进行存储。备份的目的是为了在发生事故时能够及时进行恢复, 而事故的类型可能从用户误删数据到站点的灾难性毁坏。前者需要备份处于附近位置以便于快速恢复, 而后者却要求将备份存储在其他地方。实践证明, 最好的方法是将备份存储在其他地方, 最大程度地保护数据。当需要还原时, 再将备份及时取回。

日常备份制度描述了每天的备份以什么方式、使用什么备份介质进行, 是系统备份方案的具体实施细则。在制定完毕后, 应严格按照制度进行日常备份, 否则将无法达到备份方案的目标。数据备份有多种方式: 全备份、增量备份、差分备份、按需备份等。

- 全备份: 备份系统中所有的数据。
- 增量备份: 只备份上次备份以后有变化的数据。
- 差分备份: 只备份上次完全备份以后有变化的数据。
- 按需备份: 根据临时需要有选择地进行数据备份。

全备份所需时间最长, 但恢复时间最短, 操作最方便, 当系统中数据量不大时, 采用全备份最可靠。但是随着数据量不断增大, 我们将无法每天做全备份, 而只能在周末进行全备份, 其他时间我们采用所用时间更少的增量备份, 或采用介于两者之间

的差分备份。各种备份的数据量不同：全备份>差分备份>增量备份。在备份时要根据它们的特点灵活使用。

(1) 数据存储基本策略的设定

- 数据库全备份：选择在周五（或周六）自动进行。
- 数据库增量备份：每晚做批处理前和批处理后，由 UNIX 或其他主机系统执行，批处理人员触发或由系统自动执行。
- 文件全备份：将主机系统和其他服务器的数据做全备份，选择在周日自动进行。
- 文件增量备份：在周一到周四（或周五）之间备份文件的增量。
- 系统全量：在每月初的周日备份系统及数据库的全量。
- 系统增量：在其余的时间仅备份系统和数据库配置的增量。
- 跟踪备份：实时备份系统增量（事务日志备份）。

结合以上备份策略，从便于管理和恢复的角度考虑，制订数据分组和存储介质池对应策略，将数据分门别类放在不同编号的磁带组上，并建立不同的存取权限。建议建立：

- 数据库介质。专门放置数据库信息。
- 文件介质。放置除数据库以外的文件。
- 数据库日志和系统日志介质。安全稽核和系统恢复的重要数据记录需较长时间保存，建议由安全管理员在服务器端建立管理，形成与主机系统管理人员分离的运行数据记录。
- 系统介质。备份系统和系统配置等的变化，做到快速恢复系统。

(2) 数据备份工作过程

- 自动备份进程由备份服务器发动。每天晚上，自动按照事先制订的时间表所要求的内容，进行增量或全量的备份。由于每天的备份被适当地均衡，峰值备份数据量在周五（或周六）和周日发生。
- 批前及批后备份在 UNIX 或其他主机端发起。批处理人员输入触发备份命令，自动按要求备份数据库有关内容。
- 其他文件的自由备份。进入软件交互菜单，选择要求备份的文件后备份。
- 在线跟踪备份。配合数据存储管理软件的数据库在线备份功能，可定义实时或定时备份日志。
- 灾难备份异地存放介质的克隆。自动复制每日完成后的数据，以存放在异地做灾难恢复。

(3) 灾难恢复

灾难恢复措施在整个备份制度中占有相当重要的地位。因为它关系到系统在经历灾难后能否迅速恢复。灾难恢复操作通常可以分为两类：第一类是全盘恢复，第二类是个别文件恢复。此外还有一种值得一提的是重定向恢复。

- 全盘恢复。全盘恢复一般应用于服务器发生意外灾难导致数据全部丢失、系统崩溃或有计划的系统升级、系统重组等情况，也称为系统恢复。
- 个别文件恢复。由于操作人员的水平不高，个别文件恢复可能要比全盘恢复

常见得多,利用网络备份系统的恢复功能,我们很容易恢复受损的个别文件。只需浏览备份数据库或目录,找到该文件,触动恢复功能,软件将自动驱动存储设备,加载相应的存储媒体,然后恢复指定文件。

- 重定向恢复。重定向恢复是将备份的文件恢复到另一个不同的位置或系统上去,而不是进行备份操作时它们当时所在的位置。重定向恢复可以是整个系统恢复也可以是个别文件恢复。重定向恢复时需要慎重考虑,要确保系统或文件恢复后的可用性。

