



全国计算机技术与软件专业技术资格（水平）考试辅导用书

软件设计师考试 历年试题分析与解答

计算机技术与软件专业技术资格（水平）考试研究组 编



西安电子科技大学出版社
<http://www.xduph.com>

全国计算机技术与软件专业技术资格(水平)考试辅导用书

软件设计师考试历年试题分析与解答

计算机技术与软件专业技术资格(水平)考试研究组 编

西安电子科技大学出版社

2006

内 容 简 介

本书收集了全国计算机技术与软件专业技术资格(水平)考试 2002~2006 年软件设计师级(原高级程序员级)的全部试题,并给出了详尽的分析与解答。本书有助于准备参加计算机技术与软件专业技术资格(水平)考试的应试者复习有关内容,了解试题形式,提高应试能力。相信本书对于准备参加其他类似考试的读者及需要复习有关计算机及其应用知识的读者都会有所帮助。

本书既可作为考生备考的参考书及实战训练资料,也可供大专院校的师生和计算机爱好者学习参考。

图书在版编目(CIP)数据

软件设计师考试历年试题分析与解答/计算机技术与软件专业技术资格(水平)考试研究组编.

—西安:西安电子科技大学出版社,2005.9

全国计算机技术与软件专业技术资格(水平)考试辅导用书

ISBN 7-5606-1572-4

I. 软… II. 计… III. 软件设计—工程技术人员—资格考核—解题 IV. TP311.5-44

中国版本图书馆 CIP 数据核字(2005)第 109953 号

策 划 臧延新

责任编辑 臧延新

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

<http://www.xduph.com> E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2005年9月第1版 2006年9月第3次印刷

开 本 787毫米×1092毫米 1/16 印张 19.75

字 数 472千字

印 数 10 001~12 000册

定 价 30.00元

ISBN 7-5606-1572-4/TP·0899

XDUP 1863021-3

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

计算机技术与软件专业技术资格(水平)考试自 1985 年开考以来,经历了由多省市共同举行联合考试等过程,目前已发展成由国家人事部、信息产业部等部门领导下的国家级考试。由于该考试的结果与获得专业技术职称相联系,使得考试更科学和合理,也更具有权威性和吸引力。这对提高我国计算机软件人员的技术水平,鼓励和激发计算机专业工作人员、在校学生钻研业务,提高全民的计算机应用水平都有很大的促进作用。

本书收集了 2002~2006 年计算机技术与软件专业技术资格(水平)考试软件设计师级的全部试题,并给出了详尽的分析和解答,目的在于帮助读者复习有关的知识,了解试题形式,提高应试能力。相信本书对于准备参加其他类似考试的读者或者需要复习有关计算机及其应用知识的读者都是有帮助的。本书有如下特点:

(1) 收集了这段时期的全部试题,而不是选编。

(2) 对每道试题都先给出详尽的解题思路及方法,然后再给出正确的解答。

全书按考试年限分章,每章在编排结构上分为上午试题、上午试题解析、下午试题和下午试题解析。

考虑试题结构的完整性,本书对过去的 CASL 汇编语言(此类型考题目前大纲中不要求)考题也进行了分析,并给出了答案。这可供读者学习 CASL 汇编语言程序设计时参考。

作者在本书的编写过程中,参考了许多相关的书籍和资料,在此对这些参考文献的作者表示感谢。同时也非常感谢西安电子科技大学出版社在本书出版过程中所给予的大力支持和帮助。

尽管参加本书编写的人员都是多年从事计算机专业领域科研和教学工作的专业技术人员和学者,但是书中难免会存在一些错漏和不妥之处,望读者批评指正。

编 者

2006 年 7 月

目 录

第 1 章 2002 年试题解析	1
1.1 2002 年上午试题	1
1.2 2002 年上午试题解析	9
1.3 2002 年下午试题	28
1.4 2002 年下午试题解析	38
第 2 章 2003 年试题解析	47
2.1 2003 年上午试题	47
2.2 2003 年上午试题解析	55
2.3 2003 年下午试题	71
2.4 2003 年下午试题解析	80
第 3 章 2004 年上半年试题解析	91
3.1 2004 年上半年上午试题	91
3.2 2004 年上半年上午试题解析	100
3.3 2004 年上半年下午试题	121
3.4 2004 年上半年下午试题解析	130
第 4 章 2004 年下半年试题解析	139
4.1 2004 年下半年上午试题	139
4.2 2004 年下半年上午试题解析	146
4.3 2004 年下半年下午试题	160
4.4 2004 年下半年下午试题解析	172
第 5 章 2005 年上半年试题解析	183
5.1 2005 年上半年上午试题	183
5.2 2005 年上半年上午试题解析	191
5.3 2005 年上半年下午试题	206
5.4 2005 年上半年下午试题解析	218

第 6 章 2005 年下半年试题解析	226
6.1 2005 年下半年上午试题	226
6.2 2005 年下半年上午试题解析	235
6.3 2005 年下半年下午试题	247
6.4 2005 年下半年下午试题解析	258
第 7 章 2006 年上半年试题解析	264
7.1 2006 年上半年上午试题	264
7.2 2006 年上半年上午试题解析	274
7.3 2006 年上半年下午试题	288
7.4 2006 年上半年下午试题解析	301

第 1 章 2002 年试题解析

1.1 2002 年上午试题

● 常规的数据加密标准 DES 采用 (1) 位有效密钥对 (2) 位的数据块进行加密。

- (1) A. 56 B. 64 C. 112 D. 128
 (2) A. 32 B. 64 C. 128 D. 256

● 在面向对象技术中, 多态有多种不同的形式, 其中 (3) 和 (4) 称为通用多态, (5) 和强制多态称为特定多态。

- (3) A. 参数多态 B. 过载多态 C. 隐含多态 D. 重置多态
 (4) A. 重置多态 B. 过载多态 C. 隐含多态 D. 包含多态
 (5) A. 参数多态 B. 隐含多态 C. 过载多态 D. 包含多态

● 已知一不确定的有穷自动机(NFA)如图 1.1 所示, 采用子集法将其确定化为 DFA 的过程如表 1.1 所示。

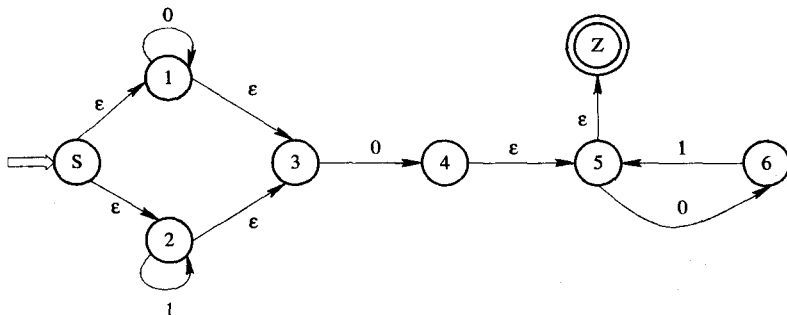


图 1.1

表 1.1

I	I_0	I_1
{S, 1, 2, 3}	{1, 3, 4, 5, Z}	{2, 3}
{1, 3, 4, 5, Z}	T1	T3
{2, 3}	{4, 5, Z}	{2, 3}
T2	{6}	T3
T1	{1, 3, 4, 5, 6, Z}	{5, Z}
{6}	T3	{5, Z}
{5, Z}	{6}	T3

状态集 T_1 中不包括编号为 (6) 的状态; 状态集 T_2 中的成员有 (7); 状态集 T_3 等于 (8); 该自动机所识别的语言可以用正规式 (9) 表示。

(6) A. 2 B. 4 C. 3 D. 5

(7) A. 1, 3, 4, 5, Z B. 2, 3

C. 6 D. 4, 5, Z

(8) A. $\{Z\}$ B. $\{6\}$ C. $\{4, 5, Z\}$ D. $\{\}$

(9) A. $(0|1)^*$ B. $(0^*|1^*)^*001$

C. $(0^*|1^*)^*0(0|1)^*$ D. $(1^*|0^*)0(0|1)^*$

● 与正规式 $(a|b)^*$ 等价的正规式为 (10)。

(10) A. $a^*|b^*$ B. a^*b^* C. $(a^*b^*)^*$ D. $(ab)^*$

● 算法是对问题求解过程的一类精确描述, 算法中描述的操作都是可以通过已经实现的基本操作在限定时间内执行有限次来实现的。这句话说明算法具有 (11) 特性。

(11) A. 正确性 B. 确定性 C. 能行性 D. 健壮性

● 快速排序算法采用的设计方法是 (12)。

(12) A. 动态规划法(Dynamic Programming)

B. 分治法(Divide and Conquer)

C. 回溯法(Backtracking)

D. 分支定界法(Branch and Bound)

● 在数据压缩编码的应用中, 哈夫曼(Huffman)算法可以用来构造具有 (13) 的二叉树, 这是一种采用了 (14) 策略的算法。

(13) A. 前缀码 B. 最优前缀码 C. 后缀码 D. 最优后缀码

(14) A. 贪心 B. 分治 C. 递推 D. 回溯

● 用递归算法实现 n 个相异元素构成的有序序列的二分查找, 采用一个递归工作栈时, 该栈的最小容量应为 (15)。

(15) A. n B. $\lceil n/2 \rceil$ C. $\lceil \log_2 n \rceil$ D. $\lceil \log_2(n+1) \rceil$

● 设有 7 项任务, 分别标记为 a、b、c、d、e、f 和 g, 需要若干台机器以并行工作方式来完成, 它们执行的开始时间和完成时间如表 1.2 所示。

表 1.2

任务 时间	a	b	c	d	e	f	g
开始时间	0	3	4	9	7	1	6
结束时间	2	7	7	11	10	5	8

在最优分配方案中完成这些任务需要 (16) 台机器。

(16) A. 2 B. 3 C. 4 D. 5

● 下述函数中渐进时间最小的是 (17)。

(17) A. $T_1(n) = n \log_2 n + 100 \log_2 n$ B. $T_2(n) = 2n \log_2 n - 100 \log_2 n$