为了防止数据丢失,我们需要做好详细的灾难恢复计划,同时还要定期进行灾难演练。每过一段时间,应进行一次灾难演习。可以利用淘汰的机器或多余的硬盘进行灾难模拟,以熟练灾难恢复的操作过程,并检验所生成的灾难恢复软盘和灾难恢复备份是否可靠。

2. 逻辑安全性

数据库的逻辑安全模型如图 13-1 所示。与数据库使用直接有关的安全措施有:用户标识和鉴定、存取控制、审计、密码存储、视图的保护和日志监视等方式。

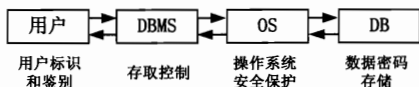


图 13-1 数据库的逻辑安全模型

1983 年,美国政府发布了“可信计算机系统评价标准(TCSEC)”,1994 年 4 月,美国国家计算机安全中心(NCSC)颁布了 TDI,即“可信计算机系统评估标准在数据库管理系统中的解释”,它将 TCSEC 扩展到数据库领域。在 TCSEC 中,系统安全从 4 个方面被划分为 7 个不同的安全级别,即 D, C1, C2, B1, B2, B3, A1,其中 A1 级别最高,D 级别最低。

TDI 沿用 TCSEC 的做法,从安全策略、责任、保证和文档 4 个方面来描述每级的安全性。一般认为,数据库及其他处理敏感商业信息的系统应达到 C2 级,在 C2 安全级上,能使用登录过程、审计跟踪并对与安全性有关的事件进行资源隔离;而处理保密的和要求更高敏感度的信息系统,安全级应达到 B1 级,这样能使用标记机制对特定的客体进行强制访问控制。

(1) 用户标识与鉴定

用户标识与鉴定是系统提供的最外层的安全保护措施。用户的标识和鉴定是两个不同的过程,标识过程负责将用户 ID 在系统中注册登记,而鉴别过程旨在将用户 ID 与真正的合法用户关联起来。目前有 3 种标识与鉴别方法。

① 利用用户的物理特征(声波、指纹、相貌、签名等)作为 ID,这在理论上是可靠的,但由于物理特征可能随时间变化且记录技术尚欠成熟等原因,使得该方法未能广泛用于数据库系统。

② 利用用户持有的证件,如身份证、机器可读卡片等,其缺点是证件可能被别人复制或冒用。

③ 利用用户知道某个能证明其身份的约定（如口令），这是当前数据库系统用来识别用户的主要方法。

- 用户名或用户标识号。在定义外模式时为每个用户提供一个用户代号存放在数据字典中。用户使用系统时，系统鉴别此用户是否是合法用户，若是，则可进入下一步的核实，否则不能使用系统。
- 口令。为了进一步核实用户，系统常常要求用户输入口令。为保密起见，用户在终端上输入的口令不显示在屏幕上，系统核对口令以鉴别用户身份。以上的方法简单易行，但用户名、口令容易被人窃取，因此还可以用更可靠的方法。
- 随机数检验。用户根据预先约定好的计算公式求出一个数值作为动态口令送入计算机，当这个值与系统算出的结果一致时，才允许进入系统。

此外，还可要求用户向鉴别机构交验机器可读身份卡片，或提供声波、指纹、签名等生物唯一性标志证明用户身份，鉴别机构对用户提供的信息进行鉴别，证明其有效性，或进行译码后给系统提供一个位串作为用户标识。

用户标识和鉴定可以重复多次。

（2）存取控制

在数据库系统中，为了保证用户只能存取有权存取的数据，系统要求对每个用户定义存取权限。存取权限包括两方面的内容：一方面是要存取的数据对象；另一方面是对此数据对象进行操作的类型。对一个用户定义存取权限就是要定义这个用户可以在哪些数据对象上进行哪些类型的操作。在数据库系统对存取权限的定义称为“授权”，这些授权定义经过编译后存放在数据库中。对于获得使用权又进一步发出存取数据库操作的用户，系统就根据事先定义好的存取权限进行合法权检查，若用户的操作超出了定义的权限，系统就拒绝执行此操作，这就是存取控制。

授权编译程序和合法权检查机制一起组成了安全性子系统。

在非关系系统中，用户只能对数据进行操作，存取控制的数据对象也仅限于数据本身。而在关系数据库系统中，DBA 可以把建立和修改基本表的权限授予用户，用户可利用这种权限来建立和修改基本表、索引、视图，因此，关系系统中存取控制的数据对象不仅有数据本身，还有模式、外模式、内模式等内容。