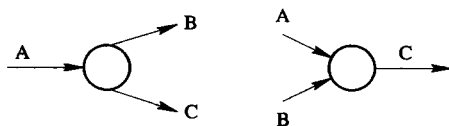
C. $T_3(n) = n^2 - 100 \log_2 n$ D. $T_4(n) = 4n \log_2 n - 100 \log_2 n$

● 对于给定的一组关键字(12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18), 按照下列算法进

行递增排序, 写出每种算法第一趟排序后得到的结果: 希尔排序(增量为5)得到 (18), 快速排序(选第一个记录为基准元素)得到 (19), 链式基数(基数为10)排序得到 (20), 二路归并排序得到 (21), 堆排序得到 (22)。

- (18) A. 2, 4, 6, 8, 10, 12, 16, 18, 20, 28, 30
 B. 6, 2, 10, 4, 8, 12, 28, 30, 20, 16, 18
 C. 12, 2, 10, 20, 6, 18, 4, 16, 30, 8, 28
 D. 30, 10, 20, 12, 2, 4, 16, 6, 8, 28, 18
- (19) A. 10, 6, 18, 8, 4, 2, 12, 20, 16, 30, 28
 B. 6, 2, 10, 4, 8, 12, 28, 30, 20, 16, 18
 C. 2, 4, 6, 8, 10, 12, 16, 18, 20, 28, 30
 D. 6, 10, 8, 28, 20, 18, 2, 4, 12, 30, 16
- (20) A. 10, 6, 18, 8, 4, 2, 12, 20, 16, 30, 28
 B. 12, 2, 10, 20, 6, 18, 4, 16, 30, 8, 28
 C. 2, 4, 6, 8, 10, 12, 16, 18, 20, 28, 30
 D. 30, 10, 20, 12, 2, 4, 16, 6, 8, 28, 18
- (21) A. 2, 12, 16, 8, 28, 30, 4, 6, 10, 18, 20
 B. 2, 12, 16, 30, 8, 28, 4, 10, 6, 20, 18
 C. 12, 2, 16, 8, 28, 30, 4, 6, 10, 28, 18
 D. 12, 2, 10, 20, 6, 18, 4, 16, 30, 8, 28
- (22) A. 30, 28, 20, 12, 18, 16, 4, 10, 2, 6, 8
 B. 20, 30, 28, 12, 18, 4, 16, 10, 2, 8, 6
 C. 2, 6, 4, 10, 8, 28, 16, 30, 20, 12, 18
 D. 2, 4, 10, 6, 12, 28, 16, 20, 8, 30, 18

● 在表示多个数据流与加工之间关系的符号中, 下列符号分别表示 (23) 和 (24)。



- (23) A. 若 A, 则((B 或 C)或(B 与 C))
 B. 若 A 或 B, 则 C
 C. 若 A 与 B, 则 C
 D. 若 A, 则(B 或 C)但非(B 与 C)
- (24) A. 若 A, 则((B 或 C)或(B 与 C))
 B. 若 A 或 B, 则 C
 C. 若 A 与 B, 则 C
 D. 若 A, 则(B 或 C)但非(B 与 C)

● 概要设计是软件系统结构的总体设计, 以下选项中不属于概要设计的是 (25)。

- (25) A. 把软件划分成模块
 B. 确定模块之间的调用关系
 C. 确定各个模块的功能
 D. 设计每个模块的伪代码

● 如果一个软件是给许多客户使用的, 大多数软件厂商要使用几种测试过程来发现那些可能只有最终用户才能发现的错误。(26) 测试是由软件的最终用户在一个或多个用户实际使用环境下来进行的。(27) 测试是由一个用户在开发者的场所来进行的。测试的目的是寻找错误的原因并改正之。

(26) A. alpha B. beta C. gamma D. delta

(27) A. alpha B. beta C. gamma D. delta

● 在下列说法中, (28) 是造成软件危机的主要原因。

① 用户使用不当 ② 软件本身特点 ③ 硬件不可靠

④ 对软件的错误认识 ⑤ 缺乏好的开发方法和手段 ⑥ 开发效率低

(28) A. ①③⑥ B. ①②④ C. ③⑤⑥ D. ②⑤⑥

● 原型化(Prototyping)方法是一类动态定义需求的方法, (29) 不是原型化方法所具有的特征。与结构化方法相比, 原型化方法更需要 (30)。衡量原型开发人员能力的重要标准是 (31)。

(29) A. 提供严格定义的文档

B. 加快需求的确定

C. 简化项目管理

D. 加强用户参与和决策

(30) A. 熟练的开发人员

B. 完整的生命周期

C. 较长的开发时间

D. 明确的需求定义

(31) A. 丰富的编程技巧

B. 灵活使用开发工具

C. 很强的协调组织能力

D. 快速获取需求

● 软件能力成熟度模型 CMM(Capability Maturity Model)描述和分析了软件过程能力的发展与改进的程度, 确立了一个软件过程成熟程度的分级标准。在初始级, 软件过程定义几乎处于无章法可循的状态, 软件产品的成功往往依赖于个人的努力和机遇。在 (32), 已建立了基本的项目管理过程, 可对成本、进度和功能特性进行跟踪。在 (33), 用于软件管理与工程两方面的软件过程均已文档化、标准化, 并形成了整个软件组织的标准软件过程。在已管理级, 对软件过程和产品质量有详细的度量标准。在 (34), 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、持续地对过程进行改进。

(32) A. 可重复级 B. 管理级 C. 功能级 D. 成本级

(33) A. 标准级 B. 已定义级 C. 可重复级 D. 优化级

(34) A. 分析级 B. 过程级 C. 优化级 D. 管理级

● 设学生 S、课程 C、学生选课 SC 的关系模式分别为 S(Sno, Sname, Sdep, Sage, Saddr)、C(Cno, Cname, Pcno)以及 SC(Sno, Cno, Grade)。与关系代数表达式 $\pi_{Sno, Sname, Grade}(\sigma_{Cname='数据库'}(S \bowtie SC \bowtie C))$ 等价的元组演算表达式为:

$\{t \mid \underline{(35)} S(u) \wedge SC(v) \wedge C(w) \wedge \underline{(36)} \wedge \underline{(37)}\}$

(35) A. $(\forall u)(\exists v)(\exists w)$ B. $(\exists u)(\forall v)(\exists w)$

C. $(\exists u)(\exists v)(\exists w)$ D. $(\exists u)(\exists v)(\forall w)$

(36) A. $u[1] = v[1] \wedge v[1] = w[1] \wedge w[1] = '数据库'$

B. $u[1] = v[2] \wedge v[2] = w[1] \wedge w[3] = '数据库'$

C. $u[1] = v[1] \wedge v[2] = w[1] \wedge w[2] = '数据库'$

D. $u[2] = v[2] \wedge v[1] = w[2] \wedge v[2] = '数据库'$

(37) A. $t[1] = u[1] \wedge t[2] = u[2] \wedge t[3] = v[3]$

B. $t[1] = u[1] \wedge t[2] = u[2] \wedge t[3] = v[2]$

C. $t[1] = u[1] \wedge t[2] = w[1] \wedge t[3] = v[2]$

D. $t[1] = u[1] \wedge t[2] = w[2] \wedge t[3] = v[3]$

● 设关系模式 R 为 $R(H, I, J, K, L)$, R 上的一个函数依赖集为 $F = \{ H \rightarrow J, J \rightarrow K, I \rightarrow J, KL \rightarrow J, JL \rightarrow H \}$, 分解 (38) 是无损联接的。关系模式 $R(A, B, C, D, E)$ 中的关系代数表达式 $\sigma_{5 < '2'}(R)$ 等价于 SQL 语句 (39)。

(38) A. $\rho = \{HK, HI, IJ, JKL, HL\}$ B. $\rho = \{HIL, IKL, IJL\}$

C. $\rho = \{HJ, IK, HL\}$ D. $\rho = \{HI, JK, HL\}$

(39) A. SELECT * FROM R WHERE $E < '2'$

B. SELECT B, E FROM R WHERE $E < '2'$

C. SELECT B, E FROM R HAVING $E < '2'$

D. SELECT * FROM R WHERE $'5' < B$

● 关系 R 、 S 分别如表 1.3、1.4 所示, $R \div (\pi_{A1, A2}(\sigma_{1 < 3}(S)))$ 的结果为 (40), R 与 S 的左外联接、右外联接和完全外联接的元组个数分别为 (41)。

表 1.3

A1	A2	A3
a	b	c
b	a	d
c	d	d
d	f	g

R 关系

表 1.4

A1	A2	A4
a	z	a
b	a	h
c	d	d
d	s	c

S 关系

(40) A. $\{d\}$

B. $\{c, d\}$

C. $\{c, d, g\}$

D. $\{(a, b), (b, a), (c, d), (d, f)\}$

(41) A. 2, 2, 4

B. 2, 2, 6

C. 4, 4, 6

D. 4, 4, 4

● 一台 PC 计算机系统启动时, 首先执行的是 (42), 然后加载 (43)。在设备管理中, 虚拟设备的引入和实现是为了充分利用设备, 提高系统效率, 采用 (44) 来模拟低速设备(输入机或打印机)的工作。

已知 A 、 B 的值, 表达式 $A^2 / (5A + B)$ 的求值过程, 且 A 、 B 已赋值, 该公式求值过程可用前驱图 (45) 来表示, 若用 PV 操作控制求值过程, 需要 (46) 的信号量。