（3）审计

审计是现代计算机系统中必不可少的功能之一，其主要任务是对用户（包括应用程序）使用系统资源（包括软硬件和数据）的情况进行记录和审查，一旦发现问题，审计人员通过审计跟踪，可望找出原因，追查责任，防止类似问题再度发生。因此，审计往往作为保证数据库安全的一种补救措施。

数据库系统中的审计工作包括如下几种：

- 设备安全审计。它主要审查关于系统资源的安全策略、各种安全保护措施及故障恢复计划等。
- 操作审计。对系统的各种操作（特别是一些敏感操作）进行记录、分析。记

录内容包括：操作的种类、所属事务、所属进程、用户、终端（或客户机）、操作时间、审计日期等。

- 应用审计。审计建立于数据库之上的整个应用系统的功能、控制逻辑、数据流是否正确。
- 攻击审计。对已发生的攻击性操作及危害系统安全的事件（或企图）进行检测和审计。

上述各种审计所用技术大致可分为以下 3 类：

- 静态分析系统技术。审计者通过查阅各种系统资源（软硬件、数据）的说明性文件，例如，软件的设计说明书、流程图等来了解整个系统，甚至定位出一些易被攻击的薄弱环节。
- 运行验证技术。运行验证的目的是保证系统控制逻辑正确，各类事务能有效执行。该技术一般又细分为实际运行测试和性能测试两种。实现时，审计者既可以根据审计需要，选择系统中一个实际事务作为样板进行审计跟踪，也可以生成专门的测试用例，通过将测试用事务在系统运行的实际结果与期望结果进行比较来评价系统；还可以设计一个专门仿真系统的程序，让仿真系统与实际系统并行工作，通过比较它们的结果来评测系统。
- 运行结果验证技术。这种技术将注意力放在运行结果——数据上。它主要涉及审计数据选择和收集、数据分析两类问题。常用的审计数据选择和收集的办法有：在应用程序中插入一个审计数据收集模块；设置专门的审计跟踪事务；兼用系统的日志库；使用由随机抽取记录组成的专用审计库，等等。

一旦获得审计数据后，审计者可以检查各类控制信息、完整性约束等内容，以达到各种审计目的。

（4）数据加密

对于那些保密程度极高的数据（如用户标识、绝密信息等）和在网络传输过程中可能被盗窃的数据，除采用上述种种安全保护措施外，一般还需采用数据加密技术，以密文形式保存和传输，保证只有那些知道密钥的用户可以访问。数据加密是防止数据库中的数据在存储和传输中失密的有效手段。有关加密技术的详细内容请参考相关章节，在此不再详细叙述。

（5）视图保护

视图机制可以将要保密的数据对无权存取这些数据的用户隐藏起来，这样就自动地提供了对数据的安全保护。

（6）日志监视

使用日志监视数据库的活动，在日志中记载日期、时间、程序名、地点、存取的数据和存取的类型等。这个日志由 DBA 监督，对于可疑的存取或高度机密数据的存取实施调查。

安全保护的策略还有很多，但要注意，任何安全的措施都是相对的，要根据具体的用户对象、系统对安全的要求和花费的代价来选择恰当的安全措施。

除了解相关技术知识，在写作方面还要注意以下几个方面的要点：

① 在文章中首先用 400~600 字的篇幅简要叙述自己参与开发与设计的软件概要和所担任的工作。

② 具体叙述在开发过程中，是如何使用安全性设计技术的，遇到过哪些实际问题，采取过哪些相应的措施。可就物理安全性或逻辑安全性进行阐述，不要面面俱到。

③ 最后要根据项目实际情况，说明采用的数据库安全性设计技术的效果，有哪些主要的使用体会和进一步的设想。

13.4.8 试题 8 写作要点

本题考查数据挖掘技术的应用。其实从应用的角度，或者从商业的角度来看，数据挖掘这一词在业内出现的频度已不如以前那么高了。因为数据挖掘通常是不独立进行的，它涉及数据源的获取问题，即先要建立一个数据仓库，再从中“挖”数据。这其实就是我们经常看到的是“BI”——商业智能。商业智能我们可以理解为是：数据仓库+数据挖掘。这也就确定了本文的项目背景。

文章最好是把这一层关系讲清楚，写商业智能的项目，如果没有项目经验，直接杜撰出数据挖掘项目来写文章，风险会很高，很容易让人看出文章的“做假”行为。除此以外，文章可按传统的写法组织内容。即按问答方式组织文章的主体脉络，并加入项目信息，同时做好承上启下的句子进行段落衔接。