(42) A. 主引导记录

B. 分区引导记录

C. BIOS 引导程序

D. 引导扇区

(43) A. 主引导记录和引导驱动器的分区表, 并执行主引导记录

B. 分区引导记录、配置系统, 并执行分区引导记录

C. 操作系统, 如 Windows 98/NT/2000/XP、UNIX 等

D. 相关支撑软件

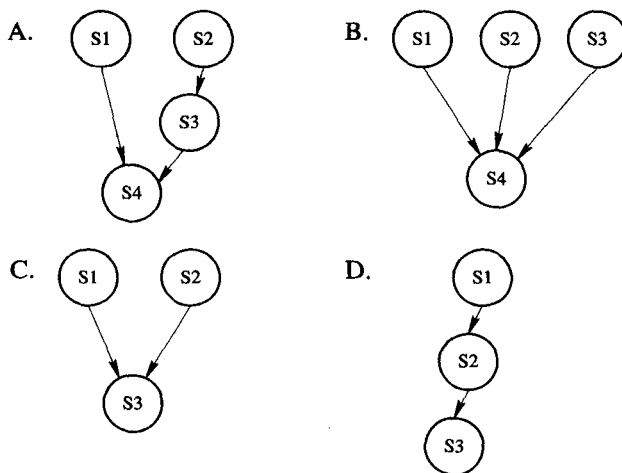
(44) A. Spooling 技术, 利用磁带设备

B. Spooling 技术, 利用磁盘设备

C. 脱机批处理系统

D. 移臂调度和旋转调度技术, 利用磁盘设备

(45)



(46) A. 3 个且初值等于 1

B. 2 个且初值等于 0

C. 2 个且初值等于 1

D. 3 个且初值等于 0

● 多媒体应用需要对庞大的数据进行压缩, 常见的压缩编码方法可分为两大类, 一类是无损压缩法, 另一类是有损压缩法, 也称 (47)。 (48) 属于无损压缩法。

(47) A. 熵编码

B. 熵压缩法

C. MPEG 压缩法

D. JPEG 压缩法

(48) A. MPEG 压缩

B. 子带编码

C. Huffman 编码

D. 模型编码

● 若每个像素具有 8 位的颜色深度, 则可表示 (49) 种不同的颜色, 若某个图像具有 640×480 个像素点, 其未压缩的原始数据需占用 (50) 字节的存储空间。

(49) A. 8

B. 128

C. 256

D. 512

(50) A. 1024

B. 19 200

C. 38 400

D. 307 200

● 声音的三要素为音调、音强和音色, 其中音色是由混入基音的 (51) 决定的。若对声音以 22.05 kHz 的采样频率、8 位采样深度进行采样, 则 10 分钟双声道立体声的存储量为 (52) 字节。

(51) A. 响度

B. 泛音

C. 高音

D. 波形声音

(52) A. 26 460 000

B. 441 000

C. 216 000 000

D. 108 000 000

● 一般来说, Cache 的功能 (53)。某 32 位计算机的 Cache 容量为 16 KB, Cache 块的大小为 16 B, 若主存与 Cache 的地址映射采用直接映射方式, 则主存地址为 1234E8F8(十六进制)的单元装入的 Cache 地址为 (54)。在下列 Cache 替换算法中, 平均命中率最高的是 (55)。

(53) A. 全部由软件实现

B. 全部由硬件实现

C. 由硬件和软件相结合实现

D. 有的计算机由硬件实现, 有的计算机由软件实现

(54) A. 00 0100 0100 1101 (二进制)

B. 01 0010 0011 0100 (二进制)

C. 10 1000 1111 1000 (二进制)

D. 11 0100 1110 1000 (二进制)

(55) A. 先入后出(FILO)算法

B. 随机替换(RAND)算法

C. 先入先出(FIFO)算法

D. 近期最少使用(LRU)算法

● 设某流水线计算机主存的读/写时间为 100 ns, 有一个指令和数据合一的 Cache, 已知该 Cache 的读/写时间为 10 ns, 取指令的命中率为 98%, 取数的命中率为 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 (56)。

(56) A. 12 ns

B. 15 ns

C. 18 ns

D. 120 ns

● 相联存储器的访问方式是 (57)。

(57) A. 先入先出访问

B. 按地址访问

C. 按内容访问

D. 先入后出访问

● 假设一个有 3 个盘片的硬盘, 共有 4 个记录面, 转速为 7200 r/min, 盘面有效记录区域的外直径为 30 cm, 内直径为 10 cm, 记录位密度为 250 位/mm, 磁道密度为 8 道/mm, 每磁道分 16 个扇区, 每扇区 512 字节, 则该硬盘的非格式化容量和格式化容量约为 (58), 数据传输率约为 (59)。若一个文件超出一个磁道容量, 剩下的部分 (60)。

(58) A. 120 MB 和 100 MB

B. 30 MB 和 25 MB

C. 60 MB 和 50 MB

D. 22.5 MB 和 25 MB

(59) A. 2356 KB/s

B. 3534 KB/s

C. 7069 KB/s

D. 1178 KB/s

(60) A. 存于同一盘面的其他编号的磁道上

B. 存于其他盘面的同一编号的磁道上

C. 存于其他盘面的其他编号的磁道上

D. 存放位置随机

● 数字签名是一种网络安全技术, 利用这种技术, 接收者可以确定发送者的身份是否真实, 同时发送者不能 (61) 发送的消息, 接收者也不能 (62) 接收的消息。Kerberos 是一种分布式环境下的 (63) 系统。为了防止重放攻击(Replay), 它使用了一次性的 (64) 和时间戳。在公钥加密的情况下, 用户必须警惕用于加密的公钥是否属于真正的接收者, 为此必须使用数字证书; 常用的数字证书格式有 (65) 证书和 X.509 证书。

(61) A. 泄露

B. 隐藏

C. 篡改

D. 否认

(62) A. 泄露

B. 否认

C. 篡改

D. 隐藏

(63) A. 数字签名

B. 身份认证

C. 数字证书

D. 公钥加密

(64) A. Key

B. Certs

C. MAC

D. Ticket

(65) A. PGP

B. SSL

C. SHTTP

D. SOCKS

● A typical (66) language contains an applicative sub-language which approximates the mathematical abstractions of “timeless” functions applied to “spaceless” values, where the actual operation sequences and use of storage space during expression evaluation

are organized behind the (67) . In this setting, values are data structures of low volume, typically a few computer words or less, which means that an illusion of spacelessness can be realized by having (68) results during expression evaluation stored at the discretion of the language implementation, and effecting parameter (69) and (70) operations through value copying.

- (66) A. imperative B. mandatory C. compulsory D. voluntary
 (67) A. foreground B. background C. screen D. scenes
 (68) A. middle B. intermediate C. previous D. final
 (69) A. tranverse B. transportation C. transmmision D. translation
 (70) A. assignment B. design C. value D. dispatch

● Most computer systems are (71) to two different groups of attacks: Insider attacks and outsider attacks. A system that is known to be (72) to an outsider attack by preventing (73) from outside can still be vulnerable to the insider attacks accomplished by abusive usage of (74) users. Detecting such abusive usage as well as attacks by outsides not only provides information on damage assessment, but also helps to prevent future attacks. These attacks are usually (75) by tools referred to as Intrusion Detection Systems.

- (71) A. vulnerable B. week C. sensitively D. helpless
 (72) A. reliable B. secure C. indestructible D. steady
 (73) A. visit B. access C. I/O D. read/write
 (74) A. power B. rights C. authorized D. common
 (75) A. searched B. checked C. tested D. detected

为方便读者, 将 2002 年上午试题的答案用表 1.5 给出。

表 1.5 2002 年上午试题答案

(1)	A	(16)	B	(31)	D	(46)	D	(61)	D
(2)	B	(17)	A 或 B	(32)	A	(47)	B	(62)	C
(3)	A	(18)	C	(33)	B	(48)	C	(63)	B
(4)	D	(19)	B	(34)	C	(49)	C	(64)	D
(5)	C	(20)	D	(35)	C	(50)	D	(65)	A
(6)	A	(21)	B	(36)	C	(51)	B	(66)	A
(7)	D	(22)	C	(37)	A	(52)	A	(67)	D
(8)	D	(23)	A	(38)	B	(53)	B	(68)	B
(9)	D	(24)	C	(39)	A	(54)	C	(69)	C
(10)	C	(25)	D	(40)	A	(55)	D	(70)	A
(11)	C	(26)	B	(41)	C	(56)	B	(71)	A
(12)	B	(27)	A	(42)	C	(57)	C	(72)	B
(13)	B	(28)	D	(43)	A	(58)	B	(73)	B
(14)	A	(29)	A	(44)	B	(59)	D	(74)	C
(15)	D	(30)	B	(45)	A	(60)	B	(75)	D

1.2 2002 年上午试题解析

试题(1)、(2)分析

本题考查的是数据加密标准 DES 的基本概念。

试题(1)正确答案为 A。美国数据加密标准 DES 是一种对称加密的算法,“对称”是指采用的保密密钥既可用于加密,也可用于解密,DES 的算法是公开的,密钥由用户自己保护。

试题(2)正确答案为 B。密钥长度为 64 bit,其中有 8 bit 奇偶校验,有效密钥长度为 56 bit,即采用一个 56 位有效密钥对 64 位的数据块进行加密。

试题答案: (1) A (2) B

试题(3)~(5)分析

本题考查的是面向对象技术方面的基本概念。

试题(3)正确答案为 A。因为,在面向对象技术中,对象在收到信息后要予以响应,不同的对象收到同一消息可产生完全不同的结果,这一现象称为多态。在使用多态技术时,用户可以发送一个通用的消息,而实现的细节则由接收对象自行决定,这样同一消息就可以调用不同的方法。

试题(4)正确答案为 D。因为多态有多种不同的形式,其中参数多态和包含多态称为通用多态,过载多态和强制多态称为特定多态。

参数多态是应用比较广泛的,被称为最纯的多态。这是因为同一对象、函数或过程能以一致的形式用于不同的类型。包含多态最常见的例子就是子类型化,即一个类型是另一类型的子类型。