具体的写作规范如下：

① 结合自己所参与的软件项目，概要介绍该项目的背景及主要内容，并明确指出在其中所承担的主要任务和开展的主要工作。

② 数据挖掘的任务主要是关联分析、聚类分析、分类、预测、时序模式和偏差分析等。

- 关联分析。两个或两个以上变量的取值之间存在某种规律性，就称为关联。数据关联是数据库中存在的一类重要的、可被发现的知识。关联分析的目的在于找出数据库中隐藏的关联网。一般用支持度和可信度两个阈值来度量关联规则的相关性。
- 聚类分析。聚类是把数据按照相似性归纳成若干类别，同一类中的数据彼此相似，不同类中的数据相异。聚类分析可以建立宏观的概念，发现数据的分布模式，以及可能的数据属性之间的相互关系。
- 分类。分类就是找出一个类别的概念描述，它代表了这类数据的整体信息，即该类的内涵描述，并用这种描述来构造模型，一般用规则或决策树模式表示。分类是利用训练数据集通过一定的算法而求得分类规则。分类可被用于规则描述和预测。
- 预测。预测是利用历史数据找出变化规律，建立模型，并由此模型对未来数据的种类及特征进行预测。预测的精度和不确定性被重点关注，通常用预测方差来度量。
- 时序模式。时序模式是指通过时间序列搜索出的重复发生概率较高的模式。

与回归一样，它也是用已知的数据预测未来的值，但这些数据的区别是变量所处时间的不同。

- 偏差分析。在偏差中包括很多有用的知识，数据库中的数据存在很多异常情况，发现数据库中数据存在的异常情况是非常重要的。偏差检验的基本方法就是寻找观察结果与参照之间的差别。

论文中须明确指出自己在该项目应用数据挖掘技术所要解决的具体问题是什么。

③ 主要的数据挖掘方法：

- 神经网络方法：神经网络由于本身良好的鲁棒性、自组织自适应性、并行处理、分布存储和高度容错等特性非常适合解决数据挖掘的问题，因此近年来越来越受到人们的关注。典型的神经网络模型主要分三大类：以感知机、BP 反向传播模型、函数型网络为代表的，用于分类、预测和模式识别的前馈式神经网络模型；以 hopfield 的离散模型和连续模型为代表的，分别用于联想记忆和优化计算的反馈式神经网络模型；以 art 模型、koholon 模型为代表的，用于聚类的自组织映射方法。神经网络方法的缺点是“黑箱”性，人们难以理解网络的学习和决策过程。
- 遗传算法：遗传算法是一种基于生物自然选择与遗传机理的随机搜索算法，是一种仿生全局优化方法。遗传算法具有的隐含并行性、易于和其他模型结合等性质使得它在数据挖掘中被加以应用。
- 决策树方法：决策树是一种常用于预测模型的算法，它通过将大量数据有目的地分类，从中找到一些有价值的、潜在的信息。它的主要优点是描述简单，分类速度快，特别适合大规模的数据处理。最有影响和最早的决策树方法是由 quinlan 提出的著名的基于信息熵的 id3 算法。它的主要问题是：id3 是非递增学习算法；id3 决策树是单变量决策树，复杂概念的表达困难；同性间的相互关系强调不够；抗噪性差。针对上述问题，出现了许多较好的改进算法，如 schlimmer 和 fisher 设计了 id4 递增式学习算法等。
- 粗集方法：粗集理论是一种研究不精确、不确定知识的数学工具。粗集方法有几个优点：不需要给出额外信息；简化输入信息的表达空间；算法简单，易于操作。粗集处理的对象是类似二维关系表的信息表。目前成熟的关系数据库管理系统和新发展起来的数据仓库管理系统，为粗集的数据挖掘奠定了坚实的基础。
- 覆盖正例排斥反例方法：它是利用覆盖所有正例、排斥所有反例的思想来寻找规则。首先在正例集合中任选一个种子，到反例集合中逐个比较。与字段取值构成的选择子相容则舍去，相反则保留。按此思想循环所有正例种子，将得到正例的规则。比较典型的算法有 michalski 的 aq11 方法等。
- 统计分析方法：在数据库字段项之间存在两种关系：函数关系（能用函数公式表示的确定性关系）和相关关系（不能用函数公式表示，但仍是相关确定性关系），对它们的分析可采用统计学方法，即利用统计学原理对数据库中的信息进行分析。可进行常用统计（求大量数据中的最大值、最小值、总和、平均值等）、回归分析（用回归方程来表示变量间的数量关系）、相关分析（用相关系数来度量变量间的相关程度）、差异分析（从样本统计量的值得