过载多态是同一变量被用来表示不同的功能而通过上下文以决定一个名所代表的功能。这纯粹是一种通过语法对不同语义的对象使用相同的名,编译能够消除这一模糊。强制多态是通过语义操作把一个变元的类型加以变换,以符合一个函数的要求,如果不做这一强制性变换将出现类型错误。类型的变换可在编译时完成,通常是隐式进行,当然也可以在动态运行时来做。

试题答案: (3) A (4) D (5) C

试题(6)~(9)分析

本题考查的是程序设计方面的基础知识。

若已知非确定的有穷自动机 $N(NFA\ N) = \{S, \Sigma, move, s_0, F\}$, 则可以构造出相应的确定有穷自动机 $M(DFA\ M) = \{S', \Sigma, move', q_0, F'\}$ 。其中 S (或 S')是一个状态集合; Σ 是一个有穷字母表; $move$ (或 $move'$)是状态转移函数,例如“ $move(s_1, a) = s_2$ ”表示当系统处于状态 s_1 且输入字母为“a”时,系统转换到状态 s_2 ; s_0 (或 q_0)是起始状态; F (或 F')为接受状态集合。在介绍构造方法之前,我们先给出一个定义。若 I 是 $NFA\ N$ 的状态集的一个子集,我们定义 $\epsilon_CLOSURE(I)$ 如下:

- (1) 状态集 I 的 $\epsilon_CLOSURE(I)$ 是一个状态集;
- (2) 状态集 I 的所有状态属于 $\epsilon_CLOSURE(I)$;
- (3) 若 $s \in I$, 那么从 s 出发经过任意条 ϵ 弧到达的状态 s' 都属于 $\epsilon_CLOSURE(I)$ 。

状态集 $\epsilon_CLOSURE(I)$ 称为 I 的 ϵ -闭包。

由以上的定义可知, I 的 ϵ -闭包就是从状态集 I 的状态出发, 经 ϵ 所能到达的状态的全体。假定 I 是 N 的状态集的一个子集, a 是 Σ 中的一个字符, 我们定义:

$$I_a = \epsilon_CLOSURE(J)$$

上式中, J 是所有那些可以从 I 中的某一状态结点出发经过一条 a 弧而到达的状态结点的全体。现在可以用子集法将一个 NFA N 确定化为一个 DFA, 也就是求解出 S' 、 $move'$ 、 q_0 和 F' 。

用子集法将非确定的有穷自动机 $N(NFA\ N)$ 确定化的算法如下:

第一步: 求出 M 的初态 q_0 , 即令 $q_0 = \epsilon_CLOSURE(\{s_0\})$, 此时 S' 仅含初态 q_0 , 并且没有标记;

第二步: 对于 S' 中尚未标记的状态 $q_i = \{s_{i1}, s_{i2}, \dots, s_{im}\}$, $s_{ij} \in S (j=1, \dots, m)$:

(1) 标记 q_i ;

(2) 对于每个 $a \in \Sigma$, 令

$$T = move(\{s_{i1}, s_{i2}, \dots, s_{im}\}, a), q_j = \epsilon_CLOSURE(T)$$

(3) 若 q_j 尚不在 S' 中, 则将 q_j 作为一个未加标记的新状态添加到 S' , 并把状态转移 $move'(q_i, a) = q_j$ 添加到 M 中;

第三步: 重复进行第二步, 直到 S' 中不再有未标记的状态时为止。

第四步: 令 $F' = \{q | q \in S' \text{ 且 } q \cap F \neq \emptyset\}$ 。

从题目给出的 NFA 的状态转换图可以看出, 状态结点 S 就是自动机的初始状态, 令 S' 的初值为空集合。根据 ϵ -闭包的定义, 可以求出 $\epsilon_CLOSURE(\{S_0\}) = \{S, 1, 2, 3\}$, 将 $\{S, 1, 2, 3\}$ 记为 q_0 , 并将 q_0 作为未加标记的状态加入 S' , 因此 $S' = \{q_0\}$; 其后求解 DFA M 的各个状态的过程如下:

根据题中所给的状态转换图以及算法中的第二步, 进行如下计算:

(1) 将 q_0 作上标记, 以“*”作为标志的记号, 即 $S' = \{q_0^*\}$;

(2) 求出 q_0 对于字母表中 0 和 1 的状态转移。

因为 $move(1, 0) = 1$, $move(3, 0) = 4$, 状态 S 及状态 2 中不存在关于字母“0”的转移函数, 所以 $move(q_0, 0) = \{1, 4\}$, 那么根据 ϵ -闭包的定义可求得 $\epsilon_CLOSURE(\{1, 4\}) = \{1, 3, 4, 5, Z\}$, 我们将 $\{1, 3, 4, 5, Z\}$ 记为 q_1 ;

同时由于 $move(2, 1) = 2$, 状态 S 、状态 1 及状态 3 中不存在关于字母“1”的转移函数, 所以 $move(q_0, 1) = \{2\}$, 因此可求得 $\epsilon_CLOSURE(\{2\}) = \{2, 3\}$, 我们将 $\{2, 3\}$ 记为 q_2 ;

(3) 将计算所得到的 q_1 和 q_2 作为未加标记的新状态添加到 S' , 即 $S' = \{q_0^*, q_1, q_2\}$;

依次类推, 重复进行以上算法的第(2)步所得到的中间结果如下:

$$\epsilon_CLOSURE(move(q_1, 0)) =$$

$$\epsilon_CLOSURE(\{1, 4, 6\}) = \{1, 3, 4, 5, 6, Z\}, \text{ 将 } \{1, 3, 4, 5, 6, Z\} \text{ 标记为 } q_3,$$

$$\epsilon_CLOSURE(move(q_1, 1)) = \epsilon_CLOSURE(\emptyset) = \emptyset, \{\emptyset\} \text{ 无须标记,}$$

$$S' = \{q_0^*, q_1^*, q_2, q_3\};$$

$$\epsilon_CLOSURE(move(q_2, 0)) = \epsilon_CLOSURE(\{4\}) = \{4, 5, Z\}, \text{ 将 } \{4, 5, Z\} \text{ 记为 } q_4,$$

$$\epsilon_CLOSURE(move(q_2, 1)) = \epsilon_CLOSURE(\{2\}) = \{2, 3\} = q_2,$$

$$S' = \{q_0^*, q_1^*, q_2^*, q_3, q_4\};$$

$\epsilon_CLOSURE(move(q_3, 0)) = \epsilon_CLOSURE(\{1, 4, 6\}) = \{1, 3, 4, 5, 6, Z\} = q_2$,
 $\epsilon_CLOSURE(move(q_3, 1)) = \epsilon_CLOSURE(\{5\}) = \{5, Z\}$, 将 $\{5, Z\}$ 记为 q_5 ,
 $S' = \{q_0^*, q_1^*, q_2^*, q_3^*, q_4, q_5\}$;
 $\epsilon_CLOSURE(move(q_4, 0)) = \epsilon_CLOSURE(\{6\}) = \{6\}$, 将 $\{6\}$ 记为 q_6 ,
 $\epsilon_CLOSURE(move(q_4, 1)) = \epsilon_CLOSURE(\{\}) = \{\}$,
 $S' = \{q_0^*, q_1^*, q_2^*, q_3^*, q_4^*, q_5, q_6\}$;
 $\epsilon_CLOSURE(move(q_5, 0)) = \epsilon_CLOSURE(\{6\}) = \{6\} = q_6$,
 $\epsilon_CLOSURE(move(q_5, 1)) = \epsilon_CLOSURE(\{\}) = \{\}$,
 $S' = \{q_0^*, q_1^*, q_2^*, q_3^*, q_4^*, q_5^*, q_6\}$;
 $\epsilon_CLOSURE(move(q_6, 0)) = \epsilon_CLOSURE(\{\}) = \{\}$,
 $\epsilon_CLOSURE(move(q_6, 1)) = \epsilon_CLOSURE(\{5\}) = \{5, Z\}$,
 $S' = \{q_0^*, q_1^*, q_2^*, q_3^*, q_4^*, q_5^*, q_6^*\}$;
 当 S' 中不再有未标记的状态时, 算法即可终止, 计算结果如表 1.6 所示。

表 1.6

I	I_0	I_1
$q_0(\{S, 1, 2, 3\})$	$q_1 = \{1, 3, 4, 5, Z\}$	$q_2 = \{2, 3\}$
$q_1(\{1, 3, 4, 5, Z\})$	$q_3 = \{1, 3, 4, 5, 6, Z\}$	$\{\}$
$q_2(\{2, 3\})$	$q_4 = \{4, 5, Z\}$	$q_2 = \{2, 3\}$
$q_4(\{4, 5, Z\})$	$q_6 = \{6\}$	$\{\}$
$q_3(\{1, 3, 4, 5, 6, Z\})$	$q_3 = \{1, 3, 4, 5, 6, Z\}$	$q_5 = \{5, Z\}$
$q_6(\{6\})$	$\{\}$	$q_5 = \{5, Z\}$
$q_5(\{5, Z\})$	$q_6 = \{6\}$	$\{\}$

对照题目中的表格, 可知 $T1$ 即为 $q_3 (= \{1, 3, 4, 5, 6, Z\})$, $T2$ 即为 $q_4 (= \{4, 5, Z\})$, $T3$ 即代表空集 $\{\}$ 。图 1.2 是以状态转换图表示的 DFA。

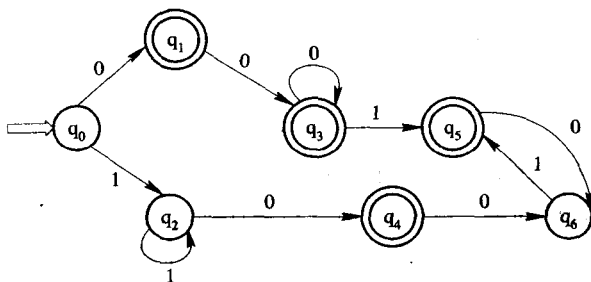


图 1.2

试题答案: (6) A (7) D (8) D (9) D

试题(10)分析

本题考查的是程序设计方面的基础知识。