出差异来确定总体参数之间是否存在差异)等。

- 模糊集方法:利用模糊集合理论对实际问题进行模糊评判、模糊决策、模糊模式识别和模糊聚类分析。系统的复杂性越高,模糊性越强,一般模糊集合理论是用隶属度来刻画模糊事物的此亦彼性的。

论文中必须明确指出使用了上述7种方法中的哪种或哪几种数据挖掘方法,并给出该方法的具体实现过程;分析所选择的数据挖掘方法的实现效果。

13.4.9 试题9写作要点

关于“企业信息系统的的核心”的论文,在写作方面大致包括以下几个方面:

- 围绕安全方面,根据你所参与分析设计的企业信息系统的核心特点,介绍系统的需求;介绍自己在其中所承担的工作和需要处理的主要问题。
- 在企业信息化建设方面,有许多需要解决的安全问题,它们并非对于每个企业都是必需的。不同企业由于其自身的特点,所关心的安全问题会各有侧重点。由于各个企业信息系统受投资的限制,在保障信息安全方面也不得不有所侧重,或在不同的工程建设阶段,有不同的建设目标。因此这部分的说明重点在于针对不同企业的特点、应用领域和投资,给出相应的合理论述。
- 在论述自己的安全解决方案之后,要根据系统的具体情况和采用的措施,客观地评价自己所采用的安全措施,并给出评价依据。这里需要注意,不可脱离实际,评价要适当。

13.4.10 试题10写作要点

① 说明软件开发项目的基本情况以及自己承担的主要工作。

② 当前的软件可靠性模型众多,但并没有一个最好的或者可以适用所用软件系统的软件可靠性模型,因此对于不同的软件系统,出于不同的可靠性分析目的,需要选择合适的软件可靠性模型。

常见的10类软件可靠性模型有种子法模型、失效率类模型、曲线拟合类模型、可靠性增长模型、程序结构分析模型、输入域分类模型、执行路径分析方法模型、非齐次泊松过程模型、马尔可夫过程模型和贝叶斯分析模型。

软件可靠性模型的选择主要需要考虑以下4个方面:

- 模型假设的适用性:模型假设是可靠性模型的基础,模型假设需要符合软件系统的现有状况,在软件系统中与假设冲突的因素达到几乎不存在的程度。往往一个模型的假设有很多,需要在选择模型时对每一条假设进行分析,评估现有软件系统中不符合假设的因素对可靠性评价有多大影响,以确定模型是否符合软件系统的可靠性评价工作。
- 模型预测的能力与质量:预测的能力和质量是指模型根据现在和历史的可靠性数据,预测将来的可靠性和失效概率的能力,以及预测结果的准确程度。因此,应尽可能选择比较成熟的、应用较广的模型。
- 模型输出值能否满足可靠性评价需求:根据可靠性测试目的来确定哪些模型

的输出值满足可靠性评价需求。重要的可靠性定量指标包括当前可靠度、平均无失效时间、故障密度、期望达到规定可靠性目标的日期、达到规定可靠性目标的成本要求等。

- 模型使用的简便性：模型使用的数据在软件系统中易于收集；模型应该简单、易懂；模型应该便于使用，最好有工具支持。

三、软件可靠性数据的收集是一项艰巨而又烦琐的工作，受到许多潜在因素的影响和制约。常见的问题如下：

- 可靠性数据规范不一致，对软件进行度量的定义混乱。
- 数据收集过程存在于整个软件生命周期，但由于成本等因素，其连续性往往不能保证。
- 缺乏有效的技术和工具支持，难以进行自动分析。
- 数据完整性不能保证，收集到的数据大多数是不完全的。
- 数据质量和准确性不能保证。
- 缺乏可靠性数据的交流与共享。

考生应叙述在项目中遇到了上述中的哪些问题。

可供采用的解决方法主要有：

- 尽早确定可靠性模型，明确需要搜集的可靠性数据，确定涉及的术语、记录方法等。
- 制订可实施的可靠性数据搜集计划，并指定专人负责。保证数据的收集和验证与软件开发过程同步进行。
- 重视软件测试特别是可靠性测试产生的测试结果的整理和分析。
- 尽可能地利用工具进行收集工作，如利用数据库进行存储和分析等